Katedra informatiky Fakulta prírodných vied
Univerzita Mateja Bela, Banská Bystrica,
Slovensko

Katedra aplikované matematiky Fakulta
přírodovědně-humanitní a pedagogická,
Technická univerzita v Liberci, Česká republika

SPOJENÁ MEDZINÁRODNÁ KONFERENCIA

DidInfo&DidactIG 2017

23. ROČNÍK KONFERENCIE

DidInfo

5. ROČNÍK KONFERENCIE

DidactIG

ISBN: 978-80-557-1216-1
ISSN: 2454-051X

BANSKÁ BYSTRICA
2017

Predsedovia programového výboru/General Chairs:
doc. Ing. Jarmila Škrinárová, PhD. *Univerzita Mateja Bela, SK*
doc. PaedDr. Jiří Vaníček, PhD. *Jihočeská Univerzita, CZ*

Programový výbor konferencie/Programm Commitee:
doc. RNDr. Gabriela Andrejková, CSc. *Univerzita Pavla Jozefa Šafárika, SK*
Sonya Armstrong, PhD. *West Virginia State University, USA*
Mgr. Jan Berki, PhD. *Technická univerzita v Liberci, CZ*
Mgr. Martin Cápay, PhD. *Univerzita Konštantína Filozofa, SK*
doc. RNDr. Miroslava Černochová, CSc. *Univerzita Karlova, CZ*
prof. Dr. Valentina Dagiene *Vilnius University, LT*
Ing. Dana Horváthová, PhD. *Univerzita Mateja Bela, SK*
prof. Dr. Mirjana Ivanovic *University of Novi Sad, SR*
Ing. Jana Jacková, PhD. *Univerzita Mateja Bela, SK*
prof. RNDr. Ivan Kalaš, PhD. *Univerzita Komenského, SK*
doc. RNDr. Zuzana Kubincová, PhD. *Univerzita Komenského, SK*
doc. RNDr. Gabriela Lovászová, PhD. *Univerzita Konštantína Filozofa, SK*
Ing. Janka Majherová, PhD. *Katolícka univerzita, SK*
Ing. Božena Mannová, PhD. *České vysoké učení technické, CZ*
RNDr. Alžbeta Michalíková, PhD. *Univerzita Mateja Bela, SK*
RNDr. Pavel Pešat, PhD. *Univerzita J. E. Purkyně, CZ*
doc. RNDr. Tomáš Pitner, PhD. *Masarykova Univerzita, CZ*
Prof. Dr. Kate Sanders *Rhode Island College, USA*
prof. Ing. Veronika Stoffová, CSc. *Trnavská univerzita v Trnave, SK*
doc. RNDr. Petr Šaloun, PhD. *Technická univerzita Ostrava, CZ*
doc. RNDr. Ľubomír Šnajder, PhD. *Univerzita Pavla Jozefa Šafárika, SK*
doc. RNDr. Pavel Töpfer, CSc. *Univerzita Karlova, CZ*
doc. Ing. Ľudovít Trajteľ, PhD. *Univerzita Mateja Bela, SK*
Dr. Livia Tudor *Petroleum-Gas University, RO*

Organizačný výbor konferencie/Organizing Commitee:
Ing. Dana Horváthová, PhD.
Ing. Jana Jacková, PhD.
RNDr. Alžbeta Michalíková, PhD.
PaedDr. Ivan Brodenec, PhD.
Katarína Gavaldová
Mgr. Patrik Voštinár

Recenzenti/Reviewers:
doc. RNDr. Gabriela Andrejková, CSc. *Univerzita Pavla Jozefa Šafárika, Košice, SK*
Mgr. Jan Berki, PhD. *Technická Univerzita v Liberci, CZ*
PaedDr. Ivan Brodenec, PhD. *Univerzita Mateja Bela, Banská Bystrica, SK*
Mgr. Martin Cápay, PhD. *Univerzita Konštantína Filozofa, SK*
Ing. Dana Horváthová, PhD. *Univerzita Mateja Bela, Banská Bystrica, SK*
PaedDr. Roman Hrušecký, PhD. *Univerzita Komenského, Bratislava, SK*
RNDr. Milan Hudec, PhD. *Univerzita Mateja Bela, Banská Bystrica, SK*
Ing. Jana Jacková, PhD., *Univerzita Mateja Bela, Banská Bystrica, SK*
doc. RNDr. Ľudmila Jašková, PhD., *Univerzita Komenského, Bratislava, SK*
prof. RNDr. Ivan Kalaš, PhD. *Univerzita Komenského, SK*
Ing. Alžbeta Kanáliková, PhD. *Žilinská univerzita, Žilina, SK*
doc. Mgr. Ján Karabáš, PhD. *Univerzita Mateja Bela, Banská Bystrica, SK*
doc. RNDr. Zuzana Kubincová, PhD., *Univerzita Komenského, Bratislava, SK*
doc. RNDr. Gabriela Lovászová. PhD. *Univerzita Konštantína Filozofa, Nitra, SK*
Ing. Janka Majherová, PhD. *Katolícka Univerzita, Ružomberok, SK*
RNDr. Miroslav Melicherčík, PhD. *Univerzita Mateja Bela, Banská Bystrica, SK*
PaedDr. Viera Michaličková, PhD. *Univerzita Konštantína Filozofa, Nitra, SK*
RNDr. Alžbeta Michalíková, PhD. *Univerzita Mateja Bela, Banská Bystrica, SK*
RNDr. Pavel Pešat, PhD., *Technická Univerzita, Liberec, CZ*
doc. RNDr. Tomáš Pitner, PhD. *Masarykova Univerzita, CZ*
doc. RNDr. Ľubomír Salanci, PhD. *Univerzita Komenského, Bratislava, SK*

prof. Ing. Veronika Stoffová, CSc. *Trnavská univerzita v Trnave, SK*
doc. Ing. Jarmila Škrinárová, PhD. *Univerzita Mateja Bela, SK*
doc. RNDr. Ľubomír Šnajder, PhD., *Univerzita Pavla Jozefa Šafárika, Košice, SK*
doc. PaedDr. Monika Tomcsányiová, PhD. *Univerzita Komenského, Bratislava, SK*
doc. RNDr. Pavel Töpfer, CSc., *Univerzita Karlova, Praha, CZ*
doc. Ing. Ľudovít Trajteľ, PhD., *Univerzita Mateja Bela, Banská Bystrica, SK*
doc. PaedDr. Jiří Vaníček, PhD., *Jihočeská Univerzita, České Budějovice, CZ*
Mgr. Patrik Voštinár *Univerzita Mateja Bela, Banská Bystrica, SK*

Editori/Editors: Ing. Dana Horváthová, PhD., Ing. Jana Jacková, PhD.,
RNDr. Alžbeta Michalíková, PhD., doc. Ing. Jarmila Škrinárová, PhD.
Copyright © 2017 autori/authors
ISBN: 978-80-557-1216-1
ISSN: 2454-051X
Recenzovaný zborník/Reviewed Proceedings

Predhovor

Koleso času sa krúti neuveriteľne rýchlo. Opäť je tu konferencia Didinfo, už jej 23. ročník. Tentokrát obohatená o českú konferenciu DidactIG a jej 5. ročník. Vzhľadom na niekoľkoročnú spoluprácu a na spoločné témy oboch konferencií sme sa rozhodli kráčať spolu.

Nachádzame sa v ére prudkého vývoja nových informačných a komunikačných, ale najmä výpočtových technológií. Podieľame sa na vývoji, používame a profitujeme z technológií a čoraz častejšie používame pojmy ako sú cloud, internet vecí, analýza obrovských a mnohorozmerných údajov, inteligentné mestá, vysokovýkonné a extrémne počítanie a mobilné technológie a aplikácie. Tieto pojmy zdanlivo spolu nesúvisia. Pri hlbšej analýze zistíme, že sú to vzájomne sa dopĺňajúce súčasti komplexného informačného, komunikačného a výpočtového systému, v ktorom množstvo procesov beží súbežne.

V tomto kontexte si kladieme otázky: Aké si máme stanovovať nové vzdelávacie ciele v informatických predmetoch, na všetkých stupňoch škôl, aby sme dokázali postupne, primerane veku žiakov, sprístupňovať a objavovať znalosti a tiež súvislosti medzi mini? Aké máme zvoliť nové postupy a metódy, ako precvičovať a spolu so žiakmi a študentami tvoriť? Ako ich pripraviť na súčasné moderné technológie, ale najmä na riešenie úloh, aj tých, ktoré ešte len prídu? Na aké nové výzvy sa máme sústrediť pri príprave budúcich učiteľov informatiky?

Odpovede na tieto a mnohé ďalšie otázky hľadáme, k niektorým sa približujeme a iné nachádzame, spolu s autormi príspevkov. Práce sú orientované najmä na výskum v oblasti vyučovania informatiky. Tradične sa viaceré práce autorov zaoberajú rôznymi metódami vyučovania algoritmizácie a programovania. V tomto roku ide najmä o objektovo – orientované programovanie. Keďže najväčším nepriateľom poznania nie je nevedomosť, ale ilúzia poznania, je potrebné identifikovať chyby v procese učenia. Dôležité a pre žiakov motivačné je tiež neformálne vzdelávanie, na báze rôznych súťaží. Cielená príprava súťažných úloh je kľúčovým prvkom k nadobudnutiu a rozvíjaniu výpočtového myslenia. Výpočtové myslenie, ako proces riešenia problémov, ktorý obsahuje metódy riešenia problémov a všeobecné intelektuálne postupy možno rozvíjať pomocou didaktických hier, vývoja mobilných aplikácií, riešenia robotických úloh, cloudových technológií a ďalších vzdelávacích aktivít. V školskej praxi je čoraz viac potrebné venovať pozornosť a preventívne pôsobiť voči počítačovej kriminalite. Preto je potrebné poznať potenciálne hrozby a účinné metódy ochrany. Dôležité sú aj aspekty, resp. legislatívne rámce medzinárodného práva v tejto oblasti.

Informatika je v súčasnosti integrovaná vo všetkých odboroch a profesiách. Máme to šťastie, že absolventi odboru informatika sú žiadaní na trhu práce a byť užitoční je nesmierne motivujúce. Každý komplexný systém, preto aj systém vyučovania informatiky na všetkých stupňoch škôl, sa skladá z veľkého počtu heterogénnych častí, ktoré majú medzi sebou množstvo väzieb a navzájom sa ovplyvňujú zložitým spôsobom. Autori príspevkov vo svojich prácach, ako určitých heterogénnych častiach tohto komplexného systému, ukázali s akou radosťou, motivačne a efektívne navrhujú, pripravujú a realizujú rôzne prvky informatického vzdelávania.

Želáme konferencii naplnenie stanovených cieľov, aktívnym účastníkom úspech a množstvo inšpirácie do ďalšieho roka!

Marec 2017

doc. Ing. Jarmila Škrinárová, PhD.
doc. PaedDr. Jiří Vaníček, PhD.

predsedovia programového výboru
konferencie DidInfo & DidactIG 2017

Obsah

Príspevky k pozvaným prednáškam

Ivan Bacigál, Tatiana Hajduková
Vyučovanie informatiky ako súčasť prevencie počítačovej kriminality....8
Ivan Kalaš
ScratchMaths: vzdelávací obsah a princípy tvorby....16

Príspevky

Brigita Albertova, Peter Nedorost, Jaroslav Kultan
Priestorová orientácia a orientácia v PC....26
Andrej Blaho, Ľubomír Salanci
Nová maturita z informatiky....32
Lucia Budinská, Monika Tomcsányiová
Abstraktné dátové typy v úlohách súťaže iBobor v kategórii Benjamíni....36
Iveta Csicsolová, Zuzana Kubincová
Wiki na strednej škole....42
Martin Cápay
Objavovanie informatických konceptov zážitkovými aktivitami....48
Rostislav Fotjík
Výuka programování pomocí robota Ozobot....54
Ján Guniš, Ľubomír Šnajder
Súťaž PALMA junior a programovanie v jazyku Python....59
Roman Hrušecký, Monika Tomcsányiová
Vývoj edukačných webových aplikácií pomocou knižnice SPRITES....64
Soňa Ivančáková, Nika Klimová, Dana Horváthová
Ako zmysluplne využiť jednu Hodinu v škole....68
Ľudmila Jašková, Mária Kaliaková
Záverečný projekt z informatiky pre zrakovo postihnutých žiakov ZŠ – skúsenosti s autentickým učením....72
Ľudmila Jašková, Mária Stankovičová
Učebné materiály pre výučbu informatiky nevidiacich žiakov základnej školy....77
Nika Klimová, Dana Horváthová
Súťaž KODU CUP ako podporný prostriedok algoritmického myslenia....83
Natália Kováčová
Koncept opakovania vo vyučovaní programovania nevidiacich žiakov základnej školy....88
Václav Králík, Patrik Sitiarik
Laboratórny model konečného automatu....93
Gabriela Lovaszova, Martin Cápay, Viera Michaličková
Informatika s tabletmi....96
Janka Majherová, Hedviga Petrušková, Pavel Valuška
Interaktívne učebné materiály vo vyučovaní algoritmizácie....101
Karolína Mayerova, Michaela Veselovská, Monika Tomcsányiová, Lucia Budinská
Analýza riešení úlohy zo súťaže iBobor pre kategóriu Juniori 2016/2017....105
Eva Mészárosová
Vývoj metodického materiálu pre vyučovanie základov programovania v jazyku Python....111

Marek Nagy
Rozvíjanie čitateľskej gramotnosti prostredníctvom detského programovacieho jazyka Scratch.....117
Ingrid Nagyová
Výuka principů objektově orientovaného programování při tvorbě her.....122
Václav Nádvorník
Výuka informatiky pomocí videonávodů.....128
Peter Potocký
Didaktika vo výučbe programovania.....131
Mária Spišáková, Ľubomír Salanci
Chyby ako súčasť motivácie programovania.....136
Veronika Stoffová
Úspešnosť študentov učiteľstva informatiky.....141
Václav Šimandl, Václav Dobiáš
Využití zkušenostního učení při výuce e-bezpečnosti.....147
Monika Tomcsányiová
Jazykovo závislé úlohy súťaže iBobor.....153
Martin Vašek
Aplikace pro výuku základních principů objektově orientovaného programování.....158
Michaela Veselovská, Karolína Mayerova
Aktivity s LEGO WeDo vo vyučovaní informatiky.....163
Patrik Voštinár
Používateľské rozhranie – dôležitý prvok pri vyučovaní programovania.....169
Michal Winczer, Ľudmila Jašková, Eva Mészárosová
Skúsenosti z úvodného kurzu programovania v Pythone na gymnáziu.....174
Michal Winczer, Monika Tomcsányiová, Peter Tomcsanyi
Programovanie interaktívnych úloh súťaže iBobor pomocou knižnice sprites.js.....179

Krátke príspevky

Dana Horváthová, Vladimír Siládi
Virtuálna realita a jej využitie v liečbe fóbií.....185
Jana Jackova, Liviana Nicoleta Tudor
Vyučovanie úvodných kurzov databázových systémov a medzinárodná spolupráca.....187
Jaroslav Kultan, Aigul Sadvakassova, Meruert Serik, Mira Rakhimzhanova
Cloud Technologies and the Taxonomy of Bloom.....189
Miroslav Melicherčík, Alžbeta Michalíková, Michal Vagač
Zapojenie študentov do riešenia projektu APVV s názvom Automatické spracovávanie trasologických objektov.....191
Eva Mészárosová, Monika Tomcsányiová
Sekcia Edu Track na konferencii PyCon SK 2017.....193
Vladimír Siládi, Ján Astaloš
Teaching Grid Technologies and Grid Computing to Undergraduate Students of Computer Science.....195
Jarmila Škrinarova, Nika Klimová
Výpočtové myslenie.....197
Jarmila Škrinárová
Podpora mladých talentov ako kľúčový prvok rozvoja vysokovýkonného počítania.....199
Patrik Voštinár
Vývoj mobilných aplikácií na Katedre informatiky FPV UMB.....201

Darina Výbohová, Vladimír Laššák, Jozef Ďurian
Zvyšovanie profesionality pedagogických zamestnancov v oblasti didaktických digitálnych kompetencií.....203

Workshopy

Organizátori

Sponzori a mediálni partneri

Príspevky k pozvaným prednáškam

Invited Papers

Vyučovanie informatiky ako súčasť prevencie počítačovej kriminality

Informatics education as a part of cybercrime prevention

Ivan Bacigál
Prezídium Policajného zboru MV SR
Pribinova 2
812 72 Bratislava
Slovenská republika
ivan.bacigal@minv.sk

Tatiana Hajdúková
Katedra informatiky a manažmentu
Sklabinská 1
835 17 Bratislava
Slovenská republika
tatiana.hajdukova@minv.sk

ABSTRACT

One of the main goals achieved through teaching computer science and information and communication technology education, is a task to gain practical knowledge which allows for effective work with information using computer-assisted techniques. Side by side with education, which starts from tender age, goes learning of how to use this newly acquired knowledge according to social conventions and social development. It is required that compulsory education of information and communication technology leads to understanding of potential risks of using modern technology and necessity of protecting oneself and their privacy from threats, especially among young children. Study describes current trends of cyber criminality, which negatively affects functioning of society and lives of individuals. The study also contains information concerning development and current state of International Cyber Law Enforcement, which is dictated by the International Convention on Cybercrime (Budapest Convention on Cybercrime).

Keywords

Cyber crime. Internet. Information technologies. Anonymity. Digital reality.

ABSTRAKT

Medzi hlavné ciele vyučovania informatiky a informatickej výchovy na základných školách patrí nadobúdanie praktických zručností umožňujúcich efektívnu prácu s informáciami s podporou výpočtovej techniky. Na strednej škole sa informatická gramotnosť žiakov rozvíja o teoretické a technické poznatky súvisiace s vývojom technológií, ktoré si moderná doba vyžaduje. Súčasne s výchovou a vzdelávaním od najútlejšieho veku však musí prebiehať aj predávanie informácií, ako tieto nadobudnuté znalosti a zručnosti využívať v súlade so spoločenskými pravidlami a pre spoločenský rozvoj. Je potrebné, aby formálna výučba v rámci vzdelávacej oblasti IKT viedla ku správnym návykom a uvedomeniu si potenciálnych rizík a potrieb chrániť seba a svoje súkromie pred nebezpečenstvami, ktoré použitie moderných technológií umožňuje. Príspevok, ktorý vnikol v rámci riešenia vedecko-výskumnej úlohy výsk. 161 Metódy spracovania policajne relevantných informácií na Akadémii Policajného zboru v Bratislave, popisuje súčasné trendy zneužívania informačných technológií, ktoré majú negatívne vplyvy na fungovanie spoločnosti a životy jednotlivcov a aktuálne trendy v oblasti počítačovej kriminality. Obsahom príspevku je aj vývoj a aktuálny stav medzinárodného práva v oblasti boja proti počítačovej kriminalite v zmysle medzinárodného Dohovoru o počítačovej kriminalite (Budapeštianskeho dohovoru).

Kľúčové slová

Počítačová kriminalita. Internet. Informačné technológie. Anonymita. Digitálna realita.

1. ÚVOD

Počítač je zariadenie slúžiace na realizáciu výpočtov alebo riadenie operácií, ktoré sú vyjadriteľné číselnými alebo logickými výrazmi. Vzhľadom na prudký vývoj v oblasti informačných technológií sú počítače stále viac implementované do rôznych inteligentných systémov využívaných v domácnostiach, motorizovaných dopravných prostriedkoch, riadení všetkých druhov dopravy, pri prevádzke budov a mnohých ďalších, pretože prispievajú k ich automatizácii, spoľahlivosti, presnosti a zvyšujú bezpečnosť a komfort. Objavovanie nových poznatkov alebo skutočností s pridanou hodnotou obvykle znamená preveriť veľké množstvo kombinácií známych informácií, ktoré počítač sprostredkováva a veľmi urýchľuje. Progres vývoja počítačov je tak prudký, že tvorba a aktualizácia súvisiacich právnych noriem často nestíha reagovať na nové technológie a súvisiace hrozby. Podľa Dohovoru o počítačovej kriminalite zverejneného v Zbierke zákonov pod č. 137/2008 je počítačový systém, zariadenie, alebo skupina vzájomne prepojených alebo súvisiacich zariadení, z ktorých jedno zariadenie, alebo viaceré zariadenia vykonávajú automatizované spracúvanie údajov na základe programu. Bez ohľadu na kritérium klasifikácie počítačov, vo všetkých prípadoch zostáva ich charakterizujúcou vlastnosťou programovateľnosť. V dnešnej dobe už poznáme aj samoučiace sa systémy, ktoré smerujú k vývoju umelej inteligencie. Programovateľnosť samoučiacich systémov je len vlastnosť podmieňujúca ich hlavnú funkciu a to schopnosť učiť sa. Používanie a rozvoj informačných technológií neustále napreduje, čo okrem pozitívnych zmien v prospech zbližovania a rozvoja spoločnosti prináša aj možnosti zneužívania ich funkcií na páchanie kriminality sofistikovanejšími spôsobmi. Súčasné problémy sveta spojené so zavádzaním nových technológií do života spoločnosti s možnosťami ich zneužívania proti obyvateľom, ako aj mnoho ďalších a ďalších udalostí, procesov a javov vo svete nevytvárajú bezpečné prostredie pre život ľudí, pre sociálnu sféru[1, s. 45]. Pre veľkú rôznorodosť informačných technológií formy páchania počítačovej kriminality nie je možné

definovať krátkym textom, ktorý by ju súčasne vystihoval v plnom význame.

2. POČÍTAČOVÁ KRIMINALITA

Pri páchaní počítačovej kriminality sa často vyskytuje záujem o obsah počítača, t. j. o údaje v ňom uložené. Krádež nehmotných elektronických údajov zvykne byť spozorovaná s časovým oneskorením, alebo dokonca nemusí byť spozorovaná vôbec, pretože po vytvorení elektronickej kópie nič nezmizne. Počítačové siete a elektronické informácie môžu byť zneužité ako prostriedok na páchanie trestnej činnosti a stopy o tejto činnosti sa môžu uchovávať na akomkoľvek mieste počítačovej infraštruktúry obsahujúcej pamäťové médium. V mnohých prípadoch jedine expertná znalosť technických možností zariadení ako aj spôsobov použitia týchto technológií v sieťach umožňuje odhaliť trestnú činnosť. Odhaľovanie a objasňovanie sa povahou svojho predmetu a svojím zázemím, v konkrétnom systéme policajného konania, podstatne odlišujú od iných policajno-bezpečnostných činností. [2, s. 14]. Nedodržanie správnych postupov môže mať za následok nenávratné znehodnotenie dôkazov potrebných na usvedčenie páchateľa a jeho prepustenie. [3, s. 222]. Navyše ani samotné zistenie nežiadúceho protispoločenského správania nemusí stačiť na potrestanie jeho páchateľov, nakoľko platná legislatíva musí byť aktuálne schopná reflektovať a akceptovať relevantnosť a informačnú hodnotu predložených digitálnych stôp. Odhaľovanie, dokumentovanie a vyšetrovanie počítačovej trestnej činnosti je postavené na štandardizovaných postupoch, ktoré aby mohli byť akceptované v trestnom konaní, musia spĺňať definované kritéria zákonnosti. Ak má mať právny systém schopnosť eliminovať a postihovať nové formy nežiadúceho protispoločenského konania, musí zmenou legislatívy dynamicky reagovať na novovznikajúce formy kriminality. Jedným z problémov je koordinované prijímanie zákonov na ochranu kybernetickej bezpečnosti, nakoľko zákony sa prijímajú na národnej úrovni v každej krajine samostatne, ale vďaka internetu páchatelia počítačovej kriminality pôsobia globálne na celom svete. [4, s. 369]. V tomto prípade máme na mysli najmä elektronické komunikačné služby, na ktorých je vyspelá spoločnosť súčasnosti vybudovaná, pretože nie sú limitované vzdialenosťou ani dlhými časovými stratami pri prenose signálu.

Rozsah problematiky počítačovej kriminality presahuje možnosti tohto príspevku, preto pohľad na ňu zúžime na oblasť dotýkajúcu sa výchovy detí a mládeže. Prevencia počítačovej kriminality a inej protispoločenskej činnosti nespočíva len vo vzdelávaní mládeže o potenciály využitia technológií vo svoj prospech, ale v cieľavedomom poskytovaní informácií o možnostiach ich zneužitia. Prostredníctvom Internetu je možné rozširovať škodlivú aktivitu nielen zo slovenských domén a IP adries, ale celého zahraničného kybernetického priestoru. Meniace sa medzinárodné, ale aj európske bezpečnostné prostredie si vyžaduje zdokonaľovanie charakteru prípravných opatrení k ochrane obyvateľstva. [5, s. 111] Nezávisí odkiaľ hrozba prichádza, v každom prípade je potrebné naučiť sa brániť. Prevencia na školách by mala spočívať aj v poskytovaní konkrétnych informácií, vrátane trestnej zodpovednosti za svoje konanie. V prípade adolescentov sa prejavuje ich nedostatočné právne povedomie, pretože si neuvedomujú dôsledky svojho konania, respektíve často ho nesprávne považujú za formu zábavy a spôsob verejného prezentovania, ktoré môže byť v skutočnosti protizákonné. Pekným príkladom môže byť komunikácia na sociálnych sieťach, ktorú považujú mnohí, nielen mladí, za anonymnú. Širokej verejnosti používateľov informačných technológií je potrebné poskytovať rady, ako sa nestať obeťou trestného činu, respektíve ako sa brániť pred počítačovými útokmi legitímnymi prostriedkami, nakoľko represívne prostriedky sú obvykle nasadzované v situáciách, pri ktorých už reálne nastali negatívne následky protizákonného konania.

2.1 Digitálna realita verzus skutočnosť

Tak ako už bolo naznačené v úvodnej časti, jednou z dôležitých úloh učiteľov súčasnosti je nielen skvalitniť vzdelávanie a výchovu detí a mládeže využívaním moderných technológií, ale aj učiť ich rozlišovať realitu od počítačového sveta. Mladí ľudia môžu moderné technológie využívať pre obohatenie zážitkov a zjednodušenie svojho života, alebo sa môžu modernými technológiami nechať spútať alebo ich dokonca zneužívať na páchanie trestnej činnosti. Minulý rok boli zverejnené informácie, že istý poskytovateľ služieb na internete svojich používateľov nepovažuje za klientov využívajúcich svoje služby, ale za tovar, s ktorým môže obchodovať (najmä za účelom poskytovania marketingových informácií používateľom). Túto filozofiu diktovania pravidiel zo vzťahu poskytovateľ služieb a používateľ je cítiť čoraz viac.

Ďalším negatívnym faktorom je, že mladí ľudia každodenne ovplyvnení informáciami (je jedno či v audiovizuálnej podobe alebo v podobe počítačových hier) si nedostatočne uvedomujú rozdiel medzi skutočným a digitálne predstieraným prostredím. Kým sa dôsledky konania hráčov v digitálnom svete a digitálnej komunite môžu zdať „cool“[1] alebo bez dôsledkov, v reálnom svete podobné konanie môže skončiť tragicky. Ale platí to aj naopak. Pocit nesmrteľnosti prežitý v digitálnom svete sa transformuje v reálnom svete do „ľahkovážneho správania“, čím sa deti stávajú obeťou v podstate dvakrát. Jedenkrát ako dôsledok konania v reálnom svete a druhýkrát ako reálna bytosť oklamaná virtuálnym svetom, v ktorom sa jej nemôže nič bolestivé stať (sled udalostí sa nedá stlačením klávesy spustiť odznovu). Alarmujúce je, že v súčasnosti vyššie uvedené neplatí iba o mladej generácii, ale čoraz viac postihuje celé spektrum ľudskej spoločnosti.

2.2 Počítačová kriminalita ako dôsledok technologického pokroku

Ako reakcia na nárast protispoločenského konania spojeného s rozvojom počítačových a komunikačných technológii, nadnárodný charakter tejto trestnej činnosti a aplikačné problémy v oblasti vymožiteľnosti práva bol 23. novembra 2001 v Budapešti otvorený na podpis medzinárodný Dohovor o počítačovej kriminalite (ďalej len dohovor). Národná rada Slovenskej republiky s dohovorom vyslovila súhlas svojím uznesením č. 583 z 23. októbra 2007, prezident Slovenskej republiky dohovor ratifikoval 12. decembra 2007. Dohovor nadobudol platnosť 1. júla 2004 v súlade s článkom 36 ods. 3[2] a 1. mája 2008 v Slovenskej republike v súlade s článkom 36 ods. 4.

[1] anglický slangový výraz – vtipný, dômyselný, úžasný

[2] Tento dohovor nadobudne platnosť prvý deň mesiaca nasledujúceho po uplynutí troch mesiacov odo dňa, keď päť

Dohovor zaviazal signatárske krajiny vrátane Slovenska definovať nasledovné protispoločenské konania ako trestné činy:

„Nezákonný prístup„ – neoprávnený prístup do počítačového systému ako celku alebo do jeho časti spáchaný porušením bezpečnostných opatrení s úmyslom získať počítačové údaje alebo s iným nečestným úmyslom, alebo vo vzťahu k počítačovému systému prepojenému s iným počítačovým systémom.

„Nezákonné zachytenie údajov“ – zachytávanie neverejných prenosov počítačových údajov do počítačového systému, z neho alebo v rámci tohto systému vrátane elektromagnetických emisií z počítačového systému, ktorý obsahuje také počítačové údaje vykonané technickými prostriedkami.

„Zasahovanie do údajov“ – neoprávnené poškodenie, vymazanie, zhoršenie kvality, pozmenenie počítačových údajov alebo zamedzenie prístupu.

„Zasahovanie do systému“ – neoprávnené závažné marenie funkčnosti počítačového systému vkladaním, prenášaním, poškodením, vymazaním, zhoršením, pozmenením počítačových údajov alebo zamedzením prístupu k nim.

„Zneužitie zariadení“ čím rozumieme:

a) výroba, predaj, obstarávanie na účely použitia, dovoz, distribúciu alebo iné sprístupnenie:

- zariadenia vrátane počítačového programu vytvoreného alebo upraveného predovšetkým s cieľom spáchať niektorý z vymedzených trestných činov,
- počítačového hesla, prístupového kódu alebo podobných údajov, ktorých pomocou je možný prístup do počítačového systému ako celku alebo do niektorej jeho časti, s úmyslom ich použiť na spáchanie niektorého z vymedzených trestných činov,
- držba veci uvedenej v odseku 1 písm. a) bode i. alebo s úmyslom ju použiť na spáchanie niektorého z trestných činov vymedzených v článkoch 2 až 5.

„Falšovanie počítačových údajov“ – vloženie, pozmenenie, vymazanie počítačových údajov alebo zamedzenie prístupu k nim, v ktorých dôsledku stratia údaje autentickosť, s úmyslom považovať ich za autentické alebo aby sa na základe nich ako autentických údajov konalo, na právne účely, bez ohľadu na to, či tieto údaje sú alebo nie sú priamo čitateľné alebo zrozumiteľné

„Počítačový podvod“ – spôsobenie majetkovej ujmy inému:

- vložením, pozmenením, vymazaním počítačových údajov alebo zamedzením prístupu k nim,
- zásahom do fungovania počítačového systému s podvodným alebo nečestným úmyslom neoprávnene získať pre seba alebo pre iného majetkový prospech.

„Trestné činy týkajúce sa detskej pornografie“. Detská pornografia“ zahŕňa pornografický materiál, ktorý zobrazuje:

a) maloletú osobu zúčastnenú na zjavnom sexuálnom správaní,
b) osobu, ktorá sa zdá byť maloletá a ktorá sa zúčastňuje na zjavnom sexuálnom správaní,
c) realistické zobrazenia maloletej osoby zúčastnenej na zjavnom sexuálnom správaní.

štátov vrátane najmenej troch členských štátov Rady Európy vyjadrí súhlas byť viazané dohovorom v súlade s ustanoveniami odsekov 1 a 2.

„Trestné činy týkajúce sa porušenia autorských a príbuzných práv“ – porušenie autorského práva vymedzeného právnym poriadkom v súlade so záväzkami, ktoré prijali signatárske štáty podľa Parížskeho aktu z 24. júla 1971, ktorým sa mení Bernský dohovor o ochrane literárnych a umeleckých diel; Dohody o obchodných aspektoch práv na duševné vlastníctvo a Zmluvy Svetovej organizácie duševného vlastníctva (WIPO) o autorskom práve, okrem osobnostných práv priznaných týmito dohovormi a v súlade so záväzkami, ktoré prijala podľa Medzinárodného dohovoru o ochrane výkonných umelcov, výrobcov zvukových záznamov a rozhlasových organizácií (Rímsky dohovor); Dohody o obchodných aspektoch práv na duševné vlastníctvo a Zmluvy Svetovej organizácie duševného vlastníctva (WIPO) o výkonoch a zvukových záznamoch, okrem osobnostných práv priznaných týmito dohovorom.

Okrem definície konkrétneho protiprávneho konania dohovor definoval aj spôsob spolupráce v trestnom konaní, akým sú napríklad vzájomná právna pomoc týkajúca sa prístupu k uloženým počítačovým údajom, postup pri výmene digitálnych stôp, či zabezpečenie vytvorenia medzinárodnej siete kontaktných miest, ktoré sú v signatárskych krajinách k dispozícii 24 hodín 7 dní v týždni, za účelom zabezpečenia poskytovania okamžitej pomoci za účelom vyšetrovania alebo konania v prípade trestných činov spojených s počítačovými systémami a dátami, alebo za účelom zhromažďovania dôkazov o trestnom čine v elektronickej forme. Účelom Dohovoru je podporiť efektívnu medzinárodnú spoluprácu, zlepšiť globálne podmienky pre dôveru a bezpečnosť medzi členskými krajinami, nakoľko v online prostredí nemajú opodstatnenosť hraničné kontroly [6, s. 59].

2.3 Trestno-právne dôsledky páchania počítačovej trestnej činnosti

V slovenskom právnom systéme boli požiadavky a záväzky dohovoru implementované hlavne do Trestného zákona , kde v zmluve uvedené protiprávne konania nájdeme implementované naprieč celým zákonom. Počítačová trestná činnosť je už v súčasnosti stanovená širšie ako bolo definované v požiadavkách dohovoru z roku 2001. Počítačovú kriminalitu dnes rozdeľujeme na počítačovú kriminalitu (computer crime) v užšom zmysle a počítačovú kriminalitu v širšom zmysle, kam býva zaradená aj kriminalita súvisiaca s počítačovými technológiami tzv. computer related crime. Medzi počítačovú kriminalitu dnes zaraďujeme najmä nasledovné:

§ 201a Sexuálne zneužívanie
Trestný čin spácha ten, kto prostredníctvom elektronickej komunikačnej služby navrhne dieťaťu mladšiemu ako pätnásť rokov osobné stretnutie v úmysle spáchať na ňom trestný čin sexuálneho zneužívania alebo trestný čin výroby detskej pornografie, pričom sám nie je dieťaťom.

§ 219 Neoprávnené vyrobenie a používanie platobného prostriedku, elektronických peňazí alebo inej platobnej karty

Trestný čin spácha ten, kto neoprávnene vyrobí, pozmení, napodobní, falšuje alebo si obstará platobný prostriedok alebo elektronické peniaze alebo inú platobnú kartu vrátane telefónnej

karty alebo predmet spôsobilý plniť takú funkciu na účel použiť ho ako pravý alebo na taký účel ho prechováva, prepravuje, použije alebo poskytne inému a ten, kto neoprávnene vyrobí, prechováva, obstará si alebo inak zadováži alebo poskytne inému nástroj, počítačový program alebo iný prostriedok špeciálne prispôsobený na spáchanie činu tohto trestného činu.

§ 226 Neoprávnené obohatenie
Trestný čin spácha ten, kto na škodu cudzieho majetku seba alebo iného obohatí tým, že neoprávneným zásahom do technického alebo programového vybavenia počítača, automatu alebo iného podobného prístroja alebo technického zariadenia slúžiaceho na automatizované uskutočňovanie predaja tovaru, zmenu alebo výber peňazí alebo na poskytovanie platených výkonov, služieb, informácií či iných plnení dosiahne, že tovar, služby alebo informácie získa bez požadovanej úhrady alebo peniaze získa neoprávnene, a spôsobí tým na cudzom majetku škodu.

§ 247 Neoprávnený prístup do počítačového systému
Trestný čin spácha ten, kto prekoná bezpečnostné opatrenie, a tým získa neoprávnený prístup do počítačového systému alebo jeho časť.

§ 247a Neoprávnený zásah do počítačového systému
Trestný čin spácha ten, kto obmedzí alebo preruší fungovanie počítačového systému alebo jeho časti neoprávneným vkladaním, prenášaním, poškodením, vymazaním, zhoršením kvality, pozmenením, potlačením alebo zneprístupnením počítačových údajov, alebo tým, že urobí neoprávnený zásah do technického alebo programového vybavenia počítača a získané informácie neoprávnene zničí, poškodí, vymaže, pozmení alebo zníži ich kvalitu.

§ 247b Neoprávnený zásah do počítačového údaja
Trestný čin spácha ten, kto úmyselne poškodí, vymaže, pozmení, potlačí alebo zneprístupní počítačové údaje alebo zhorší ich kvalitu v rámci počítačového systému alebo jeho časti.

§ 247c Neoprávnené zachytávanie počítačových údajov
Trestný čin spácha ten, kto neoprávnene zachytáva počítačové údaje prostredníctvom technických prostriedkov neverejných prenosov počítačových údajov do počítačového systému, z neho alebo v jeho rámci vrátane elektromagnetických emisií z počítačového systému, ktorý obsahuje takéto počítačové údaje alebo kto ako zamestnanec poskytovateľa elektronickej komunikačnej služby spácha tento čin alebo inému úmyselne umožní spáchať taký čin, alebo pozmení alebo potlačí správu podanú prostredníctvom elektronickej komunikačnej služby.

§ 247d Výroba a držba prístupového zariadenia, hesla do počítačového systému alebo iných údajov

Trestný čin spácha ten, kto v úmysle spáchať trestný čin neoprávneného prístupu do počítačového systému podľa § 247, neoprávneného zásahu do počítačového systému podľa § 247a, neoprávneného zásahu do počítačového údaja podľa § 247b alebo neoprávneného zachytávania počítačových údajov podľa § 247c vyrobí, dovezie, obstará, kúpi, predá, vymení, uvedie do obehu alebo akokoľvek sprístupní zariadenie vrátane počítačového programu vytvorené na neoprávnený prístup do počítačového systému alebo jeho časti, alebo počítačové heslo, prístupový kód alebo podobné údaje umožňujúce prístup do počítačového systému alebo jeho časti.

§ 283 Porušovanie autorského práva
Trestný čin spácha ten, kto neoprávnene zasiahne do zákonom chránených práv k dielu, umeleckému výkonu, zvukovému záznamu alebo zvukovo-obrazovému záznamu, rozhlasovému vysielaniu alebo televíznemu vysielaniu alebo databáze.

§ 368 Výroba detskej pornografie
Trestný čin spácha ten, kto využije, získa, ponúkne alebo inak zneužije dieťa na výrobu detskej pornografie alebo detského pornografického predstavenia alebo umožní také jeho zneužitie, alebo sa inak podieľa na takejto výrobe, potrestá sa odňatím slobody na štyri roky až desať rokov.

§ 369 Rozširovanie detskej pornografie
Trestný čin spácha ten, kto rozmnožuje, prepravuje, zadovažuje, sprístupňuje alebo inak rozširuje detskú pornografiu.

§ 370 Prechovávanie detskej pornografie a účasť na detskom pornografickom predstavení
Trestný čin spácha ten, kto prechováva detskú pornografiu alebo kto koná v úmysle získať prístup k detskej pornografii prostredníctvom elektronickej komunikačnej služby alebo sa úmyselne zúčastní detského pornografického predstavenia.

§ 376 Poškodzovanie cudzích práv
Trestný čin spácha ten kto neoprávnene poruší tajomstvo listiny alebo inej písomnosti, zvukového záznamu, obrazového záznamu alebo iného záznamu, počítačových dát alebo iného dokumentu uchovávaného v súkromí iného tým, že ich zverejní alebo sprístupní tretej osobe alebo iným spôsobom použije a inému tým spôsobí vážnu ujmu na právach.

2.4 Vývoj počítačovej trestnej činnosti

Okrem Dohovoru o počítačovej kriminalite boli pod vplyvom ďalšieho vývoja počítačovej kriminality, a to najmä znepokojujúcich rozmerov vývoja na vnútroštátnej aj medzinárodnej úrovni v oblasti sexuálneho vykorisťovania a sexuálneho zneužívania detí (vrátane trestnej činnosti súvisiacej s detskou pornografiou) prijaté aj ďalšie medzinárodné právne normy. Patria k nim napríklad smernica Európskeho parlamentu a rady 2011/92/EÚ z 13. decembra 2011 o boji proti sexuálnemu zneužívaniu a sexuálnemu vykorisťovaniu detí a proti detskej pornografii a dohovor Rady Európy o ochrane detí pred sexuálnym vykorisťovaním a sexuálnym zneužívaním v Lanzarote 25. 10. 2007, ktorý pre Slovenskú republiku nadobudol platnosť len 1. júla 2016. Hlavným účelom ich prijatia bolo zintenzívnenie vnútroštátneho a medzinárodného boja proti sexuálnemu zneužívaniu a sexuálnemu vykorisťovaniu detí a proti detskej pornografii.

Potrebné je zdôrazniť, že tieto činy majú zničujúci vplyv na zdravie a psychosociálny vývoj detí a pokiaľ sa tento vývoj nepodarí zastaviť a zvrátiť, tento problém môže mať vážny vplyv aj na vývoj celej spoločnosti.

V oblasti počítačových útokov bola prijatá smernice Európskeho parlamentu a Rady Európy 2013/40/EÚ z 12. augusta 2013 o útokoch na informačné systémy. Smernica zohľadnila nové metódy páchania počítačových trestných činov a významne rozšírila komplex protiprávnych konaní v súvislosti s neoprávneným prístupom do počítačových systémov, neoprávneným zásahom do počítačových systémov a počítačových údajov, neoprávneným zachytávaním počítačových údajov a manipuláciou s nástrojmi určenými na spáchanie týchto trestných

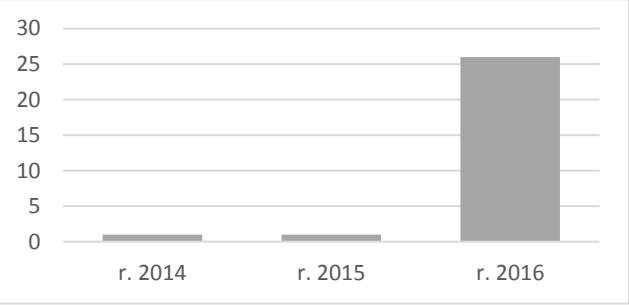
činov. Na tento účel boli vytvorené nové individuálne skutkové podstaty trestných činov definujúce jednotlivé typy nových foriem protiprávnych konaní[3].

2.5 Aktuálny stav registrovanej počítačovej kriminality v slovenskom kybernetic-priestore

Na Slovensku už vyše 30 rokov funguje Evidenčno-štatistický systém kriminality (EŠSK), ktorý je využívaný v policajnej praxi na umožnenie sledovania miery registrovanej trestnej činnosti v minulosti na území Československa, po osamostatnení len na území Slovenska. V EŠSK sa sústreďujú oznámenia o podozrení zo spáchania trestných činov, údaje o trestných činoch a údaje o známych páchateľoch. Od jeho zavedenia do praxe až do súčasnej doby bola dodržiavaná zásada, aby do systému boli zavádzané údaje len na základe procesných úkonov, ktoré sú zdrojom dát a podliehajú posudzovaniu orgánmi prokuratúry. Výstupy z EŠSK sa prostredníctvom Štatistického úradu SR poskytujú najvyšším štátnym orgánom a prostredníctvom INTERPOLU aj do zahraničia.

Na základe evidencie EŠSK za posledné tri roky je viditeľný výrazný nárast zaznamenaný v roku 2016 v počte neoprávnených zásahov do technického alebo programového vybavenie počítača podľa § 226 Neoprávnené obohatenie, zobrazené na grafe 1[4].

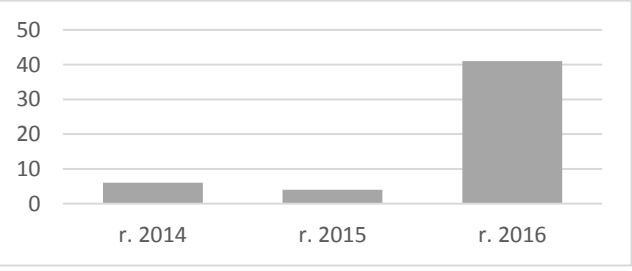
Graf 1 § 226 Neoprávnené obohatenie v SR, 2014-2016



Zdroj: Spracované podľa údajov EŠSK

Podobný vzostup za rok 2016 možno vidieť aj v prípade § 283 Porušovanie autorského práva na grafe 2, pričom objasnenosť týchto trestných činov nedosahovala ani 20%.

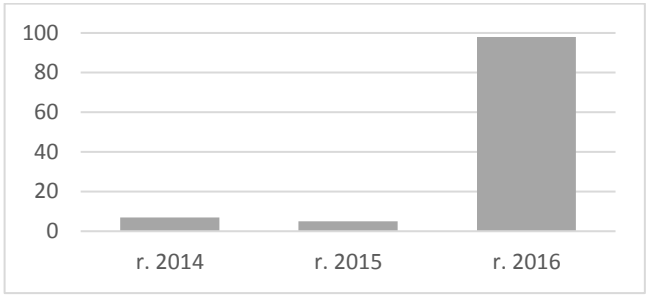
Graf 2 § 283 Porušovanie autorského práva v SR, 2014-2016



Zdroj: Spracované podľa údajov EŠSK

V takmer 100 prípadov počas roka 2016 bolo zaregistrovaných a vyšetrovaných podľa § 369 Rozširovanie detskej pornografie. Nárast evidovaných prípadov je podobne dramaticky zvýšený ako pri vyššie spomenutej počítačovej kriminalite, avšak s podstatne vyššou frekvenciou výskytu. Navyše sa v týchto prípadoch jedná výlučne o detské obete.

Graf 3 § 369 Rozširovanie detskej pornografie v SR, 2014-2016



Zdroj: Spracované podľa údajov EŠSK

Na základe vlastných skúseností však zvýšené počty v policajných štatistikách nie sú len indikátorom zvýšenia nápadu tejto trestnej činnosti, ale aj viditeľným presúvaním kapacít kriminálnej polície na oblasť odhaľovania a objasňovania počítačovej kriminality, ako reakcie vyvolanej jej nárastom a závažnosťou. Nezanedbateľným faktorom je určite aj zníženie ľahostajnosti obetí a ich zákonných zástupcov, zvyšovanie informovanosti občianskej verejnosti ako i prijatie dlho očakávaných zmien v legislatívnej oblasti, ku ktorým prišlo k 1. januáru 2016.

3. POZITÍVNY VÝVOJ PREVENCIE NA SLOVENSKU A VAROVNÉ SIGNÁLY ZO ZAHRANIČIA

Okrem smerovania policajnej štatistiky do reálnych čísel považujeme za ďalší a veľmi dôležitý pozitívny vývoj aj akceleráciu intenzity preventívnych aktivít a záujem médií o problematiku osvety v oblasti ochrany používateľov internetu a moderných počítačových a telekomunikačných technológií pred počítačovou kriminalitou a to nielen detí, ale aj dospelých.

Tu treba zdôrazniť, že prevencia formou osvety, hlavne preventívne akcie cielene zamerané na určité špecifické skupiny potencionálnych obetí (napr. diferencovane podľa veku – žiaci prvého stupňa, žiaci druhého stupňa, stredoškoláci, vysokoškoláci a mladí dospelí, dospelí a seniori) sú najúčinnejším a pritom aj najlacnejším spôsobom boja proti počítačovej kriminalite. Kým polícia nastupuje do boja proti počítačovej kriminalite vo väčšine prípadov až v momente odhaľovania, objasňovania a vyšetrovania tejto špecifickej trestnej činnosti, pracovníci a často aj nadšenci bezodplatne pracujúci na poli prevencie vedia zachrániť obete ešte pred týmto momentom, ktorý už znamená istým spôsobom aj ďalšiu prehru spoločnosti.

[3] Dôvodovú správu k smernici Európskeho parlamentu a Rady Európy 2013/40/EÚ z 12. augusta 2013 o útokoch na informačné systémy prerokovala a schválila vláda Slovenskej republiky na svojom rokovaní dňa 1. júla 2015

[4] Objasnenosť týchto trestných činov nedosiahla ani 50%

Slovensko svojím neskorším vstupom do sféry informatizácie má obrovskú príležitosť poučiť sa z chýb vývoja informatizácie v zahraničí. Vývoj možno v mnohých oblastiach označiť za alarmujúci a preto je potrebné na varovné signály upozorňovať a tým vytvárať predpoklady na to, aby sa naša spoločnosť pokúsila týmto negatívnym trendom zabrániť alebo ich aspoň minimalizovať. Z posledného obdobia môžeme upozorniť aspoň na niektoré z nich.

V marci 2017 priniesli slovenské média[5] *správy o nemeckom 19 ročnom mužovi Marcelovi Hessovi, ktorý na internete zverejnil informácie a fotografie z brutálnej vraždy 9 ročného chlapca. Na mieste činu si spravil ešte zakrvavený trendovú selfie fotografiu, ktorú tiež zverejnil. Počas vyšetrovania bolo preukázané, že zavraždil aj inú osobu, ktorú poznal z internetu. Mladý muž sa ku skutku bez emócii priznal a ako dôvod uviedol, že ho k tomu viedla frustrácia. Napriek tomu, že v internetovej komunikácii bol aktívny, psychológ uviedol, že u neho absentovali sociálne kontakty, mal nulovú emocionálnu inteligenciu a označil ho za extrémne nebezpečného. Marcel Hess bol závislí od počítačov a počítačových hier.*

Ďalšou šokujúcou správou zverejnenou v médiách[6] *v marci 2017 je článok o znepokojení nórskej polície, menovite Národného úradu pre boj proti organizovanej kriminalite a inej závažnej trestnej činnosti Kripos o veľkom počte mužov vo veku 18 až 60 rokov, ktorí si cez počítačovú sieť z Hongkongu objednávajú nafukovacie panny s podobou detí. Časť z týchto prípadov bola priamo spojená s prípadmi súvisiacimi s detskou pornografiou.*

Alarmujúce prepojenie medzi internetom a šírením terorizmu v Európe nemožno spochybniť. Už v roku 2015 vydala Európskej komisia znepokojujúcu správu o šírení islamského radikalizmu prostredníctvom internetu, ktorá je do dnes stále aktuálnou a boj proti jeho šíreniu naďalej pokračuje[7]. Tak ako bolo v správach uvedené, nebezpečenstvo terorizmu, nie je spojené len s nelegálnou migráciou, ale rizikovými osobami sú často vlastní sfanatizovaní občania Európskej únie (často veľmi mladí), ktorí sú vďaka slepému preberaniu informácií z internetových zdrojov ľahko verbovaní do teroristických zoskupení alebo manipulovaní k páchaniu závažnej násilnej trestnej činnosti v mene rôznych ideológií.

Smutné je, že sa k radikálnym skupinám pridávajú často aj nadpriemerne inteligentní ľudia.

Príkladom môže byť napríklad 20 ročný mladík odsúdený v USA, ktorý sa k Islamskému štátu pridal ako hacker[8]*. Anglický hacker Junaid Hussain zabitý v Sýrii americkým dronom pôsobil ako šéf kybernetickej jednotky Islamského štátu.*

Najsmutnejším prípadom, z tejto skupiny bola vražda kňaza vo francúzskom meste Saint-Étienne-du-Rouvray, kde mladí muži v mene Islamského štátu podrezali tamojšieho kňaza[9]*. V tomto prípade už pred týmto skutkom sami rodičia upozornili na radikalizáciu mladíka, pričom mladík pochádzal zo vzdelanej rodiny. Matka 19 ročného francúzskeho občana Adela Kermicha je profesorka a sestra je lekárka.*

Toto riziko v plnom rozsahu platí aj o šírení extrémizmu a náboru do rôznych vojenských konfliktov, nakoľko metódy šírenia propagandy sú veľmi podobné.

Na začiatku roka 2017 boli zaznamenané aj iné alarmujúce udalosti ako sú krádeže vozidiel deťmi (často svojim rodičom) a následné dopravné nehody s následkom usmrtenia alebo ťažkého zranenia účastníkov cestnej premávky. Napriek tomu, že priamu súvislosť s využívaním moderných technológií ťažko dokázať, skreslené vnímanie reality hroziaceho nebezpečenstva v narastajúcom rozsahu v týchto prípadoch je viditeľné.

Negatívnych príkladov z posledného obdobia je naozaj niekoľko. Cieľom nie je poukázať na závažnosť kriminality mladých v spojení počítačovými a inými modernými komunikačnými technológiami, ale na ich nepripravenosť čeliť týmto lákadlám, či hrozbám, ktoré na nich v internetovom svete číhajú.

Dôležitosť úlohy zavedenia určitej regulácie audiovizuálneho obsahu dostupného mladým bolo aj jednou z priorít slovenského predsedníctva SK PRES v oblasti kultúry a audiovízie, pričom výsledkom bol návrh revízie smernice o audiovizuálnych mediálnych službách[10] pod predsedníckym vedením ministra kultúry SR.

Aktualizácia a praktické plnenie smernice o audiovizuálnych mediálnych službách môže byť dobrým začiatkom zastavenia nebezpečných trendov, ktoré vplývajú na mladú generáciu.

4. PSEUDOANONYMNÁ BUDÚCNOSŤ ĽUDSTVA

Technologický pokrok v dnešnej dobe priniesol dva základné technologicko-filozofické smery. Nekontrolované zverejňovanie informácií zo súkromného života (vo väčšine prípadov pravdepodobne neuvedomelé a vynútené poskytovateľmi služieb) a na druhej strane snaha o zachovanie anonymity, ktoré niekedy

[5] https://www.aktuality.sk/clanok/422996/nemecky-tinedzer-sa-priznal-k-dvom-vrazdam/
https://svet.sme.sk/c/20480403/devatnastrocny-nemecky-mladik-sa-priznal-k-dvom-vrazdam.html

[6] http://www.topky.sk/cl/11/1613283/Norska-policia-je-znepokojena--Coraz-viac-muzov-si-dovaza-nafukovacie-panny-v-podobe-deti
https://svet.sme.sk/c/20474460/norsko-znepokojuje-narast-dovozu-nafukovacich-panien-s-podobou-deti.html

[7] http://spravy.pravda.sk/svet/clanok/374053-teroristicke-utoky-nie-su-dielom-utecencov-upozornuje-europska-komisia/
https://svet.sme.sk/c/20289595/eu-vypoveda-vojnu-islamskemu-statu-na-internete-ziada-o-pomoc-it-firmy.html

[8] http://www.zive.sk/clanok/118218/v-usa-odsudili-hackera-na-20-rokov-za-pomahanie-islamskemu-statu
https://svet.sme.sk/c/7982776/dron-zabil-hlavneho-hackera-islamskeho-statu-kradol-data-aj-blairovi.html

[9] http://www.ta3.com/clanok/1088129/vrahom-knaza-bol-tinedzer-ktoreho-zmenil-utok-na-charlie-hebdo.html

[10] http://www.eu2016.sk/sk/politicke-a-expertne-podujatia/konferencia-ochrana-maloletych-regulacne-konvergencie

hraničia až s pokusmi o nastolenie právnej anarchie. Málokto si však uvedomuje, že čokoľvek, považované za anonymné, sa ľahko môže stať verejným. Preto pri moderných technológiách musíme slovo anonymita vždy chápať vo význame pseudoanonymita. Toto platí nielen pri rôznych technológiách tzv. anonymného pripojenia na verejnú sieť internet, ale aj pri digitálnych kryptomenách, či ďalších technológiách, ktoré ešte len čakajú za rohom budúcnosti. Každý jednotlivec využívaním nových technológií fungujúcich na globálnych komunikačných sieťach stráca časť súkromia a delí sa o svoje súkromie s veľkými technologickými spoločnosťami, ktorých obchodný plán je založený na zhromažďovaní a obchodovaní s týmito informáciami. Každý jednotlivec si musí sám vybrať medzi ľahšou cestou zdieľania svojho súkromia, nepoužívania týchto technológií alebo sa vzdelávať, ako technológie používať tak, aby zásah do jeho súkromia dokázal minimalizovať. Schopnosť orientovať sa v bezpečnostnom prostredí, už nie je iba otázkou dobrej vôle, ale stáva sa čoraz viac nevyhnutnosťou a elementárnym predpokladom prežitia. [7, s. 188] Je dôležité, aby si toto mladá generácia uvedomila a osvojila. Pri vzdelávaní získavajú kľúčové postavenie osoby s odbornými poznatkami v oblasti informatiky a srdcom pedagóga.

5. ZÁVER

Keď zlyháva rodina natoľko, že je ohrozený život, zdravie a zdravý fyzický a duševný vývoj dieťaťa, je úlohou spoločnosti nahradiť tieto kľúčové výchovné funkcie a postarať sa o deti, ktoré sú budúcnosťou spoločnosti. V takýchto núdzových prípadoch musí aj poznatky, ako prežiť v digitálnom veku a ako využívať nové technológie bezpečne, sprostredkovať deťom samotná spoločnosť. Podobný princíp funguje aj v prírode a je nevyhnutným predpokladom prežitia rodu či druhu. Problémom súčasných rodín je, že sa začína strácať záujem o budovanie budúcnosti detí, nakoľko riešia existenčné problémy. Pomerne často sa obmedzuje konanie rodičov len na akési prežívanie (tzv. „život z práce do práce“). Navyše mnohí z týchto rodičov majú problém, hlavne technického charakteru, ako sa s modernými technológiami vysporiadať. Záchranným kolesom sa tak môže stať v tomto búrlivom čase rýchleho vývoja nových informačných technológií informácia ako prežiť bezpečne, ktorá príde k dieťaťu včas.

BIBLIOGRAFICKÉ ODKAZY

[1] VÁŇA, J. *Environmentálna bezpečnosť – podmienka bezpečnosti sociálnej sféry.* In *SECURITY – THE KEY REQUIREMENT OF THE PRESENT: SELECTED ISSUES OF SECURITY SCIENCE.* Bratislava : Akadémia PZ v Bratislave, 2016. s. 45-57. ISBN 978-80-8054-675-5.

[2] LISOŇ, M. *Odhaľovanie a objasňovanie trestných činov všeobecnej kriminality*. Akadémia PZ v Bratislave, 2016. s. 248. ISBN 978-80-8054-673-1.

[3] HAJDÚKOVÁ, T., BACIGÁL, I. *Využitie virtuálnej komunikácie cez internet ako prostriedok ku sexuálnemu zneužívaniu detí.* In *Mravnostná kriminalita ako spoločenský fenomén a možnosti jej kontroly*. Bratislava : Akadémia PZ v Bratislave, 2015. s. 242. 978-80-8054-646-5.

[4] KURILOVSKÁ, L., ŠIŠULÁK, S. *Internetový marketing ako metóda páchania trestnej činnosti* In *Aktuálne otázky tretsného práva v teórii a praxi*. [Online] 2015, 3. ročník. [Dátum: 12. 9. 2015.] Dostupné tiež na: http://www.akademiapz.sk/sites/default/files/KVPV/KTP/Aktu%C3%A1lne%20ot%C3%A1zky%20TP%20v%20T%20a%20P%20-%203.ro%C4%8D. %20e-verzia.pdf. ISBN 978-80-8054-637-3.

[5] METEŇKO, J., BINDEROVÁ, M. *Koordinovanie národného výskumu a politiky na zaistenie bezpečnosti pri významných udalostiach v Európe* EU SEC II : Záverečná správa medzinárodnej vedeckovýskumnej úlohy : FP7 Grant Agreement (2008-2011). Bratislava : Akadémia PZ v Bratislave, 2012. s. 34. EAN 9788080546762.

[6] ODLER, R. *Schengenský priestor verzus obchodovanie s ľuďmi.* In *Sborník příspěvků z 6. ročníku mezinárodní vědecké konference "Bezpečná Evropa 2013.* Karlovy Vary : Vysoká škola Karlovy Vary, 2013. S 110121.

[7] MURDZA, K. *Bezpečnostná dezorientácia občana a jej negatívne dôsledky* In *SECURITY – THE KEY REQUIREMENT OF THE PRESENT: SELECTED ISSUES OF SECURITY SCIENCE*. Bratislava : Akadémia PZ, 2016. s. 188-197. ISBN 978-80-8054-675-5.

[8] Stratégia prevencie kriminality a inej protispoločenskej činnosti zákona č. 583/2008 Z. z. o prevencii kriminality a inej protispoločenskej činnosti a o zmene a doplnení niektorých zákonov v znení zákona č. 403/2010 Z. z.,

[9] Dohovor o počítačovej kriminalite publikovaný pod č. 137/2008 Z.z.

[10] Smernica Európskeho parlamentu a rady 2011/92/EÚ z 13. decembra 2011 o boji proti sexuálnemu zneužívaniu a sexuálnemu vykorisťovaniu detí a proti detskej pornografii

[11] Dohovor Rady Európy o ochrane detí pred sexuálnym vykorisťovaním a sexuálnym zneužívaním publikovaný pod č. 164/2016 Z. z

[12] Smernica Európskeho parlamentu a Rady 2013/40/EÚ z 12. augusta 2013 o útokoch na informačné systémy

[13] SMAHEL, D., HELSPER, E., GREEN, L., KALMUS, V., BLINKA, L., ÓLAFSSON, K. 2012. *Excessive Internet Use Among European Children* [online] [vytvorené november 2012] dostupné na internete http://www.lse.ac.uk/media%40lse/research/EUKidsOnline/EU%20Kids%20III/Reports/ExcessiveUse.pdf. 9 p. ISSN 2045 – 256X.

[14] GREGUSSOVA, M., DROBNÝ, M. 2012. *Deti v sieti, Ako chrániť seba a naše deti na internete.* Bratislava : eSlovensko, 2013. 108 s. ISBN 978-80-970676-6-3.

[15] ÓLAFSSON, K., LIVINGSTONE, S., HADDON, L. et al. 2013. *Children´s Use of Online technologies in Europe, A review of the European evidence base.* London : LSE The school of economics and political science, [online] [vytvorené máj 2013] dostupné na internete

[16] SAKAŘ, P.: *Sexuální reviktimizace,* In Česká a slovenská psychiatrie. č. 7, ročník 103, Brno: ČSL JEP, 2007, s. 346-352, ISSN 1212-0383.

[17] VELIČKOVÁ HULANOVÁ, L. *Kybergrooming a kyberstalking*. In: KRČMÁŘOVÁ, Barbora et al. Děti a online rizika: Sborník studií. 1. vyd. Praha: Sdružení Linka bezpečí, 2012, s. 87 - 107. ISBN 978-80-904920-2-8. Dostupné z: http://www.linkabezpeci.cz/webmagazine/kategorie.asp?idk=239

[18] QUAYLE, E., TAYLOR, M. *Model of Problematic Internet Use in People with a Sexual Interest in Children*, CYBERPSYCHOLOGY & BEHAVIOR, Volume 6, Number 1, 2003, Mary Ann Liebert, Inc.

[19] Tlačový servis INHOPE. Statistiky horkých linek varují – v prostředí online roste výskyt materiálů se sexuálně zneužívanými dětmi [online]. [vytvorené 16. 6. 2014]. [cit. 2015-3-25]. Dostupné na internete: http://www.saferinternet.cz/aktuality/337-statistiky-horkych-linek-varuji-%E2%80%93-v-prostredi-online-roste-vyskyt-materialu-se-sexualne-zneuzivanymi-detmi.html.

[20] http://www.lse.ac.uk/media@lse/research/EUKidsOnline/EU%20Kids%20III/Reports/D2.2ReviewEvidenceDatabase.pdf . ISSN 2045 - 256X.

ScratchMaths: vzdelávací obsah a princípy tvorby
ScratchMaths: curriculum content and its design principles

Ivan Kalaš
Katedra základov a vyučovania informatiky
Fakulta matematiky, fyziky a informatiky
Univerzita Komenského v Bratislave
Slovensko
kalas@fmph.uniba.sk

ABSTRACT
Main goal of the ScratchMaths project is to explore how 9- to 11-year-old pupils learn to program and how the development of their computational thinking correlates with the development of their mathematical thinking. We will characterise each period of the iterative design and qualitative research of the project since September 2014. Our focus will be on the *5E pedagogical framework* which we have created and applied, the structure of the curriculum content and teaching materials we have developed, as well as summarizing several *key design principles*, which we formulated while designing and evaluating the intervention.

Keywords
Programming at KS2. Scratch. Design research. Pedagogical framework 5E. Structure of the ScratchMaths intervention. Design principles for programming curriculum content.

ABSTRAKT
Cieľom projektu ScratchMaths je skúmať poznávací proces 9- až 11-ročných žiakov pri učení sa programovania a súvislosť medzi rozvojom ich programovacích zručností a matematického myslenia. V našom príspevku charakterizujeme jednotlivé etapy iteratívneho vývoja a kvalitatívneho výskumu, ktorými projekt prechádza od svojho začiatku v septembri 2014. Ťažisko príspevku tvorí prezentácia *pedagogického rámca 5E*, ktorý sme v projekte vytvorili a aplikovali, ďalej štruktúra vzdelávacieho obsahu a učebných materiálov projektu, a tiež diskusia o niekoľkých *základných princípoch tvorby obsahu pre programovanie*, ktoré sme v projekte identifikovali.

Kľúčové slová
Programovanie na ZŠ. Scratch. Výskum vývojom. Pedagogický rámec 5E. Štruktúra výstupov projektu. Princípy tvorby obsahu pre programovanie.

1. VÝCHODISKÁ
V ostatných rokoch sme svedkami novej vlny záujmu o programovanie žiakov ako plnohodnotnej súčasti ich formálneho vzdelávania už od prvej triedy základnej školy [1]. V Anglicku nastala zásadná zmena v roku 2013, kedy vláda zrušila povinný predmet ICT (s odôvodnením, že nie je dostatočne efektívny pri rozvoji informatického poznania a myslenia, teda pri rozvoji tzv. *computational thinking* [2]) a nahradila ho novým predmetom *computing* – od prvej triedy prvého stupňa, až po ukončenie stredoškolského vzdelávania. Od septembra 2014 sa od anglických učiteľov základnej školy (teda vzdelávacích stupňov KS1 a KS2, pre deti od 5 do 11 rokov) vyžaduje, aby plnili požiadavky nového národného kurikula [3]. Vznikli rôzne iniciatívy a organizácie na podporu tejto zmeny, akou je napr. CAS – Computing At School. Vláda tiež financuje rôzne projekty, ktoré majú túto zmenu podporiť a sledovať jej napredovanie.

Vďaka tejto situácii sme na UCL Institute of Education (ďalej len UCL IoE) získali trojročný projekt ScratchMaths financovaný nadáciou EEF – Education Endowment Foundation, s cieľom vyvinúť nový vzdelávací obsah predmetu computing (jeho programovacej zložky pre žiakov 5. a 6. triedy) a jeho prostredníctvom skúmať, ako môže rozvoj programovacích zručností a informatického myslenia zároveň podporiť aj rozvoj matematického myslenia týchto žiakov. V našom príspevku chceme čitateľovi prezentovať stratégiu projektu, ktorý začal 1. septembra 2014, a jeho doterajšie výsledky, teda vlastný *vzdelávací obsah*, *pedagogický rámec*, ktorý sme pri iteratívnom vývoji a overovaní vytvorili, a tiež sériu *kľúčových princípov tvorby* tohto obsahu, ktoré sme sformulovali na základe našej spolupráce s niekoľkými základnými školami v Londýne a ďalšími 100 školami po celom Anglicku.

Aby si čitateľ mohol utvoriť lepšiu predstavu o celkovej situácii, je užitočné zdôrazniť, že:

- projekt je cielený na bežné štátne základné školy, náš obsah vyučujú štandardní učitelia primárneho vzdelávania,
- za programovací jazyk bol vopred stanovený Scratch, čo sa potvrdilo ako správna voľba,
- predmet computing sa učí naraz v celej triede, ktorá má priemerne asi 26 žiakov. Vybavenie škôl počítačmi je skromnejšie, než by sme očakávali. Škola má zvyčajne jednu, v lepšom prípade dve triedy vybavené laptopmi. Pripojenie na internet je uspokojivé, v žiadnom prípade však nie bezchybné,
- učitelia sú neobyčajne zodpovední, nové kurikulum akceptujú ako zákon a snažia sa čo najlepšie napĺňať jeho požiadavky. Majú vysokú úroveň všeobecnej digitálnej gramotnosti, niektorí z nich už majú určité nesystematické skúsenosti s jazykom Scratch (aj keď sa ukázalo, že práve to bolo pri našich vzdelávaniach zvyčajne skôr negatívum),
- učiteľ pozitívne akceptuje čokoľvek nové iba vtedy, ak uverí, že tým prospeje svojim žiakom – v našom prípade v predmetoch computing a matematika,

- v každej triede je knižnica, ktorá sa neustále používa, ale žiaci nemajú vlastné učebnice. Náš vzdelávací obsah je teda určený učiteľovi, jeho súčasťou sú pracovné listy pre žiakov, prezentácie na hodinu a pod.

Prvý rok projektu sme venovali štúdiu literatúry, pozorovaniu hodín programovania a matematiky v štyroch spolupracujúcich školách (*design schools*), organizovaniu série poldenných stretnutí s učiteľmi z týchto škôl, aby sme získali predstavu o tom, čo budú primerané pedagogické postupy a náročnosť z hľadiska informatiky a programovania – pre žiakov, ale aj pre ich učiteľov, a napokon najmä vývoju pilotnej verzie vzdelávacieho obsahu pre 5. ročník a jeho iteratívnemu overovaniu a upravovaniu. Vytvorili sme tiež sieť 100 anglických projektových škôl, ktoré potom náš nezávislý hodnotiteľ (Sheffield Hallam University) podľa metodiky RCT[1] náhodne rozdelil na 50 škôl *prvej vlny* a 50 škôl *druhej vlny*. Obrázok 1 predstavuje časový harmonogram projektu v 2. a 3. roku.

Obrázok 1. Časový harmonogram projektu v 2. a 3. roku

2. ETAPY VÝSKUMU VÝVOJOM

Kým nezávislý hodnotiteľ zo Sheffieldu má za úlohu realizovať kvantitatívne analýzy v 100 projektových školách, projektový tím na UCL IoE realizuje vlastný kvalitatívny výskum. Z pohľadu riešiteľov projektu je ScratchMaths príkladom **výskumu vývojom** (*design research*), ktorý vždy býva iteratívny, pričom každý jeho cyklus tvoria rôzne výskumné a vývojové aktivity. V našom projekte postupujeme podľa metodiky Nieveena a Folmera [4], ktorí celý proces výskumu vývojom delia na tri fázy: *(1) úvodnú fázu vývoja, (2) fázu prototypu* a *(3) fázu sumatívneho hodnotenia*.

V úvodnej fáze vývoja sme urobili *analýzu potrieb a príležitostí*, a to pozorovaním vyučovacích hodín a spoluprácou s učiteľmi štyroch projektových škôl. Naša prvá koncepcia vyplynula z predchádzajúcej expertízy členov tímu, z workshopov s učiteľmi spolupracujúcich škôl a zo štúdia odbornej literatúry, napr. aj bohatej zbierky učebných materiálov v Scratchi. Výstupom tejto etapy je náš *pedagogický rámec 5E*, prezentovaný v nasledujúcej kapitole.

Vo fáze prototypu sme navrhli štruktúru materiálov, vytvorili prototyp obsahu pre 5. ročník a v niekoľkých iteráciách ho transformovali na definitívny dizajn výslednej pedagogickej intervencie. Špeciálnu pozornosť sme venovali transformácii požiadaviek národného kurikula do vhodnej sekvencie zodpovedajúcich informatických konštruktov. Pravidelným pozorovaním vyučovacích hodín s materiálmi projektu sme realizovali iteratívne formatívne hodnotenie intervencie.

V súčasnosti sa projekt nachádza vo fáze sumatívneho hodnotenia. Na školách prvej vlny už pracujú druhý rok s celou intervenciou, na školách druhej vlny s jej prvou polovicou. Výsledky projektu nezávisle skúmajú dva tímy dvoma rozličnými stratégiami.

3. PEDAGOGICKÝ RÁMEC 5E

Najdôležitejším výstupom úvodnej fázy projektu bol explicitne sformulovaný *pedagogický rámec*, ktorý neskôr určil náš postup pri vývoji vzdelávacieho obsahu (DiSessa a Cobb takýto postup v [5] nazývajú *framework of action*). Ide o päť zásad, ktorých cieľom je pomôcť učiteľom pri voľbe konkrétnych metodických postupov. Pedagogický rámec 5E vyplýva z nášho konštrukcionistického presvedčenia[2] o tom, že žiak sa najefektívnejšie učí (teda buduje si nové položky a spoje v štruktúre svojich znalostí) vtedy, keď tvorí dáky podľa svojho presvedčenia zmysluplný produkt, či už hmatateľný alebo virtuálny. Takýto produkt má podľa [6] poskytovať príležitosť na skúmanie dôležitých myšlienok a svoje vlastné objavy. Za dôležitú považujeme skutočnosť, že päť zásad nášho rámca 5E nemá žiadne poradie (*skúmanie* sa teda neimplementuje pred *zdieľaním* a pod.) a celkom určite sa budú v rôznych kontextoch aplikovať rôznymi spôsobmi.

Explore: skúmaj. Vzdelávací obsah by mal žiakom poskytnúť dostatok príležitostí, aby mohli sami skúmať, objavovať, iteratívne rozvíjať a ladiť svoje riešenia. Aj samostatné skúmanie však treba starostlivo organizovať a do istej miery aj riadiť a vyhodnocovať. Nájsť správnu rovnováhu medzi samostatným otvoreným skúmaním a riadeným poznávacím procesom však nie je jednoduché – práve tento vzťah *medzi inštruovaním a objavovaním* je jedným z piatich protirečení, ktoré Brennan identifikuje v [7], keď uvažuje o tom, ako podporiť konštrukcionistické vzdelávanie učiteľov.

V pedagogickom rámci ScratchMaths sa snažíme žiakov vybaviť rôznymi zručnosťami pre ich vlastné skúmanie, ako napr. systematickým rozlišovaním a aplikovaním *priameho riadenia* a *riadenia programom*, pričom pri priamom riadení ešte rozlišujeme *priamu manipuláciu* a *priamu akciu*:

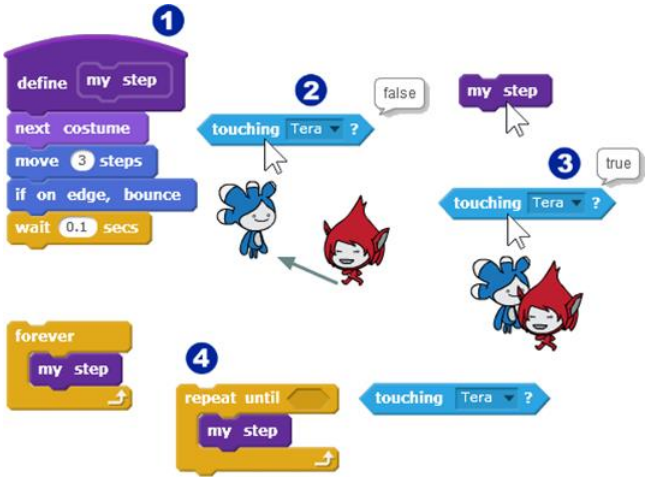
- **priamou manipuláciou** (*direct manipulation*) riadia žiaci agenta, napr. hračku, Bee-Bota a pod., priamo „fyzicky“ rukou, a tak vyriešia danú úlohu (napr. *priveď autíčko po naznačených cestách do garáže*). V jazyku a prostredí Scratch to napr. znamená premiestniť postavu na scéne ťahaním myšou alebo voľbou nového kostýmu pre agenta na záložke Kostýmy a pod.,
- **priama akcia** (*direct drive*) je prvým krokom k riadeniu agenta sprostredkovane: robotickej hračke alebo virtuálnemu agentovi zadáme jednoznačnú inštrukciu, ako napr. *urob krok vpred* alebo *otoč sa vpravo o 90 stupňov*, a ten okamžite vykoná zodpovedajúcu reakciu, teda to, čo Blackwell v [8] označuje ako jednu akciu a jeden okamžitý a viditeľný efekt, výsledok, reakciu. V metodike ScratchMaths venujeme priamej akcii veľkú pozornosť: každý blok sa totiž dá v ploche skriptov umiestniť a kliknutím naň vykonať izolovane od iných blokov. Práve to považujeme za stratégiu inkrementálneho učenia sa, pozri obrázok 2 a nasledujúci text,

[1] *randomized controlled trial*

[2] vyplýva však aj z toho, že v anglickej pedagogike pre primárne vzdelávanie je forma frontálneho výkladu neakceptovateľná

- **riadenie programom** (*computational drive*) znamená v symbolickom jazyku s jasnou syntaxou a sémantikou naplánovať budúce správanie agenta – správanie, ktoré sa zrejme bude opakovane vykonávať, ladiť, modifikovať a pod.

Obrázok 2 ilustruje tieto tri formy riadenia na aktivite *3.2.1 Spoznávame podmienky* z materiálov projektu. V predchádzajúcej aktivite už žiaci pre Gigu (pravú z dvoch postáv) naprogramovali nekonečnú prechádzku po scéne s odrazmi od okrajov pomocou skriptu **forever my step**, ktorý potom použili ako jej reakciu na udalosť **when this sprite clicked** (krok 1 na obrázku). Teraz však chceme, aby Giga kráčala smerom k Tere (postava vľavo) iba dovtedy, kým sa jej dotkne – teda *aby prišla k Tere*. Metódou priamej akcie (klikaním myšou) budú žiaci skúmať izolovaný blok **touching Tera?** (krok 2), pričom Gigu postupne posúvajú smerom k Tere buď ťahaním myšou (metódou priamej manipulácie) alebo opakovaným klikaním na izolovaný blok **my step** (metódou priamej akcie). Keď sa Giga napokon Tery dotkne, blok **touching Tera?** odpovie **true** (krok 3) – toto je prvá príležitosť, kedy žiaci objavia *podmienku* ako nástroj na riešenie problému ako spoznať, či má Giga urobiť ďalší krok k Tere. Koľkokrát má teda Tera opakovať svoj krok? *Kým sa dotkne Tery* (krok 4).



Obrázok 2. Priame riadenie ako stratégia pre skúmanie

Explain: vysvetli. Pri pozorovaní vyučovacích hodín programovania počas úvodnej fázy projektu sme si uvedomili, že metóda voľného (nekoordinovaného) skúmania[3] často vedie k nedostatočnej miere porozumenia úlohy riadiacich blokov jazyka Scratch, ktoré žiaci používali (niekedy až skoro náhodne, s niekoľkonásobným vnáraním a pod.). Preto sme v našom pedagogickom rámci začali klásť dôraz na celotriedne diskusie riadené učiteľom, a tiež na pravidelné diskusie v dvojiciach s cieľom vysvetliť spolužiakovi svoju myšlienku, názor, problém, postup či objav... teda na tzv. reflektívne rozhovory (*reflective questioning*). Harel a Papert [9] zdôrazňujú, že situácie, kedy žiaci verbalizujú svoje zámery a postupy, sú dôležité pre ich ďalší kognitívny rozvoj. Noss a Hoyles [10] sa domnievajú, že takáto slovná artikulácia je jedným z kľúčových komponentov konštrukcionistického učenia sa.

Envisage: uvažuj. Žiakov nabádame k tomu, aby uvažovali o výsledku či efekte vykonania daného programu (jedného alebo viacerých skriptov) skôr, než ho reálne vykonajú v Scratchi. Aktivity, ktorých súčasťou je takéto mentálne vyhodnocovanie programu, rozvíjajú schopnosti žiakov hlbšie uvažovať o základných prvkoch programu, výrazoch a príkazoch, o význame poradia krokov vo výpočte, o riadiacej štruktúre, udalostiach či interakciách medzi aktérmi. Tradičnou metódou logovskej kultúry, ktorú intenzívne využívame aj v našom pedagogickom rámci, je zaraďovať do vyučovania aktivity, kedy žiak vlastným telom vykonáva program pre aktéra (objekt, postavu...). Stratégia *uvažovania o programe* tým, že ho fyzicky vykonávame, akoby bol zostavený pre nás, sa nazýva *body syntonicity* [11]. Takýto postup je produktívny nielen pri uvažovaní o danom programe, ale aj pri jeho vytváraní. Aktivity zamerané na uvažovanie o programe sa dobre realizujú aj „bez klávesnice“ (*unplugged*).

Exchange: zdieľaj. Komunikácia, spolupráca a zdieľanie sú efektívne stratégie učenia sa, ktoré pomáhajú učiacim sa pri tvorbe zmysluplného produktu *prostredníctvom interakcií s ďalšími vrstovníkmi v úlohe publika, spolupracovníkov či koučov* [7]. Zdieľanie myšlienky na riešenie problému, metódy, skriptu alebo celého objektu aj s jeho správaním (programom) považuje Brennan a Resnick [12] za jeden z kľúčových informatických postupov (*computational practices*).

V našom projekte využívame tento princíp ako efektívnu metódu na podporu učenia sa medzi spolužiakmi na rôznej úrovni: aktivity musia poskytovať priestor na takéto interakcie. Vo vývojových triedach sme si napr. často všimli, ako efektívne sa v miestnosti spontánne šíria zaujímavé objavy žiakov. Zdieľanie však nemôže byť iba spontánne: naše aktivity často nabádajú k hľadaniu rôznych riešení a k tomu, aby svoje objavy potom žiaci prezentovali ostatným – a aby sa za prirodzený postup považovalo aj prevzatie myšlienky spolužiaka a jej ďalšie rozpracovanie.

Nezabúdajme však ani na to, že v tomto veku si deti ešte stále iba rozvíjajú svoje zručnosti na spoluprácu, musíme im preto ako učitelia pomáhať aj v tejto rovine rozvoja.

bridgE: prepájaj. Skúsený učiteľ využije každú vhodnú príležitosť na to, aby zdôraznil súvislosť medzi novou problematikou a inou, už známou vecou, už známym poznatkom. Jedným z cieľov každej konštrukcionistickej pedagogiky je vytvárať príležitosti pre skúmanie *veľkých myšlienok (powerful ideas)*, a myšlienky sa stávajú veľkými aj vďaka ich prepojeniu na iné oblasti, napr. na matematiku [13]. Kvalita nového poznatku a jeho zaradenia do štruktúr poznania tiež úzko súvisí s jeho viacnásobnými reprezentáciami, s našou schopnosťou pozrieť sa na to isté z inej perspektívy.

V projekte ScratchMaths nás explicitne zaujíma prepojenie programovania a matematického myslenia: hlavným cieľom projektu je vybudovať taký vzdelávací obsah a pedagogiku programovania v 5. a 6. triede, ktorá podporí rozvoj matematického myslenia novými produktívnymi spôsobmi.

4. ŠTRUKTÚRA OBSAHU

Výsledkom druhej etapy nášho vývoja bol vlastný vzdelávací obsah, ktorý sa skladá zo šiestich modulov: troch pre 5. a troch pre 6.

[3] ktorá je bežným postupom na vyučovacej hodine typu *„urob si vlastnú hru podľa odkopírovanej predlohy programu z webu alebo knihy“*

triedu. Kým v prvých troch moduloch sme kládli dôraz na základy programovania a rozvoj informatického myslenia, vývoj pre 6. triedu bol riadený snahou budovať zaujímavé prepojenia s matematikou – a, prirodzene, ďalej rozvíjať informatické myslenie a poznanie. Všetkých šesť modulov je zoradených do záväzného poradia, keďže každý konštrukt[4], s ktorými sa majú v našom obsahu žiaci zoznámiť, je starostlivo projektovaný naprieč niekoľkými modulmi. To nám umožňuje budovať preň prípravné kroky a vhodne stupňovať potrebu žiaka objaviť práve takýto pojem či postup, a potom ho opäť a opäť používať, upevňovať nový poznatok, prípadne ho ďalej rozvíjať. Príkladom takýchto konštruktov sú definícia vlastného príkazu či premenná, ktorých poznávacia trajektória sa vinie všetkými šiestimi modulmi.

Modul 1: Vytvárame vzory

Informatické konštrukty a konštrukty v prostredí Scratch

- presúvanie postavy priamou manipuláciou; jej pečiatkovanie pomocou izolovaného bloku **stamp** priamou akciou
- inicializačný skript; inicializácia udalosťou **when green flag clicked**; vykonanie skriptu kliknutím
- otáčanie postavy pomocou blokov **turn ...** so základným vstupom 15 stupňov (*default value*) – priama akcia
- modifikovanie vstupnej hodnoty bloku; izolovaný blok **move**
- spájanie (dvoch, potom troch) blokov; ich vykonanie – riadenie postavy programom
- opakované vykonávanie skriptu **move-turn-stamp**; blok **repeat** ... ; vizualizácia algoritmu – blok **wait**
- zmena kostýmu priamou manipuláciou, potom priamou akciou blokom **next costume**; **next costume** v skripte
- dva algoritmy pre vytváranie kruhových vzorov; záporný vstup pre blok **move** – návrat do stredu vzoru
- výber kostýmu z viacerých, blok **switch costume to ...**
- definovanie vlastného bloku pomocou **define**
- používanie vlastného bloku pri tvorbe zložitejších vzorov

Modul 2: Geometria chrobáka

Informatické konštrukty a konštrukty v prostredí Scratch

- pero postavy, bloky **set pen color** a **set pen size**
- bloky **pen down** a **pen up**; základné kreslenie; sekvencia
- opakovanie čiary s otočením – kreslenie regulárnych polygónov; definovanie vlastného bloku **square** s konštantnou veľkosťou strany a práca s ním; **triangle**
- používanie pripravených nových blokov s náhodnými vstupmi – **set random pen colour** a **set random pen size**
- kreslenie bodky a čiarky **dot** a **dash**
- udalosť **when ... key pressed**
- skákanie na náhodné pozície pomocou pripraveného bloku
- skúmanie náhodných vstupov **pick random ...** v definícii a ich upravovanie

Modul 3: Učíme postavy spolupracovať

Informatické konštrukty a konštrukty v prostredí Scratch

- viacero postáv a ich reakcie; paralelné procesy
- bloky **show** a **hide**: ďalšie atribúty postáv; grafické efekty; násobné kostýmy; animácia
- udalosti **when green flag clicked** a **when this sprite clicked**
- nekonečný cyklus **forever**; *implicitná podmienka* **if on edge, bounce**
- pohyby a skoky; bloky **set y to ...** a **change y by ...** ; info bloky **x position** a **y position**
- podmienka **touching *sprite*?**; hodnoty **true** a **false**; ohraničené opakovanie **repeat until ...**
- podmienené vykonanie; blok **if**; podmienka **touching *colour*?**
- info blok **y position** v relačnej podmienke > alebo <
- komunikácia postáv – posielanie a prijímanie správ; bloky **broadcast *message*** a **when I receive *message***; komunikácia *jeden s jedným*, komunikácia *jeden s viacerými*

Modul 4: Budujeme čísla

Informatické konštrukty a konštrukty v prostredí Scratch

- atribút objektu ako *nepriama premenná* – kostým postavy vyjadruje implicitnú informáciu; blok **costume #**
- výraz, ktorého vstupom je *nepriama premenná*
- aritmetický výraz s operátormi + a –
- nastavenie hodnoty atribútu prostredníctvom výrazu
- posielanie správ ako prostriedok na implementáciu závislostí medzi postavami
- duplikovanie postavy spolu s jej správaním
- počet opakovaní závisí od momentálnej hodnoty atribútu
- používateľom definovaná udalosť – konštrukcia *whenever*, čiže **forever if ...**

Modul 5: Skúmame matematické súvislosti

Informatické konštrukty a konštrukty v prostredí Scratch

- blok **ask**; vstup používateľa ako hodnota bloku **answer**
- výraz, ktorého vstupom je *nepriama premenná* **answer**
- výraz s operátorom **join**; aritmetický výraz s operátormi * a /
- nastavenie hodnoty atribútu podľa vstupu používateľa
- premenná; premenná ako vstup pre príkaz; blok **set *var* to ...**
- premenná určuje počet opakovaní
- premenná mení svoju hodnotu v cykle; blok **change *var* by ...**
- riadenie pohybu postavy klávesmi

[4] označujeme takto spoločne koncepty i operácie s nimi

Modul 6: Súradnice a geometria

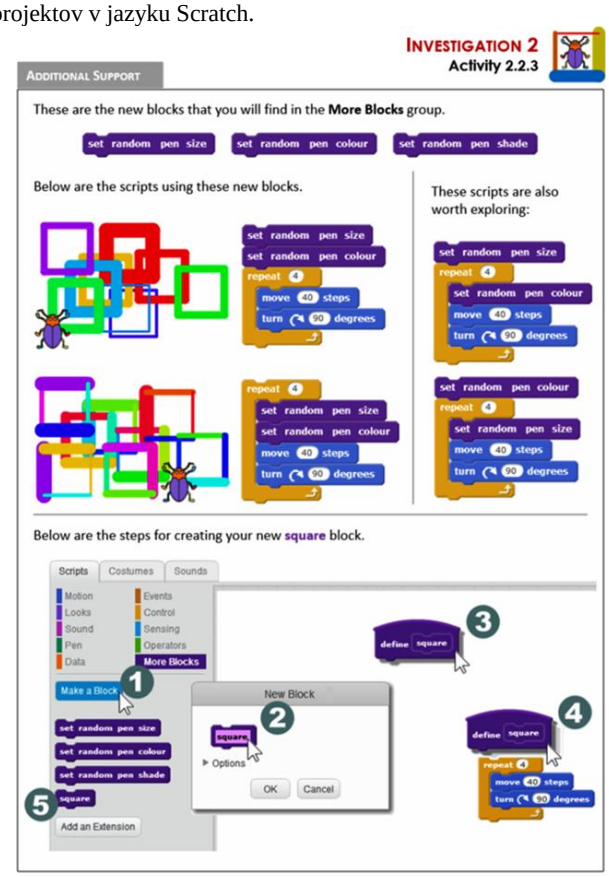
Informatické konštrukty a konštrukty v prostredí Scratch

- **blink**, **nod** a **walk** – rôzne animácie s jediným súborom kostýmov
- náhodné skákanie s obmedzením pomocou rozsahu súradníc
- aritmetický výraz s náhodným vstupom z určitého intervalu

- premenná ako vstup v aritmetickom výraze
- pohyby v sieti bodov
- výpočty s náhodným vstupom s obmedzeniami
- živý trojuholník a štvoruholník – základy dynamickej geometrie

5. UČEBNÉ MATERIÁLY PROJEKTU

Jedným z výstupov projektu je súbor materiálov, ktoré učiteľom sprístupňujeme elektronicky na projektovom portáli[5]. Tvoria ich učebné materiály pre učiteľa, prezentácie na vyučovacie hodiny pre žiakov, doplňujúce materiály a pomôcky, a tiež súbor úvodných projektov v jazyku Scratch.

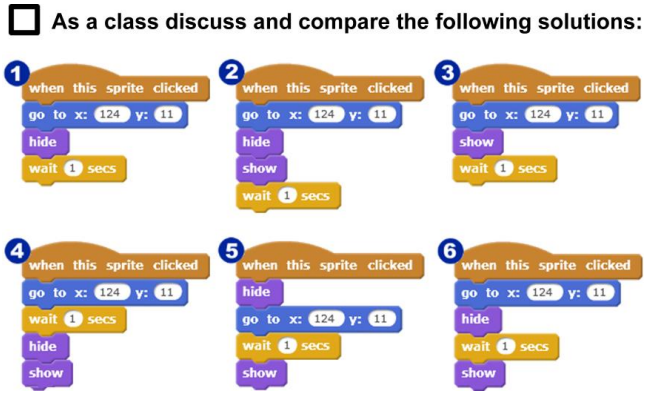


Obrázok 3. Ukážka materiálov pre učiteľa, modul 2, str. 20

Pre učiteľa: Kľúčovou časťou materiálov pre každý **modul** je rozsiahly metodický materiál. Začína stručnou špecifikáciou zamerania modulu, identifikáciou zodpovedajúcich konštruktov z informatických a matematických osnov, a tiež identifikáciou vhodných prepojení na iné vyučovacie predmety a iné oblasti poznávania. Pre moduly 4 až 6 sme do úvodu pridali aj „mapu bádaní[6] a aktivít" s vyznačením, ktoré časti považujeme za kľúčové pre postup žiakov do vyššieho modulu a ktoré sa dajú chápať ako doplňujúce či rozširujúce.

Každý modul členíme na štyri **bádania**, pričom bádanie tvoria štyri **aktivity** – tie predstavujú konkrétnu činnosť naplánovanú na jednu až dve vyučovacie hodiny. V aktivitách sa snažíme využiť viacero pedagogických princípov rámca 5E, aby poskytli žiakom príležitosť *skúmať*, *vysvetľovať*, *uvažovať*, *zdieľať* aj *prepájať*. Okrem toho, aktivity obsahujú aj sériu otázok na stimuláciu produktívnych diskusií v triede. Jedna zo štyroch aktivít zvyčajne býva „bez klávesnice": niekedy celkom mimo počítačov, kedy žiaci fyzicky animujú (vykonávajú) niektorý algoritmus, prípadne vypĺňajú vytlačený pracovný list, ktorý je súčasťou materiálu, alebo sedia na koberci pred interaktívnou tabuľou a individuálne, v dvojiciach alebo ako trieda riešia problém, prezentujú, komentujú či analyzujú riešenia a pod.

Bežnú programovaciu aktivitu v materiáloch tvorí jedna až dve strany konkrétnych krokov (zadanie), ktoré má učiteľ so žiakmi riešiť, potom námety na diskusie, ale najmä niekoľko strán detailných **metodických poznámok**, alternatívnych (i nesprávnych) riešení, komentárov, najbežnejších žiackych chýb a pod. – jednoducho, metodická pomoc učiteľovi, pozri ukážku na obrázku 3.



Obrázok 4. Šesť alternatívnych riešení toho istého problému

Pre žiakov: Každé bádanie sme doplnili o prezentáciu určenú na vyučovaciu hodinu. Obsahuje stručné zadania krokov, ktoré učiteľ môže priebežne premietať na interaktívnej tabuli, aby tak získal čas venovať sa individuálne žiakom. Túto podporu učiteľ môže alebo nemusí využiť, prípadne si môže prezentáciu upraviť podľa vlastnej predstavy[7]. Prezentácie pre žiakov obsahujú rôzne otázky a námety na diskusiu, napr. sériu alternatívnych riešení (niekedy aj nesprávnych), ktoré sme pozbierali počas iteratívneho vývoja materiálov, pozri obrázok 4. Považujeme ich za vhodné na *uvažovanie* a *vysvetľovanie*, a teda na prehlbovanie porozumenia skúmaného javu a pod. Jednotlivé riešenia alebo kroky sme vždy očíslovali, aby sa na ne mohli žiaci a učitelia v diskusii ľahko odvolávať.

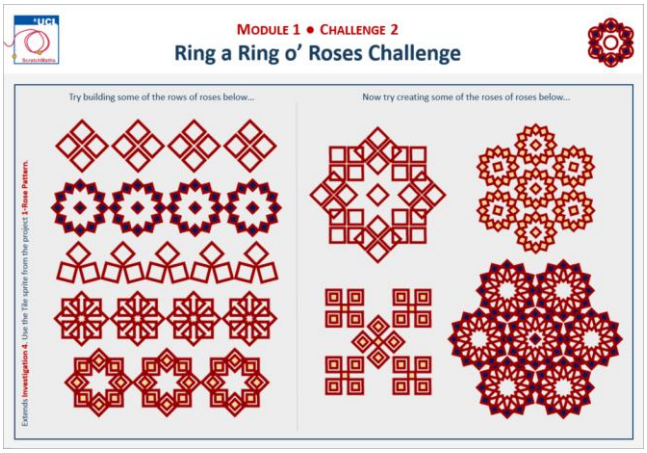
[5] Všetky dokumenty sú vo formáte vhodnom pre tlač a mnoho učiteľov si materiály skutočne tlačí.

[6] po anglicky *investigations*

[7] pri návštevách škôl sme reálne pozorovali každý z týchto troch prístupov

Doplňujúce materiály: K jednotlivým modulom sme tiež vytvorili plagáty s rôznymi užitočnými informáciami a pomôckami, napr. s paletou farieb a ich číselnými kódmi (ktoré sú, nanešťastie, v Scratchi nie celkom intuitívne), alebo vizuálnu pomôcku na vyjasnenie rozdielov medzi vnútornými, stredovými a vonkajšími uhlami pravidelných mnohouholníkov, obrázkový slovník programovacích blokov jazyka a pod. V projekte odporúčame učiteľom, aby tieto plagáty vytlačili a zavesili v triede na stenu.

Iný typ doplňujúcich materiálov tvoria jednostranové rozširujúce úlohy na tému blízku tej, ktorú žiaci a učiteľ práve riešia v danom bádaní, pozri obrázok 5. Keďže cieľom projektu je rozvíjať informatické a matematické myslenie každého žiaka (v miere, ktorá zodpovedá jeho potenciálu a záujmu), aktivity sa snažia vytvoriť pre nich **príležitosť pre *malé i veľké úspechy***. Chceme, aby mal každý možnosť urobiť správny krok v svojej *zóne najbližšieho vývinu*. To, prirodzene, kladie veľké nároky na organizáciu vyučovacej hodiny zo strany učiteľa, ktorý má podporovať poznávací proces širokej palety potenciálov žiakov. Znamená to nielen pozorne pracovať s menej zdatnými, ale aj primerane stimulovať tých najzdatnejších. Anglickí učitelia primárneho vzdelávania majú s týmto prístupom mnoho skúseností, preto ocenili naše rozširujúce zadania (*challenges*), ktoré vytlačené dávajú v triede na samostatné riešenie tým žiakom, ktorí by inak mohli na hodine stagnovať.



Obrázok 5. Jedno z rozširujúcich zadaní pre Modul 1

Úvodné projekty: Každá aktivita začína slovami: *Žiaci si otvoria projekt ...* alebo *Žiaci pokračujú vo svojej kópii projektu...* Tieto projekty sú na začiatku skoro celkom prázdne – vždy v nich však je jedna alebo niekoľko postáv s hotovými kostýmami a scéna s prázdnym či pripraveným pozadím. Niekedy je v projekte aj krátky inicializačný skript, prípadne niekoľko izolovaných blokov v ploche skriptov, výnimočne aj jedna alebo niekoľko definícií nových blokov, ktoré sme dočasne schovali vpravo „mimo obrazovku“ (nikdy sme však definície nepoužívali v zmysle „čiernej skrinky“). Naše východiská v tomto smere boli:

- v projekte sme nechceli strácať čas vyhradený na programovanie prácou s grafickým editorom Scratchu, ktorý nie je veľmi priateľský a – ako sme zistili pozorovaním žiakov a učiteľov pri návštevách škôl – prekvapujúco rýchlo sa v ňom žiaci dokážu dostať do nejasných a pre nich i pre učiteľa ťažko riešiteľných situácií,
- u žiakov sme chceli vytvoriť „kultúru“ inicializačných“ skriptov, čo sa nám ukázalo ako produktívne riešenie – žiaci si rozvíjali prístup *všetko mám pod svojou kontrolou*,
- pripravením jedného alebo niekoľkých izolovaných blokov sme podporovali stratégiu: *najprv priama manipulácia* (ak sa dá), *potom priama akcia, a až nakoniec riadenie programom*.

Ak majú žiaci s jedným projektom pracovať dlhšie ako jednu vyučovaciu hodinu, poskytli sme učiteľom aj projekty uložené v „medzistavoch“, teda projekty v stave, aký by asi žiaci dosiahli na konci predchádzajúcej hodiny. Aj toto rozhodnutie vyplynulo z našich praktických skúseností: ak žiak nebol prítomný na minulej hodine alebo projekt zle uložil alebo sa s jeho projektom či počítačom čosi prihodilo, mohol takto na ďalšej hodine pokračovať v práci spolu s ostatnými (prípadne s pomocou niektorého spolužiaka).

5.1 Riziká súvisiace s učebnými materiálmi

Vývoju a overovaniu materiálov sme v projekte venovali najväčšiu pozornosť a učitelia to aj často spomínajú a oceňujú. Napriek tomu sme si počas vývoja, testovania a školení uvedomili aj rôzne riziká, ktoré z nášho prístupu môžu vyplynúť. Tu spomenieme tri z nich.

Prvé riziko vyplýva z určitého protirečenia medzi *úplnosťou a rozsahom*. Učiteľ ocení, ak dostane kvalitné metodické materiály, ale ak obsahujú skutočne (skoro) všetko, čo môže byť pre neho užitočné, stanú sa neprakticky rozsiahle. Okrem toho, je veľký rozdiel medzi tým, čo učiteľ považuje v materiáloch za užitočné, keď s nimi pracuje prvý raz, a keď ich použije druhý či tretíkrát. Preto materiály zverejňujeme v rôznych formátoch, napr. aj ako PPT prezentácie, ktoré si učiteľ môže prípadne upraviť, skrátiť, alebo neskôr aj doplniť o vlastné rozširujúce aktivity a pod.

Druhé riziko označuje Brennan [7] ako protirečenie medzi *inštruovaním a objavovaním*, teda či má materiál (a celý obsah) formu detailných inštrukcií, alebo skôr poskytuje príležitosti pre voľné objavovanie, pre konštruovanie poznania samotnými žiakmi. Počas našich pozorovaní v triedach a na vzdelávaní sme si ako konštrukcionisti uvedomili vážnosť tohto protirečenia: Chceli by sme, aby učitelia, ktorí sú začiatočníkmi v programovaní a často zápasia aj s nízkou sebadôverou v tejto oblasti, budovali základy informatického myslenia a programovacích zručností žiakov postupmi, ktoré by buď boli konštrukcionistické, alebo aspoň obsahovali čo najviac konštrukcionistických príležitostí. Napokon sme v projekte zvolili takúto stratégiu: vzdelávacie materiály pre učiteľa sú do istej miery inštrukcionistické (aby učiteľ videl, že tento obsah dokáže realizovať), ale v každom module a vo väčšine bádaní je niekoľko príležitostí pre voľné objavovanie – avšak vždy s jasne stanoveným vzdelávacím cieľom a hodnotiacim kritériom. Pozorovali sme, že učiteľ často nedokáže produktívne usmerňovať poznávací proces žiakov, ak s nimi v triede robí jeden „dlhý voľný krok“, ale zvyčajne dokáže tento proces usmerňovať a podporovať, ak postupuje po „menších krokoch“, z ktorých niektoré sú voľným skúmaním a objavovaním s dobre definovaným cieľom. Ďalšou stratégiou bolo pre nás dôsledné dodržiavanie pedagogického rámca 5E, ktorý tiež podporuje objavovanie (a neustále reflektovanie a vysvetľovanie žiakmi) „po malých krokoch“. Okrem toho, všetky moduly pre 5. triedu končia otvoreným bádaním, ktoré môže žiakov a učiteľa inšpirovať k prekvapujúcim rozšíreniam. (Na jednej zo škôl sme napr. videli, že žiaci využili materiál Modulu 1 na výrobu papierových ozdôb na vianočný stromček potlačených vlastnými kruhovými vzormi.) Domnievame sa tiež, že miera voľného objavovania v práci s materiálom ScratchMaths bude výrazne vyššia, keď ho bude učiteľ realizovať druhý či tretí raz.

Napokon spomenieme ešte jedno vážne riziko: Niekedy sme pozorovali, že ak učiteľ dostal do ruky podrobné materiály, nevenoval dostatočnú pozornosť svojej vlastnej príprave. To niekedy viedlo

k nižšej alebo veľmi nízkej efektivite pri dosahovaní stanovených vzdelávacích cieľov. Stalo sa nám dokonca i to, že učiteľ si iba povrchne pozrel materiály a postupoval spôsobom, o ktorom sa povrchne domnieval, že je správny, zaviedol však seba i žiakov do „slepej uličky“. Takémuto javu hovoríme fatálna mutácia pôvodného zámeru autorov (*lethal mutation*) pri implementácii ich vzdelávacieho programu.

6. PRINCÍPY TVORBY OBSAHU

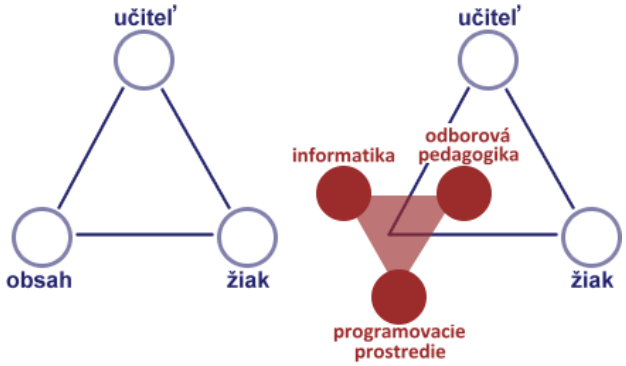
Projekt ScratchMaths je príkladom výskumu vývojom s cieľom iteratívne vyvinúť pedagogickú intervenciu, ktorá okrem vlastného nasadenia do škôl poslúži projektovému tímu aj ako inštrument pre kvalitatívny pedagogický výskum. Cieľom tohto výskumu je hlbšie porozumieť kauzálnym súvislostiam medzi rozvojom programovacích zručností (*computational thinking*) a matematického myslenia. Pozorovania učiteľov a žiakov v projektových školách nám umožňujú hĺbkovo skúmať rôzne javy, napr. aký vplyv má naprogramovanie vlastného modelu čísla zloženého z izolovaných číslic-postáv, ktoré spolu komunikujú posielaním správ, a porozumenia matematických vzťahov v pozičnej číselnej sústave, pozri [14].

Iným zaujímavým výstupom fázy sumatívneho vyhodnocovania projektu je prezentácia dôležitých princípov tvorby obsahu (*design principles*), ktoré sme identifikovali pri vývoji, overovaní a nasadzovaní nášho vzdelávacieho obsahu. Na ich usporiadanie a prezentáciu použijeme ‘tradičný’ didaktický trojuholník (pozri napr. [15] alebo [16]) s vrcholmi **učiteľ**, **obsah** a **žiak**, ktorý sa používa ako model pri skúmaní procesu učenia a učenia sa. My sa v kontexte edukačného programovania teraz zamerajme najmä na jeden jeho vrchol – **obsah**, a hrany, ktoré ho spájajú so zvyšnými vrcholmi. V [17] argumentujeme, prečo sme sa rozhodli na mieste tohto jediného vrcholu v skutočnosti rozlišovať **tri rôzne komponenty obsahu**, a to informatiku (*computing*), programovacie prostredie (*computational tool*) a odborovú didaktiku (*content specific pedagogy*), pozri obrázok 6. V spomínanej publikácii tiež upozorňujeme na to, aké je dôležité udržiavať pri tvorbe kurikula a vzdelávacieho obsahu rovnováhu v tomto sekundárnom trojuholníku: Ak by sme sa totiž vychýlili smerom k programovaciemu jazyku[8], hrozí deformácia, ktorú Papert [18] označuje ako *technocentrizmus*. Ak by sa tento stav vychýlil k informatike, môžeme analogicky hovoriť o akomsi *Computer Science centrizme*, ktorý skutočne hrozí, ak pod zástavou ‚moderného edukačného programovania pre každého‘ v skutočnosti sledujeme zvýšenie počtu budúcich univerzitných študentov informatiky. Napokon, vychýlenie smerom k odborovej didaktike môže viesť k plytkému alebo žiadnemu porozumeniu programovacích konštruktov (aké často nachádzame v populárnych učebniciach programovania).

Pozrime sa teraz na niekoľko princípov tvorby obsahu v takomto členení.

6.1 Veľké myšlienky informatiky

Ak je naším cieľom udržateľne implementovať edukačné programovanie ako úspešný a rešpektovaný predmet pre každého žiaka – a to už na základnej škole, musíme starostlivo zvažovať *výber, rozsah a formát* informatických konštruktov a ich *vývinovú primeranosť*. V projekte ScratchMaths náš obsah realizujú všeobecní učitelia primárneho vzdelávania, preto sme pojem primeranosti museli zvažovať nielen z pohľadu kognitívneho vývinu žiakov, ale aj z pohľadu kvalifikácie ich učiteľov, u ktorých nemôžeme predpokladať a ani vyžadovať akékoľvek odborné informatické vzdelanie (prekvapili nás však vysokou úrovňou svojej digitálnej gramotnosti). Preto sme sa rozhodli celý vývoj stavať na produktívnych a motivujúcich *konkrétnych zážitkoch* žiakov i učiteľov (*toto som dokázal spraviť... vyriešiť...*; *toto mojich žiakov teší...; dokážeme stále viac a viac; moji žiaci teraz lepšie rozumejú aj tomu a tomu z predmetu...; moji žiaci sa tešia na ďalší krok, objav, problém...*), na dôkladne premyslenej postupnosti aktivít, ktorou sa (často paralelne) vinú rôzne trajektórie budovania konceptov a postupov. Používame starostlivo zvolený a primeraný informatický slovník, informatické konštrukty nikdy vopred nevysvetľujeme, iba ich v diskusiách správne pomenúvame – avšak zásadne až potom, keď ich už žiaci sami zažijú alebo objavia.



Obrázok 6. Didaktický trojuholník pre edukačné programovanie

Na druhej strane však my, ako autori obsahu a vzdelávania, vedome budujeme celý materiál na starostlivo zvolených, premyslených a zoradených *veľkých myšlienkach informatiky*. Nepoužívame ich teda ako učebný cieľ, ale ako **organizačný princíp** pre nás samotných. Vytvárame kontexty, v ktorých sa žiaci učia vnímať postavy ako objekty, ktoré sami riadia, izolovaný blok ako prostriedok na dosiahnutie určitej reakcie a ako stavebný materiál na budovanie skriptu, skript ako plán či program pre budúce správanie, nový blok ako nové meno pre správanie, ktoré predtým vytvorili, posielanie správ ako nástroj na spoluprácu a pod.

V kapitole 4 sme projekt prezentovali z pohľadu modulov. Z pohľadu informatiky by sme celý vzdelávací obsah (a tiež štruktúru zodpovedajúceho vzdelávania učiteľov) mohli stručne charakterizovať súborom (nie postupnosťou) *veľkých myšlienok*:

- používame priamu akciu; upravujeme vstup(y) pre blok
- vytvárame skript(y) ako vyjadrenie budúceho správania
- vykonávame skripty; upravujeme skripty; vyvolávame udalosti
- vytvárame produkty a riadime deje opakovaním krokov
- rozširujeme jazyk, abstrahujeme (odstupujeme) od detailov
- využívame náhodnosť, definujeme pre ňu obmedzenia
- vytvárame spoluprácu postáv; paralelizmus; posielanie správ
- skúmame podmienky; obmedzujeme opakovanie podmienkou
- počítame hodnoty: výrazy
- dávame hodnotám mená: vstupy, dáta, premenné

[8] alebo akejkoľvek inej digitálnej technológii

6.2 Primeranosť programovacieho prostredia

Za rozhodujúce tu považujeme kvalifikovane posúdiť primeranosť zvoleného programovacieho jazyka a prostredia. To nám pomôže dodržať nasledujúce princípy:

Nezačnime s jazykom priskoro. V [19] a [17] sme vyjadrili presvedčenie, že primárne programovanie (teda programovanie pre žiakov približne vo veku od 5 do 11 rokov) poskytuje takú bohatú paletu vývinovo primeraných námetov a informatických konštruktov, že začať používať niektorý jazyk – napr. Scratch – skôr, než je vhodné, predstavuje dvojnásobné ‚ohrozenie' (*double leathal timing*): Ak začneme s jazykom a prostredím priskoro, vytlačíme tým z vyhradeného času príležitosti pre rozvoj základných poznatkov a aktivity, ktoré tvoria vývinovo primeraný základ a obsah primárneho programovania. Okrem toho však ešte aj ohrozíme záujem žiakov o prácu v tomto prostredí, keď už preň dosiahnu vhodný vek, ale môžu mať pocit, že... *toto sme už predsa robili*[9].

Existuje však aj opačné riziko: *Nezačnime s jazykom prineskoro*, pretože žiaci neradi pracujú s nástrojmi a pomôckami, ktoré považujú za „detské", teda za vhodné pre nižšie vekové kategórie.

S tým súvisí ešte jeden princíp: *Nepracujme s tým istým jazykom či prostredím pridlho*, pretože tým znižujeme motiváciu žiakov o objavovanie a poznávanie v „novom a nepoznanom". Naviac sa pripravujeme o kľúčovo dôležité príležitosti rozvíjať mieru porozumenia informatických konštruktov prostredníctvom ich **riadenej transformácie z prostredia do prostredia** – *Ako sme vyjadrovali takúto podmienku v ... Ako by sme čosi podobné urobili v tomto novom prostredí...?*

Vo vrchole **programovacie prostredie** sekundárneho didaktického trojuholníka z obrázka 6 chceme spomenúť ešte jeden dôležitý princíp: *Poznajme zvolené prostredie do hĺbky, aby sme dokázali rozpoznať a využiť jeho „vynikajúce vlastnosti"* (tzv. *affordances*) a aby sme sa dokázali vyhnúť jeho neproduktívnym alebo nie celkom vhodným prvkom. Stručne tu len konštatujme, že Scratch má viacero reprezentantov v oboch z týchto extrémnych polôh.

6.3 Niekoľko postrehov z odborovej didaktiky

Organizujúcim inštrumentom v tejto oblasti je pre nás dôsledné používanie všetkých *princípov pedagogického rámca 5E*, teda zaraďovanie takých aktivít, ktoré žiakom umožnia *skúmať*, *vysvetľovať*, *uvažovať*, *zdieľať* aj *prepájať*. Tu je niekoľko záverov:

- Postupujme od *priamej manipulácie* cez *priamu akciu* k *riadeniu programom*, od izolovaného k zloženému, od jedného (príkazu, vstupu, postavy, kostýmu...) k dvom, prípadne viacerým.
- Uistime sa, že žiak rozumie každému bloku skôr, než ho použije v skripte (napr. skúmaním priamou akciou).
- Používajme úvodné projekty ako *scaffolding* pri vyučovaní.
- Budujme u žiakov potrebu nového konštruktu, aby ho mohli sami objaviť a oceniť.
- Umožnime žiakom konštrukt najskôr zažiť, až potom o ňom diskutovať.
- Ku každému konštruktu sa opakovane vracajme v rôznych kontextoch.
- Veďme žiakov k tomu, aby skripty vysvetľovali a uvažovali o nich aj *blok po bloku*, aj ako *o celku*.
- Zbierajme a využívajme „zaujímavé chyby" (učitelia v Anglicku im hovoria *dobré chyby*).
- Dôsledne používajme jednotnú terminológiu, dbajme na presné formulácie. Používajme silné metafory (ako napr. *posun* pre pohyby v štýle korytnačej geometrie a *skok* pre pohyby zmenou jednej či oboch súradníc).
- Rozvíjajme u žiakov rôzne stratégie na skúmanie a ladenie (*debugging*) scriptov, napr. dekompozíciou na menšie časti alebo vložením niekoľkých **wait** blokov.
- Podporujme u žiakov rôzne prístupy k riešeniu, dajme im príležitosť, aby mohli svoje postupy prezentovať a vysvetliť ostatným.
- Opakovane pri diskusii pomenúvajme veľké myšlienky informatiky.
- Používajme rôznorodé aktivity „bez klávesnice".
- Nechajme žiakov *zahrať program* v úlohe postavy (*syntonicity*).
- Používajme rôzne papierové a iné pomôcky (napr. aj drevené hračky a pod.), podporujme žiakov, aby robili to isté.
- Podporujme a vytvárajte príležitosti pre spoluprácu, napr. prácu v dvojici, vzájomné učenie sa (napr. metódou dobrých otázok).
- Podporujme výmenu postáv, kostýmov, pozadí, skriptov alebo celých riešení.
- Organizujme výstavy produktov, nástenky, súťaže, prezentácie.
- Využime každú možnosť prepájania, napr. s matematikou, históriou, výtvarným umením a pod.

7. ZÁVER

Našim cieľom bolo v tomto príspevku prezentovať niektoré aspekty projektu ScratchMaths, s dôrazom na pedagogický rámec, ktorým sme sa riadili a ktorý reguloval vývoj celého obsahu a vzdelávanie učiteľov, na štruktúru a charakterizáciu vytvoreného obsahu, a napokon na prezentáciu niekoľkých princípov tvorby obsahu pre edukačné programovanie žiakov vo veku približne 9 až 11 rokov. Sme si vedomí toho, že na úspešné šírenie nášho obsahu bude mať vplyv veľa rôznorodých a ťažko predvídateľný faktorov. Veríme však, že materiály a pedagogika projektu ScratchMaths poslúžia ako produktívny inštrument zberu dát pre nasledujúce pedagogické výskumy v tejto oblasti a že niektoré princípy tvorby, ktoré sme identifikovali, poslúžia aj pri ďalšom vývoji iných vzdelávacích obsahov pre primárne alebo sekundárne programovanie.

POĎAKOVANIE

Ďakujem projektovému tímu ScratchMaths z UCL Knowledge Lab, profesorom R. Nossovi a C. Hoyles a kolegom L. Benton, P. Saunders a J. Carvajal, a tiež učiteľom a žiakom z našich vývojových škôl za jedinečnú atmosféru, pomoc a chuť učiť sa a učiť ma. Ďakujem nadácii EEF za financovanie projektu.

BIBLIOGRAFICKÉ ODKAZY

[1] THE ROYAL SOCIETY. *Shut down or restart? The way forward for computing*. 2012, 121 p. Dostupné na https://royalsociety.org/~/media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf

[2] PASSEY, D. 2016 Passey, D. Computer science (CS) in the compulsory education curriculum: implications for future

[9] Aj keď takáto reakcia žiakov nie je zvyčajne opodstatnená, nič to nemení na negatívnom vplyve, ktorý má takáto voľba na ich motiváciu. Tento náš uzáver sa opiera iba o izolované pozorovania (*anecdotal evidence*), identickú skúsenosť nám však potvrdili viacerí edukátori z rôznych krajín.

research. *Education and Information Technologies*: 2016, pp. 1-23.

[3] DEPARTMENT FOR EDUCATION. Computing programmes of study: key stage 1 and 2. National curriculum in England 2013. Dostupné na https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study

[4] NIEVEEN, N., FOLMER, E. Formative Evaluation in Educational Design Research. In: Plomp, T., Nieveen, N.: *Educational design research*. SLO – Netherlands institute for curriculum development: pp. 152-169.

[5] DiSESSA, A.A., COBB, P. Ontological Innovation and the Role of Theory in Design Experiments. *Journal of the Learning Sciences*, 2004: 13 (1), pp. 77-103.

[6] BERS, M. U. et al. Computational thinkng and tinkering: Exploration of an early children robotics curriculum. *Computers and Education*, 2014, 72, pp. 145-157.

[7] BRENNAN, K. Beyond Technocentrism: Supporting Constructionism in the Classroom. *Constructivist Foundations*, 2015: 10 (3), pp. 289-296.

[8] BLACKWELL, A.F. What is Programming? In *14th Workshop of the Psychology of Programming Interest Group*., 2002, pp. 204–218.

[9] HAREL, I., PAPERT, S. Software design as a learning environment. *Interactive Learning Environments*, 1990, 1, pp. 1-32.

[10] NOSS, R., HOYLES, C. *Windows on Mathematical Meanings: Learning Cultures and Computers*. 1996, Kluwer Academic Publishers, 275 p.

[11] WATT, S. Syntonicity and the psychology of programming. In Proceedings of 10th Annual Meeting of the Psychology of Programming Interest Group. 1998, pp. 75–86.

[12] BRENNAN, K., RESNICK, M. New frameworks for studying and assessing the development of computational thinking. Proc. of the *Annual meeting of the Americal Educational Research Association*, 2012, Vancouver.

[13] PAPERT, S. Mindstorms: Children, computers, and powerful ideas. Basic Books, 1980.

[14] BENTON, L., HOYLES, C., KALAS, I., NOSS, R. Bridging primary programming and mathematics: some findings of design research in England. Accepted for publication in *Digital Experiences in Mathematics Education*.

[15] KANSANEN, P. Teaching as teaching-studying-learning interaction. *Scandinavian Journal of Educational Research*, 1999: 43 (1), pp. 81-89.

[16] BERLUND, A., LISTER, R. Introductory Programming and the Didactic Triangle. Proc. 12th *Australasian Computing Education Conference*, 2010, Brisbane, pp. 35-44.

[17] KALAŠ, I. On the road to sustainable primary programming. Proc. of *Constructionism in Action*, Bangkok 2016, pp. 184-191.

[18] PAPERT, S. Information technology and education: computer criticism vs. technocentric thinking. Educational Research 1987: 16 (1), pp. 22-30.

[19] KABÁTOVÁ, M. KALAŠ, I., TOMCSÁNYIOVÁ, M. (2016). Programming in Slovak primary schools. *Olympiads in Informatics* Vol. 10 (2016), pp. 125-158.

Príspevky

Papers

PRIESTOROVÁ ORIENTÁCIA A ORIENTÁCIA V PC
SPATIAL ORIENTATION AND ORIENTATION IN PC

Brigita Albertova
RZSP Levoča
Kasárenská 16
054 01 Levoča
Slovensko
brigita.albertova@gmail.com

Peter Nedorost
RZSP Levoča
Kasárenská 16
054 01 Levoča
Slovensko
peter.nedorost@nextra.sk

Jaroslav Kultan
EU v Bratislave
Dolnozemská cesta 1
85335 Bratislava
Slovensko
jkultan@gmail.com

ABSTRAKT

Orientácia v počítači je pre nevidiaceho tak isto problematická ako aj orientácia v mikro alebo makro priestore. Problémy, ktoré má nevidiaci človek pri orientácii na stole, sú tak isté ako aj pri orientácií v počítači. Odlišnosťou je fakt, že objekty na stole sú priestorové a je možné ich nahmatať, a objekty v počítači sú virtuálne a môže o nich vedieť hlavne vďaka hlasovej informácií, ktorú vydáva príslušný program.

Článok je zameraný na využívanie postupov používaných pri vyučovaní priestorovej orientácie (mikroorientácie) pri vyučovaní informačných technológií.

Návyky, ktoré nevidiaci získajú pri orientácií na pracovnom stole, spôsoby prehľadávania jednotlivých častí a návyky o uložení jednotlivých predmetov, je možné využití pri práci s počítačom. Jednotlivé ikony, ktoré charakterizujú programy, je možné rozložiť podľa ich významu.

To, čo je pre vidiaceho samozrejmé, to pre nevidiaceho je veľmi ťažké alebo až nemožné. Jedným z hlavných problémov pre nich v bežnom živote je strata orientácie, možnosti voľne sa presunúť z miesta na miesto. Zvládnutie týchto problémov je možné čiastočne využiť na aktívne využívanie IT.

ABSTRACT

Orientation in a computer for blind also problematic and orientation in micro and macro space. The problems with orientation blind on the table are similar to the orientation of the blind in the computer. However, there are several differences. Objects on the table are space and can be palpated. Objects on a computer are virtual and can know about them mainly because the voice information issued by the program.

The article is focused on the use of procedures used when teaching spatial orientation (micro orientation) in teaching information technology. Experience gained in the orientation on a desk, individual ways to browse various parts and habits to save individual objects, you can use when working with your computer.

The icons that characterize the programs can be distributed according to their importance.

What is the seer goes without saying that the blind is very difficult or even impossible. One of the main problems for them in ordinary life, disorientation, opportunities to move freely from place to place. Solving these problems can be partly used for the active use of IT

Kľúčové slová

Informačné technológie (IT); priestorová orientácia, zdravotne postihnutí; nevidiaci , čítač obrazovky, syntetizátor reči

Keywords

Information technology (IT); quality of education; teaching methodology; disabled; blind, screen reader, speech synthesizer

1. ÚVOD

Jedným so základných predpokladov zvyšovania kvality života nevidiacich je získanie základných alebo aj pokročilých znalostí práce s počítačom. Jedným z hlavným problémom nevidiacich pri práci s počítačom je orientácia na obrazovke. To čo je pre vidiaceho normálne, to pre nevidiaceho je veľký problém. Nevidiaci nevidí rozloženie ikon na obrazovke. Programy ozvučujúce jednotlivé ikony hovoria o tom, ako sa volajú jednotlivé programy. Avšak vyhľadávanie toho pravého programu môže byť dlhý a náročný proces. Práve preto navrhujeme pri tejto činnosti využívať niektoré aspekty vyučovania mikro orientácie na stole.

Pri strate zraku nevidiaci má taký istý problém pri hľadaní vecí na stole, ako aj pri prehľadávaní obrazovky. Práve osvojenie si princípov mikro orientácie pri práci na stole a ich využití pri práci na pracovnom stole, sa môže stať pomôckou pri priestorovej orientácií na pracovnom stole počítača.

2. MIKRO ORIENTÁCIA NA STOLE

Hlavnými problémami nevidiacich je nielen problém orientácie pri pohybe v meste, orientácie v miestnosti ale aj orientácie na malom priestore. Jedná sa o vyhľadávanie predmetov na stole, ich správne uloženie na stole tak, aby proces vyhľadávania bol minimálny. Súhlasíme s tvrdením Jesenského [1], že orientáciu považujeme za východiskový prvok v oblasti poznávacieho procesu zrakovo postihnutých. Wiener [2 str.16] vymedzuje orientáciu ako „proces získavania a spracovávania informácií z prostredia za účelom skutočnej alebo len myšlienkovej manipulácie s objektmi priestoru alebo za účelom plánovania a realizácie premiestňovania sa v priestore “. Orientáciu môžeme charakterizovať orientáciu ako „proces, pri ktorom človek podľa nejakého systému súradníc určí svoju polohu v priestore“. Pri orientácii v priestore je nevyhnutné riešenie troch úloh, a to voľba smeru, udržanie smeru a objavenie cieľa, pričom riešenie uvedených úloh sa týka orientácie tak vo veľkom ako aj malom priestore. Podľa Jesenského [3] je nielen z praktického hľadiska, ale tiež na účely riešenia úloh orientácie vhodné diferencovať priestor na makropriestor a mikropriestor.

V súvislosti s tým sa v literatúre stretávame s pojmami makroorientácia a mikroorientácia.

2.1 Mikroorientácia

Začína sa práve od mikroorientácie, aj keď sa na prvý pohľad zdá, že je to presne naopak, že začiatok je od makropriestoru. Keď človek stratí zrak v neskoršom veku, je pre neho problematické dostať sa samostatne na miesta ako predtým. Avšak, aby pochopil spôsob pohybu, je potrené začínať postupne od stola.

Aj zdravému človeku sa stáva, že pri zhasnutom svetle hľadá rukou predmet, ktorý nechal na stole. Zrakovo postihnutí nemôžu takým chaotickým spôsobom hľadať každodenné veci, a to nehovoriac o potravinách na stole, horúcej káve alebo polievke. Je veľmi dôležité pochopiť systém postupnosti vyhľadávania na malom priestore. To isté platí aj pre veľké priestory aj keď sa to zdá, že je to úplne odlišné. Malé predmety hľadáme v pásoch. Ruky dáme k okrajom stola a približujeme k sebe. Po tej istej línii naspäť zasa k okraju len o niečo vyššie a zas k sebe [4]. Z akého dôvodu? Aby sa nám nestalo, že preskočíme úsek a práve tam budú tie kľúče alebo maličký predmet. Ako sa k tomu dopracovať? Hrou. Práve pri rôznych činnostiach, ktoré sme robili, keď sme boli malé deti [5]. Ťažko sa niekedy takáto situácia prijíma. Vtedy, ako deti, to mnohí nevidiaci vedeli možno lepšie ako teraz. Používame rôzne makety z rôzneho materiálu a do nich sa ukladajú predmety. Od týchto hier prechádzame *na reliéfne rozlíšenie rôznych geometrických tvarov, čiar, bodkočiarok, až do jedného bodu* (bodky). Prečo? Takéto bodky a čiarky sú potrebné, aby bola jemná motorika na vyššej úrovni.

2.2 Proces vyučovania mikroorientácie

Poznanie blízkeho priestoru a jeho spracovanie ide ruka v ruke s poznaním, ktoré presahuje pole kontaktného analyzátora. Mikroorientácia je podmieňovaná charakterom konkrétne angažovaného analyzátora. U kontaktného analyzátora hmatového (napr. ruky) je vymedzená rozsahom bimanuálneho alebo aj monomanuálneho hmatového poľa“ [4, s. 16].

Jesenský [3, s. 585-586)] uvádza, že „mikropriestor predstavuje pole nášho vnímania a konania, pri ktorom naše premiestňovanie nepresahuje viac než 3-5 krokov. Takýto charakter má väčšina sebaobslužných činností, práca v kuchyni, za pracovným stolom alebo strojom. Tzn. mikropriestor je poľom aktivít funkčnej gramotnosti, denných činností i manuálnych a duševných prác“. Pod aktivity v oblasti mikropriestoru zahŕňame: výcvik čítania a písania Braillovho písma, výcvik v tyflografike, výcvik práce s počítačom, výcvik funkčnej gramotnosti a pod [6]. Pod mikropriestor v školských podmienkach zahŕňame školskú lavicu, podložku a papier v stroji pre nevidiacich.

Pre účely tohto príspevku sa obmedzíme hlavne na orientáciu na školskej lavici, stole, podložke a prácu s reliéfnymi obrázkami.

A. Orientácia na školskej lavici, podložke:

Pri práci na školskej lavici alebo pracovnom stole sa jedná hlavne o pochopenie a nacvičovanie pojmov vľavo a vpravo, vpredu a vzadu, hore a dole [7]. Okrem toho je potrebné nacvičiť ukladanie malých predmetov na lavicu alebo podložku podľa príkazu učiteľa, tiredenie daného materiálu v súvislosti s priestorovými pojmami. Veľmi dôležitým je postu nacvičovania pohybu ruky po riadkoch zľava a doprava, pri čítaní prechod do nového riadku, nacvičovanie orientácie v texte, lokalizácie označovania cvičení a príkladov, číslovania strán. Dôležitú úloho má nacvičovanie schopnosti používať písací stroj nevidiacich.

Pri vyučovaní jednotlivých prvkov orientácie, je možné podrobnejšie rozložiť jednotlivé základné pojmy, ktoré používame pri riadení pohybu v procese vyhľadávania [8]. Jednotlivé pokyny a pohyby je možné využiť pri práci na počítači. Reč ide hlavne o príkazoch umožňujúcich prácu v nasledovných prípadoch.

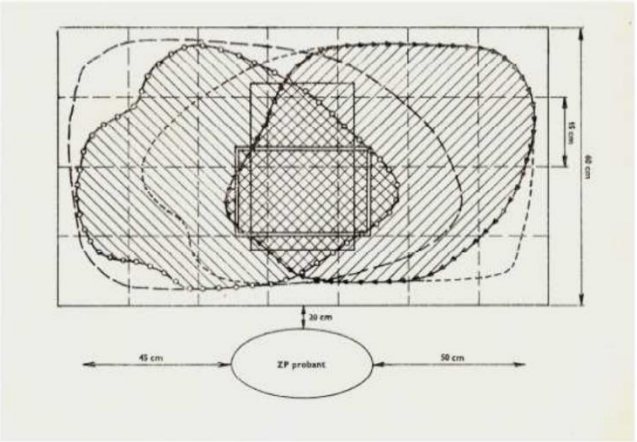
1. Práca v štyroch kvadrantoch
2. Určovanie polohy v rade predmetov, prvkov.
3. Určovanie polohy v štvorcovej sieti, na jednoduchom obrázku
4. Ukladanie predmetov podľa veľkosti
5. Určovanie polohy predmetov v stĺpcovej a tabuľkovej zostave na stole, na lavici, na podložke
6. Porovnávanie ciest vytvorených na podložke z kartónu a z rozličných materiálov, nakreslených na bielom papieri výraznou fixkou
7. Plánovanie cesty, bludiská vytvorené rozličnými technikami
8. Dokresľovanie, doplnenie druhej časti obrázka pomocou plastelíny, modelovacej hmoty, tvarovacieho špagátika
9. Porovnávanie množstva viac, menej, najviac, najmenej
10. Priraďovanie a vyhľadávanie odlišných, rovnakých predmetov, prvkov

B. Práca s reliéfnymi obrázkami

Pri práci s reliéfnymi obrázka sa zameriavame hlavne na spoznávanie jednoduchých tyflografických prvkov: dlhá, krátka, zvislá, vodorovná, šikmá čiara, vlnovka, horný a dolný oblúčik, kruh, šípka... Okrem toho na spoznávanie jednoduchých reliéfnych techník a nacvičovanie schopností čítať reliéfne obrázky, skladanie i vytváranie obrazov s pomocou využitia tyflografických prvkov. Dôležitým faktom je nacvičovanie kreslenia reliéfnych obrazov, použitia tyflografiky na čítanie a aj písanie latinky, použitia tyflografiky pri didaktických hrách.

2.3 Vyučovanie mikroorientácie pomocou reliéfnej mapy

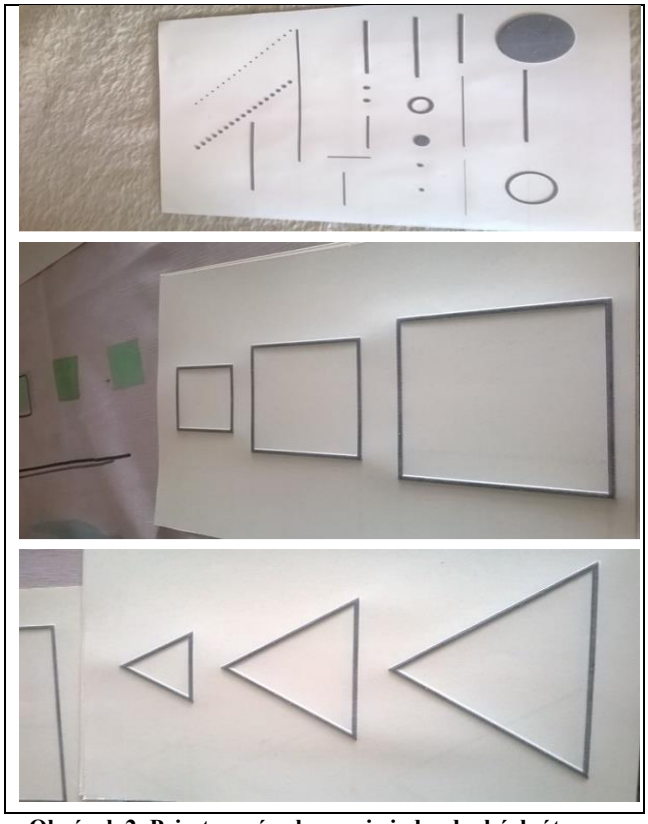
Ako bolo uvedené v prechádzajúcich častiach, základom správnej orientácie na pracovnej ploche je presné nacvičenie základných pojmov a ich implementácia do hmatovej škály nevidiaceho. Počas krátkeho obdobia je nevyhnutné vytvoriť systém návykov, ktoré umožnia ľahko a jasne rozlíšiť jednotlivé časti plochy, rozdeliť ich na sektory (Obrázok 1).



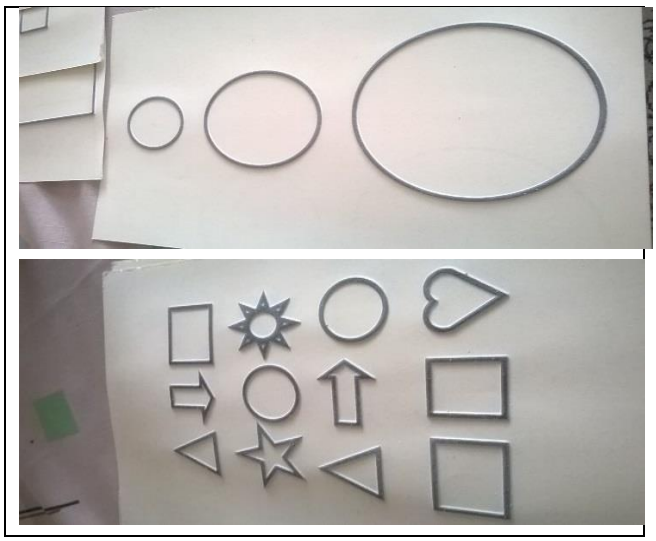
Obrázok 1. Hmatové pole

Pritom je potrebné si vytvoriť pojmový aparát nielen na základnú množinu, ale aj podmnožiny objektov, ktoré sa môžu nachádzať

v pravouholníkových, trojuholníkových, kruhových alebo nepravidelných ohraničeniach. Príklad takého postupného získavania zručností sú jednoduché tvary vytlačené na reliéfnej tlačiarni (Obrázok 2). Na začiatku sa jedná o tréning rovného pohybu, následne jednoduché geometrické tvary – obdĺžnik, trojuholník, kruh. Následne využívame plochu s viacerými tvarmi. Vrcholom danej činnosti je čítanie priestorovej mapy (Obrázok 3). Pri tomto nácviku nevidiaci musí si nielen zapamätať jednotlivé tvary, ale aj ich polohu v rámci celej mapy a vzájomnú polohu jednotlivých objektov [9]. Z takýchto objektov je potom možné vytvoriť hmatové pole ako je uvedené vyššie.



Obrázok 2. Priestorové zobrazenie jednoduchých útvarov



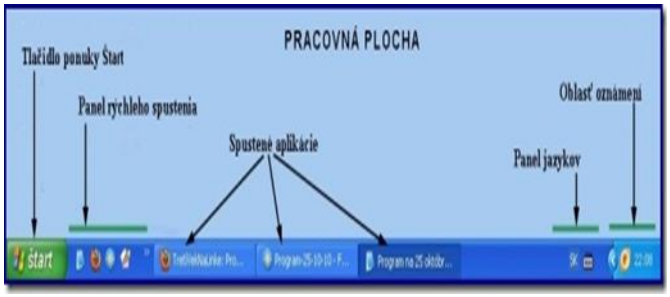
Obrázok 3. Priestorové zobrazenie zložitých útvarov

3. ZÁKLADY ORIENTÁCIE NA OBRAZOVKE POČÍTAČA

Základom úspechu akéhokoľvek projektu, do ktorého sa pustíme je dobrá organizácia. To platí obzvlášť pre nevidiacich a hlavne v práci s osobným počítačom. Nejde iba o organizáciu používateľských dát, ale aj o správnu organizáciu v nastaveniach operačného systému alebo čítača obrazovky, ktorý interpretuje informácie z obrazovky nevidiacemu používateľovi. Hovorí sa, že ak chceme získať predstavu o kvalitách a zvyklostiach používateľa osobného počítača, stačí sa pozrieť na jeho pracovnú plochu. Táto tvorí akýsi výklad počítača a trocha aj zvykov či zlozvykov používateľa.

3.1 Základné predpoklady pre vyučovanie na počítači

Kým vidiacemu stačí často jeden pohľad na plochu, aby ju celú obsiahol, pre nevidiacich plocha počítača je stolom s mnohými predmetmi, ktoré si musia „ohmatávať“ so zatvorenými očami. Aby to bolo ešte zložitejšie, ide vlastne o stôl s niekoľkými poličkami, pretože máme časť plochy s ikonami, ďalšiu časť plochy tvorí systémový panel, atď (Obrázok 4). Preto musí nevidiaci vedieť, ako sa tabulátorom medzi časťami plochy prepínať, využívať rôzne pomocné skratky operačného systému, napr. Windows ***b – prejdenie na systémový panel*** a podobne.



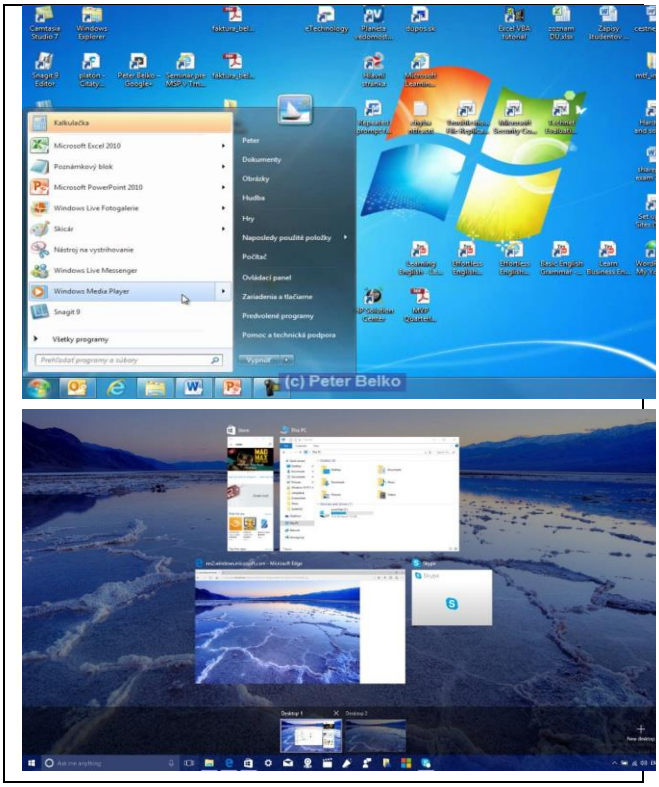
Obrázok 4 Popis jednotlivých skupín ovládacích prvkov

Ak vieme, čo sa na pracovnej ploche nachádza, môžeme sa na daný objekt (ak sme v správnej časti) rýchlo dostať stlačením jedného (alebo aj viacerých) písmen, čo navigáciu veľmi uľahčuje. Nastavenie si klávesových skratiek môže byť užitočným nástrojom, ako svoju rýchlosť pri práci ešte viac zefektívniť. Treba si však uvedomiť, že mnoho klávesových skratiek je už použitých na úrovni operačného systému, mnoho ďalších zaberajú jednotlivé aplikácia, čítač obrazovky ako taký a ak sa nevidiaci vycvičí na používanie vlastných skratiek, pri inom počítači kde zrazu nebudú definované bude ťažko odbúravať zlozvyk rýchlo otvoriť prehliadač napríklad skratkou ***ctrl shift f,...***

S navigačnými bariérami sa stretávame aj na hlbších úrovniach operačného systému. Spomeňme si len, akými rôznymi podobami prešla ponuka štart od Windows95 po Windows10. V podstate

každá verzia sa správa inak a treba sa naučiť určité postupy pre orientáciu. Ešte krikľavejším príkladom úplne zmenenej koncepcie orientácie je zmena ovládania a vyhľadávanie programov, ktorú Microsoft zaviedol v niektorých aplikáciách. A čo povedať na adresu moderného usporiadania jednotlivých aplikácií (Obrázok 5)? S týmto všetkým musí nevidiaci používateľ zápasiť.

Mnoho vecí sa dá zvládnuť jednoduchými pomôckami, možno si napríklad nastaviť zobrazenie objektov v zoznamoch miesto dlaždíc. Aj také novinky ako napr. centrum oznámení vo Windows 10 idú úspešne zvládnuť, samozrejme si nevidiaci musí nastaviť omnoho väčší čas zobrazenia oznámení na tomto paneli



Obrázok 5. Zobrazenie viacerých obrazoviek sytému Windows

Aj nevidiaci musí bojovať s roztrieštenosťou, ktorú zaviedol Microsoft do svojho aktuálneho operačného systému, kde sa niečo nastavuje postarom cez ovládací panel a niečo v modernom centre nastavení. Tu boli navyše zavedené nové ovládacie prvky. Niekedy je dokonca výhodnejšie siahnuť po starých dialógoch pre nastavenie niečoho, čo ide síce nastaviť aj v modernom centre nastavení avšak z hľadiska prístupnosti pre nevidiaceho ide o krok späť.

3.2 Základné kroky vyučovania nevidiacich

V našom zariadení (RSZP v Levoči) praktickej výuke práce s počítačom predchádza teoretická príprava. V základnom kurze naši frekventanti absolvujú predmet písanie na kancelárskom stroji, ktorý v dnešnej dobe už prebieha na počítačovej klávesnici. Tak sa naučia prstoklad a sú pripravení absolvovať kurz práce s počítačom.

Výber vhodného vstupného zariadenia – klávesnice

Pred samotným začiatkom práce s počítačom sa musia frekventanti dokonale zoznámiť so svojou pracovnou stanicou. Riešia sa veci, ktoré by vidiaceho používateľa nemuseli ani napadnúť. Na trhu je veľa druhov klávesníc. Nie všetky sú pre prácu nevidiaceho s počítačom vhodné. Napr. klávesnice, ktoré majú nad hornou šípkou tri tlačidlá, z ktorých práve to nad šípkou počítač po stlačení uspáva. Mnoho klávesníc je tiež riešených tak, že klávesy sú natlačené na úzkom priestore. Ešte horšia situácia je s notebookmi, kde sa už tiež klávesnice riešia ako jeden veľký obdĺžnik. Mnoho klávesníc na notebookoch prestáva mať základné klávesy pre prácu s čítačom obrazovky ako je insert, či kláves kontextového menu. Funkčné klávesy v hornom rade sú nahrádzané terciárnymi funkciami a pôvodné funkcie treba simulovať tlačidlom fn. Tieto všetky fakty, a mnohé iné, treba zobrať do úvahy a čo najviac prispôsobiť potrebám frekventantov pred začatím výuky.

Prvotné nastavenie operačného systému

Pred samotným začatím výuky je tiež dôležité prvotné nastavenie operačného systému, zobrazenie objektov v zoznamoch, vypnutie zoskupovania objektov, zapnutie zobrazovania prípon súborov a pod. Kto má ťažkú ruku, tiež zabránenie opakovaného písania klávesy pri jej dlhšom stlačení. Tiež si treba vštepiť pravidlá stláčania kláves, napr. býva častými chybami stlačenie klávesovej skratky ***ctrl p*** naraz dvoma rukami. Môže sa totiž stať, že p je stlačené o desatinu sekundy skôr ako kláves ctrl, čím sa nevyvolá dialógové okno pre tlač dokumentu, ale sa do dokumentu napíše písmeno p.

Úvod do vyučovania

Samotná výuka začína od najjednoduchších krokov, teda pohyb po pracovnej ploche. Pri práci s počítačom to znamená, že musia vedieť si nájsť potrebnú ikonu, otváranie a zatváranie aplikácie, prepínanie sa medzi viacerými otvorenými aplikáciami. Cez to všetko sa musia naučiť pracovať aj so svojim čítačom obrazovky. Musia sa naučiť čítať klávesové skratky pre prečítanie názvu okna, v ktorom sa nachádzajú, vyvolanie nastavení čítača, kde si nastavujú parametre hlasu ako je osoba, výška, rýchlosť a podobne. Dôležité je naučiť sa pracovať s dialógovými oknami, pohyb po jednotlivých prvkoch a ich ovládanie, pochopiť štruktúru viac kartových dialógových okien. Napokon sa učí práca s konkrétnymi aplikáciami, word, excel a pod.

4. NÁVRH METODIKY VYUŽÍVANIA MIKROORIENTÁCIE PRI ORIENTÁCII V PC

Ako bolo uvedené v kapitole 3, práca s počítačom pre nevidiaceho je omnoho zložitejšia ako pre vidiaceho. Hlavným problémom je samotná orientácia na pracovnej ploche počítača, prepojenie jednotlivých úrovní, vyhľadávanie základných kláves a pod.

Aby obyčajný osobný počítač sa stal použiteľný pre nevidiaceho človeka, je potrebné inštalovať naň dva programy:

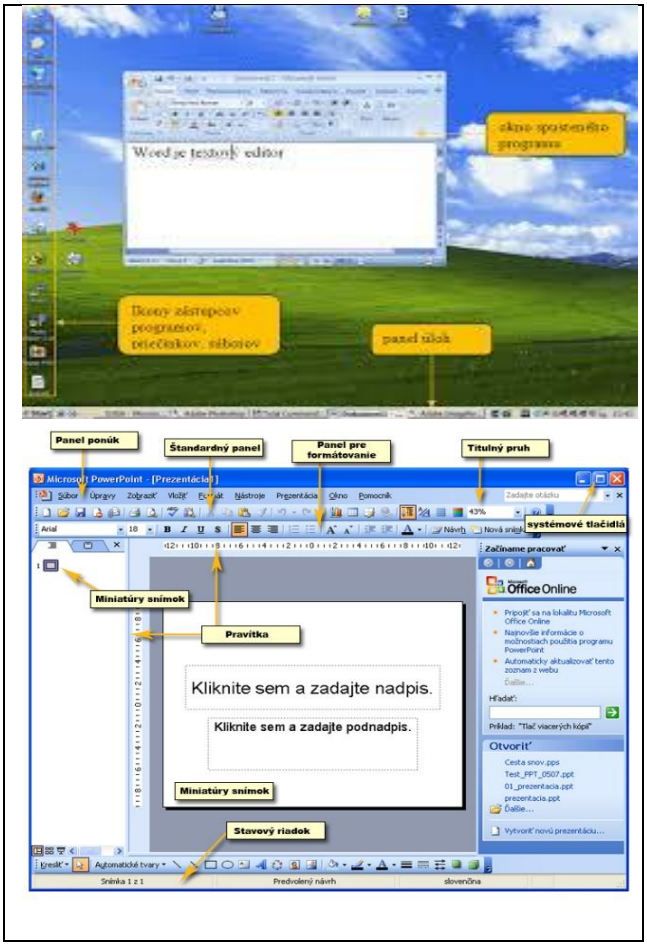
- ***Screen reader*** – program monitorového prístupu, ktorý zaznamenáva všetko čo prebieha na monitore používateľa;
- ***Syntetizátor reči*** – to je program, ktorý transformuje digitálnu informáciu načítanú screen readerom na reč.

Programov na čítanie monitora je viacero. Najviac populárne z nich sú JAWS a NonVisual Desktop (NVDA). Vo väčšine prípadov nevidiaci využívajú obidve. No druhá je populárnejšia, nakoľko je bezplatná a má otvorený vstupný kód. Syntetizátorov je tiež veľké množstvo, napr. Acapela, Vokalizer, RHVoice a iné. Mnohé čítače obrazovky a operačné sytémy už majú vstavaný svoj syntetizátor.

Na nasledujúcich obrázkoch (Obrázok 6.) je uvedených viacero spôsobov uloženia jednotlivých ovládacích prvkov na obrazovke počítača. Obrazovky predstavujú voľné rozloženie ikon a okien jednotlivých programov, ktoré sa nachádzajú v danom počítači.

Úlohou vyučujúceho je naučiť nevidiaceho základné možné rozloženie jednotlivých programov a preto je nevyhnutné vysvetliť, ako sa jednotlivé plochy nazývajú, aký je ich vzájomný vzťah a akým spôsobom je možné ich usporiadať.

Pri nácviku je možné na začiatku využívať priestorové obrazce [10]. Na papier vykreslíme zóny rozloženia jednotlivých druhov programov a na malé kúsky papiera v tvare štvorcov zobrazíme jednotlivé ikony. Názvy programov a niektoré rozloženie textov môžeme napísať Braillovým písmom. Podobné spôsoby pomoci študentom v iných oblastich sú využívané aj v zahraničí [11]. Pomocou vytvorených pomôcok si nevidiaci môžu jednotlivé ikony ukladať na fiktívnej ploche, premiestňovať ich. Ak niektorý program spustíme, alebo otvoríme adresár alebo súbor, danú ikonu zameníme zobrazením, ako je na reálnej ploche obrazovky. Pre jednoduchosť nácviku však nie je potrebné robiť identické ikony a plochy okna, ale stačí obdĺžnik či iný útvar menšej formy.



Obrázok 6 Rozloženie otvorených súborov a ich ovládacích prvkov

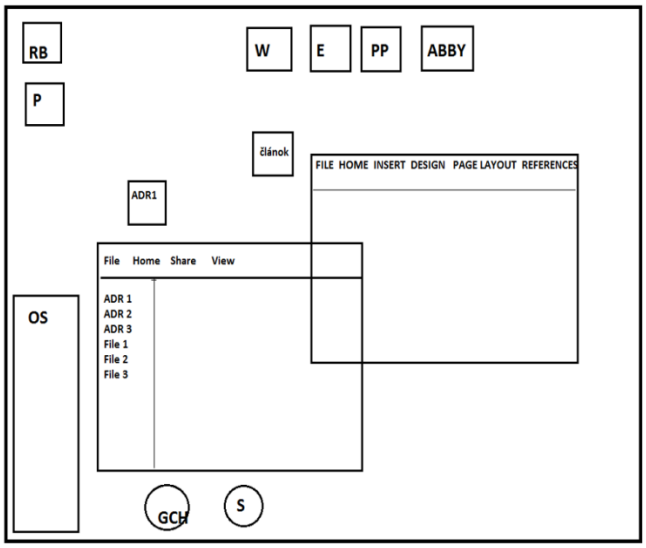
Na obrázku je uvedený jeden z možných príkladov rozloženia ikon a okien na počítači. Pri prvých krokoch s počítačom by bolo vhodné, aby si nevidiaci najskôr rozložili základné prvky na papierovej ploche, ktorej jednotlivé okná sú vyrobené z reliéfnych prvkov (Obrázok 7). Následne sa pokúsia dané rozloženie realizovať aj na počítači. Nácvik takéhoto postupu uľahčí celkový nácvik práce s počítačom a umožní nevidiacim získať prehľad medzi reálnou plochou a virtuálnou plochou na obrazovke.

Dôležitým faktorom pre lepšie uplatnenie na trhu práce je zvyšovanie celkovej digitálnej gramotnosti [12], kde sa pri online vzdelávaní uplatnia čítačky, ktoré rozširujú funkcionalitu prehliadača. V prípade online vzdelávania prostredníctvom internetového prehliadača Chrome je vhodné použiť niektorý plugin čítačky pre nevidiacich akým je napríklad ChromeVox, ktorý dobre funguje s produktmi spoločnosti Google.

5. ZÁVER

Uvedený príspevok je jednou z možností využívania prvkov vyučovania mikro orientácie na pri práci na PC. Je možné, že práve získanie základných návykov ovládania počítača umožní nevidiacim zlepšiť kvalitu svojho života. Zároveň sa použitím daného postupu pokúsime spojiť viaceré praktické zručnosti na získanie novej kvality priestorovej orientácie.

Predpokladáme, že takýto prístup urýchli postup získavania nových počítačových zručností. Samozrejme, že to nejde bez dodatočného programového vybavenia. Treba povedať, že z hľadiska techniky žijeme v zázračnej dobe, keď môžeme pomocou akéhokoľvek dnešného počítača vybaveného čítačom obrazovky byť v spojení s celým svetom, pomocou OCR programov čítať tlačené texty, pracovať v rôznych odvetviach, či len tak viac či menej užitočne tráviť čas na sociálnych sieťach.



Obrázok 7 Návrh základnej priestorovej mapy obrazovky a pohyblivých prvkov ovládania

Okrem toho, vedecký pokrok sa podpísal aj na vývoji prístrojov pre nevidiacich. V Japonsku bol vyvinutý monitor, ktorý obsahuje 3072 vydvihujúcich bodov, ktorý dovoľuje čítať zobrazenia pomocou hmatu [13]. Môžu sa dotýkať displeja. Pravda, presnosť zobrazenia je znížená v porovnaní s obyčajným monitorom viac ako sto krát. Práve pri lepšej kvalite takýchto prístrojov bude využívanie mikroorientácie ešte potrebnejšie.

6. POĎAKOVANIE

Príspevok vznikol vďaka podpore občianskeho združenia **Vzdelávanie – veda – výskum o.z.**.

7. BIBLIOGRAFICKÉ ODKAZY

[1] JESENSKÝ, J. Priestorová orientácia a samostatný pohyb zrakovo postihnutých (Zborník referátov z odborného seminára). Bratislava : ÚV ZI, 1982. 66 s.

[2] WIENER, P. 1998. Prostorová orientace zrakově postižených. Praha : MŠMT, 1998. 174 s.

[3] JESENSKÝ, J. 2007. Prolegomena systému tyflorehabilitace a metodiky tyflorehabilitačních výcviku. 1. vyd. Praha : Univerzita Jana Amose Komenského, 2007. 659 s. ISBN 978-80-86723-49-5.

[4] JESENSKÝ, J. 1988. Hmatové vnímání informací s pomocí tyflografiky. 1. vyd. Praha : Státní pedagogické nakladatelství, 1988. 226 s.

[5] Vzdelávací program pre deti a žiakov so zrakovým postihnutím. 2009. [online]. Bratislava: Štátny pedagogický ústav, 2009. [cit. 2016-02-10]. Dostupné na internete: http://www.statpedu.sk/sites/default/files/dokumenty/statny-vzdelavaci-program/vp_zp_%20isced_1_2_3_5.pdf

[6] SLOWÍK, J. 2007. Speciální pedagogika. 1.vyd. Praha : Grada, 2007. 160 s. ISBN 978-80-247-1733-3.

[7] LOPÚCHOVÁ, J. 1997. Špecifiká a osobitosti priestorovej orientácie a samostatného pohybu binokulárne chybných detí predškolského veku v hrách. In Efeta – otvor sa, roč. VII, 1997. č 1-4, s. 27-29. ISSN 1335-1397.

[8] LOPÚCHOVÁ, J. 2011. Základy pedagogiky zrakovo postihnutých. 1. vyd. Bratislava : IRIS, 2011. 249 s. ISBN 978-80-89238-61-3.

[9] LEPIL, O. 2010. Teorie a praxe tvorby výukových materiálu. 1. vyd. Olomouc : Univerzita Palackého v Olomouci, 2010. 97 s. ISBN 978-80-244-2489-7.

[10] PETTY, G.1996. Moderní vyučování. Praha : Portál, 1996. 380 s. ISBN 978-80-7367-427-4

[11] WILLINGS, C. Helping students who are blind and visually impairment learn to play [online]. [cit. 2016-02]

[12] GLATZ, M., MIŠOTA. B. 2016. Active education of the employees of the small and medium-sized enterprises. In Knowledge for Market Use 2016: Our Interconnected and Divided World, International Scientific Conference Proceedings (s. 104-114). Olomouc : Societas Scientiarum Olomucensis II, 2016. ISBN 978-80-87533-14-7

[13] Как пробраться в сеть на ощупь?» Подробнее на Medkrug.RU: http://www.medkrug.ru/article/show/kompjuter_dlja_slepyh

Nová maturita z informatiky
New Maturity Exam In Informatics

Andrej Blaho
KAI, FMFI–UK
Mlynská dolina
842 48 Bratislava
Slovenská republika
andrej.blaho@fmph.uniba.sk

Ľubomír Salanci
KZVI, FMFI–UK
Mlynská dolina
842 48 Bratislava
Slovenská republika
salanci@fmph.uniba.sk

ABSTRACT
We analyse the newest revision of standards for graduation (maturity exam) in informatics in this paper. We see from the analysis that the emphasis lies much more on algorithmic problem solving and programming skills. On the other hand, the ICT part of standards has been significantly reduced. Other areas were reformulated with regard to either technology and its short-time lasting terms or unclear legal or ethical terms. There were left only basic principles of informatics science. It turns out that currently the most valuable part of the school informatics is the area of algorithmic problem solving.

Keywords
National education program. Gymnasium (Upper Secondary General School). Maturity exam. Analysis. Evaluation.

ABSTRAKT
V tomto článku analyzujeme nové cieľové požiadavky k maturitnej skúške z informatiky. Z obsahovej analýzy vyplynulo, že nová maturita kladie veľmi vysoký dôraz na algoritmické riešenie problémov a programovanie. Naopak, bol výrazne potlačená IKT zložka. Ostatné oblasti boli očistené od technologických krátkodobo platných poznatkov, nejasných právnych a etických pojmov, vzťahov a výkonov. Ostali v nich len princípy ich fungovania. Ukazuje sa, že momentálne najhodnotnejšiu časťou školskej informatiky je oblasť algoritmického riešenia problémov.

Kľúčové slová
Štátny vzdelávací program. Gymnáziá. Maturita. Analýza. Hodnotenie.

1. ÚVOD

V predchádzajúcich príspevkoch [1], [2], [3] sme sa venovali novému štátnemu vzdelávaciemu programu z informatiky pre prvý aj druhý stupeň základnej školy a gymnáziá (ISCED 1, 2, 3A). V súčasnosti sa pripravili nové cieľové požiadavky k maturitnej skúške z informatiky.

V našom príspevku analyzujeme a hodnotíme novú maturitu z informatiky, ktorú zostavovala predmetová komisia z informatiky pri ŠPÚ na základe revidovaného štátneho vzdelávacieho programu.

2. ANALÝZA DOTERAJŠÍCH MATURÍT

V roku 2008 prebehla obsahová reforma školstva. V jej rámci vznikla aj nová verzia maturitnej skúšky. Tá však stále vychádzala do z predreformnej filozofie informatiky, v ktorej sa kládol neprimerane veľký dôraz na IKT zručnosti a technologické detaily.

Napríklad, výkonové časti požadovali množstvo vedomostí na nízkej úrovni Bloomovej taxonómie: „Charakterizovať...“, „Popísať...“, „Vysvetliť...“ atď. Znamenalo to, že sa prevažne jednalo o encyklopedické vedomosti. Vznikalo tak riziko, že stačilo, aby sa žiak naučil naspamäť správne odpovede, ale v skutočnosti poznatkom nemusel rozumieť. Pritom aktuálne trendy v definovaní výkonových štandardov v štátnom vzdelávacom programe určujú, že výkony majú byť formulované aspoň na 3. úrovni revidovanej bloomovej taxonómie.

Ďalším problémom bolo to, že mnohé témy a zručnosti z maturitnej skúšky boli na úrovni základnej školy. Napríklad: „Vyriešiť podľa zadania praktický problém...“ – úpravy rastrového obrázka, animácie, prezentácie, práca s e-mailom atď. Historicky to má korene v tom, že do roku 2008 nebola povinná informatika na základnej škole, a teda IKT zručnosti sa oficiálne učili až v rámci stredoškolskej informatiky.

V neposlednom rade požiadavky na maturitu obsahovali aj veľa technologicky orientovaných vedomostí. Napríklad: „Prenos informácie cez komunikačné médiá (medený drôt, optické vlákno, bezdrôtovo)...“, „Sieťové zariadenia, umožňujúce rozširovanie siete a rozdelenie záťaže...“, „Vysvetliť princíp paketovej komunikácie a spôsob doručovania paketov do cieľovej siete prostredníctvom medzistaníc (smerovačov).“ atď. Tu vidíme dva podproblémy.

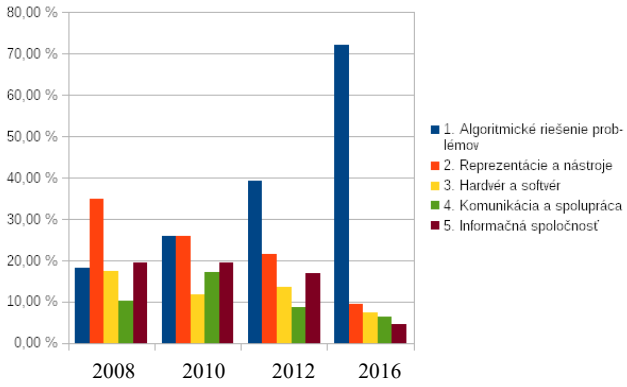
Prvý spočíva v tom, že technológie príliš rýchlo starnú a nazdávame sa, že maturitná skúška by mala skúšať vedomosti, ktoré sú trváce, nie platné iba v krátkom období. Napríklad, v maturitnom štandarde z roku 2008 sa uvádza výkon: „vysvetliť princípy jednej zo služieb rozhovoru (Talk, IRC, ICQ)“. V súčasnosti sa jedná skoro o úplne neznáme služby. Pre autorov nových maturitných štandardov z toho vyplynulo pravidlo, aby materiály formulovali tak, že budú dlhodobo platné aj pri meniacich sa technológiách.

Druhý podproblém s technologicky orientovanými vedomosťami vidíme v tom, že sú príliš úzko špecializované. Skôr by patrili do inžinierskej oblasti alebo elektrotechniky (protokoly, hardvérové detaily atď.).

Tento druhý podproblém sa netýka iba maturitnej oblasti *Komunikácie* alebo *Softvéru a hardvéru*, ale aj *Informačnej spoločnosti*. Napríklad, v maturitnom štandarde z roku 2010 sa vyžadujú práv-

ne vedomosti: „Autorské práva na softvér - registrácia softvéru, licencia, multilicencia, demoverzia, freeware, slobodný a otvorený softvér, adware, shareware, public domain, upgrade softvéru.“ Dá sa o nich nezáväzne diskutovať na hodinách informatiky (napríklad, na prvom stupni ZŠ), ale na maturitnej skúške z informatiky sa nedá objektívne zhodnotiť správnosť takejto diskusie. Môže sa stať, že žiak bude mať lepšie vedomosti v právnej oblasti ako celá komisia dohromady, a komisia nedokáže korektne posúdiť, či má žiak pravdu.

Zaujímavé je aj porovnanie textových rozsahov informatických oblastí v maturitnom štandarde. V nasledujúcej tabuľke uvádzame verzie maturitných štandardov a počty znakov, ktoré sú venované jednotlivým oblastiam. V prvom riadku tabuľky sú uvedené roky, ktoré prislúchajú rôznym verziám maturitných štandardov. V stĺpcoch sú dvojice čísel pre každý rok: prvé číslo je počet znakov v dokumente pre danú oblasť, druhé číslo hovorí o tom, aká je to percentuálna časť z celého dokumentu – teda, vidno absolútne a relatívne zastúpenie danej oblasti.



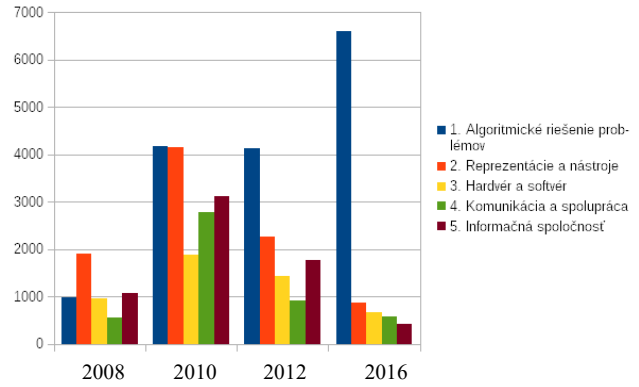
Obrázok 2. Percentuálne zastúpenie oblastí

Touto štatistikou nechceme povedať, že počet znakov vyjadruje dôležitosť oblasti. Na druhej strane však mnohí učitelia interpretujú význam informatickej oblasti práve týmto spôsobom.

Tabuľka 1. Počet znakov a percent v jednotlivých maturitných štandardoch

Verzia maturitného štandardu:	**2008 [4]**		**2010 [5]**		**2012 [6]**		**2016 [7]**	
Oblasti a inch zastúpenie:	znakov	percent	znakov	percent	znakov	percent	znakov	percent
Algoritmické riešenie problémov	988	18%	4179	26%	4140	39%	6606	72%
Reprezentácie a nástroje	1909	35%	4157	26%	2261	21%	878	10%
Hardvér a softvér	953	17%	1881	12%	1438	14%	677	7%
Komunikácia a spolupráca	553	10%	2772	17%	922	9%	583	6%
Informačná spoločnosť	1065	19%	3120	19%	1771	17%	429	5%
spolu	5468		16109		10532		9173	

Obsah tabuľky sme vizualizovali nasledujúcimi obrázkami:



Obrázok 1. Absolútne počty znakov pre jednotlivé oblasti

V obrázku 1 vidíme štyri skupiny. Každá skupina predstavuje jednu verziu maturitného štandardu. V každej skupine je 5 farebných stĺpcov pre jednotlivé oblasti. Výška stĺpca v obrázku 1 zodpovedá počtu znakov a v obrázku 2 percentuálnemu zastúpeniu.

Dá sa vidieť, že najrozsiahlejší bol materiál z roku 2010. Do roku 2012 rozsahom prevažovali nie programátorské oblasti. Malo to ten negatívny dôsledok, že algoritmické riešenie problémov a programovanie mnoho učiteľov považovalo za málo významné. Oblasť *Reprezentácie a nástroje* (v materiáli z roku 2008 to bolo v podstate IKT) postupne svojim rozsahom klesá. Naopak, algoritmické riešenie problémov významne rastie.

3. NOVÁ MATURITA

Verzia z roku 2008 bola svojim rozsahom krátka a málo špecifická. Verzia z roku 2010 bola veľmi ambiciózna a mala prehnané nároky na vedomosti maturanta. Verzia z roku 2012 mala snahu o rýchlu nápravu verzie z roku 2010. Verzia 2016 bola vytvorená s ohľadom na existujúci inovovaný štátny vzdelávací program. Nová verzia z roku 2016 vstúpi do platnosti až v roku 2018.

Cieľom novej maturitnej skúšky z informatiky je zistiť u žiakov najmä úroveň:

- schopností riešiť algoritmické problémy,
- schopností zdokumentovať a prezentovať navrhnuté riešenia,
- úroveň vedomostí a pochopenia teoretických poznatkov z informatiky.

V súčasnosti sa kladie najväčší dôraz na oblasť *Algoritmické riešenie problémov*. V minulosti bolo možné zmaturovať z informatiky na známku 3 aj bez toho, aby žiak vedel čokoľvek naprogramovať (spriemerovala sa 1 z IKT a 5 z programovania). To bol aj jeden z dôvodov, prečo sa lepšie špecifikovali témy práve z tejto algoritmickej oblasti.

Na druhej strane sa zrušila IKT zložka a nahradila sa Základmi informatiky, ktorá sa orientuje na idey a princípy fungovania z jednotlivých oblastí.

V minulom príspevku [2] sme porovnávali obsahový a výkonový štandard pre základnú školu, gymnázia a s vtedy platnou maturitou. V tabuľke 2. uvádzame porovnanie aktuálnych vzdelávacích programov (ISCED1, 2, 3A) a nových maturít.

Tabuľka 2. Značkami je znázornená hustota / náročnosť konceptov v jednotlivých ročníkoch.

Názov oblasti \ ročník, stupeň vzdelania	2. – 4.	5. – 6.	7. – 8.	gym.	mat.
Reprezentácie a nástroje					
– práca s grafikou	•	●	○	○	
– práca s textom	•	●	○	○	
– práca s príbehmi / prezentáciami	•	●	○	○	
– práca s tabuľkami		•	●	●	
– práca s multimédiami	•		•	●	
– informácie	•	•	•	•	
– štruktúry	•	•	•	•	
– digitálna reprezentácia informácií					○
Komunikácia a spolupráca					
– práca s webovou stránkou	•	●	○		
– prezent. inf. prostredníctvom web stránky				●	
– vyhľadávanie na webe	•	•	●	●	
– práca s nástrojmi na spoluprácu a zdieľanie inf.				●	
– práca s nástrojmi na komunikáciu	•	•	●	○	
– princípy komunikácie prostredníctvom nástrojov					○
Algoritmické riešenie problémov					
– analýza problému	•	•	•	•	●
– jazyk na zápis riešenia / interaktívne zostav. riešenia	•	•	•	●	●
– pomocou postupnosti príkazov	•	•	●	●	●
– pomocou cyklov		•	•	•	●
– pomocou vetvenia			•	•	●
– pomocou nástrojov na interakciu			•	•	●
– pomocou premenných + typy údajov, výrazy			•	●	○
– pomocou jednorozmerných polí					●
– pomocou podprogramov s parametrami					●
– s ciframi, znakmi, reťazcami, textovými súbormi					●
– interpretácia zápisu riešenia	•	•	•	•	●
– hľadanie a opravovanie chýb	•	•	•	•	●
– posúdenie a hodnotenie riešenia					●
Softvér a hardvér					
– práca so súbormi a priečinkami	•	•	●	○	
– práca v operačnom systéme	•	•	●	●	
– počítač a prídavné zariadenia	•	•	●	●	
– práca v počítačovej sieti a na internete	•	•	●	●	
– práca proti vírusom a špehovaniu		•	●	●	
– princípy fungovania softvéru					○
Informačná spoločnosť					
– bezpečnosť a riziká	•	•	•	•	
– digitálne technológie v spoločnosti	•	•	•	•	
– legálnosť používania	•	•	•	•	
– rôzne aspekty informačnej spoločnosti					•

Vysvetlenie značiek v tabuľke 1. Žiaci:

- • ... začínajú pracovať, zbierať zručnosti a opatrne sa zoznamujú s informatickými konceptmi
- ● ... spoznávajú základy práce, kladie sa zvýšený dôraz na nové informatické koncepty, poznatky
- ○ ... rozumejú na úrovni princípov (t.j. abstraktná úroveň)

V porovnaní s predchádzajúcimi maturitnými štandardmi boli v oblasti *Algoritmické riešenie problémov* lepšie vymedzené hranice poznatkov, ktoré majú maturanti zvládať. Napríklad, najväčšia časť tejto oblasti má názov *1.3 Riešenie problémov*. Jej obsahový štandard tvorí iba minimum pojmov:

- Pojmy:
  - algoritmus,
  - interpretácia a kompilácia.
- Vlastnosti a vzťahy:
  - chyby: syntaktické a logické,
  - analýza problému – vzťahy medzi vstupom a výstupom,
  - vstupné obmedzenia– pre ktoré hodnoty algoritmus správne pracuje.

Naopak, výkonový štandard tento časti je značne rozsiahly. Je rozdelený do menších tém. Teda, *„žiak vie“*:

- analyzovať problém,
- riešiť algoritmické problémy, vytvárať a čítať zápisy v programovacom jazyku,
- riešiť úlohy, v ktorých sa pracuje s číslami a ciframi čísla,
- riešiť úlohy, v ktorých sa využíva interakcia,
- riešiť problémy, v ktorých sa využívajú znaky a textové reťazce,
- riešiť problémy, v ktorých sa využívajú premenné,
- riešiť problémy, v ktorých sa využíva cyklus a vetvenie,
- riešiť problémy, v ktorých sa využíva generátor náhodných čísel,
- riešiť problémy, v ktorých sa využívajú podprogramy,
- riešiť problémy, v ktorých sa využíva jednorozmerné pole,
- riešiť problémy, v ktorých sa využíva textový súbor,
- opravovať syntaktické chyby,
- zisťovať, hľadať a opravovať logické chyby v programe
- intuitívne uvažovať o zložitosti algoritmu (na úlohách primeranej zložitosti).

Keď sa podrobnejšie pozrieme napríklad na tému *„žiak vie riešiť problémy, v ktorých sa využíva jednorozmerné pole“*, uvidíme tieto podrobnejšie špecifikované výkony:

- generovať obsah poľa podľa daných kritérií (napríklad: vynulovanie poľa, každý nasledujúci prvok je súčtom predchádzajúcich dvoch a pod.),
- používať pole na uchovanie väčšieho počtu údajov (napríklad: frekvenčná tabuľka výskytu nejakých hodnôt, súradnice bodov v rovine a pod.),
- hľadať prvky poľa s danými vlastnosťami (napríklad: najmenší, druhý najmenší a pod.),
- zisťovať, či pole obsahuje hodnoty s danými vlastnosťami (napríklad: či obsahuje číslo 0, prvý výskyt medzery a pod.), resp. ich počet,
- zisťovať, či pole spĺňa dané kritérium (napríklad: či sú všetky prvky rovnaké, rôzne, či sú hodnoty usporiadané a pod.),
- modifikovať prvky poľa (napríklad: vsunúť prvok na dané miesto tak, aby ostali prvky usporiadané a pod. – pozor, už nie triediace algoritmy),
- manipulovať s viacerými poľami podľa daných kritérií (napríklad: kopírovanie časti poľa, otočenie, posunutie, zlučovanie dvoch usporiadaných postupností a pod.),
- testovať hranice poľa,
- zobrazovať prvky poľa vypísaním alebo vykreslením (napríklad: stĺpcový graf).

Predpokladá sa, že predmetová komisia školy vypracuje maturitné otázky tak, aby rešpektovali stanovené hranice maturitného štandardu.

4. ZÁVER

Historicky (80. roky) bola informatika a maturita z informatiky len o programovaní a úplne bez IKT. IKT v tých časoch prakticky neexistovali. Neskôr nastúpil trend, keď sa za informatiku považovali len IKT. Do istej miery to malo zmysel, lebo v 90. rokoch boli IKT novinkou aj pre dospelých. Koncom 90. rokov sa vrátilo programovanie do informatiky, kde už bolo spolu s IKT.

Dnes je však situácia iná. Základy IKT sú náplňou informatiky už na 1. stupni. Dokonca, títo žiaci zvládajú náročnejšie zručnosti ako bolo bežné v 90. rokoch na stredných školách. Nazdávame sa preto, že v súčasnosti stráca zmysel skúšať na maturite zručnosti zo základnej školy. Maturita by mala testovať trvácne a hodnotné vedomosti, nie také, ktoré o pár rokov prestanú platiť.

BIBLIOGRAFICKÉ ODKAZY

[1] BLAHO, A.: *Informatika v štátnom vzdelávacom programe.* Zborník konferencie DidInfo2012. Banská Bystrica : 2012, s. 7 – 14. ISBN 978-80-557-0342-8

[2] SALANCI, Ľ.: *Informatika v novom štátnom vzdelávacom programe.* Zborník konferencie DidInfo2014. Banská Bystrica : Univerzita Mateja Bela, 2014. - ISBN 978-80-557-0698-6. - s. 143-148

[3] SALANCI, Ľ.*: Analýza informatiky v novom štátnom vzdelávacom programe.* Zborník konferencie DidInfo2015. Banská Bystrica : Univerzita Mateja Bela, 2015. - ISBN 978-80-557-0852-2. - s. 134-138

[4] Štátny pedagogický ústav: *Cieľové požiadavky na maturitné skúšky platné do šk. r. 2010/2011.* http://www.statpedu.sk/sites/default/files/dokumenty/cielove-poziadavky-na-maturitne-skusky/informatika.pdf

[5] Štátny pedagogický ústav: *Cieľové požiadavky na maturitné skúšky platné od šk. r 2011/2012 do školského roka 2012/2013 vrátane.* http://www.statpedu.sk/sites/default/files/dokumenty/cielove-poziadavky-na-maturitne-skusky/informatika_cp.pdf

[6] Štátny pedagogický ústav: *Cieľové požiadavky na maturitné skúšky platné od šk. r 2013/2014.* http://www.statpedu.sk/sites/default/files/dokumenty/cielove-poziadavky-na-maturitne-skusky/cp_informatika_2013_2014.pdf

[7] Štátny pedagogický ústav: *Cieľové požiadavky na maturitné skúšky platné od šk. r 2018/2019.* http://www.statpedu.sk/sites/default/files/nove_dokumenty/Cielove%20po%C5%BEiadavky%20pre%20mat.%20skusky/Informatika.pdf

Abstraktné dátové typy v úlohách súťaže iBobor kategórie Benjamíni

Abstract data types in tasks of Bebras competition in Benjamins category

Lucia Budinská
KZVI, FMFI Univerzita Komenského
Mlynská dolina 1
842 48 Bratislava
Slovensko
lucia.budinska@fmph.uniba.sk

Monika Tomcsányiová
KZVI, FMFI Univerzita Komenského
Mlynská dolina 1
842 48 Bratislava
Slovensko
monika.tomcsayniova@fmph.uniba.sk

ABSTRACT

In this paper, we talk about tasks that are focused on use of abstract data types from informatics competition Bebras. We analyze solutions of two tasks in Benjamins group from 2016/17 contest. We compare result of contestants in different grades (5^{th} to 7^{th}) and also take a look at gender differences. We think about how pupils solved tasks, how they get wrong answers and what was the main reason why and what results from that for creating this type of tasks. Moreover, we describe how tasks with abstract data types can be used in informatics at schools.

Keywords

Bebras. Benjamins. Solution analysis. Abstract data type.

ABSTRAKT

V príspevku sa budeme venovať úlohám súťaže iBobor, ktoré sa zameriavali na použitie abstraktných dátových typov. Zanalyzujeme riešenia úloh z kategórie Benjamíni v školskom roku 2016/17. Porovnáme úspešnosti jednotlivých ročníkov spadajúcich do tejto kategórie (5. až 7. ročník ZŠ), a tiež rozdiely medzi chlapcami a dievčatami. Zamyslíme sa nad tým, ako žiaci riešili dané úlohy, prečo a akým spôsobom dospeli k nesprávnym riešeniam a čo z toho vyplýva pre tvorbu úloh z tejto oblasti. Tiež popíšeme, kedy a ako sa takéto úlohy dajú použiť vo vyučovaní na základných školách.

Kľúčové slová

IBobor. Kategória Benjamíni. Analýza riešení úloh. Abstraktné dátové typy.

1. ÚVOD

Bebras, alebo po slovensky iBobor, je informatická súťaž, ktorá vznikla v roku 2004 v Litve [1]. Na Slovensku sa prvýkrát súťažilo v roku 2007 a v školskom roku 2016/17 sa konal už 10. ročník, ktorého sa zúčastnilo 62 981 súťažiacich z 921 škôl, ktorí súťažili v 5 kategóriách podľa ročníkov [2]. Najväčšou je každoročne kategória Benjamíni, do ktorej patria žiaci 5. až 7. ročníka základných škôl a prímy a sekundy osemročných gymnázií. Tento rok kvôli technických problémom súťažili dva dni (10. a 15. novembra) a dokopy súťažilo 20 615 súťažiacich. 23,2% z nich boli úspešnými riešiteľmi (t. j. dosiahli aspoň polovicu kladných bodov).

Súťaž iBobor sa snaží prinášať úlohy vhodné pre všetkých žiakov a z rôznych oblastí informatiky, pozri časť 2. My sme sa sústredili na žiacke riešenia dvoch úloh, analýze ktorých sa budeme venovať v kapitolách 3 a 4. Okrem vyhodnotenia úspešnosti žiakov v týchto úlohách, navrhneme zaradenie takýchto typov úloh aj do kontextu školskej informatiky.

2. TYPY ÚLOH V SÚŤAŽI IBOBOR

Úlohy súťaže iBobor sú v rámci databázy a ich následného využívania v prostredí Bobrovo, pozri [3], zaraďované do jednej alebo viacerých oblastí informatiky [4], v rámci medzinárodnej databázy ich zaraďujú do 5 oblastí, pozri Tabuľku 1.

Tabuľka 1 - Oblasti úloh v súťaži iBobor

INF	Spracovanie informácií (reprezentácia informácií, kódovanie, šifrovanie)
ALG	Algoritmické myslenie (algoritmy, programovanie)
USE	Používanie počítačových systémov (počítačové programy a pod.)
STRUC	Štruktúry, vzory, usporiadania (kombinatorika, diskrétne štruktúry, dátové typy)
PUZ	Hádanky (logické hádanky, hry)
SOC	IKT a spoločnosť (sociálne, etické, kultúrne a právne aspekty informatiky)

Keďže v tomto ročníku patrili medzi najmenej úspešné úlohy z oblasti algoritmického myslenia a štruktúr, zaujímalo nás, v čom konkrétne dané problémy vznikajú. Sústredili sme sa hlavne na kategóriu STRUC a konkrétne na abstraktné dátové typy, ktoré podľa Inovovaného štátneho vzdelávacieho programu [5] môžeme zaradiť až do dvoch oblastí. Základné štruktúry, ich reprezentácia, vyvodzovanie vzťahov a uvažovanie o nich sa nachádza v tematickom okruhu Reprezentácie a nástroje. Keďže ale pri práci so štruktúrami často žiaci riešia nejaké problémy, hľadajú vhodné algoritmy, alebo naučené algoritmy aplikujú, môžeme úlohy z tejto oblasti zaradiť aj do okruhu Algoritmické riešenie problémov.

Pri úlohách z tejto oblasti nemusí byť žiakom na prvý pohľad zrejmé, že sa jedná o informatickú úlohu. Ideou súťaže je ale zoznámiť žiakov so základnými informatickými konceptami a rozvíjať ich algoritmické a výpočtové (informatické) myslenie [6]. Zároveň by žiaci mali vedieť vyriešiť úlohu bez toho, aby

poznali dané koncepty, ale je potrebné, aby rozmýšľali o informáciách, štruktúrach, spracovaní a vizualizácií a pod. [7].

Keďže viaceré úlohy používajú vo svojich národných súťažiach viaceré krajiny, schopnosť riešiť úlohy nemá závisieť na školskom systéme krajiny, v ktorej sa súťaž koná. Ukazuje sa ale, že všeobecne majú žiaci vo všetkých krajinách problémy s abstraktným myslením [8], čo podporuje aj teória J. Piageta [9], podľa ktorej sú žiaci v piatom a niektorí aj v šiestom ročníku na pomedzí štádia konkrétnych a štádia formálnych operácií a abstraktné myslenie sa u nich ešte len rozvíja.

Práve úlohy z kategórie STRUC sa častejšie zameriavajú na to, že žiaci potrebujú abstraktne myslieť, reprezentovať abstraktné koncepty a snažiť sa pochopiť rôzne štruktúry. Riešením takýchto úloh sa môžu žiaci v spomenutých oblastiach zlepšiť. A to bez ohľadu na ich pohlavie. Súťaž sa snaží byť rovnako zaujímavá aj pre chlapcov, aj pre dievčatá. Výskumy z iných krajín [7, 8] naznačujú, že v tomto veku (kategória Benjamíni) neexistujú medzi pohlaviami veľké rozdiely v úspešnosti. Samozrejme, existujú nejaké odlišnosti, napríklad pri úlohách s priestorovým myslením sú úspešnejší chlapci a v článku [8] autori zistili, že chlapci častejšie volia odpoveď tipovaním, kým dievčatá sa snažia nájsť takú odpoveď, ktorá (aspoň čiastočne) vyhovuje zadaniu.

V súťaži existujú rôzne typy úloh – niektoré sú interaktívne, pri nich žiaci sami hľadajú, vytvárajú alebo upravujú odpoveď, iné úlohy majú formu výberu jednej odpovede z niekoľkých (najčastejšie zo štyroch) ponúkaných. Pri navrhovaní odpovedí k takejto úlohe je potrebné, aby autori rozmýšľali nad najčastejšími chybami, ktoré môžu žiaci spraviť a podľa toho pripravili odpovede. V [10] zadefinovali autori typické chyby, s ktorými sa stretli v súťaži Bebras. Prvým je *omyl* (keď žiak predpokladá, že to čo robí, je správne), ďalšími *chyba v stratégii* a *chyba zlého odvodenia*. Z analýzy výsledkov (a to pre oba typy úloh) je potom možné zistiť, akú chybu žiaci robili najčastejšie. To môže pomôcť nielen autorom pri tvorbe úloh do ďalších ročníkov súťaže, ale aj učiteľom informatiky, keďže vieme zadefinovať problémové oblasti[1], ktorým je potrebné venovať väčšiu pozornosť pri vyučovaní.

3. ANALÝZA ŽIACKYCH RIEŠENÍ

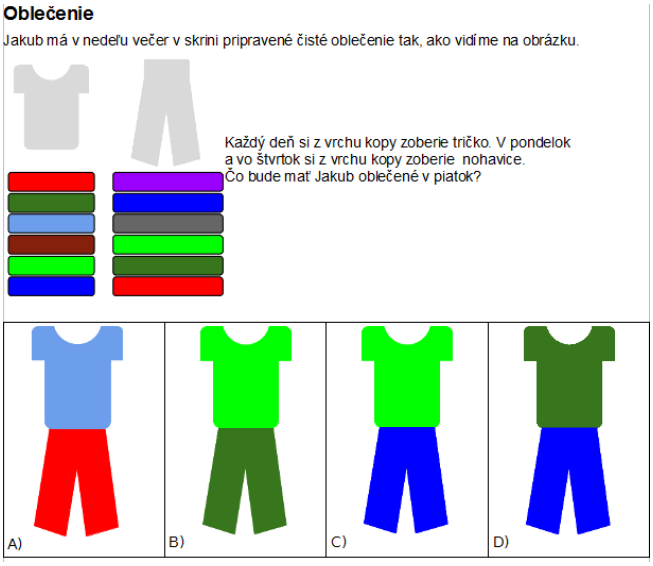
V kategórii Benjamíni súťažilo 20 615 žiakov, z toho 9 586 žien a 11 029 mužov. 32,9% súťažiacich boli piataci, 30,28% navštevovalo šiesty a 29,1% siedmy ročník základných škôl. Do tejto súťažnej kategórie patria aj prvé dva ročníky osemročných gymnázií, primánov tento rok súťažilo 3,78% a sekundánov 3,93%. Keďže žiakov z osemročných gymnázií je rádovo menej, pri analýze úloh podľa veku, sme spojili šiestakov s primánmi a siedmakov so sekundánmi. Vznikli tak porovnateľne veľké skupiny a keďže v piatom ročníku sa nachádzajú spolu žiaci, ktorí neskôr pôjdu na osemročné gymnáziá a aj tí, ktorí zostanú na základných školách, dáva zmysel tieto skupiny spojiť aj vo vyšších ročníkoch.

Jednou z našich hypotéz bolo, že žiaci v piatom ročníku budú pri riešení týchto úloh výrazne menej úspešní ako žiaci v siedmom ročníku. Ďalej sme predpokladali, že celková úspešnosť úlohy Oblečenie bude nad 55% a úlohy Reštaurácie nad 30%. A taktiež, že dievčatá aj chlapci budú približne rovnako úspešní pri riešení týchto úloh.

[1] Ktoré sa často definujú z rôznych štandardizovaných testov napr. na matematike, či vyučovacích jazykoch.

3.1 Úloha Oblečenie

Úloha Oblečenie, pozri Obrázok 1, vznikla úpravou pôvodnej maďarskej úlohy. V slovenskej verzii ju autori značne zjednodušili, preto bola zaradená medzi stredne ťažké úlohy namiesto odporúčaných ťažkých. Informatické pozadie úlohy tvorí koncept zásobníka, LIFO, ktorý je prispôsobený jazyku a veku súťažiacich. Chlapec mal dva „zásobníky“ – kôpky s oblečením, pričom pre každý z nich platí iné pravidlo, podľa ktorého sa z neho vyberá. Tričká vyberal z vrchu kopy každý deň, nohavice v pondelok a štvrtok. Ako ukázali výsledky, až tretina súťažiacich vybrala odpoveď, v ktorej bola kombinácia nohavíc a trička vytvorená pomocou pravidla pre tričko (každý deň nové).



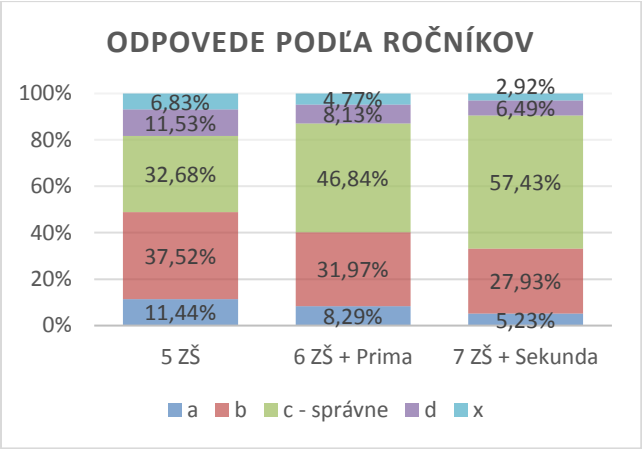
Obrázok 1 - Úloha Oblečenie (správna odpoveď C)

Celková úspešnosť úlohy dosiahla 45,68%, 4,84% žiakov ju neriešilo a zvyšné odpovede boli nesprávne. Zaujímalo nás, ako sa výber odpovedí líšil medzi jednotlivými ročníkmi. Kým piataci najčastejšie zvolili nesprávnu odpoveď B (37,52%) a správnu odpoveď dalo len 32,68% z nich, vo zvyšných ročníkoch prevažovala správna odpoveď. Kompletné výsledky sú v Grafe 1. Rozdiel medzi odpoveďou C (správnou) a odpoveďou B je v prípade piatakov -4,84 percentuálneho bodu, pre šiestakov spolu s primánmi 14,87 percentuálneho bodu a pre siedmakov spolu so sekundánmi až 29,50 percentuálneho bodu. Vidíme, že úloha bola pre siedmakov skôr ľahšia, pre piatakov bola však príliš ťažká.

Ak sa pozrieme na samotné osemročné gymnáziá, úspešnosť úlohy u primánov dosiahla 63,67% a v sekunde 75,83%. Takže v prípade týchto ročníkov by sme úlohu vedeli zaradiť medzi ľahké.

Pri návrhu odpovedí, z ktorých si majú žiaci vyberať, tzv. distraktorov, sa autori úloh snažia pripraviť také, o ktorých predpokladajú, že by mohli patriť medzi najpravdepodobnejšie chybné odpovede žiakov pre danú úlohu. V tomto prípade sa ukázalo, že odpoveď s využitím jednoduchšieho pravidla pre oba zásobníky, čo je vlastne chyba zlého odvodenia (od jedného pravidla si žiaci odvodili aj druhé), bola najčastejšou chybnou odpoveďou. Zvyšné dve odpovede, A a D mali v každom ročníku približne rovnaké zastúpenie. Kým odpoveď D sa dala dosiahnuť tak, že žiaci by použili pravidlo pre nohavice na oba zásobníky

(rovnako chyba odvodenia), alebo by použili pravidlo pre tričko a brali zospodu zásobníkov, odpoveď A, sa nedala dosiahnuť iba jednou chybou v úvahe. Práve kvôli rovnakému percentuálnemu zastúpeniu však predpokladáme, že žiaci tieto odpovede vyberali skôr tipovaním, ako riešením úlohy, čo potvrdzuje aj analýza riešení podľa pohlavia.



Graf 1 - Odpovede k úlohe Oblečenie podľa ročníkov (x znamená, že súťažiaci neodpovedal)

Rozdiely medzi chlapcami a dievčatami neboli v úspešnosti, ale pri porovnaní nesprávne zvolených odpovedí vidíme, že dievčatá častejšie volili odpoveď B a naopak, chlapci jednu z odpovedí A alebo D. Čo súhlasí s výsledkami prezentovanými v predchádzajúcej časti – chlapci častejšie vyberajú odpoveď tipovaním ako dievčatá.

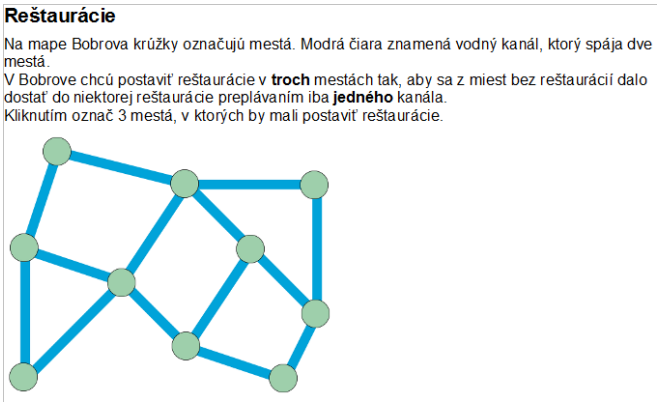
Z výsledkov predpokladáme, že žiaci nečítajú zadanie s porozumením, namiesto vytvorenia vlastnej odpovede vyberajú z možností a často vyberú prvú, ktorá im vyhovuje[2] a spätne ju neskontrolujú so zadaním. Ak úlohe nerozumejú, radšej tipujú akoby ju neriešili. Náročnosť úlohy pre piatakov môže súvisieť s tým, že ešte nemajú dostatočne rozvinuté algoritmické myslenie, respektíve sa programovaniu nevenovali a preto im môže vyhodnocovanie pravidiel a krokovanie algoritmu robiť problémy.

3.2 Úloha Reštaurácie

Druhou úlohou z oblasti STRUC v tejto kategórii bola úloha Reštaurácie, pozri Obrázok 2, ktorá zoznamovala žiakov s grafom a konkrétne sa venovala vrcholovému pokrytiu. Úlohu vymyslel švajčiarsky tím a v slovenskej mutácii sa zmenila iba motivácia – kým v pôvodnej úlohe mali žiaci na mapu umiestňovať nemocnice, v slovenskej súťaži to boli reštaurácie. Podľa náročnosti zaradili organizátori úlohu medzi ťažké, keďže predstavuje pre žiakov pomerne neznámy a abstraktný koncept.

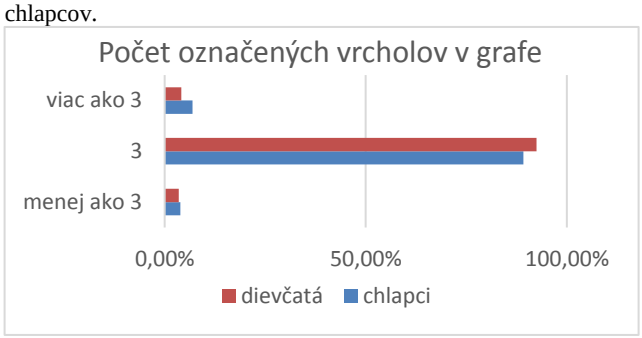
Úloha bola interaktívna, ponúkala žiakom 10-vrcholový graf spojený hranami s tým, že vrcholy sa dali označiť kliknutím. Hoci v zadaní bolo napísané, že majú umiestniť tri reštaurácie, žiaci mohli označiť aj viac resp. menej vrcholov. Správnych riešení bolo 8, dokopy ale existovalo 2^{10} možných riešení. V tejto úlohe nás preto nezaujímalo len to, koľko žiakov malo správne riešenie, ale aj ktoré konkrétne riešenie našli a prečo boli ich riešenia nesprávne.

Celková úspešnosť úlohy bola 27,98% a neriešilo ju 10,08% súťažiacich (mohlo sa stať, že niektorí súťažiaci úlohu neriešili, lebo mali problém s jej zobrazením, prípadne sa k nej už z časových dôvodov nedostali). Z pohľadu ročníkov dopadli rovnako ako pri predchádzajúcej úlohe výrazne horšie piataci – správne túto úlohu vyriešilo iba 16,6% z nich a 12,64% ju neriešilo. Šiestaci spolu s primánmi mali úspešnosť 27,24% a siedmaci so sekundánmi dokopy 40,07%. Zaujímavé je, že úspešnosť sekundánov (ktorých bolo 811) bola až 63,63%.



Obrázok 2 - Úloha Reštaurácie

Zaujímalo nás, koľko žiakov spravilo chybu práve v počte označených vrcholov. Až 5,62% žiakov, ktorí úlohu riešili, označilo viac ako 3 vrcholy a 3,72% žiakov označilo 1 alebo 2 vrcholy, pozri Graf 2. Samozrejme, výskyt tejto chyby bol častejší u piatakov a rovnako bol častejší výskyt označenia viac ako troch vrcholov, čo môže súvisieť s nedostatočným prečítaním zadania. Pri skúmaní vzťahu pohlavia a tejto chyby, sme zistili, že dievčatá častejšie označili tri vrcholy (92,48% z dievčat, ktoré úlohu riešili) ako chlapci (89,16%). Označenie jedného alebo dvoch vrcholov bolo približne rovnaké, ale chlapci oveľa častejšie označili viac ako tri vrcholy (6,92% chlapcov a 4,10% dievčat). Pre konkrétne počty vrcholov je pomer dievčat a chlapcov pomerne vyrovnaný, až na dve odchýlky – kým všetkých 10 vrcholov označilo iba 0,86% dievčat, z chlapcov to bolo až 2,18%, a 4 vrcholy označilo 2,77% chlapcov a 1,96% dievčat. To môže byť spôsobené, podobne ako pri prechádzajúcej úlohe, nejakou formou tipovania alebo „rebélie“ chlapcov.



Graf 2 - Označenia vrcholov v úlohe Reštaurácie podľa pohlavia

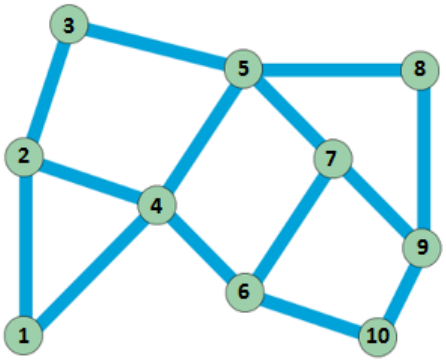
Keď sa pozrieme iba na žiakov, ktorí úlohu riešili, vidíme veľké rozdiely medzi správnymi odpoveďami. Až 68,39% odpovedí piatakov síce obsahovalo iba tri označené vrcholy, ale graf nebol

[2] žiaci vidia odpovede v náhodne vygenerovanom poradí

úplne pokrytý. Pre šiestakov s primánmi je to číslo 61,30% a pre siedmakov so sekundánmi už 49,49%.

Keďže väčšina súťažiacich označila tri vrcholy, ale odpoveď nemali správnu, skúmali sme, prečo bola ich odpoveď zlá – a teda, koľko vrcholov v grafe nepokryli. V grafe v úlohe bola maximálna miera nepokrytia pri výbere troch vrcholov 5, čo sa ale vyskytovalo pomerne málo. Celkovo, najviac žiakov nepokrylo svojím riešením dva vrcholy grafu - 38,56%, 21,38% žiakov nepokrylo jeden vrcholov grafu, 19,22% štyri, 16,47% tri a iba 4,36% riešení bolo takých, v ktorých žiaci nepokryli päť vrcholov. Častejšie všetky vrcholy nepokryli dievčatá (67,17% z dievčenských riešení, v ktorých boli označené tri vrcholy a 64,39% chlapčenských riešení).

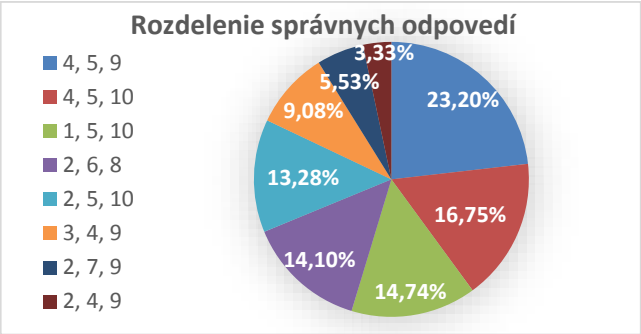
Aké boli stratégie pre správne riešenie? Ak očíslujeme vrcholy ako na Obrázku 3, medzi najčastejšie sa vyskytujúce nesprávne odpovede patrili riešenia (1, 2, 4), (5, 7, 9), (4, 5, 7), (3, 5, 8), (1, 2, 3). Z obrázku je jasné, že žiaci najčastejšie vyberali vrcholy tvoriace reťaz, čo môže byť znamením, že nepochopili zadanie úlohy. Alebo to môže byť spôsobené chybou v stratégii [10], ktorú si zle zvolili.



Obrázok 3 - Graf z úlohy Reštaurácie s očíslovanými vrcholmi

Správne riešenia sú postupne, v poradí od najčastejšie nájdených, tieto trojice: (4, 5, 9), (4, 5, 10), (1, 5, 10), (2, 6, 8), (2, 5, 10), (3, 4, 9), (2, 7, 9), (2, 4, 9). Rozdelenie podľa počtu žiakov sa nachádza v Grafe 3. Celkovo až 67,96% súťažiacich so správnou odpoveďou našlo odpoveď obsahujúcu vrchol 5. Je dosť možné, že vyberali vrcholy, z ktorých vychádza najväčším počet hrán (teda má aj najväčšie pokrytie) a postupne sa snažili vybrať také vrcholy, ktoré by pokryli zvyšok grafu. Intuitívne teda použili greedy algoritmus. Ako v [11] spomínajú, greedy koncept žiaci inštinktívne používajú na riešenie úloh, poznajú ho už z matematiky, kde sa učia úlohu rozdeliť na podproblémy a tie efektívne riešiť.

Druhou stratégiou mohol byť tiež greedy algoritmus, v ktorom sa ale prvý vrchol vyberal náhodne (každý vrchol sa nachádza v aspoň jednom správnom vrcholovom pokrytí) a ďalšie z ešte nepokrytých. Pri takomto riešení sa však mohli vyskytnúť situácie, keď museli použiť, opäť intuitívne, backtraking, ktorý je spomenutý aj vo vzorovom autorskom riešení. Pokiaľ nenašli správne trojvrcholové riešenie, mohli skúšať spätne odoberať vrcholy, interaktivita úlohy im to dovoľovala.



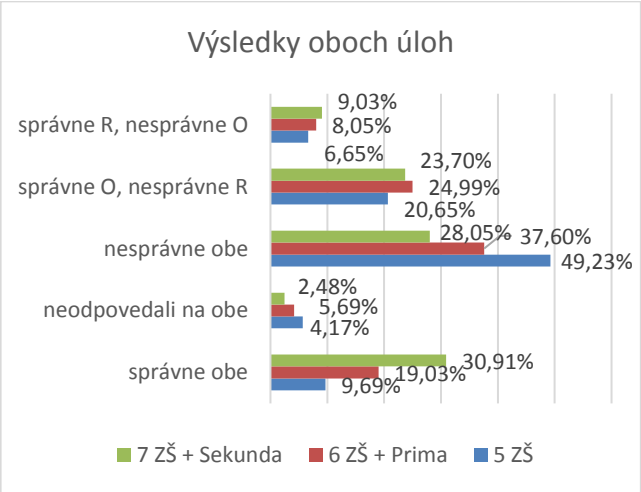
Graf 3 - Správne odpovede podľa počtu výskytu (čísla v legende označujú čísla vrcholov v grafe podľa Obrázku 3)

Úloha vyžadovala určitú mieru abstraktného myslenia pri reprezentácii mapy miest a pochopenie a správnu interpretáciu podmienky *„aby sa z miest bez reštaurácii dalo dostať do niektorej reštaurácie preplávaním iba jedného kanála“*. Je možné, že práve tieto aspekty viedli k nízkej úspešnosti úlohy v piatom a šiestom ročníku.

3.3 Vzájomný vzťah úloh

Obe úlohy sa zaoberali abstraktnými dátovými typmi, boli pomerne náročné na algoritmické a abstraktné myslenie a vyšli ako náročnejšie pre piatakov, preto nás zaujímalo, ako ich riešili žiaci dokopy. Iba 19,88% všetkých súťažiacich vyriešilo správne obe úlohy, nesprávne obe úlohy malo 38,27% súťažiacich, 25,80% žiakov malo správne úlohu Oblečenie a nesprávne úlohu Reštaurácie a naopak 8,10% malo správne úlohu Reštaurácie a nesprávne úlohu Oblečenie.

Rozdiely medzi pohlaviami nie sú výrazné, z pohľadu ročníkov to ale neplatí. Nesprávne obe úlohy malo až 49,23% piatakov a iba 9,69% z nich malo obe úlohy správne. Obe úlohy správne vyriešilo 30,91% siedmakov a sekundánov, pričom ale podobne veľká časť z nich, 28,05%, mala obe úlohy nesprávne. Ak mali z danej dvojice úloh žiaci správne iba jednu, častejšie to bola úloha Oblečenie (25,8% súťažiacich a 8,1% malo správne úlohu Reštaurácie). Kompletné výsledky zobrazuje Graf 4.

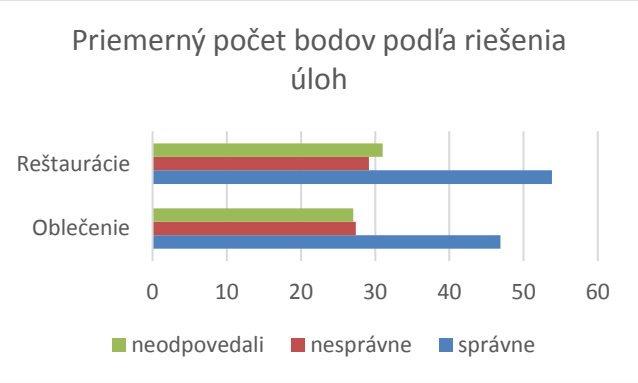


Graf 4 - Výsledky úloh Reštuarácie (R) a Oblečenie (O)

Priemerný počet bodov pre celú kategóriu Benjamíni bol 36,26.

Ak sa pozrieme na priemerný počet bodov, ktorý dosiahli súťažiaci celkovo na základe ich odpovedí v nami analyzovaných úlohách, zistíme, že úlohy vyriešili správne úspešnejší súťažiaci. Pri úlohe Reštaurácie bol dokonca priemerný počet bodov žiakov, ktorí boli úspešní v tejto úlohe, nad hranicou úspešného riešiteľa (nad 50 bodov) a priemer žiakov, ktorí správne vyriešili obe úlohy, bol až 58,32 bodu.

Rozdiely medzi žiakmi, ktorí úlohu vyriešili nesprávne a tými, ktorí ju neriešili, nie sú veľké. Ich priemerný bodový výsledok bol však približne o 20 bodov nižší, ako žiakov, ktorí boli v úlohách úspešní. Graf 5 zobrazuje úplne výsledky.



Graf 5 – Priemer celkového výsledku žiakov v súťaži podľa skúmaných úloh (maximum bolo 80b)

4. VYUŽITIE ÚLOH V ŠKOLE

Úlohy, ktoré sme rozoberali, sa dajú použiť aj v bežnom vyučovacom procese. Ako sme už v úvodnej kapitole spomínali, vieme ich zaradiť do oblastí Algoritmické riešenie problémov a aj Reprezentácie a nástroje. Pozrime sa na to konkrétne.

Podľa iŠVP [5] majú žiaci na konci 6. ročníka vedieť orientovať sa v jednoduchej štruktúre, organizovať údaje do štruktúr a interpretovať údaje zo štruktúr. Jednou z štruktúr, ktorú im môžeme vysvetliť (okrem napr. tabuliek, polí...), je aj graf. Okrem úlohy, akú sme spracúvali, ponúka archív úloh iBobra viacero grafových úloh – či ide o hľadanie najkratšej cesty, minimálnej kostry, vrcholového alebo hranového pokrytia a pod. Samozrejme, názvy problémov, či algoritmy na riešenie žiaci nepotrebujú poznať, stačí zvoliť vhodnú motiváciu a napríklad používať greedy algoritmy. Pri greedy algoritmoch si žiaci odskúšajú aj hľadanie kontrapríkladov – keďže greedy algoritmus nemusí vždy nájsť najlepšie riešenie. To súvisí s oblasťou Algoritmické riešenie problémov – analýza problému. Žiaci sa potrebujú naučiť jednak hľadať rôzne spôsoby riešenia, ale aj uvažovať, či sú dané spôsoby správne a či neexistujú prípady, pre ktoré ich riešenie nemusí byť korektné.

Úloha Oblečenie môže predstavovať úlohu na vykonávanie príkazov podľa určitých podmienok. Keďže sme zistili, že hlavne mladší žiaci majú problém vyhodnotiť dve podmienky, je dôležité, aby sa naučili rozdeliť problém na menšie časti a tie riešiť samostatne. Takáto úloha a jej podobné žiakov učia aj interpretácii príkazov a podmienok z prirodzeného jazyka.

Obe úlohy sú vhodné aj ako „unplugged“ aktivity na hodinách informatiky – keď už z rôznych dôvodov nemajú žiaci prístup k počítačom. Dajú sa ľahko riešiť na papier a dá sa k nim vymyslieť veľa rôznych variácií (pri grafovej – zmena grafu, zmena zadania, iný počet vrcholov a pod., pri zásobníkovej – iné pravidlá, viac pravidiel, viac zásobníkov, pridávanie do zásobníka a pod.). Prípadne učiteľom odporúčame využiť podobné úlohy z archívu iBobra [12] alebo úlohy z prostredia Bobrovo [3].

5. ZÁVER

V našom článku sme analyzovali dve úlohy z aktuálneho ročníka súťaže iBobor. Zamerali sme sa na úlohy obsahujúce abstraktné dátové typy v kategórii Benjamíni a vyhodnotili sme ich riešenia. Naše hypotézy o náročnosti úloh sa nepotvrdili, obe úlohy boli pre žiakov náročnejšie, ako sme predpokladali. Celková úspešnosť úlohy Oblečenie bola 45,68%, pričom sme predpokladali, že bude vyššia ako 55%. Úspešnosť piatakov v tejto úlohe bola približne 33% a siedmakov s primánmi 57%. Celková priemerná úspešnosť úlohy Reštaurácie bola 27,98%, čo je o pár percent menej ako nami predpokladaná hranica 30%, pričom piataci dosiahli úspešnosť iba približne 16% a siedmaci so sekundánmi cez 40%. To však ale potvrdzuje hypotézu, že tieto úlohy budú mať nižšiu úspešnosť v piatom ako v siedmom ročníku. Rovnako tak sa potvrdilo, že chlapci aj dievčatá riešia úlohy tohto typu s veľmi podobnou úspešnosťou.

Pri úlohe Oblečenie, ktorá obsahovala dva zásobníky, veľká časť piatakov chybným odvodením použila pre oba zásobníky rovnaké pravidlo. Vo vyšších ročníkoch bola táto chyba menej častá ako výskyt správnej odpovede. Zistili sme, ako spomínajú aj iné výskumy, že aj pri tejto úlohe boli rozdiely medzi chlapcami a dievčatami v konečných výsledkoch veľmi malé, chlapci však častejšie tipovali odpoveď, kým dievčatá vyberali častejšie odpoveď chybnou úvahou.

Pri úlohe Reštaurácie bol veľký rozdiel medzi výsledkami piatakov a siedmakov. Úloha bola ťažká, čo mohlo byť spôsobené jednak abstraktnou štruktúrou, ktorej reprezentácii mali žiaci rozumieť, a pre mladších žiakov možno nie úplne jasnému zadaniu. Predpokladáme, že správne riešenia najčastejšie žiaci dosiahli použitím intuitívneho greedy algoritmu aj s možným použitím backtrackingu. To, že interaktívna aplikácia umožňovala označiť aj iný počet vrcholov ako 3, spôsobilo určitú časť zlých riešení – chyba ale prvotne nastala pri zlom prečítaní zadania, kde bol počet vrcholov explicitne spomenutý. Rozdiely medzi chlapcami a dievčatami opäť neboli v konečných výsledkoch veľké, ale tiež sa líšili najmä v chybách, ktoré robili. Chlapci častejšie ako dievčatá označili iný počet vrcholov ako 3, ak však označili tri vrcholy, väčšie percento týchto odpovedí bolo správne ako pri dievčatách.

Obe úlohy riešili správne skôr šikovnejší súťažiaci. Priemer celkového počtu bodov súťažiacich, ktorí mali správne riešenia v týchto úlohách, sa pohyboval na hranici 50 bodov a tí žiaci, ktorí vyriešili obe úlohy správne, mali v priemere až 58,32 bodu.

Myslíme si, že takéto úlohy sú dobrými námetmi na vyučovanie informatiky podľa iŠVP pre oblasti Reprezentácie a nástroje, konkrétne časť Štruktúry, a Algoritmické riešenie problémov. Dajú sa použiť aj v neelektronickej forme, prípadne sa veľká časť podobných úloh dá nájsť v archíve súťaže iBobor a prostredí Bobrovo.

BIBLIOGRAFICKÉ ODKAZY

[1] A. Hrušecká, J. Pekárová, P. Tomcsányi a M. Tomcsányiová, „Informatický bobor – nová súťaž v informačných technológiách pre žiakov základných a stredných škôl," rev. *Didinfo*, Banská Bystica, 2008.

[2] „iBobor - štatistické vyhodnotenie 2016/17," 2016. [Online]. Dostupné na: http://ibobor.sk/vyhodnotenie 2016.php. [Cit. 2 Január 2017].

[3] D. Bezáková, A. Hrušecká a R. Hrušecký, „Využívanie prostredia Bobrovo a jeho nové možnosti," *Didinfo 2013*, pp. 40-46, 2013.

[4] V. Dagienė a G. Futschek, „Bebras international contest on informatics and computer literacy: criteria for good tasks," *R. T. Mittermeir, M.M. Syslo (Eds.), Lecture Notes in Computer Science*, vol. 5090, pp. 19-30, 2008.

[5] Štátny pedagogický ústav, „Inovnaný štátny vzdelávací program - Informatika," 2014. [Online]. Dostupné na: www.statpedu.sk/sites/default/files/dokumenty/inovovany-statny-vzdelavaci-program/informatika_nsv_2014.pdf. [Cit. 10 Január 2017].

[6] J. M. Wing, „Computational Thinking - What and Why?," 6 Marec 2011. [Online]. Dostupné na: www.cs.cmu.edu/ link/research-notebook-computational-thinking-what-and-why. [Cit. 10 December 2016].

[7] V. Dagienė a G. Stupurienė, „Bebras – a Sustainable Community Building Model for the Concept Based Learning of Informatics and Computational Thinking," *Informatics in Education*, zv. 15, %1. vyd.1, pp. 25-44, 2016.

[8] V. Dagienė, L. Mannila, T. Poranen, L. Rolandsson a G. Stupurienė, „Reasoning on children's cogni- tive skills in an informatics contest: findings and discoveries from Finland, Lithuania, and Sweden," *LNCS*, %1. vyd.8730, pp. 66-77, 2014.

[9] J. Piaget, The Psychology of Intelligence, Totowa, N.J.: Littlefield, 1972.

[10] S. Hansky, K. Schlüter a T. Brinda, „Typical errors of learners in informatics tasks of the international Bebras contest," rev. *Proceedings of the IFIP Working Conference "ICT and Informatics in a Globalized World of Education"*, 2011.

[11] A. Erdősné Németh a L. Zsakó, „The place of the greedy method concept in the progression of contestants' thinking," rev. *DidMatTech*, Budapest, 2016.

[12] iBobor, „Archív úloh," 2008 - 2016. [Online]. Dostupné na: http://ibobor.sk/sutaz_demo/. [Cit. 14 Január 2016].

DIDINFO & *DIDACTIG 2017*. **Univerzita Mateja Bela, Banská Bystrica 2017.**

Wiki na strednej škole
Wiki in high school

Iveta Csicsolová
KZVI FMFI UK
Mlynská dolina
842 48 Bratislava
Slovensko
ivetkacs@gmail.com

Zuzana Kubincová
KZVI FMFI UK
Mlynská dolina
842 48 Bratislava
Slovensko
kubincova@fmph.uniba.sk

ABSTRACT
The aim of organized education in school is to prepare the students for their professional life. Therefore, the educational process should focus not only on acquiring knowledge needed in their future job but also on the development of those competencies that are required by the employers. The teamwork competence is one of them. Integration of an appropriate tool into the teaching and well-designed educational activities can help to develop this competence in students, already in school. Wiki – a Web 2.0 tool which supports constructivist and collaborative learning, development of communication skills, critical thinking etc., can serve as such tool. In this paper we describe our experience of using wiki in informatics classes at high school.

Keywords
Wiki. Collaborative learning. Informatics. High School.

ABSTRAKT
Škola by mala pripraviť žiakov pre ich profesionálny život. Vzdelávací proces by preto mal byť zameraný nielen na to, aby žiaci a študenti nadobudli vedomosti, ktoré budú potrebovať pre svoje budúce zamestnanie, ale aj na rozvoj kompetencií, ktoré od absolventov vyžadujú zamestnávatelia. Jednou z nich je aj kompetencia k tímovej práci. Zapojením vhodného nástroja a dobre navrhnutých vzdelávacích aktivít možno túto kompetenciu rozvíjať u žiakov a študentov už v škole. Takýmto nástrojom je napr. wiki, ktorá podporuje konštruktivistické kolaboratívne učenie sa, rozvoj komunikačných schopností, kritického myslenia a pod. V tomto článku prinášame naše skúsenosti s využitím wiki na hodinách informatiky na strednej škole.

Kľúčové slová
Wiki. Kolaboratívne učenie. Informatika. Stredná škola.

1. ÚVOD
Ako potvrdzuje viacero štúdií [1, 2], medzi základné kompetencie, ktoré v súčasnosti očakávajú zamestnávatelia od svojich zamestnancov, patrí aj schopnosť efektívne komunikovať a kompetencia k tímovej práci, čiže schopnosť efektívne spolupracovať s inými ľuďmi ako člen skupiny. Podľa prieskumu medzi zamestnávateľmi v rámci projektu Uni2010 [2] však obe tieto kompetencie patria medzi päť najmenej rozvinutých kompetencií absolventov.

Keďže úlohou školy je komplexne pripraviť jej absolventov na ich ďalší profesionálny život, je potrebné zabezpečiť, aby žiaci a študenti počas štúdia získavali nielen nové vedomosti, ale aby tiež rozvíjali také svoje kompetencie, ktoré im umožnia úspešne sa zamestnať a zvládnuť efektívne riešenie rôznych pracovných situácií. Od učiteľa sa teda očakáva, že bude do svojho vyučovania zapájať aj také vzdelávacie postupy a aktivity, ktoré podporia rozvoj požadovaných kompetencií jeho žiakov, či študentov.

Pri hľadaní vzdelávacích aktivít podporujúcich rozvoj komunikačných schopností a kompetencií k tímovej práci sme sa zamerali najprv na nájdenie vhodného podporného nástroja, ktorý by umožňoval žiakom kolaboratívne pracovať na zadaných úlohách. Nakoľko súčasná generácia mladých ľudí má veľmi blízky vzťah k digitálnym technológiám a k internetu, sústredili sme sa na nástroje Webu 2.0, ktoré ponúkajú viacero zaujímavých možností využitia vo vyučovaní [3]. Vybrali sme spomedzi nich wiki, ktorú považujeme za nástroj najvhodnejší na naše účely.

V predchádzajúcom polroku tohto školského roku sme zrealizovali experiment, pri ktorom stredoškoláci využívali wiki na vypracovávanie kolaboratívnych zadaní na informatike. V tomto príspevku prinášame naše skúsenosti z experimentu.

2. WIKI
Wiki [4] je webová stránka, ktorá umožňuje používateľom upravovať jej obsah. Vo väčšine wiki systémov môže ktokoľvek nielen pridávať nový príspevok a upravovať príspevok, ktorý pridal on sám, ale môže taktiež upravovať príspevok, ktorý pridal niekto iný, prípadne ho môže aj úplne zmazať. Ďalšími užitočnými črtami wiki je možnosť pridávať ku každej stránke diskusiu, sledovať históriu vývoja každej wiki-stránky, prepájať jednotlivé stránky pomocou hypertextových odkazov a vytvárať tak sieť tematicky príbuzných stránok, používať kategorizáciu a vyhľadávanie. Vďaka týmto vlastnostiam je wiki vhodným nástrojom na kolaboratívnu prácu.

Pôvodne bola navrhnutá na kolaboratívnu tvorbu dokumentov a primárne sa nepočítalo s jej využitím na vzdelávacie účely. Avšak jednoduchosť vytvárania wiki-stránok a ich obsahu, bezproblémová navigácia, ako aj benefity, ktoré používanie wiki poskytuje, z nej robia vzdelávací nástroj s veľkým potenciálom. Podporuje konštruktivistické učenie sa [5], pri ktorom žiaci aktívne a kolaboratívne konštruujú nové vedomosti, integrujú ich do systému predchádzajúcich vedomostí, utvárajú si spoločný názor na daný problém.

Wiki môže byť vo vzdelávaní použitá rôznymi spôsobmi [6], niekoľko z nich uvádzame v nasledujúcom zozname:

- pri vývoji projektu môžu študenti použiť wiki ako projektovú dokumentáciu svojej práce

- študenti môžu vo wiki sumarizovať svoje úvahy o povinnom čítaní a vybudovať kolaboratívne anotovanú bibliografiu
- pri dištančnom vzdelávaní môže učiteľ publikovať vo wiki vzdelávacie zdroje a študenti ich môžu editovať a komentovať
- učitelia môžu wiki využiť na zdieľanie svojich skúseností a reflexií z vyučovania, vymieňať si a spoločne zdokonaľovať materiály, učebné postupy a metódy
- wiki možno použiť na mapovanie konceptov, brainstorming a tvorbu navzájom prepojenej siete vzdelávacích zdrojov
- wiki môže byť použitá na hodnotenie kurzov študentmi a pod.

O ďalších možnostiach použitia wiki vo vzdelávaní, predovšetkým na univerzite, referuje množstvo publikácii. Wiki bola napr. použitá na anotáciu študijných materiálov študentmi [7]. Po schválení učiteľom boli vylepšené materiály zdieľané a pripravené na použitie budúcimi študentmi. Na kurze *Návrh algoritmov* na University of Oviedo študenti vytvorili vo wiki knižnicu algoritmických problémov spolu s návrhmi na ich riešenie [8]. Študenti z Griffit University v Austrálii kolaboratívne pripravili vo wiki učebnicu ku kurzu o mobilných technológiách [9]. Na University of Nebrasca študenti histórie kolaboratívne písali vo wiki eseje [10].

V zahraničných publikáciách možno nájsť aj príklady experimentov s wiki na strednej škole [11, 12, 13], avšak nevieme o žiadnej relevantnej publikácii, popisujúcej použitie wiki na slovenských školách.

3. WIKI NA STREDNEJ ŠKOLE – NÁŠ VÝSKUM

Výskum bol realizovaný na troch stredných školách v spolupráci s dvoma učiteľkami, počas troch mesiacov v prvom polroku školského roku 2016/2017. Žiaci boli rozdelení do troch výskumných skupín. Skupinu A tvorili žiaci zo školy č. 1, skupinu B časť žiakov zo školy č. 2 a zvyšná časť žiakov zo školy č. 2 bola spolu so žiakmi zo školy č. 3 v skupine C. Výskumu sa spolu zúčastnilo 86 žiakov. V tomto článku sa bližšie zameriame na výskumnú skupinu C.

Cieľom nášho výskumu bolo priniesť nový rozmer do vyučovania informatiky a ukázať možnosti využitia wiki vo vyučovaní. Zaujímalo nás, akým spôsobom žiaci pracovali s wiki, aký majú na ňu názor a či by na základe tejto novej skúsenosti chceli wiki používať aj naďalej. Chceli sme poukázať aj na dôležitosť práce s informáciami (vyhľadávanie, spracovávanie a citovanie) a spoluprácu žiakov v rámci skupiny, žiakov z tej istej školy, ako aj žiakov z rôznych škôl.

3.1 Výskumná vzorka

Výskumnú vzorku tvorila skupina C, ktorá bola zložená zo žiakov z dvoch škôl. Skupinu tvorilo 47 žiakov: 24 žiakov zo školy č. 2 s anglickým vyučovacím jazykom a 23 žiakov zo školy č. 3 so slovenským vyučovacím jazykom. Žiaci z jednej školy boli tretiaci, žiaci z druhej školy boli štvrtáci, avšak v tomto prípade šlo o päťročnú bilingválnu školu, a teda žiaci vo štvrtom ročníku preberali učivo tretieho ročníka bežného gymnázia. Vyučovanie na oboch školách prebiehalo paralelne a žiakov učila rovnaká učiteľka. Všetci žiaci pracovali na báze medziškolskej kolaborácie a zároveň boli jazykovo zmiešaní. Žiaci boli učiteľkou rozdelení do 12 tímov po 4 žiakoch (až na jeden tím, v ktorom boli len traja žiaci). Tímy boli tvorené vždy dvomi žiakmi zo školy č. 2 a dvomi (jedným) zo školy č. 3. Žiaci z rozličných škôl sa medzi sebou nepoznali.

3.2 Zber údajov

Údaje do nášho výskumu sme získali z rozličných zdrojov. Využili sme dotazníkovú metódu, interview, výsledky testu, ako aj analýzu žiackych príspevkov – wiki-stránok.

Dotazník sme použili dvakrát: (a) pred začiatkom aktivity vo wiki žiaci vyplnili vstupný dotazník, ktorého cieľom bolo zistiť nakoľko sú žiaci oboznámení s kolaboratívnou prácou, či poznajú wiki a spôsob práce v nej, či na hodinách informatiky zvyknú používať niektoré z nástrojov Webu 2.0, odkiaľ zvyknú čerpať informácie pri spracovávaní referátov, či vedia citovať informačné zdroje a pod.; (b) po skončení aktivity sme sa v záverečnom dotazníku pýtali žiakov na ich názory na kolaboratívnu aktivitu vo wiki, na samotnú wiki a jej nástroje, na spoluprácu so žiakmi z inej školy, na spôsoby komunikácie v tíme, na benefity, ktoré žiakom táto aktivita priniesla a pod.

Pomocou interview sme zisťovali názory zapojených učiteliek na túto aktivitu, na prácu žiakov z pohľadu učiteľa, na plusy a mínusy použitého nástroja i celej kolaboratívnej aktivity.

3.3 Priebeh aktivít

Zadanie pre žiakov predstavovalo vytvorenie informatickej encyklopédie. Encyklopédia mala obsahovať niekoľko wiki stránok a každá stránka bola zameraná na nejakú informatickú tému prípadne oblasť alebo pojmy z tematického celku Počítačové siete a Internet. Stránka mala byť vypracovaná formou článku. Na každej stránke mal pracovať jeden tím žiakov. Žiaci si mali na základe výkladu učiteľa počas vyučovania spracovať na domácu úlohu dostatočne kvalitný a zrozumiteľný učebný materiál, z ktorého by sa neskôr mohli učiť na písomku alebo maturitu.

Priebeh celej aktivity bol rozdelený na niekoľko týždňov. Prvé tri týždne mali žiaci na vyplnenie vstupného dotazníka. Učiteľka počas týchto týždňov rozdelila žiakov do skupín a pripravovala témy vhodné na spracovanie. Našou úlohou bolo pripraviť prihlasovacie kontá pre žiakov a predpripraviť stránky vo wiki. Na účely výskumu sme použili platformu MediaWiki na portáli wiki.matfyz.sk. Žiaci pred začatím aktivity dostali inštrukcie od učiteľa a niekoľko ďalších informácií k výskumu[1].

Aktivita bola rozdelená na 4 kolá. Nepárne kolá boli určené nepárnym tímom, párne kolá párnym tímom. Zoznam spracovávaných tém a ich rozdelenie do jednotlivých kôl je uvedené v Tabuľke 1. Žiaci mali na spracovanie stránky spravidla týždeň (s výnimkou prvého kola, keď v jednej škole meškali s prebraným učivom potrebným na spracovanie témy a bolo potrebné predĺžiť kolo o jeden týždeň).

Úlohou žiakov bolo rozdeliť si prácu v tíme. Jedna časť žiakov mala spracovať danú tému v angličtine, druhá časť v slovenčine. Žiaci mali medzi sebou komunikovať pomocou diskusie priamo vo wiki. Chceli sme dosiahnuť, aby si žiaci sami riadili prácu vo svojom tíme vrátane rozdelenia práce medzi sebou, aby si

[1] Informácie pre žiakov zverejnil učiteľ na svojej webovej stránke: http://linterfop.evlyceum.sk/informacie-k-wiki/

navzájom pomáhali a komunikovali aj s členmi svojho tímu, ktorí boli z inej školy. Celý priebeh práce každého žiaka bolo možné sledovať vďaka histórií konkrétnej stránky a rozšíreniu ActivityWatch [14].

Tabuľka 1. Spracované témy

Téma	Kolo
Zaujímavé fakty o e-mailoch	1.
E-mailoví klienti	1.
SMTP server	1.
POP3 server	1.
IMAP server	1.
Diskusné fórum	1.
Interaktívna komunikácia	2.
WWW	2.
Vyhľadávanie na internete	2.
WWW prehliadače a ich funkcie	2.
Cookies	2.
Iné služby internetu	2.
Počítačová sieť	3.
Hardvér-sieťová karta a repeater	3.
Hardvér-hub a switch	3.
Hardvér-router a modem	3.
Firewall	3.
Káble-krútená dvojlinka	3.
Bezdrôtové spojenie	4.
Satelitné spojenie	4.
Optické vlákna	4.
Softvér počítačových sietí	4.
Delenie počítačových sietí podľa rozlohy	4.
Delenie počítačových sietí podľa typológie	4.

Po každom kole sme vypracovali podrobnú správu pre učiteľku, v ktorej sme analyzovali každú jednu wiki-stránku z vizuálnej i formálnej stránky (použité nástroje wiki, uvádzanie citácií a citovanie použitej literatúry, rozsah kopírovania z uvedených aj neuvedených zdrojov), ako aj z pohľadu spolupráce žiakov (akým percentom sa podieľal konkrétny žiak na danej stránke, aký rozsah obsahu na danej stránke pridal žiak). Učiteľka skontrolovala obsahovú časť danej stránky. Na základe obsahu a našej správy potom pridelila jednotlivým žiakom v skupine počet bodov. Učiteľka tiež každej skupine dávala okamžitú spätnú väzbu – priamo do diskusie napísala odôvodnenie svojho hodnotenia, upozornila na chyby a vyzdvihla dobre spracované časti.

Posledný týždeň experimentu písali žiaci oboch škôl test z celého tematického celku, ktorý mali spracovaný v podobe wiki stránok. Po teste vyplnili žiaci záverečný dotazník.

4. VÝSLEDKY

4.1 Analýza žiackych príspevkov

Ako sme už uviedli vyššie, po každom kole boli jednotlivé wiki-stránky podrobené analýze a taktiež bola skúmaná aj práca jednotlivcov.

V prvom rade sme sledovali, či sa každý člen tímu zapojil do práce na wiki-stránke a akým percentuálnym podielom. Túto možnosť nám poskytlo rozšírenie ActivityWatch. Toto rozšírenie však počíta percentá vzhľadom na počet editov stránky. Preto bolo nutné skontrolovať prácu žiakov aj v histórii. V prvom a druhom kole väčšina skupín nepoužívala citáciu. Udali zdroje, avšak bez citácie. Taktiež nepridávali obrázky, nevyužívali nástroje, ktoré VisualEditor poskytuje. Bolo to pravdepodobne spôsobené tým, že žiaci sa s wiki iba zoznamovali, nepovažovali za nutné pridať tematický obrázok a neboli zvyknutí/naučení citovať. Diskusiu používali veľmi zriedka.

Preto pred ďalším kolom boli žiaci upozornení na to, aby citovali a udávali zdroje. Taktiež aby text nekopírovali, ale ho spracovávali. Mali pridať aj tematické obrázky, ktoré môžu učivo oživiť alebo pomôcť pochopiť. Pri obrázkoch veľa skupín zabudlo udať zdroje, prípadne si neoverilo ani licenciu na používanie obrázkov.

Najdôležitejšia však bola komunikácia a spolupráca členov tímu. Žiaci svoju prácu v nasledujúcim kole naozaj zlepšili a väčšina žiakov sa zapájala aktívnejšie aj do diskusie. Mali sme možnosť sledovať v diskusii vzájomné rozdelenie práce, upozorňovanie na chyby, radenie a posúvanie zdrojov aj pre ostatných členov. V niektorých tímoch bolo vidieť spoluprácu medzi všetkými členmi, nie len členmi z rovnakej školy. V diskusii všetci komunikovali v slovenskom jazyku.

Najmä v prvých kolách zlyhala komunikácia aj z toho dôvodu, že žiaci spracovávali témy na posledný deň. Preto nemali možnosť dohodnúť sa s ostatnými členmi a každý člen tímu individuálne vypracoval niečo zo zadanej témy. V niektorých tímoch sa zlepšilo časové obdobie spracovávania na 2-4 dni pred dátumom odovzdania.

Analyzovali sme aj plagiátorstvo. Plagiátorstvo zo strany iných výskumných skupín bolo zamedzené vytvorením menných priestorov, ktoré zabezpečili, aby jednotlivé wiki-stránky videli iba členovia priestoru. Žiacke kopírovanie z iných webových stránok sme kontrolovali manuálnym kopírovaním náhodných odsekov do vyhľadávača a hľadaním zhôd. Bolo niekoľko menších prípadov, kedy bola nájdená časť odkopírovaného textu bez citácie alebo zdroja. Učiteľka vyhodnotila prípad a následne v odôvodnených prípadoch znížila počet bodov za prácu.

Každý žiak mohol získať maximálne 10 bodov za každé kolo. Žiaci spracovávajúci stránku v angličtine, získali v priemere 8 bodov, žiaci píšuci v slovenčine, získali priemerne 9 bodov. Bolo niekoľko málo prípadov, kedy žiaci získali nula bodov, pretože sa (bez odôvodnenia) nezúčastnili na aktivite. Každý žiak mal možnosť získať v každom kole bonusové body za aktivitu navyše. Bonusové body získali žiaci, ktorí opravili chyby, pomohli žiakom z iných tímov alebo prejavili nadmernú snahu pri spracovávaní wiki-stránky.

Žiaci si pravdepodobne čítali aj hodnotenia z predchádzajúcich kôl a snažili sa vyvarovať svojich chýb alebo chýb iných tímov.

Pri spracovávaní stránky si na vizuálnom hľadisku žiaci väčšinou dali záležať. Stránky mali štruktúru, používali nadpisy aj podnadpisy, zvýrazňovali dôležité časti textu, neskôr text obohacovali obrázkami. Z obsahového hľadiska bola väčšina stránok spracovaná dobre. Niekoľko stránok bolo nedostatočne spracovaných alebo sa v nich nachádzali nesprávne informácie. Na to boli žiaci upozornení a chyby opravili, aby ich wiki-stránka mohla byť použitá ako študijný materiál aj pre ostatných žiakov.

Najlepší výsledok, ktorý sme postrehli pri analýze jednotlivých kôl bol ten, že žiaci sa skutočne naučili citovať a udávať zdroje. Uvedomili si dôležitosť toho, že informácie by mali nájsť, posúdiť ich relevantnosť, pochopiť, spracovať a správne odcitovať.

Dôkazom toho bolo aj niekoľko výborne spracovaných stránok, prípadne len s malými chybami.

Učiteľka nám v interview uviedla, že žiaci občas spracovali na wiki-stránke dokonca viac, ako považovala za potrebné. Preto im v rámci spätnej väzby v diskusii ku každej téme napísala, čo sa od nich očakáva, že budú z danej témy vedieť a čo sú vedomosti nad rámec príslušného učiva.

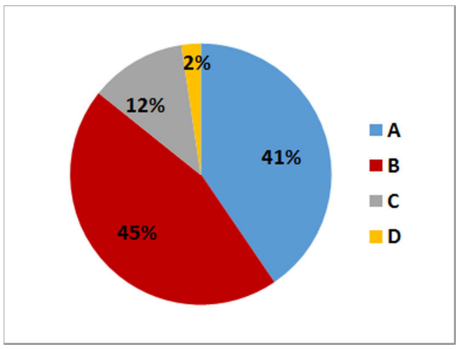
4.2 Výsledky testu

Týždeň po ukončení aktivity písali žiaci záverečný test z tematickej oblasti Počítačové siete a Internet. Test obsahoval otázky iba z tých tém, ktoré žiaci sami spracovali. Celková priemerná úspešnosť na teste bola 75%. Podľa názoru učiteľky bol vzhľadom k náročnosti testu tento výsledok „úctyhodný výkon“ a to, že si žiaci vypracovali témy sami a nedostali hotové materiály od nej, určite nemalo negatívny vplyv na ich výkony.

4.3 Názory žiakov na aktivitu vo wiki

Po teste sme požiadali žiakov o vyplnenie záverečného dotazníka. Dotazník pozostával z 15 otázok a vyplnilo ho 42 žiakov zo 47.

Na otázku, ako sa im páčila aktivita vo wiki, 45,2% žiakov uviedlo neutrálnu odpoveď („dalo sa“), 40,5% uviedlo, že aktivita sa im páčila, 11,9% uviedlo, že aktivita ich nebavila a jeden žiak uviedol inú odpoveď: „nebolo to efektívne“ (viď Obrázok 1.). Fakt, že odpovedí, ktoré boli vyslovene pozitívne, bolo 3,4-krát viac než tých, ktoré boli vyslovene negatívne, považujeme za úspech tejto aktivity.



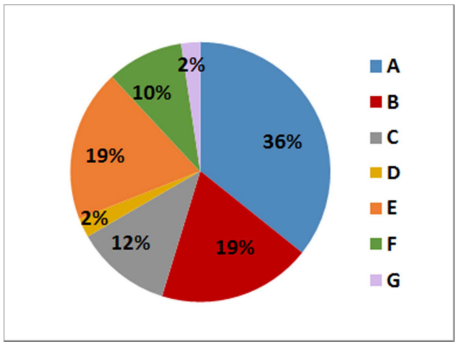
Obrázok 1. Záverečný dotazník: *Páčila sa Ti aktivita vo wiki?* A–Áno, bolo to fajn; B–Dalo sa; C–Nie, nebavilo ma to; D–Iné

Ďalej nás zaujímalo, či žiaci považovali túto aktivitu za náročnú. Odpovede boli veľmi rozdielne (viď Obrázok 2.). 35,7% žiakov uviedlo, že aktivita nebola náročná a 19% uviedlo, že bola primerane náročná v porovnaní s inými úlohami, ktoré majú. Ostatní žiaci mali problém pracovať ako tím celkovo alebo v tíme s inou školou, niektorí žiaci by vyžadovali viac času a iba jeden žiak uviedol, že nepochopil zadanie.

Žiaci dostali priestor na vyjadrenie svojho názoru na celú aktivitu. Ich reakcie boli pozitívne aj negatívne. Mnoho žiakov sa sťažovalo na malé zapájanie niektorých členov tímu. Na ukážku uvádzame niektoré zaujímavé odpovede:

„Páčilo sa mi, že sme mohli niečím prispieť, a všetci sa snažili robiť to poriadne, s víziou písomky. Resp. nikto si nedovolil odfláknuť to, pretože vedel že sa z toho bude učiť on, aj jeho spolužiaci.“

„Zo začiatku mi to prišlo chaotické a nevedela som čo mám robiť, ale potom to už bolo fajn a ani to nebolo časovo náročné. Bolo to oveľa lepšie ako programovanie.“



Obrázok 2. Záverečný dotazník: *Bola pre Teba aktivita náročná? Ako?* A–Nebola náročná, zvládol/a som to s prehľadom; B–Bola primerane náročná v porovnaní s inými úlohami, ktoré mávame; C–Bola náročná iba na čas, inak nie; D–Bolo to ťažké, nevedel/a som ani čo mám robiť, nepochopil/a som dobre zadanie; E–Samotná aktivita nebola náročná, náročné bolo pracovať ako tím; F–Samotná aktivita nebola náročná, náročné bolo pracovať v tíme s inou školou; G–Iné

„Snažila som sa zapájať do komunikácie, ale zvyšok tímu nemal veľmi záujem. Ale nie všetci sú navyknutí pracovať ako tím, preto to bolo niečo nové pre nás všetkých a určite takúto aktivitu budem podporovať.“

„Páči sa mi, že máme všetky informácie na jednej stránke, takže sme si na hodine nemuseli robiť poznámky a moli sme počúvať. Negatívne – žiaci z inej školy spravili svoju časť práce bez akejkoľvek diskusie.“

„Aktivita zrovna náročná, ale vzhľadom na to, že sme dané témy preberali na hodinách, trocha zbytočná. Na druhej strane, učenie sa bolo o niečo ľahšie.“

„Typ aktivity sa mi páčil ale s prácou v tíme som bola nespokojná. Žiaci s ktorými som mala pracovať s nami nespolupracovali a nekomunikovali. Práca nikdy nebola uroPbená načas alebo nedokončená. Ale nápad na prácu sa mi páčil.“

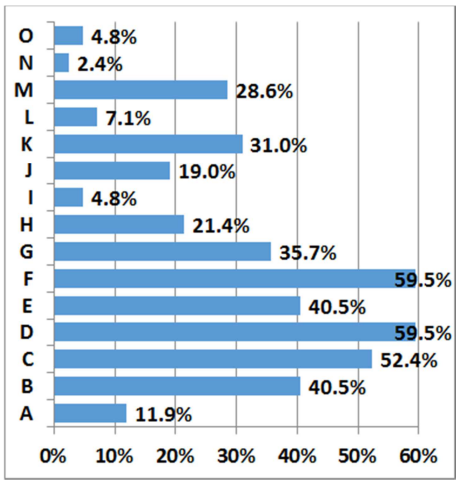
„Takéto aktivity ma veľmi nebavia.“

Zaujímali nás aj technické problémy s wiki, ktoré mali žiaci. Väčšina žiakov nemala žiadne, avšak keď na jednej wiki v rovnakom čase pracovalo viacero žiakov, u niektorých sa vyskytli problémy s uložením a žiakom sa neuložil napísaný text. Aby sme predišli problémom s orientáciou vo wiki, vopred sme vytvorili návod[2], v ktorom sme žiakom vysvetlili ako pracovať s wiki. Žiaľ, v priebehu aktivity sa ukázalo, že mnohí žiaci tento návod nečítali a radšej sa spoliehali na rady spolužiakov, alebo konzultovali svoje problémy s učiteľkou.

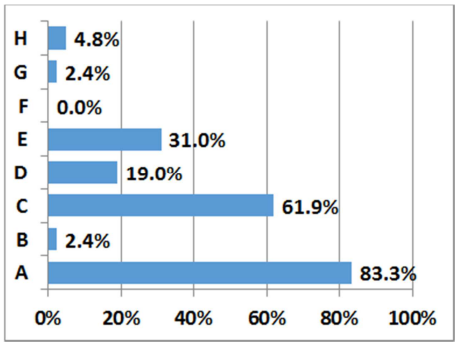
Za dôležitú pre zistenie postoja žiakov k tejto aktivite považujeme otázku, v ktorej sme zisťovali, čo im táto aktivita dala, čo nové sa naučili. Žiaci mohli zaškrtnúť ľubovoľný počet odpovedí. Najčastejšie sa vyskytujúce odpovede uvádzali, že žiaci sa naučili citovať a udávať zdroje (25), používať nový nástroj (25), že sa

[2] Návod je zvejenený vo wiki: http://wiki.matfyz.sk/N%C3%A1vod

naučili veľa o spracovávanej téme (22), spolupracovať na projektoch aj so žiakmi z inej školy (17) a vyhľadávať a spracovávať informácie (17). Všetky odpovede na túto otázku možno vidieť na Obrázku 3.



Obrázok 3. Záverečný dotazník: *Čo Ti dala táto aktivita? Čo nové si sa naučil/a*? A– Naučil/a som sa pracovať v tíme; B–Naučil/a som sa spolupracovať na projektoch aj s ľuďmi z inej školy; C–Naučil/a som sa veľa o téme, ktorú som spracovával/a; D–Naučil/a som sa citovať a udávať zdroje; E–Naučila/a som sa vyhľadávať a spracovávať informácie; F–Naučil/a som sa používať nový nástroj, ktorý som predtým nepoužíval/a; G–To, čo som spracovával/a, som lepšie pochopila a lepšie si zapamätal/a; H–Lepšie som si utvrdil/a učivo, ktoré som mal/a predtým vysvetlené od učiteľa; I–Naučil/a som sa lepšie písomne vyjadrovať v slovenskom jazyku; J–Naučil/a som sa lepšie písomne vyjadrovať v anglickom jazyku; K–Naučil/a som sa písať odbornú prácu, čo sa mi zíde v ďalšom štúdiu; L–Učivo, ktoré sme si takto spracovali, sa mi učí oveľa ľahšie, než z materiálov od učiteľa; M–Učivo, ktoré sme si takto spracovali, sa mi učí oveľa ľahšie, než z učebnice alebo iných zdrojov; N–Spoznal/a som nových ľudí z inej školy, s ktorými zostanem v kontakte; O–Iné („Nič“)



Obrázok 4. Záverečný dotazník: *Ako ste komunikovali počas práce na aktivite medzi sebou v tíme?* A–Diskusia vo wiki; B–Vnútorné komentáre vo wiki pri editácii stránky; C–Chatovanie mimo wiki; D–E-mail; E–V škole; F–Stretávali sme sa po škole; G–Telefonáty (Skype, mobil); H–Iné

Keďže komunikácia v niektorých prípadoch medzi žiakmi zlyhala, zaujímali sme sa o to, ako komunikovali medzi sebou. Žiaci mohli označiť viaceré odpovede: 35 žiakov uviedlo, že používalo diskusie vo wiki, 26 žiakov chatovalo mimo wiki, 8 žiakov komunikovalo mailom a 13 žiakov sa dohadovalo v škole (viď Obrázok 4.).

Podľa našich predchádzajúcich skúseností s používaním toho nástroja vo vzdelávacích aktivitách, študenti často uprednostňovali osobnú komunikáciu, prípadne komunikáciu prostredníctvom sociálnych sietí alebo iných komunikačných nástrojov, pred komunikáciou priamo vo wiki. Preto považujeme uvedené výsledky za veľmi priaznivé.

Pri otázke, ako by zhodnotili svoju celkovú prácu vo wiki, žiaci reagovali veľmi pozitívne. Odpovedali, že sa snažili a to bolo viditeľné aj na výsledkoch v podobe hotových wiki-stránok, ako aj na počte získaných bodov.

V závere dotazníka mohli žiaci uviesť školské aktivity, pri ktorých si vedia predstaviť používať wiki. Medzi odpoveďami sa vyskytli napr. nasledujúce:

- vypracovávanie maturitných tém
- vytvorenie školskej wiki na zdieľanie projektov, prác a informácií pre žiakov a učiteľov
- testy a dotazníky (pozn. to by bolo možné len pri skrývaní wiki-stránok, prípadne nejakým rozšírením)
- všetky informácie na jednom mieste aj z iných predmetov
- projekty a referáty
- učebné poznámky
- noviny a časopisy

Veľkým prínosom nášho výskumu bolo to, že vďaka tejto aktivite sme zistili, že žiaci môžu a dokážu spolupracovať aj s inou školou na rovnakom projekte. Žiaci si vyskúšali tímovú spoluprácu a zistili, že pri tímovej spolupráci je dôležitá komunikácia. I keď spolupráca u väčšiny tímov nebola ideálna, po upozornení učiteľkou sa v priebehu aktivity zlepšila a predpokladáme, že pri zopakovaní, resp. pokračovaní aktivity, by ju žiaci zvládli na ešte lepšej úrovni. Odpovede žiakov v dotazníku svedčia o tom, že mnohí žiaci považujú aktivitu za zaujímavú a vidia jej potenciálne využitie aj pri iných tematických celkoch z informatiky alebo dokonca aj na iných vyučovacích predmetoch.

5. ZÁVER

Výsledky nášho výskumu naznačujú, že wiki môže byť pri správnom použití vhodným nástrojom na kolaboratívne učenie sa.

Žiaci dokázali v rámci tejto aktivity spoločnými silami a bez pomoci učiteľky spracovať všetky zadané témy z preberanej tematickej oblasti na úrovni, ktorá nezriedka prekročila požadovanú úroveň ich vedomostí. Z takto pripravených materiálov boli schopní naučiť sa na test tak, aby ho zvládli absolvovať s veľmi dobrým výsledkom.

Pokiaľ ide o názory žiakov na túto aktivitu, napriek tomu, že viacerí z nich uvádzali spoluprácu v tíme ako problematickú, pozitívny postoj ku kolaboratívnej práci vo wiki zaujalo niekoľkonásobne viac žiakov, ako negatívny. Podľa odpovedí z dotazníka ako aj podľa názoru učiteľky, sa žiaci naučili spolupracovať, lepšie vyhľadávať a spracovávať informácie, správne citovať použité zdroje, zlepšili svoj písomný prejav, najmä pri písaní odborného textu a pod. Navyše, viac ako

polovica žiakov, ktorí vyplnili dotazník, potvrdila, že sa týmto spôsobom naučili veľa o téme, ktorú spracovávali.

Aj keď žiaci na aktivite tohto typu pracujú sami, úloha učiteľa je nesporne dôležitá. Mal by vopred žiakov oboznámiť s wiki a jej nástrojmi, ako aj s pravidlami písania odborného textu (napr. vyhľadávanie a overovanie zdrojov, citovanie, udávanie zdrojov, spracovávanie textu, nie len jeho kopírovanie a pod.) a najmä upozorniť ich, aby pracovali ako tím. Počas trvania aktivity by mal zasa poskytovať žiakom spätnú väzbu, poukázať na dobre i zle spracované časti ich príspevkov, povzbudzovať žiakov k aktívnej spolupráci.

Zaznamenali sme niekoľko drobných nedostatkov wiki, ktoré pri takýchto aktivitách môžu žiakom sťažiť prácu, napr. chýba automatické priebežné ukladanie práce, upozornenie na prácu viacerých členov na rovnakej stránke, možnosť vidieť, čo v rovnakom okamihu spracováva na stránke niekto iný a pod. V našej ďalšej práci sa preto plánujeme zamerať aj na ich odstránenie.

POĎAKOVANIE

Autorky ďakujú pani učiteľke PaedDr. Veronike Dropčovej, PhD., ktorá sa spolu so svojimi žiakmi podieľala na realizácii výskumu.

BIBLIOGRAFICKÉ ODKAZY

[1] ARENDÁŠOVÁ, A., KEHLÍKOVÁ A. Požiadavky zamestnávateľov na absolventov technických škôl. In *Materials Science and Technology*, 2005, vol. 3, no 5. Dostupné: http://www.mtf.stuba.sk/docs/internetovy_casopis/2005/3/arendasova.pdf

[2] CHALUPOVÁ, A. Aké kompetencie očakávajú zamestnávatelia od absolventov. In *Hospodárske noviny*, 2010, Dostupné: http://www.profesia.sk/en/cms/40398

[3] HOMOLA, M., KUBINCOVÁ, Z. Taking advantage of Web 2.0 in organized education (a survey) In *Proceedings of ICL 2009*, Villach, Austria

[4] LEUF, B., CUNNINHAM, W. The Wiki Way: Quick Collaboration on the Web, Addison-Wesley, 2001.

[5] PARKER, K. R., CHAO, J. T. Wiki as a teaching tool. In *Interdisciplinary journal of knowledge and learning objects*, 2007, vol. 3, no. 1, p.57-72.

[6] DUFFY, P., BRUNS, A. The use of blogs, wikis and RSS in education: A conversation of possibilities. In *Proceedings of the Online Learning and Teaching Conference 2006*, Brisbane. Dostupné: https://eprints.qut.edu.au/5398/

[7] GASPAR, M. I., SANTOS, M. A. Learning Communities: way to a new Teaching Model?, In *Proceedings of m-ICTE 2009*, Lisbon, Portugal

[8] PÉREZ, J., RUIZ, M., GAYO, J. Building a collaborative learning environment based in Wiki. In *Current Developments in Technology-Assisted Education, 2006*, p. 1413-1416.

[9] RUTH, A. , HOUGHTON, L. The wiki way of learning. In *Australasian journal of educational technology, 2009*, vol 25, no. 2

[10] SOH, L.K. ClassroomWiki: A collaborative wiki for instructional use with multiagent group formation. In *IEEE Transactions on Learning Technologies*, vol. 3, no. 3, pp.190-202

[11] CHANG, Y.F., SCHALLERT, D.L. The design for a collaborative system of English as foreign language composition writing of senior high school students in Taiwan. In *Fifth IEEE International Conference on Advanced Learning Technologies* (ICALT'05), 2005, p. 774-775

[12] FORTE, A., BRUCKMAN, A. Constructing text: Wiki as a toolkit for (collaborative?) learning. In *Procs. of the 2007 international symposium on Wikis*, 2007, ACM, p. 31-42.

[13] FORTE, A., BRUCKMAN, A. Writing, citing, and participatory media: Wikis as learning environments in the high school classroom. MIT Press, 2009, p. 23-44.

[14] JANAJEV, R. Tracking Students' Activity in Wiki.matfyz.sk. 2011, Univerzita Komenského, Bratislava, bakalárska práca.

Objavovanie informatických konceptov zážitkovými aktivitami

Discovering of Informatics Concepts through Experiential Activities

Martin Cápay
Katedra informatiky FPV UFK v Nitre
Tr. A. Hlinku 1
949 01 Nitra
Slovensko
mcapay@ukf.sk

ABSTRACT

The aim of the paper is to show that new concepts and principles of computer science, such as binary notation, encoding or secure data transfer, could be discovered through the constructivist games, manipulation with the objects and lots of running around. Presented activities are prepared as the experiential activities using only relatively simple teaching aids. The activities are prepared on the principles of Computer Science Unplugged method. We investigate how the visualization can help student to understand the basic concepts of computer, without unnecessary technical details and thus contribute to their better understanding.

Keywords

Computer science unplugged. Learning by discovery. Experimental learning. Visualisation.

ABSTRAKT

Cieľom príspevku je ukázať, že formou hier, pohybových aktivít a manipuláciou s predmetmi sa dajú objavovať nové pojmy a princípy počítačovej vedy, okrem iného napr. kódovanie a zobrazovanie informácií v počítači a zabezpečený prenos dát cez internet. Prezentované aktivity využívajú princíp nazývaný „Informatika bez počítača“. Aktivity sú založené na učení sa objavovaním bez priameho kontaktu s počítačom, len s použitím relatívne jednoduchých pomôcok. Ukážeme, ako môže vizualizácia prispieť k lepšiemu pochopeniu základných počítačových konceptov, bez zbytočných technických detailov.

Kľúčové slová

Informatika bez počítača. Učenie sa objavovaním. Zážitkové vzdelávanie. Vizualizácia.

1. ÚVOD

Za prejav záujmu žiaka o vzdelávacie aktivity sa všeobecne považuje dobré správanie, pozitívne pocity a uvažovanie [1]. Presne navrhnuté vzdelávacie činnosti môžu mať celkovo vplyv na úspech žiakov prostredníctvom zvýšenia behaviorálnej, emocionálnej, ako aj kognitívnej angažovanosti. Žiaci potrebujú byť vyzývaný k participácii a zodpovednosti a to bez strachu z prípadných chýb [2]. Aktívne zapojenie sa môže viesť k lepším výsledkom počas celého života [3]. Aj keď záujem žiaka môže byť založený len na dobrom správaní a/alebo emóciách, bez mentálnej potreby pochopiť podstatu aktivity [4], naše pozorovania zameriame hlavne na oblasť kognitívnej participácie.

Učenie sa objavovaním vychádza z myšlienok konštruktivizmu (nás zaujíma najmä kognitívny konštruktivizmus, ktorého autorom je švajčiarsky psychológ J. Piaget). Učiteľ pri takomto vyučovaní vedie žiaka k tomu, aby sám na niečo prišiel, aby objavil pre neho novú súvislosť alebo odhalil, ako veci fungujú. Tým dostáva dostatok priestoru a času na to, aby si sám v mysli skonštruoval nové vedomosti, ktoré hladko zapadnú do toho, čo už vie [5]. Konštruktivizmus tak spúšťa prirodzenú zvedavosť žiaka o okolitý svet [6], žiak aktívne vytvára (konštruuje) svoje vedomosti a zručnosti na základe podnetov z prostredia, manipuláciou s objektmi, hľadaním odpovedí na otázky a spory, vykonávaním experimentov, ako aj na základe svojich vrodených schopností [6] [7]. Konštruktivistické učenie sa žiakov, príkladom ktorého je aj učenie sa objavovaním, kladie dôraz na problémové situácie, ktoré sú zdrojom na objavovanie nových faktov a vzťahov. Realizácia takýchto bádateľských aktivít zahŕňa rôzne činnosti od formulácie problému, cez návrh a realizáciu postupu riešenia, zbieranie údajov pri experimentovaní a ich analýzu a vyhodnocovanie, interpretáciu výsledkov, formulovanie záverov a zdôvodňovanie objavených zistení [8]. V porovnaní s inštruktívnym vyučovaním, založenom na zadaniach a informáciách poskytovaných učiteľom priamo žiakom [6], si pri objavnom prístupe k učeniu žiaci zapamätajú viac pojmov ako aj poznatkov [9]. V procese učenia sa, sú žiaci zapojení viac ako len pri počúvaní a čítaní, rozvíjajú svoje zručnosti, analyzujú, vyhodnocujú a diskutujú medzi sebou o ich riešení. Učenie sa tak predstavuje akýsi vedecký proces aktívneho skúmania sveta [10]. Aj napriek tomu, mnohí kritici veria, že tento spôsob výučby môže viesť k nesprávnym vedomostiam [7]. To, že vysvetľovanie a tvorba konceptov nasleduje až po experimente, je podľa nás naopak obrovským prínosom vzdelávacích stratégii zdôrazňujúcich učenie sa založené na experimentovaní a objavovaní (napr. discovery learning, experience-based learning, project-based learning, inquiry-based learning).

Cieľom príspevku je ukázať, že formou hier, pohybových aktivít a manipuláciou s predmetmi sa dajú objavovať nové informatické koncepty. Prezentované aktivity sú pripravené formou zážitkového učenia sa bez priameho kontaktu s počítačom, len s použitím relatívne jednoduchých pomôcok.

2. Informatika bez počítača

Pre mnohých žiakov je tradičný prístup výučby informatických konceptov nevyhovujúci, musíme ich podporiť vizualizáciou problému a využiť ich fantáziu pri hľadaní riešenia [11]

a poskytnúť im niekedy až pohlcujúci zážitok [12]. Jednou z možností ako angažovať žiakov v praktických aktivitách je použiť prístup označovaný ako Informatika bez počítača (Computer Science Unplugged, CSU). Informatika bez počítača je séria vzdelávacích aktivít navrhnutých a realizovaných formou zážitkového učenia. Formou hier, pohybových aktivít a manipuláciou s predmetmi sa dajú urobiť prvé kroky do počítačovej vedy [13]. Aktivity špecificky zamerané na prezentovanie programátorských konceptov sú označované pojmom programovanie bez počítača (programming unplugged) [14]. CSU aktivity môžu prispieť k motivácii žiakov učiť sa aktívne a zaujímať sa o podstatu vecí okolo seba [15]. Aktivity vyžadujú tímovú prácu, čím podporujú sociálnu interakciu medzi žiakmi a medzi žiakom a učiteľom. [16]. Učiteľ podporuje žiakov v hľadaní vlastných odpovedí, pričom stále existuje priestor na usmerňovanie a spätnú väzbu [13]. Aktivity bez počítača sú často využívané počas informatických letných táborov [14] [17] [18]. Pri praktických aktivitách musí však učiteľ prispôsobiť nielen učebné pomôcky, ale aj celkový spôsob výučby. Pri návrhu a príprave aktivít bez počítača by mali platiť isté pravidlá [19], z ktorých zdôraznime potrebu cenovej dostupnosti použitého materiálu, rodovej neutrality aktivít, ako aj potrebu tímovej práce. Každá aktivita vytvorená na princípoch CSU má jasné a rýchlo vysvetliteľné pravidlá, je atraktívna pre žiaka a motivuje ho k práci smerom ku konkrétnemu cieľu alebo k hľadaniu iného, efektívnejšieho, riešenia.

2.1 Aj Informatiky je možné sa dotknúť!

Na základe vyššie definovaných pravidiel bola navrhnutá a niekoľkokrát realizovaná séria aktivít tvoriaca príbeh s cieľom nájdenia odpovede na otázku: „Ako putujú dáta zobrazené na monitore cez počítač a internet až k tvojmu kamarátovi?“ Všeobecným cieľom je s využitím princípov objavného učenia sa pochopiť koncept kódovania a zobrazovania informácií v počítači, ich prenosu cez internet ako aj ich zabezpečeniu počas prenosu cez internet. Základný cieľ je rozdelený do navzájom prepojených aktivít: (1) Binárny zápis (2) Binárne počítanie (3) Kódovanie znakov (4) Prevod signálu z digitálnej na analógovú formu (5) Zabezpečenie prenosu cez internet. Cieľ každej čiastkovej aktivity je jasný a dosiahnuteľný aj pri súčasných vedomostiach žiakov (predpokladáme pochopenie predbiehajúcich aktivít). Vychádzame teda zo sekvencie zážitkových aktivít, dopĺňaných cielenými otázkami (napr. Čo sme teraz robili? Čo sa stalo? Čo bolo najťažšie?). Praktický experiment je základnou podmienkou na zodpovedanie kladených otázok.

Zážitkové aktivity prebiehali vždy v interiéry školy, v učebni bez počítačov. Výber žiakov bol ponechaný na učiteľoch na príslušných školách. Jedinou podmienkou bolo, aby žiaci nemali predchádzajúce skúsenosti s kódovaním čísel a znakov v binárnej podobe. Keďže každá oslovená škola má rozdielnu štruktúru žiakov, ako aj rozdielne plány pre zaradenie konkrétneho učiva, vznikla situácia, že sme pracovali s rôznymi vekovými skupinami. Na realizáciu bol vyčlenený súvislý blok dvoch až troch vyučovacích hodín , v závislosti od toho, do akej hĺbky chceme žiakom jednotlivé koncepty prezentovať. Overovanie prebiehalo na troch rozdielnych typoch škôl. V januári 2015 v škole pre deti so všeobecným intelektovým nadaním (17 žiakov, 6. – 8. ročník), v júni 2015 na osemročnom gymnáziu (17 žiakov, prima) a na základnej škole všeobecného zamerania (12 žiakov, 8. – 9. ročník). V rámci mimoškolských aktivít prebehlo overenie v letnom tábore ITečk@r 2015. Na realizáciu boli použité relatívne jednoduché doma vyrobené pomôcky (obrázok 1).

Obrázok 1. Učebné pomôcky k zážitkovým aktivitám

Počas realizácií boli sledované reakcie žiakov, ich časté chyby a tímová spolupráca. Každá opakovaná aktivita bola mierne upravená, aby sme čo najlepšie mohli pracovať s chybami žiakov. Po skončení sme neriešili žiadnu formu sumatívneho hodnotenia, ani dotazníkového prieskumu. Tento príspevok sumarizuje skúsenosti získané zo všetkých vyššie uvedených realizácií. Výsledky boli šírené aj medzi učiteľmi, a to formou osobnej diskusie, praktickými ukážkami na Roadshow Moderný učiteľ 2015 a na konferencii Učíme pre život 2015, ako aj písomnou formou v časopise Dobrá škola [20][21]. Téma informatiky bez počítača bola prezentovaná aj webinárovou formou (bit.ly/webinarCSU).

2.2 Aktivita 1 - Binárny zápis

Z vlastných skúseností, ako aj z diskusií s učiteľmi môžeme konštatovať, že prevod čísla z desiatkovej do dvojkovej sústavy sa väčšinou učí pomocou algoritmu delenia dvomi a zapamätávaním si zvyškov po delení. Výhodou tejto schémy je možnosť jej aplikovania na ľubovoľnom desiatkovom čísle, pričom opakovaním stále rovnakého postupu je možné získať správny výsledok aj bez ohľadu na to, či žiak binárnemu zápisu rozumieme alebo nie. Takýto postup je podľa nás vhodný najmä pre žiaka preferujúceho prácu s písaným textom (klasifikácia podľa zmyslových preferencií), a pamäťové učenie bez väčšieho úsilia o hlbšie pochopenie (povrchový učebný štýl) [22]. Našou snahou je predstaviť spôsob vyučovania binárneho zápisu čísla založenom na preskupovaní fyzických objektov, ktorý motivuje žiakov k snahe o porozumenie učiva (hĺbkový učebný štýl) a vyhovuje žiakom s rôznymi zmyslovými preferenciami.

Cieľ aktivity: Porozumieť, že pri prevode čísla do dvojkovej sústavy sa množstvo (počet) objektov nemení, mení sa len forma zápisu tohto množstva.

Tabuľka 1 Karty reprezentujúce rády binárnej sústavy

2^4	2^3	2^2	2^1	2^0
▢▢▢▢ ▢▢▢▢ ▢▢▢▢ ▢▢▢▢	▢▢ ▢▢ ▢▢ ▢▢	▢ ▢ ▢ ▢	▢ ▢	▢

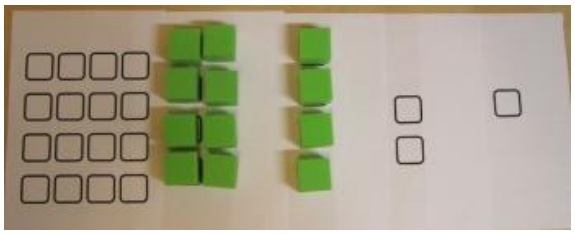
Na realizáciu potrebujeme päť kariet (osvedčil sa nám formát A5). Na každej karte bude vopred vyznačený konečný počet miest pre ukladanie objektov (Tabuľka 1). Každá z kariet reprezentuje mocninu dvojky (... 2^2, 2^1, 2^0). Ďalej potrebujeme objekty vhodné na preskupovanie (v našom prípade boli použité penové kocky).

Priebeh aktivity. Žiakov rozdelíme do menších skupín. V prvom kroku bude mať každá skupina k dispozícii sériu piatich kariet (Tabuľka 1) a rovnaký počet penových kociek (začnime napríklad počtom 12). Necháme žiakov spočítať kocky a zapíšeme

výsledok v desiatkovej sústave. Položíme otázku: „Vedeli by ste tieto kocky rozložiť na karte?“. Pokiaľ nedáme presnejšie inštrukcie, je pravdepodobné, že dostaneme rôzne, väčšinou nesprávne, riešenia (Obrázok 2).

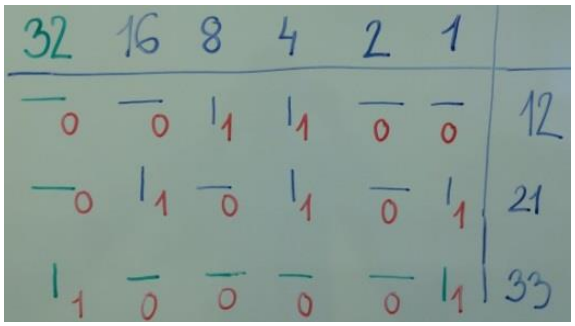
Obrázok 2. Jedno z nesprávnych riešení rozloženia 12-tich kociek

„Ktoré z riešení je však správne?“ Musíme si stanoviť spoločnú podmienku. Dáme pokyn na ukladanie kociek s pravidlom, že každá z kariet je buď plná, alebo prázdna (Obrázok 3). Položíme otázku: „Ktoré karty sme použili?“ Spoločne so žiakmi do tabuľky zaznačíme použité karty pre prvé z riešení (vo forme horizontálnych a vertikálnych čiar, Obrázok 4). „Kto má iné riešenie?“. Ukáže sa, že existuje len jedno riešenie.



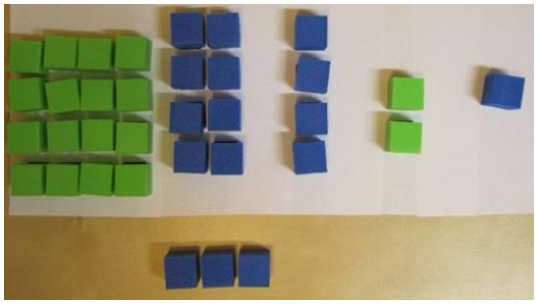
Obrázok 3. Jediné správne riešenia rozloženia 12-tich kociek

Každej skupine žiakov postupne pridávame kocky a činnosť popísanú vyššie opakujeme niekoľkokrát s novým počtom. Každý nový počet bude mať vždy jedinečné riešenie.



Obrázok 4. Zápis manipulácie s reálnymi objektmi.

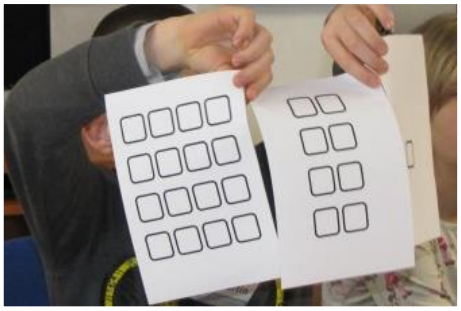
Každej skupine postupne dopĺňame kocky, až sa dostaneme do situácie, že ich je viac ako je počet všetkých bodov na kartách (Obrázok 5.). Po zaplnení kariet nám teda ostanú neuložené kocky. „Ako vyriešime túto situáciu?“ Nechajme žiakov prísť na to, že potrebujeme ďalšiu kartu na ktorej bude 32 bodov, a tiež na to že každá ďalšia karta bude mať dvojnásobný počet bodov v porovnaní s predchádzajúcou.



Obrázok 5. Počet kociek je väčší ako počet vyznačených miest

Opakujme preskupovanie aj s novou kartou a výsledky stále zapisujme do tabuľky (Obrázok 4). Keď už žiaci vedia bez problémov preskupovať ľubovoľný počet kociek, obráťme pozornosť na zapísané výsledky. Vysvetlime si, že náš zápis môžeme trochu upraviť. Položme otázku: „Koľkokrát bola použitá každá z kariet? Odpoveď je buď ani raz (1), alebo jedenkrát (0). Napríklad pre číslo 12 bolo použitých 0 jednotiek, 0 dvojok, 1 štvorka a 1 osmička (Obrázok 4). A to je vlastne zápis čísla v dvojkovej sústave; $(12)_{10} = (1100)_2$. Každú takúto jednotku alebo nulu nazývame bit (zo slova binary digit). Tu môžeme porovnať rády desiatkovej (1, 10, 100, ...) a dvojkovej sústavy (1, 2, 4, 8, 16, ..). Preskupovaním objektov si žiaci postupne sami vytvárajú vlastné predstavy o rádoch binárnej sústavy. Žiakom môžeme tiež ukázať ako sa rády zapisujú cez mocniny.

Pokračovanie aktivity. V ďalšom kroku aktivity ponechajme žiakom vyššie popísané karty, ale nepoužívajme už kocky. Položme otázku: „Viete nájsť postup ako zapísať čísla v binárnom tvare spamäti, bez manipulácie s kockami?“ Táto činnosť je o niečo zložitejšia, lebo žiak už nemanipuluje s kockami, ale priamo označuje karty, ktoré by použil a ktoré nie. Je vhodné upozorniť na to, aby karty mali žiaci celý čas položené pred sebou, a to stále v rovnakom poradí. Tie karty, ktoré nebudú použité stačí otočiť. Žiaci by mali intuitívne prichádzať na to, že na prevod čísel stačí použiť len operáciu sčítania.



Obrázok 6. Precvičovanie binárneho zápisu s kockami a bez nich.

Vyhodnotenie: Po ukončení prvej aktivity sú všetci žiaci schopní transformovať jednoduché čísla do binárneho zápisu spamäti. Mocniny dvojky sú rýchlo rozpoznané a plynulo používané pri kognitívnych operáciách. *Poznámka:* Učiteľ môže overiť, či žiak splnil vzdelávací cieľ zameraním otázok na iné číselné sústavy: “Ak by sme prevádzali čísla z desiatkovej sústavy do trojkovej, koľko kociek by sme ukladali na jednotlivé kartičky?“. Manipuláciou s objektmi by priamo experimentovali a vytvárali potrebné n-tice. Ak to situácia dovolí, žiaci môžu zapísať počet

objektov na karte aj vo forme mocniny n^x. Žiaci tak môžu úplne prirodzeným spôsobom pracovať aj s mocninami.

2.3 Binárne počítanie

Predpokladajme, že žiaci sa už naučili základný princíp prevodu založenom len na sčitovaní rádov dvojkovej sústavy.

Cieľ: Precvičovanie binárneho zápisu a hľadanie vzorov.

Na realizáciu budeme potrebovať šesť kariet (osvedčil sa formát A4), kde každá z kariet je z jednej strany prázdna a z druhej strany má zobrazenú skupinu bodov (32, 16, 8, 4, 2, 1, podobne ako v predchádzajúcej aktivite, Obrázok 7). Ideálne je mať karty kvôli manipulácii vystužené, napr. laminovaním. Vytvoríme rad šiestich žiakov. Každý z nich bude držať jednu z kariet a symbolizovať konkrétny bit v 6-bitovom binárnom zápise, ich poradie je preto dôležité. Podľa toho, ako je karta otočená, reprezentuje hodnotu 1 alebo 0. Aby sme do aktivity zapojili čo najviac žiakov, môžeme vytvoriť dve proti sebe stojace skupiny, ktoré pri zobrazovaní súťažia (chlapci vs. dievčatá). Zostávajúci žiaci môžu byť zadávatelia, merači času a hlavne kontrolóri správneho riešenia.

Nechajme žiakov zobrazovať rôzne desiatkové čísla (7, 15, 18 .. 64), na obrázku 7 je ukážka toho, ako by bolo zobrazené číslo 7 (v binárnom 6 bitovom zápise ako 000111). A naopak experimentujme s prevodom binárneho tvaru do desiatkového. „Zobrazte číslo 10100. Aké to je číslo v desiatkovej sústave?“

Obrázok 7. Číslo $(7)_{10}$ zobrazené v binárnom tvare ako $(111)_2$.

V porovnaní s aktivitou (1) Binárny zápis, kde každý žiak mal pred sebou všetky karty, tu má každý žiak kartu len jednu. A tak musí v skupine spolupracovať s ostatnými. Prípadne si každý žiak sám vypočíta, či sa príslušný bit (číslo, karta) v zápise nachádza alebo nie. Na záver prevedme číslo 64. Nechajme žiakov sformulovať odpoveď, že najväčšie 6-bitové číslo je 63. „A aké je naopak najmenšie 6-bitové číslo?“. S prihliadnutím na istú mieru nepresnosti definujme 0 ako najmenšie číslo (zápis záporných čísel nie je cieľom aktivity).

Pokračovanie aktivity. Položme otázku: „Myslíte si, že sa dajú zobraziť všetky čísla od 0 po 32?„ Žiaci zobrazujú všetky čísla postupne za sebou. Ich úlohou je sledovať správanie sa jednotlivých bitov. Po skončení sa snažíme hľadať akési vzorce správania sa. Položme otázku: „Ktorý z bitov sa narobil najviac, a ktorý najmenej?“ Samozrejme najviac sa narobil jednotkový bit, ktorý sa preklápal pri každom zvýšení čísla (00**0**, 00**1**, 01**0**, 01**1**, 100 ...). „Existuje nejaké pravidlo pre párne čísla?“ Je dobré, keď prídu na to, že párne čísla majú na konci nulu a nepárne jednotku. Takto sa dá zovšeobecniť aj správanie sa ostatných bitov, napríklad dvojkový bit sa preklápal s periódou 2 (0**0**0, 0**0**1, 0**1**0, 0**1**1, 1**0**0 ...).

Vyhodnotenie: Zo začiatku môže byť preklápanie chaotické. Žiaci však postupne prídu na to, že musia v prvom rade zistiť, ktorý je najväčší jednotkový bit v zápise a k nemu postupne pripočítať ďalšie menšie jednotkové bity (napr. pre 21 je to 16 + 4 + 1, čo je v binárnom zápise 10101). Ťažšie je nájdenie vzoru, kde sa vyšší bit preklopí len vtedy, keď sa nižší bit preklopil zo stavu 1 do stavu 0 (000, 00**1**, 01**0**, 0**11**, 1**00**). K odhaleniu takého postupu by stačilo, aby každý žiak sledoval len žiaka po ľavej strane a svoju kartu otáčal len v prípade, že u suseda nastala zmena z 1 na 0. Táto aktivita vyžaduje viac kognitívnych činností ako úvodná aktivita s manipuláciou s kockami. *Poznámka:* Učiteľ môže overiť, či žiak splnil vzdelávací cieľ kladením otázok zisťujúcich predchodcu a nasledovníka pre nejaké konkrétne binárne číslo.

2.4 Kódovanie znakov

Predpokladajme, že žiaci už vedia binárny zápis a rozumejú dôležitosti poradia bitov v zápise.

Cieľ: Porozumieť reprezentácii znakov v počítači a štandardizovanému kódovaniu.

Žiakov rozdelíme do párneho počtu skupín (ideálne štyri menšie skupiny). Položme otázku: „Ak chcete poslať e-mail, ako by ste zakódovali jednotlivé znaky v správe?“. Žiaci vedia, že počítače pracujú len s binárnym zápisom, pričom prevod čísla už zažili. Zvyčajne automaticky navrhnú priradiť jednotlivým znakom čísla. „Aké čísla by ste teda priradili jednotlivým písmenám abecedy?“ Žiaci s veľkou pravdepodobnosťou navrhnú urobiť zámenu podľa obrázka 8.

A	B	C	D	E	F	G	H	I	J	K	L	M
1	2	3	4	5	6	7	8	9	10	11	12	13
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25	26

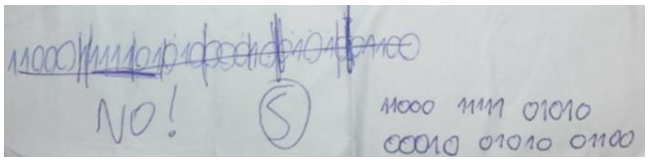
Obrázok 8 Kódovacia tabuľka pre aktivitu Kódovanie znakov

Položme otázku: „A čo keby som ja ako prijímateľ mal iné, vlastné kódovanie, napr. A by bolo 26 a Z by bolo 1? Ako zabezpečiť to, aby obe strany rozumeli, ktoré číslo sa práve prenáša?“ Po diskusii si žiaci musia stanoviť spoločné pravidlá kódovania pre všetky zúčastnené strany komunikácie. Objavia koncept štandardizácie. Aby všetky skupiny mali naozaj rovnaké a jednoduché pravidlá kódovania, poskytne učiteľ každej z nich rovnakú kódovaciu tabuľku (obrázok 8). Ak sa však skupina dohodne na vlastných pravidlách, ponechajme ich. Každá zo skupín tiež dostane správu, ktorú má s využitím kódovacej tabuľky prepísať do binárneho kódu (obrázok 9).

Obrázok 9. Kódovanie - prepis správy do binárneho kódu

Po zakódovaní všetkých správ si ich skupiny navzájom vymenia a snažia sa o odhalenie originálnej správy. Vo väčšine prípadov sa však žiaci nedopracujú k pôvodnej správe. „Ak sme použili rovnaké kódovacie tabuľky a rovnaký binárny zápis, ako je možné, že sme sa pri dekódovaní nedopracovali k pôvodnej správe?“ Žiaci postupne zistia, že jednotlivé skupiny použili na zakódovanie jedného znaku rozdielny počet bitov (obrázok 10) alebo pri prepise vynechali nuly zo začiatku správy. Objavia, že

spoločná dohoda musí obsahovať okrem kódovacej tabuľky aj informáciu o počte použitých bitov na zápis čísla.



Obrázok 10. Pokus o dekódovanie pôvodnej správy.

Vyhodnotenie: Aktivita sa zo začiatku javila ako veľmi jednoduchá. Žiaci si v skupine rozdelili úlohy prevádzačov, kontrolórov a zapisovateľov. Ukázalo sa, že pri väčšom počte žiakov v skupine, sa niektorí aktívne nezapájajú. Preto je vhodnejšie súčasne kódovať viacero správ, čím vzniknú menej početné skupiny.

2.5 Prenos cez internet

Predpokladajme, že žiaci už vedia dostatočne rýchlo pracovať s jednoduchým prevodom čísel. Nepotrebujú už manipulovať s objektmi ani pomocnými karty s číslami. Tie nahradíme kartami, kde každá má na jednej strane jednotku a z druhej strany nulu (symbolizujúc stavy jedného bitu, obrázok 11).

Obrázok 11. Každá z kariet reprezentuje jeden bit.

Cieľ: Pochopenie pojmu digitálno-analógový prevod a naopak.

Textovú správu už máme zakódovanú v binárnom tvare. Ako ju však poslať cez fyzické zariadenia pripojené k internetu? Žiaci vedia, že počítač pracuje s jednotkami a nulami. Vo fyzických kábloch sa však šíria inou formou, konkrétne ako napätie, ktoré si môžeme zobraziť formou vlny.

Obrázok 12. Napäťová vlna vizualizovaná pomocou krajčírskej gumy

V tejto aktivite je zúčastnených až 16 osôb. Prvý žiak symbolizuje znak (napr. A), ktorý má svoje poradie v kódovacej tabuľke. Ďalších 7 žiakov pomocou kariet prevedie číslo do binárneho tvaru. Žiak v strede radu symbolizuje prevodník a posledných sedem žiakov predstavuje vlnu napätia (Obrázok 12). Žiakom dáme úlohu. „Vedeli by ste navrhnúť spôsob zobrazenia vlny?“ Štandardným návrhom bude, že žiak reprezentujúci jednotku bude stáť, a ten, čo bude reprezentovať nulu sa dá do drepu. My sme však navrhli využiť krajčírsku gumu (Obrázok 12), ktorú žiaci natiahli do výšky (jednotkový bit) alebo ju nechali dolu (nulový bit). Po zmene vstupných binárnych dát, sa musela analógová vlna adaptovať.

Každé dieťa má inú výšku, čím sa dá zaujímavo vysvetliť, že aj pri reálnej vlne nie je jasný prechod z nuly na jednotku a naopak. Pracujeme so spojitým nie diskrétnym signálom. Jednotka vo vstupe môže mať rozdielnu reprezentáciu na výstupe. Pozorovať môžeme tiež nábehové hrany signálu.

Vyhodnotenie: Aktivita je veľmi dynamická. Žiaci vytvárajúci binárny zápis konkrétneho čísla môžu pre jednoduchosť zobraziť akúkoľvek kombináciu núl a jednotiek. Proces transformácie je možné vymeniť, vstupom bude analógový signál a výstupom bude binárny zápis.

2.6 Zabezpečený prenos dát

Posledná aktivita je zameraná na tému bezpečného prenosu dát cez internet.

Cieľ: Vysvetliť zabezpečenie dát pomocou dvoch šifrovacích kľúčov (asymetrické šifrovanie).

Navoďme diskusiu. „Čo treba urobiť, aby informácia, ktorú posielame skutočne prišla celá, a neupravená?“ Zahrajme sa na zasielanie správ. Použijeme drevené krabičky s poklopom (Obrázok 13). Vložme do nich správu (v našom prípade sme použili cukríky) a označme adresáta, namiesto IP adresy sme použili krstné meno žiaka. Krabičky pošleme adresátovi s dôležitou inštrukciou pre žiakov, ktorí predstavovali uzly v počítačovej sieti. Dajme pokyn: „Uvažujte, čo sa so správou môže počas prenosu stať.“

Obrázok 13. Správa s adresou uzamknutá pomocou dvoch kľúčov.

Správu posielame niekoľko krát rôznym adresátom. Žiaci postupne prídu na to, že správu môže niekto vymeniť, pripojiť k nej niečo, čo tam pôvodne nebolo, alebo sa dokonca narušiteľ môže vydávať za adresáta a nechať si celú správu. „Ako by sme mohli správu zabezpečiť proti zmene obsahu?“ Keďže skrinky majú petlicu, žiaci navrhnú použiť zámok. A tak opäť vložíme do krabičiek správu, uzamkneme ich a pošleme adresátovi. O takto uzamknutú správu nemajú už žiaci záujem a priamo ju presunú k adresátovi, ktorý sa však tiež nemôže k správe dostať, je uzamknutá. Žiaci väčšinou navrhnú, aby mu bol poslaný aj kľúčik. Vysvetlíme však, prečo je nevhodné posielať kľúč aj uzamknutú správu. Adresátovi miesto toho poskytneme vlastný zámok s kľúčom a požiadame ho, aby uzamkol správu a poslal ju pôvodnému odosielateľovi. Správa je tak uzamknutá dvoma zámkami (obrázok 13), pričom adresát ako aj odosielateľ majú svoje súkromné kľúče bezpečne v rukách. Odosielateľ odstráni

svoj zámok, aby správu následne poslal opäť do rúk adresáta, ktorý odstránením svojho zámku otvorí správu.

Vyhodnotenie: Aktivita je vizualizáciou známeho problému výmeny kľúčov v asymetrickej kryptografii. Žiakom môžeme vysvetliť, že miesto zámkov sa v praxi používajú zložité matematické funkcie.

3. Záver

Z realizovaných zážitkových aktivít môžeme vyvodiť určité závery. Zaznamenali sme, že väčšina žiakov sa aktívne a dobrovoľne zapájala do vyučovania. Počas realizácie sme sa spýtali na niekoľko otázok, ktoré vo väčšine prípadov boli zodpovedané správne, niekedy prekvapivo rýchlo. Pri prevode čísel z dvojkovej do desiatkovej sústavy (a naopak) sme sa namiesto delenia so zvyškom rozhodli použiť súčet a rozdiel, čo sa ukázalo ako dobrá voľba. Dokonca aj najmladší účastníci vedeli zapísať jednoduché čísla v binárnej forme bez použitia kalkulačky a to aj napriek tomu, že téma prevodu čísel nie je na základných školách preberaná, alebo je s ňou žiak oboznámený až v posledných ročníkoch ZŠ.

Pôvodným zámerom bolo vytvoriť len akúsi formu popularizácie informatiky. Po niekoľkých realizáciách a najmä po získaní spätnej väzby od učiteľov môžeme skonštatovať, že popísané postupy sa môžu stať súčasťou formálnej výučby.

Škola nie je len o cieľoch, o kontrole ich plnenia a o následnom sumatívnom hodnotení. Učiteľ by mal v čo najväčšej miere prepájať predmety, pracovať s chybou a využívať formatívne hodnotenie. Učiteľ by mal vzdelávať myseľ, rozvíjať kreatívnu predstavivosť a umožniť žiakom odchádzať z hodiny so slovami „To bola dobrá hodina, mohli by sme niečo také robiť aj inokedy.“ Učenie sa objavovaním cez zážitkové aktivity bez použitia počítača nám to všetko umožňuje. Ukázali sme, že vizualizácia informatických pojmov cez zážitkové učenie je spôsob, ktorý môže zmeniť pohľad žiakov aj na nie príliš populárne časti informatiky.

BIBLIOGRAFICKÉ ODKAZY

[1] FREDRICKS, J. A. *Eight Myths of Student Disengagement: Creating Classrooms of Deep Learning.* Los Angeles: Corwin. F.2014.

[2] CHEN, J. A., PAJARES, F. Implicit theories of ability of Grade 6 science students: Relation to epistemological beliefs and academic motivation and achievement in science. In Contemporary Educational Psychology, 2010, Vol.35(1).

[3] ZYNGIER, D. (Re)conceptualising student engagement: Doing education not doing time. Teaching and Teacher Education, 24. 2008.

[4] JAMES, N. P. Golden Rules for Engaging Students in Learning Activities. [online] [cit. 2017-01-30] . Dostupné z: https://www.edutopia.org/blog/golden-rules-for-engaging-students-nicolas-pino-james

[5] KABÁTOVÁ M., KALAŠ I., MIKOLAJOVÁ M., PEKÁROVÁ J. Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika: Východiská a inšpirácie. 40 s. 2009

[6] Educational Broadcasting Corporation. Workshop: *Constructivism as a Paradigm for Teaching and Learning. What is constructivism?* [online] [cit. 2017-03-03] . Dostupné z: http://www.thirteen.org/edonline/concept2class/constructivism/index.html

[7] Learning Theories. Discovery learning. [online] [cit. 2017-01-30]. Dostupné z: https://www.learning-theories.com/discovery-learning-bruner.html

[8] LUKÁČ, S., ŠNAJDER, Ľ, GUNIŠ J.,JEŠKOVÁ. 2009. Bádateľsky orientované vyučovanie matematiky a informatiky na stredných školách [online]. Košice: Univerzita Pavla Jozefa Šafárika v Košiciach, 2016, 219s. [cit. 2016-12-20]. ISBN 978 – 80 – 8152 – 471 - 4. Dostupné z: http://bit.ly/2m7vEwn

[9] BRUNER, J. S. The process of education. Harvard University Press. 2009.

[10] LLEWELLYN, Douglas. 2001. *Inquire Within: Implementing Inquiry-Based Science Standards*. Thousand Oaks, CA : Corwin Press, Inc. 2001. 192 pp.

[11] EDWARDS S. H.. Using software testing to move students from trial-and-error to reflection-in-action. In *ACM SIGCSE Bulletin*, Vol. 36, No. 1, 2004.

[12] ZAHARIJA G., MLADENOVIĆ S., BOLJAT I. Introducing basic Programming Concepts to Elementary School Children. In *Procedia - Social and Behavioral Sciences*, Vol. 106, 2013.

[13] ARMONI, M. On teaching abstraction in CS to novices. In *Journal of Computers in Mathematics and Science Teaching* 32.3 (2013), 265--284.

[14] ALAMER R. A., Al-DOWEESH W. A., AL-KHALIFA H. S., AL-RAZGAN M. S. Programming Unplugged: Bridging CS Unplugged Activities Gap for Learning Key Programming Concepts. *Fifth International Conference on e-Learning (econf)*, Manama, 2015, pp. 97-103.

[15] CÁPAY, M. Engaging Games with the Computer Science Underlying Concepts. In *Interactive Collaborative Learning (ICL 2015)*, (pp. 966-970).

[16] BELL T., WITTEN I., FELLOWS M. 2015 Computer Science without a computer. [online] [cit. 2017-01-30] Dostupné z: http://csunplugged.org/

[17] BARBARA, E., MCKLIN, T. Effective and sustainable computing summer camps. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* ACM. 2012.

[18] TIMOTHY, U., MANLEY, E.D. Generating interest in computer science through middle-school android summer camps. *Journal of Computing Sciences in Colleges*. 2013,285, pp. 211-217.

[19] BELL, T., ALEXANDER, J., FREEMAN, I., & GRIMLEY, M. Computer science without computers: new outreach methods from old tricks. In *Proceedings of the 21st Annual Conference of the National Advisory Committee on Computing Qualifications*.

[20] CÁPAY, M. Aj informatiky sa možno dotknúť In *Dobrá škol a : inšpirácia pre učiteľov, ktorí chcú byť lepší.* ISSN 1338-0338, Roč. 7, č. 5 (2016), s. 30.

[21] CÁPAY, M. Informatika bez počítača. In: *Dobrá škola : časopis o vzdelávaní v 21. Storočí.* ISSN 1338-0338, Roč. 7, č. 9 (2016), s. 32.

[22] Rovňanová, L. Učebné štýly žiakov a vyučovacie štýly učiteľov. 2015, s. 70.

Výuka programování pomocí robota Ozobot
Teaching programming using robot Ozobot

Rostislav Fojtík
Katedra informatiky a počítačů,
Ostravská univerzita
30. dubna 22
701 00 Ostrava
Česká republika
rostislav.fojtik@osu.cz

ABSTRACT
This paper aims to show the possibilities of teaching programming using robot Ozobot. The survey shows that teaching programming is seen by many science teachers neglected area. Teachers often consider teaching programming unnecessary and schools would not be them into thematic plans ever marshaled. This situation is often attributed to the fact that many teachers do not know the appropriate procedures and methodology for teaching the basics of programming and algorithms. Teachers at some schools have already started to use appropriate applications such as Scratch code.org and others. For teaching, it is advisable to use only software tools but also the hardware. Using case studies have verified the teaching of programming a robot Ozobot. It turns out that Ozobot is not only suitable for small children, but also for older pupils.

Keywords
Algorithms. Ozobot. Programming. Teaching.

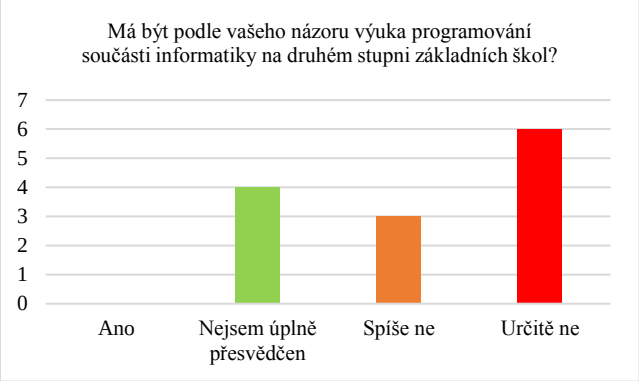
ABSTRAKT
Cílem příspěvku je ukázat možnosti výuky programování pomocí robota Ozobot. Jak ukazuje provedený průzkum, výuka programování je mnohými učitele informatiky opomíjenou oblastí. Často ji považují za zbytečnou a na základních školách by podle nich neměla být do tematických plánů vůbec zařazována. Tato situace je často zavíněna tím, že mnozí učitele neznají vhodné postupy a metodiku pro výuku základů programování a algoritmizace. Na některých školách učitelé již začali využívat vhodných aplikací jako Scratch, code.org a podobně. Pro výuku je však vhodné využívat nejen softwarové prostředky, ale rovněž ty hardwarové. Pomocí případové studie byly ověřeny možnosti výuky programování s robotem Ozobot. Ukazuje se, že Ozobot je vhodný nejen pro nejmenší děti, ale i starší žáky.

Klíčová slova
Algoritmizace, Ozobot, programování, výuka.

1. ÚVOD
Výuka programování na základních a středních školách v České republice je často opomíjená. V předmětech zabývající se výukou informatiky se obvykle preferuje uživatelský přístup a výuka ovládání konkrétních programů. Na mnohých školách se přesto objevuje snaha více se věnovat výuce programování. Tato snaha má podporu Ministerstva školství, mládeže a tělovýchovy vyjádřenou ve Strategii digitálního vzdělávání do roku 2020 [1].

2. VÝUKA PROGRAMOVÁNÍ NA ZŠ
Většímu rozšíření výuky programování na základních školách se často brání samotní učitele. Jedním z důvodů je, že mnozí programování považují za složité a sami s ním měli během svého studia problémy. Druhým důvodem je, že učitele mají pocit, že jimi vedena výuka programování by měla využívat stejné nástroje a metody, s jakými se sami setkali při svém studiu. Důležitým úkolem pro rozvoj výuky programování na základních a středních školách je učitelům ukázat nové přístupy v oblasti programování, a hlavně seznámit pedagogy s moderními metodickými postupy a pomůckami.

Mezi učiteli – studenty prvního ročníku rozšiřujícího studia informatiky pro základní a střední školy byl proveden jednoduchý průzkum. Pomocí dotazníku byli učitele dotazování na jejich postoj k výuce programování.



Obrázek 1. Má být na ZŠ vyučováno programování?

Většina ze 13 respondentů je spíše proti výuce programování na základní škole. Všichni respondenti absolvovali jeden semestr programování. Konkrétně se jedná o předmět Základy programování, ve kterém se využívá jazyk Java. Vzhledem k tomu, že jen 31 % respondentů má původní aprobaci předměty z oblasti technické nebo přírodovědné, je pochopitelné, že respondenti často považují programování za složité. Respondenti ještě neabsolvovali předmět Didaktika informatiky a mnozí mají představu, že programování lze vyučovat jen podobně jako na vysoké škole pomocí klasických programovacích a skriptovacích jazyků.

Na dotaz, zda by se programování mělo vyučovat na gymnáziu je spíše proti pouze 23 % respondentů, 46 % respondentů odpovědělo ano a 38 % respondentů je spíše pro. Naprostá většina, konkrétně 77 % respondentů, nezná žádné vhodné nástroje pro výuku programování na základních školách. Z výsledku průzkumu vyplývá, že je velmi důležité seznámit vyučující s vhodnými nástroji a metodami výuky programování.

3. ROBOTI VE VÝUCE PROGRAMOVÁNÍ

V současné době mají vyučující k dispozici nejen softwarové nástroje jako Robot Karel, Logo, Scratch, Baltík nebo code.org, ale mohou rovněž využívat hardwarové pomůcky, například robotické stavebnice, drony a výukové roboty. Ve světě se ve výuce s úspěchem používají pomůcky jako jsou například Lego roboti, Sphero, Ollie nebo Ozobot [2, 3].

3.1 Robot Ozobot

Jednou z možností, jak dětem přiblížit výuku algoritmizace a základů programování je robot Ozobot [4]. Jedná se vlastně o malou interaktivní hračku, která pomocí svých senzorů rozpoznává různě barevné čáry. Přestože Ozobot je malý robůtek o hmotnosti 17 g nabízí uživatelům mnoho možností. První z nich je využití příkazů, které jsou tvořeny kombinací barevných čar (color code language) [5, 6]. Využívá se barva černá, zelená, červená a modrá. Stačí jen papír a lihové fixy a děti mohou vytvářet čáry, po kterých robot jezdí. Druhou možností, jak ovládat robota, je využití webových stránek http://ozoblockly.com/. Stránky nabízí editor příkazů, který se trochu podobá například nástroji Scratch. Příkazy jsou rozděleny do pěti skupin podle obtížnosti. Uživatele mohou jednoduše svůj program sestavit a z monitoru nahrát pomocí světelných senzorů přímo do robota [7].

Obrázek 2. Ukázka příkazů pro robota Ozobot [10]

Mezi základní výhody Ozobota patří:

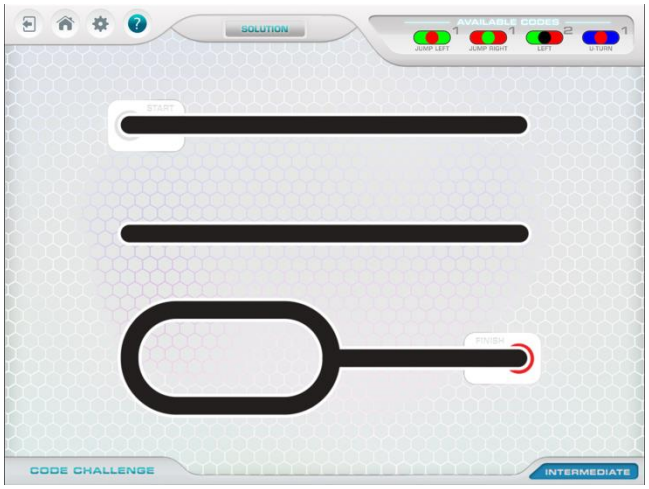
- Jednoduché ovládání,
- názornost,
- podpora motorických schopností dětí,
- přijatelná cena za robota (Ozobot Bit 2.0),
- využitelnost pro výuku u malých i starších žáků,
- množství připravených návodů a příkladů pro výuku.

Mezi nevýhody Ozobota patří:

- Omezené možnosti robota, kterými jsou pouze pohyb a změna barvy světla,
- dlouhé načítání vytvořených programů pomocí světelných senzorů,
- občasné problémy s rozpoznáním barev v příkazech nakreslených fixy,
- slabší motorky ovládající pohyb robota,
- návody, příklady a aplikace jsou pouze v anglickém jazyku,
- vyšší cena robota u verze 3.0.

3.2 Mobilní aplikace

Pro práci s Ozobotem je možné využít aplikace v mobilních zařízeních. První aplikace se nazývá Ozobot a slouží ke kreslení čar a příkazů. Ideální je aplikaci využívat na tabletech s větší velikosti displeje. Jako nejvhodnější se ukázal iPad Pro s displejem 12“. Na rozdíl od kreslení pomocí fixů odpadají problémy s rozpoznáváním barev. Během experimentu bylo zjištěno, že kreslení čar na tabletu pomocí prstů dětem nevyhovuje. Kreslení bylo méně přesné. Jako vhodnější se ukázalo použití stylusu. Velkou výhodou bylo využívaní předem připravených barevných příkazů. Děti pouze vybraly příkaz a umístili jej do příslušného místa. Sekvence barev lze natáčet podle směru čar.



Obrázek 3. Aplikace Ozobot pro tablet iPad

Během sledování dětí ve výuce byly zjištěny následující výhody aplikace Ozobot:

- Odpadají občasné problémy s rozpoznáváním barev vytvořenými pomocí fixů,
- připravené čáry a příkazy jsou robotem vždy správně rozpoznány,
- není nutné tisknout úkoly,
- chyby v nákresech se dají opravit.

Nevýhody aplikace Ozobot:

- Nutnost vlastnit pro každého žáka tablet s dostatečně velkým displejem,
- žáci neprocvičují motoriku stejně jako při práci s fixy,
- úkoly mohou vzhledem k velikosti displeje obsahovat jen omezený počet a rozměr čar.

Druhou mobilní aplikací pro ovládání Ozobota je program OzoGroove. Jedná se o aplikaci, která naučí robota „tančit“. Uživatele přidávají na časovou osu příkazy pro robota a společně se zvukovým doprovodem robot po načtení vytvořeného programu

tančí – pohybuje se podle nastavených parametrů a barevně bliká. Výhodou aplikace je zábavná forma práce s robotem, která se dětem velmi líbila a může rovněž sloužit v úvodu výuku jako motivační nástroj. Aplikace lze s úspěchem používat nejen na větších tabletech, ale rovněž na chytrých mobilních telefonech.

Obrázek 4. Aplikace OzoGroove pro iPhone

Obě aplikace jsou zdarma dostupné pro tablety s operačními systémy Android nebo iOS.

4. VÝUKA S OZOBOTEM

4.1 Sledované skupiny žáků

Pro ověření využití možností robota Ozobot pro výuku programování byla provedena případová studie. V jejím rámci se různé věkové skupiny žáků a studentů učily pracovat s Ozobotem 2.0 Bit. Děti vytvářely své kódy, jak malováním, tak rovněž skládáním příkazů na stránkách ozoblockly.com.

První sledovanou skupinou byla skupina šesti žáků základní školy ve věku 8 až 10 let. Dvě z nich byly dívky a čtyři chlapci. Uskutečnila se čtyři setkání v délce 1,5 až 2 hodiny. Tři setkání byla věnována kreslení příkazů, při čtvrtém setkání žáci sestavovali jednoduché programy na stránkách http://ozoblockly.com.

Druhou skupinu tvořilo sedm žáků ve věku 11 až 13 let. Z toho byly tři dívky a čtyři chlapci. Výuce programování byly věnovány čtyři lekce po 1,5 až 2 hodinách. První dvě setkání se děti učily ovládat Ozobota pomocí kreslených příkazů. Další dvě lekce žáci vytvářeli programy pomocí stránek http://ozoblockly.com.

Třetí skupinou byli vysokoškolští studenti učitelství informatiky. Stejně jako v minulých letech byli studenti na začátku kurzu Didaktika informatiky dotazování, zda si myslí, že by se žáci na základní škole měli učit programování. Všech šest studentů shodně zastávalo názor, že výuka programování na základní školu nepatří. Programování by připouštěli pouze v zájmovém kroužku. Studenti často přiznávají, že měli obtíže při absolvování předmětů zaměřených na programování, a proto by se raději ve své budoucí pedagogické praxi výuce programování vyhnuli. Svůj postoj k výuce programování změnili teprve po seznámení s nástroji Karel, Logo a Scratch. Nejvíce se jim však líbilo pracovat s robotem Ozobot.

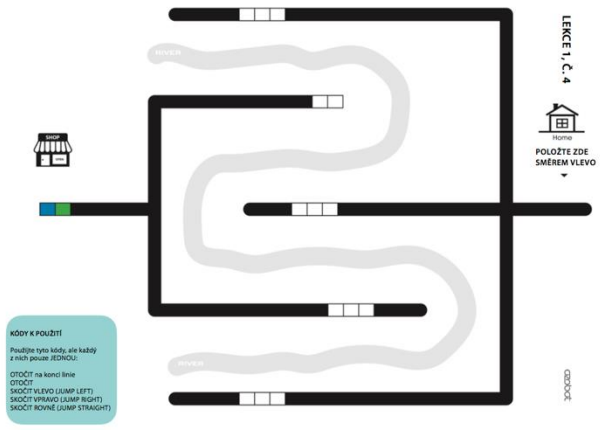
Čtvrtou sledovanou skupinou byla skupina dvanácti chlapců ve věku 11 až 17 let. Zatímco v prvních třech skupinách byli účastníci se zájmem o informatiku a programování, v poslední se žádný z chlapců se ve svém volném čase nezajímal o programování a robotiku. Chlapci zkoušeli ovládat robota pomocí kreslení barevných čar a následně řešili předem připravené úkoly.

4.2 Výsledky pozorování výuky

Cílem případové studie bylo zjistit, jak budou děti na výuku programování pomocí Ozobota reagovat a jaké bude jejich zaujetí při vytváření programů. Při sledování žáků jsme se rovněž zaměřovali na to, jak dlouho jim bude trvat zvládnout ovládání robota a vytváření programů.

Ozobot na rozdíl od Lego robotů nabízí mnohem menší možnosti. Zatímco Lego roboti jsou schopni provádět různorodé operace a je možné je modulárně sestavovat, pak Ozobot se ve své podstatě pouze umí pohybovat podle předepsaného programu a svítit v rozsahu předdefinovaných barev. Jak se během výuky ukázalo, tato omezení mohou být na druhé straně určitou výhodou. Pro některé děti se práce s Lego roboty může zdát složitá a komplikovaná. Naproti tomu ovládání Ozobota všichni účastníci pochopili velmi rychle. V rámci všech experimentů nebylo potřeba dlouze vysvětlovat ovládání a vytváření programů. Všichni účastníci byli schopni během několika málo minut pracovat zcela samostatně.

První příklady, které účastníci obvykle řešili, byly zaměřeny na seznámení s robotem a jeho ovládáním. Děti kreslily čáry a nechaly robota se podle nich pohybovat. Pak dětem byly ukázány příkazy pomocí barevných čar. Děti si opět jen zkoušely, jak robot na příkaz reaguje. Následně byly dětem předloženy připravené úkoly, ve kterých bylo potřeba doplnit pomocí barevných fixů vhodné příkazy. Byly využity základní příklady, které jsou dostupné na stránkách http://portal.ozobot.com/lessons/type/lesson. Pro začínajícího učitele je výhodné tyto sbírky příkladů využít, neboť jsou názorně metodicky popsané a nabízí úkoly vhodně setříděné od těch jednodušších až po složité. Cílem úkolů požitých ve studii bylo doplnit příkazy do nákresu tak, aby Ozobot podle čar překonal nastražené překážky, vybral vhodnou cestu a ze stanoveného startu dojel do cíle. Na obrázku číslo 5 je příklad zadání, ve kterém robot musí ze svého domu dojet do obchodu. Zadání byla přeložena z anglického do českého jazyka. Starší účastníci studie si podobné příklady vymýšleli sami.



Obrázek 5. Příklad úkolu [10]

U všech sledovaných skupin se ukázalo, že je vždy vhodné začít s Ozobotem tvořit pomocí kreslení čar. U menších dětí tato fáze přispívala nejen k lepšímu pochopení práce s robotem, ale došlo i ke zlepšení motorických schopnosti dětí. Ukázalo se, že mnohé

děti mají s kreslením čar problémy. Mezi nejčastější chyby dětí při kreslení patřily:

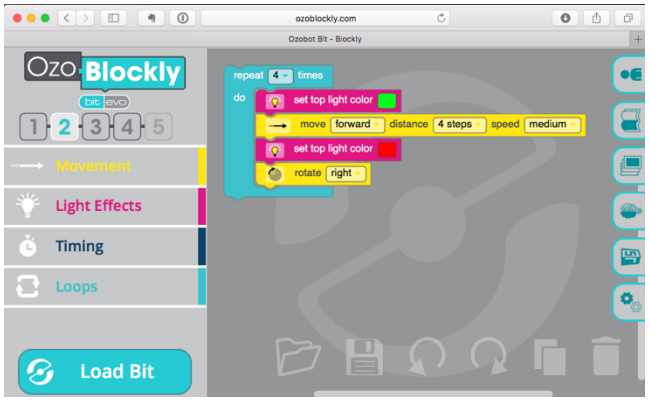
- Čáry svou šířkou neodpovídaly požadavkům,
- přechody mezi barvami nebyly dotažené, mezi barvami byla bílá místa,
- barvy se překrývaly přes sebe,
- barevné čáry v příkazech byly příliš dlouhé.

U některých dětí trvalo více než hodinu, než se naučily správně kreslit čáry a jednotlivé příkazy. Děti neměly problémy s pochopením, které příkazy v dané chvíli použít, ale nakreslit je tak, aby je robot rozpoznal. Jistou roli ve špatné interpretaci příkazů robotem hrály rovněž fixy. Hlavně zelenou barvu Ozobot občas nerozpoznal správně. Situace se zlepšila po chvíli, kdy barva na papíře trochu vyschla. Firma vyrábějící Ozoboty proto raději doporučuje jejich speciální fixy, se kterými má mít robot menší problémy s rozpoznáváním barev. Pro kreslení čar se ukázalo nejvhodnější využívat tzv. tlustých fixů s plochou koncovkou šířky kolem 4 mm. Fixy s tenkou a kulatou špičkou nejsou vhodné. Problémy s kreslením neměly děti při využití mobilní aplikace Ozobot na tabletech.

Podle provedených pozorování je možné konstatovat, že aplikace Ozobot na tabletu je vhodnější využívat pro předem připravené úkoly. Děti pouze doplňují požadované příkazy. Případné chyby lze jednoduše opravit a robot nemá problémy s rozpoznáváním barev. Jednotlivá zadání není potřeba tisknout. Během výuky se ukázalo, že zadání není možné kopírovat, neboť mnohá kopírovací zařízení snižují kvalitu černé barvy a robot ji pak nerozpozná. Děti pak musely čáry opětovně obtahovat černým fixem. Na druhé straně kreslení čar a barevných příkazů pomocí fixů podporuje větší kreativitu dětí. Nakreslené čáry se nemusí omezovat na rozměrově omezený displej tabletu. Lze kreslit na větší formáty papíru nebo spojovat čáry na více papírech. Další výhodou kreslení na papír je možnost, aby všechny děti pracovaly samostatně. Nejsme odkázání na dostupný počet tabletů. Fixy a papíry je možné zajistit pro všechny účastníky. I v případě, že má škola nebo zájmový kroužek Ozobotů méně než účastníků výuky, to nenarušuje práci. Děti si nejprve nakreslí čáry a příkazy a teprve následně zkoušejí, jak bude robot reagovat. Protože pracují různě rychle, není problém s půjčováním si robota.

Začít s kreslením čar se ukázalo vhodné i u vysokoškolských studentů. A to nejen z důvodů studijních. Cílem bylo, aby studenti věděli, jak Ozobota využít následně v učitelské práci. Jak se dalo očekávat, studenti velmi rychle zpracovali předložené úkoly. Svou invenci studenti ukázali při vytváření úkolů s příběhem, kdy robot musel projet určenými místy a provést zadané úkoly.

Druhou úroveň práce s robotem Ozobot představovalo vytváření kódu pomocí webových stránek http://ozoblockly.com. Programový kód se vytváří podobně jako například u aplikace Scratch. Příkazy jsou rozčleněny do pěti skupin podle obtížnosti [8, 9]. První skupina obsahuje jednoduché příkazy pohybu, světelných efektů a časové pauzy. Ikonky příkazů obsahují pouze grafické symboly a jsou tudíž vhodné i pro nejmenší děti. Další skupiny obsahují příkazy s doplňujícími textovými popisy. Určitou nevýhodou je, že v současné době jsou popisy i prostředí pouze v anglickém a polském jazyku. Během výuky si děti na anglický jazyk nestěžovaly. Pokud nebyly schopné samy přeložit některý z příkazů, zeptaly se učitele a po vysvětlení již dále příkazy používaly. Před samotnou prací bylo potřeba nastavit monitor na vyšší jas a kalibrovat robota, v opačném případě Ozobot nenačetl správně vytvořený program. Dětem se pracovalo lépe s tablety než stolním počítačem. Na dotykovém displeji dostatečné velikosti (alespoň 10“) žáci pohodlněji tvořili programy, které se jim lépe načítaly do robota. Toho pouze položili na tablet. Při využívání klasického monitoru, jehož obrazovka je ve svislé poloze, musely děti někdy i několik minut držet Ozobota u monitoru. Což hlavně u menších dětí byl občas problém. Někdy děti pohnuly s robotem více a bylo nutné začít s načítáním programu znovu.



Obrázek 6. Prostředí OzoBlockly [9]

Díky sledování výuky v uvedených skupinách lze konstatovat, že všechny účastníky práce s Ozobotem bavila. Vysokou motivovanost prokázali i chlapci ze čtvrté sledované skupiny, kteří se jinak o programování nezajímali. Děti nebylo nutné nutit do řešení problému. Naopak sami si vytvářely vlastní nákresy s příkazy a sami zkoumaly, co robot při použití konkrétních příkazů provede. Děti rozvíjely při vytváření i řešení svou fantazii a mnohé z nich uváděly, že normálně je kreslení nebaví, ale s Ozobotem ano. Ozobot je tedy rovněž vhodný nástroj pro rozvíjení některých motorických schopností a pro výuku programování, aniž by děti musely sedět u počítače.

5. SHRNUTÍ

Případová studie potvrdila, že Ozobot je jedna z možných vhodných pomůcek pro výuku algoritmizace a základů programování. Robot přináší do procesu výuky tolik potřebnou motivaci, ale je rovněž dostatečně názorný a zároveň jednoduchý v ovládání. Všichni účastníci případové studie byli ve velmi krátké době schopni pochopit vytváření barevných příkazů a dále pracovat samostatně. Mezi výhody Ozobota patří například možnost výuky programování bez potřeby používat počítač. V první fázi výuky postačí pouze Ozobot, papíry a fixy. Na druhé straně lze využívat kromě stolních počítačů i moderních mobilních zařízení jako jsou chytré telefony a tablety.

BIBLIOGRAFICKÉ ODKAZY

[1] *Strategie digitálního vzdělávání do roku 2020* http://www.msmt.cz/uploads/DigiStrategie.pdf

[2] GRAVES, C. The Great Robot Showdown-Ollie vs. Sphero vs Dash, 2015, https://colleengraves.org/2015/11/19/the-great-robot-showdown-ollie-vs-sphero-vs-dash/

[3] GARCIA-PEÑALO, F. J. Asurvey of resources for introducing coding into schools, *Proceedings of the Fourth International Conference on Technological Ecosystems for*

Enhancing Multiculturality, 2016, s. 19-26, ISBN 978-1-4503-4747-1

[4] http://ozobot.com

[5] HANSON, J. Tiny Ozobot Gets Kids into Block-based Programming, *School Library Journal*, February 2016 issue, http://www.slj.com/2016/02/opinion/test-drive/tiny-ozobot-gets-kids-into-block-based-programming-test-drive/

[6] KRATCOSKI, A. Campers Explore Programming Apps with Ozobots, *The Research Center for Educational Technology (RCET) Blog*, 2016, http://rcetatt.blogspot.cz/2016/07/campers-explore-programming-apps-with.html

[7] BROWN, P. Coding with Ozobots, *Tech & Learning*, 2016, http://www.techlearning.com/blogentry/10525

[8] BOUWMEESTER, J. Ozobot Bit Introduces Kids To STEM Programming Via Blockly, http://techaeris.com/2015/06/25/ozobot-bit-introduces-kids-stem-programming-via-blockly/

[9] http://ozoblockly.com

[10] http://portal.ozobot.com/lessons/type/lesson

Súťaž PALMA junior a programovanie v jazyku Python
The PALMA junior competition and programming in Python

Ján Guniš
PF UPJŠ v Košiciach
Jesenná 5
040 01 Košice
Slovensko
jan.gunis@upjs.sk

Ľubomír Šnajder
PF UPJŠ v Košiciach
Jesenná 5
040 01 Košice
Slovensko
lubomir.snajder@upjs.sk

ABSTRACT
Programming competitions are considered as one type of non-formal education which enables to develop pupils' algorithmic thinking, problem solving, critical thinking and creativity. Competitions allow mutual learning of pupils, develop their talents, and popularize programming. The PALMA junior competition (Programming, ALgorithms, and MAthematics) is dedicated for lower and higher secondary school pupils. On the competition website are published on-line materials (instructions for solving problems and their authoring solutions, commentaries to pupils' solutions) that provide scientific and methodological support for teachers. We present a methodology of competition problems development also with regard to programming languages Imagine Logo and Python, in which pupils can upload their solutions. In the second part we analyze pupils' solutions of selected competition problems.

Keywords
Programming competition. PALMA junior. Python. Imagine Logo. Analyses of problems.

ABSTRAKT
Programátorské súťaže považujeme za jednu z možností neformálneho vzdelávania rozvíjajúcu algoritmické myslenie, schopnosť riešiť problémy, kritické myslenia a kreativitu žiakov. Súťaže umožňujú vzájomné učenie sa žiakov, podporujú talenty, popularizujú programovanie. Súťaž PALMA junior (Programovanie, ALgoritmy, MAtematika) je určená žiakom základných a stredných škôl. On-line materiály (návody na riešenie úloh a ich autorské riešenia, komentáre k žiackym riešeniam) publikované na stránke súťaže poskytujú učiteľom odbornú a metodickú podporu. Prezentujeme metodiku tvorby úloh aj s ohľadom na programovacie jazyky Imagine Logo a Python, v ktorých môžu súťažiaci odovzdávať svoje riešenia. V druhej časti príspevku analyzujeme žiacke riešenia vybraných súťažných úloh.

Kľúčové slová
Programátorská súťaž. PALMA junior. Python. Imagine Logo. Analýza úloh.

1. ÚVOD
Rozvíjať schopnosť riešiť problémy, algoritmické myslenie, programátorské zručnosti, kritické myslenie a kreativitu žiakov môžeme cielene, okrem formálneho vzdelávania aj v rámci neformálneho vzdelávania. Vďaka rôznym formám neformálneho informatického vzdelávania, napr. krúžky, sústredenia, letné tábory [1, 2], letné školy [3], informatické súťaže [4] môžeme pozitívne vplývať na rôzne cieľové skupiny, napr.: talentovaní žiaci, žiaci so záujmom o informatiku, žiaci so zdravotným znevýhodnením [5]. Tradíciu na Slovensku majú najmä odborové súťaže. Na Slovensku registrujeme takmer dvadsať rôzne zameraných informatických súťaží, pri organizácii ktorých sú zainteresované najmä vysoké školy [6].

V príspevku uvádzame naše skúsenosti z programátorskej súťaže PALMA junior a zistenia zo žiackych riešení v programovacích jazykoch Imagine Logo a Python.

2. SÚŤAŽ PALMA JUNIOR
PALMA junior je súťaž v programovaní v jazykoch Imagine Logo a Python pre žiakov druhého stupňa základných škôl, prvého až tretieho ročníka stredných škôl a príslušných ročníkov osemročných gymnázií.

Do súťaže sa môžu prihlásiť dvojčlenné tímy žiakov alebo jednotlivci a súťaž prebieha v troch kategóriách (Profík, Expert, Guru). V každom školskom roku organizujeme tri on-line školské kolá a jedno finálové prezenčné kolo. Súťažné úlohy sú opravované jednotlivými hodnotiteľmi, nie automaticky. Každý súťažný tím dostane spätnú väzbu a komentár k svojmu riešeniu. Po ukončení kola zverejňujeme (http://di.ics.upjs.sk/palmaj) komentované autorské riešenia súťažných úloh (v každom kole 7 úloh) a zoznam chýb, ktorých sa súťažiaci pri riešení úloh dopustili. Súčasťou autorského komentáru je analýza riešeného problému, v ktorej naznačujeme možný postup ako by sa mohla úloha riešiť. Našim cieľom teda nie je poskytnúť len hotové riešenie úlohy, ale aj postup ako sa k tomuto riešeniu dopracovať. Tieto materiály môžu využiť učitelia vo svojej praxi a žiaci v príprave na súťaž.

3. PROGRAMOVACIE JAZYKY SÚŤAŽE
Prvým programovacím jazykom súťaže (od školského roku 2005/2006) bolo Imagine Logo. Táto voľba bola logickým dôsledkom toho, aký programovací jazyk sa používal vo vyučovaní školskej informatiky. Postupom času sme si čoraz viac uvedomovali potrebu moderného, pre žiakov možno viac populárneho jazyka. A to nielen v súťaži samotnej, ale aj vo vyučovaní informatiky.

Na základe rozhovorov s učiteľmi informatiky v rámci stretnutí Klubu učiteľov informatiky [7] sme uvažovali o programovacom jazyku Python. Učitelia nám potvrdili záujem zaradiť tento jazyk do základného kurzu informatiky namiesto v tom čase používaného jazyka. Zároveň vyjadrili svoju neistotu nad týmto krokom. Chýbali im skúsenosti, vhodné študijné a didaktické materiály. Rozhodli sme sa preto, od školského roka 2015/2016

umožniť žiakom riešiť súťažné úlohy aj v jazyku Python. Učiteľom sme týmto krokom poskytli nielen zaujímavé úlohy riešiteľné v školskej informatike v jazyku Python, ale prostredníctvom webovej stránky súťaže aj návody na riešenie a zoznam žiackych chýb, ktorých sa súťažiaci pri riešení úloh dopustili. Jazyk Python má pre nás aj ďalšie výhody. Podobne ako v Imagine Logu vieme používať korytnačiu grafiku a dátovú štruktúru zoznam. Python je distribuovaný pod otvorenou licenciou kompatibilnou s GPL a je možné ho spustiť na rôznych platformách včítane mobilných (napr. Android, iOS). Python je multi-paradigmový jazyk, vďaka čomu podporuje aj v našich školách zaužívaný procedurálny prístup.

V závere školského roka 2015/2016 sme zrealizovali dotazníkový prieskum na základných a stredných školách Prešovského a Košického samosprávneho kraja. Jeho cieľom, okrem iného bolo zistiť, aké programovacie prostredia (jazyky) učitelia vo výučbe používajú, koľko priestoru venujú učitelia výučbe algoritmizácie a programovania a o aký typ vzdelávania by mali záujem. Z odpovedí 77 respondentov sme zistili, že vo vyučovaní školskej informatiky majú programovacie jazyky nasledovné zastúpenie.

Tabuľka 1. Zastúpenie programovacích jazykov vo vyučovaní školskej informatik – najviac zastúpené jazyky

Typ školy	Imagine Logo	Baltík	Scratch	Pascal, rôzne verzie	Python	PHP	Jazyky pre robotické programovanie
ZŠ	25	10	6	1	0	0	3
G8	5	0	3	7	1	0	1
G4	6	0	1	14	3	2	0
SOŠ	5	2	1	5	1	4	4
iné	3	0	2	4	1	1	2

Keďže jazyk Python má vo vyučovaní informatiky stále minoritné postavenie, súťažiacim sme aj naďalej ponechali možnosť odovzdávať riešenia okrem jazyka Python aj v jazyku Imagine Logo. Zatiaľ čo žiaci v najnižšej kategórii uprednostňujú Imagine Logo, v najvyššej kategórii dávajú žiaci prednosť jazyku Python.

4. PRÍPRAVA SÚŤAŽNÝCH ÚLOH

Na príprave súťažných úloh pre každé kolo spolupracuje tím ľudí. Súčasťou tímu sú didaktici informatiky, študenti učiteľstva informatiky a odborníci z oblasti informatiky a prírodných vied. Výber úloh a formulácia ich zadaní je iteračný proces. Naším cieľom je, aby riešenie úlohy nebolo príliš rozsiahle (vzhľadom na množstvo kódu). Žiak nepotrebuje dokonale ovládať programovací jazyk a jeho nuansy. Vystačí si zo základmi jazyka. Kladieme dôraz na dobrú analýzu problému a návrh jeho riešenia než na programovací jazyk samotný. Okrem aspektov **P**rogramovania, **AL**goritmov a **MA**tematiky problémy formulujeme ako STEM (**S**cience, **T**echnology, **E**ngineering and **M**athematics), resp. STE**A**M (STE **A**rts M) problémy. Samotné zadanie je formulované ako zaujímavý reálny problém [8] tak, aby ho žiak bol nútený čítať s porozumením. Nepoužívame teda typicky strohé programátorské zadania úloh.

Ďalším špecifikom, ktoré je potrebné pri návrhu úloh zohľadniť, je možnosť riešiť úlohu v jednom z dvoch programovacích jazykov. Našou snahou je, aby riešenie a jeho náročnosť boli čo najviac podobné v oboch jazykoch. Tento fakt nás niekedy značne obmedzuje. Napríklad:

- V starších zadaniach úloh určených len pre Imagine Logo bolo častou úlohou analyzovať nejaký dvojrozmerný obraz vykreslený na ploche čítaním farby jeho jednotlivých bodov (farbaBodu). Keďže Python pri vykresľovaní (modul turtle) nepracuje s rastrovou grafikou, tento typ úloh je ťažko použiteľný.

alebo príklad z druhej strany:

- Python pomerne jednoducho pristupuje k prvkom a modifikácii prvkov zoznamov. V Imagine Logu je zápis takejto akcie neporovnateľne komplikovanejší. Porovnajme rozdiel pre dvojrozmerný zoznam, napr.:

  Python:
  ```
  if zoznam2D[i][j] == hodnota1:

      zoznam2D[i][j] = hodnota2
  ```

  Imagine Logo:
  ```
  ak prvok :j prvok :i :zoznam2D = :hodnota1 [

    urobTu "zoznam2D nahrad :i :zoznam2D nahrad
  :j prvok :i :zoznam2D :hodnota2

  ]
  ```

Prípravou úloh a vyhodnotením žiackych riešení súťažné kolo nekončí. Publikovaním autorského riešenia a komentáru k nemu sa snažíme žiakov ďalej vzdelávať ako pristupovať k riešeniu problémov, akú stratégiu riešenia zvoliť, ako a prečo problém analyzovať a v neposlednom rade ako korektne a zrozumiteľne zapísať riešenie v danom programovacom jazyku.

V jednotlivých domácich kolách súťaže zaraďujeme náročnejšie aj jednoduchšie úlohy. Jednoduchšie úlohy zaraďujeme z dôvodu motivácie pre začínajúcich súťažiacich, aby neskončili s nízkym, príp. nulovým ohodnotením ich riešení. Náročnejšie úlohy zaraďujeme preto, aby sme lepšie rozlíšili úroveň súťažiacich. Do finálového kola postupujú najlepší riešitelia z prvých troch domácich on-line kôl, preto sem zaraďujeme náročnejšie úlohy.

5. ANALÝZA VYBRANÝCH ÚLOH

5.1 Úloha o počte medailí

Úlohu o počte medailí sme zaradili ako tretiu do finálového kola 11. ročníka súťaže. Tretia úloha je spoločná pre najnižšiu (Profik) a strednú (Expert) kategóriu súťažiacich. Skrátené zadanie úlohy je nasledovné (kompletné súťažné zadanie je zverejnené na adrese http://di.ics.upjs.sk/palmaj/zadania/2015_2016/4/3.pdf):

„V olympiáde v informatike sú medaily udeľované podľa počtu získaných bodov tak, aby nanajvýš polovica súťažiacich získala medailu a aby počty zlatých, strieborných a bronzových medailí boli v pomere 1:2:3. Koľko ktorých medailí sa udelí pre zadaný počet tímov ak chceme dosiahnuť, aby sme medailou odmenili čo najviac súťažiacich? Predpokladajme, že žiadne dva tímy nedosiahnu rovnaký počet bodov.“

Od súťažiacich sme očakávali procedúru (Imagine Logo), resp. funkciu (Python), ktorá pre zadaný počet tímov vráti počty jednotlivých medailí.

Touto úlohou sme sledovali niekoľko faktorov:

Dôsledná analýza problému:

- počet medailí je obmedzený zhora (nanajvýš polovica súťažiacich získa medailu),
- počet medailí nie je ohraničený zdola a teda môže byť aj 0,
- pomer počtu jednotlivých medailí je 1:2:3,
- ak existujú viaceré riešenia, zoberieme to, v ktorom sa udelí najviac medailí (aby sme medailou odmenili čo najviac súťažiacich).

Dôslednejšou analýzou však zistíme, že sú situácie, v ktorých nevieme dodržať všetky pravidlá. Ak je počet súťažiacich menší ako 12, polovica z tohto počtu je menšia ako 6. Číslo menšie ako 6 už nevieme rozdeliť na tri celé čísla v pomere 1:2:3. Ak v tomto prípade neudelíme žiadnu medailu porušíme pravidlo o vzájomnom pomere počtu jednotlivých typov medailí. Ak udelíme nejaké medaily, nedodržíme ich vzájomný pomer alebo udelíme medailu viac ako polovici tímov. Úloha teda nemá vždy riešenie.

Ak zoberieme do úvahy len pravidlo o vzájomnom pomere, najmenší nenulový počet medailí, ktoré môžeme udeliť je 6 (jedna zlatá, dve strieborné a tri bronzové). Ďalšie možné počty sú 12, 18, 24, atď. Teda násobky čísla 6.

Návrh riešenia:

Na základe predchádzajúcej analýzy môžeme navrhnúť nasledovné riešenie.

Ak počet tímov je menší ako 12, úloha nemá riešenie.

Ak počet tímov je aspoň 12, nájdime najväčší násobok čísla 6, ktorý je menší alebo rovný polovici počtu tímov. Rozdelením tohto čísla v pomere 1:2:3 získame počty jednotlivých medailí.

Matematické operácie:

Hľadaný násobok čísla 6 nemusíme hľadať od najmenšieho, ale vieme ho priamo vypočítať:

$$maxPočet = \text{celá časť}\left(\frac{\text{počet súťažiacich}}{2}\right)$$

$$počet = maxPočet - zvyšok\left(\frac{maxPočet}{6}\right)$$

$$početZlatých = \frac{počet}{6}$$

$$početStrieborných = \frac{počet}{3}$$

$$početBronzových = \frac{počet}{2}$$

Použitie konštrukcií programovacieho jazyka:

Posledným krokom je zápis riešenia vo vybranom programovacom jazyku a otestovanie správnosti riešenia.

V požiadavkách na riešenie úlohy bolo uvedené, že požadujeme procedúru, resp. funkciu, ktorá vráti výsledok. Nie vypíše alebo inak zobrazí. V tomto bode nás zaujímalo, ako si súťažiaci poradia v situácii, keď výsledkom výpočtu nie je jedna, ale tri hodnoty. Jednou z možností je použiť nejakú štruktúrovanú premennú a výsledné hodnoty do nej „zabaliť“. Zatiaľ čo Python za príkazom `return` z viacerých hodnôt vytvorí automaticky n-ticu, v Imagine Logu je potrebné to spraviť explicitne (v Pythone sme akceptovali aj riešenie, v ktorom boli návratové hodnoty explicitne zabalené do zoznamu, resp. n-tice).

Ďalšou, pre nás zaujímavou časťou riešenia žiakov je, ako si poradia so situáciou, v ktorej úloha nemá riešenie. Štandardne by túto situáciu programátor zrejme riešil vyhodením nejakej výnimky (`hlásChybu`, resp. `raise`). My sme sa uspokojili s riešením, keď procedúra, resp. funkcia vrátila (nie vypísala alebo inak zobrazila) nejakú chybovú správu.

V samotnej realizácii programu môžeme niektoré kroky výpočtu spojiť. Kód autorského riešenia úlohy je nasledovný:

Riešenie v Imagine Logu:

```
viem pocetMedaili :pocetSutaziacich
  ak :pocetSutaziacich < 12 [
    vy "|Uloha nema riesenie|
  ]
  urobTu "pocet
         (cPodiel :pocetSutaziacich 2) –
         (zvysok cPodiel :pocetSutaziacich 2 6)
  vy (zoznam :pocet / 6
             :pocet / 3
             :pocet / 2)
koniec
```

Riešenie v Pythone:

```
def pocetMedaili(pocetSutaziacich):
    if pocetSutaziacich < 12:
        return "Úloha nemá riešenie."
    pocet = pocetSutaziacich // 2 - \
            pocetSutaziacich // 2 % 6
    return pocet // 6, pocet // 3, pocet // 2
```

Analyzovali sme riešenia žiakov v nasledovnom zložení:

Tabuľka 2. Zastúpenie riešení úlohy O počte medailí

kategória	riešenie v Imagine Logu	riešenie v Pythone
Profík	4	0
Expert	5	2

Analýzou žiackych riešení sme zistili, že žiaci sa dopúšťajú najmä dvoch chýb.

Nedôsledná analýza úlohy. Dôsledkom je, že až v 8 z 11 riešení žiaci neuvažovali o riešiteľnosti úlohy.

Nedôsledné, povrchné čítanie zadania úlohy. Dôsledkom je, že až v 9 z 11 riešení sa výsledok vypísal namiesto toho, aby sa použil ako návratová hodnota procedúry, resp. funkcie. V tomto prípade však nemôžeme pripísať všetko na úkor nedôsledného prečítania úlohy. Dôvodom, prečo žiaci výsledok vypísali namiesto jeho vrátenia môže byť aj fakt, že nedokázali vyriešiť ako vrátiť tri hodnoty „v jednom“. Zaujímavá je aj skutočnosť, že v obidvoch riešeniach v Pythone funkcia vrátila výsledok. Opäť môžeme polemizovať o tom, či je to dôsledok toho, že títo žiaci sú už skúsenejší alebo dôsledok toho, že použili (a zrejme používajú aj mimo súťaže) jazyk Python. Kým školské úlohy v Imagine Logu sú smerované skôr k výpisom výsledkov, Pythonovské programovanie je smerované skôr k návratovým hodnotám funkcií.

S príchodom možnosti riešiť súťažné úlohy v jazyku Python sa aj my snažíme zadania orientovať tak, aby procedúry, resp. funkcie

výsledky vracali namiesto ich vypisovania. Takéto riešenia sú univerzálnejšie a použiteľné ako súčasť iných výpočtov.

Ak si odmyslíme chyby, kde žiaci zabudli overiť riešiteľnosť úlohy, resp. výsledky vypísali boli chybné len 3 riešenia. Žiaci teda zvládnu nájsť riešenie pre bežné prípady, zabúdajú však na krajné, extrémne prípady.

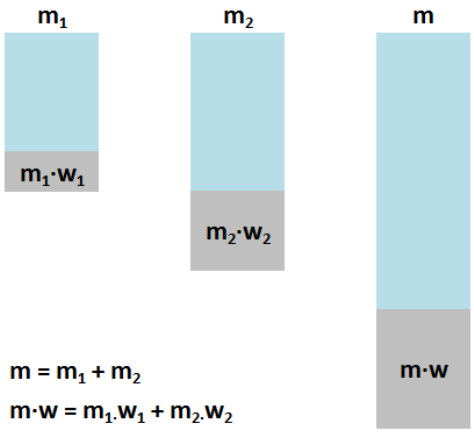
5.2 Úloha o príprave roztoku s morskou vodou

Zadanie úlohy (publikované na webovej stránke súťaže http://di.ics.upjs.sk/palmaj/zadania/2015_2016/4/5.pdf) je vložené do kontextu zoologickej záhrady, v ktorej potrebujú upraviť pôvodný slaný roztok pomocou ďalšieho slaného roztoku na výsledný roztok s požadovanou hmotnosťou a požadovaným hmotnostným percentom. Úlohou súťažiacich bolo vytvoriť funkciu, ktorá pre zadanú hmotnosť a hmotnostné percento pôvodného slaného roztoku a hmotnosť a hmotnostné percento výsledného roztoku, vypočíta hmotnosť a hmotnostné percento pridaného roztoku.

Úloha bola zaradená vo finálovom kole súťaže v školskom roku 2015/2016 ako výpočtová úloha pre strednú (Expert) a zároveň pre najvyššiu (Guru) kategóriu súťažiacich. Je zameraná na využitie logického myslenia, dôslednosť analytického uvažovania (ošetrenie vstupov a výstupov), na precvičenie použitia funkcií s parametrami, príkazov priradenia a vetvenia a zložených podmienok. Úloha patrí do kategórie STEM vďaka výpočtu hmotnostného percenta, ktoré je obsahom vyučovania chémie. Aby sme eliminovali rozdiely medzi žiakmi rôznych ročníkov, v poznámke na konci zadania úlohy uvádzame vymedzenie pojmu hmotnostné percento spolu s konkrétnym príkladom.

Na webovej stránke súťaže (http://di.ics.upjs.sk/palmaj/zadania/2015_2016/4/riesenia_komentare/uloha5.html) uvádzame komentované autorské riešenie úlohy spolu so zdrojovými kódmi v programovacích jazykoch Imagine Logo a Python.

Komentár autorského riešenia začína analýzou zadania úlohy (čo už vieme a čo potrebujeme zistiť), ktorá je zvyčajne podporená pomocnou schémou (obrázok 1).



Obrázok 1. Vzťahy medzi hmotnosťami a hmotnostnými zlomkami dvoch roztokov a výsledného roztoku

Návrh riešenia úlohy je završený odvodením vzťahov medzi vstupnými a výstupnými premennými a určením podmienok riešiteľnosti úlohy. Kód autorského riešenie úlohy v jazyku Imagine Logo vyzerá nasledovne:

```
viem vypocet :m1 :wp1 :m :wp
  urobTu "w1 :wp1 / 100
  urobTu "w :wp / 100
  urobTu  "kladne  (zároveň  :m1  >  0  :w1  >  0
                              :m > 0  :w > 0)
  urobTu "minimalne (zároveň
                      (:m - :m1) > 0
                      (:m * :w - :m1 * :w1) >= 0
                    )
  urobTu "maximalne
       (:m * :w - :m1 * :w1) <= (:m - :m1)
  ak2 (zároveň :kladne :minimalne :maximalne) [
    urobTu "m2 :m - :m1
    urobTu "w2 100 * (:m * :w - :m1 * :w1) / :m2
    výsledok zoznam :m2 :w2
  ][
    výsledok [Úloha nemá riešenie pre dané vstupy]
  ]
koniec
```

Kód autorského riešenia úlohy v jazyku Python vyzerá nasledovne:

```
def vypocet(m1, wp1, m, wp):
    w1 = wp1 / 100
    w  = wp  / 100
    kladne = (m1 > 0) and (w1 > 0) and (m > 0)and\
             (w > 0)
    minimalne = ((m - m1) > 0) and \
                ((m * w - m1 * w1) >= 0)


    maximalne = (m * w - m1 * w1) <= (m - m1)
    if kladne and minimalne and maximalne:
        m2 = m - m1
        w2 = 100 * (m * w - m1 * w1) / m2
        return m2, w2
    else:
        return "Úloha nemá riešenie pre dané \
               vstupy"
```

Uvedené programové kódy riešenia úlohy ukazujú, že riešenia sú rovnocenné v oboch programovacích jazykoch.

Analyzovali sme riešenia žiakov v nasledovnom zložení:

Tabuľka 3. Zastúpenie riešení úlohy O príprave roztoku

kategória	riešenie v Imagine Logu	riešenie v Pythone
Expert	4	2
Guru	0	2

Tri tímy vyriešili úlohu na plný počet bodov, priemerná úspešnosť riešenia úlohy bola skoro 80 %.

Úspešnosť riešenia úlohy nezávisela od použitého programovacieho jazyka, ale od kvality analýzy úlohy a hlavne v dôslednosti pri ošetrovaní požadovaných vstupov

a očakávaných výstupov. V dvoch žiackych riešeniach sa nedôslednosť prejavila v použití hmotnostných zlomkov na vstupe namiesto hmotnostných percent. Zaujímavé je zistenie, že aj keď sa v zadaní úlohy nepožadovalo explicitne vrátenie hodnoty funkciou, všetky riešenia v Pythone ho obsahujú. Z riešení v Imagine logu ho neobsahuje žiadne.

Žiacke riešenia ukázali, že chemický koncept hmotnostný zlomok, resp. hmotnostné percento žiaci pochopili a správne ho použili pri riešení tejto problémovej úlohy.

6. ZÁVER

Súťaž PALMA junior ako príklad neformálneho vzdelávania pokladáme za súčasť vzdelávania a samovzdelávania sa žiakov popri ich riadnej výučbe informatiky. Okrem rozvíjania programátorských zručností a kompetencii riešiť problémy, súťaž umožňuje rozvíjať a prepájať poznatky z matematiky a prírodných predmetov a tiež rozvíjať schopnosť čítania s porozumením. Veľký benefit súťaže vidíme aj vo vysokoškolskej príprave a ďalšom vzdelávaní učiteľov informatiky, ktorí majú v súčasnosti k dispozícii 254 súťažných úloh (zadaní, autorských riešení, komentárov k žiackym riešeniam) v jazyku Imagine Logo a 42 z nich aj v jazyku Python. Od začiatku súťaže sa naši študenti učiteľstva informatiky zúčastňujú pri tvorbe, opravovaní a vyhodnotení súťažných úloh, čím získavajú autentické skúsenosti pri práci s talentovanou mládežou. Súčasťou vysokoškolskej prípravy učiteľov informatiky na našej fakulte je aj kurz programovania v jazyku Python v rámci výučby predmetu Školské programovacie prostredia.

Na základe analýzy uvedených dvoch súťažných úloh nemožno vyvodzovať nejaké všeobecné závery. Možno nie je dôležité zistenie, že Pythonovské riešenia použili funkciu s návratovou hodnotou a tie v Imagine Logu nie. A to bez ohľadu na to, či táto požiadavka v zadaní bola alebo nie. Možno si treba uvedomiť, že rozdiel medzi Imagine Logom a Pythonom nie je len v tom, že príkazy sa volajú inak. Je to štýl programovania? Je to spôsob uvažovania, myslenia? V každom prípade môže byť problematické, ak zmenu programovacieho jazyka vo vyučovaní programovania realizujeme len „preložením“ riešení z jedného jazyka do druhého. Mali by sme zvažovať nielen typ úloh ale aj spôsob ako riešenie implementovať. Chceme využiť to, v čom je jazyk silný [9] a nie ho použiť neprirodzene na obraz iných jazykov. Aby sme sa vyhli takémuto neprirodzenému použitiu programovacieho jazyka, do budúcnosti uvažujeme v najnižšej kategórii (Profík) riešiť úlohy len v jazyku Imagine Logo a v najvyššej kategórii (Guru) len v jazyku Python.

V súčasnosti vyvíjame e-zbierku úloh súťaže PALMA junior. Určená bude súťažiacim pri príprave na súťaž, ich učiteľom pri zostavovaní súborov úloh nielen pre súťažiach, ale aj pre riadnu výučbu programovania.

POĎAKOVANIE

Táto práca bola podporovaná projektom VEGA 1/0073/15 Škálovateľné výpočtové metódy analýzy štruktúrovaných a neštruktúrovaných dát s prvkami neurčitosti.

BIBLIOGRAFICKÉ ODKAZY

[1] CÁPAY, M., LOVÁSZOVÁ, V, MICHALIČKOVÁ, V. Neformálne vzdelávanie detí formou voľnočasových informatických aktivít. In: *DidInfo 2014*. Banská Bystrica : UMB, 2014, s. 26-31. ISBN 978-80-557-0698-6. Dostupné tiež na: http://didinfo.umb.sk/public/filestore/documents/richtext/161/zbornik_didinfo_2014.pdf

[2] ŠNAJDER, Ľ., GUNIŠ, J., ANTONI, Ľ. Model of Teaching of Mobile Applications Programming in Non-Formal Education. In: *Proceedings of the 7th International Conference on Research in Didactics of the Sciences.* Kraków : Pedagogical University of Cracow, Institute of Biology, Department of Education of Natural Sciences, 2016, p.151-154. ISBN 978-83-8084-037-9

[3] PETROVIČ, P. Robotika na letných školách. In *DidInfo 2013*. Banská Bystrica : UMB, 2013, s. 188-193. ISBN 978-80-557-0527-9. Dostupné tiež na: http://didinfo.umb.sk/public/filestore/documents/richtext/153/zbornik_didinfo_2013.pdf

[4] TOMCSÁNYIOVÁ, M. Informatické koncepty v úlohách súťaže iBobor. In: *Didinfo 2014*, Banská Bystrica : UMB, 2014. s. 182-190. ISBN 978-80-557-0698-6. Dostupné tiež na: http://didinfo.umb.sk/public/filestore/documents/richtext/161/zbornik_didinfo_2014.pdf

[5] JAŠKOVÁ, Ľ., KOVÁČOVÁ, N. Prvé skúsenosti s realizáciou súťaže iBobor pre nevidiacich žiakov druhého stupňa ZŠ. In: *Didinfo 2015*, Banská Bystrica : UMB, 2014. s. 78-83. ISBN 978-80-557-0852-2. Dostupné tiež na: http://didinfo.umb.sk/public/filestore/documents/richtext/181/didinfo_2015.pdf

[6] ŠNAJDER, Ľ., GUNIŠ, J. *Tvorba úloh pre programátorské súťaže.* 1. vydanie. Košice : UPJŠ v Košiciach, 2014. 79 s. ISBN 978-80-8152-139-3. Dostupné tiež na: http://unibook.upjs.sk/image/data/Nahlady/Tvorba-uloh-pre-prog-sutaze.pdf

[7] KLUB UČITEĽOV INFORMATIKY 2017. http://skoly.upjs.sk/aktivity/course/view.php?id=8

[8] VANÍČEK, J. Bebras Informatics Contest: Criteria for Good Tasks Revised. In: *ISSEP 2014*, Berlin : Springer-Verlag, 2014. p. 17-28. ISBN 978-3-319-09957-6.

[9] BLAHO, A., TOMCSÁNYI, P. Programovací jazyk Python vo výučbe predmetu Algoritmy a dátové štruktúry. In: *DidInfo 2015*. Banská Bystrica : UMB, 2015. s. 155-159. ISBN 978-80-557-0852-2. Dostupné tiež na: http://didinfo.umb.sk/public/filestore/documents/richtext/181/didinfo_2015.pdf

Vývoj edukačných webových aplikácií pomocou knižnice SPRITES

The development of educational web applications using the library SPRITES

Roman Hrušecký, Monika Tomcsányiová
Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky,
Katedra základov a vyučovania informatiky
Mlynská dolina, 842 48 Bratislava, Slovensko
hrusecky@fmph.uniba.sk, tomcsanyiova@fmph.uniba.sk

ABSTRACT

In this paper we describe the first experience in development of educational web applications using JavaScript library SPRITES and compares the development of these applications in JavaScript without the use of said library. SPRITES library was developed by KEGA project and aims to facilitate the development of small educational applications. With it could be developing web applications also available for programmers who do not have much experience with web programming and using HTLM 5 and JavaScript, for example student teachers combinations.

As topics for pilot applications we chose the so-called interactive tasks iBobor competition, see [1]. These are the tasks that may be small educational applications already in the form in which they are in competition, but several of them have the potential for its further expansion, for example that the computer will generate a lot of similar tasks with different settings, different intensity and to check the correctness of pupil solutions.

Keywords

Educational application. Javascript. SPRITES.

ABSTRAKT

V príspevku popíšeme prvé skúsenosti z vývoja edukačných webových aplikácií pomocou javascriptovej knižnice SPRITES a porovnáme ich s vývojom týchto aplikácií v jazyku Javascript bez využitia uvedenej knižnice. Knižnica SPRITES vznikla v rámci projektu KEGA a jej cieľom je uľahčiť vývoj malých edukačných aplikácií. Vďaka nej by mohol byť vývoj webových aplikácií dostupný aj programátorom, ktorí nemajú veľké skúsenosti s webovým programovaním a využívaním HTLM 5 a JavaScriptu, napr. študentom učiteľských kombinácií.

Ako témy pre pilotné aplikácie sme si zvolili tzv. interaktívne úlohy súťaže iBobor, pozri [1]. To sú úlohy, ktoré môžu byť malými edukačnými aplikáciami už v tvare, v akom sa nachádzajú v súťaži, ale viaceré z nich majú potenciál pre svoje ďalšie rozširovanie, napr. tým, že počítač bude generovať veľké množstvo podobných úloh s rôznymi nastaveniami, rôznej náročnosti a kontrolovať správnosť žiackeho riešenia.

Kľúčové slová

Edukačná aplikácia. Javascript. SPRITES.

1. ÚVOD

V súčasnosti zažívame veľký rozmach mobilných zariadení, predovšetkým tabletov a inteligentných mobilných telefónov. Spoločnosti stále vyvíjajú nové a dokonalejšie zariadenia, no do školského prostredia sa tieto nové technológie dostávajú len veľmi pomaly. V poslednom období sa však viacerým základným a stredným školám podarilo získať tablety, ktoré sú súčasťou školských laboratórií. V súčasnosti vidíme tiež trend, v ktorom sa realizujú projekty, vďaka ktorým by sa do škôl mali dostať ďalšie mobilné zariadenia, najčastejšie opäť tablety.

Mobilné zariadenia, tablet, čítačku elektronických kníh, či mobilný telefón, používajú študenti vysokých a stredných škôl a často už aj žiaci základných škôl takmer každý deň. Najčastejšie pomocou nich vyhľadávajú informácie alebo ich používajú na komunikáciu v sociálnych sieťach. Menej ich však využívajú na vzdelávanie, t. j. svoje vlastné učenie sa.

2. CIELE A LÍNIE PROJEKTU

Hlavným cieľom projektu KEGA je podporiť tvorbu vzdelávacieho digitálneho obsahu pre mobilné zariadenia. Projekt má niekoľko cieľov. Jedným z nich je navrhnúť metodiku pre vývojára edukačných mobilných aplikácií a podľa nej vyvinúť niekoľko pilotných edukačných mobilných aplikácií, a tak prispieť k rozšíreniu digitálneho vzdelávacieho obsahu.

Keďže možností, ako vyvíjať mobilné aplikácie je veľa, postupne sme zúžili naše smerovanie na tri línie:

a) MIT App Inventor (bližšie o vývoji v MIT App Inventore sme písali v článku *Vyučovanie programovania vývojom androidových aplikácií v App Inventore*, pozri [2]),

b) prostredie Lazarus (*S Pascalom na Android*, pozri [3]),

c) programovací jazyk Javascript spolu s HTML5.

Každá z uvedených línií má svoje výhody a aj nevýhody a takisto je určená pre rôzne cieľové skupiny programátorov. V tomto článku sa zameriame na vývoj mobilných aplikácií pomocou jazyka JavaScript spolu s HTML5.

Naprogramovať aplikáciu v jazyku JavaScript si často vyžaduje nielen základné programátorské zručnosti, ale aj znalosť jazyka. Keďže našim cieľom je, aby jednoduché edukačné mobilné aplikácie mohli vytvárať aj učitelia informatiky a aj študenti učiteľských kombinácií na vysokých školách, rozhodli sme sa navrhnúť a vyvinúť javascriptovú knižnicu SPRITES. Veríme, že vďaka nej by mohol byť vývoj aplikácií v JavaScript-e dostupný aj

programátorom, ktorí nemajú veľké skúsenosti s webovým programovaním a využívaním technológie HTML 5.

3. KNIŽNICA SPRITES

Knižnica SPRITES je knižnicou bez vizuálneho a interaktívneho prostredia, ktorá programátorovi dostatočne zjednoduší prácu s vytváraním aplikácie, ale zároveň mu umožní doprogramovať aktivitu v reálnom programovacom jazyku – JavaScript-e.

Knižnica rieši viacero technických detailov, ktoré už programátor nemusí riešiť. Knižnica neobsahuje žiadne vizuálne prvky, ale poskytuje objekt Sprite a jeho niekoľko základných metód na prácu s myšou, pričom sa úplne stará o vykresľovanie objektov. Je postavená na objekte Canvas v HTML 5 (kresliace plátno), čiže všetky objekty sa vykresľujú do grafickej plochy podobne ako v iných programovacích jazykoch, napr. Delphi, C#. Knižnica bola navrhnutá a vytvorená tak, aby sa dala využiť na vývoj aplikácií, ktoré sa budú dať spúšťať nielen na počítačoch/notebookoch, ale aj na tabletoch a smartfónoch, teda dotykových zariadeniach.

Veríme, že ak programátor dostane k dispozícii knižnicu, ktorá mu poskytne riešenia viac-menej technických problémov, napr. v našom prípade vyrieši komunikáciu s webovým prehliadačom a vyrieši posielanie udalostí medzi jednotlivými objektmi, bude môcť aplikácie vytvárať väčšie množstvo ľudí, vrátane študentov učiteľského štúdia v kombinácii s informatikou, resp. učiteľov informatiky z praxe.

Pri návrhu a vývoji knižnice SPRITES sme vychádzali z tzv. interaktívnych úloh súťaže iBobor. Viaceré úlohy majú rovnaké alebo podobné programátorské „pozadie“, ktoré sme sa snažili začleniť do knižnice.

Každý objekt v knižnici je tzv. SPRITE. Ten môže byť aktívny (klikateľný, ťahací) alebo pasívny (len umiestnený, prípadne na neho môžeme umiestniť iný objekt-SPRITE).

Knižnica umožňuje tieto typy aktivít:

- Ťahanie a umiestňovanie objektov do niekoľkých cieľových pozícií, resp. na jednu pozíciu. V oboch typoch tejto aktivity môžeme ešte vyšpecifikovať, či môže byť na každej cieľovej pozícii práve jeden objekt alebo viac objektov. Pri grafickej realizácii aktivity môže ísť ešte o ďalšie dva pohľady: cieľové pozície môžu byť zobrazené ako prázdne miesta pre objekty alebo môžu byť len naznačené, napr. pomocou mrežových bodov.
- Vymieňacie aktivity tvoria druhú veľkú skupinu aktivít. Učitelia ich vo svojich malých edukačných programoch využívajú v prípade, že všetky objekty sú umiestnené na cieľových pozíciách. Počas riešenia úlohy môže žiak experimentovať s vymieňaním objektov medzi sebou.
- Ďalším typom aktivity sú také, v ktorých sa pri kliknutí na objekt zmení jeho obrázok. Pričom v niektorých prípadoch môže byť zmena obrázku pomerne nepatrná, napr. zmení sa farba výplne, obrázok sa prečiarkne alebo obrázok úplne zmení svoj tvar na iný.
- Posledný typ aktivity sú tzv. klikacie úlohy, v ktorých je úlohou používateľa nejakým spôsobom označiť obrázok. Väčšinou ide o orámovanie obrázka nejakou farbou, čím sa opticky odlíši od ostatných obrázkov.

Knižnica neobsahuje možnosť kontrolovať správnosť riešenia, ale predpokladáme, že programátor na úrovni učiteľa informatiky, resp. študenta učiteľského štúdia, dokáže navrhnúť časť programu, ktorá skontroluje správnosť žiakovho riešenia.

4. VYUŽITIE KNIŽNICE

Študenti magisterského štúdia aplikovanej informatiky (prevažne v tímoch) mali v rámci aktivít kurzu *Webové programovanie* naprogramovať hru na základe úlohy zo súťaže iBobor, pričom mali vytvoriť jednu jej verziu bez využitia knižnice SPRITES a druhú verziu s jej využitím.

Týmto zadaním sme si chceli potvrdiť, resp. vyvrátiť, jednu z hypotéz, ktorú sme si stanovili pri vývoji knižnice: či sa študenti pri programovaní tej istej úlohy bez využitia knižnice SPRITES rozhodnú pre **rovnaký alebo iný prístup** k vytváraniu a vykresľovaniu objektov v prostredí webového prehliadača.

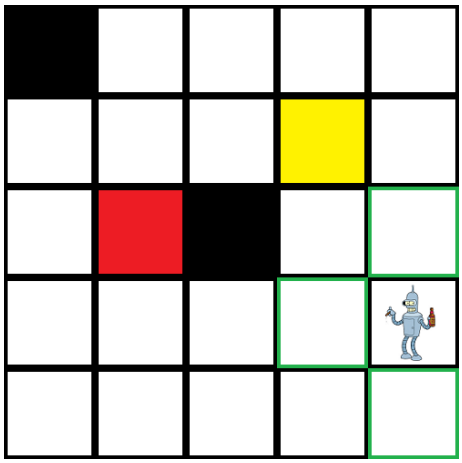
Tým, že sme si pre experiment vybrali skupinu pokročilých programátorov, ktorí mali dostatok vedomostí na vývoj aplikácií s použitím knižnice, ale aj bez nej, môžeme porovnať projekty, ktoré vytvorili s použitím knižnice a bez jej využitia.

Úloha študentov pri použití knižnice bola sťažená tým, že ku knižnici SPRITES neexistovala aktuálna verzia manuálu. V tom čase sa knižnice ešte vyvíjala a práve prechádzala väčšou zmenou. Napriek tomu niektoré tímy dokázali vytvoriť funkčnú hru aj s využitím knižnice SPRITES. Verzie bez knižnice SPRITES často využívali knižnicu JQuery, ktorá uľahčuje prácu s HTML objektmi a kaskádovými štýlmi. Pre programátora to však znamená, že musí ovládať danú knižnicu a takisto prácu s jazykom HTML a kaskádovými štýlmi aspoň na mierne pokročilej úrovni.

V ďalšej časti ukážeme niekoľko zadaní, ktoré mali študenti naprogramovať a tiež to, aký prístup si zvolili pre riešenie úlohy, v ktorom nepoužili knižnicu SPRITES.

4.1 Projekt Robot

Stručné zadanie: Robot upratuje v miestnosti žlté kocky – musí ich posúvať na správne miesta (červené). V ceste mu niekedy prekážajú čierne stĺpy. Robot sa pohybuje **VĽAVO**, **VPRAVO**, **HORE** alebo **DOLE**. Robota môžeme ovládať klávesnicou alebo myšou. Ak je v smere pohybu žltá kocka a za ňou prázdne políčko, kocku tam odtlačí.



Obrázok 1 – projekt Robot

Keďže knižnica SPRITES neobsahuje ovládanie objektov pomocou klávesnice, vytvorené programy sa líšia. Avšak len ovládaním. Inak študenti vytvorili ekvivalentné hry. Verziu hry, bez knižnice SPRITES nevytvorili pomocou objektu Canvas, ale pomocou štandardných obrázkov v HTML (objekt `<img>`). Pri analýze zdrojových kódov sme zistili, že realizácia bez knižnice

SPRITES bola mierne jednoduchšia, avšak vyžaduje si znalosť jazyka HTML a kaskádových štýlov CSS. Vďaka HTML a správnemu pomenovaniu objektov nepotrebujeme ďalšie pole, ale napr. typy objektov si môžeme pamätať priamo v objektoch plochy, prípadne im nastavovať vhodné triedy kaskádových štýlov (napr. kvôli podfarbeniu).

Ukážka testu steny v smere pohybu:

```
if($("#"+x+"a"+y).attr('value') == -1) {...}
```

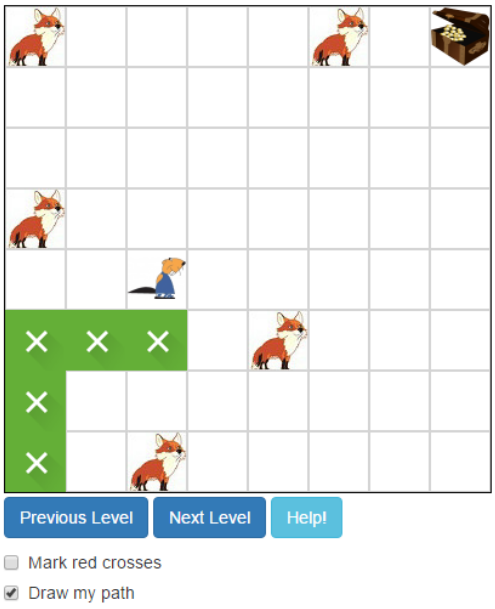
Reakcie na stlačenie klávesov študenti vyriešili štandardným spôsobom pomocou JQuery knižnice:

```
$(document).keydown(function(e){
  switch(e.which){
    case KEY.UP:
    // objekt s ASCII hodnotami klávesov
    ...
    break;
} }
```

Pri verzii riešenia s knižnicou SPRITES museli študenti použiť pole na zapamätanie si údajov. Keďže knižnica neobsahuje prácu s klávesnicou, študenti využili presúvanie objektov – funkciu **`onDragDrop(sprite)`** a robota dovolili ťahaním presúvať len na vedľajšie políčka. V rámci nej využili funkciu na nájdenie prekrývajúcich sa objektov **`sprite.findOverlapped(i)`**. Robot bol jediný objekt, ktorý sa dal presúvať. Ostatné objekty boli len „statické“.

4.2 Projekt Lov na poklad

Stručné zadanie: Bobor Kubo chce získať poklad, ktorý strážia líšky (musí sa im vyhnúť). Líška zbadá Kuba, len ak stúpi na niektoré políčko tesne vedľa nej (vo všetkých smeroch). Kuba môžeme ovládať klávesnicou alebo myšou.



Obrázok 2 – projekt Lov na poklad

Klasickú verziu študenti nevytvorili pomocou objektu Canvas, ale pomocou štandardných obrázkov v HTML (objekt **`<img>`**), rovnako ako tím, ktorý vytváral projekt *Robot*. Z analýzy kódu môžeme usudzovať, že tento spôsob bol pre daný typ hry pre nich jednoduchší.

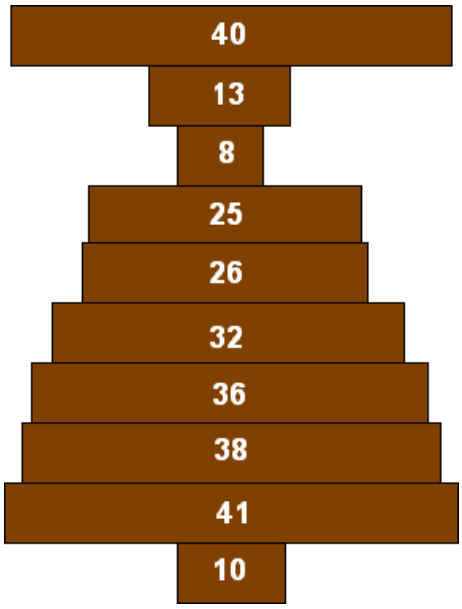
Študenti do hry doplnili dve nové nastavenia – *zobrazovanie cesty* bobra a *zobrazovanie polí, kam vidia líšky* (na ktoré nesmie bobor stúpiť).

Pri tomto projekte študenti využili knižnicu SPRITES pri editore úrovní. Z toho usudzujeme, že samotné vytváranie plochy s objektmi a hlavne presúvanie objektov bolo s pomocou knižnice jednoduchšie. Napríklad využili funkciu knižnice, že na jednom mieste nemôžu byť dva objekty, presnejšie ak presunieme objekt nad iný, vymenia si pozície.

4.3 Projekt Pyramída

Stručné zadanie: Faraón dal postaviť pyramídu z tehličiek, ktorých šírka sa zdola nahor zmenšuje. Na každej tehličke je zapísaná jej šírka. Po postavení pyramídy zistil, že je v nej niekoľko nesprávnych tehličiek, a preto ich dal nahradiť inými.

Hra môže mať niekoľko variant – pre jedného alebo dvoch hráčov, prípadne s extra tehličkami na výmenu.



Obrázok 3 – projekt Pyramída

Tento projekt sa principiálne odlišuje od predchádzajúcich dvoch – neodohráva sa na štvorcovej ploche. Preto je aj jeho realizácia odlišná od predchádzajúcich projektov.

Študenti pri štandardnom riešení využívajú objekt Canvas, a teda obe riešenia sa dajú podrobne porovnať. Funkcie a časti kódu, ktoré sú nezávislé od technológie, sú v oboch riešeniach rovnaké. Kým v SPRITES verzii vytvorili miesta pre tehličky ako „statické“ objekty a na ne umiestnili tehličky, v štandardnej verzii si toto museli študenti sami naprogramovať. Teda kód štandardnej verzie je dlhší, čo je logické.

5. ZÁVER

Na hodnotenie použiteľnosti knižnice SPRITES je zavčasu, no prvotné testovania ukazujú, že pre niektoré typy úloh (hier) je užitočná a uľahčuje vývoj aplikácie. Vytváraný kód je často kratší a programátor nemusí riešiť niektoré veci, ktoré za neho vyrieši alebo si pamätá knižnica.

Pri vývoji knižnice SPRITES sme si stanovili hypotézu: či sa študenti pri programovaní tej istej úlohy bez využitia knižnice

SPRITES rozhodnú pre **rovnaký alebo iný prístup** k vytváraniu a vykresľovaniu objektov v prostredí webového prehliadača.

Hypotéza sa nám zatiaľ nepotvrdila, aj keď nemôžeme robiť väčšie závery, keďže realizovaných projektov bolo zatiaľ málo. Zdá sa však, že skúsení programátori sa snažia využiť taký prístup, ktorý sa javí ako vhodnejší pre daný typ aplikácie. Ale takisto môžu byť v nemalej miere ovplyvnení ich programátorskými zručnosťami.

POĎAKOVANIE

Príspevok vznikol v rámci projektu KEGA č. 080UK-4/2015 *Podpora tvorby vzdelávacieho digitálneho obsahu pre mobilné zariadenia*.

BIBLIOGRAFICKÉ ODKAZY

[1] iBobor. http://ibobor.sk

[2] TOMCSÁNYIOVÁ, M., TOMCSÁNYI, P. Vyučovanie programovania vývojom androidových aplikácií v App Inventore. In *DidInfo 2015*. Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, 2015, s. 160-163. ISBN 978-80-557-0852-2. Dostupné tiež na: http://didinfo.umb.sk/sk/181/zbornik.html

[3] TOMCSÁNYI, P. S Pascalom na Android. In *DidInfo 2016*. Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, 2016, s. 106-111. ISBN 978-80-557-1082-2. Dostupné tiež na: http://didinfo.umb.sk/sk/202/zbornik.html

[4] SALANCI, Ľ.: knižnica SPRITES. nepublikovaná.

Ako zmysluplne využiť jednu Hodinu v škole
How to Use One Hour for a Meaningful Purpose

Soňa Ivančáková
Katedra informatiky,
FPV UMB Tajovského 40
974 01 Banská Bystrica Slovensko
Sona.ivancakova@gmail.com

Nika Klimová
Katedra informatiky,
FPV UMB Tajovského 40
974 01 Banská Bystrica Slovensko
nika.klimova2@gmail.com

Dana Horváthová
Katedra informatiky,
FPV UMB Tajovského 40
974 01 Banská Bystrica Slovensko
dana.horvathova@umb.sk

ABSTRACT

The paper talks about the worldwide „Hour of Code“ attended by tens of millions of students and supported by many famous people such as Bill Gates. The target is to attract people to programming. Primary school of J. Bakoss in Banská Bystrica has joined this event as one of several schools in Slovakia. We would like to introduce the attendance of the „Hour of Code“ through the paper and to show experience of many activities depending on children age and skills.

Keywords

The Hour of Code. Programming. Microsoft. The basics of programming.

ABSTRAKT

Príspevok je zameraný na priblíženie Hodiny kódu - celosvetového hnutia, ktorého sa zúčastňujú desiatky miliónov študentov z celého sveta s podporou mnohých známych osobností, vrátane Billa Gatesa. Hlavným cieľom tohto podujatia je pritiahnuť čo najviac ľudí k programovaniu. Tejto iniciatívy sa zúčastnilo viacero škôl na Slovensku vrátane Základnej školy Jána Bakossa v Banskej Bystrici. Našim príspevkom chceme predstaviť zapojenie sa tejto školy do samotnej Hodiny kódu a prezentovať skúsenosti z rôznych aktivít so žiakmi spomínanej školy so zreteľom na ich vek a schopnosti.

Kľúčové slová

Hodina kódu. Programovanie. Microsoft. Základy programovania.

1. ÚVOD

Vraví sa, že prvý dojem je dôležitý. V triedach, ktoré sme chceli oboznámiť so základmi programovania, sme si túto múdrosť zobrali k srdcu. Vybrali sme kurz Hodinu kódu, ktorý očaril už desiatky miliónov detí vo viac ako 180 krajinách po celom svete.

2. Charakteristika Hodiny kódu

Hodina kódu je celosvetová iniciatíva [https://hourofcode.com/us/sk], pod ktorú sa podpísala nezisková organizácia Code.org. Venuje sa zvyšovaniu záujmu o informatiku v spoločnosti a momentálne spolupracuje s partnermi ako Microsoft, Apple, Amazon či College Board a teší sa podpore zvučných mien v oblasti informatiky, napr. Marka Zuckerberga či Billa Gatesa. V prípade Slovenska sú to osobnosti ako olympionička Danka Barteková, spisovateľka Petra Nagyová-Džerengová, herec Juraj Kemka, či choreograf Jaro Bekr.

Ako už názov iniciatívy naznačuje, ide o programovanie, ktoré sa skladá z jednohodinových výučbových programov. Veľkou výhodou je, že sa nepožadujú žiadne predchádzajúce skúsenosti s programovaním. Aktivity sú určené pre deti i dospelých v rozmedzí od 4 – 104 rokov. Cieľom je prilákať deti k programovaniu a rozvíjať logiku, tvorivosť a riešenie problémových úloh. Výhodou je, že na využívanie programov nie je nutné sa prihlásiť, webstránka a programy fungujú aj bez registrácie.

2.1 Administrácia

Učiteľ alebo lektor má možnosť online sledovať progres svojich študentov. Po registrácii si môže vytvoriť skupinu, do ktorej sa žiaci prihlásia e-mailovou schránkou, obrázkom alebo slovom. Učiteľ dopredu zadá, v ktorej hre budú pracovať. U jednotlivých žiakov môže sledovať, koľko úrovní prešiel. Pokiaľ žiaci prejdú všetky úrovne, učiteľ im môže vytlačiť certifikáty s ich menami o tom, že kurz absolvovali.

2.2 Výber aktivít

Pri vytváraní kurzu má učiteľ alebo lektor široké možnosti. Tvorcovia sa snažili, aby sa do Hodiny kódu mohlo zapojiť čo najväčšie množstvo žiakov. Na výber je 193 aktivít trvajúcich okolo jednej hodiny. Takéto množstvo je potrebné zoskupiť podľa spoločných znakov. A tie sú:

- triedy: od predškolákov po stredoškolákov
- skúseností lektora: začiatočník / pokročilý
- skúseností študenta: začiatočník / pokročilý
- vybavenie triedy: počítače, Android, iPad / iPhone, bez internetu, bez počítačov
- témy: matematika, sociálne štúdiá, jazyky, umenie a hudba, informatika
- typ aktivity: samoštúdium, učebný plán
- dĺžka: 1 hodina, 1 hodina s pokračovaním, viac hodín
- jazyk: bloky, písanie, iné

Filtrovať sa dajú aj najpopulárnejšie hry. Medzi TOP 10 patrí:

1. Minecraft
2. Moana
3. Make a Flappy game
4. Star Wars
5. Gumball´s Coding Adventure
6. Code your own sports game
7. Kodable
8. Code with Anna and Elsa
9. Code Combat
10. Write your first computer program

Obrázok 1 Kurz Star Wars

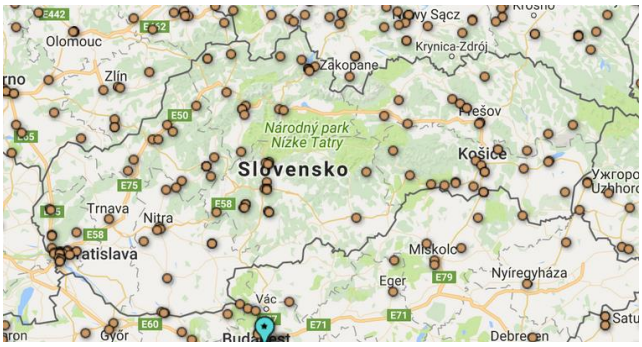
Ako si mohol čitateľ všimnúť, mnoho z týchto kurzov vychádza práve z animovaných rozprávok alebo hier, aby boli žiaci dostatočne motivovaní a zo zoznamu si môžu vyberať nielen chlapci, ale aj dievčatá.

3. Hodina kódu na Slovensku

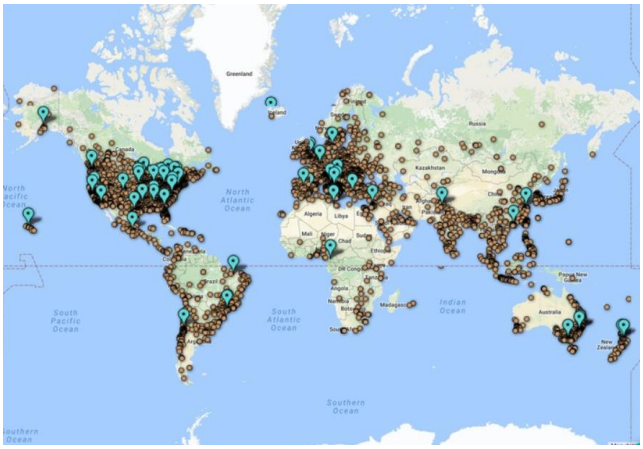
Je dôležité podotknúť, že je po Hodine kódu už značný dopyt aj na Slovensku. Nielen tým, že spoločnosť Microsoft sa snaží propagovať túto udalosť na svojej webstránke, ale aj už spomenutými osobnosťami, ktoré sa snažia slovenských žiakov prilákať k programovaniu.

Na to, aby sa mohlo do kurzu zapojiť čo najviac žiakov, bolo potrebné preložiť tieto výukové programy. Preklad zabezpečujú dobrovoľníci pod vedením Mgr. Zuzany Molčanovej.

Pokiaľ sa učiteľ alebo lektor rozhodol uskutočniť Hodinu kódu, dostal príležitosť sa zaregistrovať, aby bolo vidno, kde udalosť prebieha. Na Slovensku bolo zaregistrovaných približne 130 škôl a firiem.



Obrázok 2 Znázornené Hodiny kódu na mape Slovenska



Obrázok 3 Mapa sveta a vyznačené Hodiny kódu

4. Hodina kódu na ZŠ Jána Bakossa

V roku 2016 sa konala celosvetová udalosť Hodina kódu v rozmedzí 5. – 11. decembra. Zúčastnila sa jej aj Základná škola Jána Bakossa v Banskej Bystrici. Do udalosti sa zapojili triedy 3.-6. ročníka. Naším cieľom bolo priblížiť a zaujať mladú generáciu v snahe rozvíjať algoritmické riešenie a záujem o informatiku so zreteľom na pozitívny zážitok.

Snažili sme sa vybrať aktivity, ktoré by prilákali čo najväčšiu skupinu žiakov s cieľom rozvíjať ich logické, algoritmické a programátorské myslenie, ale aj grafickú a priestorovú predstavivosť.

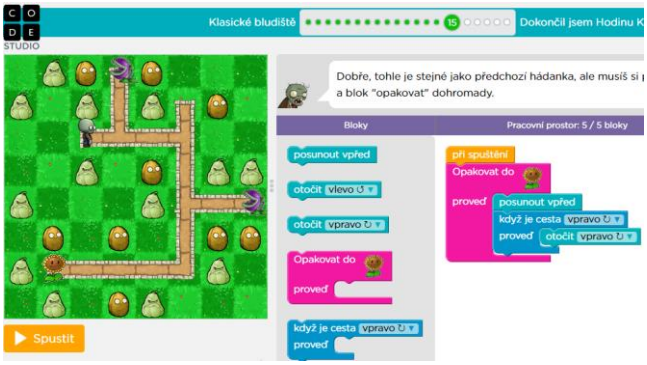
Vyučovacej hodine musela predchádzať príprava, kde sme vybrali kurzy, ktoré boli preložené buď do českého, alebo do slovenského jazyka. Pre 3. a 4. ročník sme dali žiakom na výber medzi Minecraft Designer, Star Wars a Kódovanie s Annou a Elsou. Výber bol ovplyvnený tým, že tieto hry sú v ich veku aktuálne a moderné a chceli sme docieliť aj to, aby žiaci kurzy brali skôr ako zábavné hry. Počas kurzu sme žiakom pomáhali s programovaním. Keďže v týchto ročníkoch je viac chlapcov ako dievčat, najväčší záujem bol o Minecraft Designer. Jeden zo žiakov bol užívateľom učiteľského počítača s možnosťou premietania na dataprojektor, kde žiaci mohli odpozorovať postup a realizáciu niektorých úrovní, ktoré im robili problém. Celý proces bol zahrnutý do predmetu Informatická výchova, ktorá nie je známkovaná, a teda v bežnom vyučovacom procese využívame na motiváciu nálepky, ktoré sú ekvivalentom jednotky. Približne polovica žiakov kurz úspešne absolvovala, ostatným chýbali približne 2 – 3 levely do ukončenia. V tomto prípade sme sa rozhodli, že nebudeme dávať certifikáty, ale nálepky, ktoré sme našli na webstránke www.hourofcode.com.

Obrázok 4 Ukážka hodnotenia pomocou nálepiek v triede 3.A

V prípade 5. a 6. ročníka sme zvážili kurz „Write your first computer program“, ktorá bola určená pre začiatočníkov, ktorí doteraz nemali žiadnu skúsenosť s programovaním a triedy tomuto parametru vyhovovali. Keďže naše vybrané triedy sa ešte vo výučbe nestretli s programovaním, zdali sa nám tieto kurzy pre nich vhodné. Programovanie majú zaradené podľa tematických plánov až neskôr po absolvovaní aktivity „Hour of Code“. Kurz bol vytváraný v prostredí, v ktorom boli dopredu definované bloky príkazov, ktoré žiaci mali ťahaním pridávať do okna, kde vytvárali program. Jednotlivé úrovne sa každou úrovňou stávali zložitejšími. Od príkazov typu choď dopredu k cyklom s n opakovaním až po *when* alebo *if* cykly.

Vybraný kurz bol preložený do českého jazyka, takže žiaci nemali problém s porozumením. Preto mohli pracovať samostatne. Celý proces sme sledovali prostredníctvom prechádzania sa pomedzi žiakov, aby sme zistili aktuálny stav situácie. Niektorým bolo treba poradiť, niektorí intuitívne pochopili napríklad aj cyklus *for*. Žiakom práca išla veľmi dobre, samostatne pracovali a bolo zjavné, že ich to bavilo.

Problém nastal v úrovni 15, s ktorým malo problém asi polovica triedy v oboch ročníkov. Tento príklad sme si spolu vysvetlili, keď sme ho premietli cez dataprojektor. Problém nastal u žiakov pri zvládnutí spojenia nekonečného *while* cyklu s podmienkou vo vnútri.



Obrázok 5 Úroveň 15, s ktorou mali žiaci problém

Traja žiaci v triede prejavili záujem a spýtali sa na ďalšie možnosti, v ktorých by sa mohli rozvíjať. Zaujal ich najmä kurz s prostredím MineCraft a Star Wars. Tvrdili, že by si to rady skúsili aj doma.

Nie všetci žiaci za 45 minút dokončili všetky úrovne, ale nevyskytol sa ani jeden žiak, ktorý by nezvládol minimálne 10 levelov. Na konci hodiny sme žiakov poprosili o to, aby nám vyplnili krátky dotazník o “Hour of Code“, aby sme od nich získali spätnú väzbu. Na webstránke „Hour of Code“ je možnoť využitia „Exit ticket-u“ ako spätnú väzbu pre učiteľov, ale my sme v tomto prípade využili vlastné dotazníky, ktoré mali viac otázok.

Z výsledkov dotazníka nám vyšlo, že 83,33% žiakov zaujala aktivita, ktorú sme na hodine robili a 1,2% žiakom sa aktivita vôbec nepáčila. Avšak až 50% žiakov sa vyjadrilo, že jednotlivé úlohy neboli ťažké, ale dali sa len veľmi ťažko časovo stihnúť. 27,8% žiakov si myslelo, že to bolo ľahké a časovo sa to dalo ľahko zvládnuť.

Záujem o ďalšie možnosti programovať na hodinách informatiky bol značný, lebo až 72,2% by chcelo programovať aj na ďalších hodinách informatiky. No napriek vysokému záujmu, bolo 11,2% z odpovedí, že by radšej robili niečo iné.

Z dotazníka vyplýva, že žiakom sa na aktivite páčilo rôznorodosť hier, grafika stránky i to, že to bolo zábavné a učili sa inou formou, kde získali nové skúsenosti.

5. Záver

Myslíme si, že prvotný dojem z programovania mala väčšina žiakov veľmi dobrý a to je dôležité na to, aby sa v rámci programovania mohli naďalej rozvíjať a boli motivovaní k informatike. Mnoho žiakov sa na nasledujúcej hodine informatiky a informatickej výchovy pýtalo, či budú mať zase Hodinu kódu a ktorý kurz budú mať nabudúce.

Zároveň je zaujímavé porovnať žiakov, ktorí už mali za sebou tento kurz a žiakov, ktorí ho neabsolvovali. Takéto zloženie triedy je na počítačovom krúžku (2. – 4. ročník). Žiaci majú v obľube aktivitu, kde sa na tabuľu napíšu rôzne úlohy (tleskni, otoč sa, vyskoč, pozri sa na spolužiaka) a následne sa niektoré zakrúžkujú a priradia sa k nim čísla, koľkokrát sa budú opakovať (napr. tlesk, 2x(otoč sa, vyskoč), 3x(pozri sa na spolužiaka)). Takto si žiaci precvičujú cykly bez toho, aby sme im vraveli niečo o programovaní. Žiaci, ktorí kurz absolvovali, aktivitu riešia väčšinou bez toho, aby sa pomýlili, ostatným treba správne riešenie často ukázať a až potom sú schopní aktivite porozumieť. Samozrejme, treba vziať do úvahy aj fakt, že medzi žiakmi sú vekové a intelektuálne rozdiely, teda nemôžeme jednoznačne usúdiť, že kurz „Hour of Code“ im pomohol, ale nedá sa to ani vyvrátiť.

BIBLIOGRAFICKÉ ODKAZY

Rozhlas a televízia Slovenska. Dostupné na internete: <http://www.rtvs.sk/televizia/archiv/11580/83491#1532>

[1] Programovanie Kodu na Slovensku. Dostupné na internete: <https://koduslovak.wordpress.com/category/sutaz-kodu-cup-20142015/>

[2] Blog inovatívnych učiteľov a študentov Microsoft – Vzdelávame pre budúcnosť. Dostupné na internete: <http://vzdelavame.azurewebsites.net/kodu/kodu-cup-20162017/>

[3] Fowler, A., Cusack, B.: Enhancing Introductory Programming with Kodu Game Lab: An Explorary Study [online]. 2011. Dostupné na internete: <https://www.kodugamelab.com/discussion/kodu-community/news-and-features/2013/8/introducing-the-kodu-research-series!/>

[4] Fowler, A., Fristce, T., MacLauren, M.: Kodu Game Lab: a programming environment [online]. The Computer Games Journal. 2012 . Dostupné na internete: <http://kodu.blob.core.windows.net/kodu/Kodu Research Series/tcgj_11_whitsun2012_fowlerea.pdf>

[5] KODU CUP 2016/2017 Slovensko. Dostupné na internete: <https://education.microsoft.com/gettrained/kodu-cup-slovensko>

DIDINFO 2017, Univerzita Mateja Bela, Banská Bystrica, 2017.

Záverečný projekt z informatiky pre zrakovo postihnutých žiakov ZŠ – skúsenosti s autentickým učením

Final project in Computer Science for the visually impaired pupils - experience with authentic learning

Ľudmila Jašková
KZVI FMFI UK
Mlynská dolina
842 48 Bratislava
Slovenská republika
jaskova@fmph.uniba.sk

Mária Kaliaková
SŠI pre žiakov so zrakovým postihnutím
Svrčia 6
842 11 Bratislava
Slovenská republika
maria.kaliakova@bee.sk

ABSTRACT

In this paper we describe realization of authentic projects by visually impaired pupils at the end of their compulsory education. We used authentic project as a research tool to determine what informatics skills are these pupils able to use for solving real problems. At the beginning we define the concept authentic learning, further we describe our authentic projects. In conclusion we present overview of our observations and compare them with similar researches.

Keywords

Authentic learning, visual impairment, descriptive research, creativity, informatics competences

ABSTRAKT

V článku popisujeme realizáciu autentických projektov žiakmi so zrakovým postihnutím v záverečnom ročníku základnej školy. Autentický projekt sme použili ako výskumný nástroj s cieľom zistiť, aké informatické zručnosti sú žiaci na konci povinnej školskej dochádzky schopní využiť na riešenie reálnych problémov z praxe. V úvode článku charakterizujeme pojem autentické učenie, ďalej sa venujeme popisu nami realizovaných autentických projektov. Záverom uvádzame prehľad výsledkov našich pozorovaní a porovnávame ich s niektorými podobnými výskumami.

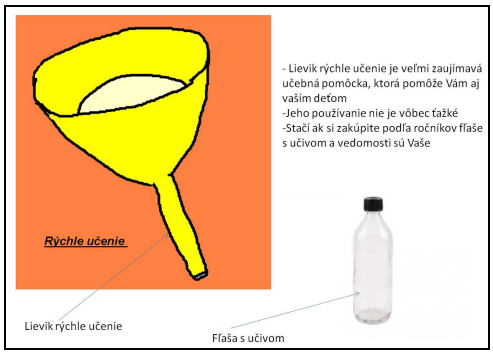
Kľúčové slová

Autentické učenie, zrakové postihnutie, deskriptívny výskum, tvorivosť, informatické kompetencie

1. ÚVOD

Pasch a kolektív [1] charakterizuje **autentické učenie** ako *proces, v ktorom je výučba organizovaná s dôrazom na zmysluplné využitie učebnej látky. Autentické učenie odráža proces, v ktorom ľudstvo počas celej svojej histórie získalo väčšinu zručností a znalostí. Iba v školách (a pri súťažiach) sa informácie cenia pre ne samotné. V bežnom svete sú fakty užitočné preto, že nám pomáhajú pochopiť okolitý svet, riešiť problémy alebo vyjadrovať svoje myšlienky. Autentické učenie umožňuje žiakom vychádzať z odpovedajúceho kontextu a používať získané poznatky zmysluplným spôsobom.*

Autentický projekt sme sa rozhodli realizovať so zrakovo postihnutými žiakmi v záverečnom ročníku základnej školy (viacerí žiaci začínali prípravným ročníkom). Našim cieľom bolo, aby si žiaci precvičili informatické vedomosti a zručnosti získané počas štúdia na základnej škole a vedeli ich využiť na riešenie reálnych problémov, s ktorými sa stretnú vo svojej budúcej profesii. Dôraz sme kládli na zmysluplné využitie digitálnych technológií.



Obrázok 1. Ukážka z prezentácie žiakov o výrobku na rýchle učenie, ktorý navrhli

Digitálne technológie (DT) sú podľa Paperta [2] *presnejším názvom pre informačné a komunikačné technológie* (IKT). Spomínaný autor ich vidí ako *súhrn dvoch rovnako dôležitých zložiek: technológie ako informačného média a technológie ako konštrukčného média.*

Podľa Resnicka výskum potvrdil [3], že *najkvalitnejšie poznávanie zvyčajne zažívame vo chvíľach, keď sme zaujatí návrhom a tvorbou, a to najmä vtedy, ak sa jedná o vytváranie zmysluplných vecí – zmysluplných buď pre nás alebo pre naše okolie.*

Dieťa sa učí tým, že vytvára čosi nové, podľa svojho uváženia, ak sa aktívne zapája do tvorby v skupine, ak využíva rôzne materiály a médiá, ak vytvára nejaký produkt. Vytvorenie produktu dieťa plánuje a realizuje, premýšľa o ňom a používa ho spolu s druhými. Jedným z najefektívnejších médií na konštrukcionistické učenie sa je podľa Paperta *počítač* [4] v [5].

Aj pre zrakovo postihnutých žiakov je dôležité, aby sa učili konštrukcionistickým spôsobom v zmysle Paperta [6]. Podľa neho je ústrednou myšlienkou konštrukcionizmu to, že *študenti sa učia efektívnejšie, ak môžu riešiť vlastné úlohy, čím si budujú svoje znalosti sami a vyjadrujú myšlienky cez médium, ktoré umožňuje získať priame skúsenosti*. Realizovali sme dva typy autentických projektov, ktoré sme nazvali **Moja firma** a **Prieskum médií**. Podrobnejšie ich popíšeme v nasledujúcej kapitole.

2. VZDELÁVACÍ SCENÁR

Projekt **Moja firma** bol určený pre vidiacich a slabozrakých žiakov, pretože v rámci neho mali žiaci pracovať aj s grafickým editorom a editorom videí, ktoré v súčasnosti nie sú vhodné pre nevidiacich. Nevidiaci žiaci realizovali projekt **Prieskum médií** vyžadujúci použitie programov, ktoré nie sú pre nich problematické (textový editor, e-mail, tabuľkový kalkulátor a prezentačný softvér).

2.1 Projekt Moja firma

V rámci projektu mali žiaci v malých skupinách alebo jednotlivo „založiť“ fiktívnu firmu, zvoliť oblasť jej podnikania a navrhnúť produkt, ktorý bude potenciálnym zákazníkom ponúkať. Mohli vymyslieť produkt, ktorý v súčasnosti na trhu chýba, napríklad technologickú novinku, zázračný kozmetický výrobok alebo liek na nevyliečiteľnú chorobu. Žiaci teda mohli využiť svoju fantáziu a tvorivosť. Navrhnutý produkt mali nakresliť v grafickom editore. Okrem toho mali navrhnúť vhodný názov firmy a produktu, tiež aj logo a výstižný slogan firmy. Ďalej mali vytvoriť reklamu na svoj produkt vo forme videa alebo animácie, vytvoriť webovú stránku firmy a prezentáciu o svojom produkte.

Časová dotácia pre realizáciu projektu bola osem vyučovacích hodín. Žiaci však mohli na ňom pracovať aj doma. Plánovaná náplň jednotlivých hodín bola nasledujúca.

- Oboznámenie sa so zadaním projektu a spôsobom jeho hodnotenia. **Nákres výrobku** a loga firmy **v grafickom editore** (1 vyučovacia hodina).
- **Tvorba sloganu a scenára** reklamy na výrobok **v textovom editore** (1 vyučovacia hodina).
- **Tvorba reklamy** vo forme **animácie** alebo **videa** (3 vyučovacie hodiny).
- **Tvorba webovej stránky** firmy (1 vyučovacia hodina).
- **Tvorba prezentácie** (1 vyučovacia hodina).
- Prezentovanie výsledkov projektu a jeho hodnotenie (1 vyučovacia hodina).

Obrázok 2. Ukážka prezentácie o výrobku nevybiteľná baterka

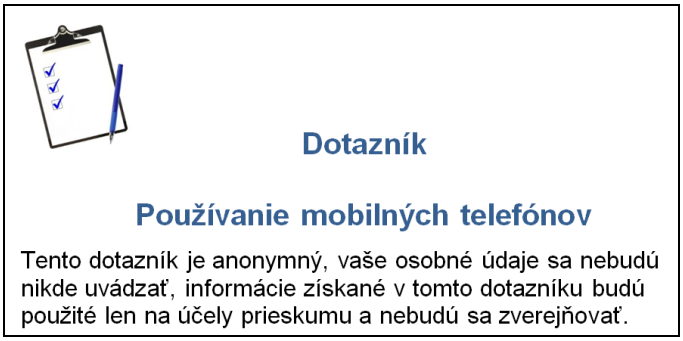
2.2 Projekt Prieskum médií

V rámci projektu **Prieskum médií** mali žiaci realizovať deskriptívny výskum. Podľa Pasch a kolektív [1] je **cieľom desktiptívneho výskumu popis skutočnosti**. Ide o čo najpresnejšie a najsystematickejšie zmapovanie súčasnej situácie.

V rámci projektu mali žiaci zistiť konkrétne informácie od desiatich náhodne zvolených respondentov týkajúce sa používania rôznych médií – rozhlasu, televízie, internetu, tlače, mobilnej siete a podobne. V rámci prieskumu mali žiaci zisťovať, ako respondenti používajú zvolené médium – za akým účelom, ako často, aké sú finančné náklady s tým spojené a podobne. Najskôr mali žiaci v textovom editore pripraviť dotazník, ktorý e-mailom rozoslali respondentom. Potom mali na základe získaných údajov spracovať dáta pomocou tabuliek, vytvoriť grafy a výsledky prieskumu prezentovať pomocou elektronickej prezentácie.

Aj tento projekt sme časovo naplánovali na osem vyučovacích hodín, pričom žiaci mohli pracovať na projekte aj doma. Plánovaná náplň jednotlivých hodín bola nasledujúca.

- Oboznámenie sa so zadaním projektu a spôsobom jeho hodnotenia. Voľba média, ktorého sa bude prieskum týkať (1 vyučovacia hodina).
- Tvorba **dotazníka** (obrázok 3) **v textovom editore** a jeho **rozoslanie e-mailom** (1 vyučovacia hodina).
- **Vypĺňanie tabuliek** podľa získaných dát (2 vyučovacie hodiny).
- **Tvorba grafov** (1 vyučovacia hodina).
- **Tvorba prezentácie** (2 vyučovacie hodiny).
- Prezentovanie výsledkov projektu a jeho hodnotenie (1 vyučovacia hodina).



Obrázok 3. Úvod dotazníka zameraného na použitie mobilných telefónov

V obidvoch projektoch sme požadovali, aby na konci žiaci odovzdali všetky vytvorené elektronické dáta a výstupy.

3. Realizácia výskumu

Náš výskum sme realizovali so žiakmi deviateho ročníka na základnej škole pre slabozrakých a nevidiacich. Použili sme metódu výskumu vývojom [7]. Vo viacerých iteráciách sme vyvíjali optimálne zadania a spôsoby hodnotenia projektov a analyzovali sme výsledné produkty žiakov a poznámky zo zúčastneného pozorovania. Našim cieľom bolo zistiť, aké informatické zručnosti sú žiaci na konci povinnej školskej dochádzky schopní využiť na riešenie reálnych problémov z praxe. Preto sme projekt realizovali ako záverečný projekt z informatiky počas posledných dvoch mesiacov školského roka. Ďalej podrobnejšie popíšeme realizáciu obidvoch projektov.

3.1 Realizácia projektu Moja firma

Projekt **Moja firma** sme realizovali trikrát (v rokoch 2013, 2014 a 2016), zakaždým s inou skupinou približne desiatich žiakov.

V roku 2013 boli v skupine prevažne vidiaci žiaci, iba jeden žiak mal ťažšie zrakové postihnutie a používal zväčšovací softvér. Žiaci pracovali vo dvojiciach. Volili si nasledujúce témy – rýchle učenie – zázračný prípravok umožňujúci rýchlo sa naučiť čokoľvek (obrázok 1), zázračná zubná pasta, „šup-šup botky“ – topánky umožňujúce rýchlu chôdzu, nápoj na zvýšenie fyzickej výkonnosti, zázračné krmivo pre psov – krmivo, ktoré zaručí, že pes bude zdravý a múdry.

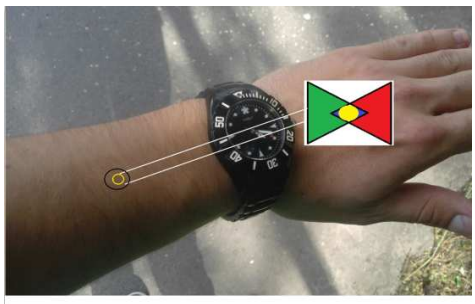
Počas tejto iterácie sme pozorovali, že žiaci mali problém efektívne pracovať v skupine. Nevedeli si rozdeliť úlohy a všetko robili spoločne. Tiež mali problém uložiť si výstupy vytvorené v rámci dvojice tak, aby ich dokázali obaja zdieľať. Často sa stávalo, že ak jeden žiak z dvojice chýbal, druhý nemal prístup k tomu, čo na predchádzajúcej hodine vytvorili a uložili na spolužiakovom konte. Tiež sme pozorovali jav, že žiaci v prípade spolužiakovej neprítomnosti neboli ochotní pracovať samostatne.

Len jedna dvojica po dokončení projektu odovzdala všetky výstupy. Väčšina dvojíc nemala všetky dáta uložené na jednom mieste. Niečo mali uložené na svojich osobných notebookoch alebo mobilných telefónoch. Najväčším problémom bolo zálohovanie videa. Mnohé výstupy žiaci dokonca zmazali hneď potom, ako ich bezprostredne po dokončení ukázali učiteľovi.

V roku 2014 projekt realizovalo 50% vidiacich žiakov a 50% ťažko zrakovo postihnutých. Pracovali prevažne vo dvojiciach, iba jeden žiak pracoval sám. V jednej dvojici boli dvaja ťažko zrakovo postihnutí žiaci, vo väčšine dvojíc bola kombinácia vidiaceho žiaka s ťažko zrakovo postihnutým. Z hľadiska tém, si žiaci zvolili nasledujúce: liek pre psov, liek proti rakovine, super rýchle tenisky, zázračný samostatne chodiaci batoh, zdravý hamburger. Počas tejto iterácie sme pozorovali nasledujúce skutočnosti.

- Žiaci mali problém vymyslieť, čo bude ich firma vyrábať. Spočiatku sa rozhodovali pre už existujúce produkty.
- Mnohí žiaci nevedeli, čo je slogan. Význam pojmu pochopili až keď sme im odporučili vyhľadať niekoľko sloganov na webe.
- Žiaci nevedeli napísať dostatočne podrobný scenár pre reklamu. Spočiatku nemali jasnú predstavu o tom, čo bude na videu. Tvorba scenára im pomohla všetko si detailne premyslieť. Po našich výzvach doplnili do scenára podrobné informácie o rekvizitách, ktoré plánovali použiť a aj dialógy postáv. Žiaci si s našou pomocou uvedomili, že video by malo názorne ilustrovať prínos ich výrobku. To znamená, že v úvode je potrebné diváka zoznámiť s problémom, ktorý sa v závere jednoducho vyrieši vďaka ich výrobku.
- Pri tvorbe webovej stránky firmy mali žiaci použiť WYSIWYG editor a mali k dispozícii rozpracované súbory, do ktorých stačilo doplniť konkrétne texty a obrázky. Jeden žiak mal však zvýšený záujem o tvorbu webových stránok a počas projektu sa samostatne naučil HTML a CSS. Všetky súbory si vytvoril sám a tiež navrhol veľmi pôsobivý dizajn firemnej stránky.
- Najväčšie problémy mali žiaci s tvorbou videa, webovej stránky a podobne ako v predošlom roku, aj so zálohovaním a zdieľaním dát v skupine. V niektorých dvojiciach sme postrehli nepomer v deľbe práce – jeden žiak pracoval výrazne viac ako druhý.

V roku 2016 projekt realizovali prevažne vidiaci žiaci, iba dvaja mali ťažké zrakové postihnutie. Všetci žiaci sa rozhodli pracovať samostatne. V porovnaní s predchádzajúcimi rokmi sa pri výbere tém výraznejšie prejavila tvorivosť a invenčnosť. Žiaci si volili neexistujúce produkty, dokonca pomerne presne popisovali princíp ich fungovania. Originálny bol produkt Invisible Music – samolepka, ktorá po aplikácii spustí v mozgu prehrávanie vhodnej hudby podľa aktuálnej nálady (obrázok 4).



Samolepka hudby podľa nálady

Samolep M25:
- Čip Play, 2 Bluethooth wave, Brain Memory
- Čip Play:
- hlavný zdroj čip prepojený cez nervovú sústavu
- pomocou podnetu, reflexu a nálady sa šíri zvuk do mozgu
- (250-320 Hz)
- 2 bluethoot wave:
- prvá časť sa používa na vnem hudby cez Brain Memory, do Čip Play
- druhá časť dokáže zdieľať pocity druhým
- Brain Memory:
- jadro, ktoré sa musí aktivovať pomocou laseru
- načíta zvuk cez mozog
- podľa nálady vyberie zvuk, ktorý je uložený v databázach piesní
- prenesie databázu cez Bluethoot wave do Čip Play

Obrázok 4. Ukážka prezentácie o výrobku Samolepka hudby podľa nálady

Zaujímavé výrobky boli aj hovoriaci puk s čipom ovládaný hlasom, lietajúce auto, nevybiteľná baterka (obrázok 2), náramok s teleportujúcimi schopnosťami – pošepkaním sa udá miesto a okamžite sa tam ocitneme (obrázok 5). Zaujali nás aj produkty Love Potion – elixír lásky a omladzovací krém „Mladšy a krajší“.

Obrázok 5. Ukážka prezentácie o výrobku teleportujúci náramok

V priebehu tejto iterácie sme pozorovali nasledujúce problémy.

- Žiaci mali spočiatku problém vymyslieť si, aký produkt navrhnú. Napriek tomu mali nakoniec originálne nápady a na projekte pracovali s nadšením. Radostnú atmosféru sme pozorovali aj pri záverečnom prezentovaní produktov. Viacerí žiaci pôsobili suverénnym dojmom a boli hrdí na výsledky svojej práce.

– Podobne ako po minulé roky, opäť sme si všimli, že žiaci nepoznajú pojmy logo a slogan. Počas projektu však pochopili ich význam.

– Žiaci mali na konci projektu pocit, že sa počas neho nenaučili nič nové, iba si zopakovali a precvičili, čo už vedeli. Neuvedomovali si, že sa naučili použiť vedomosti a zručnosti na riešenie reálnych problémov.

3.2 Realizácia projektu Prieskum médií

Projekt **Prieskum médií** sme realizovali dvakrát. Prvý raz v roku 2013 s piatimi ťažko slabozrakými žiakmi. Každý z nich realizoval prieskum týkajúci sa iného média – rozhlasu, televízie, tlače, mobilných zariadení a počítača.

Žiaci mali realizovať prieskum s 20 respondentmi. Z hľadiska časových nárokov potrebných na spracovanie a vyhodnotenie získaných dát sa ukázalo, že to je príliš veľká vzorka. Cieľom totiž nebolo, aby žiaci získali čo najhodnovernejšie výsledky, ale aby si precvičili prácu s digitálnymi technológiami pri realizácii prieskumu. V ďalšej iterácii sme sa rozhodli znížiť počet respondentov na 10.

Niektorí žiaci mali v dotazníku príliš veľa otázok, čo tiež zvyšovalo prácnosť pri vyhodnocovaní prieskumu. V ďalšej iterácii sme plánovali obmedziť počet otázok na 10.

Všimli sme si, že žiaci pri vyhodnocovaní dát v tabuľkách nevyužívali vzorce, ale výsledky do sumarizačných buniek zadávali ručne potom, ako ich získali ručným vykonávaním aritmetických operácií. Pochopili sme, že napriek tomu, že sa žiaci učili používať v tabuľkách jednoduché vzorce, toto učivo nevedeli aplikovať pri riešení reálnych problémov.

Pri práci s tabuľkami sme pozorovali tiež, že jeden žiak si vytvoril graf, aby ho mohol použiť v prezentácii výsledkov. Graf v tabuľkovom procesore však zmazal hneď potom, ako ho skopíroval do prezentácie. Neuvedomoval si, že v prípade modifikácie údajov bude musieť graf vytvárať nanovo.

Problémy s prácou v tabuľkovom procesore chceli dvaja žiaci eliminovať tak, že vyhodnocovali dáta priamo v prezentácii a tabuľkový procesor vôbec nepoužili.

V prezentáciách žiaci často uvádzali grafy bez sprievodných textov.

V roku 2016 projekt realizovali dve nevidiace žiačky. Jedna si zvolila prieskum v oblasti použitia televízie a druhá v oblasti mobilných telefónov. Žiačky pracovali na projekte aj doma, pričom na hodine sme im podrobne vysvetlili, čo majú doma robiť a žiačky si v škole nacvičili požadované činnosti. V priebehu tejto iterácie sme pozorovali viaceré zaujímavé skutočnosti.

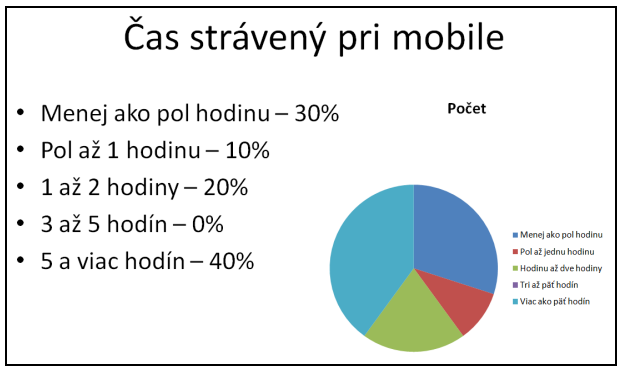
Žiaci až pri vypĺňaní tabuliek zistili, že niektoré otázky v dotazníku neboli dobre sformulované. Dôsledkom bolo, že získané odpovede boli nejasné a kategórie odpovedí boli iné, ako žiaci očakávali. Táto skutočnosť priviedla žiakov k záveru, že je dôležité v otázkach dotazníkov uvádzať aj očakávané alternatívy odpovedí, aby ich následne dokázali správne spracovať. Obmedzenie počtu otázok na 10 sa ukázalo ako optimálne. Občas sa však stalo, že dve rôzne otázky sa zameriavali na zistenie rovnakých informácií.

Žiaci zistili, že v tabuľkách je dôležité správne určiť kategórie pre vek, časové údaje a podobne. Ďalej sa naučili, že pri vypĺňaní tabuliek je dôležité vytvoriť si priečinok, do ktorého budú ukladať súbory s už spracovanými dotazníkmi. Dôležité tiež bolo kontrolovať, či súčet odpovedí pre jednotlivé otázky je zhodný s počtom respondentov. Ukázalo sa, že kvôli jednoduchšej orientácii je rozumné číslovať tabuľky podľa otázok v dotazníku.

Podľa pôvodného plánu mali žiaci najprv vytvárať grafy v tabuľkovom editore a až na ďalšej hodine ich mali kopírovať do prezentácie. Ukázalo sa však, že výhodnejšie je najprv vytvárať prezentáciu a až potom grafy, ktoré treba hneď skopírovať do prezentácie. Z hľadiska typu väčšinou používali koláčový graf, len pri otázkach, v ktorých si respondenti mohli vybrať viac možností odpovede, použili stĺpcový graf.

Pri tvorbe prezentácií sa žiaci naučili zmeniť rozloženie snímky na Nadpis, Obsah, Obsah (obrázok 6). Požadovali sme, aby na minimálne piatich snímkach bol graf, ale žiaci ho dávali na každú snímku. Viedli sme ich k tomu, aby dbali na štylistickú a jazykovú správnosť textu. Žiakom sme odporučili, aby údaje o respondentoch uvádzali v percentách, prípadne aby jasne uviedli, kedy hodnoty znamenajú počet osôb a kedy počet percent.

Pri prezentovaní výsledkov žiaci žiaľ nemohli použiť slúchadlá a čítač obrazovky. Nemali tak žiadnu spätnú väzbu z počítača a museli si pamätať, kde sa v prezentácii nachádzajú. Aj napriek týmto sťaženým podmienkam to zvládli na veľmi dobrej úrovni. Tieto sťažené podmienky však mali za následok zvýšený stres žiakov a prezentácia výsledkov bola pre ich spolužiakov pomerne nudná. Pútavosť by bolo zvýšilo, ak by na úvod pridali jednu snímku s cieľmi prieskumu, prípadne s hypotézami a na záver jednu snímku so závermi – zhrnutie výsledkov a ich porovnanie s hypotézami. Tiež by bolo vhodné pridať aj iné obrázky ako len grafy.



Obrázok 6. Ukážka snímky z prezentácie o prieskume používania mobilných telefónov

4. Závery a diskusia

Podľa našich dlhoročných skúseností sa so žiakmi deviateho ročníka po absolvovaní testovania Monitor 9 a prijímacích skúšok na strednú školu veľmi ťažko pracuje. Nie sú dostatočne motivovaní učiť sa nové veci. Opakovanie prebratého učiva vo forme autentického projektu sa ukázalo ako vhodná a dostatočne motivujúca aktivita. Jej priebeh však narúšalo časté odpadávanie hodín informatiky kvôli iným akciám, prípadne absencii žiakov zo zdravotných dôvodov. Väčšina žiakov nebola zvyknutá v prípade absencie dokončiť úlohy doma. Treba však zobrať do úvahy, že viacerí žiaci nemali doma potrebné hardvérové, či softvérové vybavenie.

Ako sme uviedli v predchádzajúcich kapitolách, ukázalo sa, že žiaci nemajú potrebné zručnosti na prácu s tabuľkami a s videom. Tiež v prezentáciách mali viacerí mnoho jazykových chýb a mali problém prezentovať spamäti. Veľkým problémom bolo aj

priebežné a systematické ukladanie dát a ich zdieľanie so spolužiakmi a učiteľom.

Daskolia a kolektív [8] popisuje výskumný projekt, v rámci ktorého študenti pedagogiky vytvárali v skupinách digitálny príbeh na tému udržateľnosti života v mestách. Sledovali aký typ technológií študenti použijú a aký bude proces a výsledok ich tvorby. Autori zdôrazňujú, že sa im osvedčil ucelený súbor presne definovaných míľnikov, a tiež považovali za dôležité dokumentovať vývoj myšlienok, ktoré viedli k tvorbe konkrétnych predmetov či artefaktov. Aj my sme zistili, že je dôležité presne definovať termíny pre dokončenie jednotlivých etáp projektu a na ich konci ohodnotiť výsledné produkty činnosti. Okrem toho je potrebné dôsledne od žiakov vyžadovať, aby ohodnotené dáta uložili na presne stanovené miesto.

Zaujala nás publikácia [9], v ktorej Šnajder a kolektív popisujú návrh, implementáciu a overenie bádateľsky orientovaných metodík v informatike. Podľa nej má mať priebeh bádateľsky orientovaného vyučovania zameraného na **získavanie, spracovanie a prezentáciu informácií** nasledujúce fázy.

1. Stanovenie výskumnej oblasti a výskumného problému.
2. Formulácia výskumných otázok a hypotéz.
3. Tvorba dotazníka a zber dát.
4. Spracovanie dát s využitím základných štatistických charakteristík súboru dát, ale aj zisťovanie súvislostí.
5. Vytvorenie záverečnej správy.

V našom prieskume sa žiaci nevenovali formulácii hypotéz, tiež sa nezaoberali zisťovaním súvislostí a chýbalo aj zhrnutie výsledkov, hoci tento aspekt by sme ocenili. Ako uvádzajú vyššie uvedení autori, pozorovali, že *závery výskumov prezentované žiakmi boli často len prerozprávaním grafov, ktoré zobrazovali absolútne alebo relatívne početnosti odpovedí na jednotlivé otázky z dotazníkov.* Podobnú skúsenosť s prezentovaním výsledkov sme mali aj my.

Na rozdiel od vyššie popísaného výskumu, naši žiaci realizovali deskriptívny prieskum a nie bádateľsky orientovaný výskum. Ďalší rozdiel je, že našim cieľom bolo precvičenie potrebných informatických a nie bádateľských zručností. Na tieto zručnosti by sme kládli dôraz až u starších žiakov – vo vyššom sekundárnom vzdelávaní.

5. BIBLIOGRAFICKÉ ODKAZY

[1] PASCH, M. et al. *Od vzdělávacího programu k vyučovací hodině*. Praha : Portál, 1998. ISBN 80-7178-127-4.

[2] PAPERT, S. What is Logo? Who Needs It? In: PAPERT, S. et al. *Logo Philosophy and Implementation*. MicroWorlds and LCSI, 1999, pp. IV-XVI. ISBN 2-89371-494-3.

[3] RESNICK, M.: Rethinking Learning in the Digital Age. In: KIRKMAN, G. (ed.): *The Global Information Technology Report: Readiness for the Networked World*, Oxford University Press, 2002.

[4] PAPERT, S. – HAREL, I. Situating Constructionism. In: Papert, S. – Harel, I. (Eds.) *Constructionism*. Norwood, NJ : Ablex Publishing, 1991. ISBN 0893917869. Dostupné na internete: http://www.papert.org/articles/SituatingConstructionism.html

[5] KALAŠ, I. et al. *Premeny školy v digitálnom veku*, Bratislava : SPN, Mladé letá, s.r.o., 2013. ISBN 978-80-10-02409-4.

[6] PAPERT, S. *Mindstorms: Children, computers, and powerful ideas*. Second Edition. New York : Basic Books, 1980.

[7] KALAŠ, I. Pedagogický výskum v informatike a informatizácii (2. časť). In: Rovan, B. (Eds.) *Didinfo 2009*. Banská Bystrica : Univerzita Mateja Bela, 2009. ISBN 978-80-8083-720-4.

[8] DASKOLIA. M. et al. Fostering collaborative creativity in learning about urban sustainability through digital storytelling. In: Futschek, G. et al.: *Constructionism and Creativity 2014*, Proceedings of the 3rd internatiobal constructionism conference, Vienna, Gesellschaft : Östereichische Computer, 2014. ISBN 978-3-85403-301-1.

[9] ŠNAJDER, Ľ. et al. Bádateľsky orientované vyučovanie informatiky – priebežné výsledky pedagogického výskumu In: TRAJTEĽ, Ľ. (Eds.) *Didinfo 2016*, Banská Bystrica : Univerzita Mateja Bela, 2016. ISBN 978-80-557-1082-2

DIDINFO 2017, Univerzita Mateja Bela, Banská Bystrica, 2017.

Učebné materiály pre výučbu informatiky nevidiacich žiakov základnej školy

Educational materials for teaching informatics blind elementary school students

Ľudmila Jašková
KZVI FMFI UK
Mlynská dolina
842 48 Bratislava
Slovenská republika
jaskova@fmph.uniba.sk

Mária Stankovičová
Centrum podpory študentov so špecifickými potrebami UK
Mlynská dolina
842 48 Bratislava
Slovenská republika
stankovicova@cezap.sk

ABSTRACT

This paper deals about teaching materials in computer science designed for blind elementary school students. At the beginning we describe our KEGA grant aimed at production of such materials. The second chapter provides an overview of the available computing textbooks for primary and lower secondary education. In the third chapter we briefly mention how blind pupils use computer and substantiate the need for the creation of teaching materials specially designed for this category of pupils. Further we offer the detailed description of the form and contents of teaching materials developed by the project. In conclusion, we describe our planed research aimed at verifing the suitability of the material formed.

Keywords

Informatics education, blind, learning material, programming, accessibility

ABSTRAKT

V tomto článku sa venujeme problematike tvorby učebných materiálov z informatiky určených pre nevidiacich žiakov základnej školy. V úvode článku uvádzame základné informácie o projekte financovanom z fondov agentúry KEGA zameranom na tvorbu takýchto materiálov, ktorého riešiteľmi sú autori tohto článku. V druhej kapitole podávame prehľad dostupných učebníc informatiky pre základné školy. V tretej kapitole sa stručne zmieňujeme o spôsobe práce nevidiacich s počítačom a zdôvodňujeme potrebu tvorby učebných materiálov špeciálne určených pre túto kategóriu žiakov. Ďalej podrobnejšie popisujeme formu a obsah jednotlivých učebných materiálov, ktoré vytvárame v rámci projektu. V závere popisujeme naše plány zamerané na výskumné overenie vhodnosti vytvorených materiálov.

Kľúčové slová

Výučba informatiky, nevidiaci, učebné materiály, programovanie, ptístupnosť

1. ÚVOD

Od roku 2008 prebieha v slovenskom školstve reforma, ktorá zaviedla informatické predmety na prvý aj druhý stupeň základných škôl, špeciálne školy nevynímajúc. V súčasnosti však stále neexistujú učebnice informatiky vhodné pre výučbu nevidiacich žiakov. Od roku 2015 sa v rámci projektu financovaného z agentúry KEGA zameriavame na odstránenie tohto nedostatku. Riešiteľmi projektu sú pracovníci Katedry základov a vyučovania informatiky FMFI UK a Centra podpory študentov so špecifickými potrebami na UK v Bratislave. Spolupracujúcou inštitúciou je aj Základná škola pre slabozrakých a nevidiacich v Bratislave.

2. Učebnice informatiky pre základné školy

Šnajder a kolektív [1] uvádzajú, že „význačným medzníkom v rozvoji vyučovania informatiky bol rok 2000, kedy sa predmet **informatika a výpočtová technika** transformoval na všeobecnovzdelávací predmet **informatika**“. Odvtedy badať úsilie, aby sa predmet informatika stále viac vzďaľoval od zúženej koncepcie používania počítača ku komplexnejšej koncepcii porozumenia. Informatika postupne získala **modernú obsahovú náplň**, ktorá má **budovať informatickú kultúru** a **poskytnúť základné informatické vzdelanie**. Má tiež primeraným spôsobom **pomôcť procesu informatizácie všetkých predmetov**. Pre druhý stupeň základnej školy vznikla kolekcia učebníc pod názvom **Tvorivá informatika** (obrázok 1). Do škôl sa dostali: **1. zošit o obrázkoch** [2], **1. zošit z programovania** [3], **1. zošit s internetom** [4], **1. zošit o práci s textom** [5], **Informatika okolo nás** [6], neskôr aj **1. zošit o číslach a tabuľkách** [7], **2. zošit s internetom** [8] a **1. zošit o prezentáciách** [9]. Tematické zošity vychádzali aj v nasledujúcich rokoch v niekoľkých reedíciách.

Obrázok 1. Niektoré tematické zošity z informatiky pre žiakov základných škôl

V rokoch 2010 až 2013 vznikli aj učebnice informatickej výchovy pre prvý stupeň ZŠ [10], [11], [12], [13].

Mnohé uvedené učebnice sú pre učiteľov bežných žiakov dobrou pomôckou, ale nie sú použiteľné pre učiteľov nevidiacich žiakov.

3. Učebné materiály pre nevidiacich žiakov

Spôsob práce nevidiacich s počítačom je diametrálne odlišný od spôsobu práce bežných používateľov. Nevidiaci používatelia nepoužívajú myš, všetky príkazy zadávajú iba pomocou klávesnice. Na interpretáciu obsahu obrazovky im slúži čítač

obrazovky, ktorý neumožňuje získať okamžitý prehľad o obsahu obrazovky, ako je to v prípade vidiacich používateľov.

Odlišný spôsob použitia počítača si vyžaduje aj odlišný spôsob učenia informatiky. Situácia je komplikovaná tým, že v súčasnosti nie sú na Slovensku kvalifikovaní učitelia informatiky pre nevidiacich žiakov. Bežní učitelia informatiky nemajú skúsenosti s výučbou nevidiacich žiakov a učitelia zo špeciálnych škôl nemajú skúsenosti s vyučovaním informatiky. Pre obidve skupiny sú kvalitné učebné materiály veľmi žiadané. Znalosť práce s počítačom je totiž pre nevidiacich žiakov nevyhnutná, aby sa mohli neskôr vzdelávať na vyšších stupňoch škôl, aby dokázali ľahšie komunikovať s inými ľuďmi a uspieť na trhu práce. Informatika je pre nich jeden z kľúčových predmetov.

V rámci projektu sa zameriavame na nasledujúce oblasti informatiky:

- **Práca s textom** – čítanie textu a schopnosť vytvoriť gramaticky správny a dobre štruktúrovaný textový dokument patrí k základnej gramotnosti.
- **Práca so zvukom** – nevidiaci ľudia žijú prevažne vo svete zvukov. Zvuky majú asociované s mnohými objektmi, pocitmi a situáciami. Pri čítaní textu z obrazovky počúvajú syntetickú reč, ktorá nie je natoľko prirodzená ako sú nahraté zvuky. Porozumieť syntetickej reči je náročné a únavné. Aktivity, ktoré im umožnia počúvať prirodzene znejúce zvuky sú pre nevidiacich príjemným oživením. Tieto aktivity podnecujú ich umeleckú tvorivosť a sú motivačne príťažlivé.
- **Algoritmy a riešenie problémov** – rozvoj schopnosti vykonávať činnosť podľa návodu, prípadne rozčleniť problém na čiastkové problémy a vedieť popísať postup na ich riešenie, to sú ďalšie dôležité schopnosti potrebné pre štúdium a úspešný život.
- **Práca s matematickým textom** – schopnosť rozumieť a vedieť zapísať matematické formuly je kľúčovou pre porozumenie matematických pojmov. Väčšinu úloh vo vyšších ročníkoch základnej školy nie je možné riešiť bez toho, aby žiaci dokázali používať matematické formuly. Braillova matematická notácia je však odlišná od bežného matematického zápisu, čím môže vzniknúť bariéra na vyšších stupňoch vzdelávania, ak učiteľ matematiky neovláda Braillovu matematickú notáciu. Riešením je vhodný matematický softvér Lambda, ktorý by mali vedieť používať nevidiaci žiaci, ak plánujú ďalšie štúdium.
- **Informácie a komunikácia** – vyhľadávanie informácií a ich následné použitie je dôležité pri vyučovaní rôznych predmetov, ako aj pre život v modernej spoločnosti. To isté možno konštatovať aj o schopnosti efektívnej komunikácie so spolužiakmi a učiteľmi pomocou digitálnych technológií.

Pri tvorbe učebných materiálov vychádzame zo skúseností, ktoré sme získali počas realizácie niekoľkoročného výskumu na základnej škole pre nevidiacich a slabozrakých [14]. Okrem toho chceme žiakom umožniť, aby sa učili **konštrukcionistickým spôsobom** v zmysle Paperta [15]. Konštrukcionizmus znamená dať žiakom dobré úlohy, témy, problémy, na ktorých môžu so záujmom pracovať a ktoré im umožnia lepšie sa učiť. Snažíme sa poskytnúť gradované úlohy, ktoré môže každý žiak riešiť vlastným tempom. Okrem samotných úloh sú súčasťou materiálu aj dátové súbory, s ktorými žiaci pracujú (modifikujú ich, dopĺňajú, odstraňujú chyby a podobne). Za kľúčové považujeme vytváranie úloh **s ohľadom na kognitívne schopnosti nevidiacich žiakov** [16]. Uvedomujeme si, že pre kvalitnejšie osvojenie si vedomostí a vytvorenie predstáv je dôležité zapájať aj hmatové vnímanie. Súčasťou materiálov sú preto aj reliéfne obrázky (napríklad ilustrujúce vzhľad rôznych formátov písma, vzhľad tabuľky, vzhľad niektorých okien a podobne).

Všetky vytvorené učebné materiály budú voľne dostupné na webovom portáli *vin.edi.fmph.uniba.sk* [17], aby si ich mohli stiahnuť a použiť učitelia informatiky zo špeciálnych aj bežných škôl, na ktorých sa nevidiaci žiaci vzdelávajú integrovane.

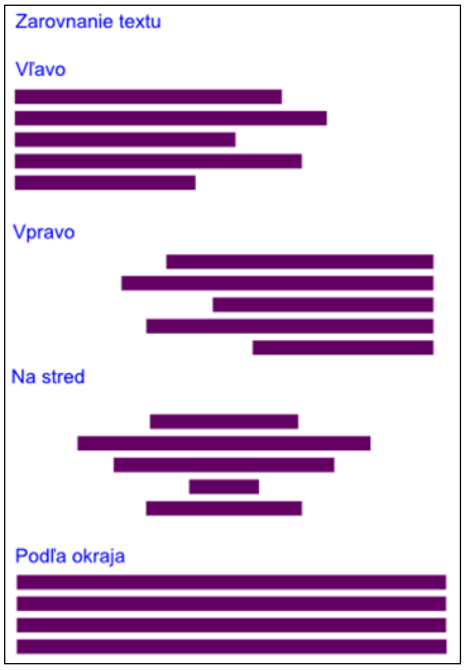
V súčasnosti sa tam nachádzajú dokončené učebné materiály pre tematické okruhy **Práca s textom** a **Práca so zvukom**. Ďalšie materiály budeme postupne dokončovať podľa harmonogramu uvedeného na stránke portálu.

3.1 Práca s textom

Učebný materiál je **určený pre žiakov, ktorí ovládajú** písanie na klávesnici desiatimi prstami, vedia písať čísla a vedia používať niektoré špeciálne klávesy (napríklad Enter, Delete, Backspace a podobne).

Cieľom je, aby sa žiaci zdokonalili v editovaní textu, naučili sa formátovať text, naučili sa vkladať do textu obrázky a tabuľky, vedeli nastavovať niektoré vlastnosti strany a spoznali základné typografické princípy.

Učebný materiál obsahuje takmer 60 úloh. Súčasťou materiálu je niekoľko desiatok dátových súborov, ktoré majú žiaci použiť pri riešení týchto úloh. Niektoré úlohy sú určené pre fázu získavania zručností a skúseností. Súčasťou takýchto úloh sú aj vysvetlenia základných pojmov, klávesových skratiek a postupov typu krok za krokom. Tieto sú viazané na prostredie editora MS Word, verzia 2007 a vyššia. Okrem toho materiál obsahuje aj úlohy určené na precvičenie a automatizáciu zručností. Takéto úlohy sú najmä v poslednej kapitole, ktorá obsahuje námety na rozsiahlejšie projekty.



Obrázok 2. Tlačená predloha pre reliéfny obrázok ilustrujúci zarovnanie textu

Prílohou materiálu je aj **séria reliéfnych obrázkov (**obrázok 2) ilustrujúca rôzne formátovanie písma, odsekov a textu. Tieto obrázky je potrebné vytlačiť na Braillovej tlačiarni Tiger.

V tabuľke 1 uvádzame jednu úlohu aj s nápoveďou, aby sme ilustrovali rozdiel medzi učebným textom určeným pre vidiacich a nevidiacich žiakov. Všimnime si, že zadanie úlohy je detailne vysvetlené v texte a nápoveď obsahuje informáciu o tom, ako treba požadované úkony vykonať pomocou klávesnice. Ak by bol materiál určený pre vidiaceho žiaka, obsahoval by menej textu a bol by doplnený obrázkom. Nápoveď by obsahovala informáciu o tom, ako pracovať s textom pomocou myši a tiež by bola doplnená obrázkom.

Tabuľka 1. Zadanie úlohy určenej pre nevidiacich žiakov

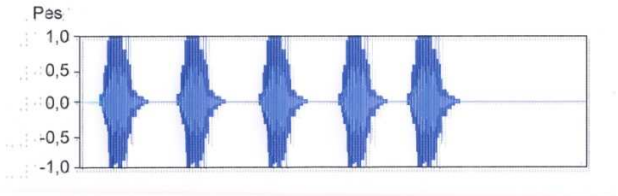
Otvorte si dokument *Nadpisy2.docx*. V dokumente sa píše o rôznych afrických zvieratách . Po nadpisoch sa však nemôžeme presúvať, pretože nemajú nastavený nadpisový štýl. V prvom riadku je text *Zvieratá Afriky*. Tomuto riadku nastavte nadpisový štýl úrovne jedna. Názvy zvierat, ktoré sú uvedené v samostatných riadkoch, by mali mať nastavený nadpisový štýl úrovne dva. Nájdite takéto riadky a nastavte im nadpisový štýl úrovne dva. Keď správne nastavíte štýly všetkým nadpisom, nastavte sa na začiatok dokumentu a presúvajte sa po nadpisoch. Zapamätaj si! • Nadpisový štýl nastavíme tak, že sa presunieme do riadku, ktorého štýl chceme nastaviť a stláčame **Alt+Shift+šípka vľavo** alebo **Alt+Shift+šípka vpravo**. • Normálny štýl nastavíme tak, že sa presuniememe do riadku, ktorého štýl chceme nastaviť a stláčíme **Ctrl+Shift+N**. • Aký štýl má text nastavený zistíme tak, že sa naň nastavíme a stlačíme **Insert+F**.

3.2 Práca so zvukom

Cieľom učebného materiálu je, aby si žiaci precvičili prehrávanie zvukových súborov, naučili sa nahrávať a prehrávať zvuky, naučili sa spájať dva a viac zvukov, vedeli použiť niektoré efekty na modifikáciu zvukov a osvojili si základnú terminológiu potrebnú pre spracovanie zvukovej informácie pomocou počítača.

Z hľadiska vstupných nárokov na vedomosti a zručnosti žiakov predpokladáme, že žiaci vedia spustiť ľubovoľnú aplikáciu, ovládajú základnú orientáciu na klávesnici, vedia uložiť, premenovať a načítať súbor z disku.

Materiál obsahuje takmer 30 úloh, ktoré predpokladajú použitie editora AudaCity [18]. Tento editor je plne ovládateľný pomocou klávesnice [19] a dobre spolupracuje s čítačom obrazovky. Snažili sme sa, aby žiaci dokázali použiť tento program ako nástroj na rozvoj vlastnej kreativity a fantázie.



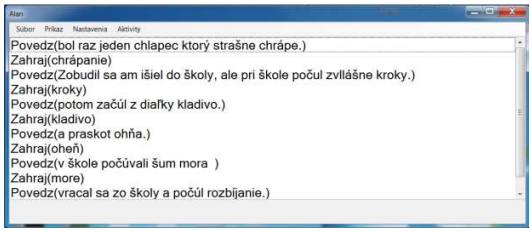
Obrázok 3. Brechot psa zobrazený ako vlnenie v programe AudaCity (reliéfny obrázok)

Súčasťou materiálu sú opäť aj reliéfne obrázky ilustrujúce niektoré prvky prostredia, s ktorými žiaci pracujú (napríklad posuvník na nastavenie hlasitosti), tiež aj reliéfne obrázky ilustrujúce zvuk ako vlnenie (obrázok 3).

3.3 Algoritmy a riešenie problémov

Rozvoj algoritmického myslenia u nevidiacich žiakov považujeme za veľmi dôležitý, nakoľko prax ukazuje, že v softvérových firmách úspešne pracuje nezanedbateľné percento nevidiacich programátorov. Uvedomujeme si, že čím skôr sa vhodne podchytí záujem žiakov o túto profesiu, tým ľahšie sa budú vedieť rozhodnúť pre ďalšie štúdium potrebné pre získanie potrebnej kvalifikácie. Cieľom učebného materiálu je, aby si žiaci osvojili základné programátorské zručnosti a pojmy, ako sú **príkaz, sekvencia príkazov, cyklus, príkaz vetvenia, podprogram, premenná**.

Učebný materiál bude obsahovať aj úlohy, ktoré sa dajú riešiť mimo počítača, či už s využitím robotických hračiek, ako sú včela Bee-Bot a bager Constructa-Bot, alebo za pomoci reliéfnej tabuľky [20]. Zvažujeme aj zaradenie úloh zo súťaže Informatický bobor s algoritmickým zameraním, prípadne ich vhodné modifikácie [21], [22].



Obrázok 4. Program žiaka vytvorený v prostredí Alan

Výučba programovania si však vyžaduje vhodné programátorské prostredie. Časť úloh bude viazaná na zvukové programátorské prostredie Alan, ktoré v spolupráci s nami vyvinul v rámci svojej bakalárskej práce študent aplikovanej informatiky [23]. Toto prostredie umožňuje programovať zvukové príbehy (obrázok 4) a popri tom rozvíjať tvorivosť a osvojiť si všetky požadované programátorské koncepty. Program Alan bude voľne dostupný spolu s učebným materiálom.

Obrázok 5. Hmatový programovací jazyk Torino [23]

Firma Microsoft Research Cambridge vyvíja hmatový programovací jazyk Torino [24] priateľský nevidiacim žiakom (obrázok 5). Toto prostredie by malo byť dostupné koncom roku 2017. Budú sa v ňom dať riešiť podobné úlohy ako v prostredí Alan. Plánujeme ho zakúpiť, preskúmať a v prípade potreby náš učebný materiál vhodne prispôsobiť, aby ho mohli učitelia použiť pri riešení úloh so svojimi nevidiacimi žiakmi.

Podľa aktuálneho harmonogramu má byť tento učebný materiál dokončený koncom roku 2017. Vzhľadom na naše ambície byť

ústretoví voči viacerým platformám, to bude pravdepodobne najrozsiahlejší učebný materiál.

3.4 Práca s matematickým textom

Cieľom učebného materiálu bude poskytnúť praktické návody na efektívne používanie matematického editora Lambda pri vyučovaní matematiky. Editor Lambda (Linear Access to Mathematics for Braille Device and Audio-synthesis – Lineárny prístup k matematike pre braillovské zariadenia a hlasový výstup) je výsledkom medzinárodného projektu Lambda Project. Cieľom projektu bolo poskytnúť nevidiacim študentom nástroj, pomocou ktorého nielen matematický príklad zapíšu, ale môžu ho aj prehľadne riešiť [25]. Editor Lambda možno teda považovať za elektronický zošit, do ktorého žiak zapisuje a z ktorého číta pomocou výstupu na Braillovom riadku a čítača obrazovky. Zároveň je možné zápis zobraziť v zabudovanom grafickom prehliadači. Nutným predpokladom je, že žiaci sa dokážu orientovať na klávesnici a majú skúsenosti s prácou v textovom editore.

Prvá časť materiálu bude venovaná oboznámeniu sa s pracovným prostredím editora. Editor má dve prostredia – matematické a textové. V každom prostredí reaguje editor na vstupy z klávesnice iným spôsobom. Uvedené budú úlohy na nácvik rozlišovania prostredí a prepínanie sa medzi nimi.

Ďalšia časť bude venovaná pomocným nástrojom editora (obrázok 6) a jej cieľom bude vytvoriť u žiakov potrebné návyky pri ich používaní. Konkrétne sa jedná o tieto nástroje:

– kopírovanie riadka,

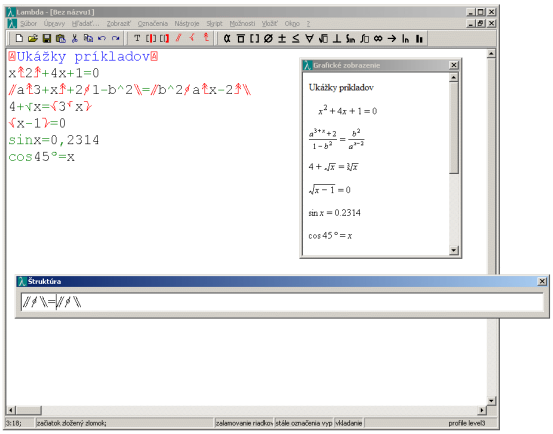
– zobrazenie rozvinutej a kompaktnej štruktúry,

– kalkulačka,

– vloženie výsledku vypočítaného výrazu na miesto kurzora,

– viacnásobná pamäťová schránka - umožňuje uložiť 9 pomocných výpočtov.

Ku každému nástroju bude uvedená praktická ukážka využitia.



Obrázok 6. Ukážka editora s príkladmi matematického zápisu, grafického zobrazenia, kompaktnej štruktúry

Ďalšie časti budú venované vkladaniu jednotlivých matematických zápisov. Dôležité bude, aby si žiaci uvedomili, že matematický zápis má vizuálnu stránku (zlomková čiara, umiestnenie exponentu, znak odmocniny a podobne) a zároveň aj významovú stránku (čitateľ v zlomku je delený menovateľom, základ mocniny násobíme toľkokrát samým sebou, aká je hodnota exponentu a podobne). Pomocou klávesových príkazov vkladáme jednotlivé časti významovej stránky. Je potrebné vkladať ich v správnom poradí, aby ich mohol korektne prečítať čítač obrazovky a tiež aby sa zápis ukázal v grafickom zobrazení. Ukážkové úlohy budú v súlade s témami Štátneho vzdelávacieho programu. [26], [27], [28].

Samostatná časť bude venovaná grafickému zobrazeniu, v ktorom je matematický zápis zobrazený v podobe, akú poznajú vidiace osoby.

Doterajšie skúsenosti ukázali, že je vhodné učiť sa používať editor Lambda s nástupom na druhý stupeň [29]. Oboznamovanie sa s editorom je potrebné mimo vyučovacích hodín matematiky. Na hodinách matematiky by sa žiaci mali sústrediť už iba na preberané učivo. Vhodným priestorom sú teda hodiny informatiky. Zručnosti v používaní editora získavajú žiaci postupne. V učebnici budú jednotlivé príklady použitia korešpondovať s úrovňou vedomostí z matematiky. Témy z učebnice bude možné použiť v rôznych ročníkoch.

3.5 Práca s internetom

Tento učebný materiál je plánovaný s výhľadom do roku 2018, preto sa možno naše súčasné predstavy nebudú úplne zhodovať s jeho finálnou verziou. Bude určený pre žiakov, ktorí vedia editovať text a ovládajú základy práce v operačnom systéme.

Cieľom bude, aby sa naučili bezpečne a eticky používať nástroje na **interaktívnu a neinteraktívnu komunikáciu** cez internet. Tiež aby si vedeli vyhľadať informácie potrebné pre osobný život, štúdium a prácu.

Zo softvérových riešení máme aktuálne dobré skúseností s prenosnou verziou programu Mozilla Thunderbird a Skype a prehliadačmi Mozilla Firefox, tiež aj s prehliadačom Internet Explorer.

Veľké množstvo webových stránok je pre nevidiacich žiakov ťažko použiteľných, pretože nespĺňajú štandardy prístupnosti [30]. V úlohách zameraných na čítanie webových stránok plánujeme použiť cvičné sídlo vyhovujúce podmienkam prístupnosti, ktoré vyvinieme iba pre tento účel. Úlohy zamerané na vyhľadávanie informácií sa budú orientovať na overené sídla, akými sú vyhľadávač Google, zvukové zdroje na sídle YouTube a informačné zdroje na sídle Wikipedia.

4. ZÁVER

Ako sme už uviedli, pri tvorbe učebných materiálov vychádzame z našich dlhoročných výskumných pozorovaní, ktoré sme realizovali na základnej škole pre slabozrakých a nevidiacich. Väčšina úloh, ktoré uvádzame, už prešla niekoľkými iteráciami pozostávajúcimi z fázy návrhu, realizácie, overenia a modifikácie. Podobne to bude aj s každým učebným materiálom ako celkom, ktorý v spolupráci s učiteľmi nevidiacich žiakov overíme formou evalvačného výskumu [31]. Na každý materiál vypracujú posudky dvaja nezávislí recenzenti a ich pripomienky zapracujeme.

V súčasnosti prebieha overovanie materiálu venovaného práci s textom. Možno konštatovať, že žiakom a učiteľom sa s ním dobre pracuje, ale pozorovali sme viaceré zaujímavé reakcie žiakov, ktoré si vyžiadajú menšie úpravy učebného textu a aj dátových súborov.

– Niektoré texty v dátových súboroch sú príliš rozsiahle. Pre nácvik požadovaného úkonu by postačovali aj kratšie texty, aby sa žiaci zbytočne neunavili.

– Je potrebné upraviť obsah niektorých dátových súborov, nakoľko obsahujú pojmy, ktorým nevidiaci žiaci nerozumejú, respektíve si ich nevedia predstaviť (napríklad pomenovanie kvetov, s ktorými sa ešte nestretli).

Niektoré návody obsahujú klávesové skratky, v ktorých treba stláčať čísla na alfanumerickej klávesnici. Ak žiaci stlačia čísla na numerickej klávesnici, nemá to požadovaný efekt. Je preto potrebné v návodoch presne špecifikovať, ktoré klávesy treba stláčať.

Zistili sme, že reliéfne obrázky musia byť nevyhnutnou prílohou učebného materiálu. Nevidiaci žiaci si bez nich nedokázali predstaviť niektoré objekty, ako napríklad odrážkový zoznam alebo komplikovanejšie vnorené zoznamy.

Niektoré ich predstavy boli nesprávne. Napríklad veľkosť riadkovania 2 si niektorí žiaci predstavovali tak, že sa veľkosť písmen v riadku zdvojnásobí.

Keďže budú všetky materiály dostupné na webovom portáli, očakávame spätnú väzbu aj od učiteľov informatiky z viacerých škôl. Po vytvorení všetkých materiálov plánujeme zorganizovať pracovný seminár pre učiteľov nevidiacich žiakov a akademických pracovníkov zaoberajúcich sa vzdelávaním nevidiacich zo Slovenska a Českej republiky. Tento seminár bude ďalšou platformou umožňujúcou výmenu skúseností a získanie spätnej väzby.

5. POĎAKOVANIE

Tento príspevok vznikol s finančnou podporou projektu Kega 014UK-4/2016 **Rozvoj kompetencií nevidiacich žiakov ZŠ potrebných pre život v 21. storočí.**

6. BIBLIOGRAFICKÉ ODKAZY

[1] ŠNAJDER, Ľ. et al. *Úvod do vzdelávania. Informatika vo všeobecnom vzdelávaní. Didaktika programovania pre ZŠ 2, Didaktika informatiky a informatickej výchovy*, Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika, Bratislava : ŠPÚ, UPJŠ, UK, UKF, UMB, ŽUŽa, 2009. ISBN 978-80-89225-53-8

[2] SALANCI, Ľ. *Tvorivá informatika. 1. zošit o obrázkoch*. Bratislava : SPN, Mladé letá, s.r.o., 2005. ISBN 80-10-00649-1.

[3] BLAHO, A. – KALAŠ, I. *Tvorivá informatika. 1. zošit z programovania*. Bratislava : SPN, Mladé letá, s.r.o., 2005. 80-10-00019-*1*.

[4] VARGA, M. – HRUŠECKÁ, A. *Tvorivá informatika. 1. zošit s Internetom*. Bratislava : SPN, Mladé letá, s.r.o., 2006. ISBN 80-10-00648-3.

[5] BLAHO, A. – SALANCI, Ľ *Tvorivá informatika. 1. zošit o práci s textom*. Bratislava : SPN, Mladé letá, s.r.o., 2007. ISBN 978-80-10-00888-9.

[6] KALAŠ, I. – WINCZER, M. *Tvorivá informatika: Informatika okolo nás*. Bratislava : SPN, Mladé letá, s.r.o., 2007. ISBN 978-80-10-00887-2

[7] KALAŠ, I. – BEZÁKOVÁ, D. *Tvorivá informatika: 1. zošit o číslach a tabuľkách*. Bratislava : SPN, Mladé letá, s.r.o., 2009. ISBN 978-80-10-01718-8

[8] VARGA, M. – HRUŠECKÁ, A. *Tvorivá informatika. 2. zošit s Internetom*. Bratislava : SPN, Mladé letá, s.r.o., 2010. ISBN 978-80-10-01630-3.

[9] KALAŠ, I. at al. *Tvorivá informatika: 1. zošit o prezentáciách*. Bratislava : SPN, Mladé letá, s.r.o., 2011.

[10] BLAHO, A. et al. *Informatická výchova pre 2. ročník ZŠ*, Aitec, 2010.

[11] MOSNÁ, M. *Informatická výchova pre 2. ročník základných škôl (Pracovný zošit)*, Aitec, 2013.

[12] MOSNÁ, M. *Informatická výchova pre 3. ročník základných škôl (Pracovný zošit)*, Aitec, 2011.

[13] MOSNÁ, M. et al. *Informatická výchova pre 4. ročník základných škôl (Pracovný zošit)*, Aitec, 2013.

[14] JAŠKOVÁ, Ľ. Ako začať s informatikou na ZŠ pre nevidiacich žiakov? In: TRAJTEĽ, Ľ. (Eds.) *Didinfo 2013*, Banská Bystrica : Univerzita Mateja Bela, 2013. ISBN 978-80-557-0527-9.

[15] PAPERT, S. *Constructionism vs. Instructionism : speech to an audience of educators in Japan*, 1980. Dostupné na internete: http://www.papert.org/articles/const_inst/const_inst1.html

[16] JAŠKOVÁ, Ľ. Kognitívne schopnosti nevidiacich žiakov druhého stupňa ZŠ a ich využitie pri vyučovaní informatiky In: TRAJTEĽ, Ľ. (Eds.) *Didinfo 2016*, Banská Bystrica : Univerzita Mateja Bela, 2016. ISBN 978-80-557-1082-2.

[17] Vyučovanie informatiky pre nevidiacich [online]. [cit. 2017-01-03]. Dostupné na internete: http://vin.edi.fmph.uniba.sk/

[18] Audacity [online]. [cit. 2017-01-03]. Dostupné na internete: http://www.audacityteam.org/

[19] *Zvukový editor Auda City* [online]. [cit. 2017-01-03]. Dostupné na internete: http://jelenovipivonelej.info/?page=audacity

[20] JAŠKOVÁ, Ľ. – KALIAKOVÁ, M. Programovacie prostredia pre deti s poruchami zraku. In: LOVÁSZOVÁ, G. (Eds.) Didinfo 2014, Banská Bystrica : Univerzita Mateja Bela, 2014. ISBN 978-80-557-0698-6.

[21] JAŠKOVÁ, Ľ. – KOVÁČOVÁ, N. Prvé skúsenosti s realizáciou súťaže iBobor pre nevidiacich žiakov druhého stupňa ZŠ In: TRAJTEĹ, Ľ. (Eds.) Didinfo 2015, Banská Bystrica : Univerzita Mateja Bela, 2015. ISBN 978-80-557-0852-2.

[22] KOVÁČOVÁ, N – JAŠKOVÁ, Ľ.. Porovnanie úspešnosti nevidiacich a intaktných žiakov pri riešení úloh súťaže iBobor In: TRAJTEĹ, Ľ. (Eds.) Didinfo 2016, Banská Bystrica : Univerzita Mateja Bela, 2016. ISBN 978-80-557-1082-2.

[23] KOVÁČ, M. Programovacie prostredie pre zrakovo postihnutých žiakov ZŠ, Bratislava: FMFI UK, bakalárska práca, 2016

[24] Torino. A physical programming language inclusive of blind children [online]. [cit. 2017-01-03]. Dostupné na internete: https://hxd.research.microsoft.com/work/torino.php

[25] Lambda Project [online]. [cit. 2017-01-03]. Dostupné na internete: http://lambdaproject.org/

[26] Štátny vzdelávací program. Matematika. Príloha ISCED 2 [online]. [cit. 2017-01-03]. Dostupné na internete: http://www.statpedu.sk/sites/default/files/dokumenty/statny-vzdelavaci-program/matematika_isced2.pdf

[27] Štátny vzdelávací program. Matematika. Príloha ISCED 3A [online]. [cit. 2017-01-03]. Dostupné na internete:

http://www.statpedu.sk/sites/default/files/dokumenty/statny-vzdelavaci-program/matematika_isced3a.pdf

[28] Vzdelávací program pre deti a žiakov so zrakovým postihnutím. [online]. [cit. 2017-01-03]. Dostupné na internete: http://www.statpedu.sk/sites/default/files/dokumenty/statny-vzdelavaci-program/vp_zp_%20isced_1_2_3_5.pdf

[29] KALIAKOVÁ, M. – STANKOVIČOVÁ, M. – VITÁLIŠOVÁ Z. Informatické kompetencie žiakov ZŠ so zrakovým postihnutím a ich využitie na vyučovaní matematiky a fyziky používaním matematického editora LAMBDA In: TRAJTEĽ, Ľ. (Eds.) *Didinfo 2015*, Banská Bystrica : Univerzita Mateja Bela, 2015. ISBN 978-80-557-0852-2.

[30] Blind Friendly Web [online]. [cit. 2017-01-03]. Dostupné na internete: https://www.blindfriendly.sk/

[31] HENDL, J. *Kvalitativní výzkum. Základní metody a aplikace.* Praha: Portál s.r.o, 2005, 408 s. ISBN 80-7367-040-2.

Súťaž KODU CUP ako podporný prostriedok algoritmického myslenia

KODU CUP Competition as a Supporting Method of Algorithmic Thinking

Nika Klimová
Katedra informatiky,
FPV UMB Tajovského 40
974 01 Banská Bystrica Slovensko
nika.klimova2@gmail.com

Dana Horváthová
Katedra informatiky,
FPV UMB Tajovského 40
974 01 Banská Bystrica Slovensko
dana.horvathova@umb.sk

ABSTRACT

This paper talks about the contest KODU CUP which aims to support the method of algorithmic thinking, creativity, developing fantasy and critical thinking. Pupils from the primary school led by one of the authors have also joined this contest supported by Microsoft Slovakia co., which is running for the 3rd time. The paper describes the contest itself, experience with its course, method of developing games, and also examples of games created by children of this team.

Keywords

Kodu Cup. Kodu Game Lab. Microsoft. Competition. Programming.

ABSTRAKT

Príspevok sa venuje predstaveniu súťaže KODU CUP, ktorá je zameraná na podporu algoritmického myslenia a kreativity detí pri vytváraní hier a snaží sa rozvíjať v deťoch fantáziu a prebúdzať kritické myslenie. Tejto súťaže medzinárodného významu, ktorá je na Slovensku organizovaná spoločnosťou Microsoft Slovakia už tretí rok, sa zúčastnili aj žiaci ZŠ J. Bakossa v Banskej Bystrici pod vedením jednej z autoriek. Príspevok približuje samotnú súťaž, skúsenosti s jej priebehom, postup vytvárania hry, ako aj ukážky vytvorených hier detí tohto tímu.

Kľúčové slová

Kodu Cup. Kodu Game Lab. Microsoft. Súťaž. Programovanie.

1. ÚVOD

Dnešné deti majú k hrám na počítačoch, smartfónoch a iných zariadeniach veľmi blízko. Súťaží v informatike je dnes už viac ako bolo v minulosti. Medzi pomerne mladé zaraďujeme aj súťaž KODU CUP, ktorej 3. ročník prebehol v jeseni 2016.

2. Vznik súťaže

V roku 2012 sa učiteľ Mgr. Peter Pallo, autor myšlienky súťaže [1], zúčastnil Konferencie moderných učiteľov v Portugalsku, kde sa stretol so softvérom KODU GAME LAB a následne v spolupráci so spoločnosťou Microsoft Slovakia vznikla súťaž KODU CUP.

V školskom roku 2013/2014 [2] sa konal nultý ročník súťaže KODU CUP, kde sa zapojili žiaci 1. a 2. stupňa ZŠ R. Dilonga v Trstenej. Počet zúčastnených žiakov sa vyšplhal na 70 a odovzdaných hier do súťaže bolo 68. Motiváciou sa stala možnosť vycestovať do centrály Microsoft Slovakia, kde sa pre víťazných žiakov zorganizovala prehliadka priestorov Microsoft. Títo žiaci upútali pozornosť vedenia spoločnosti a boli odmenení aj vecnými cenami.

3. Zadanie a cieľ súťaže

Pokiaľ sa žiaci chcú zúčastniť súťaže, je nevyhnutné, aby splnili zadanie, ktoré je stručné: musia vytvoriť vlastnú hru v Kodu Game Lab [3] a určiť jej pravidlá.

Cieľom súťaže je podporiť algoritmické myslenie, rozvíjať fantáziu a kritické myslenie a v neposlednom rade prebudiť v žiakoch kreativitu.

4. Kodu Game Lab

Ako bolo už vyššie spomenuté, na súťaž je nevyhnutné mať nainštalovaný program Kodu Game Lab, pre mladších žiakov najlepšie v češtine. Ide o [4] vizuálny programovací jazyk, ktorý je založený na blokoch. Softvérové rozhranie je blízke deťom, pretože je jednak v 3D a celková grafika je podobná komerčným hrám, ktoré sa žiaci na základných školách hrávajú. Okrem toho uvádzame porovnanie s inými programami:

Softvér/ Vlastnosti	Alice	Scratch	Game Maker	Kodu Game Lab
Izometrické 3D	✓	✗	✗	✓
Podporuje Xbox	✗	✗	✗	✓
Integrovaný hrací mód	✓	✗	✓	✓
Zdieľanie hry bez nainštalovaného softvéru	✗	✓	✗	✓
Integrované tutoriály	✓	✗	✗	✓
Cena	zadarmo	zadarmo	zadarmo	zadarmo

Tu vidíme hlavné rozdiely v porovnaní s inými softvérmi. Okrem 3D grafiky podporuje Kodu Game Lab aj konzolu Xbox 360

a Xbox One, žiaci sa môžu zamerať na ovládače z Xbox v programovaní hry.

Obrázok 1 Možnosť využitia ovládača z konzoly Xbox

Netreba zabúdať ani na to, že Kodu Game Lab má vstavaný tutoriál, ktorý dieťa naučí krok po kroku vytvoriť hru. Nakoniec významnú úlohu v školách zohráva aj cena softvéru, ktorá je v tomto prípade a aj v prípade porovnaných softvérov nulová.

4.1 Postup vytvárania hry v Kodu Game Lab

Kliknutím na nový svet sa dá dostať na zelenú plochu, tzv. svet. Vo svete je možné pohybovať sa v trojrozmernom prostredí, približovať / vzďaľovať alebo otáčať sa. Hra môže pozostávať z niekoľkých častí:

- Tvorba prostredia, kde sa hra bude odohrávať
- Nasadenie objektov
- Cesty
- Programovanie
- Export

Tvorba prostredia sa dá ovplyvniť viacerými tlačidlami. Jedným z nich je základná tvorba terénu (súše), ktorých je na výber 121 druhov. Tá sa dá formovať prostredníctvom tlačidiel kopcov / údolí, vyhladením / zdrsnením terénu a vodou. Pridanie vody v sebe obsahuje 10 druhov vrátane lávy. Okrem toho sa v prostredí nachádzajú aj iné možnosti [5]:

- postaviť / zbúrať sklenené steny (aby objekty nespadli po prekročení terénu)
- režim kamery (pevná pozícia, pevný offset, voľná kamera)
- výška a sila vĺn
- osvetlenie (8 možností)
- farba oblohy (21 možností)
- sila vetra
- hlasitosť zvukových efektov a hudby

alebo ďalšie novinky, ako:

- viditeľnosť skóre
- kompas
- priehľadnosť vody
- zobrazenie virtuálneho ovládania

Nasadenie objektov je pre vytvorenie hry nevyhnutnou súčasťou. Zahrňujú postavy, ale aj environmentálne elementy. Rozdiel medzi nimi je ten, že elementy, ako napr. skaly (horniny, ľadovce, vyvreté horniny) vodné elementy (hviezdice, lastúry, lekná, morské riasy), trubky, stromy, jablká, rakety, zámky, mince, továrne, oblaky, ventilátory a iné sa nedokážu pohybovať.

Objekty, ktoré tu nájdeme, sú:

Kodu	rover	balón	motorka
lietajúca ryba	lietadlo	vzducholoď	kanón
ponorka	hviezdica	chobotnica	svetlonos
tanier na hádzanie	oblak	bubon	loď
svetlo	mína	družica	ryba
korytnačka	palica	kľučka	chyža

Obrázok 2 Ukážka objektov

Okrem programovania objektov je možné meniť ich farbu, veľkosť, výšku, dajú sa kopírovať, rôzne meniť alebo ich otáčať. Okrem toho sa dajú zmeniť ich vlastnosti v nastaveniach, napr. zvýšiť rýchlosť pohybu vpred, akceleráciu, môžeme prikázať objektu byť nepohyblivým, nezraniteľným, neviditeľným či dokonca duchom, klonovateľným, potápateľným alebo plávajúcim na vodnej hladine a iné. Niektoré postavy majú aj špeciálne vlastnosti, napr. rover dokáže vypáliť laserový lúč do skaly alebo z nej odobrať vzorky.

Cesty môžu plniť úlohu ako cesty, mosty alebo steny. Pre dekoratívne účely je možné použiť napr. flóru, ktorá sa môže nachádzať popri ceste. Okrem toho cesta môže mať aj inú funkciu, kde objekt chodí po danej trase. Najčastejšie využitie je pri tzv. nepriateľoch, ktorí sa majú pohybovať po určitej trase a majú byť ovládaní počítačom.

Programovací jazyk, ktorý slúži na vytváranie hry, sa nazýva Kodu [5]. Každý príkaz je rozdelený do dvoch častí, a to časť *when* - *podmienka* a *do* – *akcia*. Časti kódu sú usporiadané do blokov s možnosťou nápovede, ktorá je závislá na predchádzajúcom bloku a naznačuje ukážky kódov, ktoré sa objavia pri kliknutí na blok a stlačení klávesu Y.

Obrázok 3 Programovací jazyk Kodu

V tomto programovacom jazyku je možné [5] pracovať s premennými – skóre, ktoré je vždy typu *integer* a je globálne. Na výber je 37 rôznych skóre, ktoré sa líšia abecedou alebo farbami. Skóre môže zohrávať aj úlohu časovača. Zároveň sa dá pracovať s podmienkami či naučiť sa Boolovské výrazy – negácia, ktorá je

v tomto jazyku ľahká na pochopenie cez blok „not“, konjukcia a disjunkcia. Nachádza sa tu aj možnosť učiť dedičnosť cez nastavenie „klonovateľné“ a blok „vytvoriteľné“. Strany sa používajú v tom prípade, keď je potrebné zmeniť správanie objektu počas hry, napr. rover príde k vode, kde je loď. Rover sa zmení na loď (spustí sa príkaz prechodu na stranu 2) a je schopný ovládať loď, hoci na začiatku ovládal rovera.

5. Informácie o súťaži

Súťaž je rozdelená do 3 kategórií pre základné školy [6]. Tie sú nasledovné:

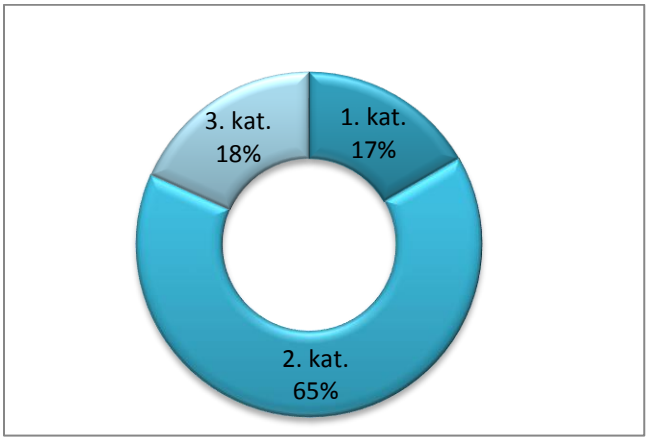
1. kategória: 1. – 3. ročník
2. kategória: 4. – 6. ročník
3. kategória: 7. – 9. ročník

Pri hodnotení mali žiaci možnosť získať až 100 bodov, ktoré boli rozdelené do viacerých kritérií: náročnosť programovania a funkčnosť hry (30 b), prepracovanosť prostredia (30 b), herná stratégia (25 b) a celkový dojem (15 b). Realizovali sa školské kolá a následne celoslovenské finále. Z každej kategórie školského kola mohol postúpiť do finále len 1 žiak, t.j. z každej školy maximálne 3 žiaci. Tí museli odovzdať:

- vyexportovanú hru z Kodu Game Lab
- dokument s cieľom a pravidlami hry
- video, kde žiak sníma obrazovku, prevedie pozorovateľov hrou a zároveň hru okomentuje. Rozsah videa bol stanovený podľa kategórií v rozsahu 2 – 3 min.

6. Priebeh súťaže na ZŠ J. Bakossa

Do súťaže KODU CUP bola zapojená Základná škola Jána Bakossa v Banskej Bystrici prvýkrát v roku 2016. Predtým žiaci tejto školy nemali skúsenosť s prostredím Kodu Game Lab. Vzhľadom na tento fakt sme neočakávali veľký záujem zo strany žiakov. Preto sme sa snažili ich motivovať, aby sa aspoň niekto do súťaže prihlásil. Najväčšou motiváciou bola nielen hlavná výhra v podobe Xbox One, ale aj možnosť vymyslieť si vlastnú hru a zrealizovať ju. Konečné číslo súťažiacich na tejto škole sa vyšplhalo na počet 60.



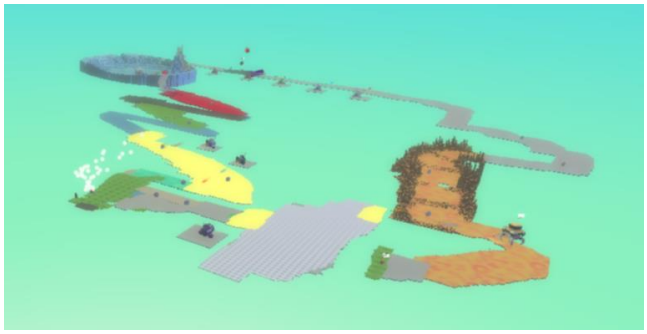
Graf 1 Percentuálne zastúpenie žiakov v jednotlivých kategóriách

Vzhľadom na fakt, že mnoho z týchto detí nikdy neprogramovalo, rozhodli sme sa, že nahráme žiakom videotutoriály s hovoreným slovom, ktoré sme rozdelili do 5 častí – inštalovanie Kodu Game Lab, tvorba prostredia, programovanie, ukončenie hry a export. Žiaci boli oboznámení s dátumom, do kedy majú hry elektronicky poslať. Hru zo 60 žiakov odovzdala len 1 žiačka. Po rozhovore so žiakmi sme zistili, že nevedia správne pracovať s videotutoriálmi. Pravdepodobne to bolo spôsobené tým, že sa s videotutoriálmi ešte nestretli a neboli na ne zvyknutí. Zároveň sme zistili aj to, že mladší žiaci mali problém prepínať medzi jednotlivým oknami. Rozhodli sme sa pre ďalšiu možnosť - školské kolo, ktoré prebehlo v škole v trvaní asi 3 hodín, kde žiaci priamo programovali svoje vopred navrhnuté hry. Počas kola mali možnosť sledovať aj premietanie cez videoprojektor, kde mohli vidieť v reálnom čase vypracovávanú hru a mohli sa tak zoznámiť s niektorými možnosťami, ktoré Kodu Game Lab ponúka. Vyhodnotenie hry bolo jednoznačné, z každej kategórie postúpil jeden žiak, ktorý mohol ešte doma svoju hru vylepšiť a zdokonaliť. Okrem toho museli vymyslieť názov hry a splniť vyššie uvedené podmienky pre účasť vo finále.

Výhercovia školského kola vytvorili hry:

1. kategória – Svet kanónov

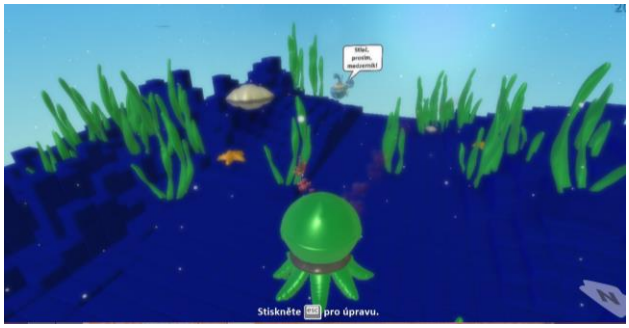
Popis hry: *„Si motorkár. Za úlohu máš pozbierať všetky mince, za ktoré dostávaš body. Jedna minca vo vode zmení vodu na lávu, ale neublíži ti. V hre je veľa nepriateľov – kanóny, lietadlá, chobotnica, loď a kľučka. Všetci do teba strieľajú a snažia sa ťa zabiť. Niektorým sa stačí vyhnúť, iných treba zabiť, aby si mohol ísť ďalej. Ty strieľaš guličky. V hre máš aj dobrého kamaráta Kodu, ktorý ti chce dať body, ale najskôr sa mu musíš pozdraviť. Keď prejdeš svet a zabiješ aj veľkú kľučku, dostaneš sa k srdiečku – cieľu. Ak ťa ale nejaký nepriateľ zabije, prehral si.“*



Obrázok 4 Ukážka hry z 1. kategórie

2. kategória – Zákon prírody

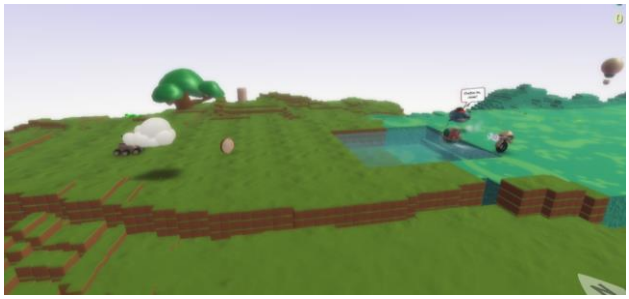
Popis hry: *„Podstata hry je vystihnutá v názve. Užívateľ je chobotnica, ktorá je hladná, a preto chce zjesť ryby. Ryby, ktoré sú zamilované, môže zjesť, za čo dostáva body. Keď dosiahne 50 bodov, vyhráva. V hre je ale aj 1 nahnevaná ryba, ktorej sa nepáči, že chobotnica ostatné ryby je a zároveň je veľká bojovníčka, preto sa snaží chobotnicu zabiť. Keď sa s ňou stretne, nemá šancu na prežitie. V hre sú aj ponorky, ktoré chcú nalákať chobotnicu, aby stlačila medzerník. Keď chobotnica stlačí medzerník, tak zomrie. Zákon prírody spočíva v hierarchii, kedy v tejto hre je najvyššie ponorka, potom chobotnica a nakoniec ryby.“*



Obrázok 5 Ukážka hry z 2. kategórie

3. kategória – Dobrodružstvo s kamarátmi

Popis hry: „*Továreň je domov rovera. Rover bol navštíviť svojho kamaráta a teraz, keď sa vracal domov, našiel prekážky, ktoré musí prekonať. Okrem toho našiel aj svojho kamaráta, ktorému by mal ako správny kamarát, pomôcť. Dostane sa rover do továrne – svojho domova a pomôže kamarátovi?*“



Obrázok 6 Ukážka hry z 3. kategórie

Zaujímavé bolo poznanie, ku ktorému sme dospeli porovnaním použitia rôznych funkcií súťažiacimi žiakmi v jednotlivých kategóriách. Zistili sme, že existuje priama úmernosť medzi kategóriou a použitými funkciami. Z nasledujúcej tabuľky 1, kde sme porovnali použitie rôznych funkcií v jednotlivých kategóriách, je vidieť, že v 3. kategórii žiak, ktorý bol najstarší, použil najviac funkcií, pričom v ostatných kategóriách bol ich počet rovnaký.

Tabuľka 1 Použité funkcie v jednotlivých kategóriách

	1. kat.	2. kat.	3.kat.
chodenie	✓	✓	✓
hovorenie	✓	✓	✓
strieľanie	✓	✗	✓
zbieranie (mincí)	✓	✓	✓
nasmerovanie rakiet	✓	✗	✓
pridanie bodov	✓	✓	✓
emócie	✗	✓	✓
porovnávanie bodov	✗	✓	✓
zobrať a dať jablko	✗	✗	✓
otočiť sa	✗	✗	✓
ukazovateľ zdravia	✗	✗	✓
blúdenie (spôsob pohybu)	✓	✓	✓
nástroj cesta	✗	✗	✓
Použité spolu	7	7	13

Treba však zdôrazniť aj pohlavie žiakov, kde chlapci v 1. a 3. kategórii použili aj strieľanie guličiek alebo rakiet a ich nasmerovanie na svetovú stranu, dievča v 2. kategórii tieto funkcie nevyužilo.

7. Celoslovenské finále KODU CUP

Dňa 5. 12. 2016 prebehlo celoslovenské finále súťaže KODU CUP. Zúčastnilo sa ho 21 žiakov z 11 škôl, ktorí sa stali víťazmi školských kôl spomedzi viac ako 230 súťažiacich z 20 základných škôl.

Finále sa začalo o 10:00 hod v Hotelovej akadémii v Bratislave, kde boli pozvané aj médiá. Prebehol kurz pod názvom Hodina kódu, kde žiaci programovali v hre Minecraft Designer a popri tom sa spojili cez Skype s rôznymi triedami na Slovensku vrátane Microsoft Inkubátoru inovatívnych učiteľov. Neskôr sa žiaci presunuli do spoločnosti Microsoft, kde boli oboznámení s novinkami z Microsoft Office a Windows 10 vrátane ukážky nových programov. Prebehlo vyhlásenie víťazov z každej kategórie, ktorí si odniesli Xbox One. Všetci súťažiaci dostali atraktívne vecné ceny. Nechýbala ani prehliadka samotného Microsoftu spojená s rôznymi informáciami, napr. koľko Horaliek mesačne zje zamestnanec firmy.

Obrázok 7 Víťazi jednotlivých kategórií

8. Záver

KODU CUP je typ súťaže, ktorá dokáže žiakov motivovať k programovaniu. Aj po skončení súťaže niektorí žiaci naďalej programujú svoje hry vo svojom voľnom čase a pripravujú sa už na budúci ročník. Aj keď sa metóda videotutoriálov v tomto prípade neosvedčila, nezamietame ju. Práve naopak, chceme žiakov dopredu viesť k tomu, ako videotutoriály používať, aby si na ne zvykli už počas roka, aby to pre nich nebola počas súťaže úplne nová vec. Myslíme si, že zapojenie žiakov do súťaže malo zmysel. Cieľ súťaže bol splnený. Okrem toho, že žiakov súťaženie bavilo, pracovali na rozvoji svojich algoritmických schopností, fantázie, kritického myslenia a kreativity.

BIBLIOGRAFICKÉ ODKAZY

[1] Rozhlas a televízia Slovenska. Dostupné na: <http://www.rtvs.sk/televizia/archiv/11580/83491#1532>

[2] Programovanie Kodu na Slovensku. Dostupné na: <https://koduslovak.wordpress.com/category/sutaz-kodu-cup-20142015/>

[3] Blog inovatívnych učiteľov a študentov Microsoft – Vzdelávame pre budúcnosť. Dostupné na: <http://vzdelavame.azurewebsites.net/kodu/kodu-cup-20162017/>

[4] Fowler, A., Cusack, B.: Enhancing Introductory Programming with Kodu Game Lab: An Explorary Study

[online]. 2011. <https://www.kodugamelab.com/discussion/kodu-community/news-and-features/2013/8/introducing-the-kodu-research-series!/>

[5] Fowler, A., Fristce, T., MacLauren, M.: Kodu Game Lab: a programming environment [online]. The Computer Games Journal. 2012 . <http://kodu.blob.core.windows.net/kodu/Kodu Research Series/tcgj_11_whitsun2012_fowlerea.pdf>

[6] KODU CUP 2016/2017 Slovensko. Dostupné na: <https://education.microsoft.com/gettrained/kodu-cup-slovensko>

Koncept opakovania vo vyučovaní programovania nevidiacich žiakov základnej školy

Repeating in programming learning of the blind pupils of primary school

Natália Kováčová
KZVI FMFI UK
Mlynská dolina
842 48 Bratislava
Slovenská republika
kovacova.naty@gmail.com

ABSTRACT

Notion of repeating, or loops is an important part of algorithmic problem solving in teaching of informatics at the second grade of primary school. In case of visually impaired pupils is difficult to put a concept of repeating to practise as these pupils find troublesome to grasp this concept. In this article we go through the most common mistakes that occur to pupils while solving the tasks focused on repeating and we also list some examples of troublesome/problematic tasks. We also analyse our experiences with introduction of the term repetition using methodics created by us to a new audible programming environment intended for visually impaired students. On base of research observations and interviews we try to understand the issues pupils have with understanding of this concept

Keywords

Loops, programming, blind pupils, computing education.

ABSTRAKT

Pojem opakovanie je dôležitou súčasťou algoritmického riešenia problémov vo vyučovaní informatiky na druhom stupni základnej školy. V prípade nevidiacich žiakov je však zavedenie konceptu opakovania náročné a nevidiaci žiaci majú viacero problémov pri pochopení tohto konceptu.

V článku sa budeme zaoberať chybami, ktoré sa pri riešení úloh zameraných na opakovanie často objavujú, uvedieme tiež rôzne príklady problematických úloh. Rozoberieme naše skúsenosti so zavádzaním pojmu opakovanie prostredníctvom nami vytvorenej metodiky k novému programátorskému prostrediu určenému pre nevidiacich žiakov. Na základe pozorovaní a rozhovorov sa snažíme porozumieť problémom, ktoré majú žiaci s pochopením konceptu opakovania.

Kľúčové slová

Opakovanie, programovanie, nevidiaci žiaci, informatika.

1. ÚVOD

Algoritmické riešenie problémov je neodmysliteľnou súčasťou školskej informatiky. V rámci tejto oblasti je potrebné, aby si žiaci osvojili programátorské zručnosti, pričom sa za dôležitý považuje aj koncept cyklu.

V literatúre je cyklus nazývaný aj ako iteratívna konštrukcia. Je považovaná za skutočnú podstatu sily počítača. Ak by iterácie neboli možné, programy by boli obrovské a neflexibilné a vyžadovali by si neprimerane veľké množstvo času na ich vytvorenie a beh, ako aj obrovské množstvo pamäte na uloženie [1].

Cyklus, resp. opakovanie s pevným počtom opakovaní definuje Armoni [2] nasledovne: *Cyklus s pevným počtom opakovaní spôsobí, že sa postupnosť príkazov vykonáva dookola. Počet opakovaní je vopred nastavený v príkaze cyklu.*

Podľa štátneho vzdelávacieho programu z roku 2014 [3] by mali žiaci pred ukončením šiesteho ročníka získať nasledujúce schopnosti:

- rozpoznať opakujúce sa vzory pri riešení zadaného problému,
- rozpoznať, aká časť algoritmu sa vykoná pred, počas a po skončení cyklu,
- stanoviť počet opakovaní pomocou hodnoty,
- riešiť problémy, ktoré vyžadujú známy počet opakovaní,
- zapísať riešenie problému s cyklom pomocou jazyka,
- interpretovať algoritmy s cyklami.

Žiaci by si mali osvojiť pojmy opakovanie, počet opakovaní, telo cyklu.

Keďže ide o veľmi dôležitú konštrukciu, je potrebné, aby si ju žiaci osvojili správne [4].

V prípade intaktných žiakov existuje viacero programovacích jazykov, v ktorých môžu využívať konštrukcie opakovania. Pre nevidiacich žiakov sú bežné programovacie prostredia neprístupné, pretože sú zvyčajne ovládané prostredníctvom počítačovej myši a nie sú kompatibilné s čítačmi obrazovky[5]. Je teda potrebné využívať rôzne iné spôsoby, ako zaradiť tento dôležitý koncept do vyučovania informatiky nevidiacich žiakov.

2. SÚŤAŽ IBOBOR

Informatickú súťaž iBobor[6] netreba špeciálne predstavovať. Je jednou z najobľúbenejších súťaží, ktorej sa zúčastňujú žiaci základných a stredných škôl, pričom súťaž je určená pre všetkých bez ohľadu na vedomosti a zručnosti. Počas našej viacročnej práce s nevidiacimi žiakmi sme navrhli a realizovali štyri ročníky súťaže pre kategóriu nevidiacich žiakov[7] [8], keďže súťaž v bežnom

formáte nie je pre nevidiacich žiakov prístupná[9]. Pri tvorbe úloh sme sa snažili pokryť všetky oblasti informatiky. Oblasť algoritmického riešenia problémov nie je výnimkou. V nasledujúcich riadkoch priblížime tri súťažné úlohy, ktoré boli zamerané na opakovanie.

2.1 Súťažné úlohy zamerané na opakovanie

Úloha **Kresliaci robot** (Tabuľka 1) sa objavila medzi súťažnými úlohami v roku 2013. Úspešnosť riešenia tejto úlohy bola pomerne nízka – z celkového počtu sedem žiakov odpovedali správne dvaja žiaci. Ďalší dvaja úlohu neriešili, ostatní traja žiaci odpovedali nesprávne. Na rozdiel od nasledujúcej úlohy Korytnačka Kasandra, sa v tomto prípade robot nielen pohybuje, ale aj otáča v príslušnom smere. Úloha je teda zameraná na ovládanie pomocou relatívneho pohybu robota. Problematike programovania postavičky žiakmi základnej školy pomocou relatívneho a absolútneho pohybu sa vo svojej práci venovala Budinská [10]. Zaujímavé je, že tri nesprávne odpovede boli rôzne (D, A, C). Z toho môžeme usúdiť, že úloha bola pre žiakov náročná a nevedeli, ako by ju mali riešiť.

Tabuľka 1. Zadanie úlohy Kresliaci robot

Kresliaci robot pozná príkazy:

- krok – urobí krok smerom, ktorým je natočený
- vpravo – otočí sa na mieste vpravo,
- vľavo – otočí sa na mieste vľavo,
- opakuj číslo [príkaz príkaz ...] – robot zopakuje príkazy v zátvorkách toľkokrát, koľko udáva číslo pred zátvorkou

Ktorým z týchto príkazov nakreslí robot obdĺžnik?
A: opakuj 4 [krok vľavo]
B: opakuj 2 [krok krok vľavo krok vľavo]
C: opakuj 2 [krok krok vľavo krok vpravo]
D: opakuj 2 [krok krok vľavo krok krok vpravo]

Tabuľka 2. Zadanie úlohy Korytnačka Kasandra

Korytnačka Kasandra žije na svojom pozemku veľkosti 4 krát 4 políčka. Každý večer musí prejsť celý svoj pozemok, aby skontrolovala úrodu. Začína na políčku, ktoré je v ľavom hornom rohu pozemku (pozemok si predstavte ako tabuľku) a postupuje podľa zadaných inštrukcií. Každá inštrukcia obsahuje slovo a číslo. Slovo udáva smer, ktorým sa má Kasandra posunúť a číslo určuje, o koľko políčok sa má v danom smere posunúť. Napríklad príkaz vľavo 4 znamená, že sa má posunúť o 4 políčka vľavo.

Vyber postupnosť štyroch inštrukcií, ktorú má Kasandra dvakrát zopakovať, aby prešla celý pozemok.

A: vpravo 4, dole 1, vľavo 4, dole 1

B: dole 4, vpravo 1, hore 4, vpravo 1

C: vpravo 3, dole 1,vľavo 3, dole 1

D: vpravo 3,dole1, vľavo 3, hore 1

Úloha **Korytnačka Kasandra** (Tabuľka 2) bola zameraná na ovládanie pomocou absolútneho pohybu robota. To znamená, že korytnačka sa neotáča, len sa pohybuje v smere, ktorý je určený príkazom. Túto úlohu vyriešil správne iba jeden žiak z celkového počtu ôsmich zúčastnených žiakov. Dvaja žiaci úlohu neriešili vôbec, ostatní žiaci odpovedali nesprávne. Spomedzi piatich nesprávnych odpovedí sa až v štyroch prípadoch objavila odpoveď D. Z analýzy rozhovorov so žiakmi, realizovanými bezprostredne po odovzdaní riešení, sme zistili, že bolo pre nich náročné predstaviť si pozemok, po ktorom sa pohybuje korytnačka. Jedna žiačka, ktorá postupne testovala všetky možnosti, nepokladala za správnu ani jednu z nich, ale tvrdila nám, že rozumela, ako sa korytnačka pohybuje. Viacerí žiaci tvrdili, že si skúšali pomôcť pohybom prsta na lavici alebo klávesnici, ale aj napriek tomu sa im úloha zdala náročná.

Tabuľka 3. Zadanie úlohy Bobrie nory

Tri bobry hrabú bobriu noru. Každý z nich sa špecializuje na iný typ nory: Anna hrabe nízke nory s dĺžkou 5 metrov, Benjamín hrabe nízke nory s dĺžkou 10 metrov, Cyril sa špecializuje na hrabanie vysokých nor. Bobry sa pri hrabaní nory striedali v rovnakom poradí a nakoniec mala ich nora nasledujúce úseky:

nízka s dĺžkou 5 metrov,

vysoká,

nízka s dĺžkou 15 metrov,

vysoká,

nízka s dĺžkou 15 metrov,

vysoká,

nízka s dĺžkou 10 metrov.

V akom poradí sa bobry striedali pri hrabaní nory?

A: Cyril – Benjamín – Anna

B: Anna – Benjamín – Cyril

C: Benjamín – Cyril – Anna

D: Anna – Cyril –Benjamín

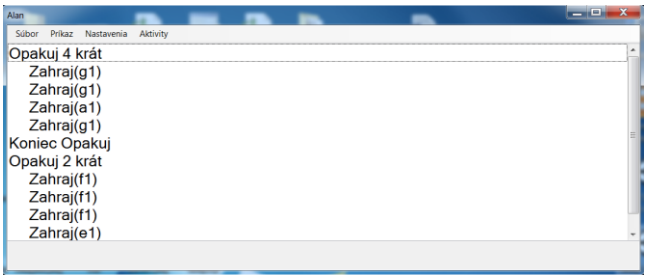
Úloha **Bobrie nory** (Tabuľka 3) bola pomerne úspešnou úlohou. Aj keď sme ju zaradili medzi náročné úlohy a v sade úloh sa nachádzala až na konci, kde je vyšší predpoklad, že túto úlohu žiaci nestihnú vyriešiť, šesť z ôsmich žiakov ju vyriešilo správne. Iba dvaja žiaci označili nesprávnu odpoveď. Obe nesprávne odpovede však boli rovnaké – žiaci označili možnosť C. Aj napriek tomu, že táto odpoveď nie je správna, žiaci pochopili, že prostredný bobor musí byť Cyril, ktorý kope vysoké nory. Neporozumeli však, že na začiatku musí byť kopanie kratšej, päťmetrovej nory. Môžeme si však všimnúť, že takáto úloha, pri riešení ktorej nie je potrebné si predstavovať pohyb postavičky v dvojrozmernom priestore, ale len lineárny sled objektov alebo udalostí, je pre nevidiacich žiakov obľúbenejšia a intuitívnejšia.

Na základe našich pozorovaní a rozhovorov s učiteľkou informatiky, ktorá pozná žiakov dlhšie a pracuje s nimi na hodinách informatiky niekoľko rokov vyplynulo, že úlohy zamerané na ovládanie objektu v štvorcovej sieti, sú pre nevidiacich žiakov veľmi náročné a natoľko abstraktné, že majú problémy s ich riešením. Preto nepovažujeme za vhodné, aby sme príkaz opakovania zavádzali do vyučovania prostredníctvom takéhoto typu úloh. Za vhodnejšiu alternatívu pre nevidiacich žiakov, ktorí sa s programovaním na informatike ešte nestretli, považujeme programovanie pomocou zvukov, ktoré sme so žiakmi vyskúšali a naše skúsenosti popíšeme v nasledujúcej časti.

3. PROSTREDIE ALAN

3.1 Popis prostredia

Prostredie ALAN vzniklo v rámci bakalárskej práce[11].Cieľom práce bolo vytvoriť prostredie, v ktorom by nevidiaci žiaci získali základné programátorské zručnosti prostredníctvom programovania postupnosti zvukov. Prostredie používa jednoduchý programovací jazyk s dvomi základnými príkazmi: *Zahraj* a *Povedz*. Príkaz *Zahraj* prehrá zvuk, ktorý si žiak môže zvoliť z viacerých kategórií – Hudba, Noty, Príbeh, Zvieratá. Učiteľ však môže do ponuky pridať ľubovoľné ďalšie zvuky. Príkaz *Povedz* zabezpečí, že program prečíta zadaný text. Okrem toho program ponúka možnosť použiť cyklus s pevným počtom opakovaní (príkaz *Opakuj*). Ďalej môžeme v programe využívať premenné, príkaz vetvenia a vytvárať podprogramy. Prostredie bolo vytvárané primárne pre nevidiacich žiakov, takže sa dá ovládať aj pomocou klávesnice. Žiadne iné prostredie s podobnými vlastnosťami nepoznáme, preto sme ho považovali za najlepšiu alternatívu pre potreby našej cieľovej skupiny žiakov.



Obrázok 1. Okno programu Alan[11]

3.2 Realizácia vyučovacích hodín

K prostrediu Alan sme vytvorili metodický materiál, ktorý sme v dvoch iteráciách overili zakaždým s inou skupinou nevidiacich žiakov piateho ročníka ZŠ. V rámci prvej iterácie sme viedli hodiny informatiky my za prítomnosti učiteľky, ktorá do vyučovacieho procesu nezasahovala, respektíve iba v prípade technických problémov. Odučili sme štyri vyučovacie hodiny, počas ktorých sme sa zamerali primárne na odhalenie a riešenie problémov pri ovládaní prostredia, ako aj pri riešení nami navrhnutých úloh. Spätná väzba od žiakov a učiteľky pomohla autorovi prostredia odhaliť niekoľko chýb, ktoré následne odstránil. Nám zasa analýza žiackych riešení pomohla upraviť jednotlivé úlohy tak, aby boli pre žiakov zrozumiteľné a aby sme pomocou nich dôsledne plnili ciele stanovené na začiatku vyučovacej hodiny.

V ďalšom školskom roku prebehla druhá iterácia overovania metodiky. Upravenú metodiku sme overili so žiakmi piateho ročníka v aktuálnom školskom roku, pričom tento raz sme boli v úlohe pozorovateľa. Vyučovacie hodiny viedla učiteľka informatiky, my sme si viedli písomnú a obrazovú dokumentáciu z pozorovaní. Na základe získanej spätnej väzby sme metodiku opäť upravili. V školskom roku 2015/2016 sme pracovali so skupinou štyroch nevidiacich žiakov piateho ročníka, v aktuálnom školskom roku sme mali k dispozícií menšiu skupinu, išlo o dve nevidiace žiačky piateho ročníka .

V tomto školskom roku plánujeme ešte realizovať tretiu iteráciu overovania metodického materiálu. V tejto iterácii by podľa neho vyučoval iný učiteľ informatiky na inej škole, pričom nepoznáme, na akej programátorskej úrovni sú jeho nevidiaci žiaci, s akými postupmi sa doteraz stretli, aké prostredia poznajú a používajú. Predpokladáme, že tretia iterácia overenia metodického materiálu nám dá veľmi cennú spätnú väzbu. Očakávame, že budeme vedieť úlohy a návody pre jednotlivé vyučovacie hodiny upraviť tak, aby ich mohol viesť akýkoľvek učiteľ informatiky bez potreby konzultovať s nami. Budeme tiež vedieť lepšie posúdiť, či sú nami navrhované úlohy a postupy vhodné pre žiakov, ktorí sa s programovaním iba zoznamujú.

Náš metodický materiál počíta s časovou dotáciou štyroch vyučovacích hodín. Počas prvých dvoch vyučovacích hodín je cieľom oboznámiť žiakov s možnosťami prostredia a prostredníctvom úloh si precvičiť použitie dvoch základných príkazov jazyka. Až keď žiaci získajú základné zručnosti s používaním týchto príkazov a s ovládaním prostredia, zaradíme počas tretej vyučovacej hodiny úlohy zamerané na opakovanie (cyklus). Cieľom štvrtej vyučovacej hodiny je naučiť sa hľadať chybu v programe. Na každej hodine pritom pracujeme so zručnosťami, ktoré žiaci získajú na predchádzajúcich hodinách, aby si ich dostatočne osvojili.

3.3 Vyučovacia hodina zameraná na použitie opakovania

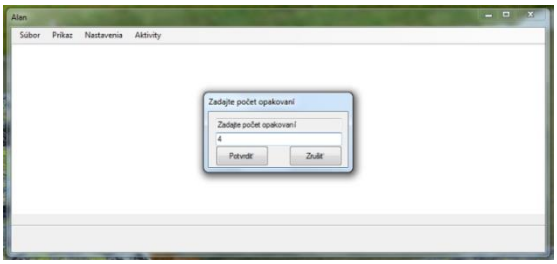
Cieľom nášho pozorovania počas tretej a štvrtej vyučovacej hodiny bolo sledovať, ako žiaci rozumejú konceptu opakovania, ktorý bol pre nich nový. Všímali sme si, ako reagujú na jednotlivé úlohy, ale aj to, či vedia rozlíšiť, v akých prípadoch je vhodné využiť opakovanie a na čo im bude užitočné.

Počas druhej iterácie overovania metodiky na začiatku tretej hodiny učiteľka diskutovala so žiakmi o tom, či sa bežne stretávajú s opakovaním nejakej činnosti. Žiaci spomenuli napríklad navliekanie šnúrok do topánok – tam musia zopakovať niekoľkokrát prevliekanie do jednotlivých dierok. keďže si žiaci počas predošlých hodín obľúbili zvuky zvierat a radi s nimi pracovali, zaradili sme nasledujúcu úlohu: *Mame mačke sa narodili mačiatka. Je ich veľmi veľa, až 15 a všetky mňaukajú. Skúste vymyslieť taký program, v ktorom bude počuť postupne mňaukanie všetkých mačiatok.*

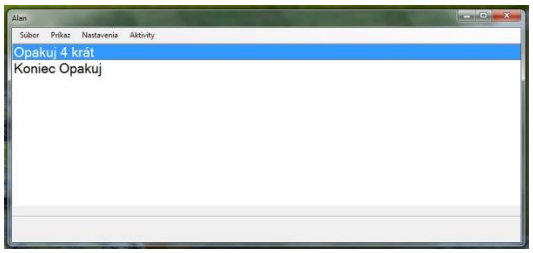
Už po zadaní úlohy sa jedna žiačka ohradila, že je to strašne veľa mačiatok a bude trvať veľmi dlho, kým taký program vytvorí. Aj napriek tomu to skúsila, no po chvíli sa prestala orientovať vo vytváranom programe. „Je to strašne veľa, už neviem, či ich tam mám 15, asi som sa pomýlila.“Druhá žiačka zatiaľ pomaly pracovala a pozorne pridávala príkazy. Musíme však spomenúť, že skôr spomenutá žiačka nemá počítačové zručnosti na porovnateľnej úrovni, preto jej rovnaká činnosť trvala dlhšie než druhej žiačke.

Po úvodnej úlohe sa jedna zo žiačok opýtala, na čo je vlastne v programe medzi príkazmi „to Opakuj“. Vtedy učiteľka prezradila, že takýto príkaz by nám uľahčil prácu a vedeli by sme úlohu o mačke a mačiatkach zapísať jednoduchšie a rýchlejšie. Učiteľka postupne začala vysvetľovať, ako sa takýto príkaz v prostredí Alan používa. Najprv bolo potrebné vybrať z ponuky príkazov príkaz *Opakuj*. Po potvrdení voľby sa zobrazilo dialógové okno, v ktorom bolo potrebné zadať počet opakovaní, teda koľkokrát chceme, aby sa niečo opakovalo. (Obrázok 2).

Následne sa v postupnosti príkazov objavili dva nové riadky predstavujúce konštrukciu opakovania (Obrázok 3). Tu je dôležité podotknúť, že príkazy sa v prostredí Alan pridávajú za označený riadok. To znamená, že príkazy, ktoré chceme opakovať, musíme vkladať za riadok *Opakuj 4 krát.* S vkladaním príkazov sa už žiaci stretli na predošlých hodinách.

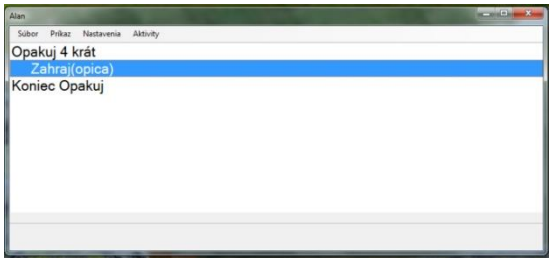


Obrázok 2. Dialógové okno Zadajte počet opakovaní



Obrázok 3. Ukážka vloženia príkazu cyklu do programu

Po vložení základnej konštrukcie opakovania sme teda mohli vložiť príkaz, ktorý mal program vykonať viackrát (Obrázok 4).



Obrázok 4. Príkaz Opakuj s vloženým príkazom v tele cyklu

Po vysvetlení použitia príkazu opakovania mali žiačky samostatne vyriešiť úlohu s mačkou a mačiatkami pomocou tohto príkazu.

Žiaci mali neskôr vytvárať program, v ktorom okrem mačky s mačiatkami bude počuť aj kozu s piatimi kozliatkami. Pridať ďalší príkaz opakovania nerobilo žiakom problém. Zložitejšie na pochopenie bolo pre žiakov použitie viacerých príkazov v tele cyklu. Jedna žiačka tomu porozumela hneď, druhá mala problém pochopiť, ako sa tieto príkazy vykonávajú. Podľa nej sa najskôr niekoľkokrát vykoná jeden príkaz a až potom sa vykoná druhý príkaz s rovnakým počtom opakovaní. Keď si ale prehrala svoj vytvorený program, pochopila, ako funguje opakovanie viacerých príkazov. Usúdili sme, že je potrebné precvičiť podobné úlohy aj neskôr a pravidelne sa k nim vracať.

Problémy sme postrehli aj pri odstraňovaní príkazov z tela cyklu. Myslíme si však, že v tomto prípade išlo skôr o problém s ovládaním samotného prostredia. Bolo totiž potrebné nastaviť sa na odstraňovaný riadok a žiaci po jeho vymazaní nedostali žiadnu zvukovú spätnú väzbu. Museli teda vždy skontrolovať všetky príkazy programu a uistiť sa, či vymazali správny riadok. Pri rozhovore so žiakmi sme sa ale presvedčili, že rozumejú, čo sa stane, ak niektorý príkaz z opakovania vymažú.

Počas overovania metodiky v oboch iteráciách nám viacerí žiaci kládli aj prirodzenú otázku, „Čo sa stane, ak použijeme opakovanie v opakovaní?“. Nechali sme žiakov vyskúšať si to, avšak upozornili sme ich, že takéto programy sú príliš zložité a Alan (tak žiaci volali prostredie aj na hodinách) by nemusel zvládnuť takéto programy vykonať. V programe totiž konštrukcia vnoreného cyklu ešte nefungovala správne, tak sme nechceli žiakov učiť takéto konštrukcie používať.

V prvej iterácií overovania metodického materiálu žiaci vytvárali programy, v ktorých pomocou príkazu *Opakuj* vytvárali hudobnú zvučku pre školskú súťaž. Musíme podotknúť, že viacero žiakov malo hudobné nadanie, preto pre nich boli podobné úlohy pútavé. Žiaci komponovali zvučky pomocou krátkych zvukov reprezentujúcich rôzne hudobné nástroje, využívali tiež zvuk základných tónov stupnice. Žiaci sa sami dožadovali širšej ponuky tónov, na ktoré sú zvyknutí z hudobnej výchovy, respektíve z krúžkov zameraných na hru na hudobné nástroje.

Obrázok 5. Žiačka počas práce s prostredím Alan

Počas štvrtej vyučovacej hodiny, ktorá bola zameraná na hľadanie a opravovanie chýb v hotových programoch, sme tiež využili úlohu s opakovaním. Žiaci mali zmeniť parametre niekoľkých príkazov, a opraviť počet opakovaní v prvom cykle. Nabádali sme žiakov, aby si najprv program prečítali a následne si ho vypočuli a snažili sa odhaliť chyby. Program vyzeral nasledovne:

Povedz(Včera som počúval výbornú hudobnú súťaž.)
Povedz(Na začiatku zaznel štyrikrát zvuk gongu.)
Opakuj 3 krát
  Zahraj(Gong)
Koniec Opakuj
Povedz(Potom prišiel prvý hudobník a zahral krátke sólo na gitare.)
Zahraj(bicie)
Povedz(Potom sa tam vystriedalo veľa nadaných súťažiacich.)
Povedz(Zaznela gitara, organ i saxofón.)
Zahraj(gitara)
Zahraj(saxofón)
Povedz(Na záver celej súťaže, keď sa vyhlásil víťaz, ešte trikrát zaznel zvuk krásnej harfy.)
Opakuj 3 krát
  Zahraj(bicie)
Koniec Opakuj

Žiakom nerobilo problém opraviť chyby v príkazoch mimo cyklu – ak sa mal zahrať zvuk gitary, ale program zahral iný zvuk, pôvodný príkaz zmazali a nahradili ho novým. V prípade opravovania chýb v konštrukciách opakovania sme zaznamenali niekoľko problémov. V prípade, keď bolo potrebné zmeniť počet opakovaní, bolo totiž nevyhnutné celý príkaz opakovania zmazať a vytvoriť podobnú konštrukciu znovu (prostredie Alan zatiaľ neumožňuje upraviť iba počet opakovaní). Žiaci tak ale strácali prehľad o tom, na akom mieste v programe sa cyklus nachádzal a čo všetko patrilo do tela cyklu. V druhom prípade, kedy bolo treba upraviť príkazy v tele cyklu, sa žiaci zorientovali lepšie. Po zmazaní príkazu totiž v programe zostávala samotná konštrukcia cyklu (*Opakuj x-krát, Koniec Opakuj*), takže žiaci presne vedeli určiť miesto, na ktoré museli doplniť nový, správny príkaz.

Na základe týchto skúseností by sme v budúcnosti rozšírili metodiku o ďalšiu vyučovaciu hodinu, počas ktorej by sme sa zamerali na hĺbkové zopakovanie získaných vedomostí.

4. ZÁVER

V rámci našej dlhodobej práce s nevidiacimi žiakmi sme vyskúšali tri typy úloh zameraných na opakovanie:

- Úlohy na programovanie hmatateľného objektu bez použitia počítača – pri nich je nevyhnutné, aby mali žiaci k dispozícii reliéfnu podložku (napr. mriežku), na ktorej by vedeli simulovať pohyb objektu. Skúsenosti [12] ale ukazujú, že takýto prístup je vhodnejší pre žiakov, ktorí majú zachované vizuálne predstavy.
- Úlohy súťaže iBobor – pri nich majú žiaci k dispozícii iba textový editor, ale naše skúsenosti ukazujú, že ho nevedia efektívne využiť na zápis poznámok počas pomocných simulácií.
- Úlohy v zvukovom programovacom prostredí. Toto prostredie sa nám zatiaľ javí ako najvhodnejšie pre všetkých nevidiacich žiakov.

V budúcnosti plánujeme využiť aj hmatový programovací jazyk Torino [13], ktorý je v súčasnosti pilotne vyvíjaný pre nevidiacich žiakov. Obsahuje hmatovo odlišiteľné diely predstavujúce jednotlivé príkazy na prehrávanie rôznych zvukov, ktoré jednoduchým spájaním vytvárajú sekvencie príkazov (Obrázok 6).

Obrázok 6. Hmatový programovací jazyk Torino[1]

Hmatový programovací jazyk Torino je ďalšou vhodnou alternatívou na sprístupnenie programovania nevidiacim žiakom. Podobne ako prostredie Alan umožňuje použitie komplikovanejších konštrukcií, akými sú cyklus s pevným počtom opakovaní, príkaz vetvenia a iné.

BIBLIOGRAFICKÉ ODKAZY

[1] SEBESTA, R. W. *Concepts of ProgrammingLanguages* (8th edition). Pearson International Edition, 2008. ISBN 0-321-50968-4.

[2] ARMONI, M. , BEN-ARI, M. *Computer Science Concepts in Scratch*[online]. [cit. 2017-01-30]. Dostupné na: https://stwww1.weizmann.ac.il/scratch/wp-content/uploads/sites/7/2016/04/scratch-14-textbook-1-0-one-side.pdf

[3] Inovovaný Rámcový vzdelávací program pre predmet informatika [online]. [cit. 2017-01-30]. Dostupné na: http://www.statpedu.sk/sites/default/files/dokumenty/inovovany-statny-vzdelavaci-program/informatika_nsv_2014.pdf

[4] SALANCI, Ľ., TOMCSÁNYIOVÁ, M., BLAHO, A. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika. Didaktika programovania 1.* 2010. ISBN 978-80-8118-037-8.

[5] JAŠKOVÁ, Ľ., KALIAKOVÁ, M. *Programovacie prostredia pre deti s poruchami zraku*. In: Didinfo 2014,LOVÁSZOVÁ, G. (Eds.), Banská Bystrica : Univerzita Mateja Bela, 2014. ISBN 978-80-557-0698-6.

[6] Bebras [online]. [cit. 2017-01-30]. Dostupnéna: http://bebras.org/.

[7] JAŠKOVÁ, Ľ., KOVÁČOVÁ, N. *Prvé skúsenosti s realizáciousúťaže iBobor pre nevidiacich žiakov druhého stupňa ZŠ*. In: Didinfo 2015, Trajteľ, Ľ. (Eds.), Banská Bystrica :Univerzita Mateja Bela, 2014. ISBN 978-80-557-0852-2

[8] KOVÁČOVÁ, N., JAŠKOVÁ, Ľ. *Porovnanie úspešnosti nevidiacich a intaktných žiakov pri riešení úloh súťaže iBobor*. In: TRAJTEĽ, Ľ. (Eds.) Didinfo 2016, Banská Bystrica :Univerzita Mateja Bela, 2014. ISBN 978-80-557-1082-2

[9] Vyučovanie informatiky pre nevidiacich. [online]. [cit. 2017-01-30]. Dostupnéna: http://vin.edi.fmph.uniba.sk/

[10] BUDINSKÁ, L. *Absolútny a relatívny pohyb vo vyučovaní programovania.*[Záverečná práca doplňujúceho pedagogického štúdia]. FMFI UK, Bratislava. 2016.

[11] KOVÁČ, M. *Programovacie prostredie pre zrakovo postihnutých žiakov ZŠ*. [Bakalárska práca]. FMFI UK, Bratislava. 2016.

[12] JAŠKOVÁ, Ľ., KALIAKOVÁ, M. *Kognitívne schopnosti nevidiacich žiakov druhého stupňa ZŠ a ich využitie pri vyučovaní informatiky*. In: TRAJTEĽ, Ľ. (Eds.) Didinfo 2016, Banská Bystrica :Univerzita Mateja Bela, 2014. ISBN 978-80-557-1082-2.

[13] Torino. A physical programming language inclusive of blind pupils. [online]. [cit. 2017-01-30]. Dostupnéna: https://hxd.research.microsoft.com/work/torino.php

[1]https://hxd.research.microsoft.com/work/torino.php

Laboratórny model deterministického konečného automatu

Laboratory Model of Deterministic Finite Automata

Václav Králík
Katolícka univerzita Ružomberok
Hrabovská cesta
034 01 Ružomberok
Slovensko
vaclav.kralik@ku.sk

Patrik Sitiarik
Katolícka univerzita Ružomberok
Hrabovská cesta
034 01 Ružomberok
Slovensko
sitiaricek@gmail.com

ABSTRACT

Fundamentals of theoretical informatics are integral parts of university curricula intended for preparation of teachers of informatics. Among other topics, the finite automata are exhibited to students in the respective courses. They regularly serve in recognition of strings of characters (elements of particular languages). This is the reason why one can find an implementation of a finite automaton in every compiler of a programming language. The explication and clarification of the working principles of the finite automata is often based on the reserve block scheme of a hardware – software machine. These were commonly used in the 1970s-1980s to load and save programmes to or from computers, e.g. EC1033. Since these devices are no longer used, the "Z" generation of IT experts has no opportunity to see these machines in operation. Some of the terms that the students encounter, such as tape, a tape-reading head, etc., rather formally for them. The paper brings a description of a proposal for a laboratory model of a finite automaton as well as a procedure of its creation. It would like to point to the possibility of a proposal for and the production of a teaching aid for the visualization of the basic concepts and activities of the finite automata using a LEGO building set.

Keywords

Deterministic Finite Automata. Model Automata. Lego.

ABSTRAKT

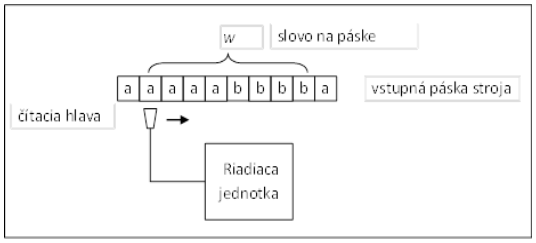
Študijné programy na vysokých školách, ktoré pripravujú budúcich učiteľov informatiky, obsahujú predmety, v ktorých sa rozoberajú základy teoretickej informatiky. V nich sa študenti oboznamujú okrem iného aj s princípmi činnosti konečných automatov (KA). Tieto nám slúžia okrem iného aj na rozpoznanie, reťazcov, jazykov, tvoria súčasť prekladačov a svojou činnosťou sú dôležité pre správnu činnosť programovacích jazykov. Pri výklade a objasňovaní princípov práce KA sa v niektorej literatúre vychádza z náhradnej blokovej schémy hardvérovo – softvérového stroja. Tieto sa používali v 70-80tych rokoch minulého storočia k načítaniu programov z diernej pásky a k ukladaniu programov v počítačoch napr. EC1033. Generácia informatikov "Z" už nemá možnosť tieto stroje, ktoré používali dierne pásky vidieť v činnosti, tieto zariadenia sa už nepoužívajú. Niektoré pojmy s ktorými sa študenti stretnú ako sú napríklad páska, čítacia hlava a iné sú pre nich skôr formálnym pojmom. Chceme poukázať na možnosť návrhu a výroby didaktickej pomôcky pre vizualizáciu základných pojmov a činnosti konečných automatov pomocou stavebnice LEGO.

Kľúčové slová

Deterministický konečný automat. Model automatu. Lego.

1. ÚVOD

Rýchly vývoj v oblasti informačných technológií spôsobuje, že v priebehu niekoľkých rokov dochádza k rýchlej obmene hardvérových ale aj softvérových prostriedkov a technológií. Niektoré systémy koncepčne zastarajú natoľko, že sa technológia v praxi prestane používať. S týmto problémom sa stretávame v prípade didaktických pomôcok používaných pri výučbe konečných automatov. V literárnych zdrojoch napr. [1,2] nachádzame náhradnú principiálnu schému páskových automatov ako je na obrázku 1.



Obrázok 1. Náhradná bloková schéma konečného automatu

Je otázne, či táto forma didaktickej pomôcky akou je bloková schéma je pre generáciu „Z" prínosom v prípade, ak si nevedia predstaviť za touto schémou reálny systém z praxe.

2. MODEL DETERMINISTICKÉHO AUTOMATU

Pri popise činnosti deterministického konečného automatu (DKA), sa často opakovali otázky študentov typu: „...existuje také zariadenie?" „....ako ta páska vlastne funguje?" „...akým spôsobom hlava číta symbol?" a mnoho podobných. Na základe týchto otázok bolo jasné, že náhradná bloková schéma ako pomôcka nestačí pre správne pochopenie problematiky. Hľadali sme preto možnosti, ako vytvoriť didaktickú pomôcku - laboratórny model deterministického konečného automatu, na ktorom by bolo možné prakticky ukázať jeho činnosť. Vizualizácia problému s použitím laboratórneho modelu zapojí do procesu vnímania a učenia aj ďalšie zmysly, ako je napríklad hmat. Na hardvérovej časti modelu bude možné prakticky ukázať princíp činnosti jednotlivých mechanických častí stroja. Spolu so

softvérom bude možné vykonávať zmeny v nastavení automatu a následne sledovať činnosť modelu DKA. Predpokladáme, že tento postup výkladu látky spolu podnetmi z praktickej práce zosilnia pamäťovú väzbu na preberanú problematiku. Pre realizáciu modelu DKA sme sa rozhodli použiť stavebnice Lego Mindstorm. Tieto stavebnice používame na katedre informatiky Katolíckej univerzity v Ružomberku pri výučbe predmetu programovanie mikrokontrolerov. Obsahom predmetu je práca s edukatívnou stavebnicou LEGO. Budúci učitelia sa oboznamujú s postupmi ako navrhovať, zostrojiť a programovať robotov. Použitie stavebnice na vytvorenie didaktickej pomôcky za týmto účelom sa ukazuje ako vhodné pre možnosť prepojenia hardvérového a softvérového vybavenia do jedného funkčného celku. Pri riešení problematiky konštrukcie DKA sme postupovali v dvoch krokoch. V prvom sme riešili mechanickú časť modelu, v druhom kroku softvér pre riadenie činnosti.

2.1 Hardvérová časť modelu DKA

Zamerali sme sa hľadanie spôsobu, akým spôsobom by bolo možné navrhovať mechanické časti DKA. Celkový dizajn zariadenia, ako aj návrhy jednotlivých častí modelu, boli zrealizované v programe Lego Digital Designer (LDD).

Obrázok 2 Mechanická časť modelu

Toto virtuálne prostredie, podporuje skladanie mechanických častí trojdimenzionálne na obrazovke počítača. Podobný spôsob návrhu zariadení za podpory 3D programov sa dnes realizuje aj v inžinierskej praxi. Týmto postupom môžeme navrhnúť a zostrojiť mechanickú časť modelu aj bez toho, aby sme mali k dispozícii stavebnicu. Po zostavení mechanických dielov do celku program umožňuje vytvoriť video návod stavby. Finálna podoba nami navrhnutého modelu DKA je na obrázku 3.

Obrázok 3. Model DKA zo stavebnice LEGO

Podávacia lišta modelu, ako je na obrázku 2 zabezpečuje proces načítania symbolov abecedy z pásky do pamäti mikrokontrolera. Pojem „páska“ je v modeli zrealizovaný priestorom, v ktorom sú v poradí za sebou uložené farebné kocky. Pojem „čítacia hlava“ prezentujeme v modeli optickým snímačom, ktorý na základe zmeranej hodnoty načíta hodnotu farby kocky, ktorá sa práve nachádza pod snímačom. Pojmu „symbol“ odpovedá farba kocky. Mechanizmus posuvu symbolov a optický snímač zabezpečujú postupné načítavanie reťazca.

2.2 Softvérová časť modelu DKA

V druhej časti realizácie modelu automatu sme vytvorili algoritmus a následne program, ktorý zabezpečuje riadenie činnosti modelu DKA. Pri ich tvorbe sme vychádzali z predpokladu, že máme úplný konečný automat, ktorý sa nachádza v počiatočnej konfigurácii. Na jeho vstup prichádzajú vstupné symboly z abecedy Σ. Automat reaguje na vstupné symboly zmenou stavu podľa prechodovej funkcie δ. V novom stave automat opäť prijíma symboly a reaguje podľa prechodových funkcií. Ak sa automat dostane do koncového stavu po prečítaní všetkých symbolov reťazca (slova), potom hovoríme, že slovo bolo prijaté automatom. Ak sa automat nedostane do koncového stavu, ale sú prečítané všetky symboly slova, potom slovo, ktoré bolo na vstupe nebolo prijaté automatom. Matematický popis DKA je vyjadrený a popísaný päticou ako je uvedené v definícii (1).

$$A = \{Q, \Sigma, \delta, q_0, F\} \quad (1)$$

kde Q .... je konečná, neprázdna množina stavov konečného automatu,
Σ .... je konečná neprázdna množina vstupných symbolov, vstupná abeceda,
δ .... je zobrazenie $\delta: Q \times \Sigma \rightarrow Q$, prechodová funkcia automatu,
q_0 .... je počiatočný stav automatu,
F .... je množina koncových stavov automatu.

Z matematického popisu deterministického konečného automatu rovnice (1) vyplývajú požiadavky pre algoritmus riadiaceho programu. Program, ktorý je určený pre riadenie činnosti modelu DKA obsahuje kroky, ktoré zobrazujú stav riešeného slová. Program je navrhnutý tak, aby o výsledku, či slovo bolo prijaté, alebo zamietnuté modelom DKA bol študent informovaný pomocou displeja a zvukového znamenia. Nami zrealizovaný program rozpozná slovo o dĺžke tri symboly. Ich poradie je možné meniť pri spustení programu. Príkladom, ako sa dá využiť model DKA ako didaktická pomôcka je úloha, pri ktorej vyučujúci zadá do riadiaceho programu slovo, ktoré sa skladá z troch symbolov. Úloha pre študentov je pomocou modelu automatu nájsť správne poradie symbolov. Informáciu o stave či je symbol prijatý, alebo nie sa zobrazuje na displeji modelu DKA. Ak sú symboly zoradené v správnom poradí, po prečítaní je slovo označené ako slovo prijaté, akceptované. Výsledný program je realizovaný v programovacom jazyku Brick C++, podporujúci prácu s mikrokontrolerom stavebnice LEGO Mindstorm. Vytvorený program a jeho grafický zápis pomocou vývojového diagramu používame pri výučbe danej problematiky ako motiváciu a vzorový príklad riešenia.

3. ZHRNUTIE

V teoretických predmetoch, ktoré rozoberajú princípy práce konečných automatov, považujeme za prínos možnosť vizualizácie činnosti mechanických častí a softvérového riešenia. Laboratórny model DKA podporuje teoretický pojmový aparát. Pri výpočtovej činnosti modelu DKA, študenti vidia postupnosť krokov práce DKA na modelu zariadenia. Za prínos modelu považujeme aj možnosť meniť parametre modelu, ako sú napríklad zmena počiatočnej konfigurácie modelu DKA, zmena prechodových funkcií, alebo zmena vstupného slova. Vytvorený model je možné ďalej modifikovať. Napríklad mechanickú časť je možné rozšíriť o zásobník typu LIFO, spolu s mechanikou

a softvérom potom môžeme simulovať činnosť zásobníkového automatu.

4. DISKUSIA

Pri hľadaní didaktických pomôcok výučby automatov nás zaujal freeware jflap, viz. link [6]. Tento program umožňuje napríklad graficky zadať, zobraziť a vykonávať výpočty s rôznymi automatmi.. Našim pôvodným zámerom však bolo napodobniť činnosť hardvérovo-softvérového stroja, ktorý používa na uchovávanie programu diernu papierovú pásku. Didaktickú pomôcku model DKA je v budúcnosti možné vytvoriť vo virtuálnom prostredí, pomôcka potom nebude závislá od dostupnosti hardvéru stavebnice LEGO.

BIBLIOGRAFICKÉ ODKAZY

[1] **HOPCROFT J. E., ULLMAN J. D.** *Formal Languages and Their Relation to Automata.* Addison-Wesley, Reading (Massachusetts), 1969

[2] **HOPCROFT, J.E., MOTWANI, R., ULLMAN, J.D.**: *Introduction to Automata Theory, Languages, and Computation*, Addison Wesley, 2. vyd., 2000

[3] **LEWIS, H. R., PAPADIMITRIOU, C. H.:** *Elements of the Theory of Computation.* Prentice Hall, Upper SaddleRiver (NJ), 1998

[4] **DEDERA, Ľ.** *Počítačové jazyky a ich spracovanie*. Vojenská akadémia, Liptovský Mikuláš, 2014

[5] **VAVREČKOVÁ, Š.** *Teorie jazyků a automatů:* . Opava: Slezská univerzita v Opavě.. Ústav informatiky, 2007, 67 s.: http://axpsu.fpf.slu.cz/~vav10ui/obsahy/tja/skripta/tja_celek.pdf

[6] http://www.jflap.org

Informatika s tabletmi
Informatics with Tablets

Gabriela Lovászová
Katedra informatiky
FPV UKF v Nitre
Tr. A. Hlinku 1
949 74 Nitra
Slovensko
glovaszova@ukf.sk

Martin Cápay
Katedra informatiky
FPV UKF v Nitre
Tr. A. Hlinku 1
949 74 Nitra
Slovensko
mcapay@ukf.sk

Viera Michaličková
Katedra informatiky
FPV UKF v Nitre
Tr. A. Hlinku 1
949 74 Nitra
Slovensko
vmichalickova@ukf.sk

ABSTRACT
In this paper, results of the project Mobile Technology in Schools for the 21st Century supported by KEGA research agency of Slovak Ministry of Education are presented. The project's objects arise from actual demand for providing teachers in Slovak schools with professional and methodical know-how to use mobile technology with their pupils successfully. The project's most important results include the original conception of meaningful using of mobile technology in secondary education together with various types of verified learning activities (mostly concerned with informatics education). All activities developed within the project enhance new forms and methods typical for the constructivist teaching and learning theories.

Keywords
Tablet. Location-based activities. Programming of mobile applications. Robotics. Student response systems.

ABSTRAKT
V príspevku predstavujeme výsledky trojročného projektu Mobilné technológie v škole pre 21. storočie podporeného Kultúrnou a edukačnou grantovou agentúrou MŠVVaŠ SR. Projekt vychádzal z aktuálnej potreby budovania odborného a metodického know-how pre úspešné využívanie mobilných digitálnych technológií v školskej praxi. Hlavným výsledkom riešenia projektu je koncepcia zmysluplného využívania mobilných technológií v sekundárnom vzdelávaní s overenými typmi učebných aktivít (najmä v predmete informatika), ktoré podporujú moderné formy a metódy konštruktivistického vyučovania.

Kľúčové slová
Tablet. Geolokačné aktivity. Programovanie mobilných aplikácií. Robotika. Hlasovacie systémy.

1. ÚVOD
Mobilné digitálne zariadenia sa väčšinou používajú na pripojenie k internetu v teréne, na prehliadanie a prehrávanie multimediálnych dokumentov (čítanie textov, počúvanie zvukových súborov, sledovanie videa, prehliadanie obrázkov), v spojení s prijímačom signálu GPS na navigáciu v neznámom prostredí, v spojení s digitálnym fotoaparátom alebo kamerou na zaznamenávanie a spracovanie obrazu, na zaznamenávanie, uchovávanie, prenos a výmenu najrôznejších údajov v teréne.

V kontexte vzdelávania je známe využitie mobilných zariadení formou m-learning. Ide hlavne o sprostredkovanie vzdelávacieho obsahu v mobilnom zariadení s možnosťou komunikácie s lektorom. Tento model využitia mobilných zariadení považujeme za vhodný skôr pre dospelých študentov, ktorí si sami organizujú čas a priestor svojho štúdia.

Na sekundárnom stupni vzdelávania nedospelých žiakov je stupeň organizácie štúdia zo strany školy a učiteľa podstatne vyšší. Tu vidíme priestor pre využitie mobilných digitálnych zariadení iným spôsobom, ktorý je zameraný viac na riešenie problémov ako na organizáciu štúdia. Mobilné technológie nám dávajú možnosť realizovať moderné vyučovanie v otvorenej škole, ktorá poskytuje flexibilný priestor pre vzdelávanie (vo vnútri aj mimo budovy školy) a vytvára podnetné prostredie pre učenie sa.

Projekt *Mobilné technológie v škole pre 21. storočie* podporený Kultúrnou a edukačnou grantovou agentúrou MŠVVaŠ SR riešený na Katedre informatiky FPV UKF v Nitre v období 2014-2016 bol zameraný na výskum možností využívania mobilných digitálnych zariadení (tablety, smartfóny, GPS navigátory) vo vyučovaní na sekundárnom stupni vzdelávania. Cieľom projektu bolo analyzovať stav využívania mobilných digitálnych zariadení vo vyučovacom procese, navrhnúť a overiť také učebné aktivity s využitím mobilných technológií, ktoré podporujú moderné formy a metódy konštruktivistického vyučovania a vytvoriť originálnu koncepciu zmysluplného využívania mobilných technológií v školskom vzdelávaní hlavne v predmete informatika.

2. TABLET AKO UNIVERZÁLNY NÁSTROJ NA UČENIE SA
Začiatky mobilných technológií sú spojené s mobilným telefonovaním. Mobilný telefón umožňuje ľuďom byť v kontakte s ostatnými vždy a všade, nielen v mieste, kde sa nachádza telefónny prístroj. Mobilný telefón je personalizovaný, viazaný na osobu svojho majiteľa, ktorý má k nemu osobný a väčšinou pozitívny vzťah.

Podobný vývoj sme zažili aj s používaním počítača. Osobný počítač slúži svojmu majiteľovi ako univerzálny nástroj na prácu aj zábavu a jeho majiteľ ho chce mať vždy poruke. To umožnili prenosné počítače zmenšené do malých (až vreckových) rozmerov s bezdrôtovým pripojením na internet a s bezdrôtovým zdrojom energie. Tablet alebo smartfón sa uplatňuje ako všadeprítomný osobný digitálny asistent a pomocník. Je preto prirodzené, ak tieto univerzálne a v osobnom živote veľmi užitočné technologické nástroje dostanú svoje miesto aj vo vyučovaní.

2.1 Vlastnosti tabletov

Mobilné technológie môžu priniesť zásadnú zmenu vo vyučovaní vďaka vlastnostiam, ktorými sa odlišujú od štandardných stolných počítačov:

- Vďaka *prenosnosti* vieme prejsť od príležitostného používania počítačov a výpočtovej techniky v špeciálnych počítačových učebniach k bežnému používaniu v triede aj mimo triedy.
- *Osobný charakter* mobilného zariadenia prináša veľký potenciál vo zvyšovaní motivácie žiakov: Používanie nástrojov, ku ktorým majú žiaci pozitívny vzťah a používajú ich v osobnom živote, posúva učenie sa bližšie k reálnemu svetu, je viac autentické, podobné riešeniu bežných problémov mimo školy.
- *Nové spôsoby ovládania* dotykmi prstov po obrazovke, pohybom, hlasom sú bezprostrednejšie ako u klasických počítačov, približujú používanie tabletov aj menším deťom, umožňujú nové spôsoby práce a nové typy aktivít.
- *Bezdrôtová komunikácia* a voľnosť pohybu so zariadením podporujú rôzne formy kolaborácie vo virtuálnom aj v reálnom prostredí.
- *Vstavané senzory* umožňujú zachytávať a priamo v mobilnom zariadení spracovávať rôzne typy digitálnych dát.

V tabuľke 1 sumarizujeme silné a slabé stránky tabletov, príležitosti na obohatenie vzdelávania, ktoré ich používanie môže priniesť, ale aj problémy, ktoré môžu učenie sa s tabletmi ovplyvniť negatívne.

Tabuľka 1. Vlastnosti tabletov ako nástrojov na učenie sa

Silné stránky	**Slabé stránky**
• prenosnosť • malé rozmery a hmotnosť • rýchly štart • bezdrôtové technológie • dotykové ovládanie • vstavané senzory • multimédiá	• výdrž batérie • malý displej • obmedzená čitateľnosť v exteriéri pri slnečnom svetle • softvérová klávesnica
Príležitosti	**Problémy**
• decentralizácia učenia sa: kedykoľvek, kdekoľvek • prirodzenejšie, bezprostrednejšie ovládanie vhodné aj pre malé deti • nové spôsoby používania: písanie rukou, kreslenie, zaznamenávanie dát, počúvanie audia, pozeranie videa • kolaboratívne učenie sa • cloud computing	• technické problémy: s Wi-Fi pripojením, s dobíjaním batérie • nekvalifikovaní učitelia • nevhodná metodika výučby zameraná na pasívne prijímanie digitálneho obsahu • odvádzanie pozornosti

2.2 Využitie tabletov vo vyučovaní informatiky

Pre vyučovanie informatiky je špecifické, že na rozdiel od iných školských predmetov sa uskutočňuje väčšinou v špeciálnej počítačovej učebni. Prínos mobilných zariadení vidíme tam, kde nielen nahrádzajú počítač s často menším komfortom používania (menší displej, softvérová klávesnica), ale umožňujú nové a zmysluplné spôsoby použitia, aké nie sú možné so štandardnými počítačmi alebo sú s tabletmi komfortnejšie.

Na základe našich skúseností z realizácie projektu klasifikujeme spôsoby využitia tabletov vo vyučovaní informatiky do kategórií uvedených v tabuľke 2.

Tabuľka 2. Využitie tabletov vo vyučovaní informatiky

Zber a spracovanie dát
zber a vyhodnocovanie dát rôzneho typu, spracovanie dát vo forme atraktívnych výstupov
Interaktívna učebnica a poznámkový zošit
práca s digitálnymi učebnými zdrojmi, tvorba poznámok bohatých na mediálne prvky, zdieľanie, spolupráca, komunikácia
Geolokačné hry
exteriérové orientačné hry typu hľadanie pokladu, počítačové hry reagujúce na polohu hráča v teréne
Programovanie
používanie mobilných aplikácií na rozvoj algoritmického myslenia, programovanie mobilných aplikácií
Robotika
ovládanie robotov tabletmi, programovanie robotov pomocou tabletov
Získavanie odozvy
testovanie vedomostí, kladenie otázok v reálnom čase, zber nápadov, hlasovanie, prieskumy názorov

3. TYPY AKTIVÍT S TABLETMI

V rámci projektu sme navrhli a v praxi overili s rôznymi cieľovými skupinami nasledujúce typy učebných aktivít s tabletmi:

- tvorba a spracovanie digitálnej fotografie a videa, tvorba 3D fotografie, rozšírená realita,
- exteriérové aktivity s GPS: geocaching, reverse caching, geolokačné hry Wherigo, GPS drawing,
- edukačné programovanie mobilných aplikácií,
- robotika s tabletmi s využitím robotov Lego MindStorms EV3, Ozozobot Bit 2.0,
- aktivity s hlasovacím systémom a s tabletmi ako hlasovacími zariadeniami.

Navrhované učebné aktivity sme overovali s rôznymi cieľovými skupinami:

- na popularizačných akciách Vedecký deň detí pre deti primárneho a predprimárneho vzdelávania a Vedecký jarmok pre žiakov základných a stredných škôl a verejnosť,
- každoročne v týždňovom detskom letnom tábore ITečk@r s informatickým zameraním pre deti vo veku 8-14 rokov,
- na workshopoch pre žiakov základných škôl na pôde univerzity alebo v domovskej základnej škole,

- na prednáškach a workshopoch pre študentov stredných škôl organizovaných na pôde univerzity alebo v domovskej strednej škole,
- na prednáškach a seminároch so študentmi učiteľstva informatiky na univerzite.

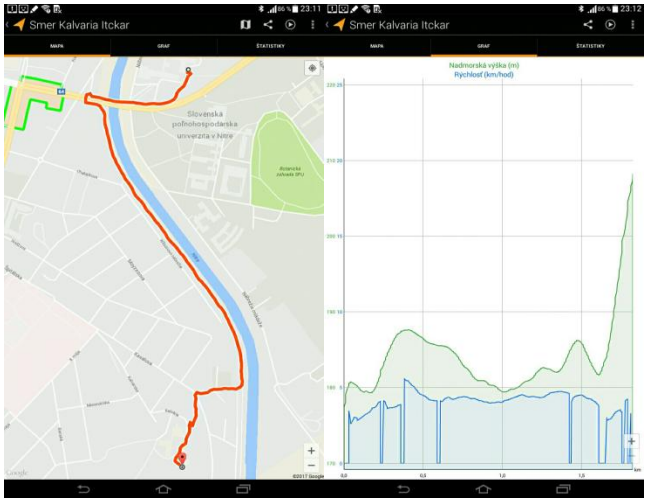
Skúsenosti s realizáciou aktivít s mobilnými zariadeniami, navrhovanú metodiku, odporúčania a zhodnotenie edukačného prínosu sme publikovali v prípadových štúdiách v učebnici [1].

3.1 Zber a spracovanie dát

V časti 2.1 sme uviedli niekoľko vlastností tabletov, ktorými sa odlišujú od štandardných stolných počítačov a vďaka ktorým je prínosné ich používať na hodinách informatiky namiesto alebo popri počítačoch v počítačovej učebni. Prenosnosť tabletov a vstavané senzory umožňujú zaznamenávať kdekoľvek (aj mimo počítačovej učebne alebo budovy školy) dáta rôzneho typu (fotografie, video, audio, záznamy o polohe) a spracovávať ich priamo v tablete bez potreby prenášania do počítača.

Na nasledujúcich obrázkoch 1, 2, 3 sú zábery z denného letného tábora: tvorba vlastnej 3D fotografie, záznam trasy pomocou GPS, hra s QR kódom. Počas týždňového tábora deti zozbierali množstvo podobných zaujímavých dát, ktoré v záverečný deň spracovávali vo forme videodenníka, prezentovali si navzájom a svojim rodičom.

Obrázok 1. Tvorba vlastnej 3D fotografie



Obrázok 2. Spracovanie záznamu trasy

Obrázok 3. Aktivita s QR kódmi

3.2 Geolokačné hry

Hra je činnosť, ktorá príjemným spôsobom zamestnáva hráča. Jej dôležitým atribútom je samoúčelnosť – človek sa hrá kvôli procesu hrania, nie kvôli výsledku (efektu). Vo vzdelávaní sa často používa metóda hry, aby sa využila vnútorná motivácia hrať sa na dosiahnutie cieľov vyučovania. Ak učiteľ vhodne dokáže ovplyvniť efekt hry, žiaci sa niečo naučia príjemným a prirodzeným spôsobom.

Rozšírenie mobilných zariadení s geolokačným senzorom podnietilo rozvoj nového typu počítačových hier založených na polohe hráča. Niektoré takéto hry a hrové aktivity (*Geocaching*, *Wherigo*, *GPS drawing*) a ich didaktický potenciál pre vyučovanie informatiky sme predstavili na konferencii Didinfo v roku 2012 [2]. Okrem spomínaných hier sa veľkej obľube medzi hráčmi v súčasnosti tešia masívne on-line geolokačné hry s rozšírenou realitou. Populárna je napríklad hra *Pokémon Go*.

Dôležitými vlastnosťami geolokačných hier, ktoré majú predpoklady pre efektívne využitie vo vzdelávaní, sú tvorivosť a otvorenosť. Otvorenosť umožňuje učiteľovi prispôsobovať hru edukačným cieľom. Hry Geocaching a Wherigo sú otvorené. Ich základnú koncepciu a technologické zázemie poskytuje komerčná firma, avšak samotné hry sú výsledkom tvorivých nápadov a kolaborácie miliónov hráčov. Preto sa dajú prispôsobiť edukačným cieľom a hrať rôznymi spôsobmi, konzumne aj tvorivo, pre zábavu aj pre učenie sa.

GPS drawing (kreslenie s GPS) je tvorivá aktivita spájajúca výtvarné umenie, fyzický pohyb v teréne a používanie mobilných digitálnych technológií. Je vhodná ako námet pre projektové vyučovanie, ktorého výsledkom je zaujímavý obrázok vytvorený zaznamenaním pohybu v teréne pomocou mobilného zariadenia s GPS (obrázok 4).

Masívne on-line geolokačné hry (napr. Pokémon Go) sú uzatvorené. Obsah hry je pripravený vývojármi hry, hráč sa nepodieľa na jej tvorbe, učiteľ nemá možnosť prispôsobiť hru svojím cieľom. Možnosti využitia uzatvorených hier na edukačné ciele sú obmedzené.

Spoločenský charakter hier a existencia hráčskej komunity zvyšujú motiváciu žiakov. Naopak niektoré technické požiadavky, ako je potreba (mobilného) pripojenia na internet, môžu využitie geolokačných hier v škole obmedzovať.

Obrázok 4. GPS kresba

Hranie geolokačných hier, hlavne tých, ktoré fungujú on-line v hráčskej komunite, prináša rovnaké hrozby ako pobyt vo virtuálnom svete na internete. Pri ich hraní však treba byť ešte opatrnejší, pretože sa odohrávajú tiež v reálnom svete vonku mimo chráneného priestoru školy alebo domova.

3.3 Programovanie

Rozumieť princípu automatického spracovania informácií ako výpočtovému procesu nad digitálnymi dátami, ktorý prebieha na základe algoritmov, je jedným z hlavných cieľov vyučovania informatiky. Pre priblíženie tohto princípu je ideálne, aby žiaci (v duchu konštruktivizmu) získali vlastné skúsenosti s tvorbou a interpretovaním algoritmov, prípadne spoznali proces vývoja softvéru zblízka prácou na malých programátorských projektoch.

Pri vyučovaní algoritmizácie a programovania sa tablet môže použiť ako zariadenie, v ktorom žiaci zostavujú a spúšťajú svoje programy. V on-line obchodoch s mobilnými aplikáciami nájdeme veľa aplikácií, ktoré formou hry rozvíjajú algoritmické myslenie detí. Zvyčajne sa v nich zostavujú sekvencie príkazov pre postavy v hre na splnenie nejakého cieľa. Takéto hry sa dobre hodia na motivovanie detí a na propedeutiku niektorých jednoduchších programátorských konceptov hlavne v neformálnom vzdelávaní. Pre systematickejšie formálne vzdelávanie je však lepšie siahnuť po prostrediach „s vyšším stropom“ a pracovať na počítači (s väčšou zobrazovacou jednotkou a s hardvérovou klávesnicou).

Druhým spôsobom, ako využiť tablety vo výučbe programovania, je programovanie aplikácií určených pre mobilné zariadenia. Tablety a smartfóny sú cieľovou platformou, ktorá prirodzene zvyšuje motiváciu žiakov učiť sa vďaka ich osobnému charakteru (žiaci majú väčšiu radosť z programovania, keď aktívne vytvárajú aplikáciu pre vlastné mobilné zariadenie). Programátorské projekty zamerané na tvorbu mobilných aplikácií sú tiež výbornou príležitosťou pre nové, atraktívne typy zadaní v porovnaní s programovaním aplikácií pre osobný počítač, v ktorých sa využíva voľnosť pohybu, ovládanie dotykom či hlasom, vstupy zo senzorov (fotoaparát, GPS prijímač, akcelerometer, gyroskop a iné).

Vývojové nástroje, ktoré pri tvorbe mobilných aplikácií používajú profesionáli, sú pre žiakov základnej a strednej školy príliš komplexné. Pre potreby edukačného programovania sú vhodnejšie špecializované programovacie prostredia, ktoré boli vyvíjané ako edukačný softvér s dizajnom zameraným na žiaka (napr. MIT App Inventor na tvorbu mobilných aplikácií pre tablety a smartfóny s operačným systémom Android), resp. začiatočníkov so žiadnou alebo minimálnou predchádzajúcou skúsenosťou s programovaním (napr. Urwigo na tvorbu geolokačných hier pre platformu Wherigo). Námety a metodiku edukačného programovania mobilných aplikácií v prostredí MIT App Inventor sme publikovali v učebnici [3].

3.4 Robotika

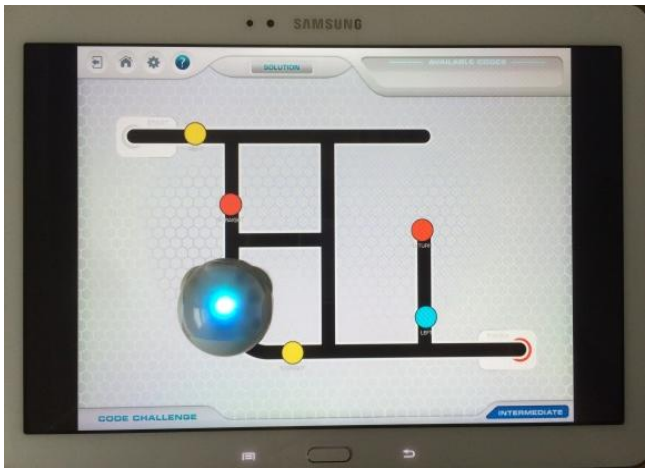
Programovanie v tabletoch je prínosné v takých situáciách, kedy využijeme dobré vlastnosti tabletov (v porovnaní so štandardnými počítačmi). Príkladom je programovanie robotov.

Robotické stavebnice od firmy Lego sú na slovenských školách tradične rozšírené. Pre najnovšie modely Lego MindStorms EV3 existuje okrem programovacieho prostredia pre osobné počítače aj mobilná aplikácia *EV3 Programmer App* pre tablety s operačnými systémami iOS a Android. Použitie tabletu je užitočné vďaka jednoduchej bezdrôtovej komunikácii tabletu s robotom (prostredníctvom technológie bluetooth). Prenosnosť a voľnosť pohybu s tabletom umožňujú, aby žiak mohol programovať v blízkosti robota, s ktorým pracuje (obrázok 5). Programuje sa v ikonickom programovacom prostredí spôsobom drag-and-drop dotykovým ovládaním.

Obrázok 5. Programovanie robota EV3 z tabletu

Veľmi užitočnú podporu ponúkajú tablety pri programovaní robotov Ozobot. *Ozobot* je miniatúrny robot (s rozmermi 2,5×2,5 cm) programovateľný cez unikátny farebný jazyk. Robot rozpoznáva špecifické kombinácie farebných kódov (Ozokódy), na základe ktorých určuje smer, rýchlosť, časovanie, špeciálne pohyby a príkazy na ukončenie činnosti. Môže sa pohybovať po dráhach, na ktorých sníma farbu podkladu. Dráhy s farebnými kódmi pre ovládanie pohybu robota je možné kresliť na papier, ale oveľa praktickejšie je použiť na nakreslenie dráhy mobilnú aplikáciu a nechať robota pohybovať sa priamo po displeji tabletu so zostavenou dráhou (obrázok 6).

Postupnosť farebných kódov je možné vygenerovať aj z programu v jazyku Blockly (OzoBlockly) a nahrať do robota v dávke – robot zosníma sekvenciu farebných zmien jednoducho priložením k displeju. Zaujímavým námetom na programovanie Ozobotov je tvorba tanečnej choreografie. Pomocou mobilnej aplikácie *OzoGroove* vedia žiaci naprogramovať a do robota preniesť sekvenciu „tanečných“ pohybov zosúladenú s hudobným sprievodom.



Obrázok 6. Ozobot na dráhe nakreslenej v tablete

3.5 Získavanie odozvy

Hlasovacie systémy predstavujú modernú digitálnu technológiu, ktorá podporuje aktívne a kolaboratívne učenie sa prostredníctvom učebných aktivít založených najmä na kladení otázok overujúcich úroveň porozumenia, motivujúcich žiakov k produktívnej diskusii, spolupráci, tvorivej činnosti a uvažovaní o vlastnom učení sa.

Okrem špeciálnych hlasovacích systémov poskytujúcich potrebné vlastné hardvérové a softvérové vybavenie existujú softvérové aplikácie, ktoré implementujú funkcionalitu hlasovacieho systému. Ako hlasovacie zariadenia sa používajú bežné výpočtové zariadenia (počítače, tablety, smartfóny) a na komunikáciu existujúca počítačová sieť.

Vo vyučovaní sa hlasovacie systémy najčastejšie používajú na hromadné testovanie vedomostí, avšak tento spôsob použitia zďaleka nevystihuje ich skutočný didaktický potenciál. Rôzne kategórie hlasovacích aktivít (testovanie, kontrolné alebo aktivizačné otázky počas vyučovania, zber nápadov, hlasovanie, prieskumy názorov) ako aj konkrétne námety realizovateľné vo vyučovaní informatiky sme prezentovali na konferencii Didinfo v roku 2015 [4].

4. ZÁVER

Ciele projektu Mobilné technológie pre školu v 21. storočí, ktorý sme riešili na Katedre informatiky FPV UKF v Nitre v uplynulých rokoch, vychádzal z aktuálnej potreby budovania odborného a metodického know-how pre úspešné využívanie mobilných digitálnych technológií v školskej praxi. Najdôležitejším výsledkom riešenia projektu je vytvorenie koncepcie zmysluplného využívania mobilných technológií v sekundárnom vzdelávaní s overenými typmi učebných aktivít (najmä v predmete informatika), ktoré podporujú moderné formy a metódy konštruktivistického vyučovania. Koncepcia je prezentovaná v dvoch knižných publikáciách [1, 3] určených pre našich študentov učiteľstva informatiky. Kvalitná odborná a metodická príprava učiteľov je základným predpokladom zmysluplného a pre dosahovanie výchovno-vzdelávacích cieľov prínosného využívania mobilných technológií vo vyučovaní na sekundárnom stupni vzdelávania. Učebnice sú určené aj pre učiteľov informatiky z praxe, ktorí majú na základných a stredných školách k dispozícii pre žiakov nové atraktívne mobilné technológie v podobe tabletov, navigačných zariadení, hlasovacích zariadení a pod. a privítajú odbornú a metodickú pomoc s ich využívaním vo vyučovaní. Výsledky riešenia projektu boli prezentované učiteľskej verejnosti aj formou prednášok a praktických workshopov na odbornom seminári, na ktorom vystúpila s pozvanou prednáškou L. Rohlíková, autorka úspešnej českej knihy o využívaní tabletov vo vzdelávaní [5]. Prípadové štúdie o učebných aktivitách s využitím mobilných technológií vo vyučovaní informatiky sú publikované na internete a dostupné z webovej stránky projektu [6].

POĎAKOVANIE

Ďakujeme za podporu projektu Mobilné technológie v škole pre 21. storočie (KEGA 055UKF-4/2014) Kultúrnej a edukačnej grantovej agentúre MŠVVaŠ SR.

BIBLIOGRAFICKÉ ODKAZY

[1] LOVÁSZOVÁ, G. a kol. *Mobilné technológie vo vyučovaní informatiky.* Edícia Prírodovedec č. 649. Nitra : UKF, 2016. ISBN 978-80-558-1104-8

[2] LOVÁSZOVÁ, G., PALMÁROVÁ, V. GPS aktivity v školskej informatike. In *DidInfo 2012* [CD]. Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, 2012, s. 129-136. ISBN 978-80-557-0342-8. Dostupné tiež na: http://didinfo.umb.sk/sk/138/zbornik.html

[3] MICHALIČKOVÁ, V. *Programovanie mobilných aplikácií v prostredí MIT App Inventor 2*. Edícia Prírodovedec č. 646. Nitra : UKF, 2016. ISBN 978-80-558-1101-7

[4] LOVÁSZOVÁ, G., MICHALIČKOVÁ, V. Hlasovacie systémy - interaktívna technológia na podporu aktívneho učenia sa. In *DidInfo 2015* [CD]. Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, 2015, s. 102-107. ISBN 978-80-557-0852-2. Dostupné tiež na: http://didinfo.umb.sk/sk/181/zbornik.html

[5] NEUMAJER, O., ROHLÍKOVÁ L., ZOUNEK J.: *Učíme se s tabletem*. Wolters Kluwer, 2015. ISBN 978-80-7478-768-3

[6] KI FPV UKF v NITRE. *Mobilné technológie v škole pre 21. storočie*. Dostupné na: https://www.ki.fpv.ukf.sk/projekty/kega_055_4_2014/index.php?m=1

DIDINFO & DIDACTIG 2017. **Univerzita Mateja Bela, Banská Bystrica 2017.**

Interaktívne učebné materiály vo vyučovaní algoritmizácie
Interactive educational materials for teaching programming

Janka Majherová
Katolícka univerzita v Ružomberku
Hrabovská cesta 1
03401 Ružomberok
Slovenská republika
janka.majherova@ku.sk

Hedviga Petrušková
Katolícka univerzita v Ružomberku
Hrabovská cesta 1
03401 Ružomberok
Slovenská republika
hedviga.palasthy@ku.sk

Pavel Valuška
Základná škola
J. Alexyho 1
960 01 Zvolen 1
Slovenská republika
valuska.p@centrum.sk

ABSTRACT
This article deals with the teaching facilities of algorithms on the subject of informatics in primary and lower secondary education. We introduce an educational program created in Imagine Logo. We further mentioned instruction for teaching in a playful way in an environment of language Scratch Junior. Finally, we describe some interactive online applications within the project Code Studio, which include courses suitable for students from the second year of primary school. The benefit of using such materials is the development of algorithmic thinking of students already in the first grade of primary school as well as the use of school tablet.

Keywords
Algorithmization. Teaching programming. Interactive materials. Primary and lower secondary education.

ABSTRAKT
V príspevku sa venujeme možnostiam výučby algoritmizácie v predmete informatika v primárnom a nižšom sekundárnom vzdelávaní. Predstavíme výučbovú aplikáciu vytvorenú v prostredí Imagine Logo. Ďalej spomenieme videonávody pre výučbu hravou formou v prostredí jazyka Scratch Junior. Nakoniec opíšeme interaktívne online aplikácie v rámci projektu Code Studio, ktoré obsahujú kurzy vhodné pre žiakov od 2. ročníka základnej školy. Prínosom použitia uvedených materiálov a aktivít je rozvoj algoritmického myslenia žiakov už na prvom stupni ZŠ ako aj využitie školských tabletov.

Kľúčové slová
Algoritmizácia. Programovanie. Interaktívne materiály. Primárne vzdelávanie. Nižšie sekundárne vzdelávanie.

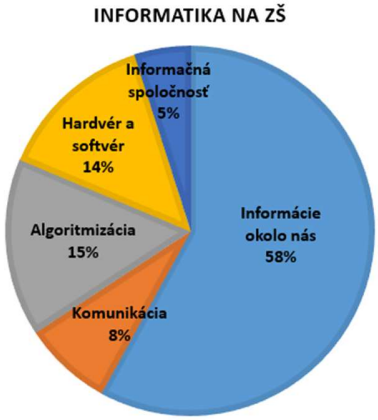
1. ÚVOD
V súčasnosti sa zavádza do praxe inovovaný Štátny vzdelávací program pre primárne a sekundárne vzdelávanie platný od školského roku 2015/2016. Obsah výučby predmetu informatika je podľa inovovaného Štátneho vzdelávacieho programu členený do piatich tematických okruhov:

- Reprezentácie a štruktúry
- Komunikácia a spolupráca
- Algoritmizácia a programovanie
- Softvér a hardvér
- Informačná spoločnosť.

Veľký dôraz v inovovanom ŠVP pre nižšie sekundárne vzdelávanie v predmete informatika sa dáva na tematický okruh Algoritmizácia a programovanie. Obsahový a výkonový štandard je pomerne detailne rozpísaný v [1], [2]. Pre učiteľov boli vytvorené podporné metodické materiály, ktoré obsahujú vzorové úlohy a rozdelenie učiva.

V rámci pedagogickej praxe študentov učiteľstva informatiky na PF KU v Ružomberku už 10 rokov sledujeme rozsah výučby algoritmizácie a programovania v porovnaní s ostatnými tematickými okruhmi. Prvý prieskum sme realizovali v rokoch 2006 - 2008, kde sme porovnali výskyt výučby programovania s výučbou práce s textom, tabuľkami, grafikou a s prezentáciami. Na základných školách sa počas pedagogickej praxe v sledovanom období študenti stretli s algoritmizáciou a programovaním v jazykoch Baltík alebo Imagine len v piatich z pätnástich základných škôl.

V rokoch 2014 - 2016 sme opäť analyzovali pedagogické denníky študentov, ktorí pozorovali výučbu alebo učili v rámci hospitačnej a výstupovej praxe v predmete informatika na základných školách. Z celkového počtu 450 vyučovacích hodín sa najväčší priestor venoval výučbe tematického celku Informácie okolo nás - Reprezentácie a štruktúry (58%). Algoritmizácii a programovaniu bolo venovaných 15% výučby, podobne ako téme Hardvér a softvér (14%) (pozri graf 1).



Graf 1. Podiel tém v učive informatiky na cvičných ZŠ v rokoch 2014 -2016

Učitelia na prvom stupni ZŠ preferovali pre výučbu programovacie jazyky Baltík a Imagine, na druhom stupni používali najmä Imagine [3]. Len ojedinele sa vyskytol jazyk Scratch, práve vďaka

overovaniu metodických materiálov, ktoré boli súčasťou záverečných prác študentov. V inovovanom ŠVP pre druhý stupeň ZŠ odporúčajú autori metodík z informatiky zvýšiť dotáciu pre výučbu algoritmizácie na 25% odučených hodín [4].

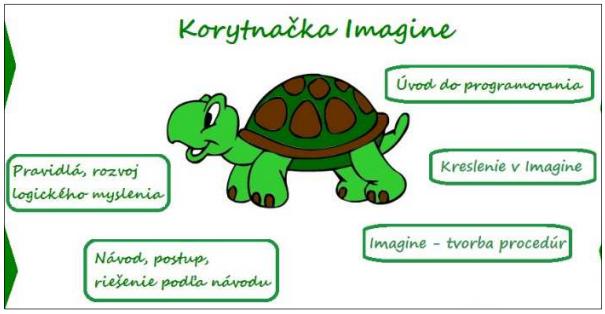
V súčasnosti pozorujeme zvýšený záujem učiteľov a mierny nárast počtu vyučovacích hodín. venovaných algoritmizácii a programovaniu na cvičných základných školách našej fakulty v Ružomberku. Učitelia sa postupne oboznamujú s novými možnosťami výučby a vítajú každú metodickú pomoc. K tomu dúfame prispeje zdieľanie učebných materiálov, možnosť účasti na rôznych workshopoch ako aj webináre zamerané na aktuálne témy (porov. [5], [6]). Dôležité je aj budúcich učiteľov informatiky oboznámiť s týmito trendami.

2. ALGORITMIZÁCIA S IMAGINE

V rámci didaktiky informatiky a v záverečných prácach vedieme študentov učiteľstva informatiky na Pedagogickej fakulte KU v Ružomberku k príprave inovatívnych metodík a učebných pomôcok. S niektorými prácami sa študenti úspešne zapojili aj do študentskej vedeckej činnosti. V rámci Česko-slovenskej vedeckej konferencie v didaktike informatiky 2013 bola úspešná súťažná práca študenta Jána Hruboša Tvorivé programovanie v jazyku Scratch [7]. Práca Martina Stopiaka Zbierka úloh pre programovací jazyk Scratch získala ocenenie na ŠVK v Ostrave v roku 2015. Súčasťou súťažnej práce študentky Anny Molčanovej bola výučbová aplikácia Korytnačka Imagine určená pre žiakov prvého stupňa základnej školy [8].

Aplikácia je rozdelená do piatich tém, ktoré nadväzujú na seba. Je možné s nimi pracovať v jednom ročníku alebo je ich možné rozdeliť do viac ročníkov. Úlohy k jednotlivým témam sú navrhnuté tak, aby zodpovedali vedomostiam žiakov daného veku. Aplikáciu Korytnačka Imagine je nutné vopred uložiť na pevný disk každého počítača, aby žiaci mohli pracovať vlastným tempom.

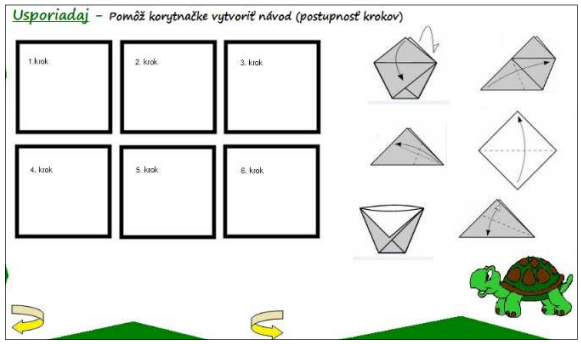
Úlohy v aplikácii sú rozdelené podľa tém nasledovne:
- Pravidlá, rozvoj logického myslenia,
- Návod, postup, riešenie podľa návodu,
- Úvod do programovania,
- Kreslenie v Imagine,
- Imagine – tvorba procedúr (obrázok 1).



Obrázok 1. Úvodná obrazovka programu

Prvé dve témy je možné použiť v rôznych ročníkoch. Ďalšie tri postupy nadväzujú na seba, preto je nutné s nimi pracovať v rámci jedného ročníka.
Tému Pravidlá, rozvoj logického myslenia je možné použiť v 2. a 3. ročníku prvého stupňa ZŠ. V tejto téme sa žiaci oboznamujú s pravidlami, chápu ich význam a pomocou aktivít ako napr. Zoraďovanie predmetov, Skladanie podľa predlohy, Čo pokračuje ďalej si rozvíjajú svoje logické myslenie.

Druhú tému Návod, postup, riešenie podľa návodu je vhodné použiť, keď sa žiaci oboznamujú s návodmi, vytvárajú vlastné algoritmy – postupnosti krokov, pri riešení úloh postupujú podľa návodov. Hlavnou náplňou úloh je to, aby sa žiaci naučili vytvárať návody, s ktorými sa stretávajú v bežnom živote. Žiaci v týchto úlohách presúvajú objekty na určené miesto a tým vytvárajú riešenie danej úlohy (obrázok 2).

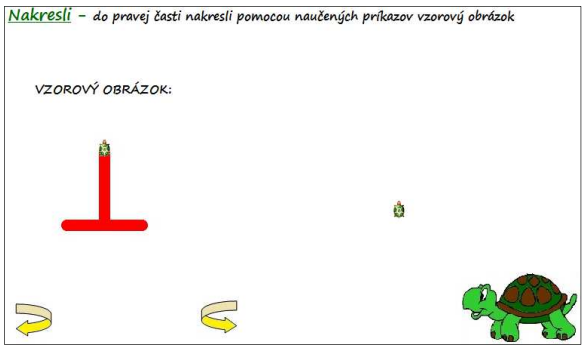


Obrázok 2. Ukážka úlohy Usporiadaj

Použitie tretej témy Úvod do programovania je vhodné zaradiť do vyučovania vo 4. ročníku prvého stupňa, kde žiaci už majú nadobudnuté základné zručnosti v používaní počítača. V tejto téme sa žiaci prvýkrát stretávajú so slovom programovanie. Pomocou úloh vo výučbovom programe Korytnačka Imagine sa oboznamujú s jednoduchým a hravým programovaním. Súčasťou tejto témy je už oboznámenie s prostredím Imagine a so základnými príkazmi.

Štvrtá téma Kreslenie v Imagine nadväzuje na predchádzajúcu tému Úvod do programovania. Žiaci si precvičia základné korytnačie príkazy, preto je možné časť z nej spojiť s predchádzajúcou témou. Jej súčasťou sú aj súbory obrázkov na precvičenie základných korytnačích príkazov a práce s perom (obrázok 3).

Poslednú tému Imagine – tvorba procedúr je vhodné zaradiť do výučby po dostatočnom precvičení predchádzajúcich dvoch tém. Žiaci sa oboznamujú s vytváraním vlastných príkazov prostredníctvom procedúr.



Obrázok 3. Ukážka úlohy Kreslenie obrázkov

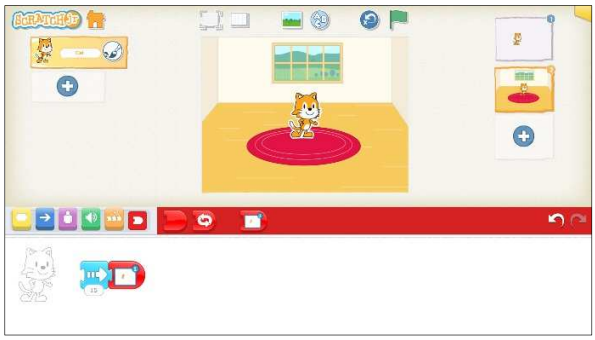
Aplikácia bola overená počas pedagogickej praxe v predmete informatická výchova v školskom roku 2014/2015 na prvom stupni základnej školy. Zamerali sme sa na zrozumiteľnosť navrhnutého výučbového programu a zložitosť jeho ovládania. Program bol pre žiakov zrozumiteľný a jednoducho ovládateľný. Pri väčšine úloh žiakom stačilo iba vysvetliť zadanie ďalších pokynov. Ovládanie programu k nájdeniu správneho riešenia úloh im nerobilo

problémy. Keďže programovací jazyk Imagine sa stále vo výučbe programovania používa, aplikácia Korytnačka Imagine je vhodná ako učebná pomôcka pri výučbe algoritmizácie na hodinách informatiky na prvom stupni.

3. PROGRAMOVANIE NA TABLETE

V rámci skúmania nových metód výučby v rámci prípravy budúcich učiteľov informatiky, sme hľadali vhodné aplikácie pre využitie tabletu na hodine informatiky. Okrem grafických aplikácií a hier sme objavili aj nové programovacie prostredie voľne dostupné pre tablety.

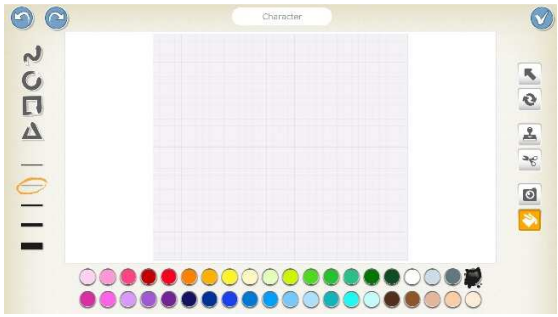
Program Scratch Junior (ScratchJr) je novinkou z prostredia MIT, bol inšpirovaný známym programom Scratch [9]. ScratchJr je vizuálny programovací jazyk, kde programujeme systémom drag and drop (chyť a polož). Výhoda tohto programovacieho jazyka je v jeho jednoduchosti. Je vhodným programovacím prostredím pre deti v predškolskom a mladšom školskom veku (uvedený je vek 5-7 rokov). ScratchJr má intuitívne prostredie a tak ho metódou pokusu a omylu dokážu ovládať deti aj bez toho, aby vedeli čítať alebo písať. Prostredie umožní deťom, aby preskúmali princípy počítačového programovania. Všetky ovládacie prvky, ktoré žiaci používajú pri vytváraní svojho programu sú príkazy, ktoré majú farebné rozloženie s veľkými tlačidlami a sú ľahko ovládateľné s prstom na obrazovke tabletu. Príkazy sa priraďujú k vybraným objektom - ťahaním príkazov do programovacej oblasti vybraného objektu, kde sa môžu príkazy navzájom spájať (obrázok 4).



Obrázok 4. Prostredie programu ScratchJr

ScratchJr je k dispozícii ako bezplatná aplikácia pre iPad a Android tablet. Informácie o programe získame na domovskej stránke http://scratchjr.org. Program má aj svoju stránku na Facebooku, kde je možné nájsť zaujímavé inšpirácie. Výhodou programu ScratchJr je aj to, že po nainštalovaní funguje bez internetového pripojenia a práca s ním je bezpečná (nevyskakujú žiadne reklamy). Na domovskej stránke sú pre učiteľov dostupné metodické materiály a vzorové pracovné listy. Aktivity môžu byť zamerané na prácu s grafikou a animáciami, rozvoj čitateľskej a matematickej gramotnosti. Autori odporúčajú prostredie ScratchJr aj k tvorbe jednoduchých detských hier s použitím postáv a programovacích blokov s náznakom objektového prístupu [10].

Program obsahuje aj jednoduchý editor, podobne ako Scratch. Umožňuje nakresliť vlastný objekt (postavu), s ktorou môžeme potom pracovať na scéne. Grafický editor umožňuje vytvárať okrem objektov aj vlastné pozadia (Obrázok 5). Zaujímavá je aj práca so zvukmi, dieťa si môže nahrať vlastný zvuk k danej postave. Projekt sa automaticky uloží, je možné ho zdieľať prostredníctvom e-mailu.



Obrázok 5. Editor v programe ScratchJr

Okrem oficiálnych materiálov sme pre študentov učiteľstva informatiky ako aj pre učiteľov, ktorí si rozširujú kvalifikáciu pre vyučovanie predmetu, pripravili videonávody, ktoré ponúkajú základné oboznámenie sa s aplikáciou ScratchJr. Slúžia pre rýchlejšie oboznámenie s jednotlivými funkciami, ktoré program obsahuje (obrázok 6). Sú určené aj pre učiteľov, ktorí nemali zatiaľ skúsenosti s daným typom programovania, nepracovali ani s jazykom Scratch.

Obrázok 6. Videonávod pre ScratchJr

K dispozícii je tiež pracovný list s názorným videokomentárom, ktorý už skúsenejšiemu učiteľovi predstaví prostredie programu ScratchJr a jeho základné funkcie. Učiteľ má dostupný materiál na vyučovanie, ktorý môžu žiaci použiť aj pri samoštúdiu (obrázok 7).

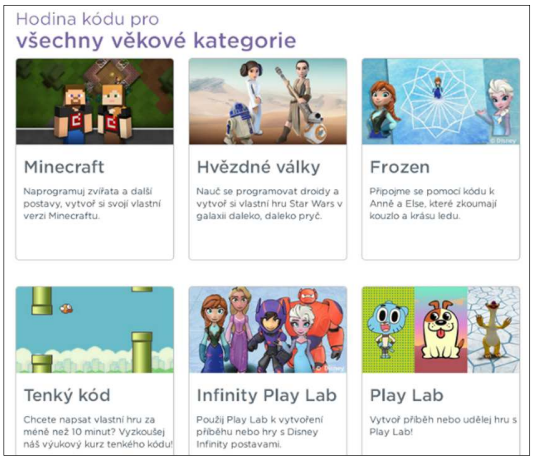
Obrázok 7. Videonávod - pracovný list ScratchJr

Videonávody sú dostupné na stránke http://tablet.jecool.net/ [11]. Plánujeme ich ďalej dopĺňať.

4. ONLINE AKTIVITY

Okrem vyššie spomenutých aplikácií pre výučbu algoritmizácie a programovania na základnej škole je možne sa inšpirovať aj viacerými projektami v rámci tzv. Hodiny kódu [12]. Je to motivačný projekt pre vzbudenie záujmu o informatiku a programovanie už u žiakov základnej školy. Domovská stránka hourofcode.org je dostupná čiastočne aj v češtine (obrázok 8). Ponúka veľký výber interaktívnych kurzov označených podľa ročníkov (od začiatočníkov na 1. stupni ZŠ). Námety kurzov sú veľmi motivujúce pre žiakov, keďže vychádzajú z obľúbených filmových príbehov, napr. Frozen, StarWars a pod. Stránka obsahuje tiež tutoriály pre učiteľa v anglickom jazyku [13].

Študenti učiteľstva informatiky v rámci pedagogickej praxe realizovali s cvičnými učiteľmi počas týždňa od 5. do 9. decembra 2016 niektoré aktivity tohto projektu a stretli sa so záujmom žiakov. Žiaci pracovali na tabletoch. Z postrehov študentov môžeme spomenúť, že žiaci 4. - 5. ročníka dobre zvládli aktivitu Minecraft (úroveň pre 2. ročník ZŠ a vyššie). Žiaci mali problémy pri kódovaní v kurze Frozen (úvod do korytnačej grafiky), keďže vyžadoval vedomosti o uhloch, ktoré získajú až koncom 5. ročníka. V metodike k danej aktivite autori odporúčajú použiť pomocnú tabuľku uhlov. Za účasť na Hodine kódu žiaci dostali aj certifikáty.



Obrázok 8. Aktivity Hodiny kódu

Ďalšia možnosť zapojiť žiakov základnej školy do interaktívnych online kurzov programovania je na stránke code.org, kde sú dostupné mnohé zaujímavé projekty dostupné na stránke studio.code.org [13]. Kurzy sú rozdelené podľa vekovej úrovne žiakov na viac vyučovacích hodín [14]. Pri aktivitách v tomto projekte žiaci zároveň získavajú jazykové kompetencie v anglickom jazyku, čo je vhodné pre použitie metódy CLIL v predmete informatika.

5. ZÁVER

Na základnej škole je vhodné algoritmizáciu a programovanie vyučovať formou hry, motivačnými úlohami primeranými veku žiakov. Žiaci sú prirodzene zvedaví, radi objavujú nové možnosti a pracujú s novými technológiami. Každý žiak má vlastné tempo pre nacvičenie príkazov a riešenie úloh.

Ak chceme, aby sa žiak učil konštruktívne, plánujme vyučovacie hodiny tak, že žiakov aktivizujeme vhodnou motiváciou. Vytvoríme dostatočný priestor na experimentovanie, zbieranie vlastných skúseností. Vhodnými úlohami posúvame žiakov o poznanie ďalej.

Niektorí učitelia informatiky na základných školách majú stále rešpekt pred výučbou algoritmizácie a programovania (nielen nekvalifikovaní). Často pristupuje k tomu evidentný nezáujem žiakov o túto tému, ktorý môže byť spôsobený aj nevhodnou metodikou výučby, „obavou z ťažkého rozmýšľania“ a pod. Riešenie vidíme v tvorbe a propagácii vhodných metodických materiálov a interaktívnych prostredí, motivácii učiteľov k účasti na zaujímavých projektoch a k príprave žiakov na súťaže ako aj kvalitnej didaktickej príprave budúcich učiteľov informatiky.

BIBLIOGRAFICKÉ ODKAZY

[1] *Inovovaný Štátny vzdelávací program.* [online]. [cit. 2017-01-02]. Dostupné na: <http://www.statpedu.sk/sk/Inovovany-Statny-vzdelavaci-program.alej>.

[2] SALANCI, Ľ.: *Informatika v novom Štátnom vzdelávacom programe.* In DidInfo 2014. Banská Bystrica: UMB. ISBN 978-80-557-0698-6, s. 143-148.

[3] MAJHEROVÁ, J. *Inovácie kurikula v informatike na ZŠ.* In *DidInfo 2015.* Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, 2015, s. 191. ISBN 978-80-557-0852-2. Dostupné na: <http://didinfo.umb.sk/sk/138/zbornik.html>

[4] KUČERA, P. *Ukážky zaradenia tém do ročníkov.* [online]. [cit. 2017-01-02]. Dostupné na: <http://www.statpedu.sk/sites/ default/files/dokumenty-old/zavadzanie-isvp-v-zs-a-gym/Pomocne_materialy/informatika/I4.pdf>

[5] JACKOVÁ, J. Learning for Mastery in an Introductory Programming Course. In *ITICSE '08 Proceedings of the 13th annual conference on Innovation and technology in computer science education.* s. 352. New York: ACM.

[6] NAGYOVÁ, I. *Analysis of Students' Behaviour in eLearning Course "Educational Technologies".* In: European Conference on e-Learning Hatfield, UK. ACPI, 2015. s.428-434. ISBN 978-1-910810-70-5

[7] HRUBOŠ, J. *Tvorivé programovanie v jazyku Scratch – metodika.* Práca Česko-slovenskej ŠVK v didaktike informatiky. Bratislava: FMFI, UK, 2013.

[8] MOLČANOVÁ, A. *Tvorba výukovej aplikácie pre výučbu algoritmov na 1. stupni ZŠ*: diplomová práca. Ružomberok: KU, 2015. 84 s.

[9] http://scratchjr.org

[10] *ScratchJr Curricula*. https://www.scratchjr.org/teach.html

[11] http://tablet.jecool.net/?p=83

[12] https://hourofcode.com

[13] https://studio.code.org/

[14] https://code.org/curriculum/course1/1/Teacher.pdf

Analýza riešení úlohy zo súťaže iBobor pre kategóriu Juniori 2016/2017

Analysis of results from the tasks in competition iBobor from category Junior 2016/2017

Karolína Mayerová
KZVI FMFI UK
Mlynská dolina 1
842 48 Bratislava
SR
mayerova@fmph.uniba.sk

Lucia Budinská
KZVI FMFI UK
Mlynská dolina 1
842 48 Bratislava
SR
budinska@fmph.uniba.sk

Michaela Veselovská
KZVI FMFI UK
Mlynská dolina 1
842 48 Bratislava
SR
veselovska@fmph.uniba.sk

Monika Tomcsányiová
KZVI FMFI UK
Mlynská dolina 1
842 48 Bratislava
SR
tomcsanyiova@fmph.uniba.sk

ABSTRACT

In this year there was tenth year of international competition iBobor in Slovakia. There were 62 981 competitors participated in this competition from 921 schools. In this paper we focus on analysis of results from one selected task from competition which called The Tunnel. Mentioned task was included into category Junior in school year 2016/2017 and it was also included into category Cadets in school year 2013/2014 with name The Bridge. We want to exhibit results from both categories and we want to focus mainly on analysis of individual solutions - we focus on answers which were selected by pupils. We are searching reasons, why pupils selected some incorrect answers. In conclusion we compare solutions from 45 pupils, which were solving both of mentioned tasks with a time difference of three years. We also obtain some results from both tasks from abroad, for example Ukraine, Slovenia and Russia.

Keywords

Contest iBobor. Analysis of solutions. Category Junior. Comparison tasks. Comparison to countries abroad.

ABSTRAKT

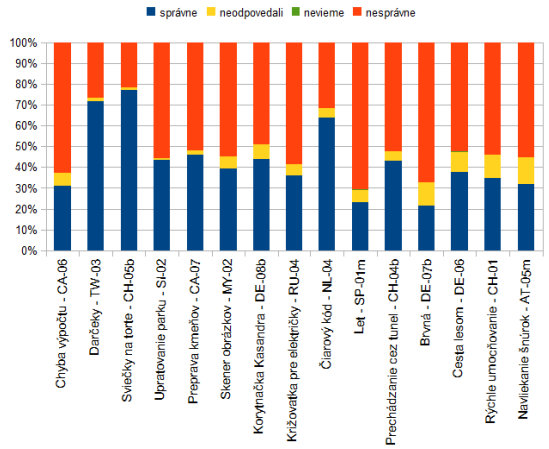
V tomto roku prebehol na Slovensku 10. ročník medzinárodnej súťaže iBobor, ktorého sa zúčastnilo 62 981 súťažiacich z 921 škôl. V tomto príspevku sa zameriame na analýzu výsledkov jednej konkrétnej súťažnej úlohy Tunel, ktorá bola zaradená do súťažnej kategórie Juniori v školskom roku 2016/2017 a do kategórie Kadeti v školskom roku 2013/2014 pod názvom Most. Predostierame výsledky z oboch kategórií a zameriavame sa predovšetkým na analýzu jednotlivých riešení, teda žiakmi zvolených odpovedí. Zamýšľame sa a hľadáme dôvody, prečo si žiaci vybrali niektoré nesprávne odpovede. V závere porovnávame výsledky 45-tich žiakov, ktorí riešili obe tieto úlohy s odstupom troch rokov. K obom úlohám sme získali aj výsledky, ktoré žiaci dosiahli v zahraničí napr.: Ukrajina, Slovinsko a Rusko.

Kľúčové slová

Súťaž iBobor. Analýza riešení. Kategória Juniori. Porovnanie úloh. Porovnanie so zahraničím.

1. ÚVOD

V tomto článku sa chceme venovať analýze dvoch úloh, ktoré sú algoritmicky rovnaké, resp. výsledkom, ktoré dosiahli žiaci pri ich riešení. Keďže úlohy používané v tomto článku sú zo súťaže, ktorá má na Slovensku už desaťročnú tradíciu, podmienky jej priebehu nebudeme podrobne opisovať, dajú sa pozrieť v [1]. Aktuálne informácie o súťaži je možné nájsť aj na webstránke [2], kde je dostupný aj archív úloh z predošlých ročníkov [3], či analýzy výsledkov a informácie o priebehu z predošlých rokov. V minulosti boli v súvislosti so súťažou iBobor na Slovensku publikované viaceré analýzy výsledkov pre rôzne kategórie ako Benjamíni [4], či Seniori [5]. Bola publikovaná aj analýza informatických tém vzhľadom na teoretickú informatiku [6], či celková analýza všetkých kategórií jedného ročníka [7]. V tomto článku sa chceme zamerať na analýzu dvoch konkrétnych úloh, ktoré majú rovnaký algoritmický postup riešenia. Jedná sa o úlohu Tunel, ktorá bola použitá v tomto roku pre kategóriu Juniori a úlohu Most, ktorú riešili Kadeti v školskom roku 2013/14. V danom roku ju riešili aj na Ukrajine a Rusku. V tomto školskom roku 2016/2017 sa konal jubilejný 10. ročník súťaže iBobor. Zúčastnilo sa ho celkovo 62 981 súťažiacich z 921 škôl. Z toho v kategórii Juniori súťažilo 8 179 súťažiacich (4794 bolo chlapcov a 3385 dievčat). Plný počet bodov z nich získalo 21 žiakov a úspešných riešiteľov bolo 1 237, čo je 15,1% z celkového počtu súťažiacich v tejto kategórie. Pre lepší obraz o celkovej úspešnosti žiakov pri jednotlivých úlohách prikladáme Obrázok 1.



Obrázok 1. Diagram úspešnosti jednotlivých úloh v kategórii Juniori 2016/17 [8]

2. Tunel

V tejto kapitole sa chceme bližšie venovať úlohe s názvom: *Prechádzanie cez tunel*, ktorá patrila medzi päť ťažkých úloh v kategórii Juniori. Keďže ide o medzinárodnú súťaž, úlohu sme vybrali zo spoločnej databázy 2016 a jej autormi sú viacerí Švajčiari.

2.1 Zadanie

Presné zadanie úlohy Tunel v našej súťaži znelo:

Štvorčlenná rodina chce prejsť cez tmavý tunel. Cez tunel môžu ísť naraz maximálne dvaja a vždy musia mať baterku. Rodina má len jednu baterku. Jednotlivým členom trvá cesta cez tunel nasledovne:

- *mama – 20 minút*
- *otec – 25 minút*
- *dcéra Anna – 10 minút*
- *syn Bruno – 5 minút*

Keď idú cez tunel dvaja, tak idú rýchlosťou pomalšieho. Za aký ***najkratší*** *čas dokáže prejsť cez tunel celá rodina?*

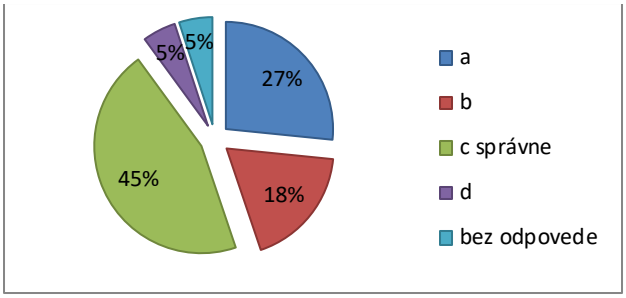
A) 35 minút
B) 45 minút
C) 60 minút
D) 80 minút

2.2 Riešenie

Správna odpoveď bola možnosť C), teda 60 minút. Jedno z možných riešení znie: Pri prvom prechode išla dcéra Anna (10 min.) a syn Bruno (5 min.). Na druhej strane ostal Bruno a k rodičom s baterkou sa vrátila dcéra Anna (10 min.). Teda celkový čas, ktorý zatiaľ „spotrebovali" bol 10+10=20 minút. V druhom prechode išli rodičia. Mame to trvalo 20 min, otcovi 25 min., takže celkový čas na prejdenie bol 25 min. Celkový „spotrebovaný" čas do tohto momentu bol už 45 min. Rodičia už ďalej nešli. Len Bruno sa vrátil s baterkou po Annu, čo bolo ďalších 5 min a spolu preli k rodičom, čo trvalo 10 min. Celkový „spotrebovaný" čas sa teda zvýšil o 15 min a vytvoril tak presne jednu hodinu = 60 min.

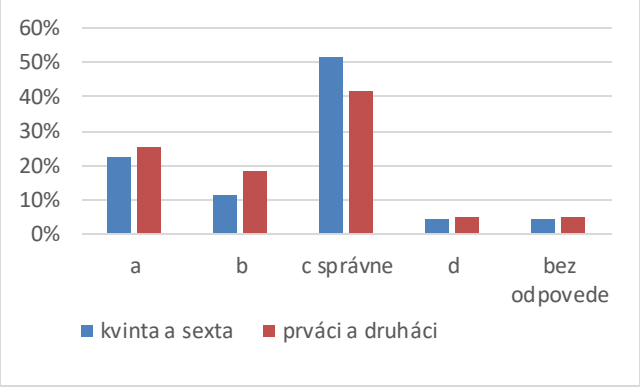
2.3 Analýza

Spôsob riešenia sa dá prirovnať aj k známej logickej hádanke o prievozníkovi, koze, kapuste a vlkovi. Je možné, že niektorí žiaci našli paralelu s touto logickou hádankou v tejto úlohe, čo znamenalo pre nich istú výhodu. Pri riešení úlohy by sa dalo predpokladať, že žiaci nájdu skôr čas vyšší ako 60 min, a teda zvolia možnosť D), ktorá bola 80 min, čiže najväčšiu zo všetkých možností. Napriek tomu túto voľbu zvolilo len 5% súťažiacich, čo je rovnako veľa, ako tých, ktorí si nezvolili žiadnu odpoveď. Prekvapujúco si správnu možnosť C) (60 min) vybralo 45% žiakov, čo je takmer polovica súťažiacich v tejto kategórii. Druhú najčastejšie zvolenú odpoveď, ktorou je možnosť B) (35 min) si zvolilo 27% žiakov. K tomuto číslu mohli dôjsť spočítaním dvoch najdlhších časov, za ktoré mohli prejsť dve dvojice na druhú stranu. Prvá dvojica by v takomto prípade bola: mama a otec a druhá dvojica by bola: Anna a Bruno. V takomto prípade žiaci ale nerátali s vrátením sa jedného bobra na druhú stranu tunela s baterkou. Možnosť B) (45 min) zvolilo 18% žiakov. Tento čas mohli získa spočítaním dvoch najdlhších časov, prípadne len náhodnou voľbou zo štyroch možností. Celkovú úspešnosť riešenia tejto úlohy môžeme vidieť zobrazenú na grafe na Obrázku 2.



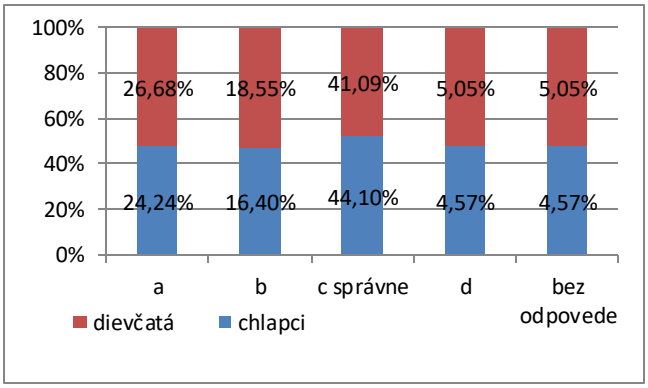
Obrázok 2. Celková úspešnosť riešenia úlohy Tunel v kategórii Juniori 2016/17

Zaujímavý je aj porovnanie odpovedí súťažiacich z kvinty a sexty (osemročných gymnázií) ak ich porovnáme s odpoveďami prvákov a druhákov (stredných škôl). Na Obrázku 3. je znázornená úspešnosť týchto dvoch skupín. Vidíme tu, že viac ako polovica žiakov osemročných gymnázií zvolila správnu odpoveď. Čo je približne o 10% viac ako žiakov stredných škôl. A vo všetkých ostatných možnostiach dosiahli tiež nižšie hodnoty.



Obrázok 3. Porovnanie riešení žiakov kvinty a sexty s prvákmi a druhákmi úlohy Tunel

Ak by sme sa pozreli na rozdiel medzi úspešnosťou chlapcov a dievčat, teda či existuje nejaká závislosť od pohlavia v tejto úlohe (Obrázok 4.), tak vidíme, že ich odpovede sú približne rovnaké. Pri správnej odpovedi je vidieť nepatrný rozdiel troch percent, čo reprezentuje približne 245 žiakov v prospech chlapcov. To je počet, o ktorý boli chlapci úspešnejší ako dievčatá. Výraznejšie rozdiely sme však nezaznamenali.



Obrázok 4. Porovnanie odpovedí na základe pohlavia žiakov v úlohe Tunel

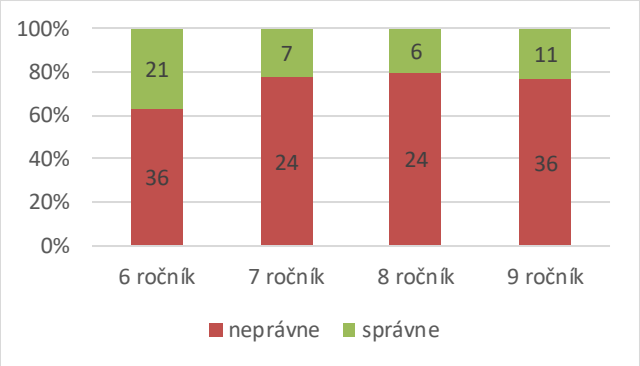
2.4 Použitie úlohy v zahraničí

K tejto úlohe Tunel sa zo zahraničnej databázy organizátorov prihlásili len Slovinci. A aj to túto úlohu použili len v druhom kole, čo znamená výrazne menšiu vzorku respondentov. Podľa slov Janes Demsar, ktorý nám údaje sprostredkoval, správne odpovedala približne len štvrtina súťažiacich. Ďalej uviedol, že medzi mladšími žiakmi (t.j. žiaci 6. a 7. ročníka) bolo oveľa viac správnych odpovedí ako medzi staršími (t.j. žiaci 8. a 9. ročníka). Zadanie úlohy znelo približne rovnako ako naše, len v závere bola pridaná požiadavka, že rodina si želá prejsť na druhu stranu za hodinu. Okrem toho mali na výber aj možnosť, že táto požiadavka je nesplniteľná. Možnosť neodpovedať nemali, keďže v Slovinsku neuplatňujú metódu udeľovania mínusových bodov za nesprávnu odpoveď. Na výber mali tieto štyri odpovede:

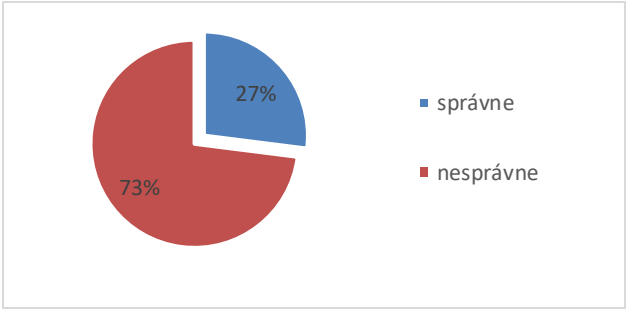
A. 35 minút
B. 45 minút
C. 60 minút
D. Nie je možné, aby rodina prešla tunelom za hodinu.

Pomer, v ktorom žiaci odpovedali na túto úlohu, rozdelený podľa ročníkov je vidieť v Tabuľke 1.

Tabuľka 1. Pomer správnych a nesprávnych odpovedí súťažiacich v Slovinsku, rozdelených po ročníkov



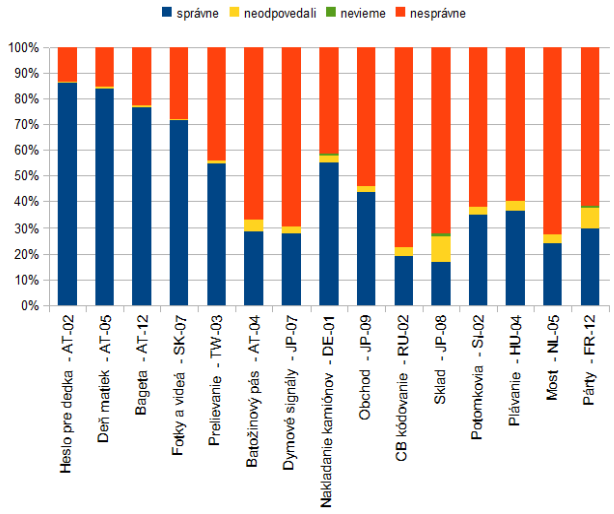
Celkový počet žiakov, ktorí súťažili v tomto druhom kole bolo 165. Z toho 45 žiakov odpovedalo správne a 120 nesprávne. Celkovú úspešnosť riešenia tejto úlohy je vidieť na grafe, ktorý je na Obrázku 5.



Obrázok 5. Celková úspešnosť riešenia úlohy Tunel v Slovinsku

3. Most

Okrem úlohy Tunel by sme sa chceli v tomto článku zaoberať aj úlohou Most. Algoritmus, ktorý bol potrebný na vyriešenie úlohy Tunel je pomerne známy a úloha s názvom *Most*, ktorá ho využívala tiež, bola ponúknutá do databázy Holanďanmi v roku 2013. Príbeh sa odohráva na moste a nie v tuneli, ale princíp riešenia ostáva rovnaký. Podľa záznamov z našej internej databázy vieme, že sme ju použili v školskom roku 2013/2014 v súťažnej kategórii Kadeti ako jednu z piatich ťažkých úloh. V tejto kategórii vtedy súťažilo 13 794 súťažiacich, z toho 7585 bolo chlapcov a 6209 dievčat. Plný počet bodov dosiahlo vtedy 6 žiakov a úspešných riešiteľov bolo 1700. Pre lepší obraz o úrovni vedomostí riešiteľov prikladáme graf úspešnosti riešených úloh z tohto ročníka (pozri Obrázok 6.).

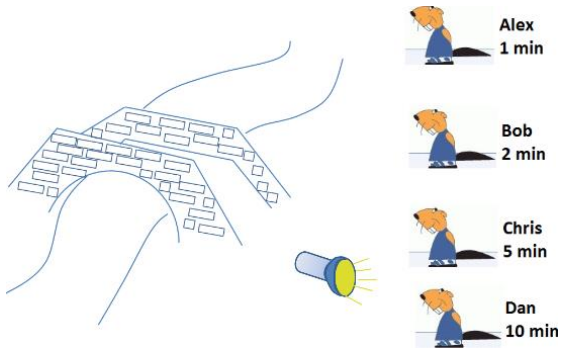


Obrázok 6. Diagram úspešnosť jednotlivých úloh v kategórii Kadeti 2013/14 [9]

3.1 Zadanie

Presné znenie úlohy *Most* v súťaži bolo:

V noci chcú prejsť cez most 4 bobry z pravej strany na ľavú. Majú len jednu baterku a po moste môžu ísť najviac 2 bobry a jeden z nich musí mať baterku. Ak je baterka na ľavej strane mostu, musí ju ostatným na pravú stranu priniesť bobor.



Obrázok 7. Ilustračný obrázok použitý v zadaní úlohy Most v kategórii Kadeti 2013/2014

Každý bobor prejde most za iný čas (Pozri Obrázok 7.). Ak idú spolu 2 bobry s baterkou, idú rýchlosťou pomalšieho bobra.

Za aký najmenší čas môžu prejsť všetci 4 na druhú stranu?

A) 10
B) 17
C) 19

D) 23

Správna odpoveď bola možnosť B).

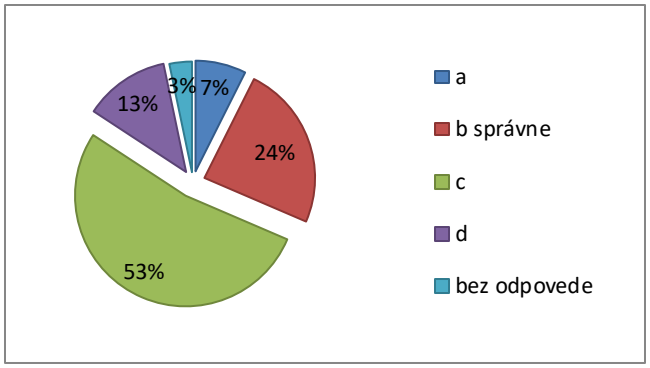
3.2 Analýza

Keďže spôsob riešenia úlohy Most je rovnaký ako v úlohe Tunel, len s inými číselnými hodnotami, prechádzame priamo k analýze žiackych riešení.

Do kategórie Kadeti spadajú žiaci ôsmeho a deviateho ročníka ZŠ a z osemročných gymnázií žiaci tercie a kvarty. Asi štvrtina súťažiacich v tejto kategórii zvolila správnu odpoveď B). Zato k možnosti C) 19 min., ktorá správna nebola, sa priklonila viac ako polovica žiakov. Týchto 19 minút sa však dá ľahko zdôvodniť. Ak si žiaci zobrali do úvahy možnosť, že sa oplatí prechádzať cez tunel vždy s „časovo najlacnejším“ bobrom Alexom, ktorému to trvá len 1 minútu, tak v takom prípade mohli za správnu odpoveď ľahko pokladať možnosť C) 19 min. Možnosť neodpovedať zvolili len 3% z celkového počtu súťažiacich, čo je približne 414 žiakov. Pre celkový obraz pozri graf na Obrázku 8.

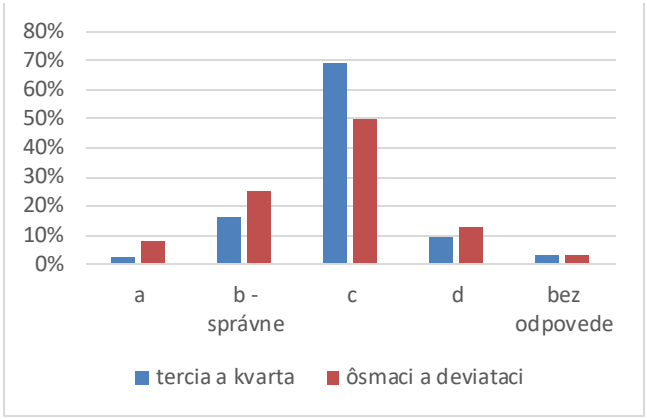
V úlohe Most mohli žiaci dosiahnuť horšie výsledky ako v úlohe Tunel, z nasledujúcich troch dôvodov:

1) Žiaci boli mladší, a to konkrétne o jednu kategóriu.
2) Medzi odpoveďami sa nachádzala aj možnosť 19 min., ktorá patri do prvoplánových možností, pokiaľ túto algoritmickú hádanku nepoznáte.
3) Medzi riešiteľmi úlohy Tunel už existovali takí, ktorí úlohu Most riešili v minulosti a mohli poznať správnu odpoveď.



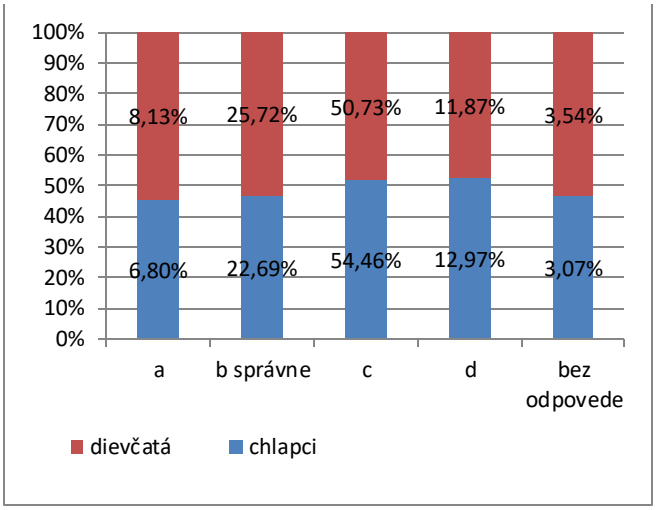
Obrázok 8. Celková úspešnosť riešenia úlohy Most v kategórii Kadeti 2013/14

Ak by sme chceli aj v tejto úlohe porovnať výsledky žiakov na základe navštevovaného druhu školy, teda či ide o žiakov základných škôl, alebo gymnazistov (pozri Obrázok 9), tak dostaneme pomerne opačné výsledky ako pri úlohe Tunel. Väčšina žiakov totiž zvolila nesprávnu odpoveď C) bez ohľadu nato, či šlo o gymnazistov, alebo žiakov základnej školy. Avšak pri správnej odpovedi je výsledok opačný. Správnu odpoveď zvolilo percentuálne viac žiakov základných škôl ako gymnazistov. Pravdepodobne k tomu mohlo prispieť celkovo nižšie zastúpenie gymnazistov (1979) ako žiakov základných škôl (11 815), ktorých bolo niekoľko násobne viac.



Obrázok 9. Porovnanie riešení žiakov tercie a kvarty s ôsmakmi a deviatakmi v úlohe Most

Ak by sme sa pozreli na rozdiel medzi úspešnosťou chlapcov a dievčat v tejto úlohe tak, ako sme sa pozerali v úlohe Tunel, dostali by sme približne rovnaké výsledky. Tiež nenaznačujú žiadnu výraznú rozdielnosť (pozri Obrázok 10.).



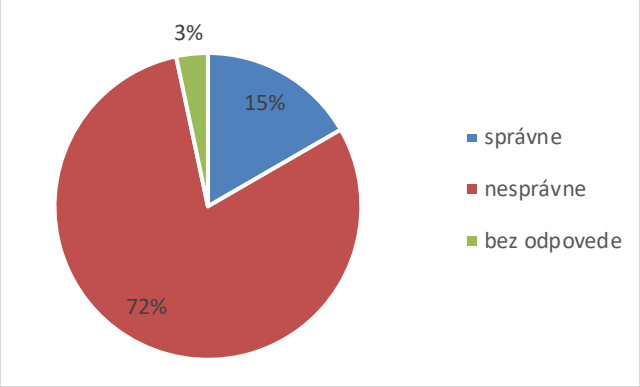
Obrázok 10. Porovnanie voľby odpovedí na základe pohlavia žiakov v úlohe Most

3.3 Použitie úlohy v zahraničí

Túto úlohu riešili v minulých rokoch napríklad Ukrajinci a Rusi. Údaje o úspešnosti riešiteľov v ich krajinách sme získali na základe malého prieskumu, ktorý sme robili pre účely tohto článku.

3.3.1 Ukrajina

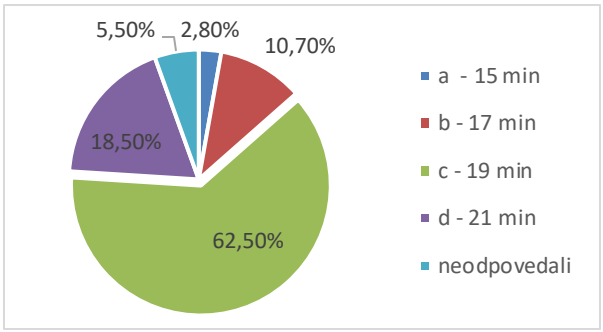
Údaje z Ukrajiny sme dostali od Rostyslava Shpakovych. Zadanie použili v rovnakom znení ako my, až na prípadné drobné rozdiely vyplývajúce z rozdielnosti povahy jazykov. Vo svojej databáze však nemajú uchovaný rovnaký druh informácií ako my, takže sme nezískali informácie o presnom počte účastníkov, ale len pomer v akom odpovedali správne, nesprávne, či neodpovedali, pozri Obrázok 11.



Obrázok 11. Celková úspešnosť riešenia úlohy Most v kategórii Juniori 2013/14 na Ukrajine

3.3.2 Rusko

Úlohu Tunel použili v roku školskom roku 2013/14 aj v Rusku, pričom sa jej celkovo zúčastnilo 690 žiakov z 11. ročníka, čo je pre nich posledný ročník daného stupňa. Žiaci v týchto ročníkoch mávajú približne16-17 rokov, čo môže zodpovedať kategórii Juniori. Výsledky dopadli o niečo horšie ako na Slovensku (pozri Obrázok 12.), napriek tomu bol pomer odpovedí približne rovnaký ako u nás. U nás zvolilo nesprávnu odpoveď 19 minút len niečo cez 50%, u nich to bolo cez 60%. U nás zvolilo správnu odpoveď približne štvrtina žiakov, u nich to bolo len niečo tesne cez 10%. Informácie nám sprostredkoval Ilya Posov.



Obrázok 12. Celková úspešnosť riešenia úlohy Most v školskom roku 2013/14 v Rusku

4. Porovnanie oboch úloh

Keďže išlo o algoritmicky rovnakú úlohu, ktorá bola použitá v súťažnom roku 2013/14 aj v roku 2016/17, chceli sme vedieť, či existujú žiaci, ktorí riešili obe tieto úlohy. Najprv ako Kadeti a neskôr ako Juniori. A ak áno, ako dopadli.

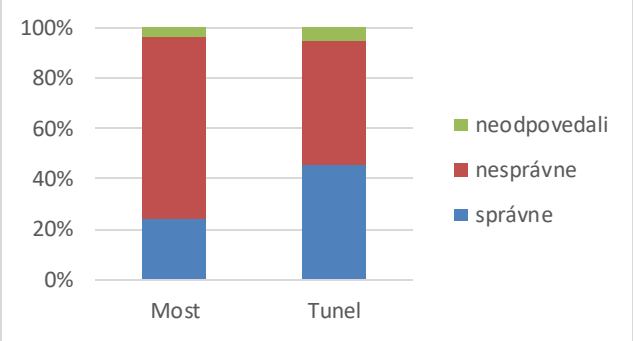
Pri vyhľadávaní týchto žiakov v databáze však nastali komplikácie. Skupinu takýchto žiakov vieme totiž dohľadať len čiastočne. Dôvody sú viaceré. Po prvé, ak učiteľ žiaka v jednom roku zaregistruje a na nasledujúci rok nastaví len prebratie informácií z predošlého roku a neregistruje ho odznova, ostáva mu ten istý kód. Avšak veľa učiteľov nahráva mená žiakov každý rok znovu a tak dostávajú nové kódy a my ich nevieme identifikovať a spätne spárovať. Týmto strácame množinu súťažiacich, ktorých výsledky by sme vedeli porovnať. Po druhé, žiaci, ktorí súťažili najprv na základnej škole ako Kadeti a v nasledujúci rok prešli na strednú školu, kde súťažia ako Juniori, majú tiež nové súťažné kódy. To je tiež pomerne veľká množina súťažiacich, ktorých údaje nevieme dohľadať. Preto sme po filtrácii v databáze získali údaje len od 45 žiakov, ktorých výsledky vieme porovnať. Z toho bolo 13 dievčat a 32 chlapcov. Podrobnejší stav o ich zlepšení či zhoršení je zaznamenaný v Tabuľke 2.

Tabuľka 2. Počet a odpovede žiakov riešiacich obe úlohy

stav	počet žiakov	dievča	chlapec
obe dobre	2	2	0
obe zle	16	4	12
"zlepšili sa"	18	6	12
"zhoršili sa"	9	1	8
Spolu	45	13	32

Aj keď celková vzorka 45 žiakov nie je veľmi významná, môžeme skonštatovať, že len dve dievčatá dokázali vyriešiť obe úlohy správne a to aj napriek tomu, že chlapcov v tejto vzorke bolo takmer trikrát toľko ako dievčat. Pozitívne však vidíme fakt, že zo žiakov, ktorí tuto úlohy riešili druhý raz, sa viac žiakov zlepšilo ako zhoršilo. Približne tretina žiakov úlohu nevedela vyriešiť ani s odstupom času pri druhom pokuse.

Celkové porovnanie úspešnosti riešení týchto dvoch úloh je znázornené na grafe na Obrázku 13, kde môžeme vidieť, že ani jedna úloha nedosiahla nad 50% správnych odpovedí. Napriek tomu úloha Most dopadla výrazne horšie. Dôvody, ktoré k tomu mohli prispieť uvádzame v kapitole 3.2.

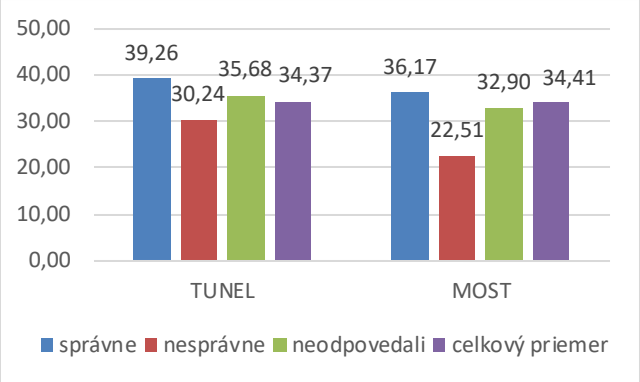


Obrázok 13. Porovnanie úspešnosti úloh Most a Tunel

Ďalšia závislosť, ktorá nás zaujímala pri týchto dvoch úlohách bola, či žiaci, ktorí odpovedali správne, patrili k tým žiakom, ktorí dosiahli v priemere väčší počet bodov, ako bol priemer získaných bodov v danej súťažnej kategórii. Na grafe, ktorý je znázornený na Obrázku 14., je vidieť takéto porovnanie pre obe úlohy.

V úlohe Most, ktorú riešili žiaci pred troma rokmi, bol celkový priemer získaných bodov približne 34 bodov. Tí žiaci, ktorí odpovedali v tejto úlohe správne, patrili do skupiny, ktorá získala priemerne 36 bodov. Teda ich priemer bol väčší. Zato žiaci, ktorí v tejto úlohe neodpovedali dosahovali v priemere nižší počet bodov ako bol celkový priemer získaných bodov.

V úlohe Tunel, ktorú už riešili starší žiaci (Juniori), je celkový priemer získaných bodov (34 bodov) nižší ako priemerný počet získaných bodov žiakmi, ktorí mali správnu odpoveď (39 bodov) a aj tých, ktorí neodpovedali (36 bodov). Jeden zo záverov, ktorý vieme z toho vyvodiť je, že žiaci už v tomto veku vedia lepšie voliť možnosť neodpovedať a nestrácajú tak body za prípadne nesprávnu odpoveď.



Obrázok 14. Priemerný počet získaných bodov podľa odpovedi v úlohách Tunel a Mosty

5. Záver

V článku sme analyzovali dve úlohy, ktoré majú rovnaké algoritmické riešenie. Na základe databázy sme vedeli, že existujú žiaci, ktorí tieto úlohy mohli riešiť raz ako kadeti a neskôr ako Juniori. Chceli sme spraviť malý prieskum, nakoľko a či vôbec sa dá zaznamenať zlepšenie vedomosti u žiakov, ktorí súťažili v tejto súťaži niekoľko rokov za sebou. Porovnali sme teda najprv celkové výsledky týchto dvoch úloh v daných rokoch, keďže ide o algoritmicky rovnako zložité úlohy. Najprv sa jedna použila v súťaži v roku 2013/2014 u Kadetov a druhá v roku 2016/2017 u Juniorov. V texte sme uviedli aj analýzu úspešnosti vzhľadom na rozdielny druh škôl, ktorý súťažiaci navštevovali. Zaujímavé, či výrazne rozdiely sme však nezaznamenali. Riešili sme aj rozdiely medzi úspešnosťou riešení v závislosti od pohlavia, avšak výrazne rozdiely sme tiež nezaznamenali. Úloha dopadla jednoznačne lepšie u starších žiakov, teda u Juniorov. Tri dôvody, ktoré to mohli spôsobiť popisujeme v kapitole 3.2. V druhej polovici článku sme sa zamerali na 45 žiakov osemročných gymnázií, ktorí súťažili v súťaži pod rovnakým kódom a na rovnakej škole a porovnávali sme ich výsledky. Z celkového hľadiska porovnania výsledkov došlo k zlepšeniu. Na základe porovnania priemerného získania bodov v súťaži sme zistili, že starší žiaci vedia pravdepodobne lepšie pracovať s možnosťou neodpovedať a nestrácať tak zbytočne body. Okrem toho sme získali údaje o výsledkoch týchto úloh z Ruska, Ukrajiny a Slovinska, pričom tieto údaje boli v rôznych formátoch, ktoré boli odlišné od našich. V porovnaní s výsledkami, ktoré nám z týchto krajín zaslali, naši žiaci vykazovali vyššiu úspešnosť.

POĎAKOVANIE

Na tomto mieste by som chcela poďakovať špeciálne Petrovi Tomcsányimu za sprostredkovanie údajov zo slovenskej databázy a za zdĺhavú komunikáciu so zahraničnými kolegami, pracujúcimi na iBobrovi, vďaka čomu sme získali dáta aj zo zahraničia.

BIBLIOGRAFICKÉ ODKAZY

[1] Hrušecká, A., Pekárová, J., Tomcsányi, P., Tomcsányiová, M. Informatický bobor – nová súťaž v informačných technológiách pre žiakov základných a stredných škôl. In Didinfo 2008. Banská Bystrica : Univerzita Mateja Bela, ISBN 978-80-8083-556-9

[2] iBobor. ibobor.sk

[3] Archív úloh ibobra. http://ibobor.sk/sutaz_demo/index.php

[4] TOMCSÁNYIOVÁ, M., TOMCSÁNYI, P. Analýza riešení úloh súťaže iBobor v kategórii Benjamíni v šk. roku 2012/13. In Didinfo 2013. Banská Bystrica : Univerzita Mateja Bela, ISBN 978-80-557-0527-9

[5] BEZÁKOVÁ, D. Analýza riešení úloh súťaže iBobor v kategórii Seniori v šk. roku 2012/2013. In Didinfo 2013. Banská Bystrica : Univerzita Mateja Bela, ISBN 978-80-557-0527-9

[6] WINCZER, M. Témy teoretickej informatiky v úlohách súťaže iBobor 2012. In Didinfo 2013. Banská Bystrica : Univerzita Mateja Bela, ISBN 978-80-557-0527-9

[7] TOMCSÁNYIOVÁ, M., & TOMCSÁNYI, P. (2014). ANALYSIS OF TASKS IN THE SLOVAKIAN BEBRAS CONTEST 2013/14. In *Journal of Technology & Information Education*, 2014, vol. 6, no. 1, p. 122-136. ISSN 1803-537X

[8] Vyhodnotenie výsledkov 2016. http://ibobor.sk/vyhodnotenie2016.php

[9] Vyhodnotenie výsledkov 2013. http://ibobor.sk/vyhodnotenie2013.php

Vývoj metodického materiálu pre vyučovanie základov programovania v jazyku Python

Designing methodology for learning programming fundamentals in Python

Eva Mészárosová
KZVI FMFI UK
Mlynská dolina
842 48 Bratislava
Slovensko
meszarosova@fmph.uniba.sk

ABSTRACT

In this paper we describe the development of a methodology for teaching programming fundamentals in the Python language. We decided to use in our materials the motivational power of turtle graphics, which supports learning by discovering and exploring. By means of it, pupils learn the basic concepts and constructs as variable and loop. We present our materials and also the process of their development and implementation to the learning process.

Keywords

Teaching programming. Python. Methodology.

ABSTRAKT

V príspevku popisujeme vývoj metodického materiálu pre vyučovanie základov programovania v jazyku Python ako súčasť všeobecno-vzdelávacieho predmetu informatika pre štvorročné gymnáziá. V materiáloch sme sa rozhodli použiť motiváciu korytnačej grafiky, ktorá podporuje učenie sa objavovaním a skúmaním. Pomocou nej si žiaci osvoja základné pojmy a konštrukcie jazyka Python ako premenná a cyklus. V príspevku predstavíme našu metodiku, jej vývoj a následné implementovanie do vyučovania.

Kľúčové slová

Vyučovanie programovania. Python. Metodický materiál.

1. ÚVOD

Programovanie je neoddeliteľnou súčasťou vyučovania informatiky na stredných školách. Vyučovanie základov programovania na stredných školách a gymnáziách je zamerané na vyučovanie algoritmov, rozvoja algoritmického myslenia a schopnosti riešenia problémov. Štátne dokumenty explicitne neurčujú konkrétny programovací jazyk a je na učiteľovi, aký programovací jazyk bude používať na hodinách informatiky. V súčasnosti sa zrejme najviac používa Pascal a prostredie Delphi alebo Lazarus. V posledných niekoľkých rokoch sa medzi jazyky, ktoré sa odporúčajú na vyučovanie základov programovania, dostáva jazyk Python. Výskumy ukazujú, že Python je vhodný na to, aby bol prvým programovacím jazykom, s ktorým sa žiaci na strednej škole stretnú a aj preto čoraz viac stredných škôl a zahraničných univerzít prechádza na vyučovanie základov programovania práve v jazyku Python. [1, 2, 3]

V rámci nášho výskumu sa sústredíme na učiteľov. Pre nich navrhneme ucelenú metodiku programovania v jazyku Python, podľa ktorej budú môcť učiť základy programovania. Súčasťou našej metodiky budú poznámky, rady a upozornenia, ktoré uľahčia učiteľom prechod z programovacieho jazyka Pascal, s ktorým majú skúsenosti, na programovací jazyk Python.

2. TEORETICKÉ VÝCHODISKÁ

Pri tvorbe našich materiálov sme sa držali rôznych princípov a taxonómií. Bloomova taxonómia vzdelávacích cieľov je kľúčová pre vzdelávací proces a tiež pre tvorbu vzdelávacích obsahov vo svete. V súčasnosti je však široko akceptovaná modifikácia tejto taxonómie – Revidovaná Bloomova taxonómia [4]. Pri tvorbe metodických materiálov je odporúčané mať na zreteli aj etapy poznávacieho procesu, ktoré definoval Hejný [5]: 1. Motivácia; 2. Izolované modely; 3. Univerzálny (generický) model; 4. Vznik poznatku. Salanci uvádza [6] mierne modifikované etapy poznávacieho procesu pre vyučovanie informatiky, kde modely nahradil získavaním praktických skúseností pri počítači: 1. Motivácia; 2. Zbieranie skúseností; 3. Zovšeobecňovanie; 4. Poznatok; 5. Trénovanie poznatku. V našej metodike sa budeme držať aj rád pre tvorbu správneho materiálu pre vyučovanie. Salanci uvádza [6], že správne učebnice kladú dôraz na to, aby žiaci zbierali skúsenosti, pričom: učebnice nemajú encyklopedický charakter, postupne sa vysvetľujú základné programátorské koncepty a dôraz sa kladie na motivujúce príklady, trasovanie a postupné objavovanie zákonitostí.

3. METODICKÝ MATERIÁL

Materiál sme navrhli pre všeobecno-vzdelávací predmet informatika pre žiakov štvorročného gymnázia. Pri jeho vývoji sme postupovali metodológiou výskumu vývojom. Materiál je rozdelený na tri časti podľa preberanej témy (úvod do jazyka, premenné a for cykly) a pozostáva z metodického materiálu pre učiteľa, ktorý je doplnený pracovnými listami pre žiakov.

V každej časti metodického materiálu sme definovali požadované vedomosti a zručnosti žiakov na začiatku preberanej témy. Stanovili sme tiež vzdelávacie ciele hodín a uviedli sme, aké kompetencie sa budú u žiakov rozvíjať na základe výkonového štandardu, ktorý je určený v iŠVP [7]. Súčasťou materiálu je aj štruktúra vyučovacej hodiny, pedagogické postupy a popis jej priebehu, spoločne so zadaním úloh pre žiakov, ich riešením a odporúčaniami pre učiteľov.

Materiál sa skladá z troch častí, podľa preberanej témy: úvod do jazyka, premenné a for cykly. Témy sme následne rozdelili, každú na niekoľko vyučovacích hodín (45 minút).

Pri návrhu úloh sme sa snažili vyberať také programy, ktoré nepotrebujú veľkú množinu príkazov, aby sme tým žiakov zbytočne nezaťažili. Sústredili sme sa na pojmy a konštrukcie, ktoré sú definované v iŠVP.

V metodike sme sa zamerali na oblasť *Algoritmické riešenie problémov*. Našou cieľovou skupinou sú žiaci 1. ročníka gymnázia so štvor- alebo päťročným vzdelávacím programom. Zamerali sme sa na základné pojmy, konštrukcie a algoritmy, preto sme nároky na vstupné vedomosti žiakov stanovili len zvládanie základných úkonov s počítačom.

Pri vyučovaní základov programovania podľa nášho materiálu je potrebné, aby mal každý žiak samostatný počítač. Učiteľ by mal mať k dispozícií počítač s projektorom a tabuľu – učiteľ vysvetľuje pomocou písania kódu do počítača a popri tom kreslí pomocné kresby na tabuľu. Ak učiteľ nemá k dispozícií projektor, odporúčame rozdeliť tabuľu na dve časti: na jednej bude písať kód tak, ako keby písal na počítači v prostredí IDLE, a druhú časť tabule využije pre pomocné kresby a vysvetlenia.

Z hľadiska priebehu vyučovacích hodín sme sa držali schémy: *Motivácia → Zbieranie skúseností → Trénovanie poznatku*, pričom v časti motivácia žiaci spolu s učiteľom riešia zadanú úlohu na tabuli (projektore) a v ďalších častiach žiaci pracujú samostatne každý pri svojom počítači.

Metodický materiál sme rozdelili na tri časti podľa preberanej témy:

1. Úvod do Pythonu a korytnačej grafiky

Pre túto tému sme vyhradili dve vyučovacie hodiny. Na prvej hodine sa žiaci zoznámia a prostredím IDLE, jeho príkazovým a programovacím režimom a základnými úkonmi ako uloženie a otvorenie programu. Žiaci sa takisto zoznámia s korytnačou grafikou pomocou kreslenia jednoduchých obrázkov.

2. Premenné

Pre tému premenné sme vyhradili jednu vyučovaciu hodinu, pričom premenné využijeme ďalej aj v nasledujúcej téme. Žiaci sa zoznámia s pojmom premenná, priradením hodnoty do premennej a zmenou jej hodnoty.

3. For cykly

Pre túto tému sme vyhradili päť vyučovacích hodín. Žiaci sa postupne zoznámia s cyklami **`for`**, a funkciou **`range()`**, naučia sa rozoznať opakujúce sa vzory, použiť riadiacu premennú a premenné v tele cyklu a vnorené cykly.

3.1 Úvod do Pythonu a korytnačej grafiky

Cieľom týchto hodín je zoznámiť žiakov s novým programovacím jazykom Python a jeho vývojovým prostredím IDLE, ako aj s ideou korytnačej grafiky. Žiaci po týchto hodinách vedia vytvoriť prvý program, ktorý po spustení nakreslí obrázok, ktorý bol určený v zadaní úlohy.

Prvé dve vyučovacie hodiny sú zamerané na to, aby sa žiaci naučili pracovať s prostredím IDLE – príkazovým a programovacím režimom, a aby sa v ňom dokázali orientovať.

Tieto hodiny sú určené na to, aby si žiaci osvojili základné príkazy na pohyb korytnačky, aby na nasledujúcich hodinách, na ktorých sa už stretnú s novými pojmami a konštrukciami v programovaní, ovládali základné korytnačie inštrukcie a prácu v prostredí IDLE. Práca s tromi samostatnými oknami, s ktorými sa v tomto programovacom prostredí stretnú, môže byť pre niektorých žiakov náročná. Preto sme pre túto tému vyhradili dve vyučovacie hodiny, aby mali žiaci dostatočný čas na precvičenie a aby mal učiteľ priestor na uistenie sa, či sa všetci žiaci vedia orientovať v tomto prostredí.

Ďalším cieľom úvodných hodín je aj to, aby žiaci uvažovali o vlastnostiach vykonávateľa – korytnačky. Žiaci na hodinách kreslia jednoduché obrázky, ktoré sú určené na automatizáciu základných príkazov na pohyb korytnačky.

1. vyučovacia hodina: Úvod do Pythonu a IDLE, úvod do korytnačej grafiky

Na začiatku hodiny učiteľ povie pár slov o programovacom jazyku Python, o jeho autorovi a využití vo svete. Ukáže žiakom vývojové prostredie IDLE, v ktorom budú pracovať. Následne ukáže žiakom importovanie knižnice **`turtle`**, vytvorenie korytnačky a základné príkazy na pohyb korytnačky **`forward(`*`pocet_krokov`*`)`**, **`back(`*`pocet_krokov`*`)`**, **`right(`*`pocet_stupnov`*`)`**, **`left(`*`pocet_stupnov`*`)`** a na zmazanie plochy **`reset()`**. Pomocou úloh ako kreslenie trojuholníka, štvorca a domčeka, sa žiaci oboznámia s kreslením jednoduchých útvarov pomocou korytnačky. Najprv žiaci pracujú v príkazovom režime, v ktorom sa príkazy vykonajú ihneď po ich zadaní a odoslaní tlačidlom Enter, čím žiaci dostanú okamžitú grafickú spätnú väzbu o správnosti svojho riešenia. Po oboznámení sa s príkazovým režimom a priamym riadením korytnačky učiteľ ukáže žiakom programovací režim, spustenie programu, ukladanie a otvorenie už uloženého programu. Následne žiaci riešia úlohy na precvičenie riadenia korytnačky a kreslenie jednoduchých útvarov. Keďže jednotlivé úlohy majú žiaci ukladať do samostatných súborov, pričom pracujú v programovacom režime, žiaci si precvičia aj prácu s prostredím IDLE a orientáciu v ňom.

2. vyučovacia hodina: Kreslenie pomocou korytnačky

Na druhej hodine si učiteľ spolu so žiakmi zopakuje prácu v prostredí IDLE, importovanie knižnice **`turtle`** a základné príkazy na riadenie korytnačky. Na tejto hodine sa žiaci zoznámia s novými príkazmi na zmenu farby a hrúbky pera korytnačky **`pencolor(`*`farba`*`)`** a **`pensize(`*`hrúbka`*`)`**, na zdvihnutie a položenie pera korytnačky **`penup()`**, **`pendown()`**. Táto hodina je venovaná precvičeniu práce v prostredí IDLE, riadenia korytnačky a kreslenia pomocou nej. V materiáli uvádzame časté chyby žiakov v syntaxi jazyka, ako napríklad to, že žiaci zabudnú zadať parametre príkazov alebo že vynechajú zátvorky pri jednotlivých príkazoch. Tiež ich upozorňujeme na to, aby si všímali prácu žiakov, či sa vedia orientovať v prostredí s tromi samostatnými oknami.

3.2 Premenné

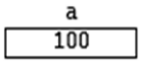
Pre tému premenná a priradenie sme vyhradili jednu vyučovaciu hodinu, avšak aj v nasledujúcich hodinách žiaci budú používať premenné. Pri vysvetľovaní premenných sme využili didaktický postup z prednášok Programovania v C++ [8] a motiváciu kreslenia štvorca a zmeny veľkosti jeho strán. Pri téme premenných je dôležité krokovanie pri postupnom priraďovaní hodnoty do premennej a znázorňovanie vyhradeného miesta v pamäti pre premennú.

Cieľom tejto vyučovacej hodiny je aby žiak porozumel pojmu premenná a jej využitiu v jazyku Python; naučil sa priradiť hodnotu do premennej a použiť hodnotu premennej; a dokázal identifikovať

zo zadania úlohy, ktoré údaje musia byť zapamätané, resp. sa menia (a teda vyžadujú použitie premenných).

3. vyučovacia hodina: Premenná

Na začiatku hodiny učiteľ zadá úlohu, v ktorej je úlohou žiakov nakresliť štvorec. Keďže žiaci ešte nepoznajú cyklus, pripravia program, v ktorom štyrikrát použijú príkazy na nakreslenie strany štvorca – napríklad `forward(100)` a následné otočenie korytnačky. Potom učiteľ uvedie problém, v ktorom prezentuje, že by sme chceli nakresliť štvorec s inou veľkosťou strany. Spolu so žiakmi vyvodí, že počet krokov, ktoré korytnačka prejde bude potrebné zmeniť v každom príkaze `forward(100)`, teda až na štyroch miestach v programe. Následne žiakom ukáže riešenie takejto situácie: zavedenie premennej a nahradenie čísla 100 touto premennou vo všetkých príkazoch `forward(100)`. Pre zovšeobecnenie učiteľ oboznámi žiakov s pojmom premenná. Využije pritom počítač s projektorom a tabuľu, na ktorú popri vysvetlení kreslí krabičky – miesto v pamäti vyhradené pre premennú, kde je uložená hodnota premennej, pozri Obrázok 1.



Obrázok 1. Znázornenie hodnoty premennej

Po využití premennej pre nastavenie strany štvorca učiteľ niekoľkokrát vykreslí štvorec s rôznymi veľkosťami strán, pričom mení len hodnotu premennej. Následne spolu so žiakmi zmení príkazy tak, aby nakreslili obdĺžnik, pričom vytvorí ďalšiu premennú. Po vykreslení obdĺžnika učiteľ oboznámi žiakov s operáciami s premennými: vytvoria premennú `sucet`, do ktorej priradia súčet hodnôt premenných `a` a `b`. Tento súčet ďalej využijú pri vypočítaní hodnoty obvodu obdĺžnika. V poradí nasleduje úloha pre žiakov, pri ktorej majú nakresliť štvorec s dĺžkou strany 100 a následne štvorec, ktorý má veľkosť strany o 50 krokov väčšiu. Pri tejto úlohe sa žiaci zoznámia so zvýšením hodnoty premennej pomocou priradenia `a = a + 50`. Žiaci ďalej riešia úlohy na precvičenie použitia premennej.

3.3 For cykly

Tému for cykly sme rozdelili do piatich vyučovacích hodín, pričom sme postupovali od for cyklov s vymenovanými hodnotami pre premennú cyklu, cez for cykly s funkciou `range(hodnota)`, for cykly s riadiacou premennou a premennou v tele cyklu až po dva vnorené for cykly. Využili sme didaktický postup z učebnice Algoritmy s Logom, pozri [9]. Cieľom týchto vyučovacích hodín je, aby žiaci dokázali rozoznať opakujúce sa vzory a dokázali zovšeobecňovať a zapisovať riešenie pomocou cyklu. Ako aj v téme premenné, aj tu je dôležité trasovanie programu, pomocou neho môžu žiaci hľadať a nájsť určité vzťahy, môže byť tiež pomôckou pri zovšeobecnení.

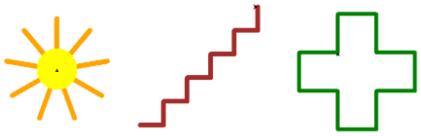
Cieľom týchto hodín je zoznámiť žiakov s cyklami for a funkciou `range(hodnota)`. Takisto aby žiaci dokázali rozpoznávať opakujúce sa vzory, zovšeobecňovať a zapisovať riešenie pomocou cyklu.

4. vyučovacia hodina: For cykly

Učiteľ na začiatku hodiny zadá úlohu na nakreslenie štvorca. Upozorní žiakov na opakujúce sa príkazy a zavedie konštrukciu cyklu, zatiaľ v tvare:

```
for i in 1,2,3,4:
    forward(100)
    right(90)
```

Nasleduje zbieranie skúseností počas riešenia úloh na kreslenie trojuholníka, 5-, 6- a 7-uholníka a obrázkov ako napríklad:



Obrázok 2. Úlohy pre tému for cykly

Po úlohách určených pre zbieranie skúseností učiteľ zhrnie nové učivo a nasledujú úlohy určené na trénovanie poznatku. Do konca hodiny žiaci pracujú samostatne pri počítačoch, pričom učiteľ ich pozoruje pri práci a pomáha im.

5. vyučovacia hodina: For cykly s funkciou range(*hodnota*)

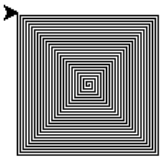
Na nasledujúcej vyučovacej hodine učiteľ uvedie problém kreslenia slnka s 24 lúčmi, pričom spolu so žiakmi si diskusiou zopakujú konštrukciu cyklu. Pri motivácii väčšieho počtu opakovania v cykle učiteľ oboznámi žiakov s príkazom na generovanie postupnosti, s funkciou `range(hodnota)`. Spoločné riešenie príkladu slnka s 24 lúčmi je teda:

```
for i in range(24):
    forward(50)
    back(50)
    right(360/24)
```

Nasleduje cvičenie pre žiakov s úlohami, v ktorých majú nakresliť rôzne obrázky použitím for cyklu a `range(hodnota)`. Do cvičení sme zahrnuli aj úlohy na precvičenie čítania a interpretácie už vytvoreného programu.

6. vyučovacia hodina: For cykly s riadiacou premennou a ďalšou premennou v tele cyklu

Učiteľ na začiatku hodiny zadá žiakom úlohu na nakreslenie špirály, pričom úlohu riešia spoločne pri tabuli alebo počítači s projektorom.



Obrázok 3. Špirála

Učiteľ pritom môže nakresliť na tabuľu pomocný obrázok znázorňujúci špirálu a dĺžky jednotlivých strán, takisto krokovanie pomocou príkazov: `fd(1)`; `rt(90)`; `fd(2)`; `rt(90)`; ...; `fd(99)`, `rt(90)`. Keďže žiaci poznajú premenné a for cykly, úlohu vedia spoločne vyriešiť pomocou využitia premennej, ktorej hodnotu postupne zvyšujú v tele cyklu. Po vyriešení príkladu pomocou premennej, učiteľ uvedie problém: „Ako by sme mali upraviť tento program tak, aby sme nemuseli použiť premennú `a`?“. Následne oboznámi žiakov s pojmom riadiacej premennej a jej využitím v tele cyklu. Spoločne vyriešia úlohu špirály:

```
for i in range(100):
    fd(i)
    rt(90)
```

Ďalej majú žiaci za úlohu meniť hodnoty parametrov v príkazoch `fd(i)` na `i*2` a `i**2` v úlohe špirály, a takisto meniť veľkosti uhla

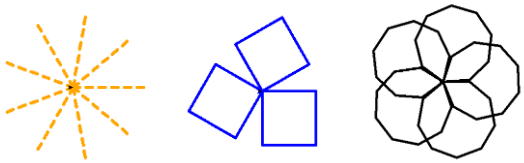
v tele cyklu. Ďalej žiaci riešia úlohy zamerané na zbieranie skúseností s premennými v tele cyklu.

7. a 8. vyučovacia hodina: Vnorené cykly

Nasledujúca vyučovacia hodina je zameraná na vnorené cykly, kde sme využili viaceré príklady z učebnice Algoritmy s Logom [9]: kreslenie štvorca prerušovanými čiarami. Učiteľ najprv zadá žiakom úlohu nakresliť prerušovanú čiaru a následne štvorec pomocou prerušovanej čiary.

Obrázok 4. Štvorec s prerušovanými čiarami

Zvyšok hodiny je venovaný zbieraniu skúseností pri riešení úloh. Žiaci kreslia obrázky ako napríklad:



Obrázok 3. Úlohy pre tému vnorené cykly

Do cvičení sme zahrnuli aj úlohy s použitím premenných v tele cyklu, kde v jednej úlohe zvyšujú hodnotu premennej a v druhej úlohe násobia hodnotu premennej. Po vyriešení úloh venovaných zbieraniu skúseností nasleduje zovšeobecnenie a žiaci ďalej riešia úlohy určené na trénovanie poznatku.

Posledná hodina v tejto téme je venovaná fixačnej a diagnostickej etape vyučovania. Žiaci riešia úlohy na precvičenie vnorených cyklov, pričom sa v úlohách objavujú aj premenné. Pre 7. a 8. vyučovaciu hodinu sme pripravili spoločný pracovný list kvôli tomu, že nové učivo sa preberie na začiatku 7. hodiny a následne žiaci pracujú samostatne pri počítačoch, postupujú vlastným tempom aj na 8. vyučovacej hodine. Učiteľovi odporúčame využiť poslednú vyučovaciu hodinu na sledovanie postupu riešenia úloh žiakov, pričom by mal sledovať, či si žiaci osvojili preberané pojmy a konštrukcie. V jazyku Python je telo cyklu určené odsadením bloku príkazov, preto je dôležité dôkladne vysvetliť žiakom význam odsadenia a dbať na podrobné znázornenie odsadenia príkazov vo vnorených cykloch.

Hodnotenie

Každý učiteľ má vlastný štýl hodnotenia, takisto školy majú rôzne harmonogramy a systémy hodnotení, preto sme priamo k popisu hodín nezahrnuli hodnotenie žiakov. Odporúčame však žiakov priebežne slovne motivovať – pochváliť ich alebo ich opravovať, aby si nadobudli správne programátorské zvyky a aby sme udržali ich motiváciu k ďalšej práci. Učiteľom odporúčame zaradiť sumatívne hodnotenie po téme for cykly, napríklad vo forme písomky, kde žiaci dostanú hotový program a ich úlohou je nakresliť obrázok, ktorý by nakreslila korytnačka po spustení programu.

4. VÝVOJ METODICKÉHO MATERIÁLU

V tejto časti popisujeme priebeh vývoja metodického materiálu. Predstavíme naše kroky pri zbieraní skúseností pred návrhom metodiky a stručný priebeh vývoja nášho metodického materiálu.

4.1 Získanie vlastných skúseností

Samotnému návrhu predchádzalo niekoľko krokov k získaniu vlastných skúseností s jazykom Python vo vyučovaní programovania:

4.1.1 Pedagogická činnosť

Doktorandské štúdium zahŕňa aj vlastnú pedagogickú činnosť na univerzite. Orientačná iterácia nášho výskumu trvala približne rok a pol (tri semestre), počas nej sme viedli cvičenia programovania v jazyku Python pre študentov prvého ročníka odboru aplikovaná informatika. Vďaka tomu sme získali nielen programátorské zručnosti v tomto jazyku, ale priamo v praxi sme videli prednášky, ktoré viedol skúsený vysokoškolský pedagóg. [10]

4.1.2 Analýza učebníc a materiálov pre jazyk Python

V rámci štúdia odbornej literatúry sme zozbierali a analyzovali knihy, učebnice a materiály o jazyku Python [11, 12]. Vytvorili sme takisto prehľad programovacích a vývojových prostredí, ktoré by sa mohli používať pri vyučovaní programovania v jazyku Python. Ako napríklad: PythonTurtle, Trinket, PyCharm Edu, Pythonroom. Pre naše materiály sme zvolili vývojové prostredie IDLE, z dôvodu ľahkej dostupnosti pre školy ako aj pre žiakov.

4.1.3 Prehľad vyučovania programovania na Slovensku

Získanie všeobecného prehľadu o stave vyučovania programovania v jazyku Python nám malo pomôcť k lepšej predstave možností vyučovania programovania v jazyku Python na gymnáziách na Slovensku. Vytvorili sme prehľad vyučovania programovania v jazyku Python na niektorých gymnáziách na Slovensku. V rámci toho sme sa rozprávali s učiteľmi, ktorí sa s nami podelili o svoje skúsenosti s vyučovaním programovania v jazyku Python. Podarilo sa nám skontaktovať s ôsmimi učiteľmi gymnázií, ktorí v súčasnosti vyučujú programovanie v jazyku Python.

4.1.4 Analýza učebníc programovania

Skúmali sme v súčasnosti najpoužívanejšie učebnice programovania jazyka Logo a Pascal: učebnicu Algoritmy s Logom a Programujeme v Delphi a Lazaruse. V obidvoch učebniciach je práca s grafikou základnou témou a ostatné pojmy a koncepty sú vysvetľované pomocou nej, a tak žiaci vizuálne vidia, či úlohu vyriešili správne. Zistili sme, že poradie niekoľkých prvých tém sa v obidvoch učebniciach zhodovalo, to znamená, že cykly, podmienky a procedúry sa dajú učiť aj pomocou klasickej, aj pomocou korytnačej grafiky. Pri návrhu našej metodiky na vyučovanie základov programovania v jazyku Python pre stredné školy budeme čo najviac vychádzať práve zo skúmaných učebníc a využívať úlohy, ktoré navrhli skúsení a uznávaní autori učebníc programovania pre stredné školy. [13]

4.1.5 Zvolenie prvotných zdrojov

Zvolili sme niekoľko materiálov, ktorých sme sa držali pri tvorbe našej metodiky. Ako prvotný zdroj, sme zvolili Učebnicu Algoritmy s Logom [9], ktorá je určená pre gymnáziá. Pri zvolení didaktického postupu a motivácií sme čerpali z materiálov z projektu ĎVUI [6] [14], prednášok z predmetov Programovanie 1 v Pythone [10], a Programovanie v C# z FMFI UK v Bratislave [8], takisto sme sa motivovali učebnicou Programujeme v Delphi a Lazaruse [15].

4.1.6 Zvolenie motivácie korytnačej grafiky

Pre naše metodické materiály sme si vybrali motiváciu korytnačej grafiky, ktorá podporuje učenie sa objavovaním a skúmaním. Našim cieľom bolo vytvoriť vzdelávací obsah pre vyučovanie základov programovania v jazyku Python pre prvý ročník gymnázia. Vychádzali sme v nich z toho, že žiaci, ktorí prichádzajú na strednú školu majú zo základnej školy skúsenosti s rôznymi detskými programovacími prostrediami. Našim cieľom je, aby sa na strednej škole stretli s vyšším programovacím jazykom, ktorý je vhodný na vyučovanie základov programovania. Zároveň sme sa v našich materiáloch snažili nájsť také námety, aby sa žiaci v tomto programovacom jazyku oboznámili aj s náročnejšími algoritmami.

4.2 Priebeh vývoja a overovania metodiky

Návrh prvej verzie materiálu prebiehal z časti pred overovaním a z časti priebežne počas jeho nasadenia do praxe. Takto sme vedeli flexibilnejšie reagovať na tempo žiakov a na potrebné korekcie plánu priebehu hodín alebo poradia niektorých tém.

Počas návrhu sme sa stretli s otázkami ako zaradiť jednotlivé témy a s ktorými príkazmi zoznámiť žiakov. Snažili sme sa navrhnúť metodiku tak, aby sme nezaťažili žiakov veľkým množstvom nových príkazov. Vyučovacie hodiny sme navrhovali tak, aby sme sa sústredili na nové pojmy a konštrukcie a nechali dostatočný priestor pre žiakov na osvojenie nového učiva. Prvý návrh obsahoval témy: úvod do Pythonu a korytnačej grafiky, for cykly, premenné.

Overovanie prvej verzie metodiky prebiehalo na jednom gymnáziu v Bratislave, kde podľa našej metodiky vyučovala jedna pani učiteľka žiakov druhého ročníka, pričom žiaci mali v prvom ročníku programovanie v prostredí Scratch. Materiál sme vyvíjali postupne, a sledovali sme aj reakcie žiakov na jednotlivé témy, ich otázky a analyzovali sme ich riešenia.

Po analýze prvej iterácie overovania metodiky sme vymenili poradie tém premenné a for cykly z dôvodu, že konštrukcia cyklu obsahuje riadiacu premennú. Po zmene poradia tém sa žiak stretne najprv s pojmom premenná a až neskôr s konštrukciou cyklu a s riadiacou premennou cyklu.

Druhý návrh metodiky sme overovali pomocou dvoch skupín prvého ročníka žiakov gymnázia. V prvej skupine sme odučili hodiny podľa našej metodiky my. Najprv prebiehalo vyučovanie v prvej skupine, po ktorom sme v materiáloch menili poradie niektorých úloh v cvičení, doplnili odporúčania pre učiteľov a upravili pracovné listy. Overovanie, ktoré sme realizovali v rámci nášho vyučovania v skupine žiakov prvého ročníka gymnázia nám poskytlo cenné skúsenosti s organizáciou vyučovacej hodiny a poskytlo náhľad do roly učiteľa pri vyučovaní základov programovania v skupine žiakov, ktorí majú rozdielne skúsenosti s programovaním. Taktiež sme získali skúsenosti s vyučovaním bez použitia projektora, pričom sme využili časť tabule na zápis príkazov a časť na pomocné kresby pri vysvetľovaní.

Následne prebiehalo overovanie v druhej skupine: jedna pani učiteľka postupovala vo vyučovaní programovania podľa našich materiálov, kde sme vystupovali ako zúčastnený pozorovateľ. Sledovali sme aj otázky učiteľky, ktoré sa týkali priebehu vyučovania a aj jazyka Python, podľa nich sme do materiálov doplnili odporúčania pre učiteľa.

Reakcie žiakov na hodiny informatiky boli kladné, aj pri začiatkoch menej motivovaní žiaci sa na posledných hodinách aktívne zapájali do vyučovacieho procesu. Sledovali sme aj postup riešenia úloh žiakov, ich otázky a chyby, z ktorých sme vyvodili niektoré nedostatky metodických materiálov.

ZÁVER

V práci sme sa zamerali na vyučovanie základných programátorských konceptov v jazyku Python. Našim cieľom bolo vytvoriť metodický materiál pre vyučovanie základov programovania v jazyku Python pre prvý ročník gymnázia, v ktorom sme sa sústredili na základné pojmy a konštrukcie: premenná a cykly. Metodický materiál sme vyvíjali iteratívne, realizovali a overovali pomocou výskumu vývojom v troch skupinách žiakov gymnázia v Bratislave.

Pri vývoji metodického materiálu, kde sme sa držali toho, aby sme nezaťažili žiakov veľkým množstvom príkazov jazyka Python, ale aby sme ich naučili základné pojmy a konštrukcie, a motivovali ich k ďalšiemu učeniu programovania. Veríme, že nami vytvorené materiály budú slúžiť k rozvoju algoritmického myslenia žiakom a k rozvoju dôležitých znalostí a zručností ako aj na napĺňanie vzdelávacích cieľov informatiky.

BIBLIOGRAFICKÉ ODKAZY

[1] ATTEQ, M., et al. 2014. *C++ of Python? Which One to Begin With: A Learners Perspective.* s.l. : IEEE. 2014 International Conference on Teaching and Learning in Computing and Engineering. ISBN 978-1-4799-3592-5/14.

[2] GRANDELL, L., et al. 2006. *Why complicate things?: introducing programming in high school using Python.* Hobart : Australian Computer Society, Inc., 2006. In: Proceedings of the 8th Australasian Computing Education Conference (ACE2006). Zv. 52, s. 71-80.

[3] KRPAN, D. a BILOBRK, I. 2011. *Introductory Programming Languages in Higher Education.Opatija* : IEEE, 2011. MIPRO, 2011. In: Proceedings of the 34th International Convention. s.1331-1336. ISBN 978-1-4577-0996-8.

[4] ANDERSON, et al. 2001. *A taxonomy for tearning, teaching, and assessing: A revision of Bloom's taxonomy of educational abjectives.* 2001. New Jersey: Pearson. ISBN 13: 978-0801319037.

[5] HEJNÝ, M. a KUŘINA, F. 2009. *Díte, škola a matematika: konstruktivistické prístupy k vyučování.* Praha: Portál, 2009. ISBN 978-80-7367-397-0.

[6] SALANCI, Ľ. et al. 2011. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika - Didaktika programovania pre stredné školy 1*. Bratislava: Štátny pedagogický ústav, 2011. ISBN 978–80–8118–079-8.

[7] ŠPÚ - Štátny pedagogický ústav. 2015. *Inovovaný Štátny vzdelávací program - príloha Informatika - gymnázium so štvorročným a päťročným vzdelávacím programom.* Bratislava, 2015. Dostupné na: http://www.statpedu.sk/sites/default/files/dokumenty/inovovany-statny-vzdelavaci-program/informatika_g_4_5_r.pdf.

[8] SALANCI, Ľ. 2016. *Programovanie v C++*. Bratislava, Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky. [Online] [Dátum: 29. 11. 2016.] Dostupné na: https://www.edi.fmph.uniba.sk/~salanci/C/index.html.

[9] VARGA, M., et al. 1999. *Informatika pre gymnáziá: Algoritmy s Logom.* Bratislava: Media Trade, spol. s. r. o., 1999. ISBN 80-08-02965-X.MOLNÁR, Z. *Moderní metody řízení informačních systémů.* Praha : Grada, 1992. 347 s. ISBN 80-85623-07-2.

[10] BLAHO, A. 2016. *Programovanie v Pythone 1 (prednášky k predmetu Programoanie (1)*. [Online] 2016. [Dátum: 30. 9. 2016.] Dostupné na: www.input.sk.

[11] MÉSZÁROSOVÁ, E. a TOMCSÁNYIOVÁ, M. 2015. *Vyučovanie základov programovania v jazyku Python na niektorých SŠ a VŠ.* Banská Bystrica : Univerzita Mateja Bela, 2015. In: Zborník z konferencie Didinfo 2015. ISBN 978-80-557-0852-2.

[12] MÉSZÁROSOVÁ, E. *Vývoj vzdelávacieho obsahu pre vyučovanie programovania v jazyku Python*. Rigorózna práca. Bratislava: Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, 2017.

[13] MÉSZÁROSOVÁ, E. 2015c. *Python and teaching programming at upper secondary schools*. Rožnov pod Radhoštěm: In: Proceedings of Information and Communication Technology in Education, 2015. ISBN 978-80-7464-763-5.

[14] BEZÁKOVÁ, D., et al. 2009. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika - Programovanie 1.* Bratislava: Štátny pedagogický ústav, 2009. ISBN 978-80-89225-65-1.

[15] BLAHO, A. 2012. *Informatika pre stredné školy: Programovanie v Delphi a Lazaruse.* Bratislava : Slovenské pedagogické nakladateľstvo - Mladé letá, s.r.o., 2012. ISBN: 978-80-10-02308-0.

Rozvíjanie čitateľskej gramotnosti prostredníctvom detského programovacieho jazyka Scratch

Developing Reading Literacy Through Children's Programming Language Scratch

Marek Nagy
Faculty of Mathematics, Physics and Informatics,
Comenius University in Bratislava
Mlynská dolina F1
842 48 Bratislava
Slovakia
mnagy@ii.fmph.uniba.sk

ABSTRACT

In the next contribution an experiment with teaching reading in the 4th grade of a primary school will be presented. It has been allocated one lesson a week for this purpose. A key element is a children's programming language Scratch where pupils animated their concepts about the read story from a book. Advances of children were continuously monitored through reading abbreviated versions of the stories in the web application Multimedia reader. From the process and the results can be concluded that such a form of education helps children improve themselves in aspects of reading literacy.

Keywords

Teaching reading. Reading comprehension. Scratch. Multimedia reader

ABSTRAKT

V nasledujúcom príspevku bude prezentovaný experiment s výučbou čítania vo 4. ročníku základnej školy. Na tento účel bola vyčlenená jedna vyučovacia hodina v týždni. Kľúčovým prvkom je detský programovací jazyk Scratch, pomocou ktorého žiaci animovali svoje predstavy o prečítanej povesti z knižky. Napredovanie detí bolo priebežne monitorované prostredníctvom čítania skrátených verzií povestí cez webovú aplikáciu Multimediálna čítanka. Z priebehu a výsledkov možno usúdiť, že takáto forma výučby pomáha deťom zlepšovať sa v aspektoch čitateľskej gramotnosti.

Kľúčové slová

Výučba čítania. Čítanie s porozumením. Scratch. Multimediálna čítanka.

1. ÚVOD

Výučba čítania je kľúčovou kompetenciou, ktorú by mali deti zvládnuť v rámci primárneho vzdelávania. V prvých ročníkoch sa cieľ zužuje skôr na nácvik hlasného čítania, ktoré sa neskôr rozšíri do čítania s porozumením. Pokiaľ deti nezískajú čitateľskú gramotnosť na prijateľnej úrovni, spôsobuje im problémy v ďalšom vzdelávaní. Bez prečítania a porozumenia logických súvislostí nie je možné napríklad riešiť matematické úlohy, vyvodzovať závery, hľadať súvislosti v dejepisných textoch, …

Na nácvik prvotného hlasného čítania je možné efektívne využiť počítačové aplikácie. Za priekopníka tejto oblasti možno označiť projekt LISTEN [1],[2]. V počiatkoch sa jednalo iba o využitie počítačového rozpoznávania reči na kontrolu nahlas prečítaného textu. Neskôr sa výskum sústredí aj na ďalšie fázy na ceste k čítaniu s porozumením [3],[4]. Či ide o nácvik správnej prozódie, využitie automatickej dopomoci k čítaniu [5] alebo dokonca automatické generovanie K-otázok (kto, čo, kde, kedy, prečo) (Wh-questions) k textu [6], ktoré preverujú porozumenie textu. V súčasnosti sa experimentuje aj so snímaním EEG signálu čítajúcich detí pomocou jednoduchých „čelenkových" snímačov.

Už v roku 1998 experimentálne porovnali prínosy projektu LISTEN s klasickou výučbou a komerčnými „čítacími" softvérmi. Ako sa uvádza na [2], deti využívajúce aplikáciu LISTEN jednoznačne dosiahli lepšie výsledky a predčili svojich rovesníkov. Dokonca predčili rovesníkov aj vo výsledkoch v tichom čítaní. Samozrejme, že sa jedná o čítanie anglicky hovoriacich detí. Dobré výsledky však s projektom dosiahli aj pri výučbe angličtiny ako sekundárneho jazyka [7].

Na nácvik hlasného čítania slovenských detí je orientovaná Multimediálna čítanka [8]. Jej vývoj začal roku 2005 a musel prejsť nevyhnutnými fázami podobne ako [2]. Najpodstatnejšou fázou je zber zvukových dát čítajúcich detí a vývoj rozpoznávača reči. Neskôr sa vytvorilo prostredie na hlasné čítanie, ktoré sa vylepšuje dodnes. K samotnému hlasnému čítaniu sa pridružilo aj testovanie otázkami.

Opodstatnenie Multimediálnej čítanky sa dlhodobo potvrdzuje na partnerskej základnej škole (Černyševského 8, Bratislava-Petržalka). Spolupráca trvá už od roku 2005. V rámci školského vzdelávacieho programu (ŠkVP) je vyčlenená v 1.-3. ročníku jedna hodina týždenne na vyučovací predmet, ktorý nesie názov Multimediálne čítanie (MMČ). Predmet figuruje samostatne, ale osnovou nadväzuje na „klasický" Slovenský jazyk (čítanie). Deti nacvičujú hlasné čítanie slov a súvislých textov pomocou počítačovej aplikácie. V prečítanom hľadajú informácie, ktoré im pomôžu individuálne zodpovedať prislúchajúce otázky. Jednoduchšie texty deti „obsiahnu" bez problémov. Pri zložitejších je to už problematické. Môžu si však pomáhať s celým textom a vrátiť sa k nemu opäť. Dosť často sa stáva, že

deti si s otázkou nevedia poradiť a skúšajú hľadať pomoc u učiteľa. Častá odpoveď je, že treba hľadať v texte. Väčšina detí začne „čítať“ text ešte raz. Slabší čitateľ to vzdá a radšej zatipuje odpoveď. Ako má dieťa hľadať odpoveď v texte? Ako má zistiť relevantnú časť textu? Odpoveďou je mentálny model, ktorý by si malo dieťa v svojej mysli vytvoriť počas čítania. Možno to prirovnať k tomu, ako sa v prvej triede učí dieťa čítať slová. Najprv mechanicky a neskôr, v druhej triede, začne vnímať aj sémantiku (význam) slova. V odbornej literatúre sa to zdôvodňuje presunom dominantnej aktivity pravej mozgovej hemisféry na ľavú [9]. Po prechode dominancie možno znateľne vidieť ako dieťa „poskočí“ v čitateľskom výkone. Rovnako možno predpokladať, že mentálny model pomôže v súvislom čítaní starších detí, čo by sa malo tiež odzrkadliť zvýšeným čitateľským výkonom.

2. VÝCHODISKÁ EXPERIMENTU

V nasledujúcom je prezentovaná idea nácviku vytvárania mentálneho modelu deja príbehu. Vychádza sa z metódy prerozprávania prečítaného príbehu, ktorá sa zvykne využívať na „klasickom“ vyučovacom predmete čítania. Využije sa vizualizácia prostredníctvom obrázkov. Tiež bežne používaná metóda, kde deti kreslia obrázok k tomu, čo prečítali. Obmena spočíva v tom, že deti budú vytvárať animácie. Štruktúra animácie bude „komixovo“ orientovaná, t.j. postavy budú aj „rozprávať“. Na tento účel sa využije detský programovcí jazyk Scratch [10]. Pozornosť detí nebude odpútaná zdĺhavým kreslením a budú len „režírovať“ (programovať) hercov-postavičky.

Na meranie napredovania detí v úrovni čítania s porozumením sa bude využívať charakteristika rýchlosti hlasného čítania. V [11] poukazujú na úzku koreláciu medzi tempom čítania a porozumením čítaného textu.

Experimentálna výučba sa realizovala v štvrtom ročníku. Vyčlenila sa špeciálna hodina jedenkrát v týždni. Deti už mali skúsenosti s jazykom Scratch. Po dohode s učiteľom Informatickej výchovy, si začiatkom roka Scratch ešte zopakovali a priebežne sa doúčali nové prvky, ktoré sa využívali. Deti predtým absolvovali v 1., 2., 3. ročníku MMČ a majú zautomatizované používanie Multimediálnej čítanky. Dôležité je to aj z pohľadu merania výkonov detí. Prostredníctvom čítania textov cez mikrofón sa zvuk nahrá a spracuje. Následne sa automaticky spočítajú čitateľské charakteristiky [12].

Celkovo možno hodnotiť, že záujem o čítanie ako také v štvrtom ročníku upadá. Deti prepadnú pasívnemu absorbovaniu informácií z televízie alebo internetu. Mohlo by sa zdať, že internet bude motivovať k čítaniu, ale nie je to tak. Za internetom skôr možno vidieť aktivity ako hranie hier, pozeranie videí, prípadne písanie jednoduchých útržkovitých informácií. Aj to bola motivácia priblížiť im prácu s textom pre nich zaujímavejšou formou.

Priebeh experimentálnej výučby bol delený do **trojhodinových cyklov**, ktoré sa opakovali. Spojivom jednej trojhodinovky bola spoločná povesť z knihy Dunajské povesti [13]. Po dohode s učiteľkou boli vybrané primerané texty.

Z povesti sa vytvorila skrátená verzia (cca 300-400 slov), ktorá bola formou príbehu v Multimediálnej čítanke použitá na **1. hodine cyklu**. Priebeh tejto hodiny nebol pre deti novinkou. Podobne postupujú aj v nižších ročníkoch na MMČ. Z dlhodobého hľadiska bolo odsledované, že použitie dlhších textov spôsobí, že slabší čitatelia nestihnú text počas hodiny ani dočítať, čo je potom problém z hľadiska sledovania čitateľských výkonov [12]. Z technického hľadiska sa automaticky spočíta tempo a plynulosť čítania iba po prečítaní celého textu. Tento typ hodiny musel byť prvý aj z iného dôvodu. Aby boli merateľné charakteristiky čítania hodnoverné, je potrebné deťom predložiť neznámy text. Túto hodinu možno označiť ako **intenzívnu**. Každé dieťa číta a rieši pripravené úlohy. Ak niektoré dieťa skončilo skôr, počúvalo ako čítali spolužiaci a hodnotilo ich formou krátkej textovej správy.

Nasledujúca **2. hodina cyklu** mala priebeh klasického skupinového čítania „v kruhu“. Deti sa striedali pri čítaní plnej verzie povesti. Vybrané povesti sa vyznačujú aj tým, že obsahujú viacero neznámych (zastaralých) slov, ktoré bolo treba vysvetliť. Túto hodinu možno označiť ako **oddychovú**. Deti pasívne počúvali čítanie, prípadne chvíľu čítali nahlas.

Posledná **3. hodina cyklu** pozostávala z pripomenutia si prečítanej povesti. Následne sa spoločne identifikovala a dohodla scénka, ktorú deti animovali. Priebeh tvorby animácie pozostával vo voľbe 2-3 hercov-postavičiek a dokreslenie kulís. Následne sa „programoval“ scenár. Prvotné animácie boli jednoduchšie komixové dialógy, kde sa využila časová synchronizácia. Neskôr pribudol pohyb a posielanie správ. Deti boli stále udržované v tom, že v Scratch-i sú len herci, ktorí si prezliekajú kostýmy a hrajú spoločné divadlo pre diváka. Preto sa ani nehovorilo o programovaní, ale o vytvorení scenáru. Túto hodinu možno označiť ako **tvorivú**. Deti na základe prečítaného zhmotňovali svoju predstavu (mentálny model) povesti.

3. REALIZÁCIA

Experimentu sa zúčastnili dve štvrtácke triedy (2x15 detí) a jedna štvrtácka trieda zdravotne oslabených žiakov (10 detí). Celkový počet detí bol 40. Z toho chlapcov 18 a dievčat 22. Vekovo boli v rozmedzí 9-10 rokov. Pre porovnanie sa využili nazbierané údaje čitateľských charakteristík zo školského roka 2013/14. Boli to tiež štvrtáci. Konkrétne trieda zdravotne oslabených žiakov (9 detí). Chlapci boli 4 a dievčat bolo 5. Tieto deti každý týždeň prečítali a pracovali s jedným príbehom v Multimediálnej čítanke [14].

V experimente boli využité povesti na základe výberu učiteľky (pozri tabuľku 1). Boli vyberané priebežne. K povestiam v pôvodnom znení boli vytvorené skrátené verzie do Multimediálnej čítanky. Podľa [15] bolo vyčíslené skóre čitateľnosti skráteného textu FRES (Flesch Reading-Ease Score)

FRES = 206.835 – 1.015*(#slov / #viet) – 84.6*(#slabík / #slov), kde znak # označuje početnosť.

Z tabuľky 1 je zrejmé, že texty sú (podľa FRES) náročné. Aspoň čo sa týka syntaktického hľadiska. Originálne plné texty neboli k dispozícii v digitálnej forme a tak k nim neboli vyčíslené charakteristiky.

Na priebeh prvej hodiny cyklu boli deti už zvyknuté. Texty boli však trochu náročnejšie, obsahovali aj ťažšie, už nepoužívané, slová (archaizmy). Každé dieťa prečítalo celý text. V závislosti od dĺžky textu čítali cca **3-5 min**. Samozrejme, že slabší čitatelia primerane viac. Ku príbehu bolo nachystaných cca 10 otvorených otázok, ktoré riešili samostatne vo vlastnom tempe. Niekedy bol problém s tým, že pomalší čitateľ sa ukrátil o čas a nestihol zodpovedať všetky otázky. Učiteľky radili spraviť špeciálne ešte jednu kratšiu verziu povesti pre slabších čitateľov. To by však bolo treba zmeniť aj otázky. Riešenie možno vidieť

skôr v tom, že text bude kratší pre všetkých a pre rýchlejšie deti sa realizuje ešte doplnková aktivita na zamestnanie.

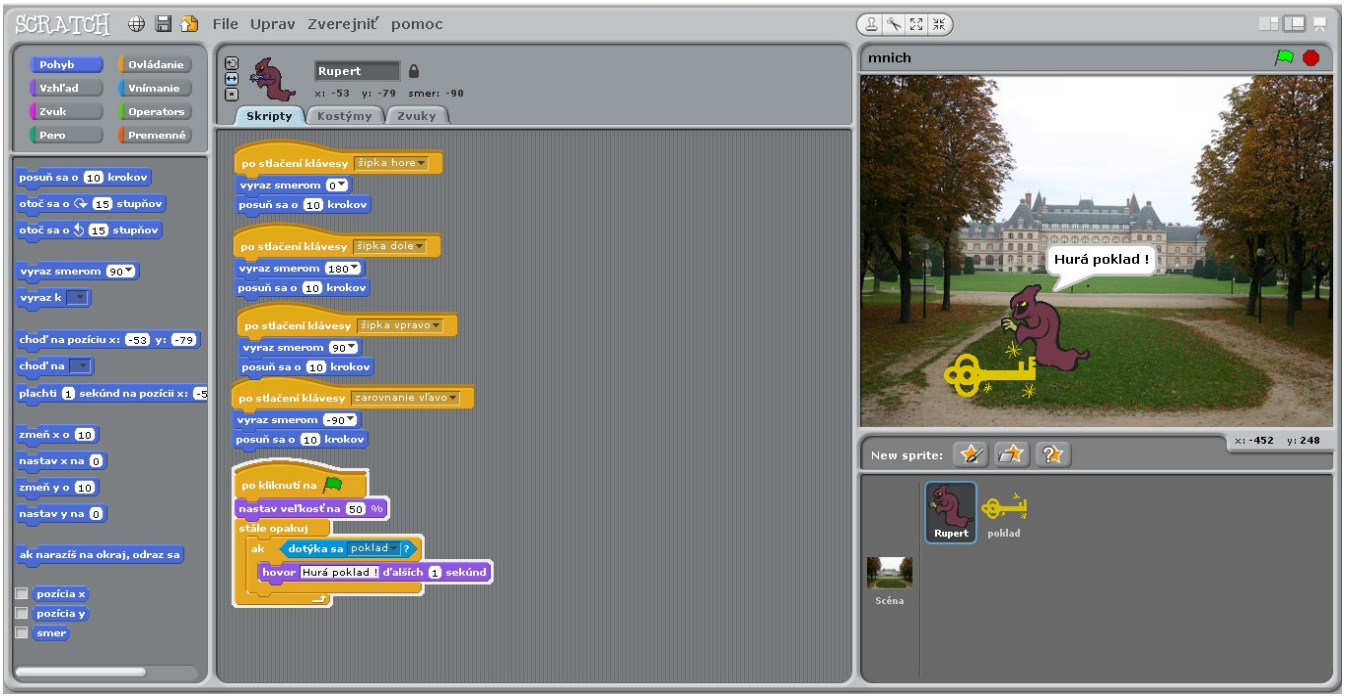
Tabuľka 1. Zoznam použitých povestí

	Skrátená verzia	
Povesť	**Počet slov**	**FRES**
Dunajská kráľovná	289	32.8
Panenská veža na Devíne	402	20.4
Žabia studňa	335	40.6
Povesť o plačúcej guli	271	18.2
Rolandova studňa	273	16.8
Bratislavský Faust	481	23.3
Kačacia fontána	416	34.5
Bukanova dcéra	341	33.5
Stolček	299	23.3

Čítanie „v kruhu“ na druhej hodine cyklu vykazovalo typické prvky nezáujmu detí. Viacero detí blúdilo zrakom po miestnosti a nesledovalo text. Ak čítal nahlas slabší čitateľ, boli nervózne všetky deti. Ak čítal výkonnejší čitateľ, slabší nestíhali a vyrušovali susedov, aby zistili, kde sa číta. V konečnom dôsledku každé dieťa nahlas čítalo iba cca **2 minúty**, zvyšok času bolo dieťa pasívne. Navyše vysvetľovanie významu náročných zastaralých slov v texte spomaľovalo napredovanie. Na oživenie hodiny sa k danej povesti premietla sfilmovaná verzia (večerníček) z archívu Slovenskej televízie.

Obrázok 1. Ukážka k povesti „O plačúcej guli”. Postavička môže vykonávať paralelne viacero súbežných akcií. Sud sa točí a zároveň „hovorí“.

Tvorenie animácii v Scratch-i na tretej hodine cyklu deti zaujalo a dožadovali sa ho. To, čo sa bude animovať, sa dohodlo vždy na začiatku hodiny. Možnosť výberu je silný motivačný faktor pre ďalšiu prácu. Zo začiatku bolo treba deti usmerňovať z „programátorského“ hľadiska. Neskôr už deti sami navrhovali, čo by postavy mohli robiť. Bolo treba ustriehnuť obtiažnosť navrhovanej scénky, aby väčšina detí animáciu na hodine dokončila. Na konci hodiny deti iniciovali aj spoločné



Obrázok 2. Ukážka k povesti „O plačúcej guli“. Ovládanie postavičky za zlatom bažiaceho mnícha klávesami.

prezentovanie hotových scénok cez dataprojektor (viď obrázky 1, 2). Z časového hľadiska to nebolo vždy možné.

Na synchronizáciu činností postavičiek v Scratch-i bol zo začiatku zvolený čas. Deťom robilo problémy správne spočítať koľko má postava čakať, aby mohla previesť ďalšiu akciu podľa plánu.

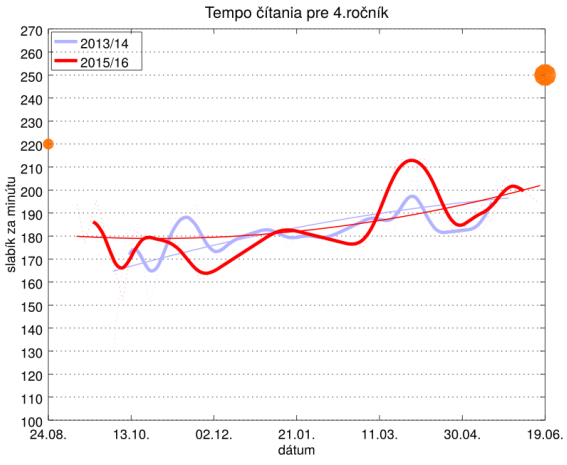
postava 1	**postava 2**	**postava 3**
hovor A sek	čakaj A sek	čakaj A+B sek
čakaj B+C sek	hovor B sek	hovor C sek
…	čakaj C sek	…

Deti zväčša spúšťali scénku stále dookola a časy nastavili štýlom pokus-omyl. Pri dlhšej scénke museli nájsť a používať istý systém, začali rátať. Poväčšine to úspešne zvládli. Neskôr sa prešlo na synchronizáciu pomocou správ. Problematické bolo správne správy pomenovať. A pri ich väčšom množstve to od detí vyžadovalo väčší mentálny manažment. Niektoré deti sa strácali.

Zaujímavý moment bol, keď deti zistili, ako môže postavička vykonávať viacero akcií súbežne. Problém bol však opäť zo synchronizáciou.

4. VÝSLEDKY

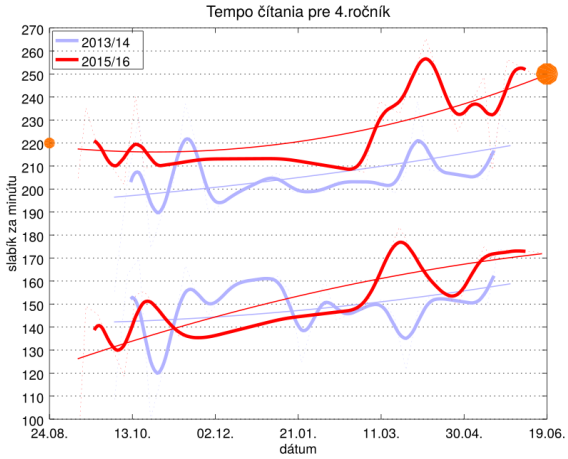
Na obrázku 3 možno vidieť graf vývoja tempa čítania zúčastnených detí. Pre porovnanie je zobrazený vývoj čítania žiakov zo školského roka 2013/14. Graf je zostrojený na týždennej báze. Na samotný výkon majú vplyv viaceré faktory. Napríklad cez ktorú vyučovaciu hodinu sa meranie realizuje, alebo návrat detí z prázdnin, prerušenie výučby pri štrajku učiteľov, … Reálne dáta majú premenlivú podobu. Aby bol graf vyhladenejší, navzorkoval sa a spriemeroval sa dolnopriepustným digitálnym filtrom (aplikovanie konvolúcie). Tento spôsob je volený hlavne z dôvodu postupného získavania a zobrazovania údajov. Po zozbieraní dát z celého školského roka je možné použiť polynomiálnu regresiu rádu 2. Oproti rádu 1 (priamka) je možné sledovať aj „kritický“ bod, reprezentovaný polrokom.



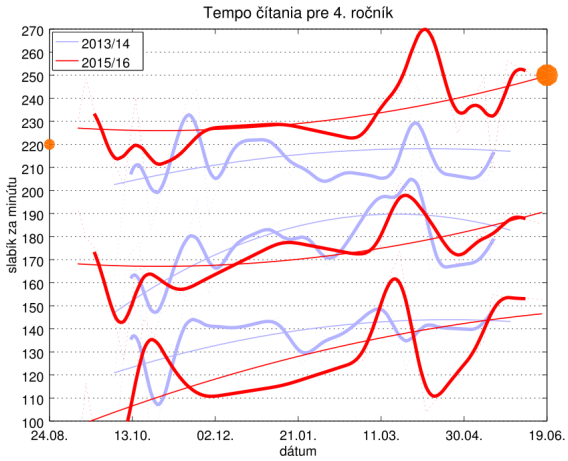
Obrázok 3. Graf priemerného napredovania štvrtákov v tempe čítania. Hrubšia čiara zodpovedá vyhladenej krivke a tenká čiara aproximácii polynómom.

V grafe sú zobrazené žltými bodmi aj úrovne hlasného čítania, ktoré by deti mali v priemere dosahovať počas 4. ročníka podľa [11]. Do výpočtu tempa čítania však, v našom prípade, nebola zahrnutá korekcia nesprávne prečítaných slov nakoľko sa jednalo iba o malé percento. Pri vyššom počte nesprávne prečítaných slov automaticky zareagoval počítačový rozpoznávač reči a nahrávku zamietol.

Okrem klasického štatistického priemeru v týždňových intervaloch je zobrazený aj 2-priemer (pozri obrázok 4). T. j. reprezentovanie dát dvomi hodnotami (využitím k-mean clustering algoritmu). K takejto vizualizácii sa pristúpilo na základe sledovania čitateľských výkonov detí z rôznych ročníkov. V každej triede možno znateľne identifikovať lepšie a slabšie čitateľské výkony, ktoré vytvárajú v nameraných dátach zhluky. Pokým priemer reprezentuje dáta ako jeden zhluk, má väčšiu chybu (rozptyl). Ak sa v dátach identifikujú dva zhluky so svojimi priemermi, ich rozptyly sú menšie. Ideálne by bolo využiť 3-priemer na reprezentáciu dobrých, priemerných a slabých čitateľských výkonov. Na obrázku 5 je orientačne zobrazený taký graf. Problematický však môže byť nízky počet detí z porovnávaného roka 2013/14.



Obrázok 4. Vývoj tempa čítania vo forme 2-priemeru.



Obrázok 5. Vývoj tempa čítania vo forme 3-priemeru.

Z obrázku 5 je zrejmé, že skupina lepších výkonov zaznamenala progres oproti začiatku šk. roka cca o 30 slabík/min. Nemožno však jednoznačne povedať, že je to nemenná skupina

detí, ale v užšom poňatí obsahuje dobrých čitateľov. Skupina priemerných výkonov postúpila o cca 20 slabík/min. A slabé výkony postúpili cca o 40 slabík/min oproti začiatku šk. roka.

Oproti údajom z roku 2013/14 vidieť hlavne „pokrok“ lepších čitateľov. Tam možno predpokladať, že ku čítaným príbehom pristupujú už viac „z nadhľadu“ a budujú si mentálny model, ktorý im pomáha v lepšej predikcii čítaného textu. Nie sú príliš upätí na dekódovanie slov a viet.

5. ZÁVER

Využitie animácií ako scénok z príbehu sa osvedčilo. Rozhodne možno tento typ hodiny odporúčať. Inak je to s čítaním „v kruhu“. Do budúcna treba určite premyslieť nahradenie tohto typu priebehu vyučovacej hodiny.

Niekedy bolo do vyučovania začlenené opakovanie. Spolu s učiteľkou deti spomínali na povesti, ktoré už čítali. Je zaujímavé, že pri spomínaní si na obsah povesti, si deti skôr spomenuli na to, čo animovali a z toho potom na celú povesť, ktorú spoločnými silami prerozprávali.

POĎAKOVANIE

Vyjadrujem poďakovanie deťom a učiteľkám zo ZŠ, Černyševského 8, Bratislava za možnosť realizovať experiment.

BIBLIOGRAFICKÉ ODKAZY

[1] MOSTOW, J., ROTH, S.F.: Demonstration of a reading coach that listens. In Hauptmann, A.G., ed.: Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology. UIST ’95, New York, NY, USA, ACM (1995) 77–78

[2] Project LISTEN, https://www.cs.cmu.edu/~listen/

[3] BOLANOS, D., COLE, R.A., WARD, W., BORTS, E., SVIRSKY, E.: Flora: Fluent oral reading assessment of children’s speech. ACM Trans. Speech Lang. Process. 7(4) (August 2011) 16:1–16:19

[4] PATEL, R., FURR, W.: Readn’karaoke: Visualizing prosody in children’s books for expressive oral reading. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI ’11, New York, NY, USA, ACM (2011) 3203–3206

[5] AIST, G.S., MOSTOW, J.: Adapting Human Tutorial Interventions for a Reading Tutor that Listens: Using Continuous Speech Recognition in Interactive Educational Multimedia. In CALL'97 Conference on Multimedia. Exeter, England, September, 1997.

[6] BECK, J. E., MOSTOW, J., CUNEO, A., BEY, J.: Can automated questioning help children's reading comprehension? Proceedings of the Tenth International Conference on Artificial Intelligence in Education (AIED2003), Sydney, Australia, 2003.

[7] MILLS-TETTEY, A., MOSTOW, J., DIAS, M. B., SWEET, T. M., BELOUSOV, S. M., DIAS, M. F., GONG, H.: *Improving Child Literacy in Africa: Experiments with an Automated Reading Tutor.* 3rd IEEE/ACM International Conference on Information and Communication Technologies and Development (ICTD2009), 129-138. Carnegie Mellon, Doha, Qatar, 2009.

[8] Multimediálna čítanka, https://www.mmcitanka.sk

[9] JOŠT, J.: Čtení a dyslexie, Grada Publishing, a.s., 2011

[10] Scratch, https://scratch.mit.edu

[11] HASBROUCK, J. AND TINDAL, G. A.: Oral Reading Fluency Norms: A Valuable Assessment Tool for Reading Teachers. Journal of The Reading Teacher, 59, pp. 636–644, 2006.

[12] NAGY, M.: Experiment výučby čítania prostredníctvom Multimediálnej čítanky, DIDINFO 2013, Banská Bystrica, Slovakia, 2013

[13] ĎURÍČKOVÁ, M.: Dunajské povesti, Mladé letá, 1996

[14] NAGY, M.: Multimediálna čítanka ako súčasť výučby čítania, Učiteľské noviny, č.42, Bratislava, 2014

[15] FLESCH, R.: A New Readability Yardstick, Journal of Applied Psychology, 32, pp. 221-233, 1948.

Výuka principů objektově orientovaného programování při tvorbě her

Teaching of principles of object-oriented programming in the creation of games

Ingrid Nagyová
Pedagogická fakulta
Ostravská univerzita v Ostravě
Mlýnská 5
701 03 Ostrava
Česká republika
ingrid.nagyova@osu.cz

ABSTRACT

In this paper we describe the process of teaching of the basic principles of object-oriented programming in a group of students, future teachers of informatics at primary schools. Projects are thematically focused on simple computer games that are implemented in teaching and on which are presented the basic elements of object-oriented programming. Classes and objects are gradually defined during the teaching of basic principles of programming and of them are created games of various difficulties – the unending games, the games that are controlled by a mouse or keyboard, the graphic games or the games with a lot of objects. Game development is for students motivating and interesting. At the same time it also follows didactic aims - to show the possibility to students how they can work in teaching programming with children in primary school.

Keywords

Object-oriented programming. Python. Encapsulation. PC game. Object. Graphic object.

ABSTRAKT

V příspěvku popíšeme proces výuky základních principů objektově orientovaného programování ve skupině studentů, budoucích učitelů informatiky na základních školách. Tematicky jsou projekty, které jsou v rámci výuky realizovány a na nichž jsou základní prvky objektově orientovaného programování prezentovány, zaměřeny na jednoduché počítačové hry. Při výuce základních principů programování postupně vznikají třídy a objekty, z kterých jsou vytvářené hry různé obtížnosti – hry, které se hrají samy, hry ovládané myši či klávesnicí nebo graficky zpracované hry využívající mnoho objektů. Tvorba her je pro studenty motivující a zajímavá. Současně sleduje také didaktické cíle – ukázat studentům možnosti, jak lze ve výuce programování pracovat s dětmi na základní škole.

Klíčová slova

Objektově orientované programování. Python. Zapouzdření. Počítačová hra. Objekt. Grafický objekt.

1. ÚVOD

Studenti připravující se na Pedagogické fakultě Ostravské univerzity na povolání učitele informatiky na základních školách mají často zkreslené a omezené pojetí informatiky jako vědy. Rámcový vzdělávací program pro základní vzdělávání [1], který jako historicky první z kurikulárních dokumentů v České republice počítá se vzdělávací oblastí Informační a komunikační technologie, omezuje vzdělání v této oblasti pouze na uživatelské dovednosti v práci s počítačem. Toto vymezení formuje vztah studentů k informatice jako vědě, která je tak značně omezena zejména v oblasti algoritmického přístupu k řešení problémů, v oblasti algoritmizace a programování [2]. Změnu lze očekávat s postupným naplňováním Strategie digitálního vzdělávání [3], která počítá s úpravou Rámcového vzdělávacího programu, s rozvojem digitální gramotnosti a zejména informatického myšlení.

V současnosti je příprava studentů, a tím i jejich motivace ke zvládnutí algoritmizace a učení se základům programování, velice obtížná. V rámci bakalářského studia studenti absolvují dva semestry výuky algoritmizace, v rámci kterých se seznámí s nejznámějšími algoritmy a postupy řešení problémů v informatice. Zvládnou práci s proměnnými základních datových typů, naučí se využívat základní programové konstrukce – sekvence, větvení, iterace, členění programu do podprogramů apod. Absolvují také úvod do abstraktních datových typů a naučí se základům práce s nimi i s ohledem na různé implementace dat. Předmět zaměřený na objektově orientované programování následuje až po úvodních kurzech algoritmizace a programování.

Příspěvek je zaměřen na výuku základních principů objektově orientovaného programování, která je určena pro studenty, budoucí učitele informatiky. Výstupem z výuky jsou jednoduché hry vytvářené v prostředí programovacího jazyka Python, který studenti znají z předchozí výuky algoritmizace. Počítačové hry jsou využity s cílem zvýšení motivace k práci a učení se. Využitím her jako tematického zaměření vytvářených děl sledujeme také didaktické cíle – ukázat studentům možnosti práce v objektově orientovaném prostředí jednoduchou a hravou formou, která by mohla být s modifikacemi uplatnitelná také při výuce žáků na základních školách.

2. ASPEKTY VÝUKY OBJEKTOVĚ ORIENTOVANÉHO PROGRAMOVÁNÍ

Objektově orientovaný přístup k řešení problémů patří dnes k nejpoužívanějším konceptům. Jeho rozšíření a využitelnost je dnes na takové úrovni, že se doporučuje výuku programování začít právě tímto konceptem (viz například také myšlenka návrhových vzorů [4]) – oproti klasické výuce vycházející

z imperativního paradigmatu. Rozhodnutí o použitém konceptu výuky (objektové versus imperativní paradigma) nutně ovlivňuje proces a postup výuky. Naznačíme proto důvody, které nás vedou k postupu výuky naznačenému v úvodu – od základů algoritmizace, základů imperativního programování k programování objektově orientovanému.

První argumentem je historický vývoj informatiky, která má staletou tradici ve vývoji nejrůznějších algoritmů, v optimalizaci výpočtů, v zjišťování složitosti algoritmů apod. Tyto poznatky, i přesto, že se celá staletí vyvíjely spíše v rámci matematiky, tvoří důležitou část myšlenkových konceptů informatiky, které by rozhodně neměly ani v dnešní době chybět v žádném informatickém vzdělávání. Algoritmická řešení problémů, často velice úspěšná a sofistikovaná, není nezbytně nutné převádět do objektové reprezentace. Objektově orientované programování by mělo být využito až tam, kde dojde k získání lepších výsledků, k vyšší přehlednosti nebo zjednodušení.

Často se deklaruje, že objektově orientovaný přístup je přehlednější a více odpovídá reálnému světu. Je nutné si uvědomit, že objektová reprezentace reálného světa vyžaduje jistou míru abstrakce, analytický přístup a abstrahování důležitých vlastností reálného světa. Ve výuce začíná proces osvojování si poznatků jejich zapamatováním (viz Bloomova taxonomie [5]) a pochopením na konkrétních příkladech, které u žáka vedou k tvorbě separovaných modelů. K zobecnění takto získaných poznatků, k tvorbě generických modelů, k jejich abstrahování a následné krystalizaci poznání dochází mnohem později [6]. Tento proces získávání poznatků je ověřen historickým vývojem poznání v informatice a je vhodné jej ve výuce následovat. To právě umožňuje výuka algoritmů, jejich procvičování, aplikace, drobné modifikace, ale také ověřování jejich správnosti a efektivnosti.

Jedním z důležitých principů informatiky, který přímo souvisí s objektově orientovaným přístupem, je myšlenka zapouzdření. Lze sledovat několik etap vývoje myšlenky zapouzdření [7]:

- využití podprogramů, které zapouzdřovaly pouze části kódu, netýkaly se přímo zapouzdření dat;
- využití lokálních proměnných, kde jsou operace podprogramu zapouzdřeny společně s potřebnými daty uloženými v lokálních proměnných;
- abstraktní datové typy umožňují odvodit libovolný počet prvků daného typu, neumožňují ale odvozování nových datových typů;
- objektově orientovaný přístup – třída odpovídá abstraktnímu datovému typu, objekt vzniká z třídy, dědičnost umožňuje odvodit a upravit nové datové typy.

Na příkladu historického vývoje myšlenky zapouzdření lze chápat objektově orientovaný přístup jako jeden vývojový stupeň, který rozhodně nevznikl náhodně a který může naznačit následující vývoj v informatice. Domníváme se, že by ve výuce měl být historický vývoj jednotlivých myšlenkových konceptů následován [7]. Při procházení jednotlivými etapami mají žáci možnost sledovat problémy jednotlivých etap a s pomocí učitele mohou znovuobjevovat, jakým způsobem byly problémy řešeny [8]. Krátké ohlédnutí do historie vede k myšlence zařadit výuku algoritmů a imperativního programování před výuku a využití objektově objektového paradigmatu.

3. POSTUP VÝUKY OBJEKTOVĚ ORIENTOVANÉHO PROGRAMOVÁNÍ

Historický vývoj myšlenky zapouzdření – jak je sledován v celkovém konceptu výuky – je vhodné následovat také při výuce objektově orientovaného programování. Začínáme proto datovým modelem řešeného problému, často také s využitím myšlenky abstraktních datových typů. Následně vysvětlíme vztahy mezi třídami a objekty a možnosti komunikace (posílání zpráv) mezi instancemi.

Pro tvorbu her je důležité grafické rozhraní hry a proto se od začátku zaměřujeme na definice tříd vybraných jednoduchých grafických objektů – kruhů, čtverců, křížků apod.

3.1 Datová struktura třídy

Základním stavebním prvkem objektově orientovaného programování je třída, která má svoji datovou a výkonovou část. Začneme příkladem, který to názorně demonstruje.

Příklad – Kreslení kolečka:

Chceme vykreslit kolečko, barevný kruh. Co všechno potřebujeme znát k tomu, abychom to mohli udělat?

Komentář: Obvykle studenti rychle odhalí, že potřebujeme vědět, jaké barvy kolečko bude, jak bude velké (poloměr), kde se bude nacházet (souřadnice, případně grafická plocha). Může se stát, že studenti problém nepochopí. Pak je dobré otázku doplnit: *Co všechno potřebujete znát, abyste nakreslili kruh na papír?*

Třída je šablona, předpis, podle kterého vytváříme objekty. Nejdůležitější metodou třídy je metoda, která inicializuje objekt, instanci. Tato metoda inicializuje všechny atributy (vlastnosti) objektu, v našem případě grafické plátno, souřadnice středu, poloměr, barvu, ale i identifikátor kolečka (kruhu). Uvedené vlastnosti náleží jednotlivému kolečku a představují datovou strukturu objektu.

```
class Kruh:
  def __init__(self, g, x, y, r, barva='red'):
    self.g = g
    self.x = x
    self.y = y
    self.r = r
    self.barva = barva
    self.id = self.g.create_oval(
      self.x-self.r,self.y-self.r,
      self.x+self.r,self.y+self.r,
      fill=self.barva)
```

Obrázek 1. Inicializace instance kruh

Již takto jednoduše definovaný objekt (viz obrázek 1) lze vykreslit vytvořením objektu dané třídy. Podle zadaných parametrů můžeme vytvořit i několik různých koleček, několik prvků daného typu, podobně, jako je tomu u abstraktních datových typů (viz vývoj myšlenky zapouzdření).

Současně se také můžeme ptát, jaké operace lze s kolečky provádět, jaké metody bychom mohli nadefinovat. S ohledem na definované vlastnosti třídy (objektů) se nabízí možnost změny pozice (posun kolečka), změny velikosti (poloměru) či změny barvy. Definici třídy lze doplnit uvedenými operacemi ve formě metod. Při jejich definici lze využít základních programových struktur – sekvenci, iteraci a alternativu.

Komentář: Obvykle se ve výuce setkáme se studenty, kteří inicializační metodu `__init__` označí pojmem konstruktor.

Tehdy je dobré pokusit se s odkazem na jiné programovací jazyky dovést studenty k tomu, že konstruktor by se mohl jmenovat `__new__` a tuto metodu do definice třídy přidat. Jednoduchým výpisem informací z obou metod ukážeme, že při vytváření objektu je nejprve volán konstruktor `__new__`, který následně volá inicializační metodu `__init__` (viz obrázek 2).

```
class Cislo:
    def __init__(self, cis):
        print('init')
        self.cis = cis

    def __new__(cls, cis):
        print('new')
        return super().__new__(cls)
a=Cislo(15)
```

Obrázek 2. Konstruktor a inicializace instance

Kromě kolečka lze podobným způsobem připravit třídy pro další grafické objekty, které využijeme při tvorbě her, například čtverec, obdélník, křížek, jednoduchý obrázek apod. Při kreslení těchto útvarů si studenti procvičují postup při deklaraci třídy a zjišťují drobné odlišnosti, zejména v definovaných vlastnostech – čtverec má délku strany, křížek délku ramene a dva identifikátory pro dvě překřížené čáry apod. Metody těchto tříd se obvykle liší pouze nepatrně. Při návrhu jednoduchých tříd využíváme pomůcku pro návrh tříd a objektů, která byla vytvořena v rámci diplomové práce [9].

Pokud používáme objektově orientované programování na úrovni deklarace třídy, představuje třída pouze vhodně strukturovaný zápis kódu – datová část třídy je propojena s částí výkonovou. Vlastnosti a principy objektově orientovaného programování se zde výrazněji neprojeví a v oblasti vývoje myšlenky zapouzdření zůstáváme na úrovni abstraktních datových typů – prozatím nevytváříme nový datové typy.

3.2 Vztahy mezi třídami a instancemi

Výhody a možnosti objektově orientovaného paradigmatu se projeví, pokud začneme zkoumat vztahy mezi třídou a objekty nebo mezi třídami navzájem. Mezi objekty lze vyhledat vztahy asociace, agregace a kompozice a najít vhodné příklady.

Důležitým vztahem, který je specifikem objektově orientovaného programování, je vztah dědičnosti (generalizace). Jedná se o hierarchický vztah tříd, v němž třída, která je potomkem, dědí vlastnosti a metody třídy, která je předkem. Kromě zděděných charakteristik může mít potomek také svoje specifické vlastnosti a zděděné charakteristiky mohou být také modifikovány. Dědičnost lze vysvětlit na grafických objektech, které studenti vytvářeli.

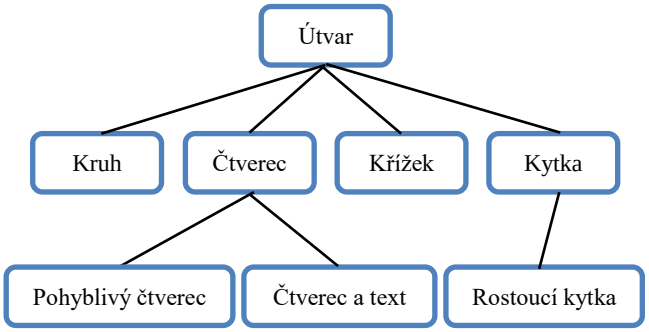
Příklad – Definice pra-útvaru:

Co mají vytvářené grafické objekty společné? Které vlastnosti a které metody jsou jim společné?

Komentář: Odpovědi na uvedené otázky se snažíme hledat společně se studenty. I přesto, že vytvářené grafické prvky jsou rozmanité, mají společné znaky. Společnými vlastnostmi jsou grafické plátno, souřadnice polohy objektu, barva a identifikátor objektu. Společnými metodami jsou metody pro posun objektu, pro změnu barvy a velikosti apod.

Společné vlastnosti a metody grafických objektů specifikují pra-útvar, který je předkem všech dalších grafických útvarů. V hierarchii dědičnosti je pra-útvar základem pro jednotlivé, již předtím definované, grafické objekty – kruh, čtverec, křížek apod. (viz obrázek 3). Ty mohou být dále rozšířeny přidáním vhodných vlastností (například vypisovaný text, jeho barva apod.) nebo metod pro práci s novými vlastnostmi či pro rozšíření stávající funkčnosti objektů (například automatický pohyb) – viz další úroveň hierarchie v diagramu dědičnosti.

Důležité je vrátit se k inicializaci instance a nutnosti inicializovat vlastnosti předka (volání `super.__init__()`). Podobně mohou nové metody využít funkcionality metod předka a tyto pouze dále rozšířit.



Obrázek 3. Hierarchický diagram grafických tříd

Dědičnost je důležitou charakteristikou objektově orientovaného programování, která umožňuje definovat nové datové typy (třídy) rozšířením stávajících. Vznikající vztahy mezi třídami a objekty představují výrazný skok ve vývoji myšlenky zapouzdření – data zapouzdřená v objektu jsou přístupná prostřednictvím metod daného objektu, které samy mají právo rozhodnout o zpřístupnění dat nebo o jejich změně.

3.3 Komunikace mezi instancemi

Vztahy mezi třídami a objekty vnáší do práce s informacemi dynamický prvek. Objekty si mohou posílat zprávy a komunikovat mezi sebou. Pravidla pro komunikaci jsou stanovena tak, aby umožňovala přístup k vlastnostem objektů (pro čtení i zápis a změnu vlastnosti), nejlépe pouze prostřednictvím metod, které tak definují komunikační rozhraní. Formy komunikace mezi třídami a objekty mohou být různé.

V návaznosti na grafické objekty jsou nejjednodušší formou komunikace zprávy jako „zjisti pozici útvaru“, „změň poloměr kruhu na 20 bodů“ nebo „posuň se o 2 body vpravo“. Tyto zprávy odkazují přímo ke konkrétním objektům. Pokud v dané části programu existuje odkaz na objekt, odpovídají tyto zprávy práci s vlastnostmi objektu (čtení nebo změna vlastnosti) nebo volání příslušných metod objektu s vhodnými parametry.

Složitějšími zprávami jsou zprávy mezi objekty různých tříd. Pokud jsou na louce rozesety květiny a prší (padají malé modré kuličky – viz obrázek 4), pak kapky deště mohou odesílat zprávy kytkám, aby rostly, když se nachází v jejich blízkosti a zavlažují je. Zpráva kapky deště může být odeslána kytce tam, kde jsou dostupné kytky i kapky, resp. odkazy na ně. Takovým objektem může být objekt reprezentující louku, jehož vlastnostmi jsou padající kapky i kvetoucí kytky. Podle vzdálenosti kytky a kapky je volána metoda kytky, která reaguje na zprávu `padla_kapka` a která kytku zvětší (zaleje ji).

Objektově orientované programování nabízí možnost definice dvou objektů různých tříd, které podporují stejnou množinu zpráv (stejné metody), ale definuji svůj vlastní model reakce na zprávu,

vlastní model výpočtu. Na louce můžeme mít kytky, které při setkání s kapkou vody povyrostou. Můžeme tam přidat žabky, které při setkání s kapkou naopak uskočí bokem. Na zprávu `padla_kapka` tak reagují objekty pro kytky i žabky různě. Tyto objekty můžeme udržovat v jednom poli a můžeme s nimi zacházet stejným způsobem – zasílat jim stejné zprávy. Jednotlivé objekty se při zpracování zprávy budou chovat každý podle svého. Při zpracování zpráv docházíme k polymorfizmu jako poslední z důležitých vlastností objektově orientovaného programování.

Spolupráce objektů a sdělení zpráv jsou základní dynamické koncepty objektově orientovaného programování, které musí být dobře pochopeny a procvičeny tak, aby je studenti byli schopni využít ve svých projektech. Důraz je nutné klást zejména na parametry metod – vstupní parametry představují požadavky zprávy, výstupní hodnoty reprezentují odpovědi na zadaný požadavek. Zpočátku se studenti učí kooperaci objektů pouze napodobováním a zkoušením. S postupným nabýváním zkušeností v jednotlivých modelových situacích dochází ke krystalizaci poznání a k schopnosti samostatně spolupráci objektů řídit.

4. POSTUP PŘI TVORBĚ HER

Tematickým zaměřením projektů řešených v rámci objektově orientovaného programování jsou hry. Ty kromě grafických prvků, jejichž tvorba umožňuje seznámit se se základy objektově orientovaného programování, vyžadují pohyb objektů a možnost ovládání hry pomocí myši nebo klávesnice.

4.1 Proč využíváme hry?

Naznačená výuka objektově orientovaného programování je určena studentům učitelství informatiky pro základní školy. V rámci přípravy na svoji budoucí profesi se seznamují s jednotlivými aspekty algoritmizace a programování v prostředí Python. Jedná se o vysokoúrovňový skriptovací programovací jazyk, který svým rozsahem a možnostmi daleko přesahuje potřeby a úroveň znalostí učitele základní školy. Snažíme se proto vybírat pouze základní aspekty programovacího jazyka a výuku studentů organizovat tak, aby naučené postupy mohli sami aplikovat při práci s žáky. Volíme tematické zaměření z oblasti, která je zajímavá nejenom pro studenty, ale která by mohla být zajímavá a při hlubší práci i efektivní pro žáky na základní škole. Hry, jejich tvorba, definice a nastavování objektů, ale také možnost ovlivňovat průběh hry zpracováním událostí jsou důležité aspekty uplatnitelné při programování nejenom v jazyce Python, ale také při práci v dětských programovacích prostředích (například Scratch [10] nebo App Invertor pro mobilní zařízení [11]), které rovněž využívají objektově orientované paradigma.

Využití her má také motivační charakter – studenti hry znají, znají různé varianty a herní strategie a často mají sami touhu si nějakou hru vytvořit.

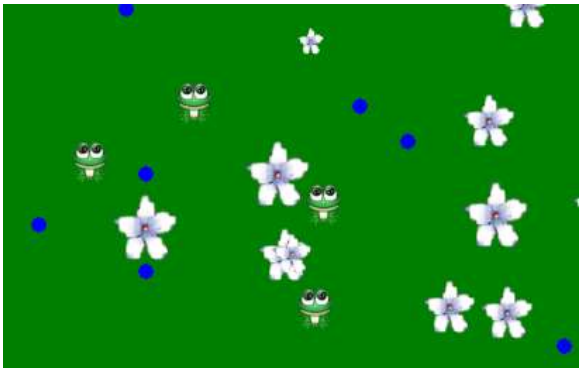
4.2 Nekonečné hry

První vytvářenou hrou je donekonečna se opakující hra, která běží podle daných pravidel sama, bez zásahu zvenčí.

Hra 1 – Nekonečný příběh louky:

Na louce se nachází kytky a žáby. Na louku neustále prší – malé kuličky. Když kapka deště dopadne na kytku, kytka naroste, když dopadne na žábu, žába odskočí (viz obrázek 4).

Tato hra vyžaduje jednak využití skupiny objektů – na louce je mnoho kytek, mnoho žab, ale také mnoho kapek deště. Pro pohyb kapek a jejich neustálé „generování“ je nutné využít časovač.



Obrázek 4. Nekonečný příběh louky

4.2.1 Skupiny objektů

Skupiny grafických objektů řadíme do pole, které umožňuje snadný přístup k jednotlivým prvkům (objektům) pomocí indexu a také snadné přidávání a mazání prvků z pole.

Pokud máme skupiny objektů různých tříd je potřeba promyslet, zda je řadit do jednoho pole nebo využít více polí pro každou třídu zvlášť. V našem případě jsme všechny použité objekty uložili do jednoho pole. Všechny objekty jsou grafické objekty tříd, které vychází z pra-útvaru `Utvar`. Kytky a žáby využívají metodu `padla_kapka`, která má pro tyto typy objektů rozdílnou funkčnost (růst nebo skok). Kapky deště lze v poli rozlišit podle názvu třídy, z které vznikly a aplikovat na ně potřebné operace.

4.2.2 Pohyb objektů – časovač

Kromě skupiny objektů vyžaduje „nekonečná“ hra pohyb. Ten lze zajistit nekonečným opakovaným voláním metod s využitím časovače, který proces pohyb objektů na jistou dobu několika milisekund pozastaví.

V projektu s loukou využívají kuličky časovač k imitaci deště, tj. k padání dolů. Náhodnost pohybu dešťových kuliček je simulována náhodným krokem posunu. Časovač je využit také pro generování nových kuliček (kapek deště). Zde využijeme rozdílnost rozestupů v čase při generování další kapky.

4.2.3 Projekty studentů

Práce se studenty naznačuje, že vymyslet opravdu hezký projekt nekonečné hry je náročné a ještě obtížnější je přivést studenty k jeho realizaci. Studentské práce obvykle kopírovaly projekt louky s drobnými obměnami – padající sníh, létající mouchy apod. Originálních námětů se vyskytlo pouze několik:

- Dva vojáci se pohybují náhodně v dolní části obrazovky a náhodně střílí na rotu vojáků sestupující shora dolů. Hra počítá pro oba vojáky body podle počtu sestřelených vojáků.
- Na začátku se vygenerují kuličky tří barevných odstínů, které začnou stoupat nahoru a rozdělí se podle barev do tří vodorovných sekcí. V těchto sekcích se dále pohybují nahoru a dolů tak, že svoji sekci neopustí.

Tvorbě nekonečných her předcházela výuka základů objektově orientovaného programování podle konceptu naznačeného v kapitole 3. Zdá se, že právě množství informací před vlastní tvorbou hry může být důvodem, pro který studenti nebyli schopni rozvinout svoji tvořivost a zejména prokázat schopnost hru vytvořit. Grafické objekty jsou nutné k tvorbě her, přesto se zde ukázalo velice užitečné začít s tvořením co nejdříve.

4.3 Tlačítkové hry

Další velkou skupinu her tvoří hry „tlačítkové“ ovládané pomocí myši. Zde jednotlivé grafické objekty mají funkci tlačítek, které vykonávají svoji funkci po kliknutí na ně nebo které se po kliknutí na dané místo grafického plátna zobrazí. Mezi tyto hry patří piškvorky, hra patnáct nebo pexeso. Při tvorbě her využijeme definované grafické objekty.

4.3.1 Ovládání hry pomocí myši

Hra 2 – Piškvorky:

V síti 10x10 čtverečků klikají střídavě dva hráči – jeden křížky, druhý kolečka. Vyhrává ten, který nejdříve sestaví řadu (vodorovnou, svislou nebo po uhlopříčce) 5 objektů.

Při tvorbě této hry se naučíme obsluhovat události myši. Metoda reagující na kliknutí musí vypočíst souřadnice středu čtverečku sítě, kde bylo kliknuto. Tyto souřadnice se stávají středem vytvářeného objektu (kolečka, křížku). Piškvorky jsou hra, při které po kliknutí vznikají na grafickém plátně nové objekty.

Poměrně jednoduchá myšlenka konstrukce hry vyžaduje větší programátorskou zručnost, pokud chceme automaticky kontrolovat výherce. Zde je nutné při každém tahu hráče (při kliknutí) zkontrolovat pro jednotlivé body sítě počet objektů v jednotlivých řadách, a pokud je tento počet 5, pak vyhrává hráč, který byl poslední na tahu.

Hra 3 – Hra patnáct:

V síti 4x4 je rozmístěno 15 hracích kamenů s čísly 1 až 15, jedno místo je volné. Úkolem hráče je s využitím volného místa přesouvat kameny tak, aby jejich číselné hodnoty byly seřazeny – viz obrázek 5.

Obrázek 5. Hra patnáct

Obsluha události při kliknutí myši spočívá v přesunu čtverce, na který bylo kliknuto, na volné místo, je-li toto místo v sousedství čtverce. Programová realizace tohoto pravidla je obtížná s ohledem na datovou reprezentaci hry, s ohledem na ukládání informací o pozicích jednotlivých hracích kamenů.

Hra patnáct nemá řešení pro libovolné počáteční rozmístění kamenů. K automatickému zamíchání kamenů lze využít simulaci mnohanásobného klikání na hrací kameny, kterou již máme vytvořenou.

4.3.2 Automatická změna velikosti

Jednoduché tlačítkové hry využívají jednoduché grafické objekty, jejichž velikost lze snadno upravit. Tyto hry lze využít k demonstraci obsluhy události při změně velikosti okna, která spočívá v poměrovém přepočtu velikostí všech zobrazovaných objektů.

4.3.3 Projekty studentů

Tlačítkové hry vytvářela pouze část studentů. Přesto vznikly mnohdy zajímavé projekty:

- Devět černých kamenů je potřeba proměnit na bílé kameny. Kliknutí na kámen způsobí prohození barev (bílé a černé) daného kamene i okolních kamenů ve čtyřech směrech.
- Postřeh – v poli 5x5 kruhů se v pravidelných intervalech zabarvuje jeden kruh červeně. Klikáním na kruhy je potřeba „chytnout“ červený kruh.
- Stojíme před třemi dveřmi, za jedněmi z nich stojí duch. Smyslem hry je uhodnout, za kterými dveřmi duch nestojí.
- Na ploše 10x10 hrají střídavě dva hráči – žlutý a fialový a kliknutím obarvují jednotlivá políčka. Pokud jeden z nich klikne na políčko tak, že „přeskočí“ políčko soupeře, toto políčko soupeře získává – políčko se obarví jeho barvou. Smyslem je získat co nejvíc políček své barvy.

4.4 Grafické hry

Posledním typem her jsou hry, které se odehrávají v prostředí reprezentovaném obrázkem na pozadí – na louce, v lese, ve vesmíru apod. Zde se nachází různé typy objektů, které mohou být i animované. Některé z objektů stojí na místě, jiné se mohou pohybovat. Hra je ovládána pomocí myší nebo klávesnice. Jedná se o hry, kde je úkolem hráče posbírat nějaké předměty, postřílet padající objekty nebo naopak se padajícím objektům vyhnout.

4.4.1 Ovládání hry pomocí klávesnice

Hra 4 – Vesmír:

Ve vesmíru plném hvězd se pohybuje ufonek ovládán pomocí kurzorových kláves. Jeho úkolem je posbírat co nejvíc hvězdiček. V pohybu mu brání tři mimozemšťané, při setkání s kterými hra předčasně končí – viz obrázek 6.

Obrázek 6. Hra Vesmír

Konstrukci hry začneme vytvořením prostředí – pozadí a hvězd. Přidáme ufonka, který je ovládán událostmi z klávesnice. Ufonek je animovaný, co vyžaduje využití časovače. Při pohybu ufonka je nutné kontrolovat jeho polohu a sledovat, zda se nenachází v blízkosti hvězd. O kontrolu se stará speciální metoda v pravidelných intervalech, tj. metoda, která využívá časovač.

Další vývoj a pravidla hry stanovujme společně se studenty. Lze přidat nepřátelské mimozemšťany pronásledující ufonka nebo automatického soupeře, který se pohybuje podle jistých pravidel a soutěží, kdo nasbírá více hvězd.

4.4.2 Projekty studentů

Grafické hry zajímají studenty nejvíce. Vyžadují sice přípravu grafických objektů, efekt z vytvořené hry je ale silným motivačním prvkem. Studenti jsou navíc v této fázi výuky již

schopni řešit vybrané úkoly samostatně, a proto se nebojí vymýšlet rozmanité varianty her. Přesto dobrá hra vzniká až kombinací dobré grafiky a jistých programátorských dovedností. Se studentských projektů lze vybrat:

- Hru na Kašpárka – Kašpárek ovládán z klávesnice střílí balonky, které k němu létají seshora. Jeho úkolem je nedovolit žádnému balónku spadnout na zem.
- Vesmírná loď ovládána kurzorovými klávesami se snaží vyhnout padajícím meteorům.
- Na ploše se nachází 10 kuliček sedmi různých barev, které se náhodně pohybují. Jedna velké kulička ovládána pomocí klávesnice mění každých 5 vteřin barvu. Jejím úkolem je pochytat všechny malé kuličky s tím, že může chytnout (zničit) pouze kuličku, která je stejné barvy. Pokud se pokusí zlikvidovat kuličku jiné barvy, na ploše přibyde nová malá kulička. Hra končí, když jsou malé kuličky zlikvidovány.

5. Závěr

Výuka základů objektově orientovaného programování je zejména u studentů, pro něž programování není prioritou, obtížná. Již řadu let se nám osvědčuje využít pro tuto výuku hry, a to zejména grafické hry (viz kapitola 4.4). Letos poprvé jsme pro výuku využili programovací jazyk Python, i s ohledem na jeho stále širší využití na školách [12].

Zkušenost ukázala, že změna programovacího jazyka (předtím probíhala výuka v Object Pascalu a později v Javě) nehraje velkou roli. Volba programovacího jazyka omezuje pouze základní terminologii a drobné rozdíly dané samotným programovacím jazykem. Přesto výuka nezaujme studenty každoročně stejnou měrou. Důvodů může být několik.

V první řadě, jak již bylo naznačeno, je vhodné maximálně zkrátit úvodní výuku teoretické části. Teorii je vhodné střídat s praxí. Teoretické znalosti tím ale ztrácí u studentů svoji důležitost a studenti nabývají falešný pocit, že teorii rozumí a orientují se v ní. Hledání optimálního vyvážení teorie a praktických činností je základem učitelského umění v tomto konceptu výuky.

Předpokladem dobré hry je originální a uskutečnitelný nápad, který je výsledkem tvořivé schopnosti studentů. Ta ale mnohdy studentům chybí a originální díla a myšlenky vznikají pouze v omezeném množství. Předmět v letošním roce absolvovalo 16 studentů, přitom nápad pro originální hru mají pouze 3-4 z nich – viz uvedené projekty studentů. Většina studentů naopak pouze kopíruje a mírně přetváří vzorová řešení her vytvářena v rámci prezenční výuky.

S výukou objektově orientovaného programování tematicky zaměřenou na tvorbu her počítáme i do budoucna. Tato oblast se osvědčuje zejména možností tematicky provázat celou výuku, budovat postupně jednotlivé komponenty a ty následně spojovat do vytvářených her. Výhodou je také motivační charakter her. Do budoucna bychom rádi podpořili také kreativní nápady studentů v oblasti her a zaměřili se na podporu výuky směrem k realizaci maximálního množství dobrých nápadů.

BIBLIOGRAFICKÉ ODKAZY

[1] JEŘÁBEK, J. A KOL. *Rámcový vzdělávací program pro základní vzdělávání.* Praha: VÚP, 2007. 124 s. [cit. 2016-1-25]. http://www.vuppraha.cz/wp-content/uploads/2009/12/RVPZV_2007-07.pdf

[2] NAGYOVÁ, I. Fundamental ideas of informatics and their relationship to didactics of informatics. In *Journal of Technology and Information Education*. 2013, vol. 5, no. 1, p. 89-95.

[3] *Strategie digitálního vzdělávání do roku 2020.* Praha, 2014. [cit. 2017-1-16] http://www.msmt.cz/file/34429/

[4] PECINOVSKÝ, R. *Návrhové vzory.* Brno: CPress, 2010. ISBN 978-80-251-1582-4.

[5] KAPOUNOVÁ, J., PAVLÍČEK, J. *Počítače ve výuce a učení.* Ostrava, PdF OU, 2002. 118 s. ISBN 80-7042-265-3.

[6] HEJNÝ a kol. *Matematika B. Učebnice pro 2. stupeň ZŠ a víceletá gymnázia.* Praha: H-mat, 2015. 80 s. ISBN 978-80-905756-1-5.

[7] SCHUBERT, S., SCHWILL, A. *Didaktik der Informatik.* Heidelberg: Spektrum, 2011. ISBN 978-3-8274-2652-9.

[8] NAGYOVÁ, I. *Podstatné aspekty informatiky jako vědního oboru.* Ostrava: OU, 2013. ISBN 978-80-7464-378-1.

[9] VAŠEK, M. *Aplikace pro výuku základních principů objektově orientovaného programování*: diplomová práce. Ostrava: OU, 2016.

[10] MAJHEROVÁ J., PETRUŠKOVÁ, H., KRÁLÍK V., LAJČIAK P. Príprava budúcich učiteľov informatiky a nové trendy vo vyučovaní programovania. In: *DidInfo 2014.* Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied, 2014. s. 97-102.

[11] ŠNAJDER, Ľ. Metodika výučby programovania mobilných zariadení. In: *DidInfo 2015.* Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied, 2015. s. 18-27.

[12] MÉSZÁROSOVÁ, E., TOMCSÁNYIOVÁ, M. Vyučovanie základov programovania v jazyku Python na niektorých SŠ a VŠ. In: *DidInfo 2015.* Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied, 2015. s. 108-111.

Výuka informatiky pomocí videonávodů
ICT lessons supported by videotutorials

Ing. Bc. Václav Nádvorník, Ph.D.
ZŠ Londýnská
Londýnská 34
120 00, Praha 2
Česká republika
nadvornik@londynska.cz

ABSTRACT
This paper does not aim to be a scientific article, but primarily an experience of a science teacher, using video tutorials that lead his classes in elementary school. It is a motivational article where some procedures are applicable across school subjects. The examples shown the use of educational video tutorials to teach 3D modelling program Google SketchUp. There are the possible working procedures, benefits, risks, and approaches feedback of different approaches and feedback of students that passed the teaching.

Keywords
Videotutorials. Flipped classroom. Primmary school. ICT.

ABSTRAKT
Tento příspěvek si nedává za cíl být vědeckým článkem, ale primárně zkušeností učitele informatiky, který pomocí videonávodů vede svoji výuku na základní škole. Jedná se o motivační článek, kde některé postupy jsou použitelná napříč vyučovacími předměty. Na příkladech je uvedeno využití výukových videonávodů při výuce 3D modelování v programu Google SketchUp. Součástí článku jsou možné postupy práce, výhody, rizika možných přístupů a také zpětné vazby žáků, kteří výukou prošli.

Klíčová slova
Videonávody. Převrácená třída. Základní škola. Informatika.

1. ÚVOD
Výuka pomocí videonávodů je v době, kdy žáci mají jednoduchý a rychlý přístup k internetu velmi efektivní metodou. Z velké části může nahradit učitelův výklad, i když práce učitele je také velmi důležitá a v globální rovině nezastupitelná. Videonávody se dají použít jak při výuce ve škole, tak i s velikou efektivitou při domácí přípravě a samostudiu. Autor se tímto tématem zabývá dlouhodobě a jeho kanál patří mezi nejsledovanějšími z kanálů, které obsahují české výukové videonávody v oblasti matematiky, fyziky a informatiky. Tento článek vychází i z dřívějších zkušeností prezentovanými na konferenci Didinfo2015 v příspěvku Videonávody na školní webové stránce[6].

2. CO JE VÝUKOVÝ VIDEONÁVOD
Výukový videonávod je video soubor, který kombinuje jak obrazovou část, tak mluvené slovo a je primárně tvořen jako učební materiál. Samotný videonávod není jediným zdrojem informací a měl by být doprovázen další aktivitou žáka a učitele. Z hlediska toho jak video vypadá jsou v České republice používány následující varianty.

Khanova škola – překládaná videa původní khanacademy.org [4], která jsou v původním znění doplněna českými titulky. Videa jsou tvořena pomocí psaní na počítačovém tabletu a většinou není vidět autor. Typická je černá barva pozadí. Vše je doplněno obrázky.

Výuková videa před tabulí – předkládaná videa jsou nejčastěji natáčena na běžnou videokameru s tím, že autor před tabulí vykládá látku. Jedním z příkladů tohoto druhu je Marek Valášek [5] a jeho projekt Matematikator. Autorů využívající tuto metodu je více.

Výuková videa, která jsou tvořená komentovaným záznamem z obrazovky počítače. Nejčastější forma videí, která vysvětluje použití počítačového software. Často se jedná o ukázku konkrétního postupu využití konkrétního programu. Slovní komentář v některých případech je součástí videa, jindy ne. Tento druh videí je často používán samotnými tvůrci programu a nahrazuje tištěné manuály.

3. VYUŽITÍ VIDEONÁVODŮ VE VÝUCE
Při výuce se principiálně dají využít buď videonávody vytvořené přímo v České republice a které jsou namluvené česky, nebo česky titulkované cizojazyčné návody, případně návody v původním znění. Mnohdy jsou právě návody pro oblast ICT v původním znění ty nejkvalitnější, ale jazyková bariéra může být na škodu. Na druhou stranu se sledováním těchto videonávodů zlepšuje právě jazyková dovednost žáků. Je tedy na volbě každého učitele, jaký jazyk zvolí.

Autoři jsou také velmi rozdílní, od profesionálních návodů tvůrců aplikací, přes dospělé nadšence, až k teenagerům a mnohdy i dětem, kteří začínají svoji činnost youtubera. Je třeba s rozmyslem volit vhodný komunikační jazyk a není třeba se právě dětských a teenagerovských návodů obávat. Je ale nutné je vždy celé předem zhlédnout.

3.1 Základní principy práce
Při vyučování se dá videonávod použít několika různými způsoby. Nejklasičtější je **princip převrácené třídy** (flipped classroom). Zuzana Kadlecová [1] princip převrácené třídy vysvětluje následujícím způsobem: „Edukační proces zde probíhá tak, že se žáci nejprve seznámí s probíranou látkou doma online, a do školy přijdou již s konkrétními dotazy. Ve výuce učitel s dětmi pracuje mnohem více individuálně. Nabízí se tak mnohem efektivnější využití času, protože je věnován pouze těm výukovým aktivitám, které jsou zacíleny na problémy, kterým děti nerozuměly. Každý žák získává možnost postupovat v látce dle vlastního tempa." Je třeba si uvědomit, že se nejedná o žádnou převratnou novinku. Styl výuky, kdy si žáci předem doma něco nastudují a poté ve škole řeší je starou didaktickou metodou. Odlišné je, že na rozdíl

od psaného textu může být videonávod mnohem názornější a tudíž pochopitelnější. Autor tento způsob výuky opakovaně vyzkoušel v různých předmětech a to jak za využití originální a dabovaných videí Khanovy školy, tak za využití jiných videonávodů, včetně návodů vlastních. Při tom narazil na zásadní problém a to ten, že aby princip převrácení třídy fungoval musí zhlédnout videonávod předem všichni žáci. Jestliže se tak nestane, pak následující část lekce, která přímo vychází z předpokladu, že žáci se svoji část nastudovali je neefektivní. Princip převrácené třídy je tedy možné použít pouze u velmi motivovaných žáků. Dříve se jako jeden z argumentů proti této metodě používalo malá možnost přístupu k internetu v domácnostech. Dnes již více jak 93% domácností s dětmi má k internetu přístup a řešit se musí ne jestli přístup je, ale jakými technologiemi, zda v počítači, nebo v mobilním zařízení. Rozdíl by mohl být v možnosti přehrát určité videoformáty. Pro platformu youtube.com však není třeba toto řešit.

Druhým principem je využití videonávodu pro opakování výkladu z hodiny a procvičování. Vytvořený videonávod může být stejný, jako v případě převrácené třídy, ale jeho zhlédnutí již není pro další vzdělávací proces nezbytné. Tento princip je založen na maximální individualizaci, kdy každý žák si samostatně volí tempo a hlavně počet přehrání videa. Je možné zadat úkol, kterým si učitel ověří, zda žák video sledoval, či nikoli. Autor příspěvku využívá při výuce matematiky právě tento postup s tím, že většinu videonávodů si sám tvoří. Na toto téma již byl na konferenci DidInfo2015 přednesen příspěvek[6].

Třetí princip je přímé využití videonávodu ve výuce. Zde je vlastně učitel nahrazen videem, které za něj provádí veškerý výklad. Učitel poté může sám zvolit svoji roli. Tento princip autor využívá při výuce informatiky, zvláště při výuce aplikací. Není vhodné však pustit několikaminutový návod bez jakýchkoli komentářů. V tom případě se stává, že pozornost žáků opadá. Autor se setkal i s tím, že odběratelé jeho kanálu[7] na serveru Youtube jej informují o tom, že autorovi videonávody jsou používány na školách místo výkladu učitele bez jeho jakéhokoli zásahu. Tento přístup může být efektivní při řešení zastupování za nepřítomného učitele tzv. suplování.

4. Výuka Google SketchUp pomocí videonávodů

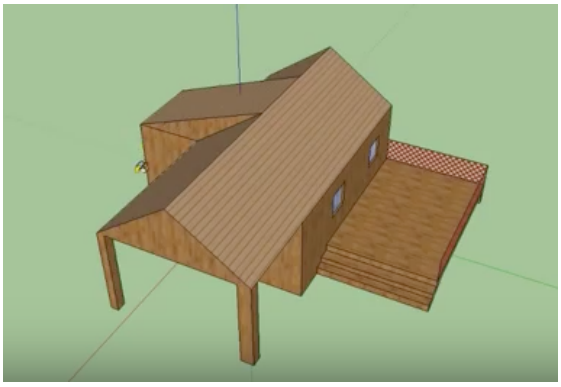
Jednou z aplikací, kde je využití videonávodů přímo ve výuce efektivní je 3D modelovací software Google SketchUp. Tento software je volně stažitelný z internetu (https://www.sketchup.com) a slouží pro jednoduché 3D modelování. Nejedná se o pokročilý profesionální CAD systém, jako je SolidEdge, nebo AutoCad, ale o snadno ovladatelný program.

Pro žáky je lehce pochopitelný a již během prvních hodin jsou v něm schopni modelovat první objekty. Výuka začíná na hranatých tvarech, ideálně domech s tím, že postupně se přechází na detailnější modely a na rotační tělesa. Server youtube.com při zadání hesla "sketchup tutorial" uvede více jak 90 000 odkazů. Vybrat tedy nejefektivnější variantu může být trochu problém.

Mírným úskalím videonávodů jsou rozdílné verze programu a to jak jeho jazykové mutace, tak verze odlišné svojí funkčností. Při výuce je pak třeba upozorňovat na některé odlišnosti.

4.1 Příklady videonávodů na téma dům

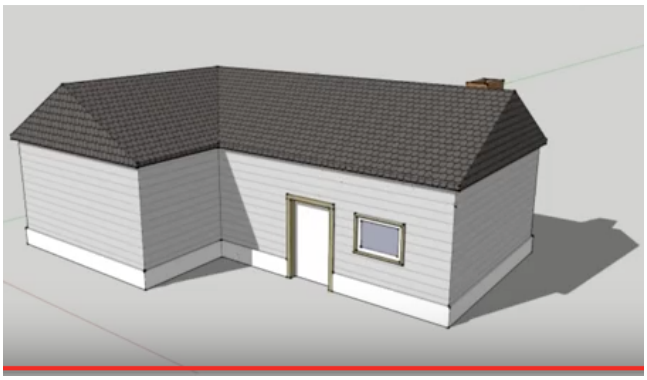
Dětský videonávod youtubera Tom Vide [3].



Obrázek 1 – Ukázka domu z videonávodu Tom Vide [3]

Jedná se o návod podle hlasu žáka základní školy, kde jsou ukázány jednoduché postupy. Přestože se při mluvení někdy autor zasekává, jedná se o povedené a hlavně použitelné dílo s více jak 8000 zhlédnutími.

Druhým příkladem je profesionálnější pojetí anglicky mluvícího youtubera Aarona Bishopa [2].



Obrázek 2 - Ukázka domu z videonávodu Aarona Biskopa [2]

Tento příklad autor využil při letošních hodinách výuky programu a byl i přes anglický jazyk velmi dobře pochopitelný pro žáky 7. a 8. ročníku základní školy.

4.2 Organizace vyučovací jednotky

Uvádím zde konkrétní příklad, jak bylo video využito s časovým snímkem vyučovací jednotky. Žáci pracovali na platformě Apple s verzí Google SketchUp 8.0. Ukázka je však použitelná v různých platformách i verzích programu. Vyučovací jednotka trvala 90 minut.

Předchozí zkušenosti s programem

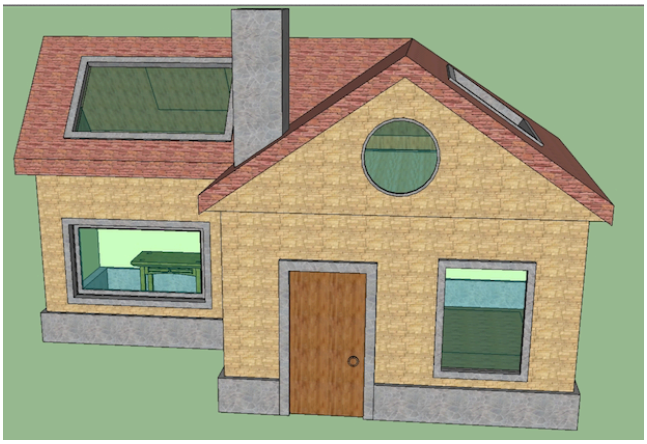
Jednalo se o druhou vyučovací jednotku práce s programem, kdy byli žáci volitelného předmětu aplikace informačních technologií v první jednotce seznámeni se základy ovládání programu a byla jim dána možnost samostatně experimentovat a objevovat jeho vlastnosti.

Vlastní hodina

V rámci motivační části (5 minut) učitel ukázal žákům již hotový vytvořený domek s informací, že na konci hodiny bude tento domek schopen vytvořit každý z nich. Samotný videonávod (15 minut) byl puštěn s přerušováním pro výklad a překlad všem žákům frontálně (na plátně) s tím, že bylo vysvětlováno co a jak bylo vytvořeno. Puštěny byly 2/3 návodu s tím, že je zde uveden

absolutní základ a vytvořena celá stavba. Žáci si poté spustili na polovině obrazovky návod ve webovém prohlížeči a na druhé polovině samotný program. Následovala samostatná práce žáků (60 minut), kdy každý z nich tvořil svůj dům vlastním tempem s ohledem na jazykové a počítačové dovednosti. Většina žáků zvládla vytvořit celý dům včetně nepovinných textur (v návodu v poslední jedné třetině). Posledních 10 minut bylo využito pro reflexi hodiny.

Příklady vytvořeného domu po konci druhé hodiny vidíte na následujícím obrázku.



Zpětná vazba žáků, kteří videonávody využívali

Jednotlivé výroky jsou autentické, bez jakýchkoli úprav.

- „Videonávody jsou super, mohu dělat jak rychle chci."
- „Videonávody někdy jsou v jiné verzi programu, což mi vadí."
- „Videonávody mne baví, zkusil jsem i jeden udělat sám."
- „Bez videonávodu bych nevěděl, jak vytvořit kouli."
- „Videonávody mi pomáhají."
- „Mám rád Vaše videonávody na matiku, dívám se na ně často."
- „Když jsem zkoušel dělat návod sám strávil jsme na tom 9 hodin práce, ale pamatuji si to."
- „Nevěřila jsem, že ten domek zvládnu."

5. Závěr

Videonávody mohou být velmi efektivní metodou práce nejen v hodinách ICT, ale i v jiných předmětech. Stejný princip byl použit při výuce práce s fotografií v programu Adobe Photoshop Elements. Samotným využitím návodů například v matematice se autor zabýval ve svém příspěvku na této konferenci v roce 2015[6].

BIBLIOGRAFICKÉ ODKAZY

[1] KADLECOVÁ, Zuzana. Khan Academy a „převrácená" třída. *Metodický portál: Články* [online]. 02. 02. 2012, [cit. 2017-01-29]. Dostupný z WWW: <http://spomocnik.rvp.cz/clanek/15039/KHAN-ACADEMY-A-„PREVRACENA"-TRIDA.html>. ISSN 1802-4785.

[2] BISHOP, Aaron. *SketchUp 8 Lessons: Making a Simple House* [online]. https://www.youtube.com/watch?v=z2f3DlI31-M&t=179s

[3] VIDE, Tom. *SketchUp - jednoduchý dům* [online]. https://www.youtube.com/watch?v=7jM3nv5Q2uI&t=319s

[4] Khan Academy. http://www.khanacademy.org

[5] Valášek, Marek, *Mathematicator 2017* [online]. http://www.matematikator.org

[6] Nádvorník, Václav. 2015. Videonávody na školní webové stránce. In: HORVÁTHOVÁ, Dana, Ivan BRODENEC a Jana JACKOVÁ. In: *DidInfo 2015. Zborník z 21. ročníka konferencie* [CD] Bánská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied v Banskej Bystrici, ISBN 978-80-557-0852-2.

[7] Nádvorník, Václav. *Adlif57 - kanál youtube* [online]. https://www.youtube.com/user/adlif57

DIDINFO & DIDACTIG 2017. **Univerzita Mateja Bela, Banská Bystrica 2017.**

Didaktika vo výučbe programovania
The didactics in a teaching programming

Peter Potocký
Vydavateľstvo Andrea
Kafendova 2
831 06 Bratislava
Slovensko
peter1potocky@gmail.com

ABSTRACT
Increased demands for teaching information technologies depend on requirements of labor market. Schools focused on education of information technologies, their founders and this conference too, should react to this situation. Thanks to these reactions, the Ministry of Education, Science, Research and Sport of the Slovak Republic has important sources to prepare documents to increase the level of education in elementary, high schools and Universities. It is the reason to support this conference. We will focus on the didactics of teaching programming.

Keywords
Teaching programming. Digital literacy. Specific field didactics of informatics.

ABSTRAKT
Zvýšené nároky na výučbu informatiky vychádzajú z požiadaviek trhu práce. Školy so zameraním na výučbu informatiky, ich zriaďovatelia i táto konferencia by mali každoročne na túto situáciu pružne reagovať, čo je ich poslaním. Ministerstvo školstva, vedy, výskumu a športu Slovenskej republiky (ďalej iba Ministerstvo) potom z týchto reakcií získava dôležité podklady pre prípravu dokumentov k potrebnému zvyšovaniu úrovne vzdelávania na takto zameraných základných, stredných a vysokých školách. To je dôvodom jeho záštity nad touto konferenciou. Zameriame sa na didaktiku komplexnej výučby programovania.

Kľúčové slová
Komplexná výučba programovania. Digitálna gramotnosť programátora. Odborová didaktika informatiky.

1. ÚVOD
Komplexná výučba programovania si vyžaduje didaktické zahrnutie celej škály tejto problematiky. Dôležitosť didaktického zahrnutia je najmä v rozvíjaní poznávacích daností žiaka, jeho logického myslenia, ktoré si vyžaduje profesia vývojového zamestnanca v oblasti programových aplikácií. Preto venujeme pozornosť tomuto konceptu výučby programovania.

Pri výučbe vychádzame z klasifikácie programovacích jazykov podľa miery ich abstrakcie od hardvéru. V samotnom priebehu výučby programovania sa sleduje dodržiavania základných didaktických zásad. Za tým účelom je nevyhnutné zoradiť témy danej výučby v súlade s týmito zásadami.

Oboznámime sa s konceptom komplexnej výučby programovania, ktorý si zdôvodníme didakticky, čiže v zmysle teórie vyučovania.

2. KONCEPT KOMPLEXNEJ VÝUČBY PROGRAMOVANIA
Koncept komplexnej výučby programovania sa viaže na každé technické zariadenie, ktoré je riadené centrálnou procesorovou jednotkou (ďalej iba CPU). Vo všeobecnosti sa jedná o osobné počítače a mikropočítače. Táto výučba predstavuje:

- Vývoj softvérových produktov **pre osobné počítače**. Dané produkty sú využívané v rôznych odvetviach výrobnej i nevýrobnej sfére prostredníctvom osobných počítačov.
- Vývoj softvérových produktov **pre mikropočítače**. Produkty sú využívané v každom väčšom meste pri svetelnom riadení dopravy na križovatkách, ale i v podobe rôznych svetelných reklám. Ďalej sú to robotické zariadenia, ktoré vo veľkom počte využíva aj automobilový priemysel. V domácnostiach ide predovšetkým o diaľkové ovládače televízorov, videí alebo prehrávačov, nastavenie programu pračiek, mikrovlniek a podobne.

V súčasnosti má prax našej krajiny zastúpenie medzi oboma oblasťami v pomere zhruba 50 : 50 a tak by tomu malo byť aj vo výučbe programovania na stredných školách so zameraním na elektrotechniku, informačné a komunikačné technológie a im podobné oblasti. S touto skutočnosťou som informoval 22. ročník národnej konferencie o výučbe informatiky DidInfo 2016, ďalej národne orientovanú čitateľskú verejnosť [1, s. 10] a rovnako aj Štátny inštitút odborného vzdelávania. Urobil som to z dôvodu nedostatočne plnenej požiadavky trhu práce strednými odbornými školami s uvedeným odborným zameraním Toto nedostatočné plnenie bolo spôsobené výpadkom výučby mikropočítačov práve na týchto stredných školách.

2.1 Popis konceptu
Koncept komplexnej výučby programovania vychádza z členenia programovacích jazykov podľa ich miery abstrakcie od hardvéru, či inak povedané, podľa spätosti toho-ktorého programovacieho jazyka s CPU.

Pre programovacie jazyky má uvedená miera abstrakcie nasledovné tri úrovne:

- **Nižšia úroveň**. Tieto jazyky sa vyznačujú najväčšou spätosťou s CPU, nakoľko využívajú inštrukcie tejto procesorovej jednotky, ktoré sú uložené v registri inštrukcií. Navrhli sme teoretickú i praktickú výučbu základov strojovo orientovaného programovacieho jazyka symbolických inštrukcií (ďalej iba JSI). V nadväznosti na ďalšiu úroveň týchto jazykov, podľa didaktickej zásady sústavnosti a postupnosti [2, s. 89 - 92], sme

taktiež navrhli realizovať ich praktickú výučbu štruktúrovaným riešením jednoduchých úloh na osobnom počítači.

- **Stredná úroveň**. Jedná sa o jazyky so strednou spätosťou s CPU, ktoré nevyužívajú inštrukcie CPU a preto ani nepoznajú vstupné a výstupné operácie. Navrhli sme výučbu základov štruktúrovaného programovacieho jazyka C, keďže je najviac blízky JSI. Základy zahŕňajú i využívanie operačného systému týmto jazykom pri realizovaní vstupných a výstupných operácií. Výučbu príkazov a funkcií tohto jazyka treba realizovať riešením jednoduchých úloh na počítači.
- **Vyššia úroveň**. Zahŕňa programovacie jazyky s najmenšou spätosťou s CPU, čo je vyššia úroveň miery uvedenej abstrakcie. Podľa didaktickej zásady sústavnosti a postupnosti [2, 89 - 92], sme navrhli výučbu objektovo orientovaného programovacieho jazyka C++, ktorý pracuje s objektovou nadstavbou jazyka C.

Rovnako sme navrhli, aby sa výučba mikropočítačov začala realizovať strojovo orientovaným JSI tak, aby sa pri riešení jednoduchých príkladov popisovala aj činnosť CPU mikropočítača. Zámerom je porozumieť hardvéru mikropočítača, čo plne zodpovedá zásade trvácnosti vedomostí a všestranného rozvoja poznávacích schopností žiakov [2, s. 92 - 93].

Zhruba v týchto intenciách vyzeral v roku 2008 koncept komplexnej výučby programovania, ktorý som si dovolil označiť pojmom **komplexná systémová výučba informatiky**. Uplatnením daného konceptu v pedagogickej praxi sa sledovalo **dosiahnutie digitálnej gramotnosti v oblasti programovania** u absolventa strednej školy so zameraním na elektrotechniku, informačné a komunikačné technológie a im príbuzné odbory. Po zavedení danej výučby programovania na uvedených stredných školách sa má pozornosť zamerať na ovládanie a uplatňovanie informačných a komunikačných kompetencií učiteľmi vo výučbe tohto odborného predmetu. Dôvodom danej požiadavky je fakt, že vo výučbe akéhokoľvek odborného predmetu učiteľovi nestačia len požadované odborné vedomosti, ale aj informačné a komunikačné kompetencie, keďže práve tieto významne ovplyvňujú záujem žiakov o ten-ktorý odborný predmet. Na tejto univerzitnej pôde som to minulý rok demonštroval so svojimi žiakmi.

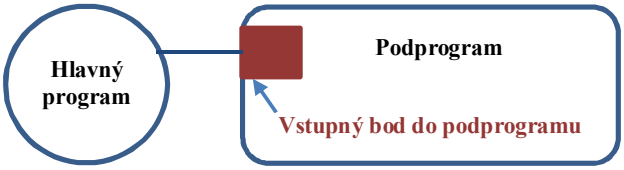
3. DIDAKTIKA KOMPLEXNEJ VÝUČBY PROGRMOVANIA

Výučba programovania na báze tohto konceptu sleduje najmä zásadu trvácnosti vedomostí a všestranného rozvoja poznávacích schopností žiakov[2, s. 92 - 93], čo nie je možné dosiahnuť bez koncentrického zosúladenia jednotlivých tematických celkov. V prioritách tohto zosúladenia sa najskôr zameriame na tvorbu programových aplikácií pre osobné počítače a potom pre mikropočítače.

3.1 Didaktika v tvorbe aplikácií pre osobné počítače

V súlade so zásadou trvácnosti vedomostí a všestranného rozvoja poznávacích schopností žiakov [2, s. 92 - 93] je pri výučbe programovania dôležité zamerať pozornosť na výklad pojmu **vstupný bod** do programu, resp. podprogramu.

Zámerom tohto výkladu je porozumieť princípu práce hlavnej vstupno-výstupnej funkcií v jazyku C, čo predpokladá porozumeniu práce procedúr v JSI. Aby sme mohli naplniť tento zámer výučby, bolo potrebné riešiť tvorbu programových aplikácií JSI štruktúrovane, ako ukazuje schéma na obr. 1:



Obr. 1 **Schéma štruktúrovane riešenej aplikácie v JSI**

V schéme štruktúrovane riešenej aplikácie v JSI predstavuje vstupný bod do podprogramu **procedúra**. Procedúru voláme z hlavného programu inštrukciou procesora CALL.

Podprogram môže obsahovať viacej procedúr, ktoré sa medzi sebou odlišujú svojim menom, čo znamená, že každá procedúra musí mať iné meno. V prípade duplicity mien procedúr prekladač vypíše chybu. Vstup do podprogramu cez inú procedúru je možný len vtedy, ak predchádzajúca procedúra ukončila svoju činnosť, čo znamená, že v podprograme sa môže realizovať iba jedna procedúra. Ukončenie činnosti procedúry sa realizuje návratom z nej do hlavného programu. Návrat z procedúry do hlavného programu sa realizuje inštrukciou procesora RET, ktorú používa procedúra ako posledný príkaz svojho tela.

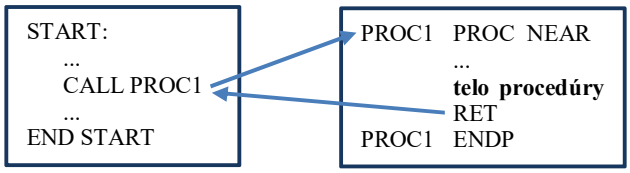
Pred výkladom činnosti párových inštrukcií procesora CALL – RET je potrebné oboznámiť žiakov so schémou tela procedúry podľa obr. 2 [10, s. 34]:

```
PROC1   PROC  NEAR
        ...
        telo procedúry
        ...
PROC1   ENDP
```

Obr. 2 **Schéma tela procedúry s menom PROC1**

Na schéme, ktorú znázorňuje obr. 2, vidíme, že telo procedúry s menom PROC1 sa nachádza medzi začiatkom procedúry a koncom procedúry. Začiatok procedúry označujeme direktívou PROC a koniec procedúry direktívou ENDP.

V inštrukcii procesora CALL uvedieme meno procedúry, ktorá bude v našom prípade predstavovať vstupný bod do podprogramu. „***Inštrukcia procesora CALL*** *zavolá procedúru takým spôsobom, že nariadi procesoru začať činnosť na novom pamäťovom mieste. Procedúra používa* ***inštrukciu procesora RET*** *(návrat z procedúry) takým spôsobom, že nariadi procesoru, aby vrátil svoju činnosť do bodu v programe, z ktorého bol volaný*“[4, s. 165]. Obr. 3 znázorňuje schému činnosti týchto inštrukcií.



Obr. 3 **Schéma činnosti párových inštrukcií CALL – RET**

S funkčnou programovou aplikáciou tohto príkladu som podrobne oboznámil túto národnú konferenciu ešte v roku 2014 [5, s. 126 - 134].

Dodržanie didaktickej zásady sústavnosti a postupnosti [2, s. 89 – 92] si vyžaduje upozorniť žiakov na inštrukcie procesora, ktoré realizujú vstupné a výstupné operácie. Oboznámenie žiakov s daným poznatkom je dôležité z toho dôvodu, že štruktúrovaný

programovací jazyk C „*nedefinuje žiadnu vstupno-výstupnú operáciu ako časť jazyka*“ [6, s. 39] v porovnaní so strojovo orientovaným programovacím JSI.

Príkazy sa v jednotlivých programových aplikáciách realizujú v takom poradí, v akom boli napísané. Pokiaľ ide o rýchlosť ich realizovania, tak najrýchlejšie sa zrealizuje aplikácia, ktorá bola vytvorená v nízkoúrovňovom jazyku a najpomalšie tá, ktorá bola vytvorená vo vysokoúrovňovom jazyku.

Vstupným bodom do programových aplikácií, ktoré boli vytvorené štruktúrovaným jazykom C, je hlavná funkcia main (). Pri štarte týchto aplikácii prostredie operačného systému cez vstupný bod tejto aplikácie otvára nasledovné 3 štandardné súbory a ukazovatele na tieto súbory:

- Štandardný vstup a jeho ukazovateľ **stdin**.
- Štandardný výstup a jeho ukazovateľ **stdout**.
- Štandardný chybový súbor a jeho ukazovateľ **stderr**.

Je tomu tak z toho dôvodu, že nasledovné tri vstupno-výstupné štandardné súbory s ich ukazovateľmi sú súčasťou vytvorenej programovej aplikácie v jazyku C, ako ukazuje schéma na obr. 4:

Obr. 4 **Činnosť operačného systému po spustení vytvorenej programovej aplikácie štruktúrovaných jazykom C**

Podľa didaktickej zásady trvácnosti vedomostí a všestranného rozvoja poznávacích schopností žiakov [2, s. 92 - 93] by mal žiak porozumieť toto: Pri vytvorení programovej aplikácie strojovo orientovaným JSI sa operačným systémom **neotvárajú** žiadne štandardné vstupno-výstupné súbory s ich ukazovateľmi z toho dôvodu, že **JSI definuje vstupno-výstupné operácie ako časť jazyka** prostredníctvom vstupno-výstupných inštrukcií procesora.

Štruktúrovaný programovací jazyk C je už jazykom strednej úrovne abstrakcie od hardvéru PC. Pri spustení vytvorenej programovej aplikácie týmto stredne úrovňovým jazykom sa operačným systémom **otvárajú** štandardné vstupno-výstupné súbory s ich ukazovateľmi z toho dôvodu, že **jazyk C nedefinuje žiadne vstupno-výstupné operácie ako časť jazyka**. Preto sa na realizáciu týchto operácii musia využívať uvedené štandardné súbory s ich ukazovateľmi. Ukazovatele uvedených troch štandardných súborov „*sú deklarované v hlavičkovom súbore* <***stdio.h***>“[7, s. 176].

Poukázali sme na uplatnenie vybraných základných didaktických zásad v komplexnej výučbe programovania. V tejto súvislosti považujeme za dôležité poukázať aj na principiálne zhody vytváraných štruktúrovaných aplikácií JSI a jazykom C. Tá je predovšetkým v činnosti tela procedúry vytvoreného podprogramu JSI, ktorá je v princípe porovnateľná s činnosťou tela hlavnej funkcii main() vytvoreného programu jazykom C. V nízkoúrovňovom JSI procedúra podprogramu odovzdáva návratovú hodnotu hlavnému programu, no v stredne úrovňovom jazyku C hlavná funkcia main() odovzdáva návratovú hodnotu operačnému systému. Avšak v odovzdávaní návratových hodnôt medzi hlavnou funkciou main() a vlastnými funkciami v jazyku C je zhoda s vytvorenou štruktúrovanou aplikáciou JSI.

Samozrejme, problematika výučby vytvárania programových aplikácií pre osobné počítače je omnoho rozsiahlejšia. Predložili sme tejto konferencií didaktické zdôvodnenie tejto výučby z toho dôvodu, aby sme poskytli kvalifikovaným učiteľom poklady pre jej realizovanie v odborných predmetoch aplikovaná informatika a programové vybavenia počítačov, ktoré sa vyučujú na stredných školách so zameraním na elektrotechniku, informačné technológie a im podobné oblasti. Obdobné didaktické zdôvodnenie poskytujeme aj pre výučbu tvorby programových aplikácií pre mikropočítače. Tieto činnosti považujeme za dostačujúce **didaktické zdôvodnenie komplexnej výučby programovania** na stredných školách s uvedeným zameraním.

3.2 Didaktika v tvorbe aplikácií pre mikropočítače

Narušenie komplexnej výučby programovania, ktoré nastalo po novembri 1989 spôsobilo, že na fakulty univerzít, ktoré sú zamerané na mikropočítačovú techniku, začali zo stredných škôl prichádzať nedostatočne pripravení žiaci na toto pomerne náročné štúdium. Reakcia týchto univerzít na danú situáciu sa premietla v ich odbornej komunikácií s uvedenými strednými školami. Ako príklad uvediem odbornú prácu pána prof. Ing. Juraja MIČEKA, PhD. z Katedry technickej kybernetiky Fakulty riadenia a informatiky Žilinskej univerzity (ďalej iba KTK FRI ŽU) v Žiline. Svojimi dvomi vysokoškolskými učebnicami z rokov 2004[8] a 2006[9] „*odovzdal zamestnancom tejto katedry potrebné teoretické poznatky na vytvorenie elektronickej stavebnice Yrobot*“[1, s. 10]. Táto stavebnica je riadená mikropočítačom ATmega16A a vďaka finančnej podpore Nadácie Volkswagen Slovakia bola poskytnutá dovedna viac ako dvadsiatim stredným odborným a priemyselným školách s uvedeným zameraním. Zamestnanci tejto katedry taktiež v rokoch 2014, 2015 a 2016 poskytli školenie učiteľom na týchto stredných školách a vybavili ich tromi učebnými pomôckami [3], [10] a [11], ktorých vydanie v prevažnej miere sponzorovala Nadácia Volkswagen Slovakia.

Po skončení tejto aktivity zamestnanci spomínanej katedry začali bezplatne poskytovať talentovaným žiakom stredných škôl školenia z tvorby programových aplikácií pre mikropočítače. V roku 2016 to bola Letná škola programovania pre stredoškolákov, ktorej obsahové zameranie vidíme na obr. 5:

Letná škola programovania

- ✓ 06.07.2016 - 08.07.2016 (začiatočníci),
- ✓ 13.07.2016 - 15.07.2016 (pokročilí).

Kde?
Katedra technickej kybernetiky, Fakulta riadenia a informatiky, Žilinská univerzita v Žiline
Univerzitná 8215/1, 010 26 Žilina

Čo?
Letná škola programovania OpenHW platformy Yrobot.

Ako?
Prihlasovanie je možné do 30.6.2016, zaslaním e-mailu na michal.hodon@fri.uniza.sk
Účať na letnej škole je bezplatná, účastník si hradí ubytovanie a stravu (VŠ internáty + VŠ Menza).

Obr. 5 **Školenie pre nadaných stredoškolákov z roku 2016**

V 2017 to bola téma IoT a Robotika, ako to vidíme na obr. 6:

Obr. 5 **Školenie pre nadaných stredoškolákov z roku 2017**

Uvedené fakty ukazujú na rozšírenú aktivitu vedecko-pedagogických zamestnancov KTK FRI ŽU v Žiline, ktorou je práca s talentovanými žiakmi stredných škôl so zameraním na elektrotechniku, ITC a im podobné odbory. Zisťovanie príčin tohto nového rozšírenia výchovno-vzdelávacej činnosti spomínanej katedry ponechávame na kompetentnejšie osoby. Tu uvádzame tento fakt z dôvodu poukázania na dôležitosť kvalifikovanej výučby mikropočítačov na stredných školách s uvedeným zameraním.

Nikto nemôže spochybniť dôležitosť uplatňovania didaktických zásad vo výučbe akéhokoľvek predmetu. Preto sa na tomto mieste venujeme didaktike vo výučbe tvorby aplikácií pre mikropočítače.

Obdobne, ako tomu bolo pri didaktike vo výučbe tvorby aplikácií pre osobné počítače, i túto výučbu začíname strojovo orientovaným JSI, nakoľko ide o „*dobrý prostriedok na vysvetlenie a pochopenie základných obvodových štruktúr a ich funkcií, ktoré sú integrované v moderných technických prostriedkoch*“[10, s. 15]. Je to v súlade s takým konceptom výučby informatiky (programovania), ktorý sa venuje stále platným princípom a mechanizmom, pretože jedine z nich „*sa bude vychádzať aj o niekoľko rokov, keď sa terajší žiaci dostanú do praxe*“[12, s. 9].

Mechanizmus práce centrálnej procesorovej jednotky (ďalej iba CPU) súčasného mikropočítača funguje na rovnakom princípe ako tomu bolo pri tom, ktorý bol vyrobený ešte v minulom storočí. Pochopenie tohto princípu výrazne uľahčuje vývojárom efektívne vytvárať programové aplikácie v strojovo orientovanom JSI. Preto je potrebné odovzdať žiakom základné poznatky aj z tejto oblasti.

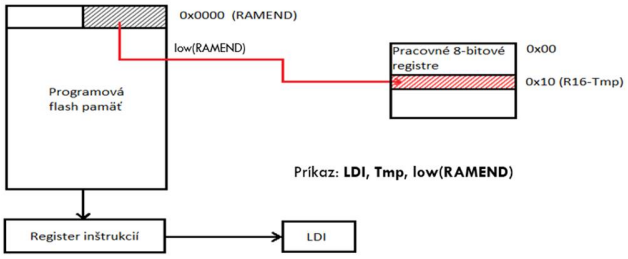
Úloha:

Podľa vytvorenej rutiny RESET zdrojového kódu strojovo orientovaným JSI vysvetlite činnosť registra inštrukcií, 8-bitového pracovného registra pre všeobecné použitie a 16-bitového ukazovateľa zásobníka (SP), jeho dolnej (SPL) a hornej (SPH) časti.

Riešenie:

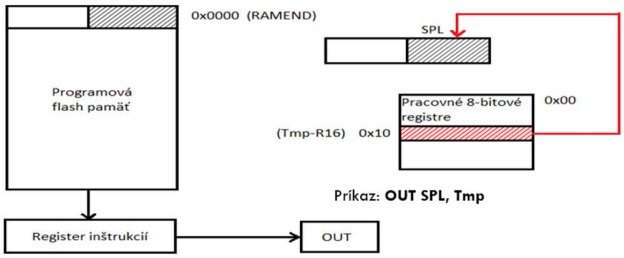
Podľa schém na uvedených obrázkoch popíšeme jednotlivé príkazy napísaného zdrojového kódu strojovo orientovaným JSI v takom poradí, v akom sa realizujú:

Príkaz 1: Presun dolnej časti konštanty RAMEND do pracovného registra pre všeobecné použitie R16 znázorňuje schéma na obr. 6:



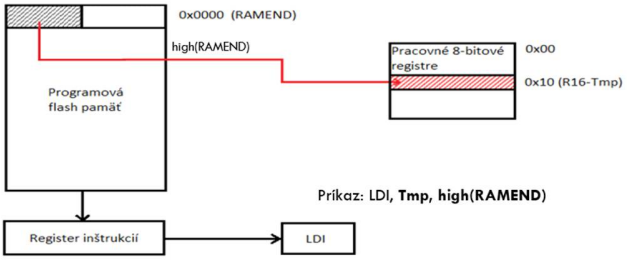
Obr. 6 **Schéma realizácie príkazu LDI, Tmp, low(RAMEND)**

Príkaz 2: Presun dolnej časti konštanty RAMEND z pracovného registra pre všeobecné použitie R16 do dolnej časti ukazovateľa zásobníka SP znázorňuje schéma na obr. 7:



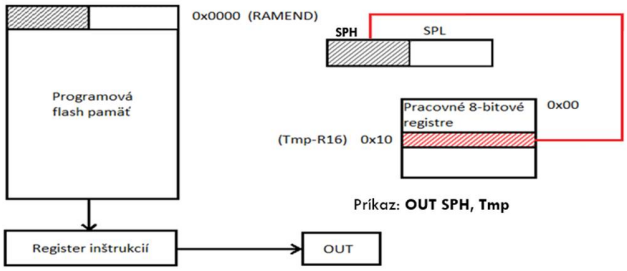
Obr. 7 **Schéma realizácie príkazu OUT SPL, Tmp**

Príkaz 3: Presun hornej časti konštanty RAMEND do pracovného registra pre všeobecné použitie R16 znázorňuje schéma na obr. 8:



Obr. 8 **Schéma realizácie príkazu LDI, Tmp, high(RAMEND)**

Príkaz 4: Presun hornej časti konštanty RAMEND z pracovného registra pre všeobecné použitie R16 do hornej časti ukazovateľa zásobníka SP znázorňuje schéma na obr. 9:



Obr. 9 **Schéma realizácie príkazu OUT SPL, Tmp**

Schémy na obr. 6 – 9 zobrazujú presun 16 bitovej konštanty RAMEND z programovej flash pamäte do ukazovateľa zásobníka. Ich prednosťou je zrozumiteľné znázornenie tohto presunu centrálnou procesorovou jednotkou mikropočítača. Pri tvorbe aplikácií štruktúrovaným jazykom C sa táto činnosť neprogramuje a z tohto dôvodu nie je možné uplatniť tento výklad bez vytvorenia jednoduchej programovej aplikácie strojovo orientovaným JSI. Nakoľko si dané učivo vyžaduje výklad na úrovni 3. stupňa poznávacieho procesu revidovanej Bloomovej taxonómie vzdelávacích cieľov je nutné pracovať s programovacím JSI.

3.3 Zhrnutie

Predkladaný **koncept digitálnej gramotnosti** pre programovanie zahŕňa aj mikropočítače. To si vyžaduje komplexnú výučbu programovania, ktorá obsahuje:

- **Vývoj programových aplikácií pre osobné počítače a mikropočítače**. Začíname strojovo orientovaným JSI. U osobných počítačoch sa zameriavame na štruktúrované programovanie. Pri mikropočítačoch využívanie programovanie v JSI nielen na porozumenie chodu CPU mikropočítača, ale aj na spôsob ovládania jeho vonkajších periférií. Na úrovni strednej školy stačí porozumieť práci údajového registra PORTx a riadiaceho registra smeru DDRx s ich ovládaním, čo je nevyhnutné vedieť pre tvorbu jednoduchých programových aplikácií. Po zvládnutí týchto základných poznatkov z JSI pristúpime k štruktúrovanému programovaciemu jazyku C pre osobné počítače a mikropočítače. Tu končia základy mikropočítača s vývojom jeho základných programových aplikácií.
- **Vývoj programových aplikácií pre osobné počítače objektovo orientovaným jazykom C++**. Téma nadväzuje na získané poznatky zo štruktúrovaného programovacieho jazyka C, ktoré realizujeme na osobných počítačoch.

V systémovej výučbe programovania sú základy programovacích jazykov na všetkých troch úrovniach. Realizácia tohto konceptu výučby na stredných školách však nespadá do našej kompetencie.

POĎAKOVANIE

Ďakujeme Katedre technickej kybernetiky Fakulty riadenia a informatiky Žilinskej univerzity v Žiline za poskytnutie materiálov pre tento príspevok. Osobne ďakujeme zamestnancovi tejto katedry pánovi prof. Ing. Jurajovi MIČEKOVI, PhD. za poskytnuté konzultácie v oblasti didaktiky programovania, ktorá je založená na tvorivom a logickom rozvoji vedomostí žiaka. Nakoľko je faktom, že pri práci vo vývojovom prostredí Arduino sa využívajú už hotové zdrojové kódy z webových stránok, čo nie je tvorivosť, ale opisovanie, plne sa zhodujeme so stanoviskom tohto pána profesora: „***...prečo nie Arduino? Najmä preto, že chceme tvoriť, nie opisovať***“.

BIBLIOGRAFICKÉ ODKAZY

[1] POTOCKÝ, P. *Cesta k prosperite Slovenska. Výučba informatiky akoby podliehala politickým zmenám*. Slovenské národné noviny 20/2006. Týždenník Matice Slovenskej.

[2] ŠLOSÁR, R., NOVÁK, J. *Odborová didaktika*. Vydalo Vydavateľstvo EKONÓM. 2. prepracované a doplnené vydanie. Bratislava 2015. ISBN 978-80-225-4037-7.

[3] POTOCKÝ, P. *Číslicová a mikroprocesorová technika pre stredné priemyselné a odborné školy elektrotechnické*. Vydalo vydavateľstvo Andrea. Bratislava 2015. ISBN 978-80-970853-4-6.

[4] IRVINE K. R.: *Assembly language for X86 processors*. Florida International University School of Computing and Information Sciences. Sixth Edition Copyright © 2011 by Pearson Education, Inc. New Jersey. ISBN-13: 978-0-13-602212-1.

[5] POTOCKÝ P.: *Modernizácia výučby strojovo orientovaných jazykov. In: DidInfo 2014*, 20. ročník národnej konferencie. Vydala Univerzita Mateja Bela, Fakulta prírodných vied. Banská Bystrica 2014. ISBN 978-80-557-0698-6.

[6] HEROUT P.: *Učebnice jazyka C – 1. díl*. České Budějovice. KOPP 2010. ISBN: 978-80-7232-383-8.

[7] KERNIGHAN B. V., RITCHIE D. M.: *Programovací jazyk C*. Vydalo Nakladatelství Computer Press. Dotisk 1. vydání. Brno 2013. ISBN: 978-80-251-0897-0.

[8] MIČEK, J. *Mikrokontroler ATmega8 – popis systému*. Vydala Žilinská univerzita v Žiline. Žilina 2004. http://fribot.sk/yrobot/wp-content/uploads/2015/04/ATmega8.pdf

[9] MIČEK, J. *MCU ATmega8 – príklady a aplikácie*. Vydala Žilinská univerzita v Žiline. Žilina 2006. http://robotika.yweb.sk/skola/AVR/ATmega-aplikacie.pdf

[10] HODOŇ, M. a kolektív *Vývoj mikropočítačových aplikácií od jazyka symbolických inštrukcií po jazyk C*. Vydalo vydavateľstvo Andrea. Bratislava 2015. ISBN 978-80-970853-3-9.

[11] MIČEK, J. a kolektív: *Sprievodca po svete Yrobota*. Vydal EDIS – vydavateľské centrum Žilinskej univerzity. Žilina 2015. ISBN 978-80-554-1120-0.

[12] BLAHO A.: *Informatika v štátnom vzdelávacom programe. In: DidInfo 2012*, 18. ročník národnej konferencie. Vydala Univerzita Mateja Bela, Fakulta prírodných vied. Banská Bystrica 2012. ISBN 978-80-557-0342-8.

Chyby ako súčasť motivácie programovania
Making mistakes as part of the motivation programming

RNDr. Mária Spišáková
KZVI FMFI UK
Mlynská dolina 1
842 48 Bratislava
Slovensko
spisakov@gmail.com

doc. RNDr. Ľubomír Salanci, PhD.
KZVI FMFI UK
Mlynská dolina 1
842 48 Bratislava
Slovensko
salanci@fmph.uniba.sk

ABSTRACT
Teachers often do not realize how pupils think about a solution, how a knowledge is developed in the mind of pupils. When developing solutions of problems, pupils have to make mistakes. Those mistakes develop the brain to its better activity. In this article we show our observations from our classrooms when pupils are discovering individual solutions of coding problems.

Keywords
Solution of the programming problem. Coding. Mistakes. Observation of the classroom.

ABSTRAKT
Učitelia si často neuvedomujú, ako v mysli žiaka vzniká poznatok, ako žiak pri riešení premýšľa, že pri riešení úloh sa musí mýliť, robiť chybné kroky a dávať chybné riešenia. Chyby rozvíjajú mozog, pri robení chýb prebieha väčšia mozgová aktivita. V článku prinášame naše pozorovania z hodín, v ktorých žiaci pracujú samostatne v skupinách, hľadajú riešenie, diskutujú o svojich riešeniach, a tak sú vťahovaní do programovania.

Kľúčové slová
Riešenie úloh. Programovanie. Chyby. Pozorovanie z hodín.

1. ÚVOD
Cieľom nášho výskumu, ktorý robíme v rámci našej vedeckej práce, je motivácia na hodinách informatiky. V predchádzajúcich príspevkoch [1], [2] sme sa venovali meraniam motivácie žiakov k výkonu a pojmu motivácie, ako takej.

V tomto príspevku približujeme spôsoby, ako žiaci pristupujú k riešeniu programátorských problémov. Tentoraz sa však zameriame na chyby a omyly, ktoré nastávajú počas poznávacieho procesu. Objasňujeme aj spôsob, ako k takýmto chybám pristupujeme, aj to, ako chyby vplývajú na ich motiváciu.

Nasledujúce pozorovania sme zaznamenali na Gymnáziu Pavla Horova v Michalovciach v rámci predmetu informatika v prvom ročníku. V tejto triede sa výučba realizuje raz do týždňa počas dvoch vyučovacích hodín.

S triedou programujeme v jazyku Python, na grafiku používame knižnicu *tkinter* (knižnicu *turtle* zatiaľ nepoužívame). Žiaci poznajú premenné, výber náhodných objektov zo zoznamu, generovanie náhodných čísel a cyklus for.

Priebeh vyučovacej hodiny sme nahrali na diktafón, ktorý bol počas celej hodiny zapnutý a zavesený v strede miestnosti mimo centra pozornosti žiakov. Súčasne sme si robili poznámky o postupoch riešení v jednotlivých skupinách. Nahrávky sme následne prepísali. Pre potreby tohto článku uvádzame dialógy medzi učiteľom a žiakmi, vysvetľujúce poznámky, kúsky kódu aj rozhovory medzi žiakmi navzájom.

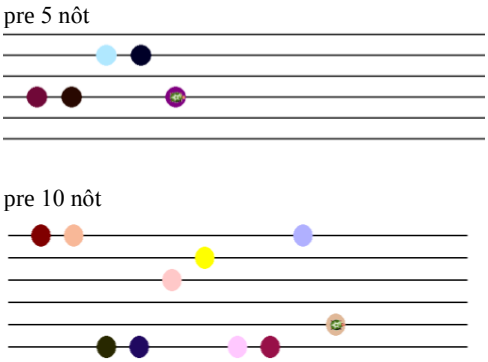
2. POZOROVANIE ŽIAKOV PRI RIEŠENÍ PROBLÉMU
Žiakov sme rozdelili do **piatich** heterogénnych skupín, po 2 – 3 žiakov. Zadali sme im úlohu, v ktorej sa mali generovať noty náhodných farieb na náhodných miestach notovej osnovy. Počet nôt zadával používateľ programu. Skupiny pracovali na úlohe samostatne, bez pomoci vyučujúceho.

Zadanie skupinovej práce znelo: [3]

Trpaslíci potrebujú pre kapelu, ktorá bude hrať na plese, notový zápis.

Vytvorte program, ktorý vykreslí noty podobne ako vidno na obrázku. Notová osnova sa bude skladať zo 6 čiar. Nota bude farebná (náhodnej farby) a bude sa generovať na náhodnej notovej čiare. Všetkých nôt bude n. To bude číslo, ktoré program načíta zo vstupu.

Noty budú náhodne umiestnené na čiarach.



Obr. 1 Ukážky možných výstupov programu

Úlohu sme prebrali zo školskej súťaže v programovaní iGePeHa boj 2015. Notová osnova má 6 čiar pretože autorka nemá hudobné vzdelanie a my sme použili doslovné znenie úlohy. Pozorovali sme žiakov pri práci na zadanom probléme. V skupinách sa vyskytovali rôzne prístupy k riešeniu, rôzne typy problémov, aj sociálnych vzťahov. Všetky skupiny problém vyriešili, ale ich riešenia boli v niečom iné.

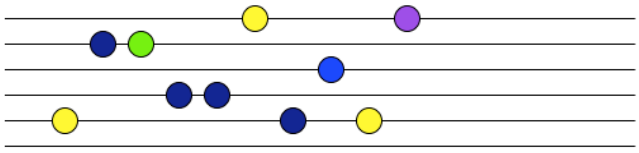
2.1 Skupina Adam, Braňo a Rado
Adam, Braňo a Rado pracovali na probléme nerovnomerne. Adam prevzal riešenie do svojich rúk, Rado bol úplne pasívny, skôr odpisoval, ako tvoril.

Ako postupovali v riešení? Najskôr si nakreslili notové čiary. Potom rozmýšľali o tom, ako generovať noty. Nastavili si najprv pevné súradnice x, y a nakreslili pár nôt s konkrétnymi súradnicami. Adam si však uvedomil, že tie noty nebudú na náhodných miestach, ale ostanú vždy na konkrétnej pozícii. Vedel, že musí použiť cyklus, ale sa nevedel zorientovať, pretože si bol vedomý, že má posúvať súradnicu x aj y a cyklus sa má opakovať podľa počtu nôt.

Rado po upozornení učiteľom, že sa nezúčastňuje riešenia, reagoval slovami: „Adam je hlava, ja sa budem zbytočne namáhať.“ Podľa nás Rado nebol nastavený na riešenie problému. Keď už Adam nevedel, ako postupovať ďalej, začal spolupracovať s Braňom a nie s Radom. Adam s Braňom vymysleli, ako je potrebné generovať výšku noty – súradnicu y. Avšak stále to bolo pri pevnom x = 40. Takže noty boli len na jednej súradnici x. Bolo treba vymyslieť posúvanie súradnice x. A nakoniec riešili farebnú výplň nôt – pomohli si už riešenými úlohami.

Pozrime sa na časť kódu, ktorý vymysleli pre generovanie nôt:

```
p = int(input("zadaj pocet"))
x = 40
for h in range(0, p):
    y = random.randrange(80, 180, 20)
    farba = random.choice((
        "red", "blue", "yellow", "black",
        "green", "orange", "navy", "blueviolet",
        "darksalmon"))
    platno.create_oval(
        x, y - 10, x+20, y+10, fill = farba)
    x = x + 30
```



Obr. 2 Výsledok skupiony Adam, Braňo a Rado

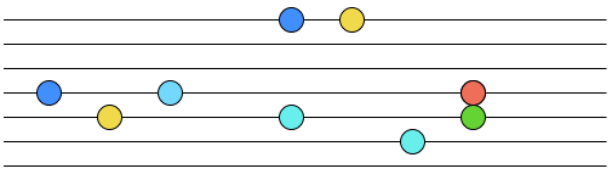
Všimnime si, ako si poradili s výpočtom súradníc: súradnicu x posúvali o 30 bodov, noty sa generovali na čiarach.

2.2 Skupina Rudo a Jana

Rudo a Jana vyriešili úlohu ako prví. Pri riešení sa narazili na viacero problémov, napríklad, ako náhodne výbrať farbu. Každý z členov to urobil iným spôsobom – Jana použila slovné názvy farieb, Rudo použil hexadecimálny zápis farieb. Obaja verili svojmu spôsobu generovania. Ostatné problémy, ktoré mali, riešili formou diskusie medzi sebou, neustálym hľadaním správneho riešenia. Problém si rozdelili na čiastočné úlohy, ktoré jednu po druhej vyriešili. Niekedy bez použitia analytického riešenia – spôsobom pokus/omyl.

Časť riešenia pre generovanie nôt:

```
p = int(input("zadaj pocet not"))
y = 50
for n in range(0, p):
    farba=random.choice((
        "#63d0ff", "#e75747", "#ebd33b", "#11ece6",
        "#3477ff", "#23cc26"))
    x = random.choice((
        30, 80, 130, 180, 230, 280, 330, 380, 430))
    y=random.choice((60, 80, 100, 120, 140, 160))
    platno.create_oval(
        x, y, x + 20, y + 20,
        fill = farba)
```



Obr. 3 Výsledok skupiny Rudo a Jana

Zaujímavé na ich riešení je to, že generovali súradnice x aj y, a tak dostali aj akordy, resp. dvojhmaty, a súčasne v notovom zázname boli aj pauzy. Generovanie súradníc x a y robili z konkrétnej množiny čísel. Rudo a Jana počítali súradnice bodov mriežky, kde noty môžu ležať a náhodne vyberali z týchto hodnôt. Preto im vychádzali aj akordy.

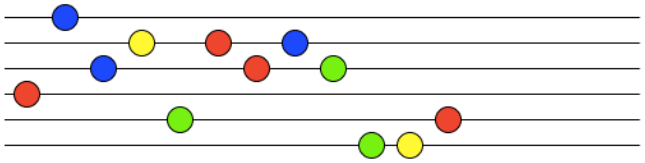
2.3 Martin, Samuel a Zlatica

Martin, Samuel a Zlatica riešila problém nerovnomerne. Vedúci skupiny bol Samuel, ktorý málo zamestnával svojich spolužiakov. Martin hľadal príkazy, ktoré potreboval zistiť Samuel, Zlatica robila zápis paralelne do svojho počítača, výraznejšie neprispievala ku riešeniu, iba komentovala chyby, ktoré boli urobené pri riešení. Prítomnosť učiteľa pri pozorovaní práce skupiny dosť rozptyľovala Samovu pozornosť, cítil sa zviazaný a nepohodlne. Na poznámku učiteľa o ukončení práce Ruda a Jany reagoval slovami: „To sú jedničky v triede, na nich nikto nemá!“ a zaklonil sa na stoličke so zdvihnutím rúk, že on už teda nemusí nič robiť. Ale potom sám ďalej pokračoval v riešení.

V tejto skupine žiaci najprv vykreslili čiary notovej osnovy, a potom riešili generovanie nôt. Skúšali riešenia, ktoré najprv vykresľovali noty len v jednom stĺpci. Noty sa generovali na náhodných miestach, ale iba na čiarach, nie v medzerách. To bol pre nich impulz na ďalšie premýšľanie – ako noty rozmiestniť po celej šírke notovej osnovy. Samuel si uvedomil, že noty musí postupne posúvať o nejakú vzdialenosť v osi x.

Časť riešenie pre generovanie nôt:

```
x=10
for h in range (0, p):
    y=random.randrange(20, 121, 20)
    farba=random.choice((
        "red", "yellow", "blue", "green"))
    platno.create_oval(
        x, y - 10, x + 20, y + 10, fill=farba)
    x = x + 30
```



Obr. 4 Výsledok skupiny Samo, Martin a Zlatica. Noty sú len na čiarach, zápis neobsahuje akordy

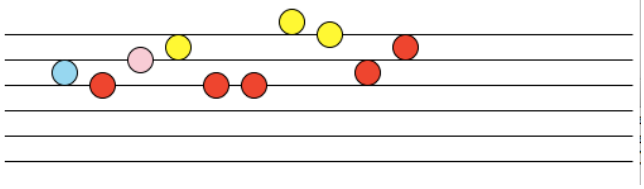
2.4 Dominika a Kristína

Táto skupina pracovala potichu, rozprávali sa navzájom len keď niektorá z dvojice nevedela ako postupovať ďalej. Dospeli k riešeniu, v ktorom sú noty aj v medzerách, zhustené v hornej polovici notovej osnovy. Oni vlastne rozšírili zadanie o ďalšiu podúlohu, ktorú sme si na hodine neuvedomili a nezadali žiakom na ďalšie riešenie. Pri predstavovaní svojho riešenia

spolužiakom, oni si to hneď všimli a skomentovali to ako chybné riešenie: „Dominika, vy nemáte správne riešenie, pretože noty mali byť len na čiarach a vaše sú aj v medzerách!“ Dominika sebavedomo odpovedala: „Ale však noty sa píšu aj do medzier notovej osnovy. My to máme správne!“

Riešenie Dominiky a Kristíny:

```
y=30
x=10
from random import *
for i in range(10) :
    y = randrange(30, 110, 10)
    x = x + 30
    farba=choice((
        "red", "brown", "pink", "yellow", "skyblue"))
    platno.create_oval(
        x, y, x + 20, y + 20, fill = farba)
```



Obr. 5 Výsledok skupiny Dominika a Kristína. Noty sú na čiarach aj v medzerách, ale iba v hornej polovici notovej osnovy, zápis neobsahuje akordy ani dvojhmaty

V našich ukážkach vidieť rôzne rozmýšľanie žiakov nad problémom z pohľadu umiestnenia nôt na notovú osnovu – do medzier, aj na čiary, akordy a bez akordov.

2.5 Skupina Stella a Kristínka

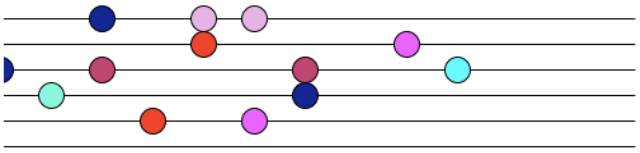
Stella s Kristínkou začali riešenie podobne, ako ostatní, nakreslením notovej osnovy a potom skúšaním polohy nôt bez cyklu. Obidve súradnice nôt generovali, tak sa mohlo stať, že sa vygenerovali akordy alebo dvojhmaty. Dievčatá navzájom spolupracovali a vzájomne sa radili pri hľadaní riešenia.

Dievčatá si najprv napísali hodnoty, z ktorých sa môže generovať súradnica y. Keďže sa im vypisovanie zdalo zdĺhavé, spomenuli si na príkaz výberu čísla z postupnosti čísel a použili ho - príkaz random.randrange.

Dievčatá však už nestihli vyriešiť načítavanie používateľom programu zo vstupu. Ako jediná skupina kreslili kružnicu použitím premennej r – polomer kružnice.

Generovanie nôt:

```
r = 10
for poc in range (13):
    farba = random.choice ((
         "navy", "green", "pink", "red", "maroon",
         "black", "aquamarine", "magenta", "cyan",
         "plum"))
    x= random.randrange(0, 500, 40)
    y= random.randrange (80, 180, 20)
    platno.create_oval(
        x - r, y - r, x + r, y + r, fill = farba)
```



Obr. 6 Výsledok programovania Stelly a Kristínky

3. POSTOJ UČITEĽA

Na žiadne riešenie sme nepovedali, že nie je dobré, alebo to má také chyby a iné má inakšie chyby. Podstatné bolo to, aby riešenie spĺňalo zadané podmienky. Ak ich prekračovalo, tak to bolo výborné. Žiaci sami videli, že ich riešenia môžu byť rôzne, originálne a to neznamená, že to bude zlé, alebo že len jedno riešenie je správne.

3.1 Typy žiackych chýb

Počas riešenia sa jednotlivé skupiny stretávali s podobnými problémami, ktoré riešili skôr spôsobom pokus- omyl, kým im napadlo efektívnejšie riešenie. Okrem bežných syntaktických chýb sa žiaci dopúšťali logických chýb.

Napríklad Stella a Kristínka generovali súradnicu x z intervalu od 0 do 500 bodov. To spôsobilo, že sa niekedy noty zobrazili na okraji plochy polovičné.

Žiaci hľadali riešenie pre posúvanie súradnice x v závislosti na vygenerovanej súradnici y. Jedna časť žiakov generovala obe súradnice a iná súradnicu x posúvala o polomer noty. Žiaci skúšali meniť súradnice v závislosti od grafického výsledku. Ak nedosiahli želaný stav, tak opäť rozmýšľali nad iným riešením.

Zaujímavé bolo sledovať, ako si žiaci poradili s generovaním náhodných čísel z danej množiny. Niektorí použili výber náhodného čísla z obsahu množiny čísel, z ktorej sa má generovať náhodné číslo. Pričom všetky čísla boli vymenované. Niektorí žiaci si spomenuli na funkciu `range`, kde sa určia len hranice intervalu a krok medzi číslami a následne na to funkcia výberu náhodného čísla z takejto oblasti `randrange`.

Žiaci pre generovanie polohy nôt na osnove používali súčasné generovanie x a y. Toto mohlo spôsobiť prekrývanie nôt. Väčšinou si to uvedomili až pri prezentovaní svojho riešenia spolužiakom, keď sa nezobrazoval príslušný počet nôt.

Tento náš postup korešponduje s fázou zbierania skúseností poznávacieho procesu v teórii učenia sa – v konštruktivizme. Ide o riešenie úlohy na základe vlastných skúseností, pri ktorých žiaci robia chyby a odstraňujú ich.

Žiaci predviedli originalitu v riešení úlohy, napríklad Dominika a Kristína (generovanie medzi čiarami), alebo Rudo a Jana (generovanie akordov a dvojhmatov). Tieto originálne riešenia vlastne znamenajú obmenu úlohy, resp. jej ďalšie rozšírenie. Počas hodiny sme si to neuvedomili, a náležite sme to nevyužili pre ostatných žiakov, aby aj oni svoje riešenia podobne rozšírili.

Počas tvorby programov žiaci žiadali vyučujúceho o pomoc. Ako pomôcku sme im poskytli mriežku so súradnicami, na ktorej si kresli noty a prepočítavali súradnice. Aj preto sú polomery nôt a rozostupy medzi notovými čiarami v každej skupine rovnaké.

Po vypracovaní programu každá skupina prezentovala svoje riešenie pred triedou. Žiakov zaujal zápis farieb pomocou

hexadecimálneho kódu u Ruda, musel im podrobne vysvetľovať, ako sa vyberajú farby a čo znamená zápis farby v RGB modeli. Riešenie Ruda a Jany vyvolalo diskusiu aj o tom, ako generovali súradnice x a y. Vyberali ich náhodne z vymenovanej množiny súradníc. Žiaci si uvedomili, že takýmto generovaním môže notový zápis obsahovať akordy, ale aj pauzy – voľné miesto v notovej osnove.

Žiaci sa pozastavili pri príkaze random.randrange(a,b,c) v riešení, ktoré prezentovala Kristínka a Stella. Niektoré skupiny zabudli na tento príkaz a vyberali súradnice z vymenovanej množiny prvkov, podobne ako to robili pri farbách.

Najväčšiu diskusiu vyvolalo riešenie Dominiky a Kristíny, ktoré najprv okomentovali kód, ale až po spustení žiaci videli, že noty sa generujú aj medzi čiarami. Ako sme spomenuli skôr, Dominika toto prezentovala ako zámer, že naozajstné noty sa vpisujú aj do medzier notovej osnovy.

Táto chyba posunula riešenie úlohy na vyššiu úroveň.

3.2 Reakcia učiteľa na zistenú chybu žiaka

Z vlastných skúseností a pozorovaní, ktoré robíme na svojich hodinách vidíme, ako reagujú žiaci, ktorí majú strach z toho, že urobia chybu.

V pozorovanej skupine prvákov sú žiaci, ktorí sa veľmi ťažko zapájajú do diskusie, aj na priame oslovenie. Napríklad Dominika, s obavou odpovie na otázku, aj keď je to vždy správne. Podobne sa chovajú aj Kristína a Stella. Kristína, Dominika získali veľmi premyslené riešenie, spolupracovali navzájom potichu, ale bolo vidieť, že sa boria z niektorými problémami – výber náhodnej farby, generovanie výšky tónu. Práve táto skupinka generovala polohu noty na čiary aj do medzier notovej osnovy. Na priame oslovenie na obhajobu svojho riešenia pred triedou prišla Kristínka a všetky kroky presne zdôvodnila. Až pri obhajobe zistila, že im chýba načítavanie počtu nôt počas behu programu. Tomuto javu sme nepripisovali veľkú váhu a vlastne sme situáciu zahovorili slovami: „ Vyriešili ste to originálne a ten vstup si určite dorobíte ešte teraz, počas hodiny. “ Opýtali sme sa ešte: „Prečo sú vaše noty aj v medzerách medzi čiarami? V zadaní bolo napísané, že noty majú byť len na čiarach“ Dominika duchaplne odpovedala: „Ale však noty sa píšu aj medzi čiary.“ Predpokladám, že ani nerozmýšľali nad tým, že noty budú generovať ináč.

Ďalšie naše reakcie na chyby žiakov boli, napríklad: „Nebojte sa, že sa vám niečo nepodarí. Bez skúšania a overovania nebudete postupovať a nedôjdete k riešeniu.“

Úlohu riešili aj iní žiaci na inej hodine v druhom ročníku: „Matúš, vidím, že tie čiary nie sú vodorovné, ako sme chceli. Zober papier a tužku a napíš si súradnice jednotlivých čiar a hľadaj nejakú spoločnú črtu, ktorú by si vedel využiť. “

Našim cieľom je to, aby sme žiakov nestrašili, neodradili zvyšovaním hlasu, alebo zhadzovaním za ich chyby pri riešení úloh.

Niekedy však žiaci vyvíjajú nátlak na to, aby sme ako učitelia dodržiavala tzv. **Didaktickú zmluvu.** O čo ide. Žiaci a aj niektorí učitelia majú „školský reflex“, ktorý spočíva v nesprávnom pochopení aktivity učiteľa a žiaka na hodine: žiak má na hodine počúvať, čo hovorí učiteľ a učiteľ má žiakovi pomáhať s riešením. [4]. Učiteľ, ktorého zavolajú žiaci na pomoc očakávajú, že učiteľ im pomôže s riešením. Učiteľ vie, že jeho úlohou je žiakom pomáhať, preto sa zapojí do riešenia úlohy a pomôže im. Tým ich však oberie o hľadanie riešenia, čo je najužitočnejšie pre poznávací proces. Takto vlastne žiaci vedia, že stačí zodvihnúť ruku a učiteľ príde a vyrieši úlohu za žiakov. V zápale práce sa občas zabudneme a žiakom pomôžeme. Vhodné je pripraviť si odpoveď pre žiakov – „Nakreslite si riešenie na papier, porozmýšľajte spoločne so spolužiakom.“

Ak prejdeme okolo všetkých skupín a vidíme, že žiadna z nich si nevie poradiť, tak na riešení pracujeme spoločne na tabuli.

Pri našich rozboroch hodín sa zamýšľame nad otázkou: „A čo ostatní žiaci? Nehnevajú sa, že sa im riešenie prezradí?“ Priznáme sa, že niekedy žiaci zakričia: „Však ste nám to nemali prezradiť, už sme na to skoro prišli!“, alebo „To ma teraz už nebude baviť riešiť túto úlohu!“

Týmto poznámkam sa veľmi teším, a ak sa niekto podobne vyjadrí, tak s riešením pomôžeme iba skupine, ktorá si evidentne nevie pomôcť.

4. CHYBY Z POHĽADU TEÓRIE

Podľa psychológov [5] [6], vždy, keď sa žiak pomýli, tak sa v mozgu vytvorí nové synaptické spojenie. Vtedy vzniká mozgová aktivita, a mozog rastie. Mozog vynakladá zväčšené úsilie vtedy, keď čelí problému. Psychológovia meraním mozgovej aktivity zistili, že keď sa človek pomýli, tak jeho mozgová aktivita je väčšia, ako keď odpovedá správne, keď sa nepomýli.

Vedci taktiež zistili, že ľudia s otvorenou mysľou vykazujú pri riešení problému väčšiu mozgovú aktivitu, ako ľudia s uzavretou mysľou [7]. Tieto štúdie ukazujú, že dopúšťať sa chýb je vlastne dobré, pretože pri nich vzniká väčšia mozgová aktivita, a vtedy mozog rastie.

V našom školstve sú žiaci trestaní za chyby, cítia sa zle, aj preto, že si myslia, že nemajú dostatočný intelekt na daný predmet.

Zo záverov vedeckých meraní vyplýva, že žiaci sa najviac naučia na takých úlohách, kedy sa budú mýliť, riešenie nebude hneď viditeľné, budú mať rôznorodé riešenia alebo prístupy.

Učiteľ však žiakom najprv musí vysvetliť, že chyby sú veľmi prospešné, že sa za ne nesmú hanbiť, že rozvíjajú ľudský mozog. Musí naučiť žiakov vnímať chyby pozitívne. Vtedy to bude mať na nich oslobodzujúci účinok.

4.1 Ocenenie chýb verzus známkovanie

Na našich hodinách sme sa rozhodli oceňovať žiakov, ktorí majú iné riešenia problému, ako ostatní a aj tých , ktorí urobili chybu. Ak im v priebehu vyučovania zdôrazňujeme, že robenie chýb je práve spôsob, ako sa učia nové veci, tak získajú sebavedomie a budú ochotnejší diskutovať o riešeniach.

Sme si vedomí toho, že podľa [8] známkovanie a odmeny potláčajú vnútornú motiváciu žiakov pre učenie. Stále riešime dilemu, ako teda žiakov hodnotiť, aby sme nepotlačili zanietenie žiakov pre programovanie a súčasne aby sme mali aj známky z predmetu.

V súčasnosti na hodinách používame slovné hodnotenie práce žiakov, ktoré si aj značíme. Známky, ktoré dávame, sú za domáce zadania a za sumatívne kontrolné práce po prebratí kapitoly.

5. ZÁVER

V našom príspevku sme popisovali priebeh riešenia úlohy žiakmi, ich uvažovanie a rozoberanie problému. Pozorovali sme, ako písanie kódu a jeho zlepšovanie vedie ku vyriešeniu úlohy, pozorovali sme ako chybné čiastkové riešenia posúvajú žiakov ku správnym riešeniam. Súčasne sme si my, ako učitelia, uvedomili, aké je dôležité jasne a zrozumiteľne formulovať zadanie. Pri podobných aktivitách je dôležité zadanie formulovať široko -

divergentne, aby úlohu bolo možné riešiť rôznymi spôsobmi, aby to neboli úlohy, ktoré hneď vyriešia správne. My chceme aby sa žiaci na chybách učili, aby sa k nim vracali a naprávali ich.

Učitelia si málokedy pripúšťajú, ako na žiakov pôsobí ich štýl vyučovania, ako môžu škodiť žiakom v tom, že ich nenechajú hľadať riešenie problémov samostatne, keď sú netrpezliví a poháňajú žiakov. Mali by sme sa zamýšľať nad vzdelávacím procesom z hľadiska kultivácie duševného sveta žiaka a nie z hľadiska metód, ktoré budú rýchlo viesť k cieľu. [9]

Učiteľ musí nechať žiakov aby riešenie hľadali, aby sa vydávali aj na nesprávne „cestičky" v riešení, aby skúmali a sami objavovali krásu poznávania a „aha-efektu". Programovanie je oblasť, kde robenie chýb, hľadanie a skúšanie, je súčasťou riešenia a spôsobuje zapojenie žiakovej mysle a prebudenie jeho záujmu o predmet.

6. BIBLIOGRAFIA

[1] M. SPIŠÁKOVÁ a Ľ. SALANCI, „Meranie motivácie žiakov ku výkonu," rev. *DIDINFO 2015*, Banská Bystrica, 2015.

[2] M. SPIŠÁKOVÁ a Ľ. SALANCI, „Motivácia vo vyučovaní informatiky," *DIDINFO 2016*, 2016.

[3] Z. TOMKOVÁ, „Gymnázium Pavla Horova Michalovce," Február 2015. [Online]. Available: http://www.gphmi.sk/fileadmin/dokumenty/informatika/IGePeHa15/zadanie-starsi.pdf. [Cit. január 2017].

[4] G. BROUSSEAU, „The crucial role of the didactical contract in the analysis and construction of situations in teaching and learning mathematics," *Theory of mathematics education: ICME 5 topic area and miniconference*, pp. 110 - 119, 1984.

[5] J. S. MOSER, H. S. SCHRODER, C. HEETER, T. P. MORAN a L. YU-HAO, „Mind Your Errors, Evidence for a Neural Mechanism Linking Growth Mind-Set to Adaptive Posterror Adjustments," *SAGE Journals: Psychological Science*, zv. 22, %1. vyd.12, pp. 1484-1489, 31 Oktober 2011.

[6] C. S. DWECK, Mindset: The New Psychology of Success, New York: Random House, 2006.

[7] J. BOALEROVÁ, Matematické cítenie - Odhaľte potenciál študentov prostredníctvom kreatívnej matematiky, motivačného prístupu a inovatívneho spôsobu učenia, Bratislava, Slovensko: Tatran, 2016, p. 325.

[8] J. NOVÁČKOVÁ, Mýty ve vzdělávaní, Bystřice pod Hostýnem: PhDr. Pavel Kopřiva, 2015.

[9] M. HEJNÝ a F. KUŘINA, Dítě, škola a matematika, Praha: Portál, 2015.

[10] O. AUBIOLA a H. S. DHINDSA, „Improving classroom practices using our knowledge of how the brain works," *International Journal of Environmental & Science Education*, zv. 1, %1. vyd.7, pp. 71-81, marec 2011.

[11] J. BOALEROVÁ, „Matematické cítenie - Odhaľte potenciál študentov prostredníctvom kreatívnej matematiky, motivačného prístupu a inovatívneho spôsobu učenia," Tatran, 2016, pp. 17 - 19.

[12] V. HRABAL, F. MAN a I. PAVELKOVÁ, Psychologické otázky motivace ve škole, Praha: Státní pedagogické nakladatelství, 1989.

[13] A. KARNI, G. MEYER, C. REY-HIPOLITO, P. JEZZARD, M. M. ADAMS, R. TURNER a L. G. UNGERLEIDER, „The acquisition of skilled motor performance: Fast and slow experience-driven changes in primary motor cortex," *Colloquium Paper*, zv. 95, p. 861–868, Február 1998.

[14] S. KASSIN, Psychologie, Albatros Media a.s, 2012.

[15] C. M. MUELLER a C. S. DWECK, „Praise for Intelligence Can Undermine Children's Motivation and Performance," *Journal of Personality and Social Psychology*, zv. 75, pp. 33-52, 1998.

[16] T. PARDEL, L. MARŠÁLOVÁ a A. HRABOVSKÁ, „Dotazník výkonovej motivácie," 1984.

[17] I. PAVELKOVÁ, „Motivace žáků," *Kapitoly ze školní pedagogiky a školsní psychologie,* pp. 93 - 103, 2008.

[18] G. PETTY, Moderní vučování, Praha: Portál, 2008.

[19] M. SPIŠÁKOVÁ a Ľ. SALANCI, „Meranie motivácie žiakov ku výkonu," *DIDINFO 2015*, pp. 139 - 143, 2015.

[20] M. SPIŠÁKOVÁ a Ľ. SALANCI, „Meranie motivácie žiakov ku výkonu," rev. *PRIT 2015*, České Budejovice, 2015.

Úspešnosť študentov učiteľstva informatiky
Success of University Students of Teacher Training in Computer Science Teaching Programme

Veronika Stoffová
Pedagogická fakulta Trnavskej univerzity
Priemyselná 4, P. O. BOX 9
918 43 Trnava
Slovenská republika
veronika.stoffova@truni.sk

Roman Horváth
Pedagogická fakulta Trnavskej univerzity
Priemyselná 4, P. O. BOX 9
918 43 Trnava
Slovenská republika
roman.horvath@truni.sk

ABSTRACT
The interest in study the teacher training programme in the informatics (computer science) field of study is declining steadily. Globally it is explained and justified by decrease in population and reducing the total number of college/university students. The authors have long been monitoring the development of the area of teacher training in computer science programmes (there are several programmes in combination with other subject). The authors seek the answers to many of the related issues in this article. They are looking for the causes of failure of the students in the first year of their university studies. The formulated answers are based on the analysis of students' answers from the Faculty of Education of Trnava University in a questionnaire and their own observations and experience of the teachers from other universities.

KEYWORDS
Teaching programming. Programming languages taught. Estimated professional focus of applicants. Estimated applicants' knowledge of programming.

ABSTRAKT
Záujem o štúdium učiteľstva informatiky neustále klesá. Globálne sa to vysvetľuje a zdôvodňuje poklesom populácie a znížením celkového počtu vysokoškolákov. Autori článku už dlhodobo sledujú tento vývoj v oblasti učiteľstva informatiky. V článku hľadajú odpoveď na mnohé s tým súvisiace otázky. Na základe analýzy odpovedí študentov Pedagogickej fakulty Trnavskej univerzity v dotazníku a vlastných pozorovaní a skúseností aj z iných univerzitných pracovísk, hľadajú príčiny neúspešnosti študentov v prvých ročníkoch vysokoškolského štúdia.

KĽÚČOVÉ SLOVÁ
Vyučovanie programovania. Vyučované programovacie jazyky. Odhadované odborné zameranie uchádzačov. Odhadované vedomosti uchádzačov z programovania.

1. ÚSPEŠNOSŤ ŠTUDENTOV PRICHÁDZAJÚCICH ŠTUDOVAŤ UČITEĽSTVO INFORMATIKY

Dlhodobý pokles záujmu maturantov o štúdium učiteľstva informatiky je znepokojujúci. Nepríjemné je aj to, že z tých študentov, ktorí prejavili záujem o štúdium a boli prijatí, nenastúpia do prvých ročníkov všetci. Navyše mnohí z tých, ktorí nastúpia, príp. sa zapíšu na učiteľské štúdium informatiky nedokončia ani len prvý semester, a do druhého ročníka sa z nich dostane približne 50 – 70 %. Na magisterskom stupni z nich pokračuje sotva 50 %. Treba poznamenať i to, že rapídne klesá počet študentov študujúcich v programoch v kombinácii informatiky s prírodo-vednými predmetmi a narastajú počty študentov, ktorí kombinujú informatiku s jazykmi. V tomto článku sú uvedené údaje, ktoré odrážajú situáciu posledných rokov na Pedagogickej fakulte Trnavskej univerzity, ktorá má akreditované iba študijné programy učiteľstva informatiky v kombinácii. Veľmi podobná je situácia aj na iných učiteľských fakultách na Slovensku, ktoré pripravujú učiteľov informatiky.

Kontinuitu udržiavania počtov študentov učiteľstva informatiky narušili neúspešná akreditácia a otvorenie iných študijných programov orientovaných na informatiku, napr. aplikovanej informatiky na fakultách, ktoré pôvodne ponúkali len učiteľský smer. (Ako príklad môžeme uviesť otvorenie študijného programu geografické informačné systémy a neskôr aplikovaná informatika na Fakulte prírodných a humanitných vied v Banskej Bystrici. Podobne možno spomenúť otvorenie bakalárskeho a neskôr aj magisterského štúdia aplikovanej informatiky na Fakulte prírodných vied v Nitre atď.) Rapídne poklesol počet prihlásených študentov na učiteľstvo informatiky v kombinácii s inými predmetmi aj na Univerzite J. Selyeho v Komárne (UJS), na ktorej po získaní akreditácie na aplikovanú informatiku zaregistrovali nielen nových záujemcov o štúdium tohto odboru, ale aj veľké preskupenie študentov medzi študovanými odbormi. Vzhľadom na to, že UJS získala akreditáciu po ukončení procesu prihlasovania sa na univerzitu, bolo prijímacie konanie na aplikovanú informatiku realizované dodatočne. Zároveň bola uchádzačom oznámená aj možnosť zmeny učiteľskej aprobácie na aplikovanú informatiku. Išlo o úsilie získať dostatočný počet záujemcov na aplikovanú informatiku, aby sa mohlo začať s výučbou už v aktuálnom akademickom roku.

Na pokles študentov učiteľstva informatiky pre ročníky 6. – 12. nášho školského systému mohli do určitej miery vplývať aj pre-orientovanie, príp. rozšírenie odborného profilu katedier techniky, a technickej výchovy o informačnú výchovu. Tým si tieto katedry do určitej miery prisvojili právo vychovávať učiteľov informačnej výchovy, príp. informatiky pre základné školy. Tieto príčiny oslabenia záujmu o učiteľstvo informatiky na Trnavskej univerzite neplatia, pretože tu nie je aplikovaná informatika ani technická a informačná výchova ako to je na mnohých univerzitách. Sú tu však ďalšie špecifiká a faktory, ktoré môžu mať vplyv na rozhodnutie uchádzača pri voľbe vysokej školy a študijného zamerania. Okrem Trnavskej univerzity sídlia v Trnave ďalšie dve vysokoškolské ustanovizne, Materiálovotechnologická fakulta Slovenskej technickej univerzity (MtF STU) a Univerzita sv. Cyrila a Metoda (UCM). Na obidvoch univerzitách je možné

študovať aplikovanú informatiku. Absolventi stredných odborných a priemyselných škôl, ktorí chcú získať vysokoškolské vzdelanie si skôr vyberú inžinierske štúdium na MtF STU než magisterské štúdium učiteľstva na Trnavskej univerzite.

Na Pedagogickú fakultu Trnavskej univerzity sa v akademickom roku 2016/17 prihlásilo na štúdium učiteľskej informatiky 50 študentov. Prijatých bolo 46 študentov, ale do prvého ročníka sa zapísalo len 29 študentov. Z tohto počtu kombinovalo informatiku s jazykom 13 študentov, čiže necelých 45 %. Prírodovedné predmety s informatikou kombinovalo 16 študentov, teda o niečo viac ako 55 %.

Do dotazníkového výskumu, výsledky ktorého uvádzame, sa zapojilo len 21 študentov. Po uplynutí skúškového obdobia zimného semestra splnilo podmienky ukončenia semestra a získalo právo pokračovať v štúdiu v letnom semestri len 14 študentov. Smutné je, že mnohí z nich sa na skúšku z kľúčových predmetov informatiky (ako je napr. programovanie) ani len neprihlásili alebo skúšku vzdali hneď na začiatku, aj keď počas semestra v priebežnom hodnotení obstáli (teda splnili stanovené kritériá hodnotenia kedysi označovaného termínom „zápočet"; pozri tabuľku 1). V prezenčných listinách zimného semestra permanentne vystupuje len asi 21 – 23 študentov a dotazník zadaný na začiatku semestra vyplnilo len 21 študentov. Svedčí to o tom, že počet študujúcich prvákov v zimnom semestri nezodpovedá počtu reálne zapísaných študentov (pozri tabuľku 1).

Tabuľka 1. Prehľad študentov prihlásených, prijatých a zapísaných do 1. ročníka

AR	PŠP	PŠA	PŠZ	P2R
2016/17	50	46 (92 %)	29 (58 %)	(14) LS
2015/16	61	53 (87 %)	32 (49 %)	24
2014/15	65	48 (74 %)	45 (69 %)	20
2013/14	74	67 (91 %)	37 (50 %)	25
2012/13	78	67 (86 %)	39 (50 %)	19

Vysvetlivky: AR – akademický rok, PŠP – počet študentov prihlásených do 1. ročníka, PŠA – počet akceptovaných študentov z počtu prihlásených, PŠZ – počet študentov zapísaných do 1. ročníka, P2R – počet študentov, ktorí postúpili do 2. ročníka, LS – stav po prvom semestri (počet študentov, ktorí postúpili do letného semestra).

Tabuľka 1 ukazuje vývoj počtov študentov prihlásených, prijatých a zapísaných do prvého ročníka za posledných päť rokov. Počty prihlásených študentov: 78, 74, 65, 61, 50 ukazujú klesajúci záujem o štúdium učiteľstva informatiky. Počet zapísaných (ale nie nastúpených) študentov sa väčšinou pohybuje v rozmedzí 55 – 60 % z tohto počtu. Počet nastúpených študentov určených podľa evidencie študentov na hodinách programovania počas prvého semestra je ešte nižší (pozri tabuľku 3, stĺpec PŠE). Výnimkou je akademický rok 2014/15, kedy sa do prvého ročníka zapísalo skoro 70 % prihlásených študentov, čo bolo až 94 % prijatých študentov (čiže takmer všetci prijatí – rozdiel medzi prijatými a zapísanými tvorili iba traja študenti).

I napriek prísnemu výberu študentov prijímaných do prvého ročníka, postúpilo do druhého ročníka relatívne malé percento zo zapísaných študentov (spravidla asi 45 %). Podobne zlý pomer očakávame aj v tomto akademickom roku, v ktorom v súčasnosti z 29 študentov zapísaných do 1. ročníka pokračuje v letnom semestri len 14. Tabuľka 2 ukazuje vývoj počtov študentov zapísaných do prvého ročníka štúdia učiteľstva informatiky a tiež aj to, aké predmety si zvolili ako druhý aprobačný predmet.

Tabuľka 2. Prehľad počtov prihlásených uchádzačov na štúdium informatiky na Pedagogickej fakulte Trnavskej univerzity v Trnave

AR ST1	PŠP/PŠA 1R	PŠZ 1R	PŠZ 2R	PŠZ 3R	PŠZ 4R	PŠZ 5R
2016/17 50/46/29	AJ: 17/15	AJ: 9	AJ: 9	AJ: 6	AJ: 5	AJ: –
	SJ: 8/8	SJ: 4	SJ: –	SJ: 2	SJ: 5	SJ: –
	M: 11/10	M: 6	M: 7	M: 7	M: 2	M: 3
	NV: –	NV: –	NV: 1	NV: 1	NV: 1	NV: –
	B: 14/13	B: 10	B: 3	B: 4	B: –	B: –
AR ST1	**PŠP/PŠA 1R**	**PŠZ 1R**	**PŠZ 2R**	**PŠZ 3R**	**PŠZ 4R**	**PŠZ 5R**
2015/16 61/53/32	AJ: 31/28	AJ: 16	AJ: 8	AJ: 10	AJ: 1	AJ: 2
	SJ: 5/4	SJ: 2	SJ: 5	SJ: 5	SJ: –	SJ: –
	M: 16/14	M: 10	M: 5	M: 6	M: 3	M: 3
	NV: –	NV–	NV: 1	NV: 1	NV: –	NV: –
	B: 9/7	B: 4	B: 6	B: 1	B: –	B: –
AR ST1	**PŠP/PŠA 1R**	**PŠZ 1R**	**PŠZ 2R**	**PŠZ 3R**	**PŠZ 4R**	**PŠZ 5R**
2014/15 65/48/45	AJ: 35/33	AJ: 20	AJ: 9	AJ: 6	AJ: 3	AJ: 1
	SJ: 5/4	SJ: 9	SJ: 6	SJ: 2	SJ: –	SJ –
	M: 16/14	M: 7	M: 2	M: 6	M: 3	M: 3
	NV: –	NV: 1	NV: 1	NV: –	NV: –	NV: –
	B: 9/7	B: 8	B: 2	B: –	B: –	B: 3
AR ST1	**PŠP/PŠA 1R**	**PŠZ 1R**	**PŠZ 2R**	**PŠZ 3R**	**PŠZ 4R**	**PŠZ 5R**
2013/14 74/67/37	AJ: 24/23	AJ: 17	AJ: 8	AJ: 5	AJ: 2	AJ: 5
	SJ: 24/21	SJ: 11	SJ: 2	SJ: 1	SJ: –	SJ –
	M: 13/12	M: 5	M: 8	M: 5	M: 2 + 1	M: 1
	NV: 5/4	NV: 1	NV: –	NV: –	NV: –	NV: –
	B: 8/7	B: 3	B: 3	B: –	B: 3	B: 9
AR ST1	**PŠP/PŠA 1R**	**PŠZ 1R**	**PŠZ 2R**	**PŠZ 3R**	**PŠZ 4R**	**PŠZ 5R**
2012/13 78/67/39	AJ: 35/30	AJ: 17	AJ: 5	AJ: 3	AJ: 4	AJ: 3
	SJ: 9/8	SJ: 6	SJ: 3	SJ: –	SJ: –	SJ –
	M: 19/17	M: 10	M: 6	M: 4	M: 1	M: –
	NV: 1/1	NV: –	NV: –	NV: –	NV: –	NV: –
	B: 14/12	B: 6	B: –	B: 7	B: 9	B: 3
	F: –	F: –	F: –	F: –	F: 6	F: –

Vysvetlivky: AR ST1 – akademický rok a stručný stav z tabuľky 1, PŠP – počet prihlásených študentov, PŠA – počet akceptovaných študentov, PŠZ – počet zapísaných študentov, 1R – 5R – do 1. – 5. ročníka, AJ – anglický jazyk a literatúra, SJ – slovenský jazyk a literatúra, M – matematika, NV – náboženská výchova (katolícka), B – biológia, F – fyzika.

Z tabuľky 2 vidieť, že z 39 študentov informatiky, ktorí sa zapísali na štúdium na Pedagogickú fakultu Trnavskej univerzity v akademickom roku 2012/13 skončili štúdium v akademickom roku 2016/17 so získaním magisterského titulu len traja. Z 37 prvákov z akademického roka 2013/14 postúpilo do 4. ročníka v akademickom roku 2016/17 len 13 študentov. Zo 45 zapísaných prvá-kov v akademickom roku 2012/13 postúpilo do 3. ročníka v akademickom roku 2016/17 len 20 študentov.

2. TYPY ŠKÔL, Z KTORÝCH UCHÁDZAČI PRICHÁDZAJÚ

Tabuľka 3 uvádza prehľad toho, z akých škôl prichádzajú študenti študovať učiteľstvo informatiky. Tabuľka je spracovaná na základe vyplnených dotazníkov. Z uvedených čísel vidieť, že počet respondentov je nižší než je počet štatisticky evidovaných študentov. Je potešujúce, že medzi typmi stredných škôl dominujú gymnáziá. Za posledné tri roky uvádzame aj percentá počtov študentov, ktorí získali maturitu na gymnáziu.

Uvádzané čísla a z toho vyplývajúce pomery, však nemusia odrážať celoslovenský stav. Úspešnosť študentov v prvom ročníku ukazuje, že gymnazisti sú v prvom ročníku štúdia úspešnejší než študenti z ostatných typov stredných škôl.

Percentuálna úspešnosť študentov, ktorí prišli študovať informatiku na Pedagogickú fakultu Trnavskej univerzity v Trnave z gymnázií sa za posledné tri roky pohybuje v intervale 72 – 88 %. Určitú deformáciu vnášajú do hodnotenia nízke hodnoty počtov študentov prichádzajúcich z určitých typov škôl, kde celú vzorku (100 % študentov) tvoria len 1 – 3 študenti. Typ strednej školy „neznámy“ sme použili v tých prípadoch, v ktorých študent túto informáciu v dotazníku nevyplnil, prípadne ak študent nevyplnil dotazník vôbec.

Tabuľka 3. Typy škôl, z ktorých prichádzajú študenti informatiky (podľa vyplnených dotazníkov)

AR	PŠE	TSŠ	PTŠ	PCP	P2R	PTŠ	PÚ
2016/17	21	GYM	8	38	(14) LS	7	87,5
		OA	1	4,8		1	100
		SOŠ	2	9,5		1	50
		iný	2	9,5		2	100
		neznámy	8	38		3	37,5
2015/16	33	GYM	11	33,3	19	8	72,7
		OA	1	3,1		0	0
		SPŠ	3	9,1		1	33,3
		SOŠ	11	33,3		7	63,6
		iný	3	3,1		3	100
		neznámy	4	12,1		0	0
2014/15	35	GYM	13	37,1	21	10	77
		OA	5	14,3		2	40
		SPŠ	5	14,3		3	60
		SOU	12	34,3		6	50

Vysvetlivky: AR – akademický rok, PŠE – počet študentov evidovaných na hodinách programovania v 1. ročníku, TSŠ – typ strednej školy, PTŠ – počet podľa typu strednej školy, PCP – percento z celkového počtu, P2R – počet študentov, ktorí postúpili do 2. ročníka, PÚ – priemerná percentuálna úspešnosť študentov.

3. ODHADOVANÉ ZAMERANIE NAŠICH UCHÁDZAČOV POSUDZOVANÉ PODĽA OBĽÚBENÝCH A NEOBĽÚBENÝCH PREDMETOV

Autorov článku dlhodobo zaujíma, aká vzorka študentov prichádza študovať učiteľstvo informatiky, aká je ich úspešnosť a čo je príčinou ich neúspechu, príp. predčasného ukončenia štúdia.

Niektoré výsledky monitorovania, štatistické hodnoty a výsledky získané počas posledných rokov na Pedagogickej fakulte Trnavskej univerzity sú zosumarizované v tabuľke 4.

Tabuľka 4. Obľúbené/neobľúbené predmety študentov

AR	PŠD	Obľúbený predmet	#	Neobľúbený predmet	#
2016/17	21 (študenti uvádzali jeden až štyri predmety v každej kategórii)	anglický jazyk	3	anglický jazyk	2
		biológia	3	občianska náuka	1
		dejepis	2	dejepis	3
		fyzika	2	fyzika	1
		chémia	1	chémia	3
		informatika	7	nemecký jazyk	1
		matematika	3	matematika	5
		počítačová grafika	1	úvod do makroekonómie	1
		priestorová tvorba	1	kresba	1
		psychológia	1	psychológia	1
		slovenský jazyk	1	slovenský jazyk	7
		telesná výchova	4		
		súčet	29	**súčet**	26

tabuľka pokračuje na ďalšej strane…

V tabuľke 4 sú uvedené obľúbené a neobľúbené predmety respondentov na strednej škole. V akademickom roku 2016/17 uvádzali študenti jeden až štyri predmety v kategórii obľúbených a neobľúbených predmetov. Pri 21 respondentoch bolo v dotazníku sumárne zaznamenaných 29 obľúbených predmetov (z 12 rôznych). Informatika a počítačová grafika boli uvedené 8-krát, ďalšie prírodovedné predmety 9-krát, jazyky 4-krát. Z neobľúbených predmetov bolo v dotazníkoch zaznamenaných 26 výskytov (z 11 rôznych). Z toho ani jeden predmet nesúvisel s informatikou, 10-krát sa vyskytli jazykové predmety a 9-krát išlo o prírodovedné predmety (5-krát matematika, 3-krát chémia a raz fyzika).

V akademickom roku 2015/16 (33 respondentov) ako aj v akademickom roku 2014/15 (35 respondentov) uvádzali študenti jeden až šesť predmetov v kategórii obľúbených a tiež v neobľúbených predmetov. Celkový počet výskytov predmetov v oboch kategóriách bol oproti poslednému sledovanému roku vyšší.

V akademickom roku 2015/16 sme pri 33 respondentoch zaznamenali 64 výskytov obľúbených predmetov (z 19 rôznych). Informatika a príbuzné predmety boli uvedené 14-krát, ďalšie prírodovedné predmety 23-krát, jazyky 14-krát. Bolo tiež zaznamenaných 57 výskytov neobľúbených predmetov (z 25 rôznych). Z toho 4-krát predmety súvisiace s informatikou, 18-krát jazykové predmety a 9-krát prírodovedné predmety (5-krát matematika, 3-krát chémia a raz fyzika).

…pokračovanie tabuľky 4

AR	PŠD	Obľúbený predmet	#	Neobľúbený predmet	#
2015/16	33 (študenti uvádzali jeden až šesť predmetov v každej kategórii)	anglický jazyk	10	anglický jazyk	7
		biochémia	1	anorganická chémia	1
		biológia	5	biológia	2
		deskr. geometria	1	dejepis	7
		ekonómia	1	ekonomika	2
		farmakológia	1	analytická chémia	1
		fyzika	3	statika	1
		grafika a multimédiá	1	počítačové siete	1
		chémia	1	chémia	5
		informatika	12	informatika	1
		konv. v ang. jazyku	1	nemecký jazyk	4
		manažment	1	logistika	1
		marketing	1	ochrana podnik. inf.	1
		matematika	12	matematika	2
		počítačové systémy	1	program. a sprac. dát	1
		príprava liekov	1	technol. liekových foriem	1
		slovenský jazyk	3	slovenský jazyk	7
		telesná výchova	8	telesná výchova	1
		účtovníctvo	1	účtovníctvo	2
				betónové konštrukcie	1
				doprava	1
				občianska náuka	4
				prax v hoteloch	1
				strojníctvo	1
				teória medzin. obch.	1
		súčet	65	**súčet**	57

tabuľka pokračuje vpravo…

V akademickom roku 2014/15 bolo pri 35 respondentoch zaznamenaných 86 výskytov obľúbených predmetov (z 29 rôznych). Informatika a príbuzné predmety boli uvedené 21-krát, ďalšie prírodovedné predmety 18-krát, jazyky 13-krát. Pri neobľúbených predmetoch bolo celkovo zaznamenaných 65 výskytov (z 18 rôznych predmetov). Z toho jeden súvisiaci s informatikou, 17-krát išlo o jazykový predmet a 19-krát o prírodovedné predmety (11-krát matematika, 5-krát chémia a 3-krát fyzika).

4. ODHADOVANÉ VEDOMOSTI UCHÁDZAČOV Z OBLASTI PROGRAMOVANIA ZÍSKANÉ NA ZÁKLADE DOTAZNÍKOV

Od roku 2008 na základných a stredných školách prebieha výučba podľa nového štátneho vzdelávacieho programu. Štandardy vytvorené v rámci tohto dokumentu vyjadrujú požiadavky na výstupné vedomosti z oblasti informatiky pre žiakov základných škôl, aj pre absolventov stredných škôl (Porubská a kol., 2008; Blaho, 2012; Salanci,°2014; Salanci,°2015). V akademickom roku 2015/16 nastúpili na vysoké školy prváci, ktorí absolvovali túto prípravu už podľa nových štandardov.

…dokončenie tabuľky 4

AR	PŠD	Obľúbený predmet	#	Neobľúbený predmet	#
2014/15	35 (študenti uvádzali jeden až šesť predmetov v každej kategórii)	anglický jazyk	15	anglický jazyk	6
		aplikovaná inf.	2		
		automatizácia	1		
		biológia	4	biológia	3
		CAD systémy	1		
		cvičenia z inf.	1		
		dejepis	2	dejepis	5
		ekonomické cvičenia	1	marketing	1
		ekonomika	3	ekonomika	6
		etická výchova	1		
		fyzika	4	fyzika	3
		geografia	2	geografia	4
		hardv. vybav. poč.	1	základy elektrotechniky	2
		chémia	1	chémia	5
		informatika	9		
		matematika	9	matematika	11
		nemecký jazyk	1	nemecký jazyk	5
		počítačová grafika	2		
		počítačové siete	1	sieťové technológie	1
		právo	2	právo pre podnikateľov	1
		programovanie	2		
		ruský jazyk	1		
		slovenský jazyk	7	slovenský jazyk	6
		služby cest. ruchu	1	občianska náuka	1
		softv. vybav. poč.	1	tovaroznalectvo	1
		sprac. obrazu a textu	2		
		španielsky jazyk	1	španielsky jazyk	1
		telesná výchova	3		
		účtovníctvo	5	účtovníctvo	3
		súčet	86	**súčet**	65

Vysvetlivky: AR – akademický rok, PŠD – počet dotazníkov vyplnených študentmi, # – počet študentov

Komentár k tabuľke 4: Žltou farbou pozadia sú zvýraznené informatické a príbuzné predmety; ružovou farbou prírodovedné predmety; zelenou farbou ďalšie prírodovedné predmety.

Žiaľ, ich vedomosti, nezodpovedajú očakávaniam a je potrebné chýbajúce vedomosti študentov doplňať v predmetoch zameraných na IKT (Pšenáková-Szabó, 2015; Pšenáková a kol., 2015). Ani po siedmych rokoch platnosti nového školského vzdelávacieho programu nenastali zlepšenia, ktoré sme očakávali. Preto navrhujeme ako riešenie ponúknuť študentom predmet kategórie C, v rámci ktorého by boli ich vstupné vedomosti doplnené na potrebnú úroveň (Czakóová, 2014a). Na základe výsledkov vstupných testov si myslíme, že by bolo vhodné študentom na doplnenie si chýbajúcich poznatkov nariadiť absolvovanie potrebných modulov (Czakóová, 2014b).

Predpokladali sme tiež, že sa zlepšia vstupné poznatky študentov učiteľstva informatiky, hlavne v oblasti základných poznatkov

algoritmizácie a programovania. V tabuľke 5 vidieť, aké programovacie jazyky sa ako dlho učili a či vôbec absolvovali predmet orientovaný na programovanie.

Tabuľka 5. Programovacie prostredia, ktoré používajú, resp. jazyky, ktoré sa učia študenti na stredných školách

AR	PŠE/PPŠ	PJ	#	%
2016/17	21/9	C++	1	10
		Delphi	1	10
		HTML/CSS	5	50
		JavaScript/Ajax	1	10
		MySQL	1	10
		PHP/ASP/CGI	1	10
2015/16	33/15	C	1	3
		C#	1	3
		HTML/CSS	14	48
		Java	1	3
		JavaScript/Ajax	6	21
		PHP/ASP/CGI	5	17
		Scratch	1	3
2014/15	35/18	C	1	6
		HTML/CSS	8	6
		JavaScript/Ajax	5	28
		Pascal	1	6
		PHP/ASP/CGI	3	6

Vysvetlivky: AR – akademický rok, PŠE – počet študentov evidovaných na hodinách programovania v 1. ročníku, PPŠ – počet študentov, ktorí na SŠ absolvovali programovanie ako učivo, PJ – používané programovacie jazyky/prostredia, # – počet výskytov uvedeného jazyka/prostredia, % – percento z celkového počtu uvedených jazykov/prostredí

Vo vyplnených a odovzdaných dotazníkoch 45 % respondentov (študentov učiteľstva informatiky) uviedlo, že programovanie ako učivo na strednej škole neabsolvovali vôbec. Avšak i sebahodnotenie tých študentov, ktorí uviedli, že hodiny programovania absolvovali, bolo nízke. Priemernou známkou, ktorú uvádzali na päťstupňovej škále od 1 (najlepšie) do 5 (najhoršie) bola štvorka.

V dotazníkoch sme študentom položili otázku, koľko hodín odhadujú, že sa s nimi učiteľ na hodinách informatiky venoval téme programovanie. V akademickom roku 2014/15 pri počte 18 študentov tvoril priemer asi 42 hodín za celú strednú školu (čo je v priemere a vzhľadom na nimi uvedený počet absolvovaných hodín informatiky asi štvrtina školského roka; z týchto študentov štyria nepostúpili do druhého ročníka).

V akademickom roku 2015/16 to boli 15 študenti a ich priemerný odhadovaný počet hodín venovaný programovaniu počas celej strednej školy bol 54 (čo je v priemere a vzhľadom na nimi uvedený počet absolvovaných hodín informatiky asi polovica školského roka; z týchto študentov šesť nepostúpilo do druhého ročníka).

Priemerný počet hodín programovania 9 študentov, ktorí v dotazníkoch z akademického roka 2016/17 uviedli, že absolvovali programovanie, bol asi 35 za celú strednú školu (čo je v priemere a vzhľadom na nimi uvedený počet absolvovaných hodín informatiky takmer polovica školského roka; z týchto študentov jeden nepostúpil do nasledujúceho semestra).

Programovaniu sa v učiteľskej príprave informatikov venuje celý rad disciplín. V prvom ročníku štúdia sa predmety algoritmizácie a programovania považujú za najnáročnejšie. Ich úspešné absolvovanie spôsobuje študentom nemalú záťaž i obavy, či budú schopní tieto predmety úspešne absolvovať.

5. ZÁVER

Kvalitný učiteľ je predpokladom nielen kvalitnej prípravy žiakov, ale aj kvality celého vzdelávacieho systému. Nestačí mať len modernými technickými prostriedkami a digitálnymi technológiami dobre vybavené základné, stredné a vysoké školy, ale potrebujeme moderného, dobre pripraveného učiteľa, ktorý okrem predmetných poznatkov, má aj didaktické schopnosti a patričné zručnosti z používania didaktickej techniky a vzdelávacích technológií. Preto na prípravu učiteľov (nielen informatiky) kladieme veľký dôraz. Aj keď o učiteľstvo informatiky zdá sa, že je dostatočný záujem, nie je jednoduché tento záujem udržať a doviezť prípravu nastúpených prvákov učiteľstva informatiky do úspešného konca. Preto nám záleží na každom študentovi, ktorý má pozitívny vzťah k učiteľskému povolaniu a má predpoklady úspešne študovať učiteľstvo informatiky. Venujeme im, hlavne talentovaným študentom mimoriadnu pozornosť, uplatňujeme individuálny prístup, ponúkame možnosť zapojiť sa do výskumu na katedre, podieľať sa na študentskej vedeckej činnosti, ponúkame konzultácie a možnosť doplniť si chýbajúce poznatky hlavne z programovania. Predmety programovania považujeme za kľúčové v študijnom programe učiteľstva informatiky. Nielen talentovaným, ale aj zaostávajúcim študentom ponúkame semináre programovania na rozvoj programátorských poznatkov a zručností, ako aj na doplnenie si chýbajúcich vedomostí.

Od učiteľov – absolventov všetkých študijných programov, nielen v kombinácii s informatikou, sa očakáva, že okrem používania hotových aplikácií a štandardných didaktických programov budú schopní aj sami takéto aplikácie vytvárať (Stoffová, 2005; Végh, 2014). Predpokladá sa tiež, že ich zručnosti v oblasti používania moderných didaktických prostriedkov a digitálnych technológií budú na dobrej úrovni.

Vytvárať didaktické aplikácie je požiadavkou, ktorá vystupuje aj v požadovaných kompetenciách na učiteľa v rámci Európskej únie (Stoffová, 2013). Preto pre všetky učiteľské smery navrhujem do skupiny predmetov B zaradiť predmet orientovaný na tvorbu didaktických aplikácií (v rôznych prostrediach), napr. v mikrosvetoch, príp. IKT vo vyučovaní predmetov.

Prvou takou lastovičkou je na Pedagogickej fakulte Trnavskej univerzity predmet pre študentov anglického jazyka a literatúry „ICT in Language Teaching“ v rozsahu 1 + 2 hodiny v zimnom a 2 + 2 hodiny v letnom semestri. Predmet je zaradený do 1. ročníka magisterského štúdia.

Študenti učiteľstva informatiky majú k dispozícii celý rad predmetov typu A a B orientovaných na získanie teoretických poznatkov aj praktických zručností z tvorby didaktických aplikácií vo

forme učebných materiálov, didaktických hier, elektronických kurzov, webových alebo mobilných aplikácií.

BIBLIOGRAFICKÉ ODKAZY

[1] BLAHO, A.: Informatika v štátnom vzdelávacom programe. In *DidInfo 2012 : Proceeding of abstracts and electronic version of reviewed contributions on CD-ROM.* Editor Gabriela Andrejková. Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, 2012, s. 7 – 14. ISBN 978-80-557-0342-8.

[2] CZAKÓOVÁ, K.: Skúmanie prístupu a postoja budúcich učiteľov k tvorbe vlastnej aplikácie (Researching access and attitudes of future teachers to creating their own applications). In Stoffová, Veronika – Chráska, Miroslav – Klement, Milan – Havelka, Martin (ed.): *Sborník z mezinárodní konference XXVII. DidMatTech 2014.* Proceeding of electronic version of reviewed contributions (CD-ROM). Olomouc : Pedagogická fakulta UP Olomouc, 2014a. s. 15 – 18. ISBN 978-80-86768-88-5.

[3] CZAKÓOVÁ, K.: Programové prostredia na rozvoj tvorivosti. In. Stoffová, Veronika (ed.): *New Technologies in Science and Education.* Győr : University of West Hungary, 2014b. s. 155 – 158. ISBN 978-963-334-184-1.

[4] PŠENÁKOVÁ, I. – SZABÓ, T.: Az m-learning lehetőségei a szlovákiai tanárképzésben In *XXI. Multimédia az oktatásban és II. IKT az oktatásban konferencia.* – Szabadka : Újvidéki Egyetem, 2015, s. 131 – 135. ISBN 978-86-87095-54-0. Dostupné na internete: http://ukftp.truni.sk/epc/12166.pdf. Dátum posledného prístupu: 15. 01. 2017.

[5] PŠENÁKOVÁ, I. a kol.: Okoseszközök felhasználási lehetőségei a felsőoktatásban In: *INFODIDACT 2015, 8. Informatika Szakmódszertani Konferencia.* Editori: Péter Szlávi a László Zsakó, Zamárdi, 2015. (príspevok na CD, 6 s.) ISBN 978-963-12-3892-1

[6] SALANCI, Ľ.: Informatika v novom štátnom vzdelávacom programe. In: *Zborník konferencie DidInfo2014*. Banská Bystrica : Univerzita Mateja Bela, 2014, s. 143-148. ISBN 978-80-557-0698-6

[7] SALANCI, Ľ.: Analýza informatiky v novom štátnom vzdelávacom programe. In: *DidInfo 2015 : 21. ročník národnej konferencie*. Editor Ľudovít Trajteľ. Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, 2015, s. 134-138. ISBN 978-80-557-0852-2

[8] STOFFOVÁ, V: Trojúrovňový model prípravy učiteľov v oblasti informačných a komunikačných technológií. In: *XVIII. DIDMATTECH 2005*. Prešov : Prešovská univerzita v Prešove, 2005, príspevok na CD.

[9] STOFFOVÁ, V.: Ideálny učiteľ v predstavách budúcich učiteľov informatiky. In: Trajteľ, Ľ. (ed.): *DidInfo 2013 : 19. ročník národnej konferencie o vyučovaní informatiky*. Editor Ľudovít Trajteľ. Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, 2013, s. 222-228. ISBN 978-80-557-0527-9

[10] VÉGH, L.: Methods of Creating Educational 3D Animation Models in Virtual Worlds. In: Stoffová, V. (ed.): *New technologies in science and education. XXVI. DIDMATTECH 2013.* Győr : University of West Hungary, 2014, s. 61-65. ISBN 978-963-334-184-1

[11] *Poradné orgány. Ústredné predmetové komisie.* Bratislava : Štátny pedagogický ústav Ministerstva školstva, vedy, výskumu a športu Slovenskej republiky. Dostupné na internete: http://www.statpedu.sk/clanky/organizacia-spu/poradne-organy, http://www.statpedu.sk/clanky/organizacia--spu-poradne-organy/ustredne-predmetove-komisie. Dátum posledného prístupu: 15. 01. 2017.

[12] *Inovovaný štátny vzdelávací program*. Bratislava : Štátny pedagogický ústav Ministerstva školstva, vedy, výskumu a športu Slovenskej republiky. Dostupné na internete: http://www.statpedu.sk/clanky/inovovany-statny-vzdelavaci--program. Dátum posledného prístupu: 15. 01. 2017.

[13] PORUBSKÁ, G. – ŠNÍDLOVÁ, M. – VALICA, M.: Príloha čísla 4/2008 – Návrh profesijných štandardov učiteľov – učiteľ nižšieho a vyššieho sekundárneho vzdelávania (diskusia). In *Pedagogické rozhľady – Časopis pre školy a školské zariadenia.* Bratislava : Metodicko-pedagogické centrum, 2008. ISSN 1335-0404. Dostupné na internete: http://www.rozhlady.pedagog.sk/cisla/p4-2008.pdf. Dátum posledného prístupu: 15. 01. 2017.

Využití zkušenostního učení při výuce e-bezpečnosti
The use of experiential learning in e-safety lessons

Václav Šimandl
KIN PF JU
Jeronýmova 10
371 15 České Budějovice
Česká republika
simandl@pf.jcu.cz

Václav Dobiáš
KIPV VŠTE
Okružní 517
370 01 České Budějovice
Česká republika
dobias@mail.vstecb.cz

ABSTRACT
This article deals with the use of experiential learning to teach e-safety. We proposed and tested a lesson in which learners were confronted with a negative experience. The learners should learn a lesson from this experience. The teaching was based on experience situations which simulated e-safety threats and which the learners should cope with. If the learners did not reveal soon enough that the situation is an e-safety attack (even though a simulated one), they would go through a negative experience. In the article, we describe not only the way how the lesson and the experience situations were implemented, but also ethical questions regarding this teaching. Moreover, we discuss the findings from pilot lesson application which was carried out on university students.

Keywords
E-safety. Experiential learning. Negative experience. Simulation. Phishing.

ABSTRAKT
Článek se zabývá možnostmi využití zkušenostního učení při výuce e-bezpečnosti. Navrhli a realizovali jsme výuku, během níž byl studentům zprostředkován určitý negativní zážitek, z nějž se studenti měli poučit. Základem výuky byly námi připravené zážitkové situace, simulující e-bezpečnostní hrozby, s nimiž se studenti měli vyrovnat. Jestliže studenti včas neodhalili, že se jedná o e-bezpečnostní útok (byť pouze simulovaný), zažili určitou negativní zkušenost. V rámci článku popisujeme nejen způsob realizace výuky a zážitkových situací, ale i etické otázky týkající se takové výuky. Diskutujeme též zjištění z pilotního ověření výuky, které jsme realizovali na skupině vysokoškolských studentů.

Klíčová slova
E-bezpečnost. Zkušenostní učení. Negativní zkušenost. Simulace. Phishing.

1. ÚVOD
E-bezpečnost se zabývá ochranou uživatele a jím používaných digitálních technologií před negativními jevy, které se vyskytují při využívání digitálních technologií [1]. Jelikož je toto vymezení poměrně rozsáhlé a patří sem celá řada rizik, v tomto článku se omezíme na tu část e-bezpečnosti, která se zabývá interakcí uživatele s technikou. Řadíme sem problematiku malware a havárií počítačů, oblast počítačových hesel a s nimi spojených krádeží identity, negativní jevy při e-mailové komunikaci (spam, hoax, phishing) a sdílení osobních dat.

Kompetence dětí a mládeže v oblasti e-bezpečnosti se staly předmětem zájmu mnoha výzkumů. Podle Cranmer et al. jsou strategie, jak se chránit před nebezpečím, u žáků prvního stupně základních škol vzhledem k věku málo propracované a rizika špatně pochopená [2], žákům druhého stupně chybí schopnost chápat a kriticky hodnotit online obsah a řídit své chování online [3]. Žáci středních škol si jsou vědomi reálnosti e-bezpečnostních hrozeb, ale chybí jim základní e-bezpečnostní pravidla a nechtějí akceptovat „pravidla dospělých“ [4]. Toto zřejmě souvisí s názorem, že v problematice e-bezpečnosti je časté přesvědčení uživatelů, že jim se problémy zcela jistě vyhnou [5] a že není potřeba se e-bezpečnostními pravidly řídit [6].

Jak uvádí Aytes and Conolly, snížení míry riskování při používání počítačů je možné díky zlepšení povědomí o této problematice a vhodnému tréninku [7]. Podle Byron nelze spoléhat, že edukaci dětí v problematice e-bezpečnosti zajistí sami rodiče [3]. Nejlepším místem, kde učit děti digitálním dovednostem potřebným k maximalizaci příležitostí a minimalizaci rizik, je podle Livingstone a Haddon škola [8]. Ta by se měla zaměřit na edukaci k bezpečnosti a odpovědnosti při užívání digitálních technologií [3], [9] a měla by nést hlavní zodpovědnost za vedení žáků ke kritickému myšlení a vhodnému chování, které je bude chránit před riziky při používání Internetu [10, 11]. Žáci potřebují pomoc školy též v rozpoznávání a eliminaci nebezpečí a tvorbě odolnosti vůči němu [12].

Abychom přispěli k diskuzi o vhodných formách výuky e-bezpečnosti, rozhodli jsme se navrhnout a realizovat vlastní přístup k výuce této oblasti. Cílem této výuky není působit pouze na znalosti, ale i na postoje učících se osob. V následujícím textu popíšeme výuku, která vychází ze zkušenostního učení podle Kolba [13]. Tato výuka byla součástí rozsáhlejšího výzkumu, do kterého bylo zařazeno vícero výukových přístupů a, jehož cílem bylo ověřit vliv různých výukových přístupů na e-bezpečnostní postoje studentů. Tímto výzkumem se blíže zabýváme v [14].

2. NÁVRH VÝUKY
Návrh naší výuky vychází z Kolbova zkušenostního učení [13]. Kolb se domnívá, že znalosti nejsou nezávislou entitou, která může být získána či předána, ale že učení je proces, během něhož jsou znalosti tvořeny transformací zkušenosti. Právě zkušenost tak má v tomto konceptu centrální roli [13].

Jelikož má na chování jedince v oblasti e-bezpečnosti značný vliv předchozí negativní zkušenost [6], [15], rozhodli jsme se realizovat výuku založenou právě na tomto jevu. V návrhu naší výuky proto Kolbovu zkušenost představuje určitý negativní zážitek, který by se studenta měl přímo dotýkat. Tento negativní zážitek je studentovi zprostředkován vyučujícím a dle našeho předpokladu by měl mít vliv na zlepšení chování jedince v problematice e-bezpečnosti.

Návrh výuky očekává, že student v rámci výuky podlehne předem připraveným e-bezpečnostním hrozbám, načež prožije určitou negativní zkušenost. Zatímco v případě skutečného e-bezpečnostního incidentu je jeho účinek umocněn jeho dlouhodobými následky [6], v případě námi připravených hrozeb jsme naopak dbali na vyloučení dlouhodobých následků, které by mohly ohrozit uživatele, jeho data či jím používané digitální technologie. Podle Kolbova modelu zkušenostního učení má po konkrétní zkušenosti následovat ohlédnutí a reflexe [13]. V rámci výuky jsme tak dbali nejen na zprostředkování negativního zážitku, ale také na následnou analýzu prožité situace a jejích skutečných i potenciálních důsledků.

2.1 Návrh zážitkových situací

Připravili jsme tři zážitkové situace, které simulují e-bezpečnostní hrozby, s nimiž se mají studenti vyrovnat. Všechny situace jsme vytvořili tak, aby student měl možnost si všimnout určitých rizikových znaků, na které by zareagoval a nebezpečí se tak vyhnul. Tento požadavek považujeme za důležitý nejen z etického hlediska, ale i proto, aby si student uvědomil možnost volby a přijal určitou míru zodpovědnosti za to, že hrozbě podlehl. V následujícím textu popíšeme jednotlivé připravené situace.

2.1.1 Situace: Riziková registrace

Obrázek 1. Úvodní stránka námi vytvořeného webu Pirate VŠTE

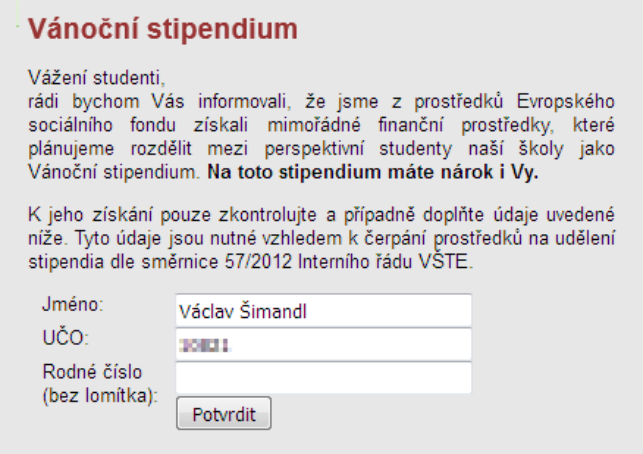
Vytvořili jsme web nabízející materiály pro usnadnění studia (například poznámky z hodin či testy z uplynulých let). Web je svým názvem Pirate VŠTE, vzhledem (viz Obrázek 1) i povahou materiálů zaměřen přímo na studenty, na nichž jsme realizovali pilotní ověření výuky. Zatímco několik materiálů je možno získat přímo, pro zpřístupnění dalších sekcí webu se má návštěvník registrovat. Během registrace uživatel zadá svou e-mailovou adresu, zvolí nějaké heslo a odsouhlasí Podmínky používání webu. Pro dokončení registrace má být zaslána na uvedenou e-mailovou adresu aktivační e-mailová zpráva. Ta však již zasílána není a registraci tak není možno dokončit. Data uvedená při registraci se ukládají do databáze – e-mailová adresa v čitelné podobě, heslo v šifrované podobě.

Studentům je několik dnů před plánovanou výukou rozeslána jménem fiktivního studenta e-mailová zpráva, ve které jsou informováni, že existuje web obsahující materiály pro usnadnění studia na škole. Ve zprávě je uveden odkaz na tento náš web.

Student si při čtení zaslané e-mailové zprávy a prohlížení webu může všimnout následujících rizikových znaků:

- Odesilatel e-mailu. Jako jméno odesilatele e-mailu je použito jméno neexistujícího studenta. Příjemce zprávy si v informačním systému školy může ověřit, zda daný student existuje. Určité pochybnosti může budit také e-mailová adresa odesilatele borec.novak@email.cz.
- Podmínky používání webu. V nich je uvedeno, že uživatel si je vědom, že na webu nejsou po registraci dostupné žádné další materiály k výuce. Tyto Podmínky musí během registrace každý uživatel odsouhlasit.

2.1.2 Situace: Nebezpečné uvádění osobních údajů



Obrázek 2. Výzva k zadání rodného čísla

Vytvořili jsme web, který svým vzhledem napodobuje školní informační systém IS. Studentovi se po přihlášení na web zobrazí stránka, na které je informován, že je mu nabízeno Vánoční stipendium, pro jehož získání je potřeba zadat rodné číslo (viz Obrázek 2).

Jestliže student zadá platné rodné číslo, zobrazí se hlášení o úspěšném provedení akce. Po několika vteřinách však je automaticky zobrazena další stránka, na které studenta varujeme, že navštívená stránka je pravděpodobně podvodná (viz Obrázek 3).

Z technického hlediska je celá situace vytvořena tak, že v počítačové učebně, kde má proběhnout tento typ výuky, jsou upraveny konfigurace počítačů, aby po zadání URL adresy informačního systému školy byl uživatel přesměrován na námi vytvořený web. Zde se přihlásí pomocí přihlašovacích údajů používaných standardním informačním systémem. Ze zadaných údajů ověřujeme pouze uživatelské jméno, s uživatelským heslem nijak nepracujeme (tedy jej ani neověřujeme, ani např. neukládáme). Uživatelské jméno je ověřováno proti námi

vytvořené databázi studentů, kteří se mají výuky účastnit. Tím je zamezeno, aby se k webu přihlásil jakýkoliv jiný uživatel. Jestliže student vyplní po přihlášení pole s rodným číslem, platnost tohoto rodného čísla je ověřena na straně klienta; na straně serveru není s tímto údajem nijak manipulováno.

Obrázek 3. Námi vytvořená stránka napodobující varování před podvodným webem

Student si při práci s webem může všimnout několika rizikových znaků:

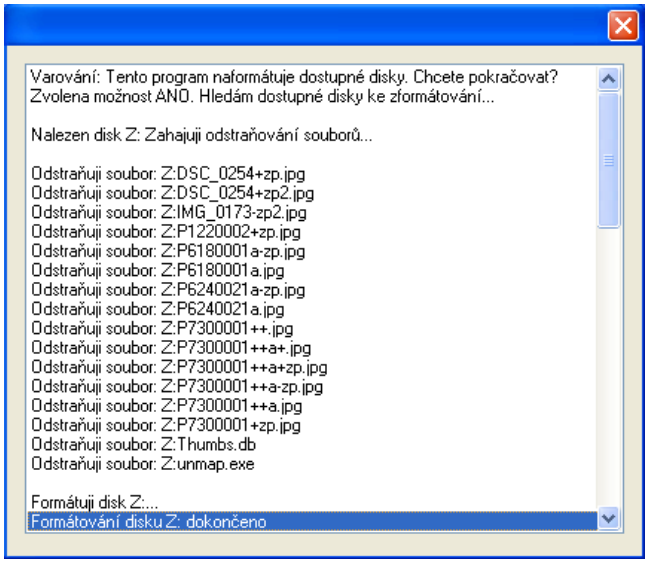
- Nezabezpečené připojení. Zatímco přihlašovací stránka standardního informačního systému je zabezpečená a je využíván protokol https (uživateli se v adresním řádku prohlížeče zobrazí před URL adresou název protokolu a symbol zámku), námi vytvořená stránka používá protokol http (kde název protokolu a symbol zámku chybí).
- Přidělování peněz prostřednictvím avizovaného stipendia. Nabízí se otázka, proč by běžný student bez jakýchkoliv aktivit nad rámec svých studijních povinností a bez nadprůměrného prospěchu měl získat mimořádné stipendium. O získání stipendia by byl student pravděpodobně vyrozuměn písemně či alespoň e-mailem. Student si může ověřit, zda směrnice uvedená v textu existuje (tato směrnice pochopitelně neexistuje) a jaké je její případné znění.
- Požadavek na zadání rodného čísla. Rodné číslo škola od studenta získala již při zápisu do studia, je patrně zaneseno ve studentově dokumentaci, a není tak objektivní důvod jej požadovat znovu.

2.1.3 Situace: Nebezpečná instalace

Vytvořili jsme web, který svým vzhledem napodobuje úvodní stránky sociální sítě Facebook. Studentovi se po přihlášení na web zobrazí stránka s informací, že pro plnohodnotné zobrazení dalšího obsahu je nutné nainstalovat určitý doplněk prohlížeče (viz Obrázek 4).

Obrázek 4. Informace o nutnosti nainstalovat doplněk prohlížeče

Jestliže student bude na výzvu reagovat, soubor si stáhne a spustí, aplikace bude simulovat odstranění veškerých souborů a složek z jeho síťového domovského disku Z: a následné zformátování tohoto disku (viz Obrázek 5).



Obrázek 5. Aplikace simulující odstranění veškerých souborů ze síťového domovského disku uživatele

Z technického hlediska je celá situace vytvořena tak, že v počítačové učebně, kde má proběhnout tento typ výuky, jsou upraveny konfigurace počítačů, aby po zadání URL adresy facebook.com byl uživatel přesměrován na námi vytvořený web. Zde se přihlásí pomocí přihlašovacích údajů používaných standardním portálem Facebooku. Se zadanými údaji nijak nepracujeme (tedy je ani neověřujeme, ani např. neukládáme).

Samotná aplikace simulující odstranění souborů z disku uživatele je vytvořena tak, že nejprve načte seznam všech souborů uložených na daném disku a disk softwarově odpojí. Poté začne vypisovat informace o odstraňování souborů a formátování disku. Aplikace po dobu svého běhu vykazuje určité časové prodlevy, aby měl student možnost si přečíst informace zobrazené aplikací.

Aplikace však běží na uživateli zcela nezávisle, tj. není možné její běh nijak ovlivnit (či ji ukončit jinak než přes Správce úloh).

Student si při práci s webem může všimnout několika rizikových znaků:

- Nezabezpečené připojení. Zatímco standardní portál Facebook je zabezpečený a je využíván protokol https (uživateli se v adresním řádku prohlížeče zobrazí před URL adresou název protokolu a symbol zámku), námi vytvořený web používá protokol http (kde název protokolu a symbol zámku chybí).
- Odmítnutí zobrazení webu. Běžné webové aplikace dovolují zobrazit většinu prvků webových stránek, i když pro zobrazení některých prvků webové stránky chybí určitý doplněk webového prohlížeče. Obvykle tak nejsou zobrazeny pouze prvky vyžadující chybějící doplněk prohlížeče. Naše stránka se však odmítne zobrazit zcela a uživatel je tak nucen doplněk prohlížeče nainstalovat, ač neví, k čemu má sloužit.
- Neznámý doplněk prohlížeče. Webová stránka požaduje instalaci neznámého doplňku prohlížeče, který uživatel nezná a o kterém nelze na Internetu nalézt žádné informace (např. k čemu slouží, kdo je vydavatelem).
- Neznámý vydavatel aplikace. Stahovaná aplikace není (na rozdíl od většiny důvěryhodných aplikací) digitálně podepsaná vydavatelem, a tudíž nelze identitu vydavatele ověřit. Některé prohlížeče před takovou aplikací varují jako před potenciálně nebezpečnou.

Tato zážitková situace zůstává pouze v testovací verzi a ve výuce nebyla doposud použita, neboť jejímu praktickému nasazení brání několik aspektů. Z etického hlediska se domníváme, že není vhodné napodobovat webové služby třetích stran (v tomto případě Facebook) a vystavovat studenty nejistotě, zda jsme nezískali jejich soukromé přihlašovací údaje k této službě. Z technického hlediska je problematické přesměrovat studenty na námi podvržený web napodobující úvodní stránku Facebooku, neboť se při komunikaci se serverem facebook.com je používán protokol https a nikoliv http. Internetový prohlížeč tak při přesměrování na námi vytvořený web pozná, že server nepoužívá odpovídající SSL certifikáty a označí web jako nedůvěryhodný resp. nebezpečný.

2.2 Návrh vyučování

Vyučování je koncipováno jako seminář o délce 90 minut, který může být realizovaný například na střední či vysoké škole v rámci výuky základů ICT. V následujícím textu nastíníme průběh takového semináře; důraz je kladen na začlenění výše popsaných zážitkových situací do výuky.

Ještě před samotným začátkem semináře (tj během „přestávky“) měla být použita zážitková situace *Nebezpečná instalace*. Bylo předpokládáno, že studenti se budou před začátkem semináře individuálně přihlašovat k sociální síti Facebook. Po zadání adekvátní URL adresy by však byli přesměrováni na námi vytvořený web, odkud by si někteří z nich stáhli aplikaci simulující odstranění souborů z disku. Po začátku semináře by učitel studenty vyzval, aby si otevřeli soubor, který mají rozpracovaný z minulého semináře. Jelikož by byl tento soubor pro některé studenty nedostupný a nemohli by s ním pracovat, učitel by na vzniklou situaci reagoval a podpořil myšlenku, že soubory byly smazány působením malware. Současně by učitel zjišťoval aktivitu studentů, která k této situaci vedla. Posléze by však studenty informoval, že nešlo o útok malware, ale o simulaci, při níž žádné jejich soubory nebyly poškozeny či zničeny.

Následovala by reflexe prožité situace, při níž by učitel vysvětlil princip útoku malware a společně se studenty by na příkladu prožité situace definoval typické způsoby, jak může k nákaze malware dojít. Na tuto fázi výuky by navázal výklad učitele o dalších e-bezpečnostních tématech, zejména o tématu zálohování dat. Jelikož však z technických důvodů nebylo možné zážitkovou situaci *Nebezpečná instalace* ve výuce používat, výše popsaná část výuky zůstává pouze ve stádiu návrhu. Seminář tak začíná až situací *Nebezpečné uvádění* osobních údajů, jak je popsáno v následujícím textu.

V úvodu semináře je použita zážitková situace *Nebezpečné uvádění osobních údajů*, kdy učitel studenty požádá, aby se přihlásili do školního informačního systému. Studenti se na základě upravené konfigurace počítačů v učebně přihlásí k námi vytvořenému webu, kde někteří z nich zadají své rodné číslo (čímž podlehnou této hrozbě). Poté jim bude zobrazeno námi vytvořené varování před podvodnou stránkou. Učitel na vzniklou situaci reaguje, ptá se studentů na jejich činnost vedoucí k zobrazení tohoto varování a podpoří domněnku, že se jedná o phishingový útok, kdy útočníci mohli získat jejich rodná čísla. Posléze však studentům oznámí, že se nejedná o reálný phishingový útok, ale o simulaci, při níž žádná jejich osobní a citlivá data (zejména rodná čísla) nejsou zneužita. Následně učitel představí výrazně nebezpečnější případy phishingu, na jejichž příkladu a na základě prožité situace společně se studenty definuje znaky, které by je od zadání požadovaných dat měly odradit (tyto znaky diskutujeme výše). Na tuto fázi výuky naváže výklad učitele o dalších e-bezpečnostních tématech.

Výklad pokračuje až k oblasti počítačových hesel, kdy je použita zážitková situace *Riziková registrace*. Učitel nejprve povzbudí studenty k definování několika způsobů, jak by jiná osoba mohla odhalit jejich hesla. Poté studentům zobrazí web Pirate VŠTE a stručně připomene, že jim patrně přišla e-mailová zpráva informující o tomto webu a že se někteří z nich na webu registrovali. Následně zobrazí tabulku zadaných e-mailových adres a zašifrovaných hesel (viz Obrázek 6).

Studentům vysvětlí, že tyto údaje byly získány na základě registrace na webu (tedy na základě podlehnutí této hrozbě), a oznámí jim, že zadaná hesla byla sice zašifrovaná (a nelze je dešifrovat), ale že by tato hesla mohla být sbírána v původní podobě. Poučí je, že by tato hesla mohla být zneužita k útoku mířenému proti jejich osobě, a uvede konkrétní příklady, k jakým účelům by tato hesla mohla být útočníkem použita. Následuje společná reflexe a nástin zásad chování, aby se potenciální útočníci nemohli k důležitým heslům uživatele dostat. Poté opět následuje výklad učitele o e-bezpečnosti.

Při probírání problematiky online služeb se učitel zaměří na souhlas uživatele s podmínkami využívání služby, kdy využije zážitkové situace *Riziková registrace*. Vrátí se přitom k tabulce z Obrázku 6, kdy oznámí, že studenti uvedení v tabulce se sběrem údajů souhlasili, neboť v rámci Podmínek používání webu Pirate VŠTE je toto uvedeno. Upozorní je, že souhlas s těmito Podmínkami studenti vyjádřili při registraci na webu nehledě na to, zda si znění Podmínek používání webu přečetli. Učitel přitom poukáže na poslední sloupec zobrazené tabulky, kde je zapsáno,

kdo si stránku s Podmínkami zobrazil (hodnota 1) a kdo nikoliv (hodnota 0). Poté až do konce hodiny opět následuje výklad.

email	heslo	precetl
wertykal@	06a0545f07d42e3da845d51ada372	1
ss-h@	29f99bdeb080f4d3f0aecfdd8f81a	0
S.Hoblinka@	36aa9b12a485fcc15f549371c4812	0
Standa009@	3771eda218a9073757b9a00591acd	0
Filikarow@	3aaadec6fc32275022b1339250b07	0
Filokarow@	3aaadec6fc32275022b1339250b07	0
vincekzbytecnosti@	5350a9323a313b6429d5046512243	1
myska44444@	7d68d78fb3b0a5d0405e132cfc1ca	0
xoilx@	8ec0ac678087d11fa272149a67b78	0
certicek007@	8f72877cb586d33e90d0ab95bc170	0
poky007@	93d46433e28ae5321567714699e0b	1
17145@	983a42f2c5ae582b292a67e380875	0
Londoran@	9d0e7e5ed3d362fb7add2add2aee5	0
sramcice@	ab00902c9c97d042a97d93a4f6ba8	0
Petr1302@	ace4b3556124d1275075206031771	0
j.a.n.a.1.4@	ad3f0a7a897a51b90b1db1d96b184	0
pinzezna@	b33eb36429c1e4b4361f740342199	0

Obrázek 6. Tabulka zadaných e-mailových adres a hashů hesel získaných z pilotního ověření výuky

2.3 Etické otázky

Při návrhu situací simulujících e-bezpečnostní hrozby i při návrhu vyučování jsme se zabývali etickými otázkami vytvářeného řešení. Jsme si vědomi, že do určité míry dojde k oklamání studentů, kteří se tohoto typu výuky zúčastní a stanou se „oběťmi" připravených hrozeb. Naším cílem nicméně není studenty poškodit či dokonce se na nich obohatit, ale sledujeme pouze pedagogické a výzkumné cíle.

Se studenty je během výuky realizován debriefing zaměřený na analýzu prožitých situací simulujících e-bezpečnostní hrozby. Studenti jsou informováni, že se jedná o simulace reálných útoků a jsou ujištěni, že žádná jimi zadaná data nebyla, nejsou a nebudou zneužita. Dále jim je ponechán prostor k vyjádření či dotazům nad proběhlou situací.

Všechny simulované hrozby jsou připraveny tak, aby při jakémkoli způsobu práce s vytvořenými weby nebyla nijak ohrožena citlivá data uživatelů.

Do připravených situací jsme se snažili zanést prvky, které mají studenty upozornit, že se jedná o pokus o e-bezpečnostní útok (viz popis rizikových znaků jednotlivých situací).

V případě webu Pirate VŠTE jsou studenti v rámci Podmínek používání webu informováni, že na webu po registraci nebudou dostupné žádné další materiály a provozovatelé webu mohou sbírat informace o uživatelích webu. Tyto Podmínky uživatelé odsouhlasí během registrace; bez jejich odsouhlasení není možné registraci dokončit.

3. PILOTNÍ OVĚŘENÍ

Pilotní ověření výuky proběhlo na Vysoké škole technické a ekonomické v Českých Budějovicích na studentech prvního ročníku studia. Abychom eliminovali vliv osoby učitele, byla veškerá výuka vedena běžným vyučujícím předmětu, kterým nebyl sám výzkumník.

Výuky se zúčastnilo 58 studentů v několika seminárních skupinách. Hrozbě *Nebezpečného uvádění osobních údajů* podlehlo 26 studentů (do tohoto počtu zahrnujeme ty studenty, kteří se přihlásili na náš podvržený web, zadali zde své ročné číslo a byla jim zobrazena stránka s varováním před phishingem). Přesný počet studentů, kteří podlehli hrozbě *Rizikové registrace* (tj. se registrovali na webu Pirate VŠTE) a zároveň se zúčastnili výuky, nelze určit. Důvodem je, že jsme nechtěli automaticky přiřazovat e-mailové adresy zadané do registračního formuláře ke konkrétním studentům. Studentů, které se nám podařilo zpětně identifikovat (například se přihlásili na naši výzvu, kdo se na webu Pirate VŠTE registroval a zúčastnil se výuky), bylo 15.

Abychom získali na výuku pohled nezávislý na učiteli či výzkumníkovi a zajistili tak určitou zpětnou vazbu, pozvali jsme do vybrané seminární skupiny na hospitaci nezávislého pedagoga. Jeho úkolem bylo pozorovat dění během výuky a vytvořit o něm zápis. Tento pedagog zůstal pro studenty v utajení – vcházel do učebny společně se studenty, vyučující na něj studenty nijak neupozorňoval a je tak pravděpodobné, že si jeho přítomnosti studenti nevšimli či jej (i vzhledem k jeho věku) považovali za svého spolužáka.

Hospitační zápis si po skončení veškeré výuky kromě samotného výzkumníka prostudoval i vyučující, kterého jsme požádali o vyjádření k proběhlé výuce z jeho pohledu. V následujícím textu popíšeme průběh výuky v hospitované hodině. Vycházíme přitom z hospitačního zápisu a z výpovědi samotného učitele.

3.1 Průběh výuky

Studenti se v úvodu výuky podle pokynů učitele chtěli přihlásit do školního informačního systému, avšak byli přesměrováni na náš podvržený web, kde někteří z nich zadali své rodné číslo. Na stránku s varováním, že se stali obětí phishingu, reagovali slovy: „dnes IS nefunguje", „měl jsem radost, že jsem dostal stipendium" nebo „něco se stalo, nevím co". Následnou informací, že stránka byla podvržena, učitel konfrontoval studenty s vážností celé situace. Pro prohloubení prožitku se učitel snažil studentům předestřít možné negativní důsledky jejich jednání. Po krátké reflexi zážitku učitel začal svůj připravený výklad o možném zneužití dat, studenti byli zaujati.

Učitel prokládal výklad mnohými příklady z praxe a audiovizuálními materiály. Přesto během výkladu pozornost postupně upadala, výrazné zlepšení nastalo při promítnutí tabulky e-mailových adres a zašifrovaných hesel získaných z databáze webu Pirate VŠTE. Ačkoliv všichni studenti velmi pečlivě studovali seznam zobrazených emailových adres, nikdo ze studentů se nepřiznal (veřejně ani svým spolužákům), že hrozbě podlehl a je v seznamu uveden. I v tomto případě se učitel snažil zvýšit prožitek studentů, k čemuž využil popis možných způsobů zneužití získaných emailů a hesel, pokud by se tyto údaje dostaly do rukou hackerů.

V průběhu následující části hodiny pozornost studentů kolísala, zvýšila se při změně stylu výuky (např. při promítnutí videa,

zobrazení grafu nebo aktivizačních dotazech učitele), naopak při výkladu bez interaktivních prvků klesala. Sám učitel hospitovanou výuku hodnotil negativněji než výuku v ostatních seminárních skupinách, což přičítal pasivnímu naladění studentů této seminární skupiny.

4. ZÁVĚR

Navrhli a realizovali jsme výuku e-bezpečnosti, která byla založena na principech zkušenostního učení. Základem této výuky jsou námi připravené zážitkové situace, simulující e-bezpečnostní hrozby, s nimiž se studenti mají vyrovnat. Oproti klasické výuce, kdy učitel v rámci svého výkladu o možných rizicích v oblasti e-bezpečnosti hovoří na teoretické úrovni, jsou studenti v rámci naší výuky přímo vtaženi do dané situace. Mohou si tak lépe uvědomit, že se hrozby spojené s využíváním ICT týkají i jich a že i oni se mohou stát obětí e-bezpečnostního útoku.

Významným problémem, který je nutno mít při takto koncipované výuce na zřeteli, je etické a psychologické hledisko. Je zde totiž zřejmé dilema mezi snahou zprostředkovat studentům poučení z určitého negativního zážitku a požadavkem studenty nikterak nepoškodit. Dalším problémem je zvýšená technická náročnost přípravy takovéto výuky, která klade nemalé požadavky na učitelovy informatické (a především programátorské) kompetence. Z těchto důvodů se zdá být vytváření vhodných situací, které by simulovaly určité e-bezpečnostní hrozby, poměrně obtížné.

BIBLIOGRAFICKÉ ODKAZY

[1] BARROW, Ch., HEYWOOD-EVERETT, G. *E-safety: the experience in English educational establishments* [online]. Coventry : Becta, 2006 [cit. 2012-07-15]. http://dera.ioe.ac.uk/1619/1/becta_2005_esafetyaudit_report.pdf

[2] CRANMER, S., POTTER, J., SELWYN, N. *Learners and technology: 7–11* [online]. Coventry : Becta, 2008 [cit. 2012-04-16]. http://dera.ioe.ac.uk/1630/7/becta_2008_learners7to11_report_Redacted.pdf

[3] BYRON, T. *Safer Children in a Digital World :The Report of the Byron Review* [online]. Department for Children, Schools and Families, 2008 [cit. 2012-03-22]. ISBN 978-1-84775-134-8. http://webarchive.nationalarchives.gov.uk/20130401151715/http://www.education.gov.uk/publications/eOrderingDownload/DCSF-00334-2008.pdf

[4] BERÁNEK, L. Information systems security education for future teacher at secondary and primary schools. In *Journal of Technology and Information Education* [online]. 2009, roč. 1, č. 2 [cit. 2012-08-15], s. 89-93. ISSN 1803-537X. http://jtie.upol.cz/pdfs/jti/2009/02/16.pdf

[5] LANG, M. et al. Social Networking and Personal Data Security: A Study of Attitudes and Public Awareness in Ireland. In *Proceedings of International Conference on Management of e-Commerce and e-Government (ICMeCG), Nanchang, China, September 16-19*. IEEE Computer Society, 2009, s. 486–489.

[6] ŠIMANDL, V. ICT Teachers and Technical E-safety: Knowledge and Routines [online]. In *International Journal of Information and Communication Technologies in Education*. 2015, roč. 4, č. 2 [cit. 2016-05-12], s. 50-65. ISSN 1805-3726. https://periodicals.osu.eu/ictejournal/dokumenty/2015-02/ictejournal-2015-2.pdf

[7] AYTES, K., CONNOLLY, T. Computer security and risky computing practices: A rational choice perspective. In *Journal of Organizational and End User Computing*. 2004, roč. 16, č. 3, s. 22-40.

[8] LIVINGSTONE, S., HADDON, L. *EU Kids Online: Final Report* [online]. London : EU Kids Online, 2009 [cit. 2011-10-24]. http://www.lse.ac.uk/media@lse/research/EUKidsOnline/EU%20Kids%20I%20%282006-9%29/EU%20Kids%20Online%20I%20Reports/EUKidsOnlineFinalReport.pdf

[9] Evropská komise. *Digitální agenda: Nová strategie pro zajištění větší bezpečnosti a lepšího obsahu pro děti a mládež na internetu* [online]. 2012 [cit. 2016-03-15]. http://europa.eu/rapid/press-release_IP-12-445_cs.pdf

[10] Becta. *E-safety : Developing whole-school policies to support effective practice* [online]. Coventry : Becta, 2005 [cit. 2013-02-02]. http://www.wisekids.org.uk/BECTA%20Publications/esafety.pdf

[11] Becta. *Signposts to safety : Teaching e-safety at Key Stages 3 and 4* [online]. Coventry : Becta, 2007 [cit. 2011-11-30]. https://www.education.gov.uk/publications/eOrderingDownload/signposts_safety_ks3and4.pdf

[12] South West Grid for Learning. *School Online Safety Template Policy* [online]. Exeter : South West Grid For Learning, 2009 [cit. 2013-03-13]. http://swgfl.org.uk/products-services/esafety/resources/creating-an-esafety-policy/Content/ESP_template-School-Template-Policy.aspx

[13] KOLB, D.A. *Experiential learning: experience as the source of learning and development* [online]. Englewood Cliffs, NJ : Prentice Hall, 1984 [cit. 2017-01-10]. https://pdfs.semanticscholar.org/f6ec/20a7a3150822140be7466353d0de572cc4bb.pdf

[14] ŠIMANDL, V. *Kompetence učitelů ICT v oblasti technické e-bezpečnosti*. České Budějovice, 2016. Dostupné též na: http://theses.cz/id/1ndn00/simandl-dp9b5.pdf. Disertační práce. Jihočeská univerzita v Českých Budějovicích, Pedagogická fakulta. Vedoucí práce Jiří Vaníček.

[15] VELŠIC, M. *Digitálna gramotnosť na Slovensku 2013* [online]. Bratislava : Inštitút pre verejné otázky, 2013 [cit. 2017-01-09]. ISBN 978-80-89345-40-3. http://www.ivo.sk/buxus/docs//publikacie/subory/Digitalna_gramotnost_2013.pdf

Jazykovo závislé úlohy súťaže iBobor
Language-dependent tasks in the Bebras contest

Monika Tomcsányiová
Katedra základov a vyučovania informatiky, FMFI UK v Bratislave
Mlynská dolina
842 48 Bratislava
Slovensko
tomcsanyiova@fmph.uniba.sk

ABSTRACT
The tasks for the Bebras competition (Informatics Beaver, iBobor in Slovakia) are prepared annually by experts in the field of informatics from more than thirty countries. Each country then chooses certain tasks and adapts them for their national competition. In the paper we describe the problems the authors and translators encounter. We focus on tasks that are language dependent. In this case it is necessary not only translate the task, but also the data contained in the task must be localized into Slovak. We show that the adaptation of these tasks can be almost as challenging and creative as the design and preparation of a new task.

Keywords
informatics, competition, Bebras, tasks, localization

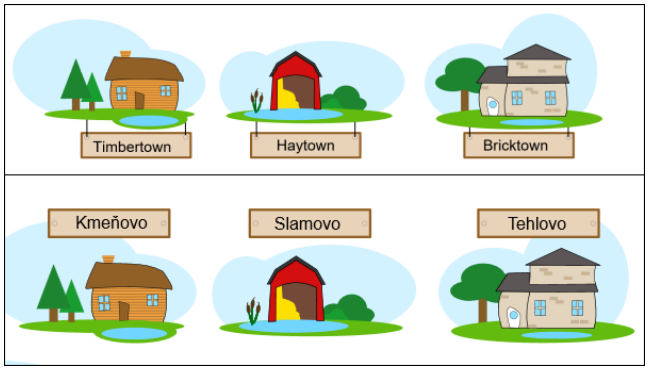
ABSTRAKT
Zadania úloh súťaže Informatický bobor každoročne pripravujú odborníci z oblasti informatiky z viac ako tridsiatich krajín. Každá krajina si potom niektoré úlohy vyberie a tie adaptuje pre svoju národnú súťaž. V článku popisujeme, s akými problémami sa stretávajú autori a prekladatelia zadaní úloh, pričom sa sústredíme na také úlohy, ktoré sú jazykovo závislé. Zadanie úlohy je v takomto prípade potrebné nielen preložiť, ale údaje, ktoré úloha obsahuje, je nevyhnutné lokalizovať do slovenčiny. Ukážeme, že adaptácia takýchto úloh môže byť takmer rovnako náročná a kreatívna ako návrh a príprava nového zadania súťažnej úlohy.

Kľúčové slová
informatika, súťaž, iBobor, úlohy, lokalizácia

1. ÚVOD
V školskom roku 2016/17 sa uskutočnil už 10. ročník súťaže Informatický bobor, pozri [1]. Najväčšou zmenou, ktorú sme tento rok urobili v organizácii a implementácii slovenskej súťaže bolo, že pre interaktívne úlohy sme namiesto prostredia Flash použili technológiu HTML 5 a JavaScript. Preto sa tohtoročné úlohy dali bez problémov riešiť aj na mobilných zariadeniach s operačným systémom Android, pozri [2].

V archíve súťaže sa v súčasnosti nachádza viac ako 600 zaujímavých informatických úloh, ktoré sú podľa náročnosti rozdelené do piatich kategórií podľa veku žiakov. Zadania úloh každoročne pripravujú odborníci z oblasti informatiky z viac ako tridsiatich krajín, ktorí formulujú zadanie svojich úloh nielen vo svojom jazyku, ale aj v angličtine. Tým sa stávajú úlohy navrhované do súťaže jazykovo závislé. Tento rok bolo pre národné súťaže odporúčaných viac ako 180. V slovenskej národnej súťaži, z rôznych dôvodov používame aj ďalšie úlohy, resp. niektoré zadania úloh modifikujeme a adaptujeme pre slovenské pomery tak, aby boli vhodné pre našu národnú súťaž, pozri [3].

2. JAZYKOVO ZÁVISLÉ ÚLOHY
Veľká časť súťažných úloh, ktoré sa nakoniec objavia v slovenskej verzii súťaže Informatický bobor, pochádza práve z medzinárodnej databázy. Jazyková závislosť každej úlohy môže mať viacero dôvodov. Jedným z nich je, že text úlohy je v angličtine. Adaptácia úlohy je v takomto prípade jednoduchá a spočíva väčšinou len v jej priamom preklade do slovenčiny. Ak je anglický text úlohy príliš dlhý, snažíme sa úlohu nielen preložiť, ale jej zadanie aj skrátiť. Zadanie úloh často obsahuje ilustračné, resp. vysvetľujúce, obrázky s anglickým textom, ktorý je potrebné preložiť a obrázok často upraviť aj iným spôsobom, aby vyhovoval, napr. zvyklostiam na Slovensku. Takto sme do tohtoročnej súťaže upravili obrázok pre úlohu Vláčik, v ktorom boli názvy železničných staníc pod obrázkami, pričom na našich nástupištiach je názov stanice väčšinou na budove, resp. kvôli prehľadnosti obrázku sme názov umiestnili nad budovu, pozri Obrázok 1 dolu.



Obrázok 1. Anglický a slovenský obrázok - Vláčik, 2016/17

Ďalšou, pomerne jednoduchou adaptáciou úlohy je také zadanie, ktoré obsahuje anglické krstné mená, resp. mená, ktoré sú bežné v krajine, z ktorej úloha pochádza. Takto boli v japonských úlohách použité mená Taro, Beba-ko, Betaro, Maiko, v nemeckých a švajčiarskych úlohách mená Fay, Ian, Alonzo, Cassy, Benno, Bernie, v lotyšských Cleveria, v holanských úlohách napr. Bart, Bubba, Hale, Serge, Maciej, v ruských Tanechka, vo švédskych úlohách mená ako Sven, Diggy, Eary, v úlohách z Taiwanu mená ako Brian, Liam. Takmer všetky tieto mená sme nahradili bežnými slovenskými menami, pozri ďalej časť 2.1.

V niektorých úlohách je ich riešenie závislé na textoch, ktoré sú súčasťou úlohy, napríklad, ak v úlohe o kódovaní použili autori anglické slovo, ktorého zakódovaním vznikne iné plnovýznamové anglické slovo. Ak chceme takúto úlohu použiť pre slovenskú

súťaž, môžeme buď ponechať anglické slová, alebo ako súčasť adaptácie úlohy, nájsť slovenské slovo tak, aby jeho zakódovaním vzniklo iné, zmysluplné slovenské slovo. Takýchto úloh bolo v súťaži niekoľko, pozri časť 2.2.

Pod typom úlohy, ktorá je jazykovo závislá budeme pre tento článok chápať dva typy úloh:

a) úloha obsahuje vo svojom zadaní **nezvyčajné krstné mená**, ktoré by mohli byť pre našich žiakov neznáme, a mohli by ich tak rušiť pri čítaní zadania úlohy,

b) úloha **obsahuje jazykovo závislé slová** (údaje), ktoré sú súčasťou zadania, a často aj riešenia, danej úlohy, a tak je potrebné nájsť inú sadu slov, ktoré vyhovujú zadaniu úlohy.

2.1 Úloha obsahuje nezvyčajné krstné mená

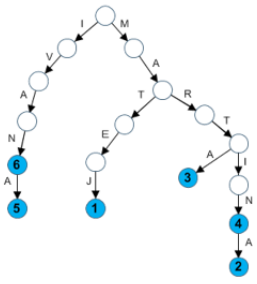
V prípade, ak ide o úlohu, v ktorej sa vyskytujú nezvyčajné krstné mená, je jej lokalizácia do slovenčiny, resp. aj do iného jazyka (súťaž Bebras prebiehala v minulom školskom roku v 38 krajinách) pomerne jednoduchá. Stačí, ak krstné meno, ktoré navrhli pôvodní autori úlohy, nahradíme takým, ktoré bude pre žiakov na Slovensku známe. Úlohy, v ktorých zadaniach sa vyskytujú mená ako Tail, Taro, Bruce, Justin, Carmen, Troy, či Buddy, by sa mohli zdať žiakom zvláštne a na prvý pohľad by bolo zrejmé, že tieto úlohy neboli navrhované pre nich, ale prevzaté zo zahraničia. Myslíme si, že ak by žiak našiel v zadaní takéto nezvyčajné mená, mohlo by to ovplyvniť aj jeho uvažovanie o úlohe a tiež postup pri jej riešení. V prípade adaptácie takejto úlohy do slovenčiny sme sa snažili nájsť také krstné mená, ktoré sú medzi žiakmi rozšírené, teda také, ktoré majú ich vrstovníci, či súrodenci. Takto sme nahradili napr. Lisa-Nina, Josh-Adam, Betaro-Timo, Bebako-Bea, Maiko-Majka, Sven-Samo, Mike-Michal. Ak má krstné meno z anglického zadania úlohy svoj ekvivalent v slovenskom kalendári, zachovali sme toto meno aj v slovenskom zadaní, resp. urobili sme v ňom len také zmeny, aby bolo zrejmé, že ide o slovenskú verziu mena, napr. Sara-Sára, Tom-Tomáš, John-Ján, Patrick-Patrik, Ann-Anna.

2.2 Slová (údaje) sú súčasťou zadania úlohy

Úlohy, ktoré obsahujú anglické slová (údaje), ktoré sú súčasťou zadania úlohy, a často aj jej riešenia, sú výzvou pre prekladateľov a autorov slovenských zadaní súťažných úloh. Pre každú z nich je potrebné najprv sa zamyslieť, či je vôbec možné lokalizovať úlohu do slovenčiny. Počas desiatich rokov súťaže bolo takýchto úloh niekoľko. Niektoré z nich sme nedokázali preformulovať tak, aby sme mohli úlohu zaradiť do našej súťaže. Iné úlohy sa nám podarilo zmeniť a nájsť také slovenské slová (údaje), ktoré vyhovovali zadaniu úlohy. V ďalšej časti článku uvedieme niekoľko takýchto úloh a spomenieme, aké úpravy bolo potrebné s každou z nich urobiť, aby sme ju mohli zaradiť pre niektorú súťažnú kategóriu.

2.2.1 Úloha Zoznam kancelárií

V školskom roku 2009/10 poslali slovenskí autori do medzinárodnej databázy úlohu Zoznam kancelárií, pozri Obrázok 2. Jej zadanie bolo založené na zobrazení slov pomocou lexikografického stromu. Keďže úlohu sme do medzinárodnej databázy posielali zo Slovenska, už pri jej vytváraní sme hľadali také slovenské krstné mená, ktoré obsahujú podreťazec iných mien. To sú napr. také, v ktorých je ženské krstné meno odvodené od mužského mena (Martin-Martina, Daniel-Daniela, Erik-Erika).



Vo firme Heker s.r.o. pracuje šesť počítačových expertov. Každý má vlastnú kanceláriu, kancelárie sú očíslované od 1 do 6. Vo vchode do firmy je namiesto obyčajného zoznamu kancelárií takýto obrázok:

Zobrazuje mená expertov a čísla ich kancelárií.

V ktorej kancelárii pracuje Marta?

A) 1
B) 2
C) 3
D) 4

Obrázok 2. Úloha Zoznam kancelárií – Benjamín, 2009/10

Ďalej sme hľadali také mená, ktoré začínajú na rovnakú skupinu jednej alebo dvoch hlások (Emil-Erik, Milan-Miloš-Michal, Martin-Marta-Marek-Matej). Keďže sme úlohu posielali do medzinárodnej databázy, museli sme sa pri jej preklade do angličtiny zaoberať anglickými krstnými menami s rovnakými vlastnosťami, aké sme hľadali v slovenských krstných menách. V anglickom zadaní sme využili mená James-Jane-Jack-Jackie-Jim.

2.2.2 Úloha PIN kód

Secret number reminder

Maciej carries a bookmark with this table on it to help him remember his secret four digit number.

1	2	3	4	5	6	7	8	9	0
A	B	C	D	E	F	G	H	I	J
K	L	M	N	O	P	Q	R	S	T
U	V	W	X	Y	Z				

If his secret number is 8526, all he has to do is to remember the word HELP.
To retrieve his number, he looks up the letters of the word HELP and finds the corresponding digits in the top row of the table. Another example: The word LOVE can be used to help Maciej remember the secret number 2525.
Maciej has to remember a new secret number. Only three of the following words produce this new number. Which one does not?
A) DOME
B) NEMO
C) NONO
D) NEWY

PIN kód

Aby si Matej zapamätal svoj PIN kód, vytvoril si takúto tabuľku:

1	2	3	4	5	6	7	8	9	0
A	B	C	D	E	F	G	H	I	J
K	L	M	N	O	P	Q	R	S	T
U	V	W	X	Y	Z				

Ak je jeho PIN kód 6541, môže si zapamätať napr. slovo PENA alebo slovo FOXU. Ak chce zistiť svoj PIN kód, tak k jednotlivým písmenám slova PENA nájde zodpovedajúce číslice vo vrchnom riadku tabuľky.
Matej si zmenil PIN kód. Tri z nasledujúcich slov mu pomôžu zapamätať si ho.
Ktoré slovo mu **nepomôže**?
A) USNE
B) KINO
C) KILO
D) AIDE

Obrázok 3. Úloha PIN kód – Benjamín, 2014/15

V školskom roku 2014/15 bola v medzinárodnej databáze úloha Secret number reminder (2014-NL-01), pozri Obrázok 3 hore. Ak má mať takto pripravené zadanie reálny význam, t. j. pomocou kódovacej tabuľky a nejakého slova si budeme pamätať svoj PIN kód, musí byť slovo, ktoré si má človek zapamätať zmysluplné slovenské slovo. Ak by sme úlohu preformulovali tak, aby sme si pamätali len skupinu ľubovoľných štyroch písmen, mohli by žiaci a učitelia, ktorí sa s úlohou stretnú v súťaži, argumentovať tým, že takto pripravené zadanie nie je reálne a že v takomto prípade je vlastne jedno, či si máme zapamätať skupinu číslic alebo skupinu písmen, ktorá nedáva zmysel. Vidíme, že v anglickom texte zadania sa vyskytovali slová HELP a LOVE a v rámci odpovedí

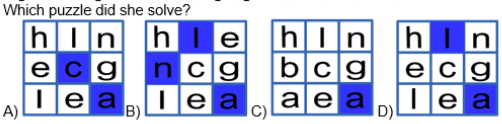
DOME, NEMO, NONO a NEVY. Ak by sme zadanie pripravovali pre starších žiakov, napr. pre Juniorov, zrejme by sme anglické slová nechali aj v slovenskom zadaní úlohy. Pre kategóriu Benjamín sme sa však pri adaptácii úlohy do slovenčiny snažili nájsť slovenské slová, resp. slová, ktoré znejú slovensky. Kvôli zachovaniu princípu zadania a jeho náročnosti bolo nevyhnutné, aby tri z navrhnutých slov používali písmená z kódovacej tabuľky, ktoré sú v stĺpci pod rovnakou číslicou, pričom štvrté slovo by sa malo odlišovať od zvyšných troch v treťom alebo štvrtom písmene. Druhú podmienku sme sa rozhodli zachovať kvôli tomu, aby úloha nebola príliš jednoduchá a aby žiak pri hľadaní riešenia nevidel, že hneď prvé písmeno odlišuje slová tak, že jedno z nich nemôže byť správnou odpoveďou. Skupinu slov sme našli a úlohu sme zaradili do slovenskej súťaže, pozri Obrázok 3 dolu.

2.2.3 *Úloha Hlavolam*

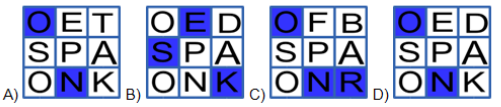
9-Letter Puzzle
Sandra likes to create word puzzles. She found in an Asian Newspaper a nice scheme to describe a word puzzle with a 3x3 grid of letters. To solve such a puzzle one has to form out of these 9 letters as many English words as possible which

- contain all highlighted letters
- contain at least 4 letters.

She created several puzzles and solved one of them.
Here is her solution:
angel, challenge, clan, clean, eagle, glance, hall, heal, lane, lean
Which puzzle did she solve?



Hlavolam
Sandra sa rada hrá s písmenami a vytvára z nich slová. V časopise našla písmenkový hlavolam, v ktorom je v tabuľke 3x3 zapísaných 9 písmen. Úlohou je nájsť všetky slová, ktoré majú aspoň 4 **písmená** a obsahujú **všetky podfarbené písmená**.
Sandra vyriešila jeden z nasledujúcich hlavolamov a našla takéto slová:
SPONA, OPONA, NAOKO, SENO, SNOP, ONAK, KANOE, OKNO, SONDA, PONAD
Ktorý hlavolam vyriešila?



Obrázok 4. Úloha Hlavolam – Benjamín, 2016/17

Úloha Hlavolam (2016-AT-02), pozri Obrázok 4 hore, z medzinárodnej databázy bola na úrovni údajov úplne jazykovo závislá a v tejto forme sme ju nemohli použiť pre našu národnú súťaž. Na prvý pohľad sa zdalo, že nebude problém nájsť hlavolamy, ktoré by obsahovali nejaké slovenské slová. Pre adaptáciu úlohy bolo potrebné nájsť vhodných deväť písmen a označiť niekoľko písmen, ktoré musia slová pri riešení obsahovať. Adaptácia tejto úlohy znamenala v podstate na základe daného princípu tvorbu novej úlohy. Pri vytváraní hlavolamu bolo, kvôli zachovaniu náročnosti úlohy, potrebné myslieť aj na to, aby slov, ktoré sa dajú z písmen vytvoriť bol dostatočný počet (v anglickom zadaní ich bolo desať). Ako prvú iteráciu sme viac-menej náhodne zvolili štyri samohlásky a päť spoluhlások, všetky bez diakritiky. V pôvodnom hlavolame nebolo dôležité, ako sú písmená v tabuľke usporiadané, takže aj my sme ich do tabuľky usporiadali náhodne. V prvom kroku prípravy úlohy nebolo dôležité ani to, ktoré písmená sú podfarbené. Podfarbenie niektorých písmen sme urobili dodatočne a tak, aby nám v hlavolame ostalo čo najviac slov. Následne bolo potrebné pripravený hlavolam vyriešiť, t. j. nájsť slovenské slová, ktoré sa dajú z navrhnutých písmen vytvoriť. Na riešenie úlohy sme využili počítač a program, ktorých postupne vytváral všetky kombinácie písmen a v pomerne veľkom slovníku slovenských slov bez diakritiky s viac ako 2700 slovami vyhľadával príslušnú skupinu písmen. Počítač našiel 50 slov, z toho 25 má aspoň štyri písmená: DEKA, DEKAN, DNES, DOSKA, KANOE, KONOPE, KOPA, KOSA, ODSEK, OKNO, OPAK, OPASOK, OPONA, PENA, PODNOS, PONAD, SAKO, SANE, SENO, SNOP, SONDA, SOPKA, SPODOK, SPONA, SPONKA. Keďže sa nám podarilo nájsť dostatočne veľa slov, rozhodli tento hlavolam použiť pre slovenské zadanie úlohy. Zvyšné tri nesprávne odpovede do úlohy sme navrhli tak, že v odpovediach B) a C) sme podfarbili ešte jedno písmeno, čím sa dajú tieto odpovede pri hľadaní správnej odpovede pomerne rýchlo vylúčiť. V prípade odpovede A), ktorá sa najviac podobá správnemu riešeniu sme v tabuľke zmenili písmeno D na T, pozri Obrázok 4 dolu.

2.2.4 *Úloha Telefónny zoznam*

Beaver Maxime is searching for her friend's phone number on a very long webpage. She is not sure how her friend's name is spelled so she uses special characters to search:
? is used when exactly one character is not known
& is used when exactly two consecutive characters are not known
% is used when the rest of the name is not known
For example, The% gives search results Theresa, Theodor, etc.
Maxime enters: S?rah B&cht%
Which name might Maxime be searching for?
A) Sirah Birchman
B) Sara Bilchdrain
C) Sarah Birchtree
D) Billy Beachtram

Bobrica Maxima hľadá vo veľmi dlhom zozname telefónne číslo svojej priateľky. Nevie, ako sa presne píše jej meno. Vyhľadávací program umožňuje použiť pri vyhľadávaní takéto znaky:
? keď nepoznáme práve jeden znak
& keď nepoznáme práve dva po sebe idúce znaky
% keď nepoznáme zvyšok mena
Napríklad, ak napíšeme Te% program nájde mená Tereza, Teodor, atď.
Maxima zadá: S?ra H&bo%
Ktoré z týchto mien program nájde?
A) Sára Hrubovská
B) Sandra Hraboňová
C) Sura Hubová
D) Saša Hladká

Bobor Max hľadá vo veľmi dlhom zozname telefónne číslo svojho priateľa. Nevie, ako sa presne píše jeho meno. Vyhľadávací program umožňuje použiť pri vyhľadávaní takéto znaky:
? keď nepoznáme práve jeden znak
& keď nepoznáme práve dva po sebe idúce znaky
% keď nepoznáme zvyšok mena
Napríklad, ak napíšeme Ti% program nájde mená Tibor, Tina, atď.
Max zadá: E?o B&zi%
Ktoré z týchto mien program nájde?
A) Edo Brezinka
B) Elo Bazinovský
C) Edon Breza
D) Emil Brezička

Obrázok 5. Úloha Telefónny zoznam – Junior, Senior, 2015/16

Anglické zadanie úlohy Telephone book (2015-AT-07), pozri Obrázok 5 hore, obsahovalo anglické mená aj priezviská, pričom súčasťou úlohy bolo, že žiak má doplniť časť mena a priezviska podľa pravidiel zo zadania, teda jedným alebo viacerými písmenami. Počas prekladu úlohy sme anglické mená Theresa a Theodor zachovali, pretože mená Tereza a Teodor sa dajú nájsť aj v slovenskom kalendári. V slovenskom zadaní sme zmenili len „The“ na „Te“. V odpovediach úlohy sa v anglickom texte vyskytovali mená Sára, Sirah, Sarah a Billy, ktoré sme vo výslednom zadaní z časti zachovali – Sára a z časti nahradili – Sandra, Saša, pozri Obrázok. 5 v strede. Priezvisko, ktoré sa vyskytovalo v anglickom zadaní, by v slovenských pomeroch znelo zvláštne, a tak sme sa snažili nájsť inú skupinu štyroch priezvisk, pričom sme chceli zachovať princíp zadania. Nakoniec sa nám to podarilo a priezviská v odpovediach znejú slovensky. Pri práci na preklade tejto úlohy do slovenčiny sme prehľadali množstvo priezvisk a našli sme aj niekoľko ďalších, ktoré vyhovovali rovnakému princípu vyhľadávania podľa chýbajúcich

písmen. Preto sme navrhli aj zadanie podobnej úlohy pre kategóriu Juniori, pozri Obrázok 5 dolu.

2.2.5 *Úloha Abecedné mená*

A name is called alphabetical when it can be assembled from the alphabetically sorted list of its letters by following this algorithm:

- Create the sorted list of letters.
- Start with the first letter from the list.
- Take the next letter from the list and add it to the left or to the right of the name.
- Continue till the list is empty.

The name HENRY is alphabetical: its alphabetical list of letters is EHNRY and we can put it together:
E -> **H**E -> HE**N** -> HEN**R** -> HENR**Y**
The name LUCY is not alphabetical: its alphabetical list of letters is CLUY. It is impossible to add second letter L to the first letter C because letters C and L are not adjacent in the word LUCY.
Which of these names is alphabetical?
A) JOHNNY
B) SUSAN
C) ISIDOR
D) ROBERT

Meno nazývame **abecedné**, ak sa dá zostaviť zo všetkých jeho písmen nasledujúcim postupom:

- vytvor abecedne usporiadanú postupnosť všetkých písmen mena
- začni prvým písmenom z postupnosti
- zober nasledujúce písmeno z postupnosti a pridaj ho vľavo alebo vpravo od vytváraného mena
- pokračuj, kým nie je postupnosť prázdna

Napr. meno TOMÁŠ je abecedné a postup, ktorým ho vytvoríme je:
ÁMOŠT
Á → **M**Á → **O**MÁ → OMÁ**Š** → **T**OMÁŠ
Ktoré z nasledujúcich mien je abecedné?
A) PETER
B) LUCIA
C) TIBOR
D) MONIKA

Křestní jména se nazývají abecedová, jestliže je lze postupně sestavit ze svého abecedního seznamu písmen podle následujícího algoritmu:
1. vytvoř z písmen jména abecední seznam
2. začni skládat jméno prvním písmenem seznamu
3. vezmi další písmeno ze seznamu a přidej ho zleva nebo zprava k tvořenému jménu
4. pokračuj bodem 3, dokud nedojdeš na konec abecedního seznamu

Jméno Romeo je abecedové: jeho abecední seznam písmen je emooR a můžeme jej postupně skládat
e> – me> – ome> – omeo> – Romeo
Jméno Julie není abecedové: jeho abecední pořadí písmen je eiJlu a k základu slova ie nelze přidat třetí J – písmena J a i neleží ve slově Julie vedle sebe.
Které z těchto jmen je abecedové?
(ch v úloze považujeme za 2 písmena, c + h)
A) Robert
B) Helena
C) Monika
D) Michal

Obrázok 6. Anglické, slovenské a české zadanie – Abecedné mená, 2016/17

Na prvý pohľad sa zdalo, že úloha Abecedné mená (2016-CZ-05), pozri Obrázok 6 hore, sa bude dať do slovenčiny preložiť a adaptovať pomerne jednoducho. Anglické krstné mená, ktoré obsahovala nám síce nevyhovovali, ale keďže v zadaní úlohy sa vyskytovali iba dve mená, ktoré majú spĺňať uvedené pravidlá pre abecedné meno (jedno v ukážke a jedno v správnom riešení), rozhodli sme sa, že úlohu pripravíme do slovenskej súťaže. Pri adaptácii úlohy sme však pomerne rýchlo pochopili, že nájsť hoci len dve abecedné mená bude problém. Nakoniec sa nám to podarilo. Na Obrázku 6 dolu vidíme české zadanie. Organizátori súťaže Bobřík informatiky, pozri [5], pri adaptácii tejto úlohy do češtiny použili namiesto bežného významu písmena „ch“ v češtine zvlášť písmeno „c“ a zvlášť písmeno „h“ a v zadaní úlohy museli toto nezvyčajné použitie dovysvetľovať.

2.2.6 *Úloha Pečiatky*

Beaver Code

For her birthday beaver Barbara has been given two stamps. With the first one she can produce a little flower, with the second one a little sun. Being a clever girl she thinks of a way to write her own name with these two stamps by encoding each letter with a sequence of flowers and suns as follows:

Letter	B	A	R	E	Y
Code					

The name "Barbara" would be encoded as:

She then goes on to also write down the names of her friends. But unfortunately they got all mixed up. Drag the sun-flower-codes to the names of her four friends:

Abby		
Arya		
Barry		
Ray		

Pečiatky

Nina dostala na narodeniny dve pečiatky. Na jednej je kvet a na druhej slnko. Nina sa rozhodla, že pomocou nich zakóduje písmená abecedy. Tu je niekoľko kódov:

písmeno	N	A	I	D	J
kód					

Zakódované meno NINA vyzerá takto:

Potom Nina zakódovala aj mená svojich kamarátok, ale už nevie, ktorý kód patrí ku ktorému menu. Ťahaním umiestni k menám ich kódy.

DANA

ANNA

JANA

DIANA

Pečiatky

Lea dostala na narodeniny dve pečiatky. Na jednej je kvet a na druhej slnko. Rozhodla sa, že pomocou nich zakóduje písmená abecedy. Tu je niekoľko kódov:

písmeno	E	A	L	N	K
kód					

Zakódované meno LEA vyzerá takto:

Potom Lea zakódovala aj mená svojich kamarátiek, ale už nevie, ktorý kód patrí ku ktorému menu. Ťahaním umiestni k menám ich kódy.

LANA

LENA

NELA

ELENA

Obrázok 7. Úloha Pečiatky – Benjamín, Kadet, 2016/17

Úloha Pečiatky (2016-CH-12) obsahovala v pôvodnom anglickom zadaní kódy pre päť písmen: B, A, R, E, Y., pozri Obrázok 7 hore. Na prvý pohľad nám bolo zrejmé, že nenájdeme skupinu piatich slovenských mien, ktoré by obsahovali práve tieto písmená,

dokonca aj meno Barbara, ktoré použili vo vysvetlení sa v slovenčine nachádza v tvare Barbora, takže obsahuje ďalšie písmeno, pre ktoré by sme museli pridať nové kódovanie a zmeniť tým pôvodné kódovanie. V tejto podobe bola teda úloha pre slovenskú súťaž nepoužiteľná. Ak sme pri adaptácii úlohy chceli z pôvodnej úlohy zachovať počet kódovaných písmen a aj ich kódovanie, bolo pre jej adaptáciu do slovenčiny potrebné nájsť inú skupinu piatich písmen. Prípravu slovenského zadania sme začali prezeraním mien v slovenskom kalendári. Adaptácia tejto úlohy bola pomerne náročná a dlhší čas sa zdalo, že mená, ktoré by sa skladali z piatich písmen nedokážeme nájsť. Napokon sme našli skupinu takých piatich písmen, z ktorých sa dá poskladať päť slovenských mien. Dokonca sa nám podarilo nájsť aj ďalšie mená a tak vznikli dve slovenské interaktívne zadania úlohy Pečiatky pre kategórie Benjamín a Kadet, pozri Obrázok 7 v strede a dolu.

3. ZÁVER

Adaptácia anglických zadaní úloh súťaže iBobor z medzinárodnej databázy do slovenčiny môže byť niekedy veľmi jednoduchá a do slovenčiny stačí preložiť iba text úloh. Niekedy však ide o náročnejší proces, keď je potrebné prispôsobiť nielen zadanie úlohy, ale aj anglické údaje, ktoré úloha obsahuje. Je na organizátoroch národných súťaží a autoroch úloh v každej krajine, či preložia krstné mená v zadaniach, či preložia slová v obrázkoch zadaní a či prispôsobia údaje v zadaní úlohy svojmu jazyku. Slovenskí organizátori súťaže iBobor sa snažia adaptovať úlohy v plnom rozsahu. Prekladajú a upravujú aj údaje v úlohách, s ktorými je pri riešení úlohy potrebné niečo urobiť (zakódovať, zobraziť v údajovej štruktúre, vyhľadať alebo nahradiť ich časti), aby tak žiakom uľahčili pochopenie zadania a veríme, že následne aj nájdenie riešenia danej úlohy.

BIBLIOGRAFICKÉ ODKAZY

[1] Hlavná stránka súťaže Informatický bobor. http://www.iBobor.sk

[2] Winczer, M., Tomcsányiová, M., Tomcsányi, P.: Programovanie interaktívnych úloh súťaže iBobor pomocou knižnice sprites.js. *DidInfo 2017* [CD]. Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, 2017.

[3] Tomcsányiová, M.: Úlohy súťaže iBobor zamerané na rôzne stratégie riešenia problémov. *DidInfo 2010* [CD]. Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, 2010. ISBN 978-80-8083-952-9.

[4] Tomcsányiová, M., Kabátová, M.: Categorization of pictures in tasks of the bebras Contest. *Informatics in Schools : Sustainable Informatics Education for Pupils of all Age*. Berlin : Springer, 2013. pp. 184-195.

[5] Hlavná stránka českej súťaže Bobřík informatiky. http://www.ibobr.cz

Aplikace pro výuku základních principů objektově orientovaného programování

Application for teaching the basic principles of object-oriented programming

Mgr. Martin Vašek
Ostravská univerzita, Pedagogická fakulta
Katedra informačních a komunikačních technologií
Fráni Šrámka 3, Ostrava – Mariánské Hory
Česká republika
martin.vasek1990@gmail.com

ABSTRACT

The following article focuses on various aspects of the application for teaching the basic principles of object-oriented programming and describes its content and options for correct usage in the classroom. Text can be in some ways understood as a brief guide, as a tool to work correctly with application. Conclusions contained herein are based on lessons supported by this teaching tool. The application is used by the Department of Information and Communication Technology, Pedagogical Faculty of University of Ostrava.

The main idea of teaching aid is to enable design and creation of classes in specific form which allows to define and set appropriate parameters without the necessary knowledge of programming language. Application for teaching the basic principles of object-oriented programming is freely available to download and link is contained in this publication.

Keywords

Object-oriented programming. Object. Class. Java. Python. Object Pascal.

ABSTRAKT

Následující článek se zaměřuje na jednotlivé aspekty aplikace pro výuku základních principů objektově orientovaného programování a popisuje její obsah a možnosti správného využití ve výuce. Text lze v jistém ohledu chápat i jako stručný návod, jak s pomůckou korektně pracovat. Závěry zde uvedené vychází z již absolvované výuky podporované právě pomůckou. Aplikace je využívána na Katedře informačních a komunikačních technologií Pedagogické fakulty Ostravské univerzity.

Hlavní myšlenkou pomůcky je umožnění návrhu a tvorby třídy specifickou formou, která dává možnost definovat a nastavit příslušné parametry bez nutné znalosti programovacího jazyka. Aplikace pro výuku základních principů objektově orientovaného programování je volně přístupná a odkaz pro stažení je obsažen v této publikaci.

Klíčová slova

Objektově orientované programování. Objekt. Třída. Java. Python. Object Pascal.

1. ÚVOD

Objektově orientované programování je v razantní míře využíváno při vytváření desktopových i webových aplikací a tvoří nedílnou součást povinných znalostí u lidí, kteří se pohybují nebo chtějí pohybovat ve světě informačních technologií. Jedná se o paradigma, které je využíváno již desítky let. Myšlenka zapouzdření, která se jeví jako určující prvek ve vývoji programování [1] dává předpoklad, že objektově orientované paradigma bude i nadále velmi aplikovaným modelem programování. Velké množství programovacích jazyků umožňuje práci s objekty a vlastnostmi kolem objektů, které sebou právě koncept objektově orientovaného programování přináší. Je navíc velmi příjemným faktem, že využití objektů, jejich hlavní myšlenka i samotné názvosloví v celém objektově orientovaném programování, je v jistém smyslu stejné, ať se podíváme na kterýkoli programovací jazyk s podporou tohoto modelu. To umožňuje jakousi přenositelnost znalostí člověka, který se naučil v jednom jazyce, do jazyka jiného [1].

S cílem dosažení spolehlivého a rychlého osvojení si základů objektově orientovaného programování (dále jen „OOP“) byla v roce 2015 vytvořena desktopová aplikace. Tato softwarová vzdělávací pomůcka v sobě nese jak textové studijní materiály, tak i praktickou a názornou pomůcku v programovacích jazycích Java, Python a Objektový Pascal. Pomůcka má za cíl vhodným způsobem studenům ukázat, jak fungují jednotlivé vlastnosti v objektově orientovaném programování a jak se mezi sebou tyto vlastnosti ovlivňují. Během práce s pomůckou musí student aktivně pracovat a tvořit třídu a případné objekty, čímž se má docílit jednoduššího vštípení znalostí.

Aplikace byla vytvořena pro studenty studijního oboru Informační a komunikační technologie ve vzdělávání. Za typický profil studenta lze zde považovat člověka, který má s informačními technologiemi zkušenosti, ale i přesto se z odborného hlediska jedná o začátečníka, který není programátorem a samotné programování pro něj není prioritou. Paradigma OOP je pro cíleného studenta v počátcích práce s aplikací neznámou oblastí.

Na Katedře informačních a komunikačních technologií Pedagogické fakulty Ostravské univerzity byla aplikace úspěšně vyzkoušena ve výuce a v konečném hodnocení byla vyhodnocena jako přínosná a vhodná pro využití ve výuce OOP.

Odkaz pro stažení aplikace je následující:

http://www1.osu.cz/~d16481/OOP aplikace.zip

2. ZÁKLADNÍ RYSY APLIKACE

Jedná se o ryze vzdělávací software, který byl vytvořen se záměrem umožnit studentům nahlížet na principy OOP i z jiného hlediska než pouze psaní kódu daného programovacího jazyka.

Aplikace podněcuje aktivitu na straně studenta pomocí aktivní stavby třídy (viz kapitola 3), kdy student sestavuje třídu z jednotlivých komponent OOP. Další funkcí aplikace je obsažení textových studijních materiálů.

2.1 Programovací jazyky

V rámci aplikace se studenti mohou seznámit s návrhem a tvorbou tříd v jazycích Java, Python a Object Pascal, jež jsou pro výuku na českých středních školách využívány nejčastěji [2].

Před začátkem práce s programem je samozřejmostí výběr jednoho jazyka. To má za cíl oddělení jednotlivých jazyků, aby nedocházelo k prolínání několika syntaxí, což by studenty mohlo mást.

Všechny tři zvolené jazyky jsou v základu již objektově orientované a není tedy nutné, aby před prací s nimi byly instalovány případné externí knihovny. Lze říci, že Java a Python jsou moderní programovací jazyky, a proto je jejich zařazení v aplikaci vhodné. Object Pascal vychází z Pascalu, který byl původně určen hlavně jako jazyk k výuce programování. Proto je Object Pascal ideálním kandidátem pro studenty, kteří se chtějí naučit základy objektově orientovaného programování [3].

2.2 Cílová skupina

Cílovou skupinou jsou vysokoškolští studenti oboru Informační a komunikační technologie ve vzdělávání, který připravuje studenty pro práci ICT specialisty na základních a středních školách. Studenti absolvuji výuku proto, aby se seznámili s principy OOP a pochopili je.

2.3 Způsoby přenosu znalostí

Záměrem pro využití aplikace je výuka za podpory vyučujícího. Neboli hlavní představa práce s aplikací je taková, že studenti pracují s pomůckou dle instrukcí, které jim zadal vyučující (více k formám výuky s aplikací v kapitole 4).

Obsah aplikace je rozdělen na dvě části:

- Aktivní stavba třídy.
- Studijní textové materiály.

2.3.1 Aktivní stavba třídy

Jedná se o nejdůležitější prvek aplikace, ve kterém je student aktivně zapojen do práce, a vytváří své vlastní výstupy. Cílem této praktické pomůcky **je vyzkoušet si vytvořit třídu a objekty, a to v názorné a přehledné formě**, kdy jsou studentovi, pomocí jednotlivých komponent, ukazovány prvky a vlastnosti třídy. Pokud by se student učil vytvářet třídu bez aplikace, jeho aktivita by byla pravděpodobně zaměřena čistě na práci s vývojovým prostředím, kde by byl výstupem jeho práce pouze zdrojový kód třídy a deklarace objektu. Je záměrem, aby aplikace pomáhala tím, že tvorbu třídy představuje studentovi ve více vizuální podobě bez nutné znalosti konkrétních příkazů programovacího jazyka.

Hlavní výstupy praktického studia s pomůckou by měly být následující:

- Vizuální zobrazení jednotlivých prvků třídy.
- Zdrojový kód buďto u samostatných prvků třídy nebo třídy celé.
- Vygenerování souboru se zdrojovým kódem třídy.

Aktivní stavba třídy je stěžení součástí aplikace a je proto podrobně popsána v kapitole 3 této publikace.

2.3.2 Studijní textové materiály

Aplikace obsahuje studijní materiály, které jsou rozděleny do 10 větších tutoriálů. Ty se věnují základům programování v konkrétních jazycích a dále pak rozebírají podrobněji jednotlivé aspekty OOP v obecné rovině. Mimo tyto materiály, odkazuje aplikace na 15 PDF souborů s příklady konkrétních příkazů programovacích jazyků.

V jistém smyslu jsou studijní materiály odděleny od praktické části programu. Je na ně odkazováno a v rámci vytváření třídy v aplikaci je vždy možné, aby si student texty jednotlivých problematik zobrazil. Důležitou vlastností aplikace je fakt, že lze mít zobrazeny studijní texty a zároveň vytvářet třídu v okně aplikace.

3. AKTIVNÍ STAVBA TŘÍDY

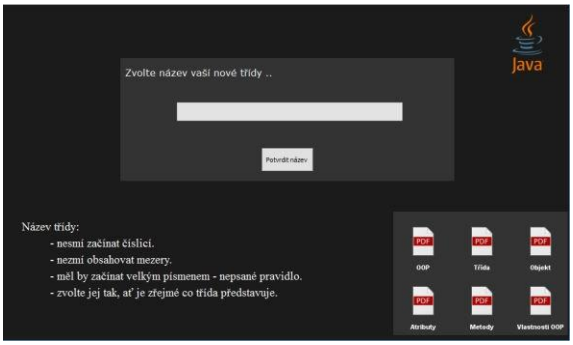
Praktickým úkolem aplikace je umožnit studentům, aby vytvořili třídu v objektově orientovaném programování, a to ne za pomoci psaní zdrojového kódu třídy, ale pomocí sestavování jednotlivých prvků třídy krok po kroku.

3.1 Výběr programovacího jazyka a nastavení názvu třídy

Menu pro zvolení programovacího jazyka je úvodní obrazovkou po spuštění aplikace (viz obrázek 1 níže). Lze odtud postoupit ke studijním materiálům.

Obrázek 1: Volba programovacího jazyka

Po výběru programovacího jazyka student nastaví název své třídy (obrázek 2). Aplikace jej kontroluje, zdali není nastavený název chybný. Pokud chybný je, studentovi není umožněno pokračovat dále a musí název opravit.



Obrázek 2: Nastavení názvu třídy

3.2 Tvorba obsahu třídy

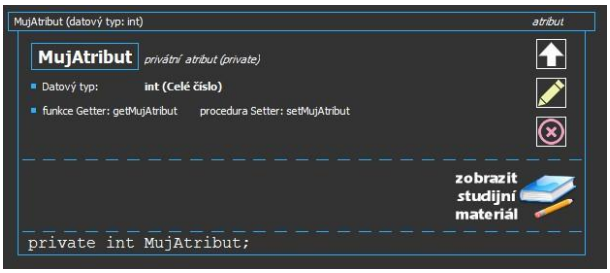
Při vstupu do této části aplikace má vytvářená třída definovaný pouze název a předpokládá se, že student bude editovat následující prvky:

- Rodičovská třída, abstraktnost třídy a případně úprava názvu třídy.
- Vložení atributů třídy.
- Vložení běžných metod třídy.
- Vložení speciálních metod – konstruktory a destruktory.
- Deklarace objektů vytvářené třídy.

3.2.1 Vkládání prvků třídy jako komponent

Jednotlivé komponenty třídy (metody, atributy, konstruktor, destruktor) lze vložit pomocí menu tlačítek na pravé straně okna aplikace. Po vložení se zobrazí panel dané komponenty ve větší oblasti nalevo, a právě tam student jednotlivé prvky třídy upravuje.

Po vložení například atributu je zobrazen modrý panel, ve kterém student postupně nastaví název atributu, vlastnosti zapouzdření a datový typ. Během vytváření prvků třídy je vždy dynamicky zobrazován a aktualizován zdrojový kód, který se k danému prvku váže. Po vytvoření je panel vyplněn informacemi o atributu a umožňuje prvek třídy editovat, odstranit nebo panel schovat pro lepší přehlednost a orientaci mezi ostatními panely (viz obrázek 3).



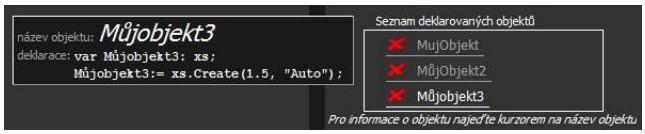
Obrázek 3: Podoba panelu s atributem a jednotlivými informacemi o něm

Pokud student najede kurzorem myši například na informaci o datovém typu, zdrojový kód níže zvýrazní žlutou barvou právě příkaz, který datový typ nastavuje. Jedná se o velmi názornou ukázku pro studenty, kteří mohou díky ní identifikovat všechny příkazy v deklaraci a udělat si přehled o jejich vlivu na, v tomto případně, vytvářený atribut.

Podobnou podobu jako panel atributu mají panely funkcí (oranžová barva), konstruktorů a destruktorů (světle zelená barva).

3.2.2 Objekty

Deklarace objektů probíhá pomocí správce objektů. V něm lze vytvořit libovolný počet objektů třídy, které lze pomocí správce odstranit. Při najetí myší na název objektu se o něm zobrazí základní informace, a to jeho název a zdrojový kód pro deklaraci (obrázek 4).



Obrázek 4: Správce deklarovaných objektů

Velkou výhodou je propojení vytváření objektů a již definovaných konstruktorů ve třídě. Pokud vytváříme objekt, můžeme zvolit konstruktor, který bude volán pro inicializaci, a lze tedy zvolit i počáteční hodnoty atributů, které má student již ve třídě navoleny.

3.2.3 Výstupy tvorby třídy

Výstupem, mimo samotnou práci na třídě, je zdrojový kód, a to formou buďto zobrazení čistě textu zdrojového kódu (a možnost jeho zkopírování) nebo vygenerování zdrojového souboru ve formátu programovacího jazyka, se kterým student pracuje. Výstupními formáty jsou *.java (Java), *.py (Python) nebo *.pas (Object Pascal).

4. ROLE VYUČUJÍCÍHO VE VZDĚLÁVACÍM PROCESU S APLIKACÍ

Je samozřejmostí, že výuka, kterou aplikace pro OOP podporuje, musí probíhat kontrolovaně a vyučující si musí uvědomovat svou roli vůči žákům. Pomůcka umožňuje učiteli zaujmout několik rolí [4]. Přitom nelze samozřejmě vyloučit existenci i jiné formy výuky s pomůckou, které tvůrce aplikace nezamýšlel a které zde nejsou popsány. Kreativní vyučující může být schopen vytvořit vlastní novou metodu, jak s aplikací ve výuce pracovat.

4.1 Vyučující jako průvodce

Hlavní náplní tohoto modelu vyučování je zapojení učitele jako pomocníka, a tedy i jako průvodce studenta za poznáním**. Student se k informacím dopracuje převážně samostatně**, a to na základě diskuze a interakce s vyučujícím.

Vyučující zasahuje v případě, že:

- student (nebo případně skupina studentů) je neschopen samostatně pokročit v řešení problému a je nutné mu ukázat cestu,
- vyučující vidí nezbytnou potřebu vysvětlit část kurikula tak, aby bylo zajištěno, že studenti danou problematiku pochopí správně.

Po nalezení řešení studentem je doporučena zpětná vazba ze strany vyučujícího, kdy je student za své řešení odměněn, a to slovní pochvalou nebo případně jinou formou, která je motivující k další aktivitě a snaze studenta.

4.2 Vyučující jako vzor

Tento model je postaven na výkladu a aktivitě vyučujícího, kdy pracuje před studenty s aplikací a oni mají za úkol postupovat dle jeho instrukcí, a hlavně dle jeho konání. Neboli studenti se pokouší napodobovat kroky vyučujícího, který pracuje s aplikací. Je potřeba, aby učitel byl předem připraven a měl dobře naplánováno, který problém bude před studenty řešit. Výklad musí správně popisovat jeho zacházení s pomůckou a zároveň by měl objasňovat jednotlivé obsahy učiva.

Jedná se o **vhodnou metodu pro naučení se používat aplikaci**. Právě první výuka podporovaná aplikací by měla probíhat touto formou, aby byla studentům ukázána správná manipulace s pomůckou.

4.3 Vyučující jako zadavatel

Zde je vyučující postaven do role, kdy má připraven úkol (či větší projekt) a prací učitele je tedy seznámit studenty se zněním zadání. Hlavním prvkem je zde ponechání studentů, aby řešili problém samostatně a od vyučujícího je očekávána nižší aktivita. Naproti tomu je na studenty kladen požadavek, aby vypracovali řešení od začátku až do konce sami. Po ukončení práce studentů je nutné, aby

vyučující jejich práci vyhodnotil a aby tak vznikla zpětná vazba studentům.

V případě, že se studentům nebude dařit v pokročení s řešením problému, je možné, aby vyučující zasáhl formou průvodce (viz kapitola 4.1) a vhodným způsobem je navedl.

Tento model není primárně určen k získání nových znalostí, ale jeho podstatou je upevnění si již nabytých znalostí. Je zde očekávána dobrá zpětná vazba od studentů, a informace pro vyučujícího, zda studenti danou problematiku chápou či nikoli.

5. UČEBNÍ ÚLOHY

Během práce s aplikací by měli být studenti postaveni před problémy, které lze nazvat jako učební úlohy. Jedná se o takové úlohy, které vyžadují určitou úroveň myšlení a mají svou danou náročnost. Nejznámější kategorizace učebních úloh je taxonomie dle Tollingerové (rozpracováno také například v [5]), kdy je 27 typů úloh rozděleno do 5 kategorií, které postupně zvyšují úroveň náročnosti:

1. Úlohy vyžadující pamětní reprodukci.
2. Úlohy vyžadující jednoduché myšlenkové operace s poznatky.
3. Úlohy vyžadující složitější myšlenkové operace s poznatky.
4. Úlohy vyžadující sdělení poznatků.
5. Úlohy vyžadující tvořivé myšlení.

Následující podkapitoly popisují možné znění úloh pro práci s aplikací. Úlohy pro pamětní reprodukci nejsou zahrnuty, jelikož se aplikace zaměřuje spíše na práci studenta a jeho praktickou činnost s objekty a třídami.

5.1 Jednoduché myšlenkové operace

Zde lze nalézt příklad úlohy **na vyjmenování a popis faktů**, a to kupříkladu v následujícím znění: „Vytvořte v aplikaci atribut u naší třídy člověk. Tento atribut ponese informaci o tom, jaké má člověk zaměstnání. Zkuste navrhnout vhodnou formu atributu a to tak, aby nabýval pouze číselné hodnoty a názvy jednotlivých zaměstnání by přebíral například z externího souboru.“ Při řešení této úlohy nemůže být v atributu obsažen celý název zaměstnání, ale pouze číselný odkaz (popis faktu). V praxi bylo postupováno tak, že si studenti vytvořili atribut s typem řetězce znaků a pak byli postaveni před problém, kdy mají jeden typ zaměstnání u více objektů a potřebují ono zaměstnání upravit. řešení tohoto úkolu může student v budoucnu propojit s funkčností databází a ošetřením redundancí a nekonzistencí. Postup studentů vedl k tomu, že si studenti museli uvědomit praktičnost číselného odkazu, což je odkaz číselnou hodnotou na jeden jediný řetězec znaků s názvem zaměstnání. Když studenti došli k pochopení toho, že potřebují číselnou hodnotu, změnili v aplikaci již existující atribut s řetězcem znaků na číselný atribut.

Dalším typem úlohy je **zjištění vztahů mezi fakty**. Otázka takovéto úlohy může znít takto: „Co se stane během vytvoření objektu?“ Zde se student má dopracovat vztahu mezi konstruktorem a počátečním stavem objektu. Tato úloha se v praxi ukázala jako obtížná, protože studenti museli mít znalosti o tom, co je třída a jaký má vliv na vytvářený objekt. Fakt, že třída je pouze jakýmsi předpisem toho, jak bude objekt pracovat a čím bude tvořen, je pro studenty mnohdy těžce pochopitelný.

Jiným úkolem by mohlo být nalezení vztahu mezi tzv. metodami getter a setter: „V aplikaci vidíme vytvořené metody getNumber a setNumber. Která metoda nastavuje hodnotu atributu Number a která naopak vrací onu hodnotu? Zkuste se zamyslet nad tím, která z těchto metod bude funkcí a která procedurou.“ Tato úloha ukázala, že pro studenty není těžké pochopit funkčnost getterů a setterů. Oproti předešlé úloze, která se věnovala pochopení teorie vytvoření objektu, je tato úloha spíše praktického rázu a při výuce ji studenti byli schopni lépe uchopit i díky výukové aplikaci, protože výsledné settery a gettery měli před sebou a mohli s nimi pracovat.

Jako poslední v kategorii pro jednoduché myšlenkové operace lze zmínit úlohy **pro abstrakci, zobecnění a konkretizaci**. Dobrým příkladem je zadání v následujícím znění: „Máme třídu žirafa, která obsahuje své atributy a metody. Popište, co všechno u žirafy třída definuje. Pak si pár objektů pro žirafy pomocí aplikace vytvoříme.“ Jedná se o vhodnou úlohu pro orientování se v těle třídy. Při využití takovéto úlohy studenti získali přehled o umístění deklarací atributů, metod a dále také deklarace samotného názvu třídy. Ukázalo se, že při vhodném postupu vyučujícího a malého navedení z jeho strany, jsou studenti schopni veškeré součásti třídy identifikovat sami.

5.2 Složitější myšlenkové operace

Do této kategorie patří úlohy, které se zaměřují **na výklad**. Obecně se jedná o jakékoli otázky, které mají iniciovat vysvětlení problematiky studentem, jako například: „Vysvětli, proč je dobré mít parametry u konstruktoru.“ Zde lze nejen dobře upevnit studentovy znalosti, ale také rozvíjet jeho komunikační dovednosti. Zároveň má vyučující možnost zjistit, zdali student problematiku chápe správně. U zmíněné úlohy s konstruktorem si studenti většinou nevědí rady, protože propojení existence objektu s deklarací konstruktoru a jeho voláním je obtížné. Navíc uvědomění si faktu, že parametry jsou cestou, jak inicializovat počáteční hodnoty atributů u objektů, vyžaduje více zkušeností s tvorbou tříd a deklarováním atributů.

Úlohou **na vyvozování** může být v rámci aplikace zadána následující práce: „Pomocí aplikace zjistěte, jak a kde se ve zdrojovém kódu třídy deklarují atributy.“ Student zde musí v aplikaci nejprve atribut zadat. Pomocí zobrazeného zdrojového kódu poté uvidí přesnou deklaraci atributu, kterou musí nejprve sám vyhledat. Tato úloha patří spíše k jednodušším a to obzvlášť, pokud již student ovládá dobře práci s aplikací.

5.3 Sdělení poznatků

Jedná se o všechny úlohy, které jsou cílené na zisk výstupu z aplikace. Buďto zdrojového kódu v textové podobě nebo konečného souboru se zdrojovým kódem. Vhodným příkladem je úloha, která vyžaduje slovní popis například atributu nebo metody, a to podle zdrojového kódu deklarace.

5.4 Tvořivé myšlení

Zde se řadí obtížné úlohy, u kterých musí student ukázat kreativitu a důkladnou analýzu problému. Úloha pro tvořivé myšlení v rámci aplikace může mít toto znění: „Vytvořte k naší třídě dvě metody na vykreslení objektu tak, aby první metoda byla připravena vykreslit objekt modrou barvou. Druhá metoda bude mít prostředek k tomu, abychom zadali jakoukoli barvu pro vykreslení při volání této metody.“ Student zde musí důkladně promyslet to, co bude vytvářet, což jsou v tomto případě dvě metody. Musí se zamyslet nad tím, jestli budou mít obě metody parametry a případně který datový typ bude využit.

Další úlohou je již celistvá práce s třídou a jejími objekty. Tato úloha může vypadat takto: „Děláme virtuální zoologickou zahradu a každý z vás do ní vytvořte svou třídu jednoho zvířete. Objekty pak budou zvířata, která naši zahradu zaplní.“ Studenti zde v praxi

museli popřemýšlet, jaké zvíře vytvoří a pak museli definovat obsah třídy dle toho, jak své zvíře chtěli realizovat. V této úloze se nalézá to nejobtížnější, co paradigma OOP obsahuje, a to nutnost přemýšlet dopředu a vytvářet atributy a metody, které budou vyžadovány. Zároveň je nutné znát jednotlivé vlastnosti OOP, aby mohly být studentem využity. V praxi se ukázalo, že čím více tříd studenti v minulosti vytvořili, tím rychleji a efektivněji budou v budoucnu třídy sestavovat.

U úloh této obtížnosti je nezbytné, aby již student uměl bezchybně zacházet s aplikací a mohl se tedy soustředit čistě na funkčnost a obsah objektů a tříd, se kterými pracuje.

6. ZÁVĚR

Aplikace pro výuku základních principů OOP je dobrým doplňkem pro studenty, kteří studují na vysokoškolské úrovni. Pomůcka cílí na vhodné podání těch znalostí, které by každý student informačních technologií měl znát a zároveň odstraňuje potřebu znát jednotlivé příkazy daného programovacího jazyka. Dá se říci, že aplikace didakticky správným způsobem zobecňuje problematiku OOP. Student je podněcován k práci s jednotlivými principy programování, aniž by musel psát konkrétní příkazy, které by měly být až dalším krokem ve studiu.

Aplikace při svém využití ve výuce napomáhá k lepšímu pochopení studentova uchopení problematiky OOP. Neboli aplikace pomohla získat znalosti o tom, co studentům není jasné a co jim znesnadňuje tvorbu tříd a deklaraci objektů. Toto je vhodné pro vyučujícího, aby studentům objasnil právě ty skutečnosti, o kterých již bude vědět, že mohou studentům dělat problém.

Nejčastěji byla aplikace využita pro výuku programovacího jazyka Python. Ze zpětné vazby získané od studentů, kteří podstoupili výuku s aplikací, lze vyvodit závěr, že pomůcka jim pomohla k zisku nových znalostí a k ujasnění si problematiky z dřívějška, kterou studenti neovládali. I z tohoto důvodu je počítáno s využitím aplikace v budoucí výuce základů OOP.

BIBLIOGRAFICKÉ ODKAZY

[1] NAGYOVÁ, Ingrid. *Podstatné aspekty informatiky jako vědního oboru.* Vyd. 1. Ostrava: Ostravská univerzita v Ostravě, 2013. ISBN 978-80-7464-378-1.

[2] KOTEK Lukáš. Výzkum používaných programovacích jazyků na středních školách. In*: Olympiáda techniky 2013*. Plzeň, 2013. Dostupné z: https://dspace5.zcu.cz/bitstream/11025/16677/1/kotek_vyzkum_clanek.pdf

[3] VAŠEK, Martin. *Aplikace pro výuku základních principů objektově orientovaného programování*. Ostrava, 2016. Diplomová práce. Ostravská univerzita.

[4] PODLAHOVÁ, Libuše. *Didaktika pro vysokoškolské učitele.* Praha: Grada, 2012. Pedagogika (Grada). ISBN 978-80-247-4217-5.

[5] TOMÁŠ, Martin. Učební úlohy. In: *Dielektrika* [online]. Plzeň: Fakulta pedagogická Západočeské univerzity v Plzni, 2009 [cit. 2017-01-10]. Dostupné z: http://dielektrika.kvalitne.cz/ulohy.html

[6] PECINOVSKÝ, Rudolf. *OOP: naučte se myslet a programovat objektově*. Vyd. 1. Brno: Computer Press, 2010. ISBN 978-80-251-2126-9.

[7] SPILKOVÁ, Vladimíra. *Současné proměny vzdělávání učitelů.* Brno: Paido, 2004. ISBN 80-7315-081-6.

[8] PECINOVSKÝ, Rudolf. Výuka objektově orientovaného programování žáků základních a středních škol. In: *Objekty* 2003 [online]. Ostrava, 2003 [cit. 2017-01-10]. Dostupné z: http://vyuka.pecinovsky.cz/prispevky/2003_OB_Vyuka_OOP_zaku_ZS_a_SS.pdf

[9] BLAHO, Andrej. *Programovanie v Pythone*. Bratislava, 2016. Dostupné z: http:/python.input.sk/

Aktivity s LEGO WeDo vo vyučovaní informatiky
LEGO WeDo activities in Informatics learning

Michaela Veselovská
KZVI FMFI UK
Mlynská dolina 1
842 48 Bratislava
SR
veselovska@fmph.uniba.sk

Karolína Mayerová
KZVI FMFI UK
Mlynská dolina 1
842 48 Bratislava
SR
mayerova@fmph.uniba.sk

ABSTRACT
In this article we characterize curriculum for educational robotics, which we iteratively developed within our dissertation research. We developed 11 activities together with complex methodical materials, which teachers can productively integrate into their daily teaching and thus prepare attractive introduction for educational programming for pupils using them. We reckon that curriculum developed in this way can serve mainly for fulfilling modern educational goals of Informatics. With our new intervention pupils can develop important skills, knowledge and abilities, which they will use not only in school, but in their personal lives as well.

Keywords
Informatics. Activities in compulsory education. Educational robotics. LEGO WeDo. Lower secondary school.

ABSTRAKT
V tomto článku stručne charakterizujeme vzdelávací program pre edukačnú robotiku, ktorý sme iteratívne vyvíjali v rámci nášho dizertačného výskumu. Tento program zahŕňa 11 aktivít spolu s komplexnými metodickými materiálmi, ktoré môžu učitelia začleniť do svojej bežnej výučby a pomocou nich realizovať pre žiakov pútavým spôsobom úvod do edukačného programovania. Domnievame sa, že vzdelávací program vyvinutý takýmto spôsobom poslúži k napĺňaniu moderných cieľov informatiky a v neposlednom rade i k rozvoju mnohých dôležitých zručností a znalostí žiakov, ktoré využijú nielen vo vzdelávaní, ale i vo svojom osobnom živote.

Kľúčové slová
Informatika. Aktivity na vyučovaní. Edukačná robotika. LEGO WeDo. Nižšie sekundárne vzdelávanie.

1. ÚVOD
Výsledky analýzy viacerých publikácií ukazujú pozitívny rozvoj vedomostí a zručností pri využití edukačnej robotiky. Žiaci nadobudli nové vedomosti v priebehu vyučovania *programovania*, *geografie* a *robotiky* [1, 2, 3], pri učení sa základných princípov *evolúcie* [4] a pri vyučovaní *systémov* [5]. Medzi nadobudnuté zručnosti patrili spôsobnosti potrebné na prírodovedné skúmanie [5], zručnosti sociálnej interakcie [6] a zručnosti na riešenie problémov [3]. Autori [7] sa domnievajú, že robotické modely nepredstavujú iba špičkové moderné technológie, ale zároveň reprezentujú aj hmotný výsledok procesu učenia sa žiakov, čo je len ďalším dôvodom využívať ich vo vzdelávaní. Gura v [8] dokonca tvrdí, že pri skúmaní vzdelávacích štandardov takmer každého predmetu môžeme nájsť produktívne prepojenie s edukačnou robotikou. Hovorí tiež, že čím dlhšie ju budeme využívať pri vyučovaní, tým viac možností nájdeme, ako s jej pomocou rozvíjať vzdelávacie ciele na konkrétnom predmete. V našom doterajšom výskume sme sa zameriavali prevažne na výučbu na základnej škole. Či už išlo o prvý alebo druhý stupeň ZŠ. Jedným z dôvodov bol aj nedostatok materiálov pre implementáciu edukačnej robotiky pre zvolenú vekovú kategóriu žiakov.

2. METÓDY VÝSKUMU
Hlavným cieľom nášho dizertačného výskumu bol *návrh, implementácia a overenie vzdelávacieho programu pre edukačnú robotiku, čiže integrácia edukačnej robotiky do informatiky na 2. stupni ZŠ.* Vzdelávacím programom tu rozumieme materiály pre pedagógov zahŕňajúce vzdelávacie ciele a rozvíjané kompetencie žiakov, obsahujúce vybranú metodiku a obsah výučby a sú v súlade s platným Štátnym vzdelávacím programom. Preto sme si položili nasledovnú výskumnú otázku: *Aké špecifické ciele, obsah a formu môže mať vzdelávací program pre edukačnú robotiku v predmete informatika pre žiakov vo veku 10 až 12 rokov?* To znamená, že sme navrhovali, implementovali a overovali v praxi jeden z možným scenárov vzdelávacieho programu pre edukačnú robotiku na 2. stupni ZŠ. Vo výskume sme využili ***metódy kvalitatívneho výskumu*** [9, 10]. Keďže sme realizovali iteratívny vývoj aktivít, do ktorého sme aktívne vstupovali, použili sme stratégiu výskum vývojom (*Design based research*). Pri hľadaní odpovede na položenú výskumnú otázku sme zbierali dáta z viacerých zdrojov, a tak sa snažili zabezpečiť i kvalitu nášho výskumu. Vo výskume sme si zvolili nasledujúce metódy zberu a analýzy dát: pološtruktúrovaný skupinový rozhovor, štruktúrovaný rozhovor s otvorenými otázkami, zúčastnené, neštruktúrované pozorovanie a metóda terénnych zápiskov a obsahovú analýzu produktov. Pri analýze získaných dát sme využili i metódu konštantnej komparácie, techniku „vyloženia kariet“, techniku kontrastovania a pod. [10]. V našom výskume sme teda využili viaceré metódy zberu i analýzy dát. Triangulóciou zdrojov i použitých metód sme tak chceli podporiť validitu nášho výskumu. Na výskume sa podieľali prevažne dvaja výskumníci, ktorí aktivity navrhovali, overovali a upravovali. Priebežné výsledky výskumu sme tiež konzultovali s viacerými kolegami a spolupracovali sme i s učiteľkou informatiky, ktorá disponovala s viac než 15-násť ročnou praxou. Tejto učiteľke sme v dostatočnom predstihu dodávali metodické materiály, podľa ktorých následne viedla vyučovaciu hodinu. Na vyučovacích hodinách boli prítomní a zaznamenávali potrebné dáta obaja výskumníci. S uvedenou pani učiteľkou sme viedli aj viaceré rozhovory a konzultovali prípadne otázky či nezrovnalosti. V priebehu výskumu sme pracovali s viac ako 100 žiakmi rozdelenými do niekoľkých skupín. V jednej

skupine sa nachádzalo približne 8 až 14 žiakov. Skupiny sa skladali z dievčat i chlapcov v rôznych pomeroch. Prezentované aktivity sú teda výsledkom štvorročného výskumu, počas ktorých prešli niekoľkými iteráciami úprav.

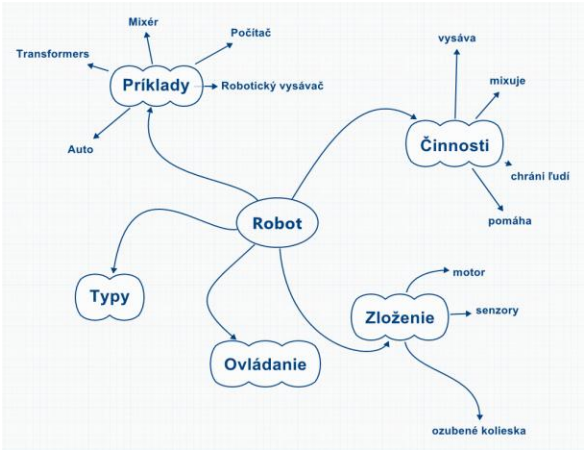
3. AKTIVITY

Pri tvorbe aktivít v rámci vyvíjaného vzdelávacieho programu sme sa pridržiavali ***princípov návrhu učebných činností*** podľa [11], medzi ktoré patria: princíp sústredenia sa na kľúčové pojmy, princíp predchádzajúcich znalostí, princíp kognitívnej náročnosti, ktorý zameriava učebné činnosti k autentickému projektu, princíp aktívneho spracovania, princíp rozmanitosti. Pasch a kolektív sa v [11] domnievajú, že tieto princípy môžu slúžiť ako orientačné vodítko nielen pri plánovaní nových častí vzdelávacieho programu, ale aj pri revízii daného učiva. Do nami vyvíjaného vzdelávacieho programu sme sa systematicky snažili zakomponovať aj ***konštrukcionistické prvky výučby*** [12, 13, 14], medzi ktoré patria: učenie sa robením (angl. *learning by doing*), aktivity s aktívnym zapojením (angl. *hands-on activities*), voľba vlastných produktov, voľba vlastných problémov a tvorba vlastných riešení (angl. *genuine achievement and own solutions, problem finding*), náročná zábava (angl. *hard fun*) a zábavné učenie sa (angl. *playful learning*), digitálne technológie ako stavebný materiál, možnosť robiť chyby a poučiť sa z nich, tímová práca, komunikácia, kolaborácia, rozdeľovanie úloh v tíme a spolupráca na spoločnom riešení. Konštrukcionistické učenie sa tiež utvára príležitosti pre rozmanité reprezentácie vedomostí a učiva a podporuje rozličné typy učebných štýlov [15]. V niektorých aktivitách sme využívali aj ***inštruktívne prvky výučby***, pretože i moderný poznávací proces si vyžaduje oba prístupy, a to inštrukcie i konštruovanie poznatkov. Tieto prístupy sa vzájomne podporujú a sú pevne poprepájané v procese učenia sa [14]. Inštrukcie mnohokrát užitočne predchádzajú konštruovaniu poznatkov a naopak [16]. Navrhovanými aktivitami vo vzdelávacom programe sme sa navyše usilovali zaujať žiakov celej triedy. Rusk a kolektív sa v [13] snažia o to, aby bola robotika rovnako atraktívna pre chlapcov i dievčatá, nie len pre žiakov so záľubou konštruovať autá, ale i pre žiakov so záujmom o umenie, hudbu a rozprávanie príbehov. Preto sme v aktivitách, v ktorých žiaci vytvárali vlastné robotické modely, sledovali ***princípy „tvorivej informatiky pre každého“*** [13], medzi ktoré patria: zameranie sa na tému, kombinovanie umenia a konštrukcie, podnecovanie rozprávania príbehov a organizovanie výstav. Rusk a kolektív v [13] tiež tvrdia, že pri dodržaní týchto princípov sa nám podarí zaujať viacerých žiakov s rôznorodými záujmami.

V nasledujúcej časti stručne charakterizujeme výsledný metodický materiál pre edukačnú robotiku, teda pedagogickú intervenciu, ktorá je výstupnou intervenciou nášho výskumu vývojom pre všeobecnovzdelávací predmet informatika na 2. stupni ZŠ. Tento materiál tvorí 11 aktivít určených pre žiakov piateho a šiesteho ročníka. Každá aktivita je určená na jednu vyučovaciu hodinu a v našej intervencii ju reprezentuje metodický materiál pre vyučujúceho. Každý metodický materiál obsahuje identifikáciu tematického okruhu alebo okruhov aktivity podľa Štátneho vzdelávacieho programu (ďalej len ŠVP) predmetu informatika [17]; ďalej odporúčaný ročník; všeobecné a špecifické vzdelávacie ciele; rozvíjané kompetencie na základe výkonového štandardu podľa ŠVP [17]; požadované vedomosti a zručnosti žiakov na začiatku vyučovacej hodiny; štruktúru vyučovacej hodiny s predpokladaným časovým trvaním jednotlivých úloh; zoznam didaktických pomôcok pre učiteľa a žiaka; pedagogické postupy a priebeh celej vyučovacej hodiny, spoločne so zadaním úloh pre žiakov a odporúčaniami pre učiteľov; navrhované postupy hodnotenia práce žiakov [18] a ďalšie prílohy, ako napr. návody na stavbu robotických modelov, pracovné listy a pod. Pri každej nasledujúcej aktivite neuvádzame úplne všetky kompetencie rozvíjané v danej aktivite, ale len tie, ktoré sú oproti predchádzajúcim aktivitám nové (zatiaľ ešte neuvedené). V priebehu všetkých aktivít žiaci pracovali prevažne v dvojčlenných skupinách.

3.1 Aktivita 1: Čo je robot?

Cieľom úvodnej aktivity je, aby si žiaci utriedili a ujasnili svoje predchádzajúce neformálne znalosti o pojme robot, a to prostredníctvom rozhovoru a spoločnej tvorby myšlienkovej mapy (Obrázok 1.). Žiaci v závere hodiny dokážu na základe vytvorenej mapy sformulovať svoju vlastnú definíciu robota.



Obrázok 1. Myšlienková mapa o pojme robot

Touto aktivitou sa žiaci učia: vlastnou činnosťou objavovať a vytvárať nový význam skúmaného pojmu, vyjasniť si a slovne vyjadriť svoje názory, vyjadrovať a obhajovať svoje názory argumentmi, kriticky posudzovať vlastné názory a názory druhých, kultivovane komunikovať a spolupracovať, sumarizovať a syntetizovať získané informácie.

3.2 Aktivita 2: Vytvorme si vysnívané autíčko

Cieľom tejto aktivity je, aby sa žiaci oboznámili s obsahom robotickej stavebnice LEGO WeDo a jej programovacím prostredím, aby si prostredníctvom stavby atraktívneho modelu (Obrázok 2.) rozvíjali schopnosť postupovať podľa návodu, aby sa oboznámili s prevodom osi otáčania motora v horizontálnej rovine a aby si rozvíjali svoju jemnú motoriku. Cieľom je tiež, aby ovládali svoj model prostredníctvom príkazov *otáčaj motor vpravo* alebo *vľavo, spusti motor na istý čas*, *nastav motoru istú rýchlosť* a *zastav motor*. Žiaci budú tiež využívať príkaz na *spúšťanie zvukov*.

Obrázok 2. Model auta, ktorý žiaci stavali podľa návodu

Touto aktivitou sa žiaci učia: skladať prvky do konštrukcií podľa návodu, skonštruovať prevod osi otáčania motora v horizontálnej rovine o 90°, riešiť problém skladaním príkazov do postupnosti,

používať programovací jazyk na popis riešenia problému, používať vo vlastnom programe zadané príkazy, slovne interpretovať svoj program ostatným, rozvíjať schopnosť zostaviť a poslať správu danému príjemcovi prostredníctvom konkrétneho e-mailového nástroja, spolupracovať v dvojčlennom tíme, efektívne komunikovať svoje myšlienky, tolerovať názory iných.

3.3 Aktivita 3: Aj Ľubka chalúpka na robotickej nôžke vie tancovať

Cieľom tejto aktivity je, aby si žiaci pomocou stavby robotického modelu (Obrázok 3. vľavo) rozvíjali schopnosť postupovať podľa návodu, aby sa zoznámili s ďalším spôsobom prevodu osi otáčania motora z horizontálnej roviny do vertikálnej, a aby si rozvíjali svoju jemnú motoriku. Cieľom tiež je, aby si riešením gradovaných úloh v pracovnom liste prehlbovali vedomosti a skúsenosti s prácou s príkazmi na ovládanie motora a *spúšťanie zvukov* (na Obrázku 3. Vpravo vidíme ukážku jednej z úloh). Ďalším cieľom je, aby si žiaci skupinovou prácou rozvíjali komunikačné a kolaboračné schopnosti. Žiaci sa tu prvýkrát stretnú s príkazmi *čakaj istý čas* a *opakuj*.

Obrázok 3. Model chalúpky, ktorý žiaci stavali podľa návodu (vľavo) a ukážka programu, ktorý žiaci popisovali do pracovného listu (vpravo)

Touto aktivitou sa žiaci učia: skonštruovať prevod osi otáčania motora z horizontálnej roviny do vertikálnej, vytvoriť program podľa písomných inštrukcií, aplikovať pravidlá konštrukcie programu na zostavenie jednotlivých príkazov, písomne i slovne interpretovať postupnosť príkazov, hľadať chybu v postupnosti príkazov a opraviť ju, rozpoznať, aká časť algoritmu sa má vykonať počas cyklu, stanoviť počet opakovaní pomocou hodnoty, zapísať riešenie problému s cyklom pomocou konštrukcií jazyka, interpretovať algoritmy s cyklami, rozvíjať tímovú spoluprácu a kooperáciu.

3.4 Aktivita 4: Môj prvý vlastný pomocník

Cieľom tejto aktivity je, aby si žiaci rozvíjali tvorivé myslenie pri stavbe a programovaní vlastného robotického modelu (ukážky dvoch robotických modelov vidíme na Obrázku 4.). Ďalším cieľom je, aby si žiaci plnením zadania utvrdzovali vedomosti a zručnosti spojené so stavbou a programovaním modelu nadobudnuté na predchádzajúcich hodinách. Cieľom je tiež, aby si skupinovou prácou rozvíjali schopnosti vyjadriť a obhájiť svoj názor a dohodnúť sa v skupine na spoločnom riešení problému.

Obrázok 4. Robotický model postavený s inšpiráciou z návodu (vľavo) a model postavený podľa vlastných predstáv (vpravo)

Touto aktivitou sa žiaci učia: rozvíjať schopnosť porozumieť zadaniu, rozvíjať si tvorivé myslenie pri konštruovaní modelu, pri tvorbe vlastného programu využívať zadané príkazy, rozpoznať, že program funguje nesprávne, hľadať chybu vo vlastnom nesprávne pracujúcom programe a opraviť ju, diskutovať a argumentovať o správnosti riešenia (svojho aj iných), navrhnúť vylepšenie riešenia.

3.5 Aktivita 5: Matúšove vznášadlo z budúcnosti

Cieľom tejto aktivity je, aby si žiaci pomocou stavby jednoduchého robotického modelu rozvíjali schopnosť postupovať podľa návodu, aby sa zoznámili s využitím pohybového senzora v modeli a aby si rozvíjali svoju jemnú motoriku. Ďalším cieľom je, aby sa oboznámili s funkcionalitou pohybového senzora a možnosťami využitia ikony reprezentujúcej pohybový senzor v programovacom prostredí. Žiaci sa prvýkrát stretnú so spojením príkazu *čakaj* s parametrom pohybového senzora. Cieľom tiež je, aby žiaci pomocou úloh v pracovnom liste (na Obrázku 5. vidíme ukážku jednej z úloh) skúmali správanie robotického modelu reagujúceho na pohybový senzor. Cieľom je aj to, aby si skupinovou prácou rozvíjali komunikačné a kolaboračné schopnosti.

Obrázok 5. Ukážka programu z pracovného listu, pomocou ktorého žiaci skúmali správanie robotického modelu

3.6 Aktivita 6: Martinina predstava strateného prepravného prostriedku

Cieľom tejto aktivity je, aby si žiaci pomocou stavby modelu podľa obrázka rozvíjali schopnosť rekonštrukcie neúplných informácií do konkrétneho modelu a aby si pomocou dostavania modelu rozvíjali tvorivé myslenie a jemnú motoriku. Ďalším cieľom je, aby si žiaci prehĺbili vedomosti spojené s prácou s pohybovým senzorom pomocou rôznorodých úloh v pracovnom liste. Cieľom tiež je, aby si žiaci ozrejmili rozdiely medzi vybranými príkazmi (Obrázok 6.) a objavovali príkazy na *výpis do virtuálneho okna*.

Obrázok 6. Príklad zadaných programov v pracovnom liste, kde žiaci popisujú ich hlavné rozdiely

Touto aktivitou sa žiaci učia: rozvíjať schopnosti vytvoriť robotický model iba na základe čiastočných informácií podaných v podobe jeho fotografie, rozvíjať si tvorivé myslenie pri dotváraní robotického modelu podľa zadaných požiadaviek (ako nutnosť využiť motor a pod.), hľadať chybu v zadanom nesprávne fungujúcom programe a opraviť ju, doplniť, dokončiť a modifikovať rozpracované riešenie, rozvíjať schopnosť niesť zodpovednosť za prácu vytvorenú v skupine.

3.7 Aktivita 7: Najlepší pomocník je na svete

Cieľom tejto aktivity je, aby si žiaci rozvíjali tvorivé myslenie pri stavbe a programovaní vlastného robotického modelu (požadované príkazy k programovaniu tohto modelu vidíme na Obrázku 7.). Cieľom tiež je, aby žiaci plnením zadania demonštrovali svoje

vedomosti a zručnosti nadobudnuté na predchádzajúcich hodinách a aby si rozvíjali schopnosť využiť pohybový senzor pri stavbe a programovaní robotického modelu. Ďalším cieľom je, aby žiaci získavali viac skúseností s využívaním pohybového senzora ako parametra na nastavovanie príkazov *rýchlosti motora*, *času točenia motora*, *zvukov* a nastavenia príkazu *čakaj*. Cieľom je aj to, aby si žiaci skupinovou prácou rozvíjali schopnosti vyjadriť a obhájiť svoj názor, dohodnúť sa v skupine na spoločnom riešení problému a niesť zodpovednosť za prácu vytvorenú v skupine.

Obrázok 7. Požadované príkazy k programovaniu vlastného robotického modelu

Touto aktivitou sa žiaci učia: rozvíjať si tvorivé myslenie pri konštruovaní modelu s využitím zadaných súčiastok zo stavebnice, medzi ktoré patrí motor a pohybový senzor, rozpoznať, či program funguje správne alebo nesprávne (svoj aj cudzí).

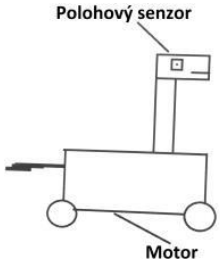
3.8 Aktivita 8: Pohyb modernej korytnačky

Cieľom tejto aktivity je, aby si žiaci pomocou stavby jednoduchého robotického modelu (Obrázok 8. vľavo) rozvíjali schopnosť postupovať podľa návodu, aby sa zoznámili so spôsobom využitia polohového senzora v robotickom modeli a aby si rozvíjali svoju jemnú motoriku. Cieľom tiež je, aby sa žiaci oboznámili s funkcionalitou polohového senzora a s možnosťami využívať ikonu reprezentujúcu polohový senzor pri programovaní robotického modelu (na Obrázku 8. vpravo vidíme ukážku jedného z programov, ktoré žiaci popisovali do pracovného listu). Žiaci sa prvýkrát stretnú so spojením príkazu *čakaj* s parametrom polohového senzora. Ďalším cieľom je, aby žiaci pomocou úloh v pracovnom liste skúmali správanie robotického modelu reagujúceho na polohový senzor. Cieľom je aj to, aby si žiaci skupinovou prácou rozvíjali komunikačné a kolaboračné schopnosti.

Obrázok 8. Model korytnačky (vľavo) a ukážka programu, ktorý žiaci popisovali do pracovného listu (vpravo)

3.9 Aktivita 9: Navrhujeme rozprávkovú krajinu

Cieľom tejto aktivity je, aby si žiaci pútavou formou rozvíjali schopnosť uvažovať o možnostiach a obmedzeniach konštrukcie vlastného robotického modelu vzhľadom na obsah robotickej stavebnice. Ďalším cieľom je, aby si rozvíjali schopnosti uvažovať o možnostiach a obmedzeniach programovania svojho robotického modelu vzhľadom na programovacie prostredie LEGO WeDo. Cieľom je aj to, aby si žiaci pri tvorbe nákresu (Obrázok 9.) a písomného popisu správania svojho robotického modelu rozvíjali tvorivosť a schopnosť realizovať svoje myšlienky tvorivou činnosťou. Ďalším cieľom je, aby celkovou prácou na návrhu robotického modelu demonštrovali svoje doterajšie poznatky a zručnosti nadobudnuté na predchádzajúcich hodinách. Cieľom je tiež, aby si žiaci skupinovou prácou rozvíjali schopnosť plánovať, organizovať svoju prácu a niesť zodpovednosť za prácu vytvorenú v skupine.

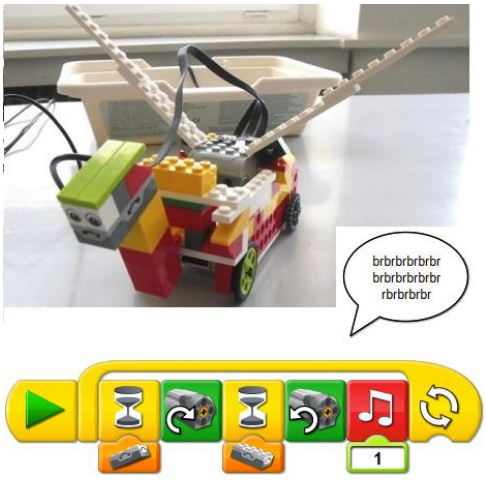


Obrázok 9. Náčrt robotického modelu draka

Touto aktivitou sa žiaci učia: uvažovať o obmedzeniach, ktoré súvisia s riešením úlohy, uvažovať o rôznych alternatívnych riešeniach, použiť konkrétne nástroje editora na tvorbu a úpravu obrázkov, skúmať nové nástroje v konkrétnom editore, používať konkrétne nástroje editora na tvorbu a úpravu textu, ukladať produkt do súboru, otvoriť rozpracovaný produkt zo súboru, rozdeliť si úlohy v tíme, prezentovať svoju prácu.

3.10 Aktivita 10: Budujeme rozprávkovú krajinu

Cieľom tejto aktivity je, aby žiaci realizovali návrh svojho robotického modelu z predchádzajúcej hodiny (Obrázok 10.) a aby si pri tom rozvíjali tvorivé myslenie. Ďalším cieľom je, aby žiaci v priebehu konštruovania a programovania svojho modelu demonštrovali doterajšie vedomosti a zručnosti nadobudnuté v priebehu predchádzajúcich hodín. Cieľom je tiež, aby v rámci skupinovej práce využívali silné stránky členov skupiny na dosiahnutie spoločného cieľa a aby si rozvíjali schopnosť niesť zodpovednosť za prácu tvorenú v skupine.



Obrázok 10. Výsledný robotický model draka (jeho náčrt vidíme na Obrázku 9.) a program na jeho ovládanie

Touto aktivitou sa žiaci učia: transformovať vlastný návrh do skutočného robotického modelu a programu na jeho ovládanie, rozvíjať si tvorivé myslenie pri konštruovaní modelu s využitím vybraných súčiastok zo stavebnice ako je motor a jeden zo senzorov, využívať písomný popis svojho návrhu na tvorbu vlastného programu, rozpoznať, či program funguje správne alebo nie (i na základe slovného popisu v návrhu).

3.11 Aktivita 11: Rozprávková krajina aj pre ostatných

Cieľom tejto aktivity je, aby si žiaci zhodnotili postup práce na vlastnom robotickom modeli, aby zhodnotili svoju prácu vrátanie svojich silných a slabých stránok a aby zhodnotili obmedzenia konštrukcie svojho modelu a programu. Ďalším cieľom je, aby žiaci porovnali svoje prvé predstavy o výzore modelu a popise programu s jeho reálnou konštrukciou a konkrétnym programom a zdôvodnili, prečo ich nemohli realizovať. Cieľom je tiež, aby si žiaci skupinovou prácou rozvíjali schopnosť diskutovať a obhajovať svoje názory.

Touto aktivitou sa žiaci učia: používať konkrétne nástroje editora na tvorbu a úpravu prezentácií, rozvíjať si kritickú reflexiu svojich vedomostí a ich hraníc, orientovať sa v konkrétnej štruktúre priečinkov, použiť nástroj na manipuláciu so súbormi a priečinkami, zostaviť a posielať správu danému príjemcovi prostredníctvom e-mailového nástroja, pripojiť prílohu správy, zefektívňovať vlastný časový manažment.

4. ZÍSKANÉ POZNATKY Z PROCESU VÝVOJA AKTIVÍT

V priebehu prvej etapy výskumu sme na základe analýzy dostupných vedeckých prác zistili, že výučba s edukačnou robotikou v školskom prostredí prebiehala prevažne konštrukcionistickým či konštruktivistickým spôsobom. Preto sme sa rozhodli viesť výučbu našich aktivít konštrukcionistickým spôsobom a nechávali sme žiakom priestor na skúmanie robotickej stavebnice i programovacieho prostredia. Na základe našich predchádzajúcich skúseností s vývojom vzdelávacieho programu pre edukačnú robotiku na prvom stupni ZŠ sme umožnili žiakom tvoriť robotické modely podľa vlastnej fantázie. Žiaci však:

- často kládli neuskutočniteľné požiadavky na výzor i správanie svojich robotických modelov,
- nemali nacvičenú prácu s časom, ani efektívnu kolaboráciu,
- v rámci skupiny nespolupracovali, nereagovali na výzvy svojich spolužiakov či učiteľov pracovať spoločne.

Vo všeobecnosti skupiny žiakov pracovali s rozdielnou efektívnosťou. Jedna skupina sa viac zamerala na vytváranie príbehu o svojom modeli, ďalšia na jeho konštruovanie a iná zas na jeho programovanie. Napriek drobným neúspechom skupiny žiakov pracovali s nadšením. Vzhľadom na rozdielnosť schopností a zručností jednotlivých žiakov bolo pomerne náročné vytvoriť vhodný druh aktivity, navrhnúť rozdelenie do skupín a odhadnúť čas, ktorý žiaci potrebovali na prácu s robotickými stavebnicami. Preto sme sa rozhodli zaradiť do vyvíjaného vzdelávacieho programu také aktivity, počas ktorých by sa žiaci mohli zoznámiť s rôznymi typmi konštrukcie robotických modelov a rôznymi možnosťami ich programovania ešte pred prácou na svojich prvých vlastných modeloch.

V druhej etape výskumu sme najskôr vytvorili špecifikáciu vzdelávacích cieľov pre edukačnú robotiku [19]. Proces jej tvorby nám pomohol sformulovať rôznorodé typy úloh na objasnenie a hlbšie pochopenie jednotlivých konštrukčných a programátorských konceptov. Ďalej sme overovali vyvíjaný vzdelávací program, ktorý vyučovala vyučujúca v rámci svojich bežných hodín informatiky. V ňom sme pre žiakov najskôr pripravili aktivity, v ktorých sa zoznámili s robotickou stavebnicou a softvérom na jej ovládanie. Potom si žiaci skúšali stavbu rôznych modelov podľa návodov s inštrukciami a ich programovanie podľa úloh v pracovných listoch. Tiež sme žiakom poskytli priestor na objavovanie fungovania pohybového senzoru a na objavovanie rôznych možností jeho programovania. Žiaci však samostatne len so značnými ťažkosťami zisťovali, ako senzor funguje, alebo akým spôsobom ho môžu použiť na ovládanie modelu. Preto sme v nasledujúcej iterácií výskumu upravili spôsob zoznamovania sa so senzormi, kde sme zaradili aj frontálnu formu výučby a pridali viacero inštruktívnych prvkov. Ďalej sme upravili i znenie návrhu záverečného projektu, ktorý sme zamerali na tvorbu výzoru modelu a na písomný popis jeho správania. Popis príbehu, v ktorom vystupuje model, sme zaradili medzi rozširujúce úlohy.

V tretej etape výskumu sme sa zamerali na návrh rozličných typov úloh v rámci našich aktivít a vytváranie ich hodnotenia [18]. V tejto etape výskumu sme tiež zisťovali, koľko úloh je pre žiakov primerané pripraviť na danú aktivitu a akých chýb sa dopúšťajú pri tvorbe svojich modelov. Žiaci sa najčastejšie robili **chyby pri stavbe** vlastného robotického modelu, v ktorom mali použiť konkrétne komponenty stavebnice, napr. pri prvom vlastnom modeli to bol motor. **Pri programovaní modelov** sa žiaci najčastejšie dopúšťali chýb v úlohe, kedy mali do pracovného listu písomne popísať zadaný program. Pri vytváraní programu podľa zadania žiaci tvorili nesprávne postupnosti príkazov alebo im v týchto postupnostiach chýbali niektoré príkazy. Pri tvorbe vlastných programov žiaci nevyužívali všetky požadované príkazy. Aj keď vedeli, ktoré príkazy majú použiť, pokiaľ im nekorešpondovali s ich vlastnou predstavou o správaní modelu, tak ich nevyužili. Žiaci ďalej robili chyby nielen **pri vysvetľovaní programov** z pracovných listov, ale aj pri vysvetľovaní svojich vlastných programov. Často vytvárali zaujímavé programy, avšak pri záverečnom prezentácií svojej práce len spustili program bez podrobnejšieho správania svojho modeli. Na základe viacerých vyššie uvedených zistení sme vytvorili výstižnejšie a pre žiakov zrozumiteľnejšie zadania, upravili sme i rozsah úloh v jednotlivých aktivitách, a taktiež sme modifikovali aj všetky metodické materiály pre učiteľov.

V štvrtej etape výskumu sme upravili poradie jednotlivých aktivít, sústredili sme sa na reakcie vyučujúcej a žiakov na náš metodický materiál. Táto etapa potvrdila nasýtenosť dát nášho výskumu a obsahuje finálne úpravy. Vo výskumnej vzorke sa ako v predchádzajúcich iteráciách objavila skupina žiakov, ktorá sa nedokázala dohodnúť na jednom spoločnom funkčnom robotickom modeli. Žiaci však dokázali svoju prácu plánovať, analyzovať, vylepšovať, a tiež zhodnocovať jej výsledky, čo považujeme za dôležitý prínos nášho vzdelávacieho programu.

5. ZÁVER

V tomto článku sme popísali metódy a nástroje nášho pedagogického výskumu. Ako výsledok nášho skúmania uvádzame konkrétne aktivity, ktoré sme vytvorili a overili počas niekoľkých etáp. Zdôrazňujeme, že nejde o jediný, ale o jeden z možných prístupov integrácie edukačnej robotiky do vyučovania povinného predmetu informatika na bežných školách. Aktivity teda nie sú určené na voľnočasové krúžky, či pre mimoriadne nadané deti, alebo pre inak selektovaný výber študentov. V rámci hlavných cieľov nášho výskumu sme pomocou uvedených aktivít žiakom priblížili a ozrejmili viaceré programátorské koncepty, rozvíjali sme u nich rôzne informatické zručnosti ako aj zručnosti potrebné pre život v 21. storočí a využívali sme pritom ako nástroj robotické stavebnice. Okrem vývoja samotného obsahu týchto aktivít, sme získavali aj veľmi vzácne poznatky o vhodnej štruktúre aktivít a vycibrili sme si odhad na nastavenie náročnosti učiva pre

edukačnú robotiku v primárnom vzdelávaní. Vytvorená štruktúra samotných aktivít, ale aj poradie medzi nimi navzájom, sa nám overila ako najvhodnejšia na predávanie poznatkov a zároveň dávala aj žiakom priestor na učenie sa objavovaním.

BIBLIOGRAFICKÉ ODKAZY

[1] BARKER, B.S., ANSORGE, J. Robotics as means to increase achievement scores in an informal learning environment. In: *Journal of Research on Technology in Education*. 2007, vol. 39, no. 3, p. 229-243. ISSN 1539-1523.

[2] NUGENT, G. et al. The effect of 4-H robotics and geospatial technologies on science, technology, engineering, and mathematics learning and attitudes. In: *World conference on educational multimedia, hypermedia and telecommunications*. Vienna: Chesapeake; VA: AACE, 2008, p. 447–452. ISSN 1055-8896.

[3] NUGENT, G. et al. The use of digital manipulatives in k-12: robotics, GPS/GIS and programming. In: *FIE'09 Proceedings of the 39th IEEE international conference on Frontiers in education conference*. 2009, p. 1-6. ISBN 978-1-4244-4715-2.

[4] WHITTIER, L.E., ROBINSON, M. Teaching evolution to non-English proficient students by using lego robotics. In: *American Secondary Education*. 2007, vol. 35, no. 3, p. 19-28. ISSN 00031003.

[5] SULLIVAN, F.R. Robotics and science literacy: Thinking skills, science process skills and systems understanding. In: *Journal of Research in Science Teaching*, 2008, vol. 45, no.3, p. 373-394. DOI 10.1002/tea.20238.

[6] MITNIK, R., NUSSABAUM, M., SOTO, A. An autonomous educational mobile robot mediator. In: *Autonomous Robots*, 2008, vol. 25, no. 4, p. 367-382. ISSN 0929-5593.

[7] MUBIN, O., et al. A review of the applicability of robots in education. In: *Journal of Technology in Education and Learning*, 2013, vol. **1**. DOI 10.2316/Journal.209.2013.1.209-0015.

[8] GURA, M *Getting Started with LEGO Robotics: A Guide for K-12 Educators*. United States of America: International Society for Technology in Education, 2011. 137 p. ISBN 978-1-56484-298-5.

[9] CRESWELL, J. Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research. 3 vyd. New Jersey: Pearson Education, 2008. ISBN-13: 978-0-13-207208-0.

[10] ŠVAŘÍČEK, R. et al. *Kvalitativní výzkum v pedagogických vědách*. Praha : Portál, 2007. ISBN 978-80-7367-313-0.

[11] PASCH, M. et al. *Od vzdělávacího programu k vyučovací hodině: jak pracovat s kurikulem*. 2. Vyd. Praha: Portál, 2005. ISBN 80-7367-054-2.

[12] PAPERT, S. *The Eight Big Ideas of the Constructionist Learning Laboratory*. Unpublished internal document. South Portland, Maine, 1999. Cited in STAGER, G. Papertian Constructionism and the Design of Productive Contexts for Learning. Plennary Session Paper – EuroLogo X, Warsaw, Poland, 2005. [cit. 2017-1-20]. Dostupné tiež na: http://www.stager.org/articles/eurologo2005.pdf

[13] RUSK, et al. New Pathways into Robotics: Strategies for Broadening Participation. In: *Journal of Science Education and Technology*, 2008, vol. 17, no. 1, p. 59-69. DOI 10.1007/s10956-007-9082-2.

[14] KABÁTOVÁ, M. *Konštrukcionistický prístup vo vyučovaní robotiky v príprave budúcich učiteľov* [dizertačná práca]. Bratislava: Fakulta matematiky, fyziky a informatiky, Univerzita Komenského v Bratislave, 2010.

[15] KAFAI, Y., RESNICK, M. *Constructionism in practice*. Mahwah, New Jersey: Lawrence Erlbaum Asociates, 1996. ISBN 0-8058-1985-1.

[16] FEURZEIG, W. Demystifying Constructionism. CLAYSON, J., KALAS, I. (Eds.) *Proceedings of Constructionism 2010*. Paris: The American University of Paris, 2010. ISBN 978-80-89186-66-2.

[17] ŠPU: 2015, *Inovovaný štátny vzdelávací program: Informatika. Vzdelávacia oblasť: Matematika a práca s informáciami. Príloha ISCED 2*. Bratislava: Štátny Pedagogický Ústav. [cit. 2017-1-27]. Dostupné tiež na: http://www.statpedu.sk/sites/default/files/dokumenty/inovovany-statny-vzdelavaci-program/informatika_nsv_2014.pdf

[18] VESELOVSKÁ, M., MAYEROVÁ, K. Assessing robotics learning at lower secondary school. In: *Information and Comunication Technology in Education 2015*. Ostrava: Pedagogical Faculty, University of Ostrava, 2015, s. 240-247. ISBN 978-80-7464-763-5.

[19] KALAŠ, I., MAYEROVÁ, K., VESELOVSKÁ, M. Špecifikácia vzdelávacích cieľov pre edukačnú robotiku. In: *Journal of Technology & Information Education* 2014, vol. 6, no. 1, s. 30-44.

Používateľské rozhranie – dôležitý prvok pri vyučovaní programovania

User Interface - an important element in teaching programming

Patrik Voštinár
Katedra matematiky FPV UMB
Tajovského 40
974 01 Banská Bystrica
Slovensko
patrik.vostinar@umb.sk

ABSTRACT

The paper deals with procedures, which should be created before programming of applications. The main goal of the paper is to describe the documents that teachers of informatics could start to use within programming teaching at all educational stages. In the paper three documents (Feasibility Study, Software Requirements Specification and Design Document), which should be a part of development of specific desktop, mobile or web application, are described. A separate chapter is devoted to the environment Pencil, which is suitable for creating design of screens for various platforms. In the end of the paper there is a chapter that describes experience with the use of such documents in educational process.

Keywords

User interface design, wireframe, Pencil, programming

ABSTRAKT

Článok sa zaoberá postupmi, ktoré majú byť vytvorené už pred programovaním aplikácií. Hlavným cieľom článku je opis dokumentov, ktoré by mohli začať používať učitelia informatiky pri výučbe programovania na všetkých stupňoch vzdelávania. V článku sú opísané tri dokumenty (štúdia uskutočniteľnosti, špecifikácia požiadaviek na softvér a návrhový dokument), ktoré by mali byť súčasťou vývoja konkrétnej desktopovej, mobilnej alebo webovej aplikácie. Samostatná kapitola je venovaná prostrediu Pencil, ktoré je vhodné pre vytvorenie vzhľadu obrazoviek pre rôzne platformy. V závere článku je kapitola, ktorá opisuje skúsenosti s použitím takýchto dokumentov vo vyučovacom procese.

Kľúčové slová

Používateľský vzhľad, wireframe, Pencil, programovanie

1. ÚVOD

S programovaním aplikácií a webových stránok sa stretávajú už žiaci na základných školách. Výučba väčšinou prebieha tak, že učiteľ ukáže, respektíve povie žiakom, čo a ako budú vytvárať. Analogicky ako na základných školách, prebieha výučba aj na stredných a vysokých školách. Absolventi škôl, ktorí si zvolia programátorské povolanie, alebo sa chcú venovať webovému dizajnu, sú po príchode do praxe prekvapení, že sa musia zoznamovať s postupmi, ktoré predchádzajú samotnému vývoju. Konkrétne sa jedná o návrhové dokumenty, ktoré určujú, ako sa bude aplikácia správať a ako bude vyzerať. Vhodné je, keď si žiaci už na základných školách osvoja tieto postupy. Učiteľ má vytvoriť a ukázať žiakom, čo všetko musia urobiť ešte pred samotným vývojom. Na základnej škole je postačujúce, keď si žiaci už pred tvorbou aplikácie alebo webovej stránky vytvoria návrh dizajnu. Na stredných a vysokých školách môžu k návrhu dizajnu študenti pripojiť ďalšie dokumenty s požiadavkami na aplikáciu.

2. DOKUMENTY POTREBNÉ K VÝVOJU

Firmy zaoberajúce sa vývojom aplikácií vytvárajú pred samotným vývojom aplikácií dokumenty, podľa ktorých sa programátori riadia [1]:

- **Štúdia uskutočniteľnosti** (Feasibility Study) – V prípade, že existuje zámer vytvoriť nový, alebo modernizovať existujúci informačný systém, medzi prvé kroky patrí spracovanie analýzy s názvom štúdia uskutočniteľnosti. Cieľom tohto dokumentu je zistiť, či sa môže projekt vytvoriť s danými obmedzeniami, rizikami a či to je vôbec ekonomicky výhodné.
- **Špecifikácia požiadaviek na softvér** (Software Requirements Specification) – V tomto dokumente sú opísané základné informácie o aplikácií – všeobecný popis systému, funkčné a mimo-funkčné požiadavky (opis chovania aplikácie).
- **Návrhový dokument** (Design Document) – Opisuje, ako by mal byť softvér implementovaný, aby vyhovoval požiadavkám. Návrhový dokument obsahuje napríklad návrh databázy, dátových štruktúr a vzhľad obrazoviek (wireframe).

3. ŠTÚDIA USKUTOČNITEĽNOSTI

Dokument štúdia uskutočniteľnosti posudzuje okrem finančných a ekonomických aspektov návrh projektu z viacerých hľadísk, významných pre realizáciu projektu. Účelom tohto dokumentu je preukázať efektívnosť a uskutočniteľnosť navrhovaného projektu.

Dokument by mal mať nasledujúcu štruktúru:

- Úvod – základné informácie o projekte.
- Začiatočný stav a zdôvodnenie realizácie projektu.
- Analýza potrieb.
- Harmonogram realizácie.
- Technické a technologické riešenie projektu.
- Finančná a ekonomická analýza.
- Analýza rizík.
- Zhodnotenie projektu.

4. POŽIADAVKY NA SOFTVÉR

Požiadavky predstavujú prianie budúceho používateľa. Opisujú funkcie a vlastnosti, ktoré má budúci softvér obsahovať. Neurčujú však, ako bude softvér vytvorený (aký programovací jazyk a nástroje sa použijú).

Požiadavky delíme na [2]:

- **Funkčné požiadavky** (Functional requirements) – určujú chovanie systému.
- **Mimo-funkčné požiadavky** (Non-functional requirements) – špecifikujú vlastnosti a obmedzenia systému.

Príklad funkčných požiadaviek:

- Používateľ sa musí registrovať, aby mal prístup k obsahu stránky.
- Aplikácia bude overovať emailovú schránku odoslaním potvrdzovacieho emailu pri registrácií.
- Systém automaticky odhlási používateľa po 5 minútach nečinnosti.

Príklad mimo-funkčných požiadaviek:

- Softvér musí byť použiteľný pri záťaži 10 000 používateľov.
- Na komunikáciu medzi serverom a klientom bude použitý protokol XY.
- Softvér podporuje iba jeden jazyk – anglický.

Súčasťou požiadaviek na softvér sú aj tzv. používateľské príbehy (User-stories). Tieto príbehy opisujú, čo všetko môžu používatelia robiť v aplikácií.

Napríklad, aplikácia bude podporovať dvoch používateľov – administrátora a bežného používateľa. V dokumente s požiadavkami určíme, čo všetko môže administrátor a bežný používateľ robiť v aplikácií:

- Ako administrátor môžem vidieť detaily registrovaných používateľov.
- Ako administrátor môžem zablokovať príspevky, alebo kontá používateľov.
- Ako používateľ sa môžem registrovať do aplikácie a následne sa prihlásiť.
- Ako používateľ si môžem preposlať nové heslo, ak zabudnem staré heslo.

5. NÁVRHOVÝ DOKUMENT

Návrhový dokument musí obsahovať tzv. wireframy, ktoré definujú textový a grafický obsah. Určujú rozmiestnenie funkčných prvkov, nadpisy, texty a môžu definovať aj akcie, ktoré sa vykonajú po stlačení napr. tlačidla.

Najjednoduchší a najlacnejší, aj keď nie príliš vhodný spôsob pre návrh dizajnu aplikácie, je papier a ceruzka. Medzi hlavné nevýhody tohto spôsobu vytvárania je, že v prípade zmeny musíme nakresliť nový návrh a pomocou papiera nevieme vygenerovať použiteľný kód.

5.1 Programy na tvorbu obrazoviek

V prípade, že nechceme použiť na tvorbu návrhu obrazoviek papier a pero, na výber máme množstvo špeciálnych programov. Medzi najdostupnejší program patrí skicár, ktorý sa nachádza na každom počítači s operačným systémom Windows. Vhodnejšie dizajnérske programy sa nachádzajú v tabuľke 1.

Tabuľka 1. Programy na tvorbu obrazoviek

Program	Cena
Pencil[1]	bezplatný program
Maqetta[2]	bezplatný program
Cacoo[3]	bezplatná verzia pre študentov a učiteľov
Balsamiq Mockups[4]	7 dní skúšobná verzia, platená verzia 89$/licencia
Mockinbird[5]	5 dňová skúšobná verzia, platená verzia 12€/mesiac
Wireframe Sketcher[6]	14 dní skúšobná verzia, platená verzia 99$/licencia

5.2 Pencil

Pencil je bezplatný open-source program na tvorbu grafických návrhov. Je dostupný pre operačné systémy Windows, Linux a MacOS. Stiahnuť si ho môžeme ako desktopovú aplikáciu, alebo rozšírenie pre webový prehliadač Firefox [3].

Pencil umožňuje vytvárať návrhy pre desktopové, webové a mobilné aplikácie, ako je možné vidieť na obrázku 1.

Komponenty, ktoré môžeme použiť rozdeľujeme na:

- **Bežné tvary** – text, kruh, obrázok, čiara, mnohouholníky, atď.
- **Základné webové prvky** – html text, hypertextový odkaz, nadpis h1-h5, tabuľka.
- **Desktopové widgety** – záložky, tlačidlá, prepínače, zaškrtávacie tlačidlá, zoznamy, atď.
- **Vývojový diagram** – šípky, podprogramy, cykly, podmienený výraz, atď.
- **Android widgety** – telefón, tablet, Status Bar, Action Bar, android tlačidlá (stlačené, nestlačené), indikátor priebehu, ikonka, klávesnica, atď.
- **iOS widgety** – iPhone, iPad, iOS tlačidlá, prvky, atď.

[1] http://pencil.evolus.vn/

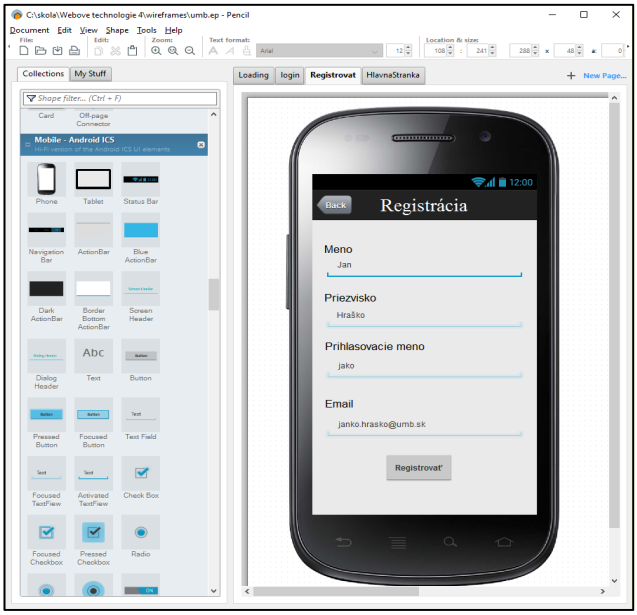
[2] http://maqetta.org/

[3] https://cacoo.com/

[4] https://balsamiq.com/products/mockups/

[5] https://gomockingbird.com/

[6] http://wireframesketcher.com/

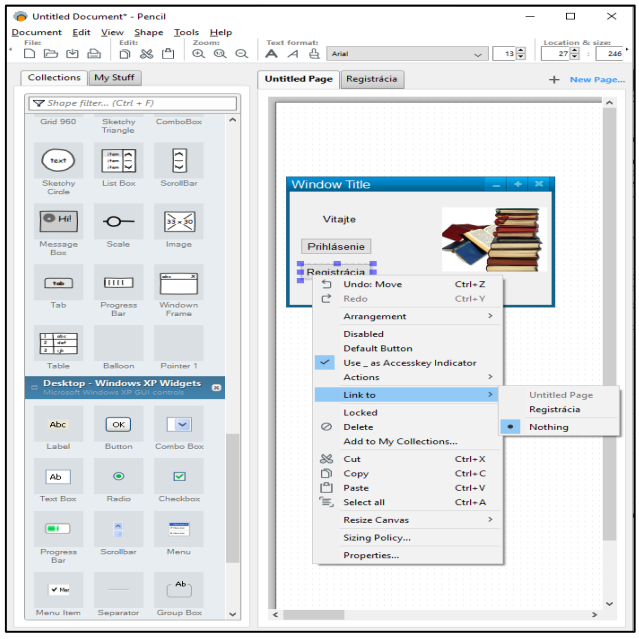


Obrázok 1. Aplikácia Pencil - ukážka mobilnej aplikácie

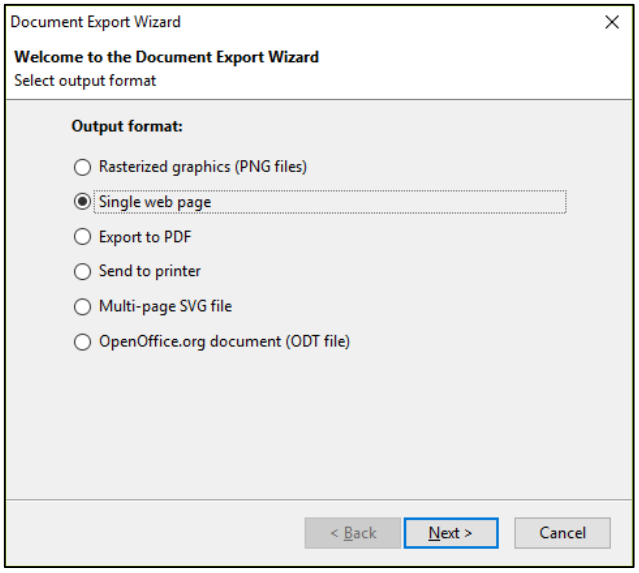
Okrem vstavaných komponentov, program Pencil umožňuje stiahnuť a doinštalovať ďalšie komponenty.

Niektorým komponentom, ako sú napr. tlačidlá môžeme nastaviť akciu, ktorá sa vykoná v prípade ich stlačenia – napr. spustí sa ďalšia obrazovka s inými komponentami (pozri obrázok 2). V prípade použitia komponentu hypertextového odkazu sa spustí webový prehliadač s daným odkazom.

Po dokončení obrazoviek si používateľ vyberie alternatívu, v akej forme chce prezentovať tieto obrazovky. Na výber má exportovanie obrazoviek do PNG obrázkov, vytvorenie jednoduchej webovej stránky, exportovanie do formátu PDF, SVG alebo do OpenOffice dokumentu, ako je vidieť na obrázku 3.



Obrázok 2. Aplikácia Pencil - ukážka desktopovej aplikácie



Obrázok 3. Aplikácia Pencil - možnosti exportovania

6. POUŽITIE PRI VÝUČBE

V kapitole 2 je spomenuté, že firmy zaoberajúce sa vývojom softvéru vytvárajú zvyčajne ešte pred samotným vývojom tri dokumenty: štúdia uskutočniteľnosti, špecifikáciu požiadaviek na softvér a návrhový dokument. Absolventi škôl sú počas ich praxe zaskočení existenciou takýchto dokumentov. Aj z tohto dôvodu sa autor rozhodol implementovať tieto dokumenty do vyučovacieho procesu predmetov Programovanie 3 a predmetu Webové technológie 4.

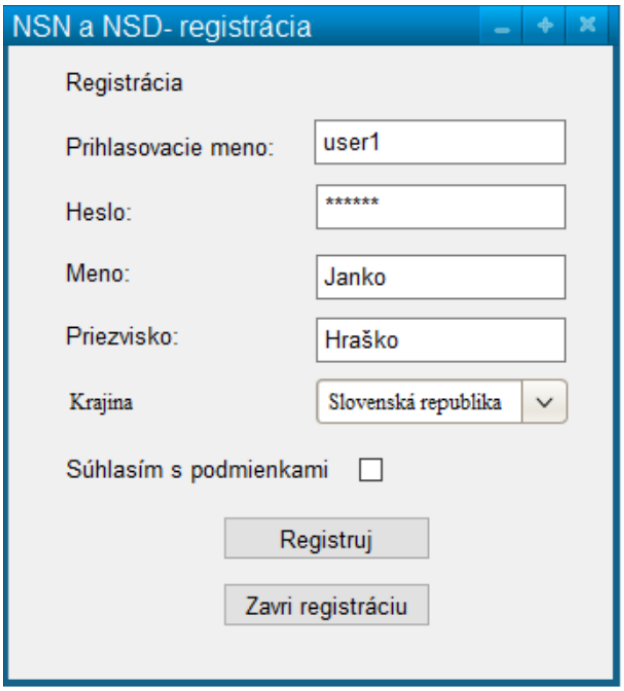
6.1 Implementovanie dokumentov do vyučovacieho procesu – zadanie učiteľa

V predmete Programovanie 3 autor vytvoril stručnejšiu formu špecifikácie požiadaviek a návrhový dokument. Tieto dokumenty obsahovali presne stanovené požiadavky na funkcionalitu a vzhľad aplikácie, ktorú mali študenti za úlohu vytvoriť, ako záverečnú prácu.

Návrhový dokument obsahoval obrazovku:

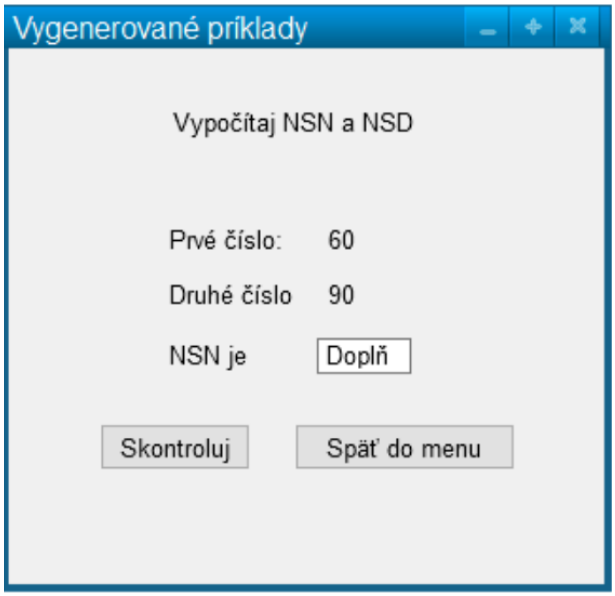
- prihlásenie,
- registrácia,
- dialógové okno pre úspešné alebo neúspešné prihlásenie,
- hlavné menu aplikácie, ktoré sa zobrazí po prihlásení,
- pomocník - postup riešenia a ukážka príkladov,
- vybratie typu príkladu,
- vygenerované náhodné príklady,
- dialógové okno oznamujúce, či je výsledok správny alebo nesprávny.

Na obrázku 4 je ukážka zadania konkrétnej obrazovky, ktorú mali študenti naprogramovať. V tomto prípade je to obrazovka, kde sa používatelia aplikácie musia zaregistrovať, aby mohli vstúpiť do aplikácie.



Obrázok 4. Ukážka časti zadania (obrazovky) - registrácia používateľa

Obrázok 5 zobrazuje príklad na úlohu najväčší spoločný násobok. Používateľ aplikácie musí doplniť výsledok a stlačiť tlačidlo Skontroluj.



Obrázok 5. Obrazovka pre výpočet NSN

Študenti na základe takéhoto zadania vedeli jednoznačne naprogramovať aplikáciu. Učiteľovi stačilo pri záverečnom hodnotení skontrolovať, nakoľko študenti splnili zadanie, či dodržali zarovnania, texty, názvy obrazoviek a funkcionalitu.

V prípade nepoužitia takýchto dokumentov závisí od učiteľa, ako sformuluje zadanie. Väčšinou sú však žiaci neistí, nevedia, čo všetko musí obsahovať ich aplikácia, či sa bude učiteľovi páčiť alebo, či na niečo nezabudli. Učiteľ ich zvyčajne musí častejšie kontrolovať, alebo konzultovať priebežné riešenie.

6.2 Implementovanie do vyučovacieho procesu – zadanie študentov

V predmete Webové technológie si študenti vytvorili dokumenty – špecifikácia požiadaviek a návrhový dokument sami. V úvodných hodinách si študenti premysleli mobilné aplikácie, ktoré budú vyvíjať a následne ich odprezentovali pred spolužiakmi a učiteľom. Štúdiu uskutočniteľnosti teda nemuseli vytvárať, nakoľko ju odkomunikovali ústne. Po odsúhlasení témy mobilnej aplikácie študenti vytvorili dokumenty špecifikácie požiadaviek na softvér a návrhový dokument. Po konzultácií s učiteľom študenti začali pracovať na svojich aplikáciách.

Príklad špecifikácie požiadaviek, ktoré vytvoril študent:

Informácie o projekte

Pomocou mobilnej aplikácie si jej používatelia môžu prečítať úvodníky k najnovším článkom spravodajských portálov o Banskej Bystrici priamo v prostredí OS Android bez toho, aby museli navštíviť ich webové stránky. Úvodník článku, ktorý ich zaujme, slúži ako odkaz na článok na webovej stránke portálu, na ktorej si ho môže používateľ prečítať celý pomocou webového prehliadača. Pomocou menu aplikácie si používateľ môže vybrať, ktoré z ponúkaných portálov bude sledovať, počet zobrazovaných úvodníkov na jednej stránke, jazykovú verziu a vybrať si medzi svetlým a tmavým vzhľadom aplikácie.

V časti požiadavky na systém, študent napríklad uviedol nasledujúce požiadavky na funkcionalitu:

P_002 – Odoberanie spravodajských zdrojov

Spravodajské zdroje sú odoberané, keď je zariadenie pripojené na internet pomocou verejne prístupných RSS kanálov, ktoré na svojich webových stránkach poskytujú. Článok, ktorý neposkytnú vo svojom RSS kanále sa v aplikácii nezobrazí.

FP_001 – Jazyková podpora

Systém podporuje dva jazyky – Slovenský (SJ) a Anglický (AJ). Systém určí, ktorý jazyk sa má použiť podľa jazyka použitého v nastaveniach zariadenia. V prípade, že jazyk zariadenia je iný ako SJ a AJ, tak sa použije AJ. Používateľ aplikácie má možnosť v nastaveniach aplikácie zmeniť používaný jazyk za iný.

V kapitole nefunkčné požiadavky dokument obsahoval napríklad nasledujúce požiadavky:

01_Výkon

Nie sú žiadne požiadavky na výkon mobilných zariadení.

02_Dostupnosť

Aplikácia je dostupná v obchode Google Play.

03_Minimálna podporovaná verzia OS Android

Aplikácia je dostupná pre všetky zariadenia od verzie OS Android 4.0 Ice Cream Sandwich..

Návrhový dokument obsahoval všetky obrazovky, ktoré sa v aplikácií môžu vyskytnúť. Na obrázku 6 je zobrazená obrazovka, ako bude vyzerať hlavné okno aplikácie.

Obrázok 6. Wireframe študentského zadania

7. ZÁVER

Hlavným cieľom príspevku bolo prezentovať dokumenty a nástroje, ktoré sú vhodné pre učiteľov pri vyučovaní programovania. Aplikácia takýchto dokumentov môže vzbudiť väčší záujem o preberanú látku a zároveň môže pripraviť žiakov na budúce povolanie softvérového inžiniera. V článku sú spomenuté viaceré nástroje, ktoré je vhodné používať pri vytváraní obrazoviek – wireframov. Samostatná kapitola je venovaná možnému použitiu takýchto dokumentov vo vyučovacom procese. Každý učiteľ si vyberie alternatívu, ktorá mu viac vyhovuje – či vytvorí takéto dokumenty a študenti následne podľa nich zostavia výsledné aplikácie, alebo nechá tvorbu spomínaných dokumentov na študentov. Tieto dokumenty používajú firmy zaoberajúce sa vývojom aplikácií, vo vyučovacom procese však nemusia študenti pracovať so všetkými dokumentami. Učiteľ sa rozhodne, či stačí, že študenti budú pracovať s návrhovým dokumentom, alebo budú pracovať aj s ďalšími dokumentami.

BIBLIOGRAFICKÉ ODKAZY

[1] ROUDENSKÝ, P., HAVLÍČKOVÁ, A. *Řízení kvality softwaru*. Brno : Computer Press, 2013. 208 s. ISBN 978-80-251-3816-8.

[2] ARLOW, J., NEUSTADT, I. *UML 2 a unifikovaný proces vývoja aplikací.* Brno : Computer Press, 2007. 568 s., ISBN 978-80-251-1503-9.

[3] Stránka Pencil [online]. [cit. 2017-01-12]. Dostupné na internete: http://pencil.evolus.vn.

Skúsenosti z úvodného kurzu programovania v Pythone na gymnáziu

Experience from programming introduction class in Python at upper secondary school

Ľudmila Jašková
KZVI FMFI UK
Mlynská dolina
842 48 Bratislava
Slovensko
jaskova@fmph.uniba.sk

Eva Mészárosová
KZVI FMFI UK
Mlynská dolina
842 48 Bratislava
Slovensko
meszarosova@fmph.uniba.sk

Michal Winczer
KZVI FMFI UK
Mlynská dolina
842 48 Bratislava
Slovensko
winczer@fmph.uniba.sk

ABSTRAKT

V tomto príspevku opisujeme naše skúsenosti s používaním učebného materiálu pre úvodný kurz programovacieho jazyka Python na strednej škole. Učebný materiál sme skúšali v dvoch školách s troma skupinami žiakov. Zamerali sme sa skôr na opis reakcií žiakov, prípadne zaujímavých situácií, ktoré sa počas vyučovania vyskytli než na hodnotenie samotného materiálu.

ABSTRACT

In presented paper we describe our experience using methodology of introductory programming class in Python in upper secondary school. We ran experiment in three groups of pupils. We describe how pupils reacted during the class. We do not evaluate the methodology.

Kľúčové slová

Python. Úvodný kurz programovania. Gymnázium, Korytnačia grafika.

Keywords

Python. Introductory programming class. Upper secondary school, Turtle graphics.

1. ÚVOD

Základy programovania sú dnes neoddeliteľnou súčasťou ŠVP v predmete informatika [1]. Voľba programovacieho jazyka, ktorý by sa mal vo vyučovaní informatiky na strednej škole používať a bol všeobecne akceptovaný, nie je jednoznačná. Vplýva na ňu viacero faktorov. Spomeňme aspoň niektoré: dostupnosť, existencia a jednoduchosť vývojového prostredia, jednoduchosť jazyka, rýchlosť a zložitosť vytvorenia aplikácie. Programovací jazyk Python vyhovuje viacerým z týchto a aj eventuálnym ďalším kritériám. Vďaka tomu sa v posledných rokoch teší stále väčšej obľube ako prvý programovací jazyk. Existuje mnoho materiálov opisujúcich jazyk Python, väčšina z nich je určená najmä profesionálnym programátorom. Jazyku Python v súvislosti s možnosťou jeho použitia pri výučbe sa venovali aj niektoré príspevky na konferencii DIDINFO [2, 3]. V ostatných rokoch vznikli aj viaceré materiály, ktoré môžu poslúžiť pri úvodnom kurze programovania v jazyku Python na strednej škole [4, 5, 6, 7].

V našom príspevku sa nebudeme venovať priamo použitej metodike, ale pokúsime sa opísať skúsenosti s použitím rovnakého učebného materiálu pri vyučovaní troch rôznych skupín žiakov v 1. ročníku gymnázia. Všimneme si čo bolo v rámci troch pozorovaní spoločné a čo sa naopak vyskytlo iba ojedinele. Uvedomujeme si, že veľmi pravdepodobne budú naše pozorovania v súlade s pozorovaniami každého pozorného učiteľa informatiky. Napriek tomu sa však odvažujeme naše zistenia publikovať v nádeji, že priblížia situáciu v triede aj širšiemu publiku. Zaujala nás odlišná úroveň jednotlivých skupín a dokonca napriek malej početnosti a veľkosti skupín aj ich nehomogénnosť – v rámci skupín a aj v rámci každej skupiny.

2. OPIS PODMIENOK EXPERIMENTU

Experiment prebiehal počas celej prvej polovice školského roka na dvoch stredných školách – v troch triedach prvého ročníka štvorročného gymnázia. V predmete informatika bola každá trieda rozdelená na skupiny veľkosti desať až dvanásť žiakov. Každú skupinu učil iný učiteľ, pričom pri vyučovaní postupovali podľa rovnakého učebného materiálu [8]. Skupiny označíme A, B, C, aby sme sa v ďalšom texte mohli na ne odvolávať. Vyučovacia hodina mala 45 minút. Učitelia skupín A a C medzi sebou komunikovali počas trvania experimentu, avšak táto komunikácia neovplyvnila ich postup pri organizácii vyučovania.

Skupina A

Žiaci mali informatiku jednu vyučovaciu hodinu týždenne, prvú hodinu v úvode týždňa. V skupine bolo 11 žiakov z toho dvaja chlapci. Hodina prebiehala v počítačovej učebni, kde pri každom počítači sedel jeden žiak. Počítače boli umiestnené v dvoch radoch za sebou tak, že žiaci boli otočení tvárou k svojmu počítaču a zároveň aj k učiteľovi a plátnu s výstupom z jeho počítača. Učiteľ mal k dispozícii počítač s dataprojektorom, ale keď pri ňom pracoval a vysvetľoval učivo, nevidel na obrazovky žiakov. Žiaci však mohli zároveň sledovať výklad učiteľa a aj pracovať na svojom počítači. Prácu žiakov sledoval aj výskumný pracovník, ktorý občas odpovedal na ich otázky a po skončení hodiny diskutoval o jej priebehu s učiteľom. V úvode hodiny učiteľ vždy stručne zopakoval učivo z predchádzajúcej hodiny a nastolil nový problém, ktorý riešil spolu so žiakmi. Otázkami ich nabádal, aby diktovali riešenie úlohy, ktoré na učiteľskom počítači zapisoval učiteľ, alebo niektorý zo žiakov. Niektorí žiaci už počas výkladu učiteľa editovali na svojom počítači rovnaký

kód, ako videli na dataprojektore. Potom žiaci riešili ďalšie úlohy podobného typu samostatne, prípadne si vzájomne pomáhali. Učiteľ chodil medzi žiakmi, radil im, riešil prípadné problémy a kontroloval splnenie zadaných úloh. Na konci hodiny slovne zhodnotil prácu žiakov a najlepšie výkony odmenil známkou.

Skupina B

Žiaci mali informatiku dvakrát týždenne jednu vyučovaciu hodinu (v dva rôzne dni, raz piatu a raz šiestu hodinu). V skupine bolo 10 žiakov, z toho tri dievčatá. Žiaci v učebni sedeli každý za jedným počítačom. Keď žiaci pracovali pri počítači, boli otočení k vyučujúcemu chrbtom, čo umožňovalo učiteľovi ľahko kontrolovať ich prácu. Hodiny boli organizované nasledovne: na začiatku hodiny si všetci spoločne pripomenuli, čo robili na minulej hodine alebo veľmi stručne prebehol výklad novej látky. Vtedy sa žiaci otočili a sledovali učiteľa, výklad prebiehal pri tabuli. Zvyšok hodiny žiaci pracovali samostatne a podľa potreby privolávali učiteľa (obyčajne, keď nastal nejaký problém). V prípade, že si učiteľ všimol u viacerých žiakov nesprávne, prípadne nevhodné používanie nejakej programátorskej konštrukcie, prerušil prácu všetkých žiakov a upozornil ich, ako sa prípadnej chybe vyvarovať. Učiteľ v tejto skupine nepoužíval dataprojektor.

Skupina C

Informatiku mali žiaci dvakrát týždenne jednu vyučovaciu hodinu (v dva rôzne dni, štvrtú a piatu hodinu). V skupine bolo 10 žiakov, z toho štyri dievčatá. Učebňa bola rovnaká ako pri skupine B, žiaci boli otočení k vyučujúcemu a tabuli chrbtom. Žiaci sedeli samostatne pri počítači. Vyučujúci nepoužíval dataprojektor. Hodiny boli organizované podobne ako v skupine B: na začiatku si všetci spoločne zopakovali doteraz prebrané príkazy a konštrukcie. Následne učiteľ v rámci výkladu zadal úlohu, ktorú spoločne riešil so žiakmi. Počas riešenia úlohy prebiehala diskusia, pri ktorej sa žiaci zoznamovali s novým učivom. Žiaci mali k dispozícii pracovné listy. Počas zvyšku vyučovacej hodiny pracovali samostatne s prípadnou pomocou učiteľa.

3. POZOROVANIA

Priebeh vyučovania sme analyzovali na základe poznámok vyučujúcich, ktoré si zapísali v krátkom čase po skončení hodiny.

Použitý učebný materiál intenzívne využíval vývojové prostredie IDLE, ktoré je súčasťou balíka Python 3.5 a korytnačiu grafiku (modul turtle). Témy boli v tomto poradí: kreslenie jednoduchých obrázkov, premenná, for cyklus, podprogram bez parametra a s parametrom, zmena polohy korytnačky, náhodné čísla, podmienený príkaz. Vymenovaním tém chceme čitateľovi priblížiť celkový rámec experimentu. Ako sme už uviedli, samotný učebný materiál a jeho overovanie nie je predmetom tohto článku.

Cieľom použitej metodiky bolo zaviesť, prípadne upevniť základné programátorské koncepty a osvojiť si programátorské zručnosti – editovanie prehľadného kódu, simulovanie a ladenie programu, odstraňovanie chýb.

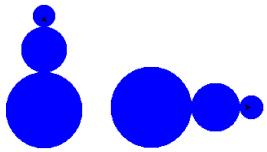
V samostatných častiach ďalej popíšeme detailnejší priebeh hodín a najčastejšie chyby, ktorých sa žiaci dopúšťali v rámci jednotlivých skupín A, B a C.

3.1 Skupina A

Z hľadiska programátorských zručností boli v tejto skupine žiaci, ktorí v minulosti programovali len minimálne, prípadne vôbec. Pokladali sme ich preto za programátorských začiatočníkov a podľa toho sme sa snažili prispôsobiť tempo výučby a spôsob vedenia hodín – učiteľ pomerne často vysvetľoval, čo je úlohou a pomocnými otázkami viedol žiakov k riešeniu. Často sme používali tabuľu, kde sme spoločne simulovali vykonávanie programu. S touto skupinou sme prebrali všetky vyššie uvedené témy, okrem zmeny polohy korytnačky a náhodných čísel, ktorým sa budeme venovať v blízkej budúcnosti.

Programovanie pohybu korytnačky bolo pre žiakov silnou motiváciou. Pri osvojovaní **základných vykresľovacích príkazov** mali žiaci problémy len so zadaním správneho parametra v príkaze otáčania. Pri kreslení rovnostranných trojuholníkov a pravidelných mnohouholníkov sme si museli zopakovať učivo o výpočte veľkosti vonkajšieho uhla.

Keď sa žiaci naučili meniť parametre pera (farbu, veľkosť), potešili sa, že dokážu vykresľovať komplikovanejšie obrázky. Dopúšťali sa však viacerých chýb. Zaujímavé chyby sa vyskytli napríklad pri kreslení snehuliaka (obrázok 1). Žiaci zabúdali pred vykreslením snehuliaka otočiť korytnačku, takže snehuliaka vykreslili na ležato.

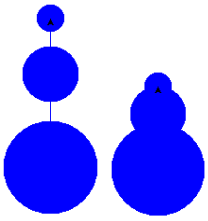


Obrázok 1: Požadovaný obrázok snehuliaka a snehuliak na ležato

Pri vykresľovaní kruhov (snehových gulí) bolo potrebné nastaviť hodnotu pre veľkosť pera rovnú priemeru kruhu a pri presúvaní korytnačky (pera) ju bolo potrebné zmeniť na hodnotu 1. Ak na to žiaci zabudli, snehuliak mal iný tvar (obrázok 2).

Obrázok 2: Snehuliak nakreslený s nesprávnou hodnotou pera pri presúvaní korytnačky (pera)

Ďalšou častou chybou bolo, že žiaci nesprávne zadali vzdialenosť, o ktorú sa má korytnačka posunúť medzi vykreslením jednotlivých kruhov (obrázok 3).

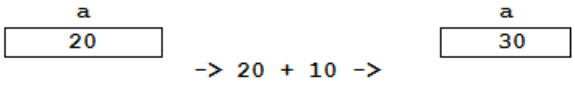


Obrázok 3: Nesprávna dĺžka presunu medzi vykresľovaním kruhov (príliš veľká vľavo, príliš malá vpravo)

Zaujímavé bolo, že rovnaké chyby sa vyskytli aj neskôr, keď sme obrázok snehuliaka kreslili pomocou procedúry s parametrom.

Pojem premenná – ako pomenované miesto v pamäti – bol pre žiakov nový. Najväčší problém mali so zmenou hodnoty premennej a konštrukciou typu *a=a+10*, ktorá sa im z matematického hľadiska zdala byť nezmyselná. Museli sme

zdôrazniť, že znamienko rovnosti je pre Python príkazom priradenia. Tiež sme kreslili na tabuľu, čo sa deje v pamäti počítača (obrázok 4).

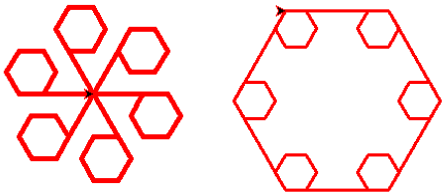


Obrázok 4: Zmena hodnoty premennej pri príkaze priradenia

Samotná **konštrukcia cyklu** s pevným počtom opakovaní (for cyklus) nerobila žiakom problémy a dokázali si ju pomerne rýchlo osvojiť. Pri riešení úloh zameraných na jej využitie žiakom nerobilo problém určiť a zadať správny počet opakovaní. Vyskytli sa však problémy iného typu. Žiaci si nedokázali uvedomiť, kde je pero nastavené na začiatku vykonávania programu a čo treba kresliť s využitím cyklu. Tiež si nedokázali predstaviť, kde je pero nastavené vždy na začiatku cyklu a ktorým smerom je otočené. Pozorovali sme aj, že často nastavovali farbu pera v tele cyklu, aj keď to stačilo urobiť pred vykonávaním cyklu. Pri riešení úloh, kde bolo potrebné využiť riadiacu premennú v tele cyklu, chceli žiaci spočiatku používať ďalšiu premennú (Obrázok 5).

Obrázok 5: Veľkosť bodiek sa postupne zväčšuje o 1 bod

Väčší priestor sme museli venovať úlohám zameraným na použitie **vnorených cyklov**. Slovne sme analyzovali, čo sa kreslí vo vonkajšom a čo vo vnútornom cykle, kde je pero nastavené a ako je otočené na začiatku a na konci každého z nich. Pri vykresľovaní obrázkov, v ktorých bolo vo vnútornom cykle potrebné použiť príkaz na spätný posun pera do stredu obrázka, žiaci na tento úkon často zabúdali (obrázok 6).



a) b)

Obrázok 6. Ak žiaci zabudli vo vnútornom cykle presunúť pero do stredu, namiesto útvaru a) sa nakreslil útvar b)

Viaceré úlohy zamerané na použitie vnorených cyklov by sa pravdepodobne dali riešiť jednoduchšie, ak by žiaci vedeli použiť podprogram. V budúcnosti preto plánujeme prehodnotiť poradie tém.

V čase dokončovania tohto článku sme začali preberať tému podprogramy. Zatiaľ žiaci riešili iba jednoduché úlohy, takže vážnejšie problémy sme nepozorovali.

Všimli sme si, že viacerí žiaci sa vzdávali, keď videli komplikovanejší obrázok, ktorý hneď nevedeli nakresliť. Keď sme si však rozobrali postup, pracovali s nadšením.

3.2 Skupina B

Na začiatku školského roka sme žiakom položili v dotazníku niekoľko otázok. Na otázku, či sa učili na ZŠ programovať, odpovedali piati záporne. Zo zvyšných žiakov dvaja uviedli, že sa učili programovať v Delphi a Imagine, ďalší žiaci uviedli len jednu z nasledujúcich možností: „Imagine“, „korytnačka a Scratch“, „áno učili, ale nepamätám si čo“. Pri analýze odpovedí na otázku, čo si žiaci predstavujú pod pojmom informatika, bolo zrejmé, že informatiku chápu len ako školský predmet, nie ako inžiniersku, či vedeckú disciplínu. Bolo to očividné z ich odpovedí: „hranie sa na PC“, „voľná hodina“, „tvorba prezentácií“ alebo „neviem“.

Pre úplnosť spomenieme, že pred začiatkom práce podľa učebného materiálu [8], sme so žiakmi približne mesiac programovali robota Karla. Tu sa žiaci stretli s cyklom s pevným počtom opakovaní aj s cyklom s podmienkou a podmieneným príkazom, ale nestretli sa s premennými.

Učiteľ žiakom zriadil webovú stránku so zadaniami úloh. Úlohy boli v podstate z uvedeného učebného materiálu. V prípade, že sa vyskytla nová funkcia alebo učiteľ chcel žiakov upozorniť na nejakú konštrukciu, do dokumentu s úlohami vložil tieto informácie vizuálne oddelené od zadaní úloh. Nádejal sa, že ak si žiaci pozrú úlohy mimo hodiny, budú tam mať aj doplňujúce poznámky. Žiaľ, čítanie textu (so zadaniami úloh) s porozumením nebolo pre všetkých žiakov samozrejmosťou. Ale na druhej strane, dostupnosť zadaní úloh umožnila každému žiakovi postupovať svojím tempom. Opakovane sa potvrdilo, že reakcia žiakov na úlohy zadané písomne, bola lepšia než slovné zadanie úlohy pri tabuli.

V učebnom materiály, podľa ktorého sme postupovali, bola jedna z prvých tém premenná. Žiakom sa jednoduché úlohy darilo riešiť pomerne dobre. Bolo vidieť, že niektorí sa už s premennými v minulosti stretli. Naproti tomu, použitie premennej niektorým žiakom robilo problémy aj tretí mesiac po tom, ako sa s ňou prvý raz stretli. Nevedeli premennú použiť na uloženie hodnoty, ktorú budú neskôr viackrát potrebovať. Zdá sa, že koncept premennej netreba podceniť, akokoľvek jednoducho vyzerá na prvý pohľad. V skutočnosti jeho zvládnutie vyžaduje relatívne dlhý čas. Pojem premennej v informatike a v matematike, kde sa väčšinou spája s neznámou v rovnici, žiaci nemajú takmer vôbec prepojený.

Ďalšou témou bol cyklus s pevným počtom opakovaní. Napriek tomu, že jednotlivé úlohy viac-menej úspešne postupne vyriešili všetci žiaci, čas od času sa vyskytla na hodine situácia, keď sa zdalo, akoby ani po ôsmych vyučovacích hodinách práce s cyklami nebolo niekomu z nich jasné, v akom poradí sa vykonávajú príkazy v cykle.

Samostatným problémom, vyplývajúcim s používaným jazykom Python, je spôsob zápisu blokov formou odsadzovania. Nesprávne odsadenie príkazov zvyčajne spôsobí chybu. V lepšom prípade ide o syntaktickú, na ktorú upozorní Python ešte pred spustením programu, alebo v horšom prípade program vykoná niečo iné ako očakáva žiak.

V riešeniach rôznych žiakov sa aj napriek tomu, že na hodine nesedeli vedľa seba, vyskytlo niekoľko, v rámci jazyka Python správnych, ale neočakávaných konštrukcií. Napríklad cyklus, ktorého telo sa vykoná iba raz:

```
for i in range(1)
```

alebo definovanie funkcie v cykle

```
for f in range(100):
    def hviezda(d):
        a=random.randrange(30,45)
        for y in range(d):
            fd(a)
            fd(-a)
            rt(360/d)
```

Napriek tomu, že sme žiakom vysvetlili, prečo je takéto použitie nevhodné a zopakovali sme si správne použitie, prekvapilo nás, že sa vyskytovalo aj naďalej.

Napriek našej neustálej podpore akejkoľvek programátorskej aktivity, iba traja študenti prišli s nejakým vlastným nápadom, čo by chceli naprogramovať. Jedna dvojica svoj nápad s našou pomocou aj zrealizovala. Žiakov zaujala animovaná ikona, ktorá sa používa počas čakania na ukončenie nejakej dlhšie trvajúcej akcie (obrázok 7). Aktivita sa dala zrealizovať, kruhy zadanej farby sme vykresľovali, takže žiaci museli len napísať správny cyklus. Postačil nám cyklus s pevným (podľa potreby dostatočne veľkým) počtom opakovaní. Druhý nápad zrealizoval žiak so svojím bratom počas vianočných prázdnin – vytvorenie slepej mapy na geografiu. Tento projekt bol ale nad momentálne sily žiaka.

Obrázok 7 Ikona ilustrujúca čakanie na dokončenie procesu

Do Vianoc sme nezadávali domáce úlohy. Sledovali sme žiakov na hodine a na základe toho, ako pracovali, sme mali pocit, že preberanej látke rozumejú. Po vianočných prázdninách nastal zlom a žiakov sa nám nepodarilo ani počas niekoľkých vyučovacích hodín „naštartovať“. Zaviedli sme bodovanie riešení úloh. Čo nevyriešili na hodine, mohli riešiť doma do daného termínu s tým, že získajú rovnaký počet bodov ako na hodine. Po tomto termíne mohli žiaci riešenia odovzdávať, ale za menší počet bodov. Tento prístup pomohol žiakov aktivizovať. Pri vyváženom počte úloh a bodov, ktoré žiakom zaručia určitú známku, by mohol predstavovať istú formu vonkajšej motivácie.

Úlohy, ktoré žiaci riešili, mali všetky grafický výstup, čo im umožňovalo okamžite po spustení programu vidieť, či pracuje správne. Keď sa vyskytla v ich programe chyba, iba ojedinele vedeli zistiť, čo ju spôsobilo a následne ju nájsť a odstrániť. Najhoršie bolo, keď program nehlásil chybu, ale jeho výsledkom nebol želaný obrázok. V takom prípade pomohla až intervencia učiteľa. Našim úsilím ale nebolo nájsť žiakom chybu. Našim cieľom bolo ukázať im techniky a postupy, ako chybu nájsť: testovanie programu, či ešte lepšie systematická tvorba a postupné testovanie vznikajúceho programu. Toto by malo byť neoddeliteľnou súčasťou každého kurzu programovania.

3.3 Skupina C

Žiaci z tejto skupiny s nadšením uvítali, že sa budú učiť programovať. Aj im sme položili otázku, či sa už stretli s programovaním. Zistili sme, že máme v skupine žiakov s rôznymi skúsenosťami: osem žiakov už programovalo v Imagine Logo, traja žiaci sa už stretli s jazykom Python, ale bližšie si nevedeli spomenúť, čo prebrali. Traja žiaci spomenuli, že sa doma sami venovali programovaniu – buď v jazyku Scratch, Python alebo C#. Vyučovanie prebiehalo podľa uvedeného učebného materiálu [8]. Žiaci nemali k dispozícii dokument s už prebranými príkazmi, ale v priebehu prvých hodín, počas opakovania preberaného učiva, učiteľ napísal príkazy na tabuľu. Približne po troch hodinách si ich už žiaci zapamätali a nebolo potrebné ich opakovať. Pracovné listy sme žiakom poskytli postupne na našej webstránke. Žiaci väčšinou pracovali samostatne, prípadne riešenie prediskutovali so susedmi, navzájom si pomáhali nájsť chyby v riešení, alebo privolali učiteľa na pomoc. Približne do polovice kurzu sme po každej druhej vyučovacej hodine zadávali krátku domácu úlohu. Neskôr sme od domácich úloh upustili, ale niektorí žiaci prichádzali na hodiny s tým, že dokončili doma úlohy, ktoré nestihli dokončiť cez hodinu.

V tejto skupine sme žiakov nemuseli motivovať ani známkami, ani písomkou. Žiaci boli na hodinách aktívni, spolupracovali a v niektorých prípadoch si vyžiadali aj náročnejšie úlohy, alebo si vymysleli vlastné vylepšenia úloh.

Medzi časté chyby žiakov patrilo, že aj pri náročnejších úlohách najprv napísali celé riešenie a až potom spustili program. Aj preto sa stávalo, že program nekreslil požadovaný obrázok a z naprogramovaného obrázka sa nedalo priamo zistiť, v ktorom kroku spravili chybu. Pri takýchto chybách už požadovali pomoc učiteľa a väčšinou stačilo opraviť len hodnotu parametra v niektorom príkaze. Snažili sme sa ich upozorniť na to, aby riešenie náročnejších úloh rozdelili na časti a postupne overovali svoje riešenie.

Všimli sme si, že niektorí žiaci mali problémy s premennými a parametrami, keď chceli parameter použiť aj mimo definície funkcie ako premennú.

Pri niektorých žiakoch sme si všimli negatívnu reakciu na chybové hlásenia a neochotu opravovať chyby. V takýchto prípadoch sme pomohli pri opravení programu a slovne podporili ďalšiu prácu žiaka. Po vysvetlení príčiny chybového hlásenia, sme sa čoraz menej stretávali s takými syntaktickými chybami žiakov, ktoré by si sami nedokázali opraviť.

4. ZÁVERY Z POZOROVANÍ

Na základe pozorovaní, vykonaných vo všetkých skupinách, sme dospeli k nasledujúcim záverom.

- Je potrebné zadávať domáce úlohy, aby si žiaci prebranú látku precvičovali aj doma a nemali problém s vybavovaním príkazov v prípade absencie alebo ak nejaká hodina odpadne. Vhodné je, aby všetky domáce úlohy a prípadne aj iné materiály z vyučovania, mali žiaci k dispozícii na internete. V skupine A napriek tomu, že mali úlohy dostupné, nikto zo žiakov doma nič neriešil. V skupine B žiaci začali úlohy doma riešiť až po zavedení bodovania riešení. V skupine C žiaci riešili doma úlohy aj bez bodovania.
- Žiaci si na informatike nerobia písomné poznámky. Ako by mali pocit, že všetko, čo budú potrebovať, si zapamätajú. Samozrejme to nie je pravda. Neustále sme žiakom pripomínali, že danú úlohu je lepšie si najprv vyriešiť (niekedy stačí len čiastočne) na papieri, alebo načrtnúť v programe Skicár, pretože napríklad vyžaduje nejaké elementárne počítanie, ktoré z hlavy bez dostatočných skúseností nespravia (veľkosti uhlov, resp. vzdialenosti, o ktoré sa má korytnačka posunúť).
- Je potrebné nabádať žiakov, aby sa nebáli skúšať. Napríklad, čo je výsledok danej funkcie, špeciálne keď Python poskytuje interaktívne prostredie, ktoré to umožňuje.
- Nesmieme zabúdať dávať žiakom aj úlohy typu *„Máme takýto program, čo nakreslí korytnačka?“* Koniec koncov, musia čítať aj vlastné programy, keď si hľadajú chybu, alebo keď chcú pokračovať po nejakom čase v programe. Skúšali sme takýto typ úloh vo všetkých skupinách. V skupine B a C sme zadali rovnakú úlohu. V skupine C ju vyriešili všetci žiaci za 15 minút. V skupine B ju celú vyriešili za 20 minút traja žiaci, a šiesti vyriešili tretinu, jeden úlohu nevyriešil. V

skupine A sme zadali ľahšiu verziu úloh a až na jednu žiačku, ktorá si zamieňala príkazy vľavo a vpravo, nemal s ich riešením nikto väčšie problémy.

- K motivovaniu žiakov prispieva (napríklad bodové) ohodnotenie ich práce.
- Žiaci sa väčšinou zľaknú, keď nerozumejú chybovej správe – je dôležitá podpora zo strany učiteľa, aby nestratili motiváciu pri riešení zadanej úlohy ani po prvotnom *„neúspechu“* pri spustení programu.
- Dôležité je viesť žiakov k postupnému riešeniu úlohy – zložitý problém rozdeliť na menšie časti, a tie riešiť definovaním vlastných príkazov (podprogramov). Vznikajúci program je potrebné postupne spustiť, aby žiaci zistili, či je ich riešenie správne (testovanie).
- Špecificky v jazyku Python musíme dbať na dôkladné vysvetlenie odsadenia bloku príkazov – napríklad, ktoré príkazy sa vykonajú v rámci cyklu a ktoré mimo neho.

Zaujímavé sa ukázalo aj porovnanie úrovne jednotlivých skupín. Skupiny A a B si boli na začiatku veľmi podobné. Významným rozdielom medzi nimi bola iba časová dotácia predmetu informatika, vďaka ktorej skupina B postupovala rýchlejšie. Myslíme si, že rýchlejší postup nebol spôsobený len dvojnásobnou hodinovou dotáciou, ale aj tým, že medzi jednotlivými hodinami boli menšie prestávky. Čas potrebný na opakovanie bol kratší, a teda ho ostalo viac na riešenie nových úloh, či preberanú látku. Skupinu C tvorili žiaci, ktorí boli vybraní na základe prísnejších prijímacích kritérií (úspešnosť asi 1:10). Skupina mala úplne inú úroveň. Žiaci boli naučení pracovať aj samostatne, čo umožňovalo preberať látku ešte rýchlejšie a riešiť ťažšie úlohy.

BIBLIOGRAFICKÉ ODKAZY

[1] Štátny vzdelávací program. Informatika. Príloha ISCED 3A [online]. [cit. 2017-01-03]. Dostupné na internete: http://www.statpedu.sk/sites/default/files/dokumenty/statny-vzdelavaci-program/informatika_isced3a.pdf

[2] MÉSZÁROSOVÁ, E., TOMCSÁNYIOVÁ, M.: Vyučovanie základov programovania v jazyku Python na niektorých SŠ a VŠ, *DidInfo 2015,* Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, 2015, s.108-111.

[3] HORVÁTHOVÁ, D., KABÁČOVÁ, N.: Multimédia v prostredí jazyka Python, *DidInfo 2016,* Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, 2016, s. 22-26.

[4] KUČERA, P.: Programujeme v Pythone, učebnica informatiky pre stredné školy, 2016, ISBN 978-80-972320 (pdf)

[5] BLAHO, A.: Programovanie v Pythone časť" 1, 2016 https://zona.fmph.uniba.sk/fileadmin/fmfi/sluzby/elektronicke_studijne_materialy/Python1.pdf

[6] BELAN, A.: Python, učebný text pre septimu osemročného gymnázia, 2013, ŠPMNDaG http://www.smnd.sk/anino/moje/Python.pdf

[7] HORÁČKOVÁ, H.: Výuka programování v jazyce Python pro střední školy, diplomová práce, Masarykova Univerzita v Brně, 2014, http://is.muni.cz/th/39773/fi_m/dp.pdf

[8] MÉSZÁROSOVÁ, E. *Vývoj vzdelávacieho obsahu pre vyučovanie programovania v jazyku Python*. Rigorózna práca. Bratislava: Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, 2017.

Programovanie interaktívnych úloh súťaže iBobor pomocou knižnice sprites.js

Using the library sprites.js for programming interactive tasks in iBobor contest

Monika Tomcsányiová
Katedra základov a vyučovania informatiky, FMFI UK v Bratislave
Mlynská dolina
842 48 Bratislava
Slovensko
tomcsanyiova@fmph.uniba.sk

Peter Tomcsányi
Katedra základov a vyučovania informatiky, FMFI UK v Bratislave
Mlynská dolina
842 48 Bratislava
Slovensko
tomcsanyi@fmph.uniba.sk

Michal Winczer
Katedra základov a vyučovania informatiky, FMFI UK v Bratislave
Mlynská dolina
842 48 Bratislava
Slovensko
winczer@fmph.uniba.sk

ABSTRACT

In this paper, we describe programming of interactive tasks in the iBobor contest with help of created sprites.js library. In the past interactive tasks used Flash, which is declining slowly a therefor we decided use Javascript in connecting with HTML 5.

Keywords

JavaScript, iBobor contest, interactive tasks.

ABSTRAKT

V tomto článku opíšeme programovanie interaktívnych úloh súťaže iBobor pomocou knižnice sprites.js. V minulosti sa v úlohách, ktoré mali byť interaktívne používala technológia Flash, ktorá postupne prestáva byť vo webových prehliadačoch podporovaná, preto sme sa ju rozhodli nahradiť použitím JavaScriptu a HTML 5.

Kľúčové slová

JavaScript, iBobor, interaktívne úlohy

1. ÚVOD

Tento školský rok sa uskutočnil už 10. ročník súťaže Informatický bobor, pozri [1]. Každoročne sa do nej zapája pomerne veľké množstvo žiakov. V relatívnom počte vzhľadom k počtu obyvateľov si Slovensko udržiavalo niekoľko rokov prvenstvo, tento rok sa súťaže zúčastnilo 1,16% obyvateľov Slovenska, väčší podiel dosiahlo len Slovinsko (1,41%).

Úlohy súťaže sú pripravované v rámci medzinárodnej spolupráce odborníkov v oblasti informatiky z celého sveta. Väčšina úloh je pripravených tak, aby na ne žiak odpovedal výberom jednej zo štyroch možností, ale niektoré úlohy sú navrhnuté tak, aby pri jej riešení mohol žiak interagovať s počítačom, napr. aby mohol myšou presúvať obrázky. Takéto úlohy nazývame interaktívne a každý rok je takýto typ úloh v každej súťažnej kategórii.

2. INTERAKTÍVNE ÚLOHY

Tento rok sme sa rozhodli pre podstatnú zmenu v technickej realizácii interaktívnych úloh súťaže, viac o interaktívnych úlohách pozri [2]. Namiesto doteraz používanej technológie Flash Player sme sa rozhodli pre interaktívne úlohy použiť JavaScript v spolupráci s HTML 5. Táto inovácia sa nám po technickej stránke podarila a umožnila nám zbaviť sa závislosti od, dnes už zastarávajúcej, technológie Flash. Vďaka tomu bolo tento rok možné súťažiť aj na tabletoch, prípadne smartfónoch.

Okrem zmien v súťažnom systéme bolo potrebné navrhnúť, pripraviť a naprogramovať JavaScript–ovú knižnicu, pomocou ktorej by dokázali autori interaktívnych úloh súťaže pomerne rýchlo naprogramovať súťažné úlohy.

2.1 Návrh a vývoj knižnice sprites.js

Pri návrhu knižnice bolo dôležité, aby bola na jednej strane dostatočne komplexná, ale na druhej strane aby bola zároveň aj otvorená a umožnila programátorovi interaktívnych úloh doprogramovať ďalšie vlastnosti jeho programu.

Najdôležitejšou vlastnosťou knižnice, ktorú sme potrebovali implementovať bolo, aby dokázala spolupracovať s kresliacou grafickou plochou, ktorú ponúkajú súčasné webové prehliadače. Týmto sme chceli dosiahnuť to, aby túto technickú stránku realizácie interaktívnej úlohy nepotreboval zvládať programátor súťažných úloh. Ten by sa potom mohol plne sústrediť na programovanie samotnej úlohy, pričom udalosti klikania, ťahania a pustenia objektu, väčšinou obrázku, by dokázala za neho spracovať samotná knižnica.

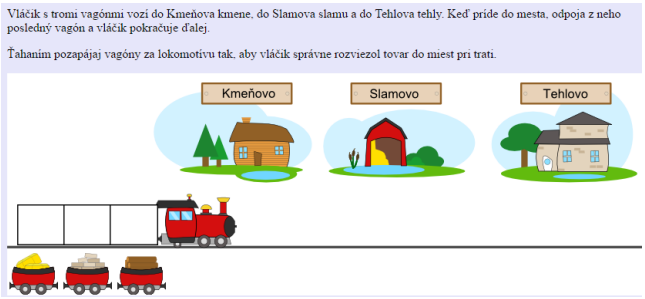
Vývoj knižnice `sprites.js` [2] prebiehal v niekoľkých fázach. Skúsený programátor najprv navrhol knižnicu na základnú prácu s grafikou v prostredí JavaScript v spolupráci s HTML5, ktorá by rovnako dobre fungovala vo webovom prehliadači mobilného zariadenia ako na desktopovej platforme. Prvý prototyp knižnice obsahoval vytváranie objektov a základné spracovanie udalostí ťahania a umiestňovanie objektov do konkrétnych oblastí. Ďalšiu prácu s objektmi, napr. obrázky objektov, vykresľovanie oblastí, kreslenie do pozadia prostredia, si mohol tvorca aplikácie doprogramovať priamo v prostredí JavaScript, mimo knižnice. Knižnicu sme začali používať a pomocou nej sme naprogramovali niekoľko úloh súťaže. Pri testovaní sa ukázalo, že programovanie takýchto aktivít pomocou súčasnej funkčnosti knižnice si vyžaduje pomerne pokročilé programátorské skúsenosti, ktoré priamo nesúvisia s naprogramovaním súťažnej úlohy. Preto bolo potrebné knižnicu upraviť a pokročilejšie programátorské techniky presunúť do jej vnútra a poskytnúť ich v tvare funkcií a metód objektov typu `Sprite`. Výslednú knižnicu sme použili na vývoj všetkých interaktívnych úloh súťaže.

3. TYPY INTERAKTÍVNYCH ÚLOH

Počas predchádzajúcich ročníkov súťaže sa ukázalo, že interaktívne úlohy súťaže môžeme rozdeliť do niekoľkých typov, pozri [3].

Aby sme knižnicu **`sprites.js`** dokázali pri vývoji interaktívnych úloh efektívne používať, je potrebné ukázať, že pomocou nej dokážeme implementovať všetky typy úloh v súťati a ujasniť si, ako a aké udalosti objektov typu **`Sprite`** pritom využijeme.

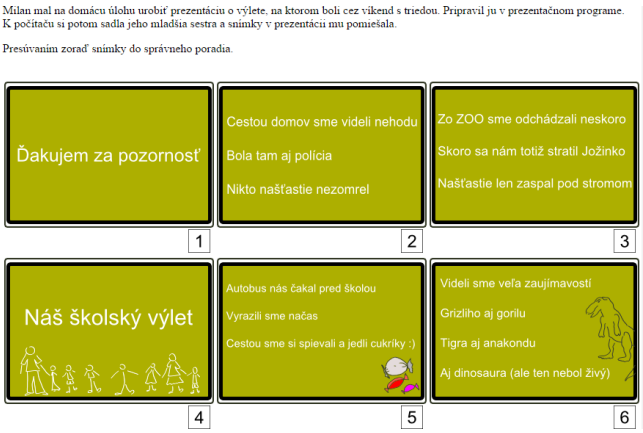
- **Ťahanie** a umiestňovanie objektov do cieľových pozícií, ktoré sú väčšinou zobrazené ako prázdne rámiky, pozri Obrázok 1.



Obrázok 1. Úloha Vláčik, 2016/17, Bobrík

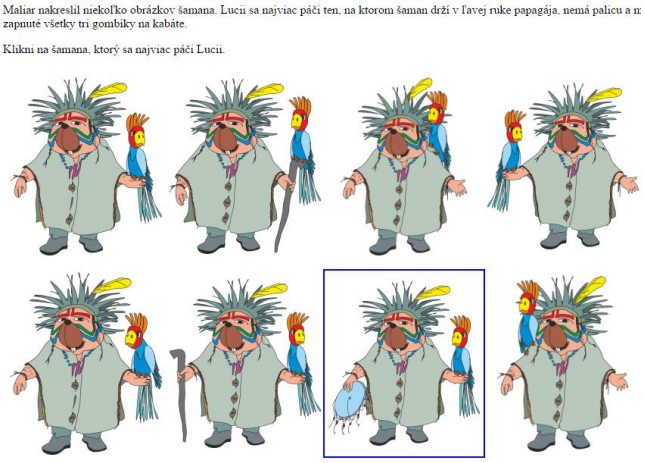
Pre tento typ aktivít je v knižnici pripravená metóda objektu **`sprite.placeAt(ciel)`**. Pri realizácii aktivity si v premennej, pamätáme, ktoré políčka určujú ciele aktivity a pomocou metódy **`sprite.findOverlapped(ciele)`** z knižnice **`sprite.js`** zistíme, nad ktorým cieľom bol objekt pustený. Základné správanie objektu, ktorý je pustený nad niektorým cieľovým políčkom je také, že ak sa na tomto cieľovom políčku nachádza iný objekt, tento objekt sa premiestni na svoju domácu pozíciu a ťahaný objekt sa umiestni na cieľové políčko, nad ktorým bol pustený. V prípade, ak chceme, aby sa na jednu pozíciu mohlo umiestniť viacero objektov, využijeme pre tieto objekty umiestňovanie pomocou metódy **`sprite.placeInto(ciel)`**, pričom objekt sa umiestni do oblasti, v ktorej je cieľové políčko.

- **Vymieňanie** objektov navzájom, t. j. objekt, ktorý ťaháme si vymení miesto s objektom, nad ktorým ho pustíme a teda na ploche sa nenachádzajú prázdne orámikované miesta pre objekty, pozri Obrázok 2.



Obrázok 2. Úloha Prezentácia, 2016/17, Benjamín

- **Kliknutie** na objekt a jeho označenie farebným rámikom. Tento typ aktivít môžeme rozdeliť ešte na dva typy – pri riešení môže byť označený iba jeden alebo viacero objektov, pozri Obrázok 3.



Obrázok 3. Úloha Šaman, 2016/17, Benjamín

V knižnici **`sprites.js`** je implementovaná udalosť **`onClick`**, ktorá zabezpečuje reagovanie objektu na kliknutie. V nej autor úlohy typicky zmení farbu obdĺžnika nakresleného okolo objektu zmenou hodnoty objektovej premennej `highlight`. Programátor tak napr. môže zrušiť označenie všetkých objektov a následne označiť objekt (časť kódu pre osem obrázkov šamanová):

```
for (var i = 0; i < 8; i++) {
    ciele[i].highlight = null;
}
sprite.highlight = "#0000FF";
```

V predchádzajúcej časti sme opísali typy interaktívnych úloh a ich súčasnú implementáciu pomocou knižnice **`sprites.js`**. Tým sme zároveň vyšpecifikovali aj všeobecné funkcie, ktoré autor knižnice implementoval.

V tomto ročníku súťaže sme využitím knižnice naprogramovali všetkých 19 interaktívnych úloh. Knižnicu pritom použil programátor súťažných úloh, ktorý spolupracoval s autorom knižnice pri jej vývoji a dolaďovaní funkcií pre objekty **`Sprite`**.

Knižnicu sme pre vývoj ďalších troch interaktívnych úloh dali spolu so zadaním a šablónou jedného typu úloh k dispozícii aj ďalšiemu programátorovi. V časti 4 sú popísané jeho skúsenosti a práca s knižnicou.

4. ŠABLÓNA PRE ŤAHACIE AKTIVITY

Programátor dostal okrem knižnice **`sprites.js`** k dispozícii aj šablónu vo forme html súboru, ktorá obsahovala príklad použitia objektov **`Sprite`** z knižnice, pozri Tabuľku 1. Programátor pred tým neprogramoval v JavaScripte, ale má dlhoročné a pomerne bohaté skúsenosti s čítaním cudzích programov v rôznych jazykoch. Šablóna, pozri Tabuľku 1, obsahovala riešenie úlohy na Obrázku 1. Táto úloha bola typu Ťahanie, pozri časť 3. Typy interaktívnych úloh.

V šablóne môžeme okrem základného obalu html dokumentu rozoznať niekoľko častí JavaScript-u. Funkciu **`onDragDrop`**, ktorá je súčasťou knižnice **`sprites.js`** a ktorá určuje správanie objektu **`Sprite`** pri jeho pustení myšou, t. j. objekt sme ťahali

a práve sme ho pustili na nejakej pozícii. V našom prípade je potrebné, aby sa objekt umiestnil na cieľovú pozíciu v prípade ak bol nad niektorou z nich pustený.

Tabuľka 1. Šablóna pre interaktívnu úlohu

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <script src="../sprites.js"></script>
  <script>
    var ciele = [];
    onDragDrop = function(sprite) {
      var ciel = sprite.findOverlapped(ciele);
      sprite.placeAt(ciel);
    }
    var nazov = "2016-AU-07_";
    var n = ["slama", "tehly", "kmene"];
    for (var i = 0; i < 3; i++){
      (new Sprite(nazov + n[i] + ".svg",
       35 + i * 70, 260, dragSprite)).idxy = n[i];
      ciele[i] = new Sprite(nazov + "prazdna.svg",
                            45 + i * 60, 197);
    };
    (new Sprite(nazov + "pozadie.svg",410,120));
  </script>
</head>
<body style="margin: 0; padding: 0; background-
color:#E6E6FA">
  <canvas id="canvas" style="display: block;
width: 843px; height: 288px"></canvas>
</body>
</html>
```

Premenná **ciele** obsahuje objekty typu **Sprite**, ktoré určujú cieľové pozície kam sa majú umiestniť objekty. Druhá časť šablóny obsahuje vytváranie obrázkov a cieľových pozícií. Obe polia obsahujú objekty typu **Sprite**, ale s rôznymi vlastnosťami. Objekty sa vytvárajú pomocou konštrukcie **new Sprite**. Prvý argument je súbor s obrázkom, pričom obrázky musia byť v tom istom adresári ako html súbor, aby nebolo potrebné uvádzať celú cestu. Druhý a tretí argument je x-ová a y-ová súradnica pozície, kam sa obrázok pri inicializácii umiestni. Posledný, nepovinný, argument určuje, či sa objekt bude dať ťahať, pričom cieľové políčka nemajú tento argument, a teda sa s nimi nebude dať hýbať. Objekt typu **Sprite**, ktorý predstavuje ťahateľný objekt má nastavený ešte atribút **idxy**. Ten slúži na to, aby sme pri kontrole riešenia vedeli zistiť, či sú objekty na svojom mieste. V tejto úlohe sme použili na tento účel meno príslušného obrázka. Programátor si však môže vymyslieť vlastný názov atribútu a samozrejme si doň uložiť hodnotu, aká sa mu bude hodiť pri kontrole riešenia. Keďže je v tejto rovnaký počet ťahateľných objektov a ich cieľových pozícií, môžeme ich vytvárať v tom istom cykle. Môžu sa však vyskytnúť úlohy, v ktorých potrebujeme, aby počet objektov, ktoré sa dajú ťahať, bol väčší než počet cieľových pozícií. Vtedy ich zrejme budeme vytvárať v dvoch rôznych cykloch. Ak potrebujeme vytvoriť viacero rovnakých objektov na sebe, vytvorené objekty jednoducho umiestnime na rovnaké pozície, pozri Obrázok 4, v ktorom je na sebe niekoľko obrázkov každej šípky, aby žiaci pri riešení úlohy mohli každú z nich použiť niekoľkokrát. Pri ťahaní sa vždy odoberie vrchný objekt a ostatné objekty ostávajú na svojom mieste. V prípade odobratia objektu z niektorej cieľovej pozície sa objekt znovu umiestni na svoje pôvodné miesto k ostatným rovnakým objektom.

Funkcia **getResult()** (pozri Tabuľku 2.) slúži na komunikáciu so súťažným systémom, ktorý požaduje od programu aj vyhodnotenie správnosti.

Tabuľka 2. Funkcia getResult()

```
vysl = ["mene", "slama", "tehly"];

function getResult() {
  var ok = true;
  for (var i = 0; i < ciele.length; i++) {
    if (ciele[i].item != null)
      ok = false
  }
  if (ok)  //ani sa toho nechytil(a)
    return "x";
  else {
    ok = true;
    for (var i = 0; i < ciele.length; i++) {
      if (ciele[i].item == null)
        ok = false; // ak je niektora cielova
                    // pozicia prazdna
      else
        if (ciele[i].item.idxy != vysl[i])
          ok = false; // ak niečo nie je na
                      // svojom mieste
    }
    if (ok)
      return "b";
    else
      return "d";
  }
}
```

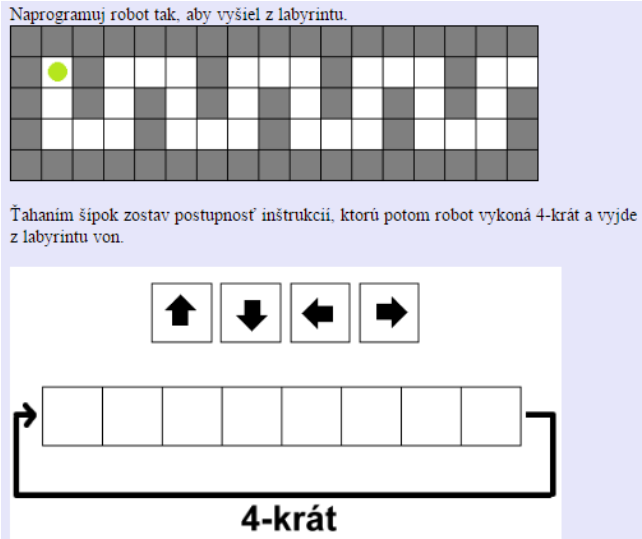
Funkcia vráti **"x"**, ak používateľ zanechal úlohu v jej počiatočnom stave (vtedy predpokladáme, že na úlohu nechce odpovedať). Funkcia vráti **"d"**, ak výsledok nie je správny (neúplný) a výsledok **"b"** znamená, že riešenie je správne. V tomto príklade sa správnosť riešenia zistila jednoduchým porovnaním, či sú objekty na cieľových pozíciách v správnom poradí. Tento prístup sa dá použiť vždy vtedy, keď má úloha jediné riešenie. V prípade viacerých riešení sa programátor musí rozhodnúť, ako bude kontrolovať správnosť riešenia.

4.1 Skúsenosti s programovaním

Pri programovaní interaktívnych úloh nenastal prakticky žiaden problém, čo sa týka použitia knižnice **sprites.js**. Asi najväčšie úsilie vyžadovalo hľadanie a odstraňovanie chýb v programe, a tak nevyhnutnosť použiť ladiace prostriedky, ktoré sú v súčasnosti súčasťou prehliadačov. Jediný vážnejší problém spôsobovala skutočnosť, že nie všetky prehliadače umožňujú vývoj JavaScript-ových programov na lokálnom počítači. To je zapríčinené rôznou bezpečnostnou politikou, ktorú aplikujú jednotlivé prehliadače a ktorá sa časom mení s novými prichádzajúcimi verziami prehliadačov. Špeciálne Chrome v posledných mesiacoch v jednotlivých verziách obmedzil lokálne čítanie súborov, takže pri ladení programu treba využiť lokálny webserver.

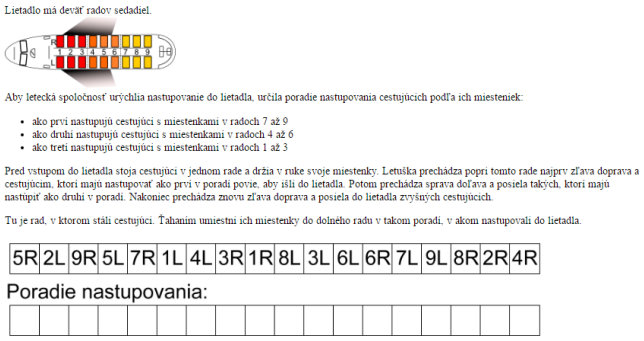
Šablónu pre interaktívne úlohy sme využili pri príprave troch úloh – Let, pozri Obrázok 5, a Robot, pozri Obrázok 4, pričom úloha Robot bola tak podobná pre kategórie Bobrík a Benjamín, že bol program pre obe kategórie takmer identický a líšil sa len počtom vygenerovaných kartičiek so šípkami, počtom voľných pozícií pre riešenie úlohy a samozrejme kontrolou správneho riešenia úlohy.

Postup, ktorý sme pri programovaní úloh využili bol s využitím šablóny pomerne rutinný. V každej z úloh bolo potrebné uvedomiť si, zmeny pri generovaní kartičiek, samozrejme zmeny obrázkov a zmeny cieľových pozícií, na ktorých sú umiestnené prázdne kartičky.



Obrázok 4. 32 kartičiek so šípkami (osem rovnakého druhu) v úlohe Robot. 2016/17, Bobrík

Časovo najnáročnejšou činnosťou vytvárania interaktívnych úloh bola príprava obrázkov. Ako jednu z úloh, ktorú sme programovali bola úloha Let, pozri Obrázok 5, v ktorej bolo potrebné nakresliť osemnásť kartičiek s číslom a písmenom a tiež prázdnu kartičku, ktorá určovala cieľové pozície.



Obrázok 5. Osemnásť kartičiek s číslom a písmenom v úlohe Let. 2016/17, Senior

Pri príprave týchto obrázkov išlo v podstate o rutinnú činnosť, ktorá by sa z veľkej časti dala nahradiť vhodných počítačovým programov, pozri časť 5.2. Generovanie kartičiek s textom.

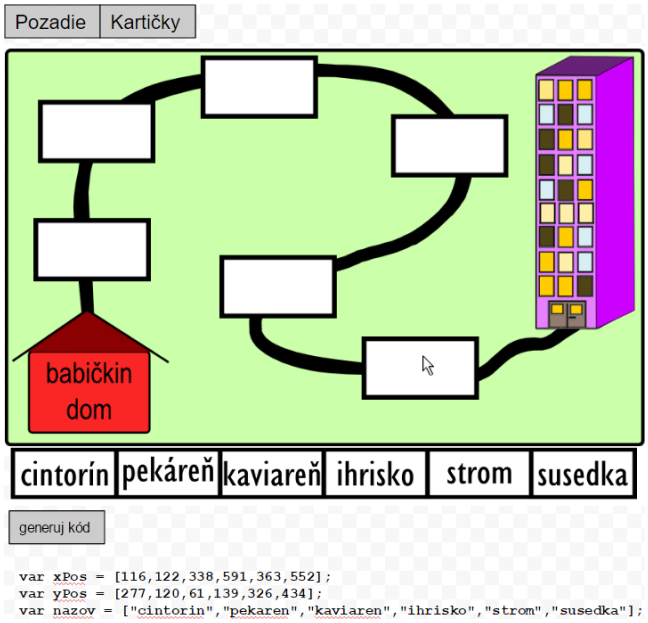
5. PODPORNÉ PROGRAMY PRE VÝVOJ INTERAKTÍVNYCH ÚLOH

Počas tohtoročného vývoja interaktívnych úloh okrem vytvorenia šablón pre rôzne typy interaktívnych úloh sme zistili aj to, ktoré činnosti sú potrebné pri ich programovaní a vyšpecifikovali sme dve také, ktoré by sa dali pomerne jednoducho zautomatizovať.

Na základe tohto sme sa rozhodli navrhnúť špecifikáciu dvoch pomocných programov, ktoré by mohli v budúcom ročníku zrýchliť a zefektívniť prípravu interaktívnych úloh. Jedným z nich je program na generovanie pozícií kartičiek, ktoré majú nepravidelné umiestnenie na hracej ploche teda sú na hracej ploche rozložené podľa toho, ako vyzerá pozadie úlohy a ako ich umiestnenie navrhol tvorca úlohy. Druhým program by mal dokázať generovať kartičky s textom.

5.1 Generovanie programu s pozíciami kartičiek

Pri tvorbe interaktívnych úloh môžeme rozlišovať dva druhy umiestnenia kartičiek a cieľových pozícií. Ak sú kartičky pravidelne rozmiestnené pozíciách v nejakom rastri, je ich generovanie veľmi jednoduché a dá sa urobiť priamo v JavaScript-ovom kóde pomocou cyklu. Iná situácia nastane, ak autor úlohy presne určil umiestnenie kartičiek do konkrétneho pozadia. Vtedy sa pozície pre kartičky nedajú generovať cyklom a bolo by vhodné, keby mal programátor k dispozícii editor dizajnu, ktorý by umožňoval načítať pozadie, sadu kartičiek a prázdnu kartičku. Programátor potom prázdnu kartičku postupne umiestni na cieľové pozície, ktoré určil autor úlohy pre dané pozadie. Takto navrhnutý program by umožňoval nielen interaktívne ťahať prázdnu kartičku na miesta, kam pri riešení patria jednotlivé kartičky, ale dokázal by z takto pripraveného dizajnu priamo vygenerovať JavaScript–ový kód, pozri Obrázok 6, dolu. Autorovi interaktívnej úlohy by potom stačilo takto vygenerované súradnice a názvy súborov priamo preniesť do jeho programu pre danú interaktívnu úlohu.

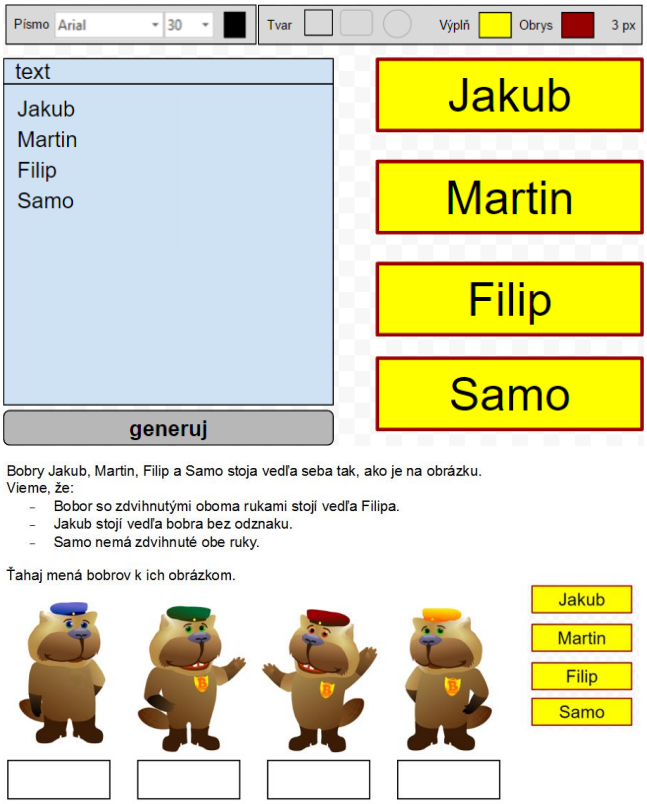


Obrázok 6. Program na návrh interaktívnych úloh s presným umiestnením kartičiek

5.2 Generovanie kartičiek s textom

V niektorých interaktívnych úlohách sú na kartičkách, ktoré má žiak umiestniť na cieľové pozície, mená alebo čísla, resp. iné, jednoduché, texty, pozri Obrázok 7. Pri tvorbe takýchto úloh je v súčasnosti potrebné každú kartičku vytvoriť v niektorom grafickom editore a každú z nich zapísať ako samostatný obrázkový súbor. Takýto typ práce pri príprave a programovaní úloh nie je veľmi tvorivý a dal by sa zautomatizovať. Preto sme navrhli vytvoriť samostatný program, Editor kartičiek, ktorý by dokázal pre zadaný text vygenerovať sadu kartičiek s textom, ktoré by mali rovnaké vlastnosti – veľkosť, farbu výplne

a orámovania, farbu, typ a veľkosť písma. Program by mal umožniť export kartičiek do samostatných súborov v png, resp. v svg formáte. Zatiaľ sme pripravili návrh dizajnu takéhoto programu, pozri Obrázok 7. Podobný program začína v súčasnosti vznikať v rámci predmetu pre študentov aplikovanej informatiky na FMFI UK v Bratislave.



Obrázok 7. Program na generovanie kartičiek s textom. Použitie vygenerovaných kartičiek v úlohe Kto je kto. 2016/17, Kadet

6. ZÁVER

Interaktívne úlohy tvoria v súťaži iBobor takmer tretinu súťažných úloh. Tento školský rok sme zmenili technológiu ich tvorby a namiesto Flash sme začali využívať JavaScript. Aby bolo programovanie týchto úloh jednoduchšie, navrhli a iteratívne sme vyvinuli knižnicu `sprites.js`, ktorá zabezpečuje základnú komunikáciu s webovým prehliadačom, prácu s obrázkami a udalosťami myši. Tiež sme pripravili niekoľko šablón, ktoré umožnia programátorom efektívne a pomerne rutinne vyvíjať úlohy súťaže. Pri príprave úloh sme zistili, že niektoré činnosti ani tak nesúvisia s programovaním ako so samotným vývojom úloh a tak sme navrhli dizajn a funkčnosť dvoch podporných programov na generovanie kartičiek s textom a dizajnovanie úloh s nepravidelným umiestnením kartičiek.

PODAKOVANIE

Článok vznikol v rámci projektu KEGA Podpora tvorby vzdelávacieho digitálneho obsahu pre mobilné zariadenia (č. 080UK-4/2015).

BIBLIOGRAFICKÉ ODKAZY

[1] Hlavná stránka súťaže Informatický bobor. http://www.iBobor.sk

[2] SALANCI, Ľ: knižnica sprites.js. nepublikované, pre internú potrebu KZVI FMFI UK.

[3] TOMCSÁNYIOVÁ, M.: Interaktívne úlohy v súťaži Informatický bobor [elektronický dokument] CD. *DidInfo 2009*. - Banská Bystrica : Univerzita Mateja Bela, 2009, s. 174-177., ISBN 978-80-8083-720-4

Krátke príspevky

Short Papers

Virtuálna realita a jej využitie v liečbe fóbií
Virtual reality and it`s use in the phobia treatment

Dana Horváthová
Katedra informatiky,
FPV UMB Tajovského 40
974 01 Banská Bystrica Slovensko
dana.horvathova@umb.sk

Vladimír Siládi
Katedra informatiky,
FPV UMB Tajovského 40
974 01 Banská Bystrica Slovensko
vladimir.siladi@umb.sk

ABSTRACT
Virtual reality tools can help to make the treatment of certain types of phobias more efficient. Our paper introduces the methods of creating virtual environments, models and applications designed for medical therapies of patients at the Department of Computer Science, FNS, Matej Bel University in Banská Bystrica.

Keywords
Virtual reality. Virtual environments. Treatment of phobias.

ABSTRAKT
Nástroje virtuálnej reality môžu efektívne napomáhať pri liečbe niektorých typov fóbií. Náš príspevok prezentuje metódy tvorby virtuálnych prostredí, modelov a aplikácií, ktoré sme na katedre informatiky FPV UMB v Banskej Bystrici navrhli pre terapiu pacientov trpiacich fóbiami.

Kľúčové slová
Virtuálna realita. Virtuálne prostredie. Liečba fóbií.

1. ÚVOD
Dnešné technické vynálezy prenikajú veľkou mierou do všetkých oblastí nášho života. Nie je teda prekvapivé, že zasahujú aj do oblasti medicíny, či psychoterapie. V dôsledku toho môžu byť rôzne fóbie prekonané vďaka terapii pomocou virtuálnej reality. Jedná sa o naozajstný pokrok pre pacientov trpiacich fóbiami [1].

Na katedre informatiky, Fakulty prírodných vied, Univerzity Mateja Bela v Banskej Bystrici sa už niekoľko rokov venujeme problematike liečby fóbií pomocou virtuálnej reality. Pripravujeme materiály pre potreby psychoterapeuta na podporu liečby rôznych druhov fóbií, ako je napríklad strach z pavúkov (arachnofóbia), strach z plazov (herpetofóbia), alebo strach z výšky, hĺbky (hypsofóbia. akrofóbia), z otvoreného priestranstva (agorafóbia) a pod. Pre účely terapie vytvárame podklady vo forme modelov objektov, obrazov, videí, 360˚panorám a zvukov, simulujúcich reálne prostredia a situácie, ktoré majú psychoterapeutovi uľahčiť proces liečby a pacientovi urýchliť navodenie želaného stavu.

2. FÓBIE A ICH LIEČBA
V klinickej psychológii fóbia patrí medzi poruchy úzkosti. Tento termín popisuje nepretržitý strach, považovaný za neadekvátny a nepochopiteľný, z predmetu alebo zo situácie, ktorý núti bojacu sa osobu radšej sa tejto situácii či predmetu vyhnúť. Fóbie môžeme vo všeobecnosti rozdeliť do troch hlavných kategórií: špecifické fóbie, sociálne fóbie a agorafóbie. Moderná medicína ponúka viacero typov liečby týchto ochorení, ako sú najmä farmakologická a psychologická liečba alebo hypnoterapia [2].

V posledných rokoch sa vo svete začína využívať ďalšia metóda, ktorá umožňuje pacientom nový typ liečby: terapia prostredníctvom virtuálnej reality. V dôsledku toho, že ľudské zmysly sú veľmi ľahko oklamateľné, je jednoduché jedinca ponoriť do prostredia vytvoreného, modifikovaného a prezentovaného pomocou nástrojov virtuálnej reality.

Zahĺbenie sa do virtuálnej reality ponúka viacero výhod v porovnaní s vystavením pacienta reálnemu stimulu. Liečba sa môže uskutočniť v ordinácii terapeuta, čo je menej nákladné pre pacienta a menej obmedzujúce pre terapeuta, ktorý nemusí sprevádzať pacienta, aby ho vystavil reálnemu stimulu. Navyše, tento typ terapie sa môže použiť aj u pacientov, ktorí majú veľký strach zo skutočného podnetu [3].

3. TVORBA VIRTUÁLNEHO SVETA
Existuje niekoľko metód na vytvorenie virtuálneho prostredia [4].

Modelovanie prostredníctvom počítačovej grafiky, čo vyžaduje vysokú kvalitu použitých obrázkov, aby objekt vyzeral reálne, ale aj kompatibilitu zariadení, potrebných na správnu funkciu aplikácie. Táto metóda je vhodná na tvorbu objektov, ktoré by sme snímaním reálneho sveta získali len veľmi ťažko.

Animácia 3D modelu, ktorý je možné pomocou animačného programu nielen rozpohybovať, ale je možné mu priradiť zmenu farby, tvaru, veľkosti, či trajektórie. V prípade niektorých objektov postačuje vytvoriť jeden model, ktorý sa potom nakopíruje do scény a rozanimuje v požadovanom množstve, napr. pavúky, hady a pod.

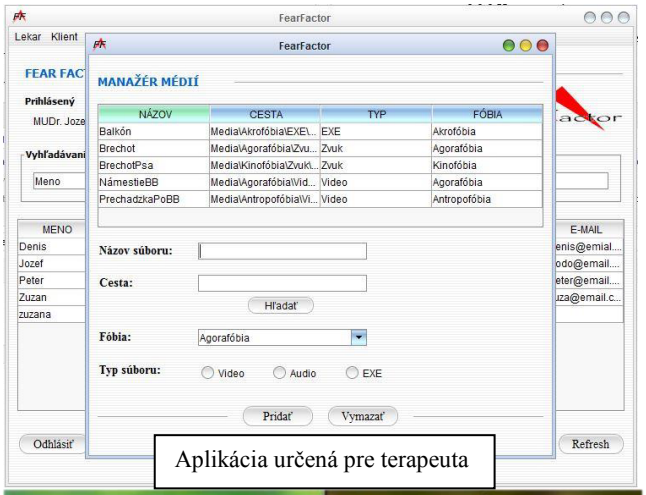
Panoramatická fotografia alebo video, ktoré zachytáva obraz prípadne dianie scény v 360 stupňoch. Táto metóda umožňuje vytvoriť virtuálne prostredie jednoduchšie ako modelovaním, pod podmienkou, že fotografovaná alebo filmovaná scéna je jednoducho prístupná.

Kombinácia z uvedených troch metód umožňuje prispôsobiť vzhľad prostredia a objektov v ňom individuálnym potrebám pacienta. Čím reálnejšie dokáže virtuálna realita zobraziť simulované prostredie a objekty v ňom, tým je väčšia šanca na oklamanie zmyslov a navodenie pocitu, že človek sa v tom prostredí naozaj nachádza.

4. APLIKÁCIA URČENÁ NA TERAPIU
Ďalšou pomôckou, ktorú sme vytvorili pre psychoterapeuta je aplikácia, ktorá mu umožňuje nielen evidovať dôležité informácie o pacientovi (meno a základné informácie o pacientovi, druh liečenej fóbie, dátumy kontrol, priebeh terapie, posledný stav pacienta a pod.), ale aj promptne spúšťať podľa potreby a stavu

pacienta jednotlivé situácie, či prostredia [5]. Do tejto aplikácie sa nám podarilo vytvoriť v priebehu posledných rokov niekoľko objektov a prostredí, ktoré neustále dopĺňame o nové a nové médiá. Vďaka zapožičanému hardvéru (dátová rukavica, Head tracker – snímač pohybu hlavy, virtuálne okuliare, panoramatická kamera a pod.) overujeme spôsob a kvalitu zobrazenia, a prispôsobujeme všetko požiadavkám terapie.

5. UKÁŽKY VYTVORENÝCH PRÁC



Aplikácia určená pre terapeuta

Strehúň škvrnitý

Prostredie na liečbu hypsofóbie

Prostredie námestia na liečbu agorafóbie

Prostredie lesa na liečbu agorafóbie

Model hada na liečbu ofidiofóbie

Obrázok 1. Ukážky prác

6. ZÁVER

Náš výskum na KI FPV UMB sa rozbehol vďaka námetu Dr. Jána Záskalana (Mind Solutions) a materiálnej podpore firmy Nettech s.r.o. (zastúpenou pánom Petrom Sojkom). Do spolupráce sa zapojila aj katedra pedagogiky a psychológie PF UMB (zastúpená pani Evou Lackovou)

Keďže sme ešte len na začiatku a nemáme spätnú väzbu od skutočných pacientov, testovacia fáza prebieha len v laboratórnych podmienkach na simulovaných pacientoch. Veríme, že naše úsilie, aj ďalších výskumníkov v dohľadnej dobe aj na Slovensku prinesie potvrdenie našej hypotézy o efektívnosti liečby fóbií prostredníctvom virtuálnej reality.

BIBLIOGRAFICKÉ ODKAZY

[1] HORVÁTHOVÁ, D., SILÁDI, V., SOJKA, P.: *Využitie virtuálnej reality na liečenie fóbií.* In: Zborník abstraktov z podujatia Konferencia NITT SK 2014 Transfer technológií na Slovensku a v zahraničí. pp 44, ISBN .978-80-89354-33-7. (34:33:33). AFH Abstrakty príspevkov z domácich konferencií

[2] HORVÁTHOVÁ, D., SILÁDI, V., LACKOVÁ, E.: *Phobia treatment by help of virtual reality*. In Informatics 2015 Proceedings. IEEE 13th International Scientific Conference on Informatics, November 18-20, 2015 Poprad, ISBN 978-1-4673-9867-1, IEEE catalog number CFP15E80-PRT

[3] MOLNÁR, R., HORVÁTHOVÁ, D.: *Tvorba interaktívneho prostredia na liečbu fóbií.* In.: Zborník príspevkov zo ŠVK Prírodovedec 2016, s. 28-37, - Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, ISBN 978-80-557-1127-0

[4] HORVÁTHOVÁ, D., SILÁDI, V.: *Creating virtual environments for phobia treatment.* In: Open Computer Science. Volume 6, Issue 1, ISSN (Online) 2299-1093, DOI: 10.1515/comp-2016-0012, October 2016

[5] HORVÁTHOVÁ, D., SILÁDI, V., MOLNÁR, R., VITÉZ R., ŠLAJFERČÍK, L., PALKOVIČ, M., OBROČNÍK, J., TÓTH, J.: *Liečba fóbií s podporou virtuálnej reality.* In: Zborník príspevkov zo ŠVK Prírodovedec 2014. - S. 248-255. - Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, ISBN 978-80-557-0721-1

Vyučovanie úvodných kurzov databázových systémov a medzinárodná spolupráca

Teaching introductory courses of database systems and international cooperation

Jana Jacková
Univerzita Matej Bela, FPV
Tajovského 40
974 01 Banská Bystrica
Slovakia
jana.jackova@umb.sk

Liviana Nicoleta Tudor
Petroleum-Gas University of Ploiesti
Bd. Bucuresti Nr. 39
Ploiesti,100680
Romania
LTudor@upgploiesti.ro

ABSTRACT

We inform about international cooperation in teaching introductory courses of database systems in bachelor study between the Department of the Faculty of Natural Sciences, Matej Bel University and the Department of Computer Science, Information Technology, Mathematics and Physics from the Petroleum-Gas University Ploiesti in 2016 and 2017. This cooperation is within Erasmus+ teacher stay mobilities and work on some common documents and activities.

Keywords

Database Systems. Curriculum. SQL. Bachelor study. Teaching. Erasmus+. Cooperation.

ABSTRAKT

Príspevok informuje o medzinárodnej spolupráci v rámci vyučovania úvodných kurzov databázových systémov v bakalárskom štúdiu medzi Katedrou informatiky Fakulty prírodných vied Univerzity Mateja Bela v Banskej Bystrici a Katedrou informatiky, informačných technológií, matematiky a fyziky z Petroleum Gas-University of Ploiesti v rokoch 2016 a 2017. Táto spolupráca prebieha v rámci učiteľských mobilít programu Erasmus+ a práce na niektorých spoločných dokumentoch a aktivitách.

Kľúčové slová

Databázové systémy. Kurikulum. SQL. Bakalárske štúdium. Vyučovanie. Erasmus+. Spolupráca.

1. ÚVOD

Na základe zistenia spoločného záujmu autoriek príspevku o problematiku vyučovania databázových systémov a rôzne výučbové prostredia [1, 2], sme v roku 2016 iniciovali bilaterálnu spoluprácu medzi našimi univerzitami pre mobility v rámci medzinárodného programu Erasmus+ [3] učiteľov a študentov.

Na Katedre informatiky Fakulty prírodných vied Univerzity Mateja Bela v Banskej Bystrici [4] (ďalej FPV UMB) sa vyučuje úvodný kurz databázových systémov v zimnom semestri 2. ročníka bakalárskeho štúdia študijných odborov Aplikovaná informatika a Učiteľstvo informatiky v kombinácii predmetov. Katedra informatiky, informačných technológií, matematiky a fyziky z univerzity v Ploiesti (Petroleum-Gas University of Ploiesti, fakulta Faculty of Letters and Sciences) [5] poskytuje základné znalosti z databázových systémov v letnom semestri 2. ročníka bakalárskeho štúdia aj v rámci predmetov Spracovanie dát a prezentácia (Data Processing and Presentation) a Spracovanie dát a informačné systémy v ekonomickej doméne (Data Processing and Information Systems in the Economic Domain) pre niektoré študijné odbory v rámci 2 fakúlt tejto univerzity. Na obidvoch pracoviskách používame pri výučbe týchto predmetov rovnaké databázové platformy (MS Access, Oracle) a na podporu niektorých aktivít aj e-learningové prostredia v Moodle [6, 7].

2. SPOLUPRÁCA V RÁMCI ERASMUS+

Na základe zistenia podobného kurikula v úvodných kurzoch databázových systémov na pracoviskách autoriek príspevku sme v akademickom roku 2015/16 zahájili vzájomnú spoluprácu v oblasti vyučovania databázových systémov.

V dňoch 22. – 26. septembra 2016 sa učiteľskej mobility v rámci programu Erasmus+ [3] zúčastnila na Katedre informatiky FPV UMB Liviana Tudor (Obrázok 1).

Obrázok 1. Učiteľská mobilita Erasmus+ Dr. Liviany Tudor z Petroleum-Gas University, Rumunsko, v Banskej Bystrici v septembri 2016 (dole, druhá sprava)

Počas tejto mobility Dr. Tudor odprednášala v 1.týždni zimného semestra akademického roka 2016/17 študentom predmetu Databázové systémy 1 úvod do databázových systémov a relačných

databáz. Na cvičeniach sa študenti pod jej vedením (Obrázok 2) oboznámili so základmi práce v prostredí Oracle (Oracle Database Express Edition) [8]. Výučba prebehla v anglickom jazyku. Zaujímavosťou bolo, že v tejto študijnej skupine bol zapísaný aj riadny študent FPV UMB, ktorý má trvalý pobyt v Rumunsku a ovláda slovenský aj rumunský jazyk (na strednej škole študoval v Rumunsku, vyučovacím jazykom na jeho škole bola aj slovenčina). Takže študenti predmetu mali možnosť prípadného pretlmočenia aj z rumunčiny do slovenčiny. Túto možnosť nebolo potrebné pri výučbe využiť, ale po skončení výučby sa vyučujúca porozprávala s týmto študentom v rumunčine.

Obrázok 2. Vyučovanie predmetu Databázové systémy 1 na Katedre informatiky FPV UMB s Dr. Tudor, 26. 9. 2016

V letnom semestri akademického roka 2016/17 plánujeme uskutočniť obdobnú mobilitu na rumunskom pracovisku. Našim zámerom v rámci tejto spolupráce programu Erasmus+ je hlbšie sa oboznámiť s kurikulárnymi dokumentami týkujúcich sa vyučovania databázových systémov, čo by malo viesť k prehĺbeniu spolupráce na úrovni spoločnej tvorby niektorých dokumentov a aktivít príbuzných predmetov.

3. ĎALŠIA SPOLUPRÁCA

Cieľom našej spolupráce v rámci úvodných kurzov databázových systémov je skvalitnenie vyučovania predmetu, jeho zatraktívnenie a zlepšenie výstupných vedomostí študentov. Rozvinutie našej ďalšej spolupráce vidíme v dvoch nasledujúcich oblastiach.

V oblasti štandardných podporných študijných materiálov k predmetom pracujeme na slovenskom pracovisku najmä s učebnicami [9, 10, 11] a u študentov učiteľstva aj s [12]. Na rumunskom pracovisku sú podporné materiály sústredené na [8]. Inšpirované prácami Ahadi et al. [13, 14] sme začali pripravovať spoločné úlohy zo základov jazyka SQL a z návrhu relačnej databázy. Z týchto úloh plánujeme vyšpecifikovať typy úloh pre výstupný test, ktorý by na konci semestra mohli riešiť študenti na obidvoch pracoviskách. Naše výsledky by sme potom chceli porovnať s publikovanými výsledkami výskumov v tejto oblasti. Pri našej spolupráci v tomto projekte využívame platformu Moodle [7].

Keďže študenti na obidvoch pracoviskách vytvárajú v rámci týchto predmetov aj vlastné malé databázové projekty, radi by sme ich v budúcnosti tiež zapojili do vzájomnej spolupráce. Mohlo by ísť o spoluprácu v týchto formách: spoločná kolaboratívna práca na databázových projektoch, online prezentácia výsledných projektov alebo offline oboznámenie sa so študentskými prácami spolupracujúceho pracoviska, prípadne aj peer review [15] niektorých čiastkových výsledkov. Domnievame sa, že takéto aktivity by mohli prispieť k záujmu študentov o študentské mobility Erasmus+ medzi našimi univerzitami, ktoré by im mohli rozšíriť ich vyhliadky na uplatnenie v praxi.

BIBLIOGRAFICKÉ ODKAZY

[1] TUDOR, L. N. Virtual Learning Laboratory about Query Optimization against XML Data. In *ITiCSE '15*. New York : ACM, 2015, p. 348-348. DOI= http://dx.doi.org/10.1145/2729094.2764861

[2] JACKOVÁ, J. K navrhovaniu databázy a jej implementácii v MS Access v učebniciach pre stredoškolákov. In *DidInfo 2014*. Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, 2012, s. 63-68. ISBN 978-80-557-0698-6.

[3] *Erasmus+* [online]. [cit. 2017-02-27]. https://www.umb.sk/medzinarodne-vztahy/informacie/erasmus/

[4] *Katedra informatiky* [online]. [cit. 2017-02-27]. http://www.fpv.umb.sk/katedry/katedra-informatiky/

[5] *Departmentful ITIMF* [online]. [cit. 2017-02-27]. https://timf.upg-ploiesti.ro/www/home/

[6] *Virtuálna Univerzita Mateja Bela:Katedra informatiky FPV, 2. ročník (Bc.)* [online]. [cit. 2017-02-27]. https://lms2.umb.sk/course/index.php?categoryid=33

[7] *E-learning Platform* [online]. [cit. 2017-02-27]. programare.biz/moodle/

[8] TUDOR, L. N. *Baze de date Oracle* [online]. [cit. 2017-02-27]. http://www.programare.biz/BD/ORACLE/ora.html

[9] OPPEL, A. *Databáze bez předchozích znalostí.* Brno : Computer Press, 2006. 319 s. ISBN 80-251-1199-7.

[10] KRAJČI, S. *Databázové systémy.* Košice : Univerzita Pavla Jozefa Šafárika, 2005. 272 s. ISBN 80-7097-608-X.

[11] MATIAŠKO, K., VAJSOVÁ, M., ZÁBOVSKÝ, M., CHOCHLÍK, M. *Databázové systémy a technológie.* Bratislaava : STU, 2006. 693 s. ISBN 978-80-227-3035-8.

[12] KUBINCOVÁ, Z., CÁPAY, M., HRUŠECKÁ, A., SALANCI, Ľ. *Úvod do databáz*. Bratislava : Štátny pedagogický ústav, 2010. 40 s. ISBN 978-80-8118-093-4.

[13] AHADI, A., PRIOR, J., BEHBOOD, V., LISTER, R. Students' Semantic Mistakes in Writing Seven Different Types of SQL Queries. In *ITiCSE '16*. New York : ACM, 2016, p. 272-277. DOI= http://dx.doi.org/10.1145/2899415.2899464

[14] AHADI, A., BEHBOOD, VIHAVAINEN, A., PRIOR, J., LISTER, R. Students' Syntactic Mistakes in Writing Seven Different Types of SQL Queries and its Application to Predicting Students' Success. In *SIGCSE '16*. New York : ACM, p. 401-406. DOI= https://doi.org/10.1145/2839509.2844640

[15] HERZOG, M. A., KATZLINGER, E., STABAUER, M. Embedding Interuniversity Peer Review in Virtual Learning Groups. In *SETE 2016, LNCS 10108.* Cham : Springer, 2017, p. 614-623. DOI= https://doi.org/10.1007/978-3-319-52836-6_65

Cloud Technologies and the Taxonomy of Bloom

Meruert Serik
ENU Astana
010009, Astana
Kazakhstan
+77078432750
serik_meruerts@mail.ru

Jaroslav Kultan
EU Bratislava
82107, Bratislava
Slovakia
+4210904394892
jkultan@gmail.com

Aigul Sadvakassova
ENU Astana
010009, Astana
Kazakhstan
+77712020721
Sak79@bk.ru

Mira Rakhimzhanova
ENU Astana
010000, Astana
Kazakhstan
+77051656164
r.mira@mail.ru

ABSTRACT

This article describes issues related to study of cloud technologies in improving the training of future teachers of computer science. Distribution of cloud applications is presented according to the levels of Bloom's taxonomy. The teaching methodology of the disciplines introduced on cloud technologies is given in accordance with the levels of educational activity.

Keywords

Cloud technologies, Cloud storage, Bloom's taxonomy.

1. INTRODUCTION

Development and introduction of new educational disciplines related to cloud computing in educational process will allow modern graduates to get the necessary skills and knowledge for further successful work with the cloud.

There are several options for studying the cloud technologies. The first approach lies in the training of cloud technologies on the part of specialist, i.e. how to make applications for clouds, how to develop cloud software, how to ensure the safety of cloud applications and data storages.

The second is the user's approach, i.e., how to learn to use the tools of cloud technologies. This approach considers the use by graduates of their knowledge in the future pedagogical activity. For these purposes we have developed and implemented special courses for Bachelor's and Master's programs such as "Fundamentals of cloud technologies", "Cloud technologies and virtualization technologies". Moreover, we have also developed the academic cloud storage to be applied in the educational process.

To understand the educational objectives and levels of mastering of educational material on cloud technologies, we offer the following review of the cloud applications and tools divided into categories according to the level of learning objective under the Bloom's taxonomy [1], [2].

2. ABOUT BLOOM'S TAXONOMY

Referring to Bloom's taxonomy and putting the reviewed cloud services and applications in conformity with the levels of its hierarchy, it is possible to obtain a classification in terms of the use in educational activity of students. Such classification, in particular with respect to Google services, has been presented by Kathy Schrock [3].

The Benjamin Bloom's hierarchy of the levels of educational activity or rather its version made by his students (Lorin Anderson and others) may be presented as follows (Figure1).

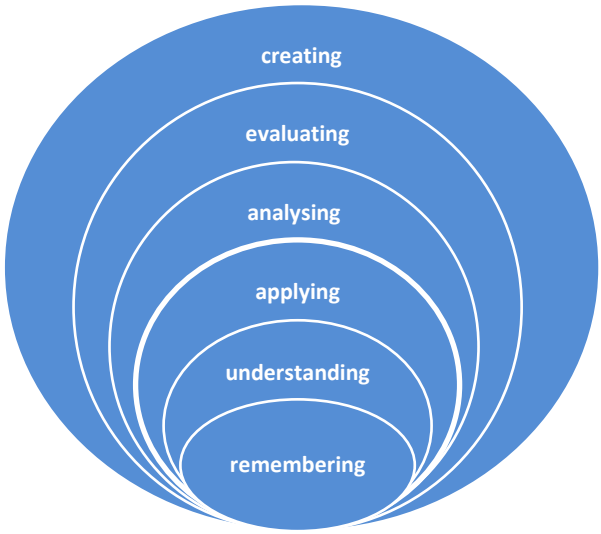


Figure 1. Bloom's Taxonomy

However, the discipline content should be built on the basis of the logic of the process of improving the future teacher training; the model is presented by the levels of formation of future computer science teacher's readiness to work with cloud technologies: remembering, understanding, applying, analysing, evaluating, creating. Besides, we define the last two points as a high degree of readiness. The distinctive characteristics of each stage consist in achieving the various objectives; each subsequent stage includes results of the previous one and can only exist on their basis.

Third section is described all of stages Bloom's Taxonomy in the using of the specific cloud technologies.

3. DISTRIBUTION OF CLOUD APPLICATIONS ACCORDING TO THE LEVELS OF BLOOM'S TAXONOMY

At the remembering level the student memorizes and reproduces the material studied, knows terminology, etc. Among detailed services and Internet services at this level the student may use social networks, e-mail, voice and video chats, search engines to study the relevant literature, discuss difficult moments with the teacher or

other students, questionnaires, etc. Tools for making questionnaires can be found among Google services and others [4], [5].

At the understanding level such services as team work with documents (we mean not only text documents, but also presentations and electronic worksheets), forums and chats rooms, search engines, wiki can be used. Now the student can interpret the algorithm studies, classify algorithms according to various signs while participating, for example, in creation of wiki-site page.

The applying level at which the student is able to apply this knowledge is also characterized by application of services of team work with documents, forums and chats. Furthermore, the student may participate in joint development of projects, knows how to use cloud services and how to perform cloud computing.

Analysing stage involves the student's abilities to allocate stages, types, classifications, compare different points of view and make schemes and lists the basic properties.

Evaluating stage involves the student's ability to adequately assess the level of applications and software products developed by him/her, to justify the concepts he/she works with, discuss, comment messages, and moderate information. Moreover, the student can independently assess and perform complete analysis of surveys developed within the virtual class and by external tools (Google Form).

Finally, at the last creating level the student develops own cloud storages himself on open platform (Owncloud) Figure 2.

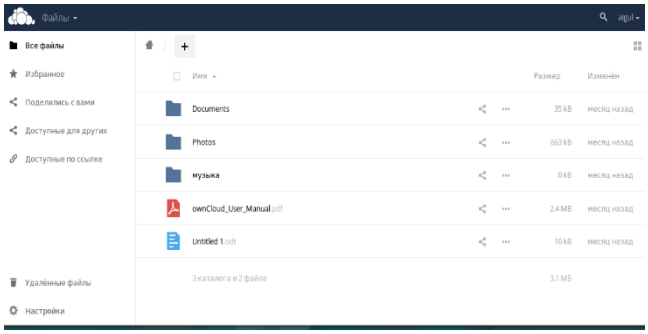


Figure 2. Created Owncloud storage by students

In addition, student plans activities to work on it using cloud calendars (Google Calendar), and can distribute it via the planners (Owncloud planner). As a result of his work the student creates and presents the obtained the software product (Adobe Acrobat.com Presentations, Prezi). Also during learning process students learn to make their own cloud storage in local network with the help of OwnCloud service, study possibilities of its usage in the learning process and their future professional activities.

4. CONCLUSION

We have tried to present the modern landscape of cloud technologies, as well as our own understanding of this sphere and opportunities opened up by it.

In our view, cloud technologies offer the possibilities for educational institutions to effectively organize educational activities based on the experience of the best universities. Understanding the subject of cloud technologies is important for teachers both from the perspective of training professionals who plan to work in schools and from the users' point of view: we should teach students how to use the tools and resources of cloud technologies.

REFERENCES

[1] Bloom's Digital Taxonomy / Educational Origami [Electronic resource]. – URL: http://edorigami.wikispaces.com/Bloom's+Digital+Taxonomy

[2] Bloom's and ICT tools/ Educational Origami [Electronic resource]. – URL: http://edorigami.wikispaces.com/Bloom%27s+and+ICT+tools

[3] Bloomin' Apps // Kathy Schrock's Guide to Everything [Electronic resource]. – Last updated: 10.01.12. – 2011. – URL: http://www.schrockguide.net/bloomin-apps.html

[4] M. Serik cloud-based / manual. - Astana, 2015. 93p

[5] M. Serik, J. Kultan, A.Sadvakasova Implementation of cloud technology in education process //Journal of knowledge society /international scientific journal/ ISSN 2336-2561, Číslo 2/2016, Roč. 4. , 88-94p.

Zapojenie študentov do riešenia projektu APVV s názvom Automatické spracovávanie trasologických objektov

Involving the students into the APVV project with the name Automatic processing of traceology objects

Miroslav Melicherčík
Katedra informatiky,
FPV UMB Tajovského 40
974 01 Banská Bystrica
Slovenská republika
miroslav.melichercik@umb.sk

Alžbeta Michalíková
Katedra informatiky,
FPV UMB Tajovského 40
974 01 Banská Bystrica
Slovenská republika
alzbeta.michalikova@umb.sk

Michal Vagač
Katedra informatiky,
FPV UMB Tajovského 40
974 01 Banská Bystrica
Slovenská republika
michal.vagac@umb.sk

ABSTRACT

In this paper we describe the process of development of application for the automatic processing of traceology objects developed in the APVV project and we point out the possibility of involvement of students in the various stages of its solution. In the first step we analyze the topic and aims of the project, then we focus on problems which were necessary to overcome during the working on project where the individual problems were solved through cooperation between teachers and students at our department. At the end of the article we mention the results, which were achieved in the project as well as the outcomes of students' work.

Keywords

APVV project. Traceology. Cooperation of students and teachers.

ABSTRAKT

V tomto článku opíšeme postup vývoja aplikácie pre Automatické spracovávanie trasologických objektov vyvíjanej v rámci projektu APVV a poukážeme na možnosti zapojenia študentov v jednotlivých fázach jeho riešenia. Najprv uvedieme riešenú problematiku, poukážeme na problémy, ktoré bolo počas riešenia projektu potrebné prekonať, pričom jednotlivé problémy boli riešené pomocou vzájomnej spolupráce pedagógov a študentov našej katedry. Na záver uvedieme dosiahnuté výsledky jednak samotného projektu, ako aj výstupy práce študentov.

Kľúčové slová

Projekt APVV. Trasológia. Spolupráca študentov a pedagógov.

1. ÚVOD

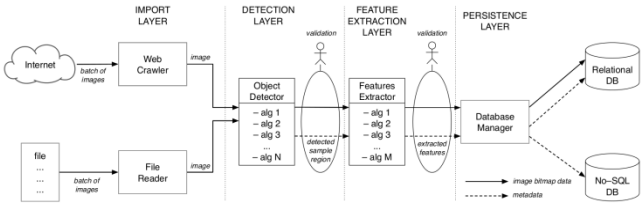
Na Katedre informatiky Fakulty prírodných vied UMB v Banskej Bystrici sa v súčasnej dobe zaoberáme vývojom aplikácie pre Automatické spracovávanie trasologických objektov v rámci projektu APVV. Na riešení projektu spolupracujú tri organizácie a to Fakulta prírodných vied UMB v Banskej Bystrici, Kriminalistický a expertízny ústav Policajného zboru SR a Matematický ústav SAV. Hlavným cieľom projektu je vytvorenie funkčnej aplikácie, ktorá bude schopná automaticky rozpoznať charakteristické atribúty zadaných trasologických objektov (odtlačky podošiev a pneumatík) a vybrať z rádovo 10 000 podobných objektov úzku množinu objektov, ktoré sú s nimi vizuálne najpodobnejšie. Vytvorenie takejto aplikácie predpokladá kvalitné znalosti z oblasti programovania, takisto predpokladá získanie vedomostí a zručností pri práci s metódami počítačového videnia. Pri vytváraní aplikácie sa ukázalo ako veľmi nápomocné ponúknuť študentom katedry informatiky možnosť participovať na riešení projektu. Navyše na našej katedre predmet počítačové videnie nie je zaradený ako vyučovací predmet, preto sme sa rozhodli vytvoriť špeciálny seminár, v ktorom sa stretávajú študenti z rôznych ročníkov a spolu s pedagógmi z katedry riešia čiastkové úlohy tohto projektu. Osvedčilo sa nám zadávať čiastkové úlohy projektu aj ako témy bakalárskych, resp. diplomových prác.

2. TRASOLÓGIA

Jedným z odvetví vedného odboru kriminalistika je trasológia, ktoré patrí medzi klasické kriminalistické disciplíny a dlhodobo významnou mierou prispieva k objasňovaniu trestnej činnosti. Predmetom trasológie je najmä zaisťovanie, skúmanie a porovnávanie odtlačkov podošiev obuvi a pneumatík, zaistených v súvislosti s objasňovanou kriminalisticky relevantnou udalosťou. K identifikácii trasologických stôp s objektom, ktorý ich vytvoril, je nevyhnutné disponovať vhodným porovnávacím materiálom, ktorý v čase porovnávania obsahuje identifikačné znaky zobrazené na stope. K tomuto účelu forenzné laboratóriá disponujú zbierkami a databázami obrazových vzorov dezénov podošiev obuvi a vzorov dezénov pneumatík. Zbierky a databázy je potrebné udržiavať aktualizované, permanentne ich dopĺňať novými vzorkami, napr. pri uvedení nových sezónnych kolekcií obuvi a pneumatík. Obsahovo sa jedná o zmes rôznorodých digitálnych fotografií, ktoré boli získané napr. fotografovaním kontrolných odtlačkov vyhotovených na papieri, fotografovaním dostupných objektov priamo v obchodnej sieti, alebo získané a spracované z rôznych reklamných a propagačných materiálov obchodníkov vrátane webových lokalít. A práve získanie vhodných obrazových vzoriek z webových stránok a ich automatické spracovanie je cieľom tohto projektu.

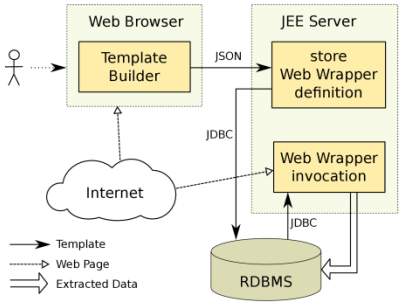
3. NÁVRH APLIKÁCIE

Navrhovaná aplikácia bude slúžiť na získavanie obrazových vzoriek z webových stránok a ich automatické spracovávanie. Vzhľadom na rozsah príspevku sa zameriame len na opis hlavných krokov pri spracovávaní odtlačkov podošiev. V projekte sa venujeme aj spracovávaniu odtlačkov pneumatík, ktoré tu neuvádzame [1]. Jednotlivé kroky tvorby aplikácie popisuje schéma algoritmu (pozri Obrázok 1).



Obrázok 1. Schéma systému pre automatické spracovávanie trasologických objektov [2]

Na získanie obrázkov z internetových obchodov sme vyvinuli tzv. Web Crawler (Obrázok 2), ktorý spolu s obrázkom uloží aj informácie o obrázku z danej webovej stránky.



Obrázok 2. Schéma systému Web Crawler [3]

Získané obrázky sú rôznej kvality a rôzneho typu. Ideálne by bolo, keby sme získali obrázok samostatnej podošvy topánky (Obrázok 3). Často sa však stretávame s tým, že podošva topánky je čiastočne prekrytá iným objektom (Obrázok 4). V našej práci používame aj takéto obrázky. V ďalšom kroku program na obrázku identifikuje podošvu topánky (Obrázok 5a), následne na podošve určí hlavné príznaky (Obrázok 5b), extrahuje ich z obrázka (Obrázok 5c) a porovná so vzorkou získanou z miesta činu (Obrázok 6). Na základe porovnania systém určí zadaný počet obrázkov najviac podobných so vzorkou z miesta činu.

Obrázok 3. Podošva topánky vhodná na spracovanie

Obrázok 4. Podošva topánky čiastočne prekrytá iným objektom

Obrázok 5. (a) Identifikácia podošvy topánky, (b) Určenie príznakov, (c) Extrakcia príznakov

Obrázok 6. Porovnanie príznakov so vzorkou získanou z miesta činu

4. VÝSLEDKY A ZÁVER

Pri riešení jednotlivých krokov algoritmu aplikácie sme odskúšali viacero metód a na základe získaných výsledkov sme vždy implementovali najlepšiu z nich. Do riešenia projektu sme zapojili študentov Katedry informatiky Fakulty prírodných vied UMB v Banskej Bystrici. Výsledkom spolupráce je 33 bakalárskych a 25 diplomových prác, z ktorých 34 už bolo úspešne obhájených. Okrem toho sa niekoľko študentov zúčastnilo Študentskej vedeckej konferencie, pričom jeden z príspevkov skončil na 3. mieste v rámci Česko – Slovenskej študentskej vedeckej konferencie. Okrem vzniknutých záverečných prác vidíme veľký prínos zapojenia študentov najmä v tom, že sa naučili pracovať v tíme, stretli sa s riešením reálnych problémov z praxe, kde mohli uplatniť vedomosti získané počas štúdia aj samostatným štúdiom vedeckých materiálov.

POĎAKOVANIE

Tento príspevok vznikol vďaka podpore Agentúry na podporu výskumu a vývoja, konkrétne projektu s číslom APVV-0219-12.

BIBLIOGRAFICKÉ ODKAZY

[1] MICHALÍKOVÁ, A., VAGAČ, M. A Tire Tread Pattern Detection Based on Fuzzy Logic. In *Flexible Query Answering Systems 2015*. Springer International Publishing, 2016, p. 381-388.

[2] VAGAČ, M., MELICHERČÍK, M. Improving image processing performance using database user-defined functions. In *International Conference on Artificial Intelligence and Soft Computing*. Springer International Publishing, 2015, p. 789-799.

[3] VAGAČ, M. et al. Crawling images with web browser support. In *Scientific Conference on Informatics*, 2015 IEEE 13th International. IEEE, 2015, p. 286-289.

Sekcia Edu Track na konferencii PyCon SK 2017
The Edu Track of PyCon SK 2017

Eva Mészárosová
KZVI FMFI UK
Mlynská dolina
842 48 Bratislava
Slovensko
meszarosova@fmph.uniba.sk

Monika Tomcsányiová
KZVI FMFI UK
Mlynská dolina
842 48 Bratislava
Slovensko
tomcsanyiova@fmph.uniba.sk

ABSTRACT
In this paper we describe the content of the Edu Track, section for teachers, at the conference PyCon SK 2017. The purpose of this section was to introduce teachers the benefits of the programming language Python and ways of how to use it for educational purposes.

Keywords
Teaching programming. Python. Conference PyCon SK.

ABSTRAKT
V príspevku popíšeme prednášky, ktoré odzneli v rámci sekcie Edu Track, ktorá bola súčasťou konferencie PyCon SK 2017. Táto sekcia bola primárne určená učiteľom informatiky a jej cieľom bolo priblížiť učiteľom možnosti, vlastnosti a výhody používania programovacieho jazyka Python vo vyučovaní.

Kľúčové slová
Vyučovanie programovania. Python. Konferencia PyCon SK.

1. ÚVOD
Konferencie PyCon sú určené pre pythonovskú komunitu a sú organizované nielen v Európe (PyCon UK, PyCon FI, PyCon FR, PyCon DE), ale aj celosvetovo, ako napríklad PyCon US, PyCon AU (Australia), PyCon India, PyCon ZA (South Africa), PyCon PH (Philippines), PyCon Argentina, pozri [1]. Na viacerých z nich je zvykom zaradiť do programu aj prednášky o vyučovaní programovania v jazyku Python, ktoré sú určené pre učiteľov a žiakov na stredných školách a univerzitách. Keďže sa jazyk Python dostáva aj do povedomia slovenských učiteľov, rozhodli sme sa v rámci konferencie PyCon SK 2017 zorganizovať sekciu prednášok pre učiteľov informatiky Edu Track, pozri [2]. Naším cieľom bolo priblížiť učiteľom možnosti používania jazyka Python vo vyučovaní. Prednášky boli primárne určené učiteľom, ktorí sa zatiaľ nestretli s jazykom Python, avšak obsah sekcie Edu Track sme sa snažili prispôsobiť tak, aby boli vhodné a užitočné aj pre učiteľov, ktorí v tomto jazyku už programovanie vyučujú. Veríme tiež, že vďaka tejto aktivite na konferencii PyCon sa začne budovať komunita slovenských učiteľov, ktorí si budú navzájom pomáhať, radiť a zdieľať svoje skúsenosti z vyučovania programovania na rôznych typoch stredných škôl.

2. PREDNÁŠKY SEKCIE EDU TRACK
Ako hlavný rečník sekcie Edu Track vystúpil RNDr. Andrej Blaho, PhD. z FMFI Univerzity Komenského v Bratislave. Je vysokoškolský učiteľ, ktorý štvrtý rok vyučuje programovanie v jazyku Python pre odbor Aplikovaná informatika. Ďalším pozvaným rečníkom bol prof. RNDr. Ivan Kalaš, PhD., ktorý sa sústredil na všeobecnejšie otázky, ktoré sa bezprostredne dotýkajú vyučovania programovania na našich školách. Ako ďalší prednášajúci vystúpili učitelia (a dokonca aj jeden žiak), ktorí z rôznych dôvodov začali vyučovať programovanie v jazyku Python. Vo svojich prezentáciách predstavili aké pedagogické prístupy, motivácie a materiály používajú vo svojom vyučovaní.

Andrej Blaho: Prečo Python?
Cieľom prednášky bolo oboznámiť učiteľov s jazykom Python a jeho kladnými a zápornými vlastnosťami pre vyučovanie základov programovania. Väčšina učiteľov v súčasnosti používa programovací jazyk Pascal (prostredie Lazarus alebo Delphi). V rámci tejto prednášky mohli učitelia vidieť a priamo na častiach kódu porovnávať zápisy programov v oboch jazykoch. Mohli tak bezprostredne vidieť rozdiely, s ktorými sa stretnú, ak sa vo svojom vyučovaní rozhodnú použiť jazyk Python.

Ivan Kalaš: Programovanie v škole: Primerané, udržateľné, rešpektované a integrované. Platí či neplatí?
Prof. Kalaš vo svojej prednáške poukázal na to, že hoci viacero krajín deklaruje zavedenie edukačného programovania do povinného predmetu s názvami ako ICT, Computing, či IKT, je pomerne ťažké nájsť jeho systematickú, vývinovo primeranú, rešpektovanú a správne podporenú implementáciu. V prednáške odzneli otázky, v ktorých sa prof. Kalaš zamýšľal na tým, čím sa líši súčasná vlna obnoveného záujmu o edukačné programovanie od tých predchádzajúcich. Prednášajúci hľadal aj zdôvodnenia, prečo je tento cieľ dosiahnuteľný a ktoré faktory musíme zvažovať, aby sme dosiahli všeobecne akceptované a stabilné školské programovanie pre každého.

Eva Mészárosová: Základy programovania v jazyku Python pomocou korytnačej grafiky
Prednáška bola zameraná na predstavenie metodického materiálu pre vyučovanie základov programovania, pozri [3], pre cieľovú skupinu učiteľov a žiakov 1. ročníka gymnázia. Používanie korytnačej grafiky vo vyučovaní programovania podporuje učenie sa objavovaním a skúmaním. V metodickom materiáli sa pomocou tohto prístupu žiaci oboznámia so základnými pojmy a konceptami programovania v jazyku Python. Dôležité je, že sa zároveň zoznámia s vyšším programovacím jazykom. V prednáške odzneli aj skúsenosti z vývoja, overovania a implementovania metodického materiálu do vyučovania základov programovania pre žiakov prvého ročníka gymnázia.

Marek Mansell: MicroPython – Python a hardvér patria k sebe
Na tejto prednáške bolo zaujímavé, že ju prezentoval žiak strednej školy. Predstavil v nej jazyk MicroPython, ktorý je určený pre programovanie mikroprocesorov. Marek Mansell ukázal učiteľom prácu s hardvérom, a keďže má sám skúsenosti s vedením workshopov, ponúkol tiež rady a tipy pre začiatočníkov. V závere

prednášky ukázal niekoľko zaujímavých námetov pre využitie hardvéru na vyučovaní informatiky, s prierezovými témami s fyzikou a biológiou.

Jaroslav Výbošťok: Skúsenosti s vyučovaním programovania v jazyku Python na strednej škole
Pán učiteľ Výbošťok z gymnázia v Bratislave sa v rámci svojej prednášky podelil o svoje skúsenosti s prechodom na vyučovanie programovania v jazyku Python. Predtým vo svojom vyučovaní používal jazyk Pascal, ale vďaka odporúčaniu svojich žiakov sa rozhodol, že prejde na Python. Vo svojej prednáške predstavil spôsob, akým vyučuje programovanie a aké motivácie a typy úloh používa v jazyku Python. Ukázal tiež niekoľko príkladov, ktoré jeho žiaci riešili na hodinách. Kombinuje v nich motiváciu práce s grafickou plochou a klikaním myšou s úlohami s matematickým kontextom a s výpismi textu v príkazovom režime IDLE. Takisto popísal problémy, s ktorými sa sám stretol počas prechodu na jazyk Python a uviedol niektoré svoje odporúčania pre učiteľov, ktorí sa rozhodli prejsť na vyučovanie programovania v jazyku Python.

Peter Kučera: Programujeme v Pythone
Peter Kučera je učiteľom na gymnáziu v Bratislave. Vo svojej prednáške opísal svoj prístup vyučovania programovania v jazyku Python, v ktorom sa rozhodol využiť motiváciu kreslenia do grafickej plochy a zmeny kresieb pomocou klikania myšou. Pre svoje vyučovanie navrhol materiál v rozsahu približne 33 vyučovacích hodín, príručku pre učiteľa a zošit, v ktorom uvádza testy a námety na rôzne projekty, pozri [4]. V prednáške odzneli aj jeho skúsenosti z vedenia kurzov jazyka Python pre učiteľov a informácie o aktivitách Klubu učiteľov informatiky.

Ľubomír Šnajder a Ján Guniš: Jazyk Python v programátorskej súťaži PALMA junior
Do programátorskej súťaže PALMA junior, ktorá je určená pre žiakov základných a stredných škôl, môžu od roku 2015 posielať súťažiaci svoje riešenia aj v jazyku Python. Predtým mohli úlohy riešiť len v jazyku Imagine Logo. Keďže prednášajúci sú zároveň autormi úloh, súčasťou ich prednášky boli predovšetkým ukážky zadaní úloh a to, ako zadania pripravujú. Motivácie úloh sa snažia čerpať z reálneho života, a keďže v súčasnosti môžu žiaci v riešeniach používať buď Python alebo Imagine Logo, je primárnou snahou autorov to, aby bol zápis riešenia súťažnej úlohy porovnateľne náročný v oboch jazykoch.

Zuzana Tkáčová: Zážitkové vyučovanie vo výučbe programovania na strednej škole
Ing. Tkáčová zoznámila učiteľov so zážitkovým vyučovaním programovania. Vo svojej prednáške predstavila niektoré zážitkové techniky, ako aj východiská a úskalia takéhoto prístupu. Podelila sa s nimi o svoje skúsenosti z ich uplatňovania v praxi pri vyučovaní programovania v jazyku Python. Počas prednášky si vyslúžila od účastníkov aj spontánny potlesk, keď spomenula percento žiakov, ktorí maturujú z informatiky na ich škole a podiel dievčat medzi týmito žiakmi.

Anino Belan: Ako funguje evolúcia?
Pán učiteľ Belan vo svojom vystúpení predstavil, ako využíva programovanie evolučných algoritmov pri vyučovaní témy funkcie s poliami v škole pre mimoriadne nadané deti. Pre svojich žiakov pripravil aj učebnicu, pomocou ktorého sa učia programovanie v jazyku Python vlastným tempom, pozri [5].

Michal Kaukič: Grafika v Jupyter notebookoch a vplyv programovacích prostriedkov na dušu študenta i učiteľa
V rámci prednášky ukázal Michal Kaukič možnosti využitia Jupyter notebookov, ich interaktívnych zmien a prvkov grafického rozhrania nielen pri vyučovaní informatiky, ale aj matematiky.

Prednášky, ktoré by mohli zaujať učiteľov informatiky odzneli aj mimo sekcie Edu Track. V sobotu v rámci anglického dňa konferencie vystúpil Valdemar Švábenský s prednáškou *Make learning great again: It's the how, not what*, v ktorej sa podelil o svoje skúsenosti s vyučovaním programovania na univerzite, pričom ako študent vedie cvičenia z programovania v jazyku Python. V nedeľu si účastníci konferencie mohli vypočuť prednášku Horst Jens s názvom *Teaching python to (your) children*, v ktorej prednášajúci porozprával o svojich skúsenostiach s vedením počítačových kurzov pre deti v Rakúsku. Konferenciu obohatili žiaci z Gymnázia Roudnice nad Labem, ktorí pripravili výstavu svojich diel.

3. ZÁVER

V rámci sekcie Edu Track sa stretli učitelia, ktorí už poznali Python, ale aj takí, ktorí oň majú záujem, pretože z rôznych zdrojov počuli o jeho výhodách pri vyučovaní základov programovania. Obsah prednášok sme sa snažili prispôsobiť týmto cieľovým skupinám tak, aby sa učitelia podrobnejšie a hlbšie zoznámili s jazykom Python a aby boli po ich absolvovaní motivovaní a rozhodli sa prejsť na vyučovanie programovania práve v tomto jazyku. Učitelia mohli v rámci prednášok vidieť viaceré spôsoby, ktorými môžu využiť vlastnosti jazyka Python v prospech kvality a efektívnosti svojho vyučovania programovania na strednej škole. Veríme, že teraz sa môžu kvalifikovanejšie rozhodnúť, či budú vo vyučovaní používať motiváciu korytnačej grafiky, prácu v grafickej ploche s klikaním myšou, či a ako použijú kombináciu príkazového režimu s okamžitou spätnou väzbou pre žiaka, alebo či budú žiakov učiť programovanie využitím hardvéru.

Konferencie sa zúčastnili učitelia stredných škôl, gymnázií a univerzít, takmer z celého Slovenska. Špeciálne na sekciu Edu Track prišli aj niekoľkí učitelia z Česka. V dotazníku napísali:

„Páčilo sa mi, že aj keď téma bola rovnaká, obsah bol rozdielny. Programovanie v IDLE, grafika, korytnačia grafika, zážitkové vzdelávanie...“

„...pre mňa to bolo veľmi prínosné pre moju ďalšiu výučbu... zloženie prednášateľov a programu bolo pestré a vyvážené“

„Ďakujem za organizáciu konferencie. Myslím, že mám dosť inšpirácií, ktoré mi pomôžu v septembri.“

“...inšpirácie, možnosti a diskusie. Výborné.”

Videozáznamy z prednášok sú zverejnené na YouTube kanáli SPy o.z., pozri [6]. Stretneme sa na Edu Track PyCon SK 2018.

BIBLIOGRAFICKÉ ODKAZY

[1] Python Software Foundation. www.python.org
[2] PyCon SK. https://www.pycon.sk/2017/
[3] MÉSZÁROSOVÁ, E. *Python a korytnačia grafika. Metodický materiál pre vyučovanie základov programovania pre gymnáziá*. Knižničné a edičné centrum, FMFI UK, Bratislava, 2017. ISBN 978-80-8147-080-6
[4] KUČERA, P. *Programujeme v Pythone. Učebnica informatiky pre stredné školy.* Mgr. Peter Kučera, 2016. ISBN 978-80-8147-080-6
[5] BELAN, A. *Python, eUčebnica pre septimu osemročného alebo 3. ročník štvorročného gymnázia.* Bratislava : Druska Books, 2013. ISBN: 978-80-89646-35-7
[6] YouTube. SPy o.z. www.youtube.com/channel/UC8Tzn82aG4zdvtqlrfsLgSA

Teaching Grid Technologies and Grid Computing to Undergraduate Students of Computer Science

Vladimír Siládi
Faculty of Natural Sciences, Matej Bel University
Tajovského 40
974 01 Banská Bystrica, Slovak Republic
+421 48 446 7126
vladimir.siladi@umb.sk

Ján Astaloš
Institute of Informatics, Slovak Academy of Sciences
Dúbravská cesta 9
845 07 Bratislava, Slovak Republic
+421 55 602 5123
jan.astalos@savba.sk

ABSTRACT
Grid computing courses are usually designed for PhD. students. Usually, undergraduate students have possibility to learn flying information within subjects related to distributed computing, etc. This paper summarizes experience in teaching a subject aimed at grid computing to bachelor students of applied informatics at Matej Bel University. The grid computing course has been developed with relationship to Slovak grid initiative and National project Slovak Infrastructure for High Performance Computing (SIVVP).

Keywords
Grid computing. Distributed computing. Computer science education. Best practices. Grid education. Hands-on training.

1. INTRODUCTION
Grid technologies and grid computing were new phenomenon at the end of the millennium (since 1997) as a new infrastructure for research of the 21st century [2]. The building of the Large Hadron Collider within CERN has had big influence on this trend. In Slovakia, some growth of the attention of researchers to grid computing could be observed, when the Slovak grid initiative has been established (2005). An availability of the grid infrastructure has opened a need to train its potential users. Mainly, researchers and PhD. students have been users of this infrastructure. A couple of educational forms have been offered to obtain fundamental knowledge and skills, e.g. on-line courses, grid-café by CERN, textbooks, workshops, testbeds, etc. In that time the grid computing entered into Computer Science programs originally as graduate-level topic [9] or within courses of undergraduate-level such as distributed computing or high-performance computing. Our aim was to teach this technologies undergraduate students of computer science (applied informatics) within autonomous subjects. We have begun to implement this kind of courses since 2006.

2. RELATED WORKS
Published works describe activities which are dated to the same period as process at our university. A part of publications, which deal with the presented topic, are focused on PhD students [3][4][5] and papers from universities in the USA have got the target group of undergraduate students [1][9][10][10]. A generalization of learning material for both kinds of students has been done e.g. by B. Wilkinson in book *Grid Computing: Techniques and Applications* [8]. All of them deal with middleware Globus. In our case, we did experiments with middleware Condor, at the beginning.

Very specific field of grid computing, which is interesting for limited group of students (researchers), has found some support by e-learning or MOOCs [6][7]. In case of the course at the University of Ss. Cyril and Methodius (UCM), the course is aimed at the technology of virtualization, preparation, and administration of computer cluster to join it into the grid infrastructure [6].

3. COURSE DESCRIPTION
Our Grid computing course is offered to students as an optional course in a spring term of the 3rd year of their bachelor's study. The prerequisite is optional course Grid technology, which has got the same structure as the course at UCM [6]. Students should have the knowledge of computer networks, parallel computer architectures, programming in C, distributed operating systems and basic experience with UNIX as an operating system.

The Grid Computing course consists of 10 lectures (40 minutes per week) and 10 laboratory sessions (40 minutes per week). The syllabus expects self-study and self-experiments, too. The course provides a practically oriented introduction to current technologies, which is the same as in cited works [3][4].

An e-learning course was developed as a source of basic study materials. In comparison with the cited papers [3][4][8][10][10], our course has been focused on four basic areas: 1) The grid infrastructure and virtual organizations, 2) Safety in the grid, 3) Using the grid through a command line, and 4) Using a Scheduler to Submit a Job.

3.1 Labs
Students in the labs have to solve a couple of problems. First of all, they generate a pair of keys to login on the grid user interface and then they generate a private key and a certificate request to issue a grid certificate by certification authority. The virtual Linux labs server is used for these activities. The pair of keys is generated by PuttyKeyGen, too. Students have got the possibility to try both, Windows and Linux platforms.

Following tasks are designed to gain skills in using of the grid infrastructure by middleware gLite. They learn to create a job submission file, submit the job to the gLite resource allocation and agreement. Experiments are made with sequential programs and parallel programs written in C language. The sequential programs include three cases: with implicit input/output files (standard input/output), with explicit input/output files and with parametric input. In case of parallel programs, the students adapt parallel program for matrix multiplication for the grid environment using MPI framework (usually copy and paste from free examples). A special problem are DAG (Direct Acyclic Graph) jobs, because it is hard to find a suitable testbed. We were able to test it only once in virtual organization VOCE (in case of preparation of a bachelor thesis).

3.2 Testbed
Implementation of experiments by students was a problem in the beginning, because the grid infrastructure of SlovakGrid could be

used only for research purposes. We could use the virtual organization VOCE, but it would be illegal and unfair. Teachers used screenshots of a real response on a user interface, when the VOCE was used to demonstrate real use cases. Testbed GILDA was not suitable for students due to the end of its services and reliability.

Institute of Informatics of Slovak Academy of Sciences has helped us. It has provided a testbed for our students to run experiments (all except DAG). The test grid is primarily designed to test updates of the middleware gLite. It consists of three servers, one is hosting all necessary grid services (User Interface, Top level BDII, Workload Management System and Logging & Bookkeeping service, MyProxy, Logical File Catalogue, as well as testing Grid site: Site BDII, Storage Element and Computing Element with one Worker Node) inside virtual machines and two other servers were added as Worker Nodes for testing MPI parallel applications. Current hardware configuration of primary server is: IBM System x3250 M5, Intel Xeon CPU E3-1241 v3 @ 3.50GHz, 8GB RAM and 1TB disk. Two additional Worker Nodes are based on older PC servers with hardware configuration: Intel Core2 Quad CPU Q8300 @ 2.50GHz, 4GB RAM and 750GB disk. X509 testing certificates are issued using dedicated TestGrid Certification Authority (based on openssl ca) which is located on primary server. Thirty accounts have been created for students. Students use their pair of keys to login. Virtual organization TESTVO is available to simulate a real virtual organization.

4. RESULTS

The number of participants in the course changes year by year. It depends on study programs accreditation (years 2007, 2008), interest in study program and university, quality of students, etc. The number of course participants is showed in Table 1. The information system contains records about distance students since 2012, therefore the cells corresponding to years before 2012 contain question marks. An updated course Grid and cloud computing has been moved to master study since 2016 and the original course Grid computing is going to be finished in year 2018 (two rows in the cases A and D, years 2016 and 2017) within the ERASMUS schema.

Table 1. Number of participating students

Year	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
A	1	17	36	30	25	14	52	35	25	39	27 / 37	25 / 31
B	?	?	?	?	?	?	27	13	17	8	8	8
C	21	22	12	5	7	8	8	5	4	6	8	2
D	n	1	1	12	4	n	17	6	7	15	10 / *31*	14 / *30*
E	?	?	?	?	?	?	n	n	11	0	n	5
F	21	6	1	0	1	0	0	0	0	1	0	0

Number of students: A – applied informatics (full-time study), B – applied informatics (distance study), C – teacher students, D – participating course cat. A, E – participating course cat. B, F – participating course cat. C. n – course not offered.

The course was offered for university researchers a couple times too and also for students at University of Applied Sciences in Oulu (Finland) and at Liepaja University (Latvia).

5. CONSLUSION

At this time the grid technologies have been included into category PaaS of cloud computing services. Therefore, both subjects Grid technologies and Grid computing have been included in new, more general courses: Grid and cloud technologies and Grid and cloud computing, to reflect current trends in this area, because the current form of grid computing has got actually two ways: cluster computing (quasi-return) and cloud computing in research cloud infrastructure.

REFERENCES

[1] APON, A., BAKER, M. Teaching Condor Grid Computing to Beginning Programming Students. In: *EEE Distributed Systems Online*, 2007. vol. 8, no. 4, art. no. 0704-o4002.

[2] Ian FOSTER. I., KESSELMAN, C. (Eds.). *The Grid: Blueprint for a New Computing Infrastructure*, 1998. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. ISBN 1-55860-475-8.

[3] PETCU, D. Teaching Grid Technologies to PhD Students, Part 1: Using Best Practices to Build the Course, 2008. In: *IEEE Distributed Systems Online*. vol. 9, no. 3, 2008, art. no. 0803-o3002.

[4] PETCU, D. Teaching Grid Technologies to PhD Students, Part 2: Course Structure and Experiences. 2008, In: *IEEE Distributed Systems Online*, vol. 9, no. 4, 2008, art. no. 0804-o4001.

[5] ŠIMON, M., HURAJ, L., HOSŤOVECKÝ, M.: IPv6 Network DDoS Attack with P2P Grid. In: *Creativity in Intelligent, Technologies and Data Science*. Springer International Publishing, 2015. pp. 407-415.

[6] ŠKRINÁROVÁ, J., VESEL, E. Model of education and training strategy for the high performance computing. In *ICETA 2016*. 14th IEEE international conference on emerging eLearning technologies and applications, November 24-26, 2016, Starý Smokovec, Slovakia [elektronický zdroj]. 1. vyd. - [s. l.] : IEEE, 2016. ISBN 978-1-5059-4701-7, cD-ROM, s. 315-320. Also available at: http://ieeexplore.ieee.org/document/7802080/

[7] VESEL, E., ŠKRINÁROVÁ, J. Teaching effectiveness of high-performance computing using an online course. In *International education and research journal.* Ahmedabad, Gujarat : Adhiyamaan college of engineering, 2016. vol. 2, no. 7, pp. 96-97. ISSN 2454-9916.

[8] WILKINSON, B. *Grid Computing: Techniques and Applications* (1st ed.), 2009.Chapman & Hall/CRC. ISBN 1420069535.

[9] WILKINSON, B., FERNER, C. Towards a top-down approach to teaching an undergraduate grid computing course. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education (SIGCSE '08)*, 2008. ACM, New York, NY, USA, 126-130. DOI=http://dx.doi.org/10.1145/1352135.1352179

[10] WILKINSON, B., FERNER, C. Teaching Grid Computing in North Carolina: Part I. In: *IEEE Distributed Systems Online*, 2006. vol. 7, pp. 3. ISSN: 1541-4922.

[11] WILKINSON, B., FERNER, C. Teaching Grid Computing in North Carolina: Part II. In: *IEEE Distributed Systems Online*, 2006. vol. 7, no. 7, 2006, art. no. 0607-o7003.

DIDINFO & *DIDACTIG 2017*. **Univerzita Mateja Bela, Banská Bystrica 2017.**

Výpočtové myslenie
Computational thinking

Jarmila Škrinárová
Fakulta prírodných vied UMB v Banskej Bystrici
Tajovského 40
97401 Banská Bystrica
Slovensko
jarmila.skrinarova@umb.sk

Nika Klimová
Fakulta prírodných vied UMB v Banskej Bystrici
Tajovského 40
97401 Banská Bystrica
Slovensko
nika.klimova2@gmail.com

ABSTRACT
In this article we deal with computational thinking as the contribution of informatics in everyday human life. Computational thinking is a process of solving problems (tasks), which consists of methods of problem-solving and intellectual practices. The importance of developing computational thinking is demonstrated with a simple example. An example is designed for pupils primary school.

Keywords
Computational thinking. Algorithmic thinking. Problem-solving methods. Intellectual processes.

ABSTRAKT
V tomto článku sa zaoberáme výpočtovým myslením ako prínosom informatiky do každodenného živote človeka. Výpočtové myslenie je proces riešenia problémov (úloh), ktorý pozostáva z metód riešenia úloh a intelektuálnych postupov. Dôležitosť rozvíjania výpočtového myslenia demonštrujeme na jednoduchom príklade. Príklad je určený pre deti na základných školách.

Kľúčové slová
Výpočtové myslenie. Algoritmické myslenie. Metódy riešenia problémov. Intelektuálne postupy.

1. ÚVOD
Inteligencia je schopnosť organizmu prispôsobovať sa. Jestvuje však viacero typov inteligencie. Do istej miery spolu korelujú, avšak človek s vyvinutým jedným typom inteligencie ešte nemusí vynikať v druhom. Inteligencia je vrodená, ale najmä sa rozvíja alebo utlmuje vplyvom prostredia [1].

Každý má iné schopnosti potrebné na osvojenie si učiva, analyzovanie a formulovanie poznatkov, zovšeobecnenie vlastností a hľadanie riešenia problémov,. Štýly učenia sa úzko súvisia s typom inteligencie. Moderný učiteľ pozná typy inteligencie svojich študentov a určitý druh inteligencie vie primerane rozvíjať. Trénovanie výpočtového myslenia prispieva k rozvoju výpočtovej inteligencie.

2. VÝPOČTOVÉ MYSLENIE
Prínos informatiky do každodenného života ľudí nespočíva len v digitálnej gramotnosti, ale predovšetkým vo výpočtovom myslení. Prispieva k celkovému rozvoju osobnosti človeka. Pojem výpočtové myslenie predstavila Jeannette Wingová [2]. Výpočtové myslenie je proces riešenia problémov, ktorý obsahuje:

- metódy riešenia problémov,
- všeobecné intelektuálne postupy.

K metódam riešenia problémov patrí:

- Reprezentácia informácií prostredníctvom abstrakcií, akými sú modely a simulácie.
- Logické štruktúrovanie a analýza údajov.
- Automatizácia riešenia problémov prostredníctvom algoritmického myslenia. Algoritmické myslenie obsahuje presne opísané sekvencie krokov z presne vymedzeného zoznamu základných operácií.
- Identifikácia, analýza a implementácia možných riešení s cieľom dosiahnuť najúčinnejšiu kombináciu krokov a prostriedkov (ľudských a/alebo technických).
- Formulácia problémov (úloh) takým spôsobom, ktorý uľahčuje používanie počítača a informačných technológií, ktoré ich pomáhajú riešiť.
- Proces zovšeobecnenia riešenia problému pre širokú škálu problémov.

Tieto techniky sú dôležité pre všetkých ľudí, nielen pre ich priamu prácu s počítačmi, sieťami a softvérom, ale ako nástroj na riešenie rôznych problémov v mnohých disciplínach.

Súčasťou všeobecných intelektuálnych postupov je:

- Poznanie zložitosti (komplexnosti) problému. Napr. softvérové systémy dosahujú určitý stupeň časovej alebo pamäťovej zložitosti. Inak povedané treba poznať koľko krokov vedie k zabezpečeniu riešenia problému.
- Stálosť, resp. stabilita pri práci so zložitými problémami.
- Tolerancia k nejednoznačnosti. Treba zosúladiť riešenie problému s povolenými nepresnosťami.
- Schopnosť vyrovnať sa s otvorenými problémami.
- Schopnosť vyrovnať sa s kombináciou ľudských požiadaviek a technických aspektov.
- Schopnosť komunikovať a spolupracovať s ostatnými na dosiahnutí spoločného cieľa alebo riešenia.

3. PREDMET INFORMATIKA NA ZÁKLADNÝCH A STREDNÝCH ŠKOLÁCH
Metódy riešenia problémov a intelektuálne postupy majú svoje prirodzené miesto v modernom vzdelávaní. Metódy riešenia problémov aj súčasťou slovenského Štátneho vzdelávacieho programu pre primárne vzdelávanie v predmete Informatika. Informatika ako predmet v učebnom pláne [3]:

• Podporuje kreativitu tým, že ukazuje rôzne spôsoby a metódy riešenia problémov.

• Aplikuje konštruktívne metódy, ako napr. navrhovanie algoritmov. Poznanie a najmä vytváranie algoritmov je práca, ktorá produkuje viditeľné alebo virtuálne objekty.

• Pomocou riešenia informatických problémov pomáha riešiť zložité problémy v iných oblastiach.

• Vychováva k presnosti a k hľadaniu presných riešení. Tvorba dobrých počítačových programov si vyžaduje presnosť v každom detaile.

Úloha informatiky v primárnom a sekundárnom vzdelávaní, rovnako ako rola matematiky, je dvojaká: praktická a vzdelávacia.

Praktická úloha informatiky je zabezpečiť zručnosť pre študentov, aby sa uplatnili na informatických pozíciách v 21. storočí. Vzdelávacia úloha informatiky predstavuje neoceniteľný intelektuálny nástroj pre vývoj základnej koncepčnej zručnosti, ktorá bude slúžiť študentom počas ich kariéry vo všetkých oblastiach práce.

4. ÚLOHA ZAMERANÁ NA VÝPOČTOVÉ MYSLENIE

Cielene rozvíjať výpočtové myslenie môžeme už základnej škole. Teraz si ukážeme príklad jednoduchej úlohy, pri ktorej môžeme, ale nemusíme používať počítač.

Úloha: Janko ide na výlet. Rozhodol sa, že pôjde na turistiku. Počasie môže byť premenlivé, preto si berie aj ďalšie oblečenie a pršiplášť. Plánuje, že keď sa vráti z turistiky do hotela, pôjde do sauny a potom si oddýchne v izbe. Na druhý deň pôjde do Tatralandie, kde si chce zaplávať a večer sa chystá na párty. Pomôžeš mu zbaliť batoh v správnom poradí? Správne poradie je také, v ktorom bude Janko jednotlivé veci potrebovať. Pozri obrázok 1.

Obrázok 1. Úloha s batohom - začiatok

Postup úlohy: Zober farebný obdĺžnik s vecami, ktoré patria do batoha a umiestni ho na obrázok batoha. Pozri obrázok 2.

Keď máš umiestnené všetky veci v batohu, môžeš odkryť správne riešenie, tak, že presunieš čierny obdĺžnik s názvom riešenie na iné miesto. Pozri obrázok 3. V prípade, že nie všetky deti umiestnili veci do batoha správne, nasleduje krátka diskusia.

Obrázok 2. Úloha s batohom – priebeh riešenia úlohy

Obrázok 3. Úloha s batohom – správne riešenie

Po krátkej diskusii deti hľadajú ďalšie úlohy. Napríklad: v škole pripravujeme podujatie, na ktorom budú vystupovať aj deti. Navrhneme niekoľko aktivít podujatia. Deti majú znova určiť správne poradie aktivít.

BIBLIOGRAFICKÉ ODKAZY

[1] ŠKRINÁROVÁ, J.. *Metodika dištančného vzdelávania a e-learningu*. Banská Bystrica: Univerzita Mateja Bela, 2011.

[2] WING, J. Computational Thinking, In *Communications of the ACM*, vol. 49, no. 3, 2006, s. 33 – 35

[3] GANDER, W. et al. *Informatics education: Europe cannot afford to miss the boat. Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education April 2013*. 21 s. http://www.informatics-europe.org/images/documents/informatics-education-acm-ie.pdf

DIDINFO & *DIDACTIG 2017*. **Univerzita Mateja Bela, Banská Bystrica 2017.**

Podpora mladých talentov ako kľúčový prvok rozvoja vysokovýkonného počítania

Young talents supporting as a key element of HPC development

Jarmila Škrinárová
Fakulta prírodných vied, Univerzita
Mateja Bela v Banskej Bystrici
Tajovského 40
97401 Banská Bystrica
Slovensko
jarmila.skrinarova@umb.sk

ABSTRACT

In this paper we describe the High Performance Computing at Matej Bel University in Banská Bystrica. High Performance Computing at UMB covers the High Performance Computing Center (HPCC UMB). The paper specifies an example of young talents supporting as a key element of European HPC development.

Keywords

High Performance Computing. Computer cluster. Slovak Infrastructure for High Performance Computing. Talent supporting.

ABSTRAKT

V tomto článku opisujeme vysokovýkonné počítanie (HPC) na Univerzite Mateja Bela v Banskej Bystrici. Vysokovýkonné počítanie (HPC) na UMB zastrešuje Centrum vysokovýkonného počítania (HPCC UMB). V článku špecifikujeme príklad podpory mladých talentov, ako kľúčový prvok rozvoja európskej HPC.

Kľúčové slová

Vysokovýkonné počítanie. Počítačový klaster. Slovenská infraštruktúra pre vysokovýkonné počítanie. Podpora talentov.

1. ÚVOD

Cieľom Programu stratégie pre vedu (Strategic Research Agenda, SRA, http://www.etp4hpc.eu/en/sra.html) a Európskeho výskumného programu (European Research agenda, ERA, http://ec.europa.eu/research/era/index_en.htm): zvýšiť efektívnosť národných výskumných systémov, zlepšiť nadnárodnú spoluprácu, súťaže a efektívne pracujúce zariadenia, ako systém kľúčovej výskumnej infraštruktúry, optimalizovať obeh a prenos vedeckých poznatkov, aj prostredníctvom digitálnych prostriedkov. Na európskej úrovni, predstavitelia projektov medzi EGI-Engage a PRACE-4IP, podpísali dohodu s cieľom vzájomne koordinovať implementácie digitálnych technológií, ktoré urýchľujú výskum a inovácie, vedú k pevnejšej európskej elektronickej infraštruktúre a tým lepšie podporujú ERA, v súlade s Európskou technologickou platformou pre HPC „ETP4HPC Strategic Research Agenda (SRA)“. Cieľ Európskej iniciatívy pre Extrémne údaje a výpočty (European Extreme Data & Computing Initiative, EXDCI, pozri https://exdci.eu/) je podporiť vývoj a realizáciu spoločnej stratégie pre Európsky HPC ekosystém. Jeden zo 4 základných pilierov tvorí: **podpora mladých talentov, ako kľúčový prvok rozvoja európskej HPC.**

UMB je prostredníctvom Centra pre vysokovýkonné počítanie a na báze národného projektu Slovenská infraštruktúra pre vysokovýkonné počítanie, zapojená do Slovenskej a Európskej gridovej infraštruktúry (projekt European Grid Infrastructure, EGI). V rámci projektu boli vybudované špecializované strediská disponujúce modernou vysokovýkonnou výpočtovou technikou, superpočítačmi a vysokokapacitnými úložiskami dát na UMB a vo vybraných pracoviskách, prevažne na technických univerzitách (STU v Bratislave, ŽU v Žiline a TU v Košiciach) a SAV. Hlavným cieľom pracoviska HPCC UMB je efektívne prevádzkovať vysokovýkonný počítačový klaster a poskytovať služby vedcom, ktorí vo svojich výskumoch potrebujú realizovať výpočty na vysokovýkonnej výpočtovej infraštruktúre. Zapojenie UMB do európskej gridovej infraštruktúry ponúka používateľom ďalšie výpočtové možnosti a prístup k ďalším výpočtovým a softvérovým zdrojom.

2. VYSOKOVÝKONNÝ POČÍTAČOVÝ KLASTER

V rámci projektu SIVVP prevádzkuje UMB (obr. 1), ktorý pozostáva zo 41 výpočtových uzlov IBM dx360M3 a IBM dx360M4. Dva výpočtové uzly sú osadené grafickými kartami GPGPU NVIDIA Tesla M2070 a 3 uzly sú osadené dvojicami GPGPU NVIDIA Tesla K20. Ide o špeciálne grafické karty, ktoré slúžia na paralelné výpočty. Klaster má celkovo 560 fyzických výpočtových procesorových jadier a 15872 (grafických) paralelných výpočtových jadier. Ďalšie uzly slúžia na riadenie a zabezpečenie prevádzky systému.

Obrázok 1. Vysokovákonný klaster UMB

V roku 2014 prebehol proces pripájania vysokovýkonného klastra UMB do Európskej gridovej infraštruktúry (angl. European grid infrastructure, EGI). V tejto súvislosti je potrebné udržiavať predpísanú funkcionalitu a striktne dodržiavať bezpečnostné politiky. Pripojenie klastra do EGI, na strane nášho klastra, vyžaduje udržiavanie 5 virtuálnych serverov. Okrem virtuálnych serverov je pre používateľov gridu vyhradených 8TB z kapacity diskového poľa a 5 výpočtových uzlov. UMB poskytuje EGI malú infraštruktúru (CPU: 5 x 12 = 60 jadier, RAM: 5 x 48GB = 240GB RAM a HDD: 8TB) a k dispozícii na naše výpočty máme celú európsku gridovú infraštruktúru. Klaster UMB je monitorovaný aj systémami SlovakGrid, t.j. národnej gridovej infraštruktúry NGI a EGI (obrázok 2).

Obrázok 2. Členstvo UMB BB v EGI council.

Zapojením klastra do Európskej gridovej infraštruktúry sme sa v roku 2016 stali členmi a participantami v EGI Council a EGI Federation dátových centier, ktoré sú lokalizované najmä v Európe, ale aj integrované vo výpočtových zdrojoch Kanade, USA, Latinskej Amerike a severnej Afrike. Kliknutím na odkaz UMB – BB – Slovakia sa otvorí stránka nášho HPCC, http://www.hpcc.umb.sk/.

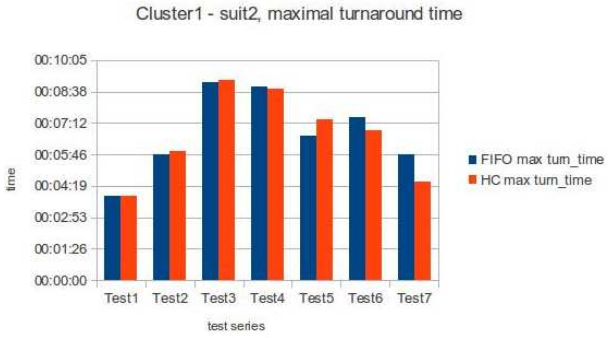
3. PODPORA MLADÝCH TALENTOV AKO KĽÚČOVÝ PRVOK HPC

Vzdelávanie a výchova novej generácie vedcov, pre a používateľov HPC infraštruktúry je neodmysliteľnou súčasťou spolupráce HPCC a Katedry informatiky FPV UMB. V súčasnosti realizujeme výučbu v predmetoch študijného programu Aplikovaná informatika (AI), ktoré úzko súvisia s využívaním vysokovýkonnej výpočtovej ako aj predmety zamerané na ich inštaláciu a administráciu, čo prispieva k lepšiemu pochopeniu problematiky vysokovýkonného počítania. V rámci prípravy študentov Katedra informatiky v roku 2016 ponúkala nový predmet, analýza mnohorozmerných údajov, ktorý využíva štatistické metódy a IT technológie, vhodné na spracovanie a dolovanie informácií z údajov (resp. tokov údajov), v súlade s najnovšími svetovými trendmi. Aktuálne prvýkrát vyučujeme predmet manažovanie vysokovýkonných systémov. Predmety 1. stupňa študijného programu Aplikovaná informatika sú programovanie, operačné systémy, distribuované operačné systémy a webové, gridové a cloudové technológie. V 2. stupni sú to najmä predmety paralelné a distribuované výpočty, analýza mnohorozmerných údajov a manažovanie vysokovýkonných systémov. V roku 2016 sme pripravili e-learningový kurz „Spúšťanie úloh na vysokovýkonnom klastri“, ktorý je dostupný v LMS Moodle.

Mladí vedci sa už počas svojho štúdia zapájajú do výskumných projektov školiteľov a riešia výskumné úlohy nielen, ale aj v rámci svojich bakalárskych, diplomových a dizertačných prác.

Práca s talentami je radosťou pre každého učiteľa. Spolupráca, tvorivá atmosféra a vzájomná dôvera pri riešení výskumných úloh so študentami motivuje všetkých zúčastnených a vedie k vzájomnej úcte učiteľa a študenta a súčasne k lepším výsledkom. Za všetky príklady uvedieme jeden viac ako dobrý príklad:

Mgr. M. Krnáč už na bakalárskom štúdiu pripravil simulátor plánovania úloh vo vysokovýkonnom počítaní. Simulátor bol upravený tak, aby sa dal využívať aj vzdialene pomocou webového rozhrania a stal sa súčasťou vzdelávacieho kurzu [1]. M. Krnáč sa umiestnil v roku 2011 na prvom mieste fakultnej ŠVOČ. Výskumná práca ďalej pokračovala a M. Krnáč vytvoril model určený na modelovanie a simuláciu plánovacích algoritmov na báze virtuálnych strojov. Model používal manažovací systém TORQUE, ktorý umožňoval implementovať vlastné plánovacie algoritmy [2]. Neskôr, v praktickej časti diplomovej práce [3] navrhol plánovacie algoritmy, na báze optimalizácie tzv. horolezeckého algoritmu (pozri obr. 3) alebo optimalizácie založenej na princípe sociálnych javov správania sa roja častíc. Algoritmus implementoval do manažovacieho systému vysokovýkonného klastra UMB [4] a navrhol metodiku tvorby, implementácia a testovania plánovacích algoritmov v prostredí manažovacieho systému TORQUE. S diplomovou prácou sa Mgr. Krnáč umiestnil vo finále na 6. mieste medzinárodnej súťaže ACM SPY (Student Project of the Year) a na 1. mieste fakultnej ŠVOČ.



Obrázok 3. Príklad výsledkov testovania algoritmu.

BIBLIOGRAFICKÉ ODKAZY

[1] ŠKRINÁROVÁ, J., KRNÁČ, M. E-learning course for scheduling of computer grid, In *Proceedinds of International Conference Interactive Collaborative Learning ICL 2011*. Wien : International Associaton of Online Engineering, 2011. IEEE Computer Society, p. 352-356.ISBN 978-1-4577-1746-8.

[2] ŠKRINÁROVÁ, J., KRNÁČ, M. Particle Swarm Optimization Model for Grid Scheduling, In *Proceedings of: Second International Conference on Computer Modelling and Simulation*, CSSIM 2011, Brno, Czech Republic, September 5-7, Brno University of Technology, 146-153

[3] KRNÁČ, M. *Algoritmy vyvažovania na gridovom prostredí* : diplomová práca. Banská Bystrica : UMB, 2013. 84 s.

[4] ŠKRINÁROVÁ, J., KRNÁČ, M. Particle Swarm Optimization for Grid Scheduling, In *Proceedings of: Informatics 2011*, Eleventh International Conference on Informatics, Roznava, Faculty of Electrical Engineering and Informatics of the Technical University of Kosice, 2011, 153-158.

Vývoj mobilných aplikácií na Katedre informatiky FPV UMB

Mobile Application Development at the Department of Informatics FNS MBU

Patrik Voštinár
Katedra matematiky FPV UMB
Tajovského 40
974 01 Banská Bystrica
Slovenská republika
patrik.vostinar@gmail.com

ABSTRACT

The main aim of the contribution is to show newly-created subject Web Technologies 4, which is being taught at the Department of Informatics of FNS MBU from the academic year 2016/2017. The course deals with the development of mobile applications for the Android operating system.

Keywords

FPV UMB. Android. e-learning.

ABSTRAKT

Cieľom príspevku je priblížiť novo-vytvorený predmet Webové technológie 4, ktorý sa začal vyučovať na Katedre informatiky na FPV UMB v akademickom roku 2016/2017. Predmet sa zaoberá vývojom mobilných aplikácií pre operačný systém Android.

Kľúčové slová

FPV UMB. Android. e-learning.

1. ÚVOD

Vyučovanie informatiky má na Univerzite Mateja Bela už dlhú tradíciu. Prvýkrát sa tu informatika začala vyučovať 1.9.1989. Za toto pomerne dlhé obdobie sa na katedre vyučovalo veľa rôznych predmetov. Tento školský rok 2016/2017 pribudol na katedre predmet s názvom Webové technológie 4. Cieľom tohto predmetu je priblížiť študentom vývoj mobilných aplikácií. Žijeme v dobe, kde mobilné zariadenia patria k životnému štýlu hlavne mladých ľudí. Katedra informatiky chce nasledovať tento trend a ponúknuť svojim študentom pohľad na vývoj mobilných aplikácií, preto sa rozhodla začleniť tento predmet do výučby.

2. PREDMET WEBOVÉ TECHNOLÓGIE 4

Hlavným cieľom tohto predmetu je, aby študenti získali a pochopili podstatné fakty, pojmy, princípy a teórie vzťahujúce sa k tematike tvorby mobilných aplikácií.

Tento predmet si môžu vybrať študenti prvého magisterského ročníka, ako jeden z povinne voliteľných predmetov, za ktorý získajú 4 kredity. Prihlásiť sa môžu študenti dennej a externej formy odboru Aplikovaná informatika.

Študenti majú k dispozícii e-learningový kurz, ktorý sa nachádza v systéme LMS Moodle[1]. V tomto kurze sa nachádzajú e-lekcie, ktoré prevedú študenta od základov, až po publikovanie aplikácií v špecializovanom mobilnom obchode Google Play.

V tomto e-learningovom kurze učiteľ oboznamuje žiakov s procesom vývoja aplikácií pre operačný systém Android, zatiaľ čo študenti si v dvojiciach naštudujú súčasný stav problematiky mobilných vývojových platforiem a následne ich odprezentujú svojim spolužiakom. Študenti majú možnosť vybrať si tému z nasledujúcich vývojových platforiem: HTML 5, Windows phone, iOS, BlackBerry, PhoneGap, jQuery mobile, Rhodes, Mono, Adobe AIR, Titanium Mobile, Firemonkey a NativeScript.

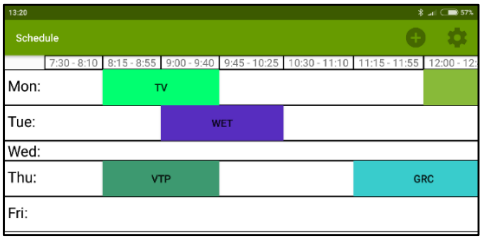
2.1 Záverečné študentské mobilné aplikácie

Výučba predmetu Webové technológie 4 prebiehala akademický rok 2016/2017 v zimnom semestri. Počas tohto semestra študenti prvého ročníka magisterského denného štúdia naprogramovali viacero mobilných aplikácií, napríklad:

- TaskApp - zoznam úloh, ktoré musí prihlásený používateľ urobiť v určitom časovom intervale,
- MX File Manager – aplikácia na spravovanie súborov mobilného zariadenia,
- BB News – aplikácia, ktorá spája novinky zo spravodajských portálov o Banskej Bystrici,
- Inteligentný alarm – upozorňuje používateľov na nastavený čas, priblíženie sa k určitému miestu a ku pridaným kontaktom,
- Rozvrh – pomocou tejto aplikácie si študenti, alebo učitelia všetkých druhov škôl, dokážu efektívne vytvoriť rozvrh podľa svojich požiadaviek.

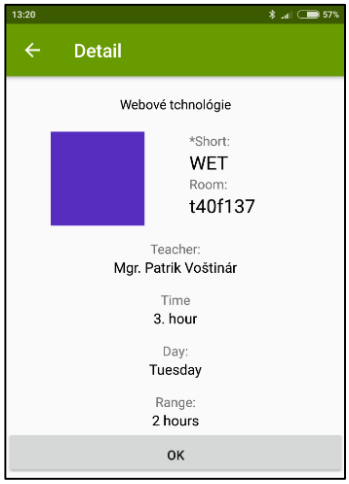
Hlavná obrazovka poslednej spomínanej aplikácie je zobrazená na obrázku 1. Používateľ aplikácie si pridal päť predmetov do svojho rozvrhu.

[1] https://lms2.umb.sk/projekty/course/view.php?id=60



Obrázok 1. Hlavná obrazovka aplikácie Rozvrh

Po kliknutí na konkrétny predmet sa zobrazí dialógové okno s informáciami o danom predmete – skratka, miestnosť, vyučujúci, čas a deň (pozri obrázok 2).



Obrázok 2. Okno s detailom konkrétneho predmetu

Aplikácia Rozvrh, okrem hore uvedených obrazoviek, obsahuje ďalšiu funkcionalitu a iné obrazovky:

- podpora dvoch jazykov,
- ukladanie informácií do databázy SQLite.
- vyplnenie základných informácií po prvom spustení - názov rozvrhu, dĺžka vyučovacej hodiny, prestávky, začiatok výučby,
- pridávanie hodín do databázy,
- odstránenie predmetu,
- zobrazenie detailu po kliknutí.

3. ZÁVER

Príspevok sa zaoberal opisom novo-vytvoreného predmetu Webové technológie 4, ktorý priblížil študentom prvého magisterského ročníka odboru Aplikovaná informatika proces vývoja mobilných aplikácií. Študenti získali po absolvovaní predmetu prehľad aktuálnych trendov vo vývoji mobilných aplikácií a vytvorili si svoje vlastné mobilné aplikácie.

4. BIBLIOGRAFICKÉ ODKAZY

[1] Informačné listy predmetov (kritérium KSP-B2), študijný program: Aplikovaná informatika.

http://www.fpv.umb.sk/studium/akreditovane-studijne-programy/magisterske-studium-2015.html

[2] VOŠTINÁR, P. *Webové technológie 4*. [online]. Banská Bystrica : UMB, [vid. 2017-03-01].

http://lms2.umb.sk/projekty/course/view.php?id=60

Zvyšovanie profesionality pedagogických zamestnancov v oblasti didaktických digitálnych kompetencií

Development of Teachers professionalism in the area of digital didactic competences

Darina Výbohová Vladimír Laššák Jozef Ďurian

Metodicko-pedagogické centrum
Ševčenkova 11
850 05 Bratislava
Slovensko

darina.vybohova@mpc-edu.sk vladimir.lassak@mpc-edu-sk jozef.durian@mpc-edu.sk

ABSTRACT

The poster describes experience of Metodicko-pedagogicke centrum (Institute for In-service Teacher´s Education and Training) to be related to increasing of professional competences of teachers through training activities in the area of didactic digital competences. It also indicates future orientation of in-service training of teachers in didactic digital competences.

Keywords

Pedagogy. Didactics. Digital technologies. Professional Standard. Competence framework. Didactic Digital Competence. Teachers´ Development.

ABSTRAKT

Príspevok opisuje skúsenosti Metodicko-pedagogického centra so vzdelávacími aktivitami zameranými na zvyšovanie profesijných kompetencií pedagogických zamestnancov v oblasti digitálnych kompetencií. Uvádza východiská pre vzdelávanie pedagogických zamestnancov v oblasti digitálnych kompetencií v budúcnosti.

Kľúčové slová

Pedagogika. Didaktika. Digitálne technológie. Didaktické digitálne kompetencie. Vzdelávací program.

1. DOTERAJŠÍ ROZVOJ DIGITÁLNYCH KOMPETENCIÍ UČITEĽOV

Metodicko-pedagogické centrum (ďalej MPC) riešilo v rokoch 2007-2015 v súčinnosti s MŠVVaŠ SR viacero projektov zameraných na rozvoj kompetencií pedagogických a odborných zamestnancov v oblastiach využívania informačných a komunikačných technológií v školách.

Cieľom národného projektu Profesijný a kariérový rast pedagogických a odborných zamestnancov (ďalej PKR) bolo vytvoriť a zdokonaľovať efektívny systém kontinuálneho vzdelávania pre PZ/OZ. Na rozvíjanie digitálnych kompetencií a podporu IKT bolo zameraných 85 akreditovaných vzdelávacích programov (ďalej AVP), ktoré ukončilo 31.240 absolventov. Z toho 5 programov bolo zameraných na informatiku a informatickú výchovu od primárneho, až po vyšší sekundárny stupeň vzdelávania.

V rámci národného projektu Vzdelávanie pedagogických zamestnancov MŠ ako súčasť celoživotného vzdelávania MPC dodalo do 4.400 MŠ informačno-komunikačné technológie a realizovalo vzdelávania v 7 AVP zameraných na rozvoj digitálnych kompetencii pedagogických zamestnancov pre predprimárne vzdelávanie.

Do národného projektu Aktivizujúce metódy vo výchove sa celkom zapojilo 6.183 účastníkov. Jeho cieľom bolo zvýšiť úroveň mediálnej gramotnosti a zdokonaliť zručnosti PZ /OZ vo využívaní aktivizujúcich metód vo výchovnej činnosti s využitím interaktívnych digitálnych pomôcok (IDP) obsahovo zameraných na oblasť dopravnej, etickej, mediálnej a zdravotnej výchovy. Z projektu bolo dodaných 1.550 interaktívnych tabúľ s príslušenstvom a Edukačný balíček s celkom 132 edukačnými videoprogramami, 13 softvérovými riešeniami interaktívnych prvkov a 264 predvolenými, odborne garantovanými náplňami. Vzdelávanie a obsah edukačných filmov zahrňoval tiež témy súvisiace s rozvojom mediálnej gramotnosti a bezpečnosti na internete.

Zámerom národného projektu Nové trendy vzdelávania učiteľov anglického jazyka na základných školách – EnglishOne bolo v relatívne krátkom čase docieliť kvalitnejšiu výučbu anglického jazyka na základných školách kombináciou vzdelávania a dodaním moderných didaktických digitálnych prostriedkov, vďaka ktorým sú PZ schopní učiť angličtinu kvalitnejšie bez ohľadu na mieru ich kvalifikácie. Vzdelávanie v projekte úspešne absolvovalo 3.000 PZ.

MPC sa podieľalo aj na národnom projekte Elektronizácia vzdelávacieho systému regionálneho školstva - Digiškola. Dodalo do škôl tablety a realizovalo ďalšie vzdelávanie PZ na používanie tabletov a manažmentu triedy.

Problematika rozvoja digitálnych didaktických kompetencií (ďalej DDK) je vo vzdelávacích aktivitách MPC integrovaná obsahovo a prierezovo do rôznych vzdelávacích a výchovných oblastí, vrátane mimoškolských aktivít (Centrá voľného času, ZUŠ, Školské kluby) a pre všetky úrovne vzdelávania (predprimárne vzdelávanie, primárne vzdelávanie, nižšie sekundárne vzdelávanie a vyššie sekundárne vzdelávanie). Vzdelávacie programy a aktivity reflektujú zistené požiadavky a potreby jednotlivých cieľových skupín. Dôraz je kladený na rozvoj zručností narábať s konkrétnymi digitálnymi pomôckami (napr. interaktívna tabuľou, tabletmi, digitálnou hračkou Bee-bot a pod.), aplikáciami

a počítačovými programami (napr. programy MS Office, AutoCAD, Baltik, Zoner a mnohé iné).

Atestácia je overenie získaných kompetencií vymedzených profesijným štandardom pre príslušnú kategóriu PZ/OZ, ktorí ich získali kontinuálnym vzdelávaním, sebavzdelávaním a výkonom pedagogickej činnosti alebo výkonom odborných činností. Na MPC v rokoch 2010 – 2016 bolo celkovo 13.810 atestovaných PZ/OZ, z toho 9.076 bolo atestovaných v rámci 1 atestácie a 4.734 v rámci 2 atestácie. Témy a oblasti, na ktoré sa uchádzači v prácach pre 1. a 2. atestáciu zamerali sú v tabuľke 1.

Tabuľka 1. Zastúpenie tém a oblastí atestácií

Zameranie odborno-metodického problému riešeného v atestačnej práci	1 AT	2 AT	Spolu
1. Obsahy, programy, softvér a učebné plány	31	26	57
2. Zručnosti pri používaní aplikácií a zariadení	45	35	80
3. Metódy, inovácie a kvalita vzdelávacieho procesu	85	86	171
Spolu	161	147	308

V 1. oblasti **Obsahy, programy, softvér a učebné plány** boli témy atestačných prác zamerané na tvorbu pracovných listov pre hry v programovacom prostredí Scratch, rozvoj algoritmického myslenia, programovanie v Imagine na 2. stupni ZŠ, programovanie s využitím robotických stavebníc, využitie programovateľnej hračky Bee-Bot, vytvorenie zbierky príkladov v Lazaruse, tvorbu panelu na simuláciu počítačovej siete, bezpečnosť detí na internete, na sociálnych sieťach a iné.

V 2. oblasti **Zručnosti pri používaní aplikácií a zariadení** boli témy atestačných prác zamerané na: výučbu s využitím programu GeoGebra, spracovanie grafických informácií, využívanie hypertextov v informatike, využívanie vektorovej grafiky vo vzdelávaní, vytváranie animácií, využitie interaktívnej tabule vo vyučovaní informatickej výchovy a iné.

V 3. oblasti **Metódy, inovácie a kvalita vzdelávacieho procesu** boli témy atestačných prác zamerané na: využitie aktivizujúcich metód na podporu informačnej gramotnosti, zvyšovanie záujmu a motivácie žiakov o predmet informatika, o počítačovú grafiku v škole a iné, tvorba úloh s využitím aktivizujúcich metód na podporu informačnej gramotnosti vo vzťahu k informačnej spoločnosti, využitie ročníkových prác a projektového vyučovania na hodinách informatiky.

2. VÝCHODISKÁ PRE ROZVOJ DIGITÁLNYCH KOMPETENCIÍ V BUDÚCNOSTI

V programovacom období 2014 - 2020 MPC pripravuje národný projekt Inovácia prípravy učiteľov a budúcich učiteľov v Centrách inovačného vzdelávania (NP CIV), ktorý bude financovaný z operačného programu Ľudské zdroje (OP ĽZ). Projekt je členený do dvoch hlavných aktivít. Prvá je zameraná na inovácie pedagogických praxí budúcich učiteľov a druhá na inovácie v oblasti kontinuálneho vzdelávania pedagogických a odborných zamestnancov. Časť vzdelávacích programov a aktivít v projekte bude zameraná na rozvoj didaktických digitálnych kompetencií učiteľov (DDK). Vzdelávacie programy budú vytvorené na základe priebežného zisťovania vzdelávacích potrieb PZ v školách. Budú reflektovať aktuálne problémy a potreby spojené s vyučovaním informatických predmetov a využívaním moderných digitálnych didaktických prostriedkov vo vyučovacom procese v školách - napríklad nedostatočnú spoluprácu učiteľov informatických predmetov s učiteľmi iných predmetov, malé využívanie IKT v rámci medzipredmetových vzťahov a prierezových tém, pri tvorbe žiackych prác a projektov, využívanie IKT ako prostriedku aktivizácie a motivácie žiakov, rozvoja kritického myslenia, schopnosti riešiť praktické životné problémy a pod. V projekte bude vytvorený referenčný rámec DDK, ktorý vymedzí súbor očakávaných kompetencií pedagogických zamestnancov pre prácu s digitálnymi technológiami vo vyučovacom procese, napr. spôsobilosť využívať digitálne technológie na komunikáciu s kolegami, žiakmi, rodičmi a inými zainteresovanými skupinami na profesionálne výchovno-vzdelávacie účely, spôsobilosť efektívne využívať digitálne technológie na podporu výchovno-vzdelávacieho procesu v prostredí rôznorodosti žiakov, spôsobilosť personalizovať a digitalizovať výchovno-vzdelávací obsah a využívať ho v interaktívnom vzdelávaní, spôsobilosť zvyšovať záujem žiakov o učenie sa prinášaním prepojenia vedy s praxou využívaním digitálnych nástrojov a zdrojov, spôsobilosť dosahovať vzdelávacie ciele na základe kontinuálnej reflexie/spätnej väzby s využitím digitálnych technológií a vyhľadávať príležitosti na zlepšovanie svojich digitálnych kompetencií v spolupracujúcom prostredí a pod.

K referenčnému rámcu bude vytvorená metodika posudzovania úrovne rozvinutia kompetencií a certifikácie DDK, vrátane sebahodnotiaceho mechanizmu. Referenčný rámec bude vytvorený po posúdení existujúcich národných a nadnárodných štandardov a na základe zohľadnenia výsledkov zisťovania využívania digitálnych technológií a zisťovania vzdelávacích potrieb pedagogických zamestnancov. Referenčný rámec bude vytvorený v súlade s Koncepciou informatizácie a digitalizácie rezortu školstva s výhľadom do roku 2020 a Akčným plánom rozvoja digitálnej kompetencie a zručností.

Súčasťou výstupov národného projektu budú digitálne edukačné obsahy, využiteľné v rôznych vzdelávacích programoch, resp. ako obsah využiteľný v rámci neformálneho a informálneho vzdelávania učiteľov. Vzájomné učenie sa a zdieľanie praktických skúseností bude realizované prostredníctvom spracovania osvedčených praktických skúsenosti (OPS), formou odborných príspevkov spracovávaných učiteľmi z praxe. Dôraz pri spracovaní OPS bude kladený najmä na didaktické aspekty využívania moderných digitálnych technológií v školách. V rámci inovácie poradenských a konzultačných činností v projekte plánujeme zaviesť systém reflexívnej spätnej väzby. Tento systém umožní zaznamenať priebeh vyučovacích hodín a následne tento záznam analyzovať a konzultovať jedným, resp. viacerými poradcami.

Opisované východiská a zámery sa významne viažu na realizáciu spomínaného národného projektu, ktorý je momentálne vo fáze prípravy a schvaľovania na MŠVVaŠ SR. Začiatok realizácie projektu je plánovaný na začiatok školského roka 2017/2018.

BIBLIOGRAFICKÉ ODKAZY

[1] KONCEPCIA INFORMATIZÁCIE A DIGITALIZÁCIE REZORTU ŠKOLSTVA S VÝHĽADOM DO ROKU 2020. https://www.minedu.sk/koncepcia-informatizacie-a-digitalizacie-rezortu-skolstva-s-vyhladom-do-roku-2020

Workshopy

Workshops

Algoritmizácia a štruktúry – postrach alebo výborná zábava?

Martina Chalachánová, Eva Uličná

Dozviete sa ako postupne pestovať u žiakov algoritmické myslenie a o čo vlastne ide v štruktúrach. Zažijete množstvo zábavných úloh spájajúcich matematiku a informatiku. Zistíte, že informatiku nemusíte učiť len s počítačmi.

Programujeme v Pythone

Peter Kučera

Priblížime vám, ako sme prešli z vyučovania programovania v Lazaruse/Delphi na jazyk Python. Vysvetlíme vám dôvody zmeny programovacieho jazyka, ukážeme materiály a metodiku, podľa ktorej vyučujeme. Priblížime vám naše skúsenosti, odozvy študentov. Budete si môcť vyskúšať úvod do programovania v Pythone s využitím knižnice Tkinter.

Robotika na tablete

Martin Cápay

Ozobot je miniatúrny programovateľný robot s vlastnou inteligenciou založenou na náhodne generovaných rozhodnutiach. Z pohľadu učiteľa je Ozobot vynikajúca pomôcka aj k propredeutike programovania. Na workshope predstavíme úvodné aktivity, vhodné aj na prvý stupeň ZŠ.

Práce se senzorovou deskou PicoBoard ve Scratchi

Jiří Vaníček

Vyzkoušejme si senzorovou desku Picoboard (https://www.sparkfun.com/products/11888), kterou lze ovládat z programovacího prostředí Scratch, a objevme možnosti výuky programování se vstupy z reálného světa. Na workshopu budou představeny možnosti ovládání programovacího prostředí Scratch pomocí vnějších senzorů, umístěných na desce připojené k počítači. Po zvládnutí základního ovládání bude následovat několik aktivit pro úroveň žáků ZŠ, které ukážou možnosti využití při výuce.

Organizátori
Organisers

Katedra informatiky FPV UMB v Banskej Bystrici

Katedra informatiky je pevnou súčasťou Fakulty prírodných vied Univerzity Mateja Bela. Má priznané právo udeľovať titul „bakalár“ a akademický titul „magister“ absolventom dennej formy štúdia a externej formy štúdia v študijnom programe Aplikovaná informatika a aj absolventom dennej formy štúdia v študijnom programe Učiteľstvo informatiky v kombináciách, ktoré sú zriadené podľa zákonných predpisov a uskutočňované na Fakulte prírodných vied Univerzity Mateja Bela. Katedra informatiky má priznané právo udeľovať akademický titul „doktor pedagogiky“ (v skratke „PaedDr.“) a akademický titul „doktor prírodných vied“ (v skratke „RNDr.“) absolventom študijných programov, ktorí získali titul „magister“, po vykonaní rigoróznej skúšky v študijnom odbore, v ktorom získali vysokoškolské vzdelanie alebo v príbuznom študijnom odbore uskutočňovanom na Fakulte prírodných vied Univerzity Mateja Bela. Katedra informatiky vykonáva aj výučbu mnohých celofakultných predmetov a celouniverzitných predmetov, ktoré sú buď povinnými, povinne voliteľnými alebo výberovými predmetmi na katedrách FPV UMB v Banskej Bystrici, resp. na iných katedrách; aj v rámci spoločného základu.

Zameranie edukačnej, vedeckej a výskumnej činnosti Katedry informatiky FPV UMB sa odvíja od hlavnej výskumnej línie fakulty, profilácie katedry a od záujmu jednotlivých členov katedry v symbióze s potrebou obohacovať predmety v študijných programoch o najnovšie poznatky vedy a výskumu v odbore.
Hlavnými smermi činnosti katedry sú:

- Teória vyučovania informatiky.
- Softvérové inžinierstvo.
- Umelá inteligencia.
- Aplikovaná informatika.

Katedru informatiky charakterizuje:

- vysoký dopyt po absolventoch našich odborov,
- široké spektrum uplatnenia absolventov v učiteľskej i odbornej praxi,
- kontakt so školami i s firmami z oblasti IT už počas štúdia,
- viacročná tradícia v príprave učiteľov a odborníkov pre oblasť informatiky,
- ústretové výučbové prostredie,
- tím skúsených akademických a vedeckých pracovníkov katedry,
- previazanosť teoretických a praktických poznatkov,
- primerane vybavené laboratóriá,
- možnosť absolvovať časť odborného štúdia na zahraničnej partnerskej univerzite.

Katedra je medzinárodne uznávaným a akceptovaným pracoviskom. Má kontakty s rôznymi domácimi aj zahraničnými pracoviskami a organizáciami, s ktorými dochádza k pravidelnej výmene učiteľov, študentov i ďalšej spolupráci v oblasti vedy a vzdelávania: Technická univerzita Ostrava, Metropolitní univerzita a ČVUT Praha, Česká republika; Oulu University of Applied Sciences, Oulu, Fínsko; Atilim University, Ankara a Fatih University, Istambul, Turecko; Utrecht University, Utrecht, Holandsko; Liepaja University, Liepaja, Lotyšsko; Corvinus University of Budapest, Maďarsko; Śląska Wyższa Szkoła Informatyczno-Medyczna, Chorzów, Poľsko; Technische Universität Chemnitz, Nemecko a ďalšie.

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta přírodovědně-humanitní a pedagogická

Na katedře najdete lidi se dvojím odborným zaměřením. Jednak jsou zda, jak název napovídá, matematici. Ti se zaměřují především na výuku matematiky aplikované v ostatních disciplínách. Jedná se především o diskrétní matematiku, lineární algebru a statistiku. Druhou skupinou jsou informatici. Ti se podílí na dalším rozvoji ICT gramotnosti studentů bez ohledu na jejich profilaci. U budoucích učitelů se zaměřujeme na zvládnutí vybraných technologií, aby pak dále měli dobrý základ pro rozvoj počítačem podporované výuky ve svých specializacích. Pro budoucí učitele informatiky pak katedra zajišťuje část odborných předmětů, jako jsou kódování a šifrování či matematika pro informatiky. Informatická část katedry se pak soustředí v rámci přípravy těchto učitelů na didaktiku informatiky, programování a počítačem podporovanou výuku. Je to pracoviště, pod nímž jsou akreditovány všechny studijní obory připravující budoucí učitele informatiky na univerzitě – jak v bakalářském, tak navazujícím magisterském stupni. Část kolegů se specializuje na GIS a jejich využití.

Členové katedry se také zapojují do komunitního života v Jednotě školských informatiků a v Jednotě českých matematiků a fyziků. Spolupodílíme se na organizaci Bobříka informatiky a Matematické olympiády. Katedra se významně podílí i na řešení různých projektů na fakultě. Mezi nejvýznamnější projekty z poslední doby patří:

- Zvýšení kvality vzdělávání žáků, rozvoje klíčových kompetencí, oblastí vzdělávání a gramotností
- Prameny spojují krajiny a lidi
- Staré mapy Liberecka a Jizerských hor jako součást přeshraničního kulturního bohatství
- Modely extrémních hodnot založené na homogenním a nehomogenním Poissonově procesu ve studiu změny klimatu
- ICT v odborné přípravě studentů FP TUL
- Zapojení týmu KLIMATEXT do mezinárodní spolupráce

Zároveň jsme jako pracoviště zapojeni do přípravy dvou velkých projektů v rámci implementace digitální strategie – Podpora rozvoje digitální gramotnosti a Podpora rozvoje informatického myšlení. Jsou to projekty připravované napříč všemi pedagogickými fakultami v ČR, což prohloubí již tak dobrou spolupráci na domácí půdě. Tu se snažíme prohlubovat od roku 2013 pořádáním české obdoby DidInfa, která nese název DidactIG.

Zahraniční spolupráci jsme navázali například právě s Univerzitou Mateje Bela v Banské Bystrici, Univerzitou P. J. Šafárika v Košiciach, Univerzitou Komenského v Bratislave, Vilniaus universitetas nebo Carl von Ossietzky Universität Oldenburg. Bohužel vzhledem k malému počtu pracovníků informatické části neprobíhá spolupráce tak intenzivně, jak bychom chtěli. Na druhou stranu díky propojení konferencí jsme nyní v kontaktu více.

Sponzori a mediálni partneri

Sponsors and media partners

Continental
The Future in Motion

ZP
1840
Informatika
s.r.o.
člen skupiny ŽELEZIARNE PODBREZOVÁ GROUP

PMS Delta s.r.o.
Autorizované zastúpenie
Vernier

triony
technologies

PC REVUE

Quark
Magazín o vede a technike

Názov:	**DidInfo&DidactIG 2017**
Editori:	Ing. Dana Horváthová, PhD., Ing. Jana Jacková, PhD., RNDr. Alžbeta Michalíková, PhD., doc. Ing. Jarmila Škrinárová, PhD.
Vydanie:	1. vydanie
Vydavateľ:	Univerzita Mateja Bela, Fakulta prírodných vied v Banskej Bystrici, Slovensko
Rok:	2017
Rozsah:	212 strán
Formát:	elektronický zborník
Náklad:	120 ks
ISBN:	978-80-557-1216-1
ISSN:	2454-051X

Title:	**Proceedings of conference DidInfo&DidactIG 2017**
Editors:	Ing. Dana Horváthová, PhD., Ing. Jana Jacková, PhD., RNDr. Alžbeta Michalíková, PhD., doc. Ing. Jarmila Škrinárová, PhD.
Edition:	First edition
Publisher:	Matej Bel University, Faculty of Natural Sciences in Banská Bystrica, Slovakia
Year:	2017
Pages:	212
Format:	electronic proceedings
Number of copies	120
ISBN:	978-80-557-1216-1
ISSN:	2454-051X