Katedra informatiky, Fakulta prírodných vied, Univerzita Mateja Bela
Slovenská informatická spoločnosť
Metodicko-pedagogické centrum, alokované pracovisko B. Bystrica
Ústav informácií a prognóz školstva, Školské výpočtové stredisko
Banská Bystrica

DidInfo 2009

15. ročník medzinárodnej konferencie

Banská Bystrica
2009

Predseda programového výboru:
prof. RNDr. Branislav Rovan, PhD.

Programový výbor konferencie:
Doc. RNDr. Gabriela Andrejková, PhD., Univerzita Pavla Jozefa Šafárika, Slovenská republika (SK)
Prof. Ing. Mária Bieliková, PhD., Slovenská technická univerzita, Slovenská republika (SK)
Doc. RNDr. Miroslava Černochová, PhD., Univerzita Karlova, Česká republika (CZ)
Prof. RNDr. Milan Hejný, CSc., Univerzita Karlova, Česká republika (CZ)
PaedDr. Ladislav Huraj, PhD., Univerzita Mateja Bela, Slovenská republika (SK)
Dr. Piotr Chrząstowski-Wachtel, University of Warsaw, Poľsko (PL)
Prof. Dr. Mirjana Ivanovic, University of Novi Sad, Srbsko (SR)
Doc. Dr Przemyslaw Kajetanowicz, Politechnika Wrocławska, Poľsko (PL)
RNDr. Miloš Kravčík, PhD., Open Universiteit Nederland, Holandsko (NL)
Doc. RNDr. Zdena Lustigová, CSc, Univerzita Karlova, Česká republika (CZ)
RNDr. Jaroslav Pelikán, PhD., Masarykova Univerzita, Česká republika (CZ)
prof. RNDr. Branislav Rovan, PhD., Univerzita Komenského, Slovenská republika (SK)
PaedDr. Miloslava Sudolská, PhD., Univerzita Mateja Bela, Slovenská republika (SK)
Prof. Márta Turcsányi-Szabó, Ph.D., Eötvös Loránd University, Maďarsko (HU)
RNDr. Michal Winczer, PhD., Univerzita Komenského, Slovenská republika (SK)

Organizačný výbor konferencie:
PaedDr. Miloslava Sudolská, PhD.
PaedDr. Ladislav Huraj, PhD.
PaedDr. Darina Výbohová
Mgr. Stanislav Slačka

General Chair:
prof. RNDr. Branislav Rovan, PhD.

Programm committee:
Doc. RNDr. Gabriela Andrejková, PhD., Pavol Jozef Šafárik University, Slovak Republic (SK)
Prof. Ing. Mária Bieliková, PhD., Slovak University of Technology, Slovak Republic (SK)
Dr. Piotr Chrząstowski-Wachtel, University of Warsaw, Poland (PL)
Doc. RNDr. Miroslava Černochová, PhD., Charles University, Czech Republic (CZ)
Prof. RNDr. Milan Hejný, CSc., Charles University, Czech Republic (CZ)
PaedDr. Ladislav Huraj, PhD., Matej Bel University, Slovak Republic (SK)
Prof. Dr. Mirjana Ivanovic, University of Novi Sad, Serbia (SR)
Doc. Dr Przemyslaw Kajetanowicz, Politechnika Wrocławska, Poland (PL)
RNDr. Miloš Kravčík, PhD., Open Universiteit Nederland, Netherland (NL)
Doc. RNDr. Zdena Lustigová, CSc, Charles University, Czech Republic (CZ)
RNDr. Jaroslav Pelikán, PhD., Masaryk University, Czech Republic (CZ)
prof. RNDr. Branislav Rovan, PhD., Comenius University, Slovak Republic (SK)
PaedDr. Miloslava Sudolská, PhD., Matej Bel University, Slovak Republic (SK)
Prof. Márta Turcsányi-Szabó, Ph.D., Eötvös Loránd University, Hungary (HU)
RNDr. Michal Winczer, PhD., Comenius University, Slovak Republic (SK)

Organizing Committee:
PaedDr. Miloslava Sudolská, PhD.
PaedDr. Ladislav Huraj, PhD.
PaedDr. Darina Výbohová
Mgr. Stanislav Slačka

Editor: prof. RNDr. Branislav Rovan, PhD.
Technický editor: PaedDr. Ladislav Huraj, PhD.
Copyright © 2009 autori
ISBN: 978-80-8083-720-4
Recenzovaný zborník.

Predhovor

Pätnásty ročník konferencie DIDINFO bol príležitosťou, pri ktorej organizátori z UMB v Banskej Bystrici požiadali Slovenskú informatickú spoločnosť o pomoc a podporu zorganizovania konferencie. Slovenská informatická spoločnosť podporila konferenciu poskytnutím finančných prostriedkov na pobyt piatich pozvaných prednášajúcich, ktorých príspevky sú uverejnené v tomto zborníku, a tiež prevzatím garancie nad programom konferencie.

Ako vidieť, do programového výboru DIDINFO 2009 sa podarilo získať rad významných odborníkov, ktorí pomohli posúdiť články ponúknuté autormi k prezentácii na DIDINFO. Temer všetky články boli posúdené aspoň tromi členmi programového výboru. Týmto by som rád poďakoval všetkým členom programového výboru, ktorí poslali svoje posudky včas. Na základe posudkov a následnej diskusie, ktorú veľmi uľahčil svojou softwarovou podporou konferenčný systém EasyChair vybral programový výbor 35 príspevkov zo 44 ponúknutých. Tieto príspevky sú uverejnené v tomto zborníku a budú prednesené na konferencii. Ponúknuté príspevky prezentujú výskum ale najmä mnoho informácií a skúseností z bežnej pedegogickej práce učiteľov. Poskytnutie priestoru na výmenu skúseností a informácií je významnou funkciou konferencie DIDINFO.

Dôležitou súčasťou programu konferencie sú aj workshopy a diskusie na aktuálne témy súvisiace so školskou reformou, maturitami a tiež využitím otvoreného softvéru a rôznych techologických platforiem v oblasti vzdelávania.

Rád by som poďakoval pozvaným prednášajúcim za ich príspevok do programu konferencie, všetkým autorom, ktorí ponúkli svoje príspevky a tým umožnili zostaviť program, ktorý Vám predkladáme, ďalej organizačnému tímu, ktorý sa pod vedením dr. Sudolskej a dr. Huraja postaral o organizačné zabezpečenie konferencie a dr. Hurajovi aj za technickú podporu práce Programového výboru a prípravu zborníka konferencie do tlače.

Branislav Rovan
Predseda Programového výboru
a
Predseda Výkonného výboru
Slovenskej informatickej spoločnosti

Marec 2009

K pätnástemu ročníku konferencie DIDINFO

Sme právom hrdí na **tradíciu vzdelávania** v Banskej Bystrici.

Počiatky rozvoja humanizmu a reformácie sa už v 15. a 16. storočí odrazili v zriadení latinskej školy. Vo vzdelanosti následne pokračovala škola nemecká.

Zavádzanie **prírodovedných predmetov** do škôl v Banskej Bystrici datujeme do obdobia 17. storočia. V 18. storočí zvýšil prestíž banskobystrického vzdelávania **Matej Bel**, ktorý svojou pedagogickou a výskumnou prácou prispel k jeho dobrému postaveniu v rámci celého Uhorska.

Už v polovici 19. storočia sa podarilo v Banskej Bystrici založiť učiteľskú preparandiu a v 20. storočí sa presadzovanie požiadavky vysokoškolského **vzdelávania učiteľov** napokon naplnilo zákonom z roku 1946 - vznikla Pedagogická fakulta pri Slovenskej univerzite v Bratislave s pobočkami v Banskej Bystrici a v Košiciach. Ďalšiu významnú etapu v histórii prírodovedného vzdelávania v Banskej Bystrici prednaznačil od roku 1959 **Pedagogický inštitút**. Neskôr, v roku 1964, na základe zlúčenia pedagogických inštitútov v Banskej Bystrici a Martine, vznikla samostatná **Pedagogická fakulta**. No a od nej, cez rok 1992, kedy sa Pedagogická fakulta rozčlenila na **Fakultu humanitných a prírodných vied** a Pedagogickú fakultu (a tieto sa stali základom budovania **Univerzity Mateja Bela v Banskej Bystrici**) bolo treba spraviť len krôčik k roku 1995 a k našej dnešnej **Fakulte prírodných vied UMB** a na nej fungujúcej **Katedre informatiky**.

So všetkou skromnosťou možno konštatovať, že počas viac ako 15 rokov existencie pod hlavičkou Katedry informatiky Fakulty prírodných vied sme získali dôstojné miesto medzi katedrami Univerzity Mateja Bela i medzi akademickými pracoviskami podobného zamerania na Slovensku. Zásluhu na súčasnom postavení katedry majú viacerí pedagógovia, vedeckí pracovníci i technickí a administratívni pracovníci. Dobré meno katedry šíria aj desiatky našich absolventov, ktorých doteraz „vychovali“ naše študijné programy. Všetkým úprimne ďakujem a som presvedčený, že najbližšie obdobie, charakterizované aj výraznou spoločenskou zmenou vychádzajúcou z kladov vstupu Slovenska do Európskej únie, bude zas obdobím ďalšieho rozvoja našej katedry. Získavanie postavenia akceptovaného Európou, obecne zahraničím, je úloha, ktorej bude venovaná výrazná pozornosť každého z nás.

Okrúhle jubileá sú v ľudskom živote míľnikmi, pri ktorých sa zastavíme v toku dní, aby sme sa obzreli späť a zaspomínali na dni minulé. Ak oslavuje pracovisko – Katedra informatiky – **15. (!) výročie konania dnes už celoslovensky významnej zahraničnej konferencie DIDINFO**, potom takéto zastavenie a obzretie sa späť znamená zvlášť slávnostný okamih. Pri tomto pohľade je dôležité si uvedomiť, **v čom je základ úspechu našej konferencie**.

Tu by som najradšej konštatoval, že zdedenou výhodou nášho slovenského školstva (a zrejme teda aj základom úspechu našej konferencie) je už viac ako tri desiatky rokov budovaný **samostatný predmet informatika**. Keby to však konečne už bola pravda a tú výhodu by bolo vidno. Viac však pozorujem **pretrvávajúcu neistotu v koncepcii predmetu informatika a**, do istej miery, stále ešte **aj nedoceňovanie jej významu**, najmä na základných a stredných školách. Táto „pretrvávajúca nevýhodnosť postavenia informatiky“ je však, paradoxne, práve základom úspechu našej konferencie; heslo: „**Je najvyšší čas niečo s tým** (=informatkou) **systematicky a koordinovane robiť“ -** spája nás všetkých a každoročne nás ženie dopredu v ústrety lepšiemu zajtrajšku informatiky.

Život v modernej informačnej spoločnosti a efektívna účasť každého na znalostnej ekonomike **si vyžaduje, aby vzdelávanie** už na základných a stredných školách **rozvíjalo kritické vnímanie, myslenie, vyššie poznávacie funkcie, flexibilné rozhodovanie a zvládanie epizodických problémov a situácií.** Každý si uvedomuje, že **žiadna profesia sa dnes nezaobíde bez kontaktu s informatikou**. **Nepopierateľnosť nutnosti informatiky tkvie dnes aj v globálnej bezpečnosti**. Ak sa nezvýši **informatická gramotnosť**, budeme mať problémy spôsobované hlúpymi používateľmi informačných technológií. Preto s informatikou začíname **hravou formou už v predškolskej príprave**, pokračujeme **v dobre pripravenej podobe na základnej škole** a **v starostlivo volenej šírke predmetov na strednej škole**. **A** vzdelanie uzatvárame **na vysokej škole cielene a efektívne pripravenými študijnými programami. Všade dnes už aj s podporou IKT a všade dnes už tak, aby sme, my všetci, ktorým na informatike záleží, o sebe vedeli**. Čas vyhradený informatike totiž musí na školách vytvoriť dostatočný rámec nielen pre učenie sa kľúčových informatických pojmov, ale aj pre samostatnú prácu s IKT, pre dôkladné precvičenie látky, pokusy, budovanie súvislostí a nadhľadu, pre učenie vyjadrovať sa, zaujať vlastné stanovisko a prezentovať výsledky práce – skrátka pre rozvoj myslenia každého jedinca.

O tomto všetkom naša konferencia pojednáva, kvôli tomuto všetkému ju radi organizujeme a Vy navštevujete, pre toto všetko je úspešná.

Aj tento rok je **ústredným motívom** konferencie poskytnúť účastníkom najnovšie poznatky z oblasti vyučovania predmetu Informatika, vrátane poznatkov z informačných a komunikačných technológií (IKT) a rovnako aj edukačných technológií; umožniť im prezentovať výsledky vlastnej vedecko-výskumnej činnosti so špeciálnym zameraním na didaktické aspekty informatickej výučby. Naša katedra, fakulta, ako aj všetky spolupracujúce pracoviská, sa zorganizovaním konferencie snažia využiť rokovací čas konferencie na vyjadrenie svojho vlastného „pedagogického a vedeckého postoja“ k svojmu žiakovi a študentovi, postoja, ktorý je vo svojom princípe otvorený a založený na tradičných európskych demokratických hodnotách. Konferencia je práve z tohto dôvodu organizovaná aj pre pedagógov, ktorí vyučujú študentov - budúcich učiteľov predmetu Informatika na strednej a základnej škole. Tešíme sa však z účasti všetkých učiteľov a rovnako náš teší aj prítomnosť zástupcov z vedecko-výskumných pracovísk a zástupcov z iných príbuzných odborov.

Vážení priatelia, po zoznámení sa s obsahom referátov, ktoré odznejú na našej konferencii, som zistil, že **charakteristickou črtou všetkých príspevkov je to, že integrujú v sebe rozvoj viacerých aspektov premeny školy: budovanie technickej, technologickej a informačnej infraštruktúry školy – systematickú prípravu učiteľov - tvorbu moderného edukačného obsahu – informatický výskum a informatickú vedu.** To je veľmi dobre! Lebo spoločnosť si vyžaduje stále viac digitálne gramotných ľudí. **Je to impulz pre organizovanie konferencií, ako je táto naša.**

Nech teda žije konferencia DIDINFO 2009!

Doc. Ing. Ľudovít Trajteľ, PhD.
Vedúci Katedry informatiky FPV UMB

Marec 2009

Obsah

Pozvané prednášky (Invited Papers)

OLYMPIÁDA V INFORMATIKE
A. BLAHO 10

PEDAGOGICKÝ VÝSKUM V INFORMATIKE A INFORMATIZÁCII (2. ČASŤ)
I. KALAŠ 15

THE PEDAGOGY OF PROGRAM DESIGN
V. KRŇANOVÁ PROULX 25

PEDAGOGIKA A NÁSTROJE E-LEARNINGU
J. MIKULECKÁ 35

ŽÁKOVSKÉ PROJEKTY VE VÝUCE INFORMAČNÍCH TECHNOLOGIÍ
J. VANÍČEK 41

Vybrané vedecké prednášky (Selected Research Papers)

VYHODNOTENIE RELIABILITY HODNOTENIA OLYMPIÁDY V INFORMATIKE
M. FORIŠEK 50

PROGRAMOVATEĽNÉ HRAČKY A ALGORITMICKÉ MYSLENIE PREDŠKOLÁKOV
J. PEKÁROVÁ 54

O DIDAKTIKE PROGRAMOVANIA
Ľ. SALANCI 58

ÚVAHA O DIMENZIÁCH ALGORITMICKÉHO MYSLENIA A JEHO ROZVÍJANÍ
Ľ. ŠNAJDER, J. GUNIŠ 62

Krátke prednášky (Short Papers)

DRUHÁ VZDELÁVACIA AKTIVITA V NÁRODNOM PROJEKTE ĎALŠIEHO VZDELÁVANIA UČITEĽOV V PREDMETE INFORMATIKA verzus ROZŠIRUJÚCE ŠTÚDIUM INFORMATIKA
G. ANDREJKOVÁ 67

MODELOVANIE A SIMULÁCIA V ŠKOLSKEJ MATEMATIKE A FYZIKE S EXCELOM
J. BEŇAČKA 71

OPENING THE GATE TO OBJECT ORIENTED APPROACH IN TEACHING PROGRAMMING
I. BRODENEC 75

K SENZOROM V NAŠICH ZÁKLADNÝCH ŠKOLÁCH
P. CVIK 78

INFORMATIKA A MULTIMÉDIÁ V ŠKOLE A MIMOŠKOLSKEJ ČINNOSTI
V. DRÁB 82

DISTANČNÍ KURZ INTERNET JAKO PROSTŘEDEK PRO PODPORU VÝUKY
Z. FILIPI, T. PŘIBÁŇ, V. VRBÍK, ... 86

EDITOR LAMBDA – NÁSTROJ NA SPRÍSTUPNENIE MATEMATIKY NEVIDIACIM A JEHO ROZŠIROVANIE NA SLOVENSKU
M. HORŇANSKÝ, P. LECKÝ 90

IMPLEMENTÁCIA PROGRAMÁTORSKÉHO ROZHRANIA PRE TVORBU INTERAKTÍVNYCH VZDELÁVACÍCH MATERIÁLOV
R. HORVÁTH 94

INOVÁCIE KURIKULA A VÝBER VÝUČBOVÝCH METÓD
J. JACKOVÁ 98

PRÍPRAVA BUDÚCEHO UČITEĽA V REÁLNOM PROSTREDÍ ŠKOLY
B. JAKUBÍKOVÁ, Z. JUROŠKOVÁ, L. LÍŠKOVÁ 101

SÚ TESTY ECDL VHODNÉ NA EVALUÁCIU POČÍTAČOVEJ GRAMOTNOSTI BUDÚCICH UČITEĽOV?
Ľ. JAŠKOVÁ, J. PEKÁROVÁ 105

KOGNITÍVNY TÚTOR PRE ZAČIATOČNÍKOV V PROGRAMOVANÍ
M. KABÁTOVÁ 109

MOŽNOSTI VYUŽITIA ALTERNATÍVNYCH NÁSTROJOV GIS PRI VYUČOVANÍ GEOGRAFIE A EKOLÓGIE
J. KRNÁČ 113

VESELÁ ŠKÔLKA LIENKA. DVA POHĽADY NA EDUKAČNÝ SOFTVÉR PRE PREDŠKOLÁKOV
A. KROMMEROVÁ, J. PEKÁROVÁ 117

DVUI – NÁRODNÝ PROJEKT ĎALŠIEHO VZDELÁVANIA UČITEĽOV ZÁKLADNÝCH ŠKÔL A STREDNÝCH ŠKÔL V PREDMETE INFORMATIKA
Z. KUBINCOVÁ, M. CHALACHÁNOVÁ 121

DIDAKTICKÝ TEST AKO SPÄTNOVÄZOBNÝ PRVOK PRE FORMU E-LEARNINGOVEJ EDUKÁCIE
A. KÚTNA 125

INFORMATICKÉ SÚŤAŽE NA SLOVENSKU
JULIANA LIPKOVÁ 129

INFORMATICKÁ VÝCHOVA: ELEKTRONICKÉ PRACOVNÉ LISTY
G. LOVÁSZOVÁ 132

TVORBA VZDELÁVACÍCH CIEĽOV VO VYUČOVANÍ INFORMATIKY
J. MAJHEROVÁ 136

AKTIVITY VO VIDEOEDITORE PRE II. STUPEŇ ZŠ
K. MIKOLAJOVÁ 140

OD ELEKTRONICKEJ UČEBNICE K UČENIU SA
M. MORAVČÍK, J. PEKÁROVÁ 144

POČÍTAČOVÁ GRAFIKA A JEJ EDUKÁCIA
D. ONAČILOVÁ 148

VYUŽITIE POČÍTAČOVÝCH HIER VO VÝUČBE PROGRAMOVACIEHO JAZYKA IMAGINE
H. PALÁSTHY, I. ČERNÁK 152

VÝUČBA TVORBY WEBOVÝCH STRÁNOK NA STREDNEJ ŠKOLE
D. POLČIN, L. BIELÁKOVÁ 155

ORIENTED GRAPHS IN THE INFORMATICS
B. SIVÁK, M. SUDOLSKÁ 159

METODICKÉ POZNÁMKY K VÝUČBE OBLASTI KOMUNIKÁCIA PROSTREDNÍCTVOM IKT INTERNET = WEB + E-MAIL?
Ľ. ŠNAJDER, J. GUNIŠ, V. GUNIŠOVÁ, Ľ. JAŠKOVÁ 162

MOŽNOSTI NASADENIA OTVORENÉHO SOFTVÉRU NA ŠKOLÁCH
M. ŠRÁMEK 166

NÁROČNOSŤ ÚLOH V SÚŤAŽI INFORMATICKÝ BOBOR
P. TOMCSÁNYI 170

INTERAKTÍVNE ÚLOHY V SÚŤAŽI INFORMATICKÝ BOBOR
M. TOMCSÁNYIOVÁ 174

DĚTI A LINUX
Š. VAVREČKOVÁ 178

INFORMATIKA – ČO TO JE?
M. WINCZER 182

KATEDRA INFORMATIKY FPV UMB
SLOVENSKÁ INFORMATICKÁ SPOLOČNOSŤ
METODICKO-PEDAGOGICKÉ CENTRUM ALOKOVANÉ PRACOVISKO B. BYSTRICA
ÚSTAV INFORMÁCIÍ A PROGNÓZ ŠKOLSTVA ŠKOLSKÉ VÝPOČTOVÉ STREDISKO BB

APPLE V ŠKOLE
DIGITÁLNA ŠKOLA – GAMO
SUN MICROSYSTEMS PODÁVA ŠKOLÁM POMOCNÚ RUKU

Pozvané prednášky

Invited Papers

OLYMPIÁDA V INFORMATIKE

RNDR. ANDREJ BLAHO

Katedra aplikovanej informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina, Bratislava,
tel. 02/602 95 442, e-mail: andrej.blaho@fmph.uniba.sk

Článok stručne charakterizuje 24 rokov olympiády v informatike z pohľadu organizátorov súťaže. Ukazuje vývoj charakteru zadaní v priebehu existencie súťaže ale aj pozadie fungovania samotnej olympiády. Článok je určený najmä učiteľom informatiky, ktorí pripravujú, resp. sa chystajú pripravovať svojich žiakov do tejto súťaže.

Kľúčové slová: *olympiáda v informatike, matematická olympiáda, informatika v škole, programovanie, programovacie jazyky*

1. ÚVOD

Olympiáda v informatike je jedna z najmladších predmetových olympiád pre žiakov stredných škôl. Historicky začala pred 24 rokmi v šk. roku 1985/86 hoci nie hneď ako olympiáda v informatike, ale zatiaľ len ako súčasť už dobre zabehnutej matematickej olympiády. Keďže v tých časoch sa školská informatika sústredila hlavne v predmete programovanie, nová kategória dostala názov kategória P. To, že sme začali fungovať pod matematikmi, bolo v tých časoch výborné rozhodnutie. Vďaka tomu nám matematická komunita pomohla organizačne naštartovať veľmi dobre premyslenú súťaž, ktorá už od prvého svojho roku bežala na vysokej odbornej úrovni. Organizačne až do roku 1993 bola táto kategória riadená z troch centier, ktoré boli na fakultách v Bratislave, v Prahe a v Brne (neskôr pribudli Košice). Našu olympiádu v tých časoch v ústrednej komisii MO zastupoval prof. Branislav Rovan. Po rozdelení republík sa rozdelila aj naša olympiáda na dve republikové olympiády, ktoré ale naďalej veľmi úzko navzájom spolupracujú. Veď už 15 rokov vytvárame zadania pre všetky kolá spoločne s kolegami v Čechách.

Tak, ako aj iné olympiády, má aj tá naša za cieľ podchytiť talenty. Hoci by nás iste potešila veľká účasť súťažiacich, predsa len naozajstných talentov v riešení ťažších algoritmických problémov bude na Slovensku obmedzený počet. Našim súpútnikom je korešpondenčný seminár z programovania, ktorý vznikol na FMFI UK o dva roky skôr ako samotná olympiáda. Tento seminár už od úplných začiatkov súťaže pomáha vychovávať najlepších olympionikov a vlastne naznačuje hranice reálneho počtu súťažiacich.

Organizačne informatika prevzala model matematickej olympiády a preto už od prvých rokov súťaže to boli tzv. "teoretické" úlohy. Postupom času sa viac a viac súťažných úloh rieši nie na papieri, ale na počítači, tzv. "praktické" úlohy.

2. AKÁ JE TO SÚŤAŽ?

Žiaci v každom kole riešia sadu náročných algoritmických úloh. Pri hodnotení týchto riešení sa najväčší dôraz kladie na efektívnosť riešenia. Samozrejme, informatici vedia, že pri algoritmických problémoch je práve toto to najťažšie – často aj komisia, ktorá posudzuje riešenia, sa dosť natrápi, aby nielen porovnala úroveň efektívnosti rôznych riešení, ale niekedy aj odhalila správnosť niektorých veľmi netypických a pritom často správnych postupov.

3. AKÝ TYP ÚLOH ŽIACI RIEŠIA?

Ako sme už spomenuli, sú to len algoritmické úlohy. Snahou je pokryť čo najväčší okruh problémov na rôznych údajových štruktúrach od práce s reťazcami, cez polia, geometrické problémy, až po najrôznejšie typy stromov a grafov. Môžeme sa ale pochváliť jednou špecialitou našej národnej súťaže (teda aj v Čechách), a to tým, že v každom ročníku súťaže sa objavuje jedna séria úloh (v každom kole jedna úloha) z nejakej oblasti niektorého teoretického modelu. Spomeňme napr. Minského stroj, Turingov stroj, parelalený počítač, prekladací automat, zásobníkový automat, grafový automat atď. – v každom ročníku súťaže je to vždy úplne iný model.

4. AKO SA POSTUPNE VYVÍJALA ŠTRUKTÚRA ZADANÍ

Prvých 9 ročníkov súťaže mali riešenia všetkých úloh teoretický charakter , t.j. žiaci zapisovali algoritmy väčšinou v ľubovoľnom programovacom jazyku, resp. ho popísali slovami a opravovatelia tieto riešenia prechádzali bez kontroly na počítačoch. V 10. ročníku súťaže, teda v šk. roku 1994/95, sa prvýkrát objavili tzv. praktické úlohy: jeden súťažný deň celoštátneho kola žiaci riešili dve úlohy na počítači – mohli si vybrať riešenie buď v jazyku Turbo Pascal alebo Turbo C.

Teraz sa už dosť ťažko predstavujú problémy, ktoré v tých časoch bolo treba z organizačného pohľadu riešiť. Koncom 80-tych rokov, t.j. v prvých rokoch súťaže, väčšina žiakov pravdepodobne nemala doma žiadne počítače a samozrejme, že v tých časoch pre nás neexistoval nielen web ale ani email. Študijná literatúra pre žiakov nebola skoro žiadna a informatická osveta sa na stredných školách riešila len prostredníctvom korešpondenčných seminárov.

Začiatkom 90-tych rokov sa štartuje medzinárodná olympiáda v informatike (prvé IOI bolo na jar 1989), ktorá nám ukázala cestu, ako sa dá efektívnosť algoritmických úloh hodnotiť pomocou počítača. V prvých svojich rokoch aj IOI hľadala správnu cestu, ako zadať úlohy, ktorých vyhodnotenie by sa dalo čo najviac zautomatizovať. Dá sa povedať, že v roku 1994 už boli tieto idey dostatočne rozpracované a teda sme to naštartovali aj my. Tým sme sa priblížili nielen k medzinárodnému spôsobu súťaženia, ale pre našich žiakov začali byť dôležité nielen zručnosti v teoretickom riešení úloh, ale aj zvládanie praktického programovania s reálnymi obmedzeniami pamäte a času. Zrejme nie každý, kto má talent "na papieri" vyriešiť aj najťažšie algoritmické problémy, má šancu uspieť v medzinárodnej súťaži, kde tieto svoje schopnosti musí predviesť na počítači.

5. AKÁ JE IDEA ZADANÍ

Súťaž začína domácim kolom, v ktorom žiaci riešia 4 rôzne algoritmické úlohy. Na toto majú čas niekoľko mesiacov: zadania sú väčšinou zverejnené v septembri a ich riešenia treba odovzdať v priebehu novembra. Každá úloha je samozrejme úplne iná a pokrýva jedny z algoritmických oblastí. V tomto domácom kole sa ale nepredpokladajú riešenie veľmi náročných algoritmov – cieľom je zoznámiť sa s problematikou a navrhnúť prvé verzie algoritmov nad zadanými štruktúrami. Už v tomto domácom kole je vždy štvrtá úloha z nejakej teoretickej oblasti informatiky. Žiaci dostávajú niekoľko stranový študijný materiál, ktorý vysvetľuje túto novú oblasť a ukazuje vzorové riešenia niekoľkých úloh. V domácom kole sa teda predpokladá, že si to žiaci naštudujú a budú schopní riešiť prvé jednoduché úlohy z často veľmi náročných oblastí teoretickej informatiky.

V domácom kole sa samozrejme predpokladá pomoc učiteľa informatiky – na mnohých školách títo učitelia vedú programátorské krúžky, v ktorých sa žiaci zoznamujú s novými dátovými štruktúrami a s rôznymi technikami v teoretickej oblasti zo štvrtej úlohy. Učiteľ má totiž k dispozícii komentáre k riešeniam úloh, vďaka čomu aj ak danú problematiku zatiaľ ešte nepoznal, môže žiakov správne viesť.

Súťaž ďalej pokračuje krajským kolom: žiaci sa v jeden súťažný deň zídu v každom kraji a počas jedného doobedia riešia 4 úlohy. Charakter úloh je zvyčajne ten istý ako bol v domácom kole, žiaci väčšinou riešia náročnejšie varianty úloh domáceho kola. Samozrejme, že tu už riešenia nemôžu konzultovať ani s učiteľom ani so spolužiakmi. Majú na to čas len 4 hodín, čo je naozaj veľmi veľkým obmedzením. Hovoríme, že úlohy sú teoretické, nakoľko sa riešia bez pomoci počítača len na papieri. Nebazíruje sa ani na presnej syntaxi, často sú povolené slovné popisy niektorých častí programu. Okrem poslednej úlohy, ktorá je viazaná na konkrétnu tému a teda na špecifický jazyk zápisu riešenia, žiaci môžu zapísať riešenie v Pascale alebo C/C++.

Súťaž vyvrcholí na celoštátnom kole. Sem sa dostávajú najlepší zo všetkých krajov. Opäť sa riešia náročnejšie varianty úloh z predchádzajúcich kôl, ale aj úplne iné témy. Celoštátne kolo je rozdelené na dva súťažné dni: prvý deň sa riešia tri "teoretické" úlohy, druhý deň sú to dve "praktické" úlohy pri počítači. Charakter týchto praktických úloh je rovnaký ako tých teoretických, len ich bude treba naozaj kompletne odladiť. Pri hodnotení teoretických je najdôležitejšia dobrá idea a jej popis, praktické úlohy musia byť úplne dokončené, nakoľko sa hodnotí len správnosť výstupných údajov a už vôbec nie slovný popis idey algoritmu.

6. KAM SA DOSTALA OLYMPIÁDA DNES

Za tých necelých 25 rokov trvania olympiády v informatike sa v toho dosť zmenilo:

- Zmenila sa samotná **informatika v škole** – v tých časoch to bolo na 90% percent o programovaní, ale väčšinou vo Fortrane a Basicu. Len zopár škôl na Slovensku malo vôbec nejaké počítače, na ktorých by sa dalo aj programovať. Pascal sa začal objavovať koncom 80 rokov vďaka PMD-85. Naproti tomu, súčasná informatika pokrýva široké spektrum tém od multimédií a grafiky až po operačné systémy, počítačové siete, vírusy a riziká IT. Hoci algoritmy a programovanie sú súčasťou dnešnej informatiky, pre väčšinu škôl ostávajú len na najtriviálnejšej úvodnej úrovni.
- Zmenil sa **prístup žiakov k počítačom** – od prvých rokov súťaže, keď sa niektorí žiaci dostali k počítačom len v škole aj to s veľmi obmedzeným prístupom, až po súčasnosť, keď pravdepodobne všetci súťažiaci majú doma svoj vlastný počítač a to s niekoľkými kvalitnými vývojovými programátorskými prostrediami. Všetci dnešní súťažiaci prakticky celý svoj život žijú vo svete počítačov, kde sú od malička obklopení najrôznejšími typmi technológií.
- Zmenili sa **programovacie jazyky** – prvé roky boli poznačené bojom za Pascal proti Basicu.

V súčasnosti je to boj za uchovanie Pascalu a akceptovanie aj jazykov, ktoré hoci nie sú didakticky vhodné pre stredné školy, ale sú v profesionálnom svete najrozšírenejšie. Okrem dvojice jazykov C a C++, ktoré sú súčasťou oficiálnych jazykov aj na medzinárodnej súťaži, sa pravdepodobne k nim časom pridá aj Java. Uvidíme, ako sa to bude vyvíjať v nasledujúcich rokoch.

- Zmenil sa **prístup k odborným materiálom** – v prvých rokoch boli všetky materiály len v tlačenej forme, prakticky neexistovali skoro žiadne učebnice použiteľné pre súťažiacich (spomeňme Wirthovu klasickú knihu Algoritmy a štruktúry údajov). V tých časoch jediným zdrojom boli študijné materiály k riešeniam KSP (korešpondenčný seminár z programovania), prípadne postupne šírené zadania a riešenia olympiády z predchádzajúcich rokov. Naproti tomu, v súčasnosti žijeme v opačnom extréme: súťažiaci sú zahltení nestráviteľne veľkým množstvom zdrojov najmä z internetu. Majú k dispozícii nielen študijné materiály zo Slovenska (napr. z rôznych súťaží a z vysokoškolského štúdia), ale aj z programátorských súťaží na celom svete.
- Zmenila sa **náročnosť informatických tém** v súťažiach – náročnosť niektorých úloh v prvých ročníkoch môže teraz vyvolávať úsmev. Organizátori prvých ročníkov museli naozaj brať do úvahy to, že žiaci nemajú absolútne žiadne možnosti vopred si naštudovať ani niektoré základné dátové štruktúry a algoritmy. Často sa s nimi prvýkrát stretli až počas súťaže a prakticky "za jazdy" museli objavovať aj niektoré triviálne postupy. Až časom sa začali objavovať aj náročnejšie témy a programátorské techniky, keď bolo vidieť, že tí najvyspelejší súťažiaci zvládajú pôvodnú úroveň náročnosti príliš dobre.
- Zmenili sa a stále sa ešte veľmi výrazne menia **technické možnosti realizácie priebehu** samotnej súťaže – v prvých rokoch to bolo len písanie riešení na papier, neskôr ladenie niektorých úloh na počítači, v súčasnosti automatické vyhodnocovanie kvality riešenia už počas súťaže.

7. ČO VŠETKO JE ZA ORGANIZOVANÍM JEDNÉHO ROČNÍKA SÚŤAŽE

Vysvetlili sme, ako prebieha samotná súťaž: súťažiaci sa postupne prebojuje cez tri kolá až do výberového sústredenia na medzinárodnú súťaž. Myslíme ale, že by bolo zaujímavé aspoň čiastočne predstaviť, čo sa deje skôr ako samotný ročník súťaže vypukne a tiež pozadie organizácie samotnej súťaže. Nepôjdeme do detailov, len si ukážme najdôležitejšie činnosti.

V predstihu celý jeden rok sa ako prvé pripravia zadania všetkých troch kôl. Našťastie vďaka spolupráci s českými kolegami, ktorí sa starajú o svoju republikovú olympiádu, každý druhý rok pripraví kompletnú sadu všetkých úloh buď slovenská alebo česká strana. Takže jeden rok sa zostavujú príklady v Čechách a my ich len prekladáme do slovenčiny a druhý rok je to naopak. Pripraviť zadania pre jeden celý ročník je náročná tímová práca a vyžaduje obrovské skúsenosti a samozrejme neuveriteľnú tvorivosť. Kolektív tvorcov sa snaží naformulovať čo najzaujímavejšie témy, ale pritom "neuletieť " s náročnosťou, aby boli úlohy riešiteľné a najlepší zo súťažiacich mali šancu ukázať, že zvládajú aj veľmi náročné postupy najefektívnejšieho riešenia.

Súčasne so zadaniami sa pripravujú aj vzorové riešenia, a pre praktické úlohy sa začínajú navrhovať aj sady testovacích dát, na ktorých bude testovací softvér kontrolovať kvalitu riešenia.

Toto všetko je skoro úplne pripravené v čase, keď ešte ani neskončil predchádzajúci ročník súťaže.

Paralelne s týmto, ale na inej organizačnej úrovni, vzniká kalendár všetkých súťaží. Toto má na starosti Iuventa a hlavným cieľom je, aby minimálne kolidovali tie súťaže, v ktorých je šanca, že sa v nich zúčastňujú rovnakí žiaci (napr. matematická, informatická a fyzikálna olympiáda). Pre našu olympiádu musíme samozrejme koordinovať súťaž aj s Českou republikou, čím sa toto rozvrhovanie stáva ešte zložitejším.

Keďže najneskôr v septembri sa zverejňujú zadania domáceho kola, ešte pred letom sa pripravia definitívne znenia úloh v publikovateľnom tvare aj s komentármi a s pokynmi na zápis riešení. Zároveň vzniká aj definitívny tvar komentárov a vzorových riešení pre učiteľov, ktoré sa samozrejme zverejňovať nemôžu – dostávajú ich začiatkom školského roku priamo učitelia do škôl. Okrem tohto sa pripraví aj návod pre budúce opravovanie týchto príkladov – opravovatelia neskôr dostanú tieto podrobné pokyny, ako majú ohodnotiť každé riešenie.

Od tohto školského roku (2008/09) sa "praktické" úlohy objavujú aj v domácom kole. Každý súťažiaci musí odovzdať dve z úloh cez webové rozhranie, pričom vyhodnocovací softvér mu okamžite oznámi počet bodov, ktoré za dané riešenie získava. Súťažiaci má potom šancu riešenie vylepšiť a odovzdať znovu, aby získal väčší počet bodov. Vďaka tejto novinke musel realizačný tím pripraviť nielen vyhodnocovací softvér, ale aj kompletné webové rozhranie s evidovaním súťažiacich a odovzdávaných riešení. Samozrejme pre vyhodnocovací softvér museli pripraviť veľmi premyslenú sadu testovacích dát, pomocou ktorej sa rozhodne o kvalite riešenia.

Predstavme si, že už uplynul čas domáceho kola. Všetci súťažiaci odovzdali svoje riešenia a tieto sa dostali k ľuďom v príslušných krajských komisiách. Teraz nastupujú tímy opravovateľov pri každom krajskom výbore, ktorí na základe pripravených pokynov, vyhodnotia riešenia a zostavia zoznamy úspešných riešiteľov pre postúp do krajských kôl. Krajská komisia v každom kraji pripraví a rozpošle na školy pozvánky do krajských kôl. Samozrejme, že na príslušný dátum zorganizuje priebeh krajskej súťaže.

Zadania úloh krajského kola spolu aj so vzorovými riešeniami a komentármi musia byť pripravené dávno pred samotným dňom súťaže. Našťastie vďaka internetu a rýchlym tlačiarňam nie sme už veľa rokov odkázaní na klasickú poštu. Samozrejme, že sa pritom nezabúda na bezpečnosť utajenia, aby nás šikovní informatici neprekvapili svojim talentom.

Riešenia krajského kola sa opäť podľa pripravených pokynov vyhodnotia a potom sa pošlú do centra na FMFI. Tu už väčší tím opravovateľov opäť prejde všetky riešenia (samozrejme, že to už ide rýchlejšie) a zostaví výslednú tabuľku všetkých riešiteľov. Táto poslúži pre pozývaných do celoslovenského kola.

Medzičasom sú už vo veľkom rozbehnuté prípravy na celoštátne kolo. Napr. Iuventa spolu s krajským výborom, kde bude bežať celoštátne kolo, sa stará o zabezpečenie ubytovania, stravy, súťažných priestorov a sprievodného programu. Organizačný tým už musí mať pripravené nielen testy zadaní a riešení, ale najmä súťažný softvér, ktorý sa tesne pred súťažou nainštaluje na počítače. Zrejme kľúčovým bude pripraviť aj server, na ktorom bude prebiehať samotné vyhodnocovanie praktických úloh a tiež sa tu nachádza webové rozhranie na odovzdávanie vyriešených úloh počas súťaže. Keďže tento organizačný tým je už perfektne zabehnutý a pripravený aj na tie najnepravdepodobnejšie problémy, priebeh súťaže sa pre súťažiacich zdá absolútne jasný a jednoduchý. Hneď po prvom súťažnom dni tím opravovateľov prejde a oboduje všetky riešenia, čo je činnosť často namáhavejšia a náročnejšia ako samotné riešenie úloh súťažiacimi. Niekedy treba napríklad uhádnuť nejakú "geniálnu" ideu riešenia len na základe veľmi hrubej kresby, nepochopiteľného komentáru a šialene prekoplikovaného úryvku programu.

Keď máme celoštátne kolo úspešne za sebou aj s odovzdanými diplomami, cenami a rôznymi materiálmi, ktoré si žiaci odnášajú domov, tak pre organizačný tým to vôbec ešte neznamená oddych. Treba pripraviť výberové sústredenie pre 12 najlepších, z ktorých sa vyberie reprezentácia pre obe medzinárodné súťaže IOI (medzinárodná olympiáda v informatike) a CEOI (stredoeurópska olympiáda v informatike). Toto sústredenie je vlastne ďalšou ale výrazne náročnejšou súťažou, ako bolo celoštátne kolo: žiaci celý týždeň od rána do večera riešia veľmi ťažké úlohy na úrovni medzinárodnej olympiády. Samozrejme, že sa títo žiaci poriadne potrápia, ale nie menej sa natrápi organizačný tím, ktorý tieto úlohy nielen zostavuje ale aj priebežne vyhodnocuje. Ale záver sústredenia je najväčšou odmenou – konečne poznáme reprezentantov na medzinárodné súťaže, ktorí opäť celému svetu ukážu, že naši žiaci sú zaslúžene jedni z najlepších a konkurujú aj veľmociam ako Čína, Rusko a USA.

My už vieme, že organizačný tím ani počas sústredenia ani po ňom neoddychuje, veď vtedy sa už vo veľkom finišuje nasledujúci ročník a teda zadania, riešenia, testovací softvér, testovacie dáta a pod.

Tento stručný popis priebehu jedného roka z pohľadu organizátorov sme tu uviedli najmä preto, aby sme poďakovali všetkým obetavým nadšencom, ktorí v súčasnosti ťahajú túto olympiádu, ale aj desiatkam organizátorov, ktorí to v priebehu minulých skoro 25 rokov ťahali aj v tých bývalých "romantických" časoch, keď nebol internet, keď nebol Linux, keď nebol Excel, keď sa vyhodnocovalo iba ručne a tak ďalej.

8. ZÁVER

Ešte by sa dalo o olympiáde toho povedať naozaj veľa zaujímavého. Napr. veľmi dôležitou novinkou je vznik kategórie B pre začiatočníkov. Podľa vzoru matematickej olympiády, teda okrem základnej A kategórie, máme aj B kategóriu a dá sa predpokladať, že časom budú aj u nás vznikať kategórie pre základnú školu. Informatika začala prenikať aj na prvý a druhý stupeň ZŠ, tak zrejme o niekoľko rokov sa rozšírime aj o tieto detské kategórie.

Ak by sme rozmýšľali ešte viac do budúcnosti, pravdepodobne časom úplne zmizne papierová verzia riešenia úloh – o niekoľko rokov bude asi bežnejšie písať do notebooku ako na papier. Možno vznikne aj nejaký špecializovaný softvér na on-line riešenie nielen praktických ale aj teoretických úloh.

Okrem rozširovania vekových kategórií a vylepšovania technológií pre odovzdávanie riešení sa časom možno aj samotná školská informatika rozdelí na viac špecializácií a možno vznikne buď viac paralelných súťaží, alebo iná štruktúra kategórií. Možno v ďalekej budúcnosti sa aj samotné programovanie posunie takým smerom, že súčasný spôsob riešenia problémov sa bude zdať smiešny, podobne, ako sa teraz usmievame nad problémami programovania v strojovom kóde. Nechajme sa prekvapiť.

LITERATÚRA

[1] Niklaus Wirth: Algoritmy a štruktúry údajov, Alfa 1987

[2] Oficálne stránky olympiády v informatike na Slovensku:
http://www.oi.sk/ resp. http://ksp.sk/oi/

[4] Oficiálna stránka IOI - International Olympiad in Informatics:
http://www.ioinformatics.org/index.shtml
http://olympiads.win.tue.nl/ioi/

[5] KSP - Korešpondenčný seminár z programovania:
http://www.ksp.sk/

[6] Predmetové olympiády na Iuvente: http://www.olympiady.sk/

[7] Matematická olympiáda - kategorie P v Českej republike:
http://mo.mff.cuni.cz/p/

[8] Programátorská súťaž PALMA (Programovanie, Algoritmy, Matematika): http://palma.strom.sk/

[9] ACM International Collegiate Programming Contest:
http://cm2prod.baylor.edu/

[10] IPSC - Internet Problem Solving Contest: http://ipsc.ksp.sk/

[11] Olympiáda v informatike v Poľsku: http://www.oi.edu.pl/

[12] Timus Online Judge - largest Russian archive of problems from different programming contests:
http://acm.timus.ru/

PEDAGOGICKÝ VÝSKUM V INFORMATIKE A INFORMATIZÁCII (2. ČASŤ)

PROF. RNDR. IVAN KALAŠ, PHD.

Katedra základov a vyučovania informatiky, Fakulta matematiky, fyziky a informatiky
Univerzita Komenského, Mlynská dolina, 842 48 Bratislava
e-mail: kalas@fmph.uniba.sk

V druhej časti našej malej série o pedagogickom výskume v informatike a informatizácii prezentujeme vlastnú cestu od „insitného" či intuitívneho experimentovania k tomu, ako sme „objavili" a ako používame kvantitatívne a kvalitatívne metodológie. V ostatných rokoch cítime rastúcu potrebu erudovane poznať tieto paradigmy – a korektne ich používať vždy, ***keď je to vhodné****. Často však narážame na problémy, pri riešení ktorých nám tradičné výskumné postupy nedávajú do rúk tie nástroje, ktoré potrebujeme. Preto sa spolu s odborníkmi z oblasti TEL (Technology Enhanced Learning) pokúšame spoznať a aplikovať aj iné, moderné a alternatívne výskumné paradigmy. Jednu z nich –* ***výskum vývojom*** *– v tejto časti predstavíme detailnejšie a prezentujeme argumenty, prečo ju považujeme za mimoriadne vhodnú pre edukačný výskum v našej oblasti. Na záver vyslovíme niekoľko zásad, ktoré by mali výskumníci v oblasti TEL dodržiavať, aby sa zaručil ďalší rast kvality nášho odboru na Slovensku.*

Kľúčové slová: *pedagogický výskum, výskum vývojom, technológiami podporované učenie sa, zásady korektného výskumu*

1. ÚVOD

Nič nie je také praktické ako dobrá teória.
J.C.Maxwell (1831 – 1879)

Je to práve teória, ktorá určuje, čo môžeme pozorovať.
A.Einstein (1879 – 1955)

V prvej časti našej série príspevkov o edukačnom výskume v informatike a informatizácii, pozri [10], sme predstavili dve základné výskumné stratégie a prezentovali argumenty, prečo sa musíme systematicky venovať ich spoznávaniu a aplikovaniu vo výskumných projektoch. Povedali sme, že iba tak sa teória vyučovania informatiky môže ďalej rozvíjať ako vedecká disciplína – veď práve korektné postupy pri výskume môžu zaručiť, že na otázky (napr. o poznávacom procese), ktoré si kladieme, dokážeme **korektným spôsobom** nachádzať **primerané odpovede**.

V tejto časti série predstavíme zaujímavý príklad, ako sme v ostatnom čase využili kvantitatívnu stratégiu spôsobom, ktorý je pre ňu nanajvýš primeraný. Vzápätí však vysvetlíme, prečo obe stratégie považujeme v niektorých prípadoch za nedostatočné a obmedzujúce a prečo sa obzeráme aj po ďalších metodológiách výskumu. Podrobnejšie predstavíme jednu z takýchto nových, alternatívnych paradigiem – ***výskum vývojom***, ktorý momentálne azda najlepšie zo všetkých známych stratégií pomáha práve nám – výskumníkom v oblasti informatizácie a v teórii vyučovania informatiky.

2. Z KRÁTKEJ HISTÓRIE NAŠICH VÝSKUMOV

Začnime tým, že stručne a trochu zjednodušene charakterizujeme sedem etáp, ktorými sme ako „začínajúci výskumníci" s kolektívom kolegov a doktorandov na katedre prešli za ostatných desať rokov:

- *Insitné* či *intuitívne obdobie*
  Vývoj edukačného softvéru a iných inovatívnych materiálov na podporu poznávacieho procesu, ktorý sme robili od začiatku 90-tych rokov, vychádzal najmä z osobných skúseností a intuície autorov. Náš príklon k výskumnej práci nám však kládol otázky, na ktoré sme výlučne intuíciou odpovedať nedokázali.
- *Obdobie objavenia kvantitatívnej metodológie a jej nekorektné používanie*[1]
  Spoznali sme niektoré zásady kvantitatívneho výskumu, ale pri jeho používaní sme nedbali o validitu či objektivitu... Najväčšou chybou však bolo, že sme robili zovšeobecnenia uzáverov, na ktoré sme nemali dostatočne reprezentatívne vzorky.
- *Objavenie kvalitatívnej metodológie počas medzinárodných projektov MATCh, Playground, a CoLabs*[2] *a jej využívanie v dizertačných výskumných projektoch*
  Najmä naša dlhodobá spolupráca s kolegami z londýnskeho Institute of Education a London Knowledge Lab a intenzívna účasť v pracovných výboroch WG3.1 a WG3.5 výboru federácie IFIP TC3 pre vzdelávanie nás naučila, že výskumník nemusí pracovať s veľkou vzorkou detí, ak chce korektne hľadať odpovede na svoje výskumné otázky. Ak intenzívne pracujeme s malou skupinou (možno iba jedným, dvoma alebo piatimi deťmi), dozvieme sa o nich viac a môžeme vysloviť teóriu o ich poznávacom procese – ale nesmieme naše pozorovania zovšeobecňovať. V [10] sme predstavili niekoľko výskumných

[1] na svoju obhajobu poznamenajme, že podobné chyby na Slovensku – a inde – robili a robia aj iní výskumníci
[2] pozri projektové stránky [20], [24] a [26]

projektov, ktoré sme realizovali – alebo naďalej realizujeme – práve takouto stratégiou.

- *Objavenie problémov pri používaní kvalitatívnej metodológie vo viacerých výskumných projektoch*
  V ostatných rokoch opakovane narážame pri rôznych výskumných projektoch na takéto protirečenie: výskumník, ktorý používa kvalitatívnu stratégiu, vstupuje do niektorého učiaceho sa prostredia, aby skúmal určitý jav, napr. *stav používania IKT na základných školách*, pozri [18]. Stáva sa teda buď zúčastneným alebo izolovaným pozorovateľom (svedkom) určitého javu, skutočnosti, vzťahov... Skúma ich, analyzuje a vysvetľuje. To isté chceme robiť aj my – ale realitu, do ktorej chceme vstúpiť a skúmať, zároveň aj **vytvárame**, **rozvíjame** alebo **modifikujeme**! Ak chceme napr. skúmať, aké digitálne technológie pomáhajú v predškolskej príprave efektívne naplňať jej didaktické ciele, musíme takéto situácie projektovať, vyvíjať, implementovať a analyzovať! Ak sa chceme dozvedieť, či a ako môžeme rozvíjať programátorské zručnosti a kompetencie pre riešenie problémov u žiakov druhého stupňa ZŠ, **musíme to zároveň robiť**, a zároveň svoju vývojársku prácu evaluovať a korigovať korektnými výskumnými metódami! Kto iný než my by takúto zmenu, takúto **intervenciu**[3] urobil – aby sme ju potom my výskumníci mohli už „iba“ skúmať?

  Zo začiatku sme tento problém iba cítili a nevedeli ho pomenovať. Postupne sme však, najmä účasťou v sieti Kaleidoscope a diskusiami so zahraničnými a domácimi odborníkmi[4] pochopili, že podobné problémy trápia aj iných výskumníkov a didaktikov v príbuzných či podobných oblastiach.

- *Objavenie TEL v sieti excelencie Kaleidoscope*[5]
  Túto etapu charakterizujeme v kapitole 3.

- *Znovuobjavenie kvantitatívnej stratégie a jej korektné používanie*
  Túto etapu charakterizujeme v kapitole 4.

- *Objavenie novej výskumnej paradigmy – výskumu vývojom*
  Túto etapu charakterizujeme v kapitole 5.

3. TEL[6] – TECHNOLÓGIAMI PODPOROVANÉ UČENIE SA

Pri našej práci si celkom prirodzene kladieme otázku, čo vlastne tvorí predmet nášho výskumu a čo sú pre nás – napr. v európskom vedeckom kontexte – príbuzné či podobné výskumné oblasti. Hlavným predmetom nášho skúmania je všeobecno-vzdelávací predmet **informatika** (a **informatická výchova**) a jeho didaktika na ZŠ a SŠ, ale tiež rôzne súvislosti tohto vzdelávania pre ďalšie vekové skupiny: predškolákov, budúcich učiteľov či učiteľov v praxi. Predmetom nášho výskumu je však aj **informatizácia poznávacieho procesu** v každom predmete a aspekte, na každom stupni vzdelávania, v každom veku a vo všetkých formách, od vzdelávania formálneho až po vzdelávanie neformálne a neorganizované.

Označiť tento široký súbor záujmov slovami *teória vyučovania informatiky* nie je dostatočné. Ako vhodnejší sa nám javí nový, zatiaľ prevažne európsky smer výskumu **TEL** – technológiami podporované učenie sa. V roku 2004 sme sa stali zakladajúcim partnerom siete excelencie Kaleidoscope, ktorá si v rámci 6. RP zvolila za cieľ vytvoriť pre túto novú disciplínu komplexný európsky výskumný priestor, pozri [25]. V súčasnosti predstavuje Kaleidoscope spoločenstvo viac než 1000 výskumníkov z 24 rôznych krajín, ktorí sa zaoberajú **vývojom nových spôsobov a metód na skúmanie učenia sa s podporou digitálnych technológií**.

... Kaleidoscope stavia učiaceho sa do centra pozornosti multidisciplinárneho výskumu s teoretickými základmi v kognitívnych a pedagogických vedách, rovnako ako v informatike a oblastiach, ktoré vyvíjajú nové technológie. Sieť má silnú praktickú orientáciu, je zameraná na šírenie inovácií a súťaživosti a generovanie nových foriem kultúrnych skúseností a skúseností v procese učenia sa.

Obr. 1 Logo siete excelencie Kaledoscope

Oblasť TEL, ktorú Kaleidoscope rozvíja, je zatiaľ iba v začiatkoch, postupne si vyjasňuje svoj predmet výskumu a preň vhodné výskumné metódy (jednej z metodológií výskumu vhodných pre TEL sa budeme venovať v kapitole 5). Podľa [11]: ... *Základným problémom výskumníkov v oblasti TEL je nedostatok uceleného porozumenia svojej oblasti. To znižuje našu schopnosť prichádzať s fundamentálnymi teóriami a metódami na riešenie kľúčových problémov pri učení sa s podporou technológií. To ďalej znamená, že nemôžeme tieto absentujúce teórie aplikovať na úrovni návrhu a vytvárania* ***intervencií do poznávacieho procesu****...*

Ak sa pozrieme na prírodné vedy, vidíme priamy súvis medzi ***súčasným*** *a* ***predchádzajúcim*** *výskumom. Tento súvis zaručuje postupné rozširovanie komplexného poznania v oblasti. Naša oblasť sa*

[3] tento pojem podrobnejšie charakterizujeme ďalej v texte
[4] menovite by sme radi spomenuli R. Nossa, D. Pratta a z našich P. Gavoru a J. Zouneka
[5] pozri projektový portál [25] a obr. **1**
[6] TEL – Technology Enhanced Learning

nachádza medzi informatikou a spoločenskými vedami, v ktorých sa takáto ***linearita*** *dosahuje len ťažko, pretože sociálne vedy sú svojou podstatou zakotvené v zložitosti a protirečivosti reálneho sveta. A čo je ešte náročnejšie, skúmajú tie nevypočítateľné tvory nazývané ľudia…*

Po piatich rokoch našej účasti v sieti Kaleidoscope môžeme konštatovať, že sme veľa získali (a pravdepodobne aj niečo priniesli ďalším partnerom siete v rôznych formách spolupráce[7]). Za najvýznamnejší zisk pre náš kolektív považujeme:

- pomoc pri formovaní koncepcie moderného doktorandského štúdia v odbore Teória vyučovania informatiky, a vďaka zapojeniu sa do Virtual Doctoral School siete tiež nárast kvality dizertačného výskumu, ktorý už na katedre prebiehal,
- príležitosti spoznať kvalitný štýl vedeckej práce v oblasti TEL, nové metódy pedagogického výskumu, prístup k rozsiahlym zdrojom vedeckej literatúry, vytvorenie početných osobných kontaktov s odborníkmi z mnohých krajín. Spoznali sme rôzne procesy a postupy v projektovaní, realizácii a evaluácii vedeckej práce, ktoré v súčasnosti v sieti Kaleidoscope dosahujú (či predstavujú) svetovú úroveň.

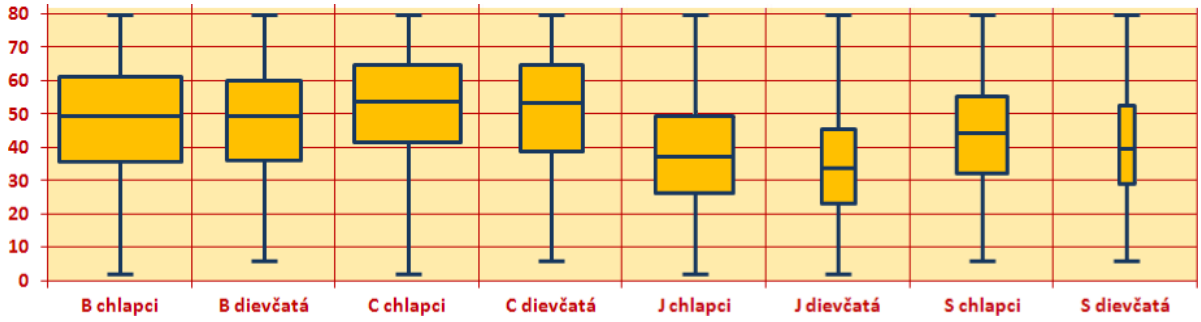
Aj keď sme vďaka sieti prenikli do oblasti TEL a nových metodológií edukačného výskumu, zároveň podstatne vzrástol aj náš rešpekt voči tradičnej kvantitatívnej metodológii. Toto konštatovanie by sa na prvý pohľad mohlo javiť ako protirečivé, ale opak je pravdou: výskumník sa musí naučiť zvoliť si pre svoj konkrétny výskumný cieľ tú najproduktívnejšiu metodológiu. Práve dvom zdanlivým „extrémom" v palete výskumných stratégií venujeme dve nasledujúce kapitoly.

4. KVANTITATÍVNA METODOLÓGIA A ANALÝZA SÚŤAŽE iBOBOR

Vďaka vysokému záujmu žiakov o účasť na druhom ročníku súťaže Informatický bobor, skrátene iBobor, pozri napr. [5], [16], sa nám tento rok dostal do rúk rozsiahly súbor dát[8] o 9317 riešiteľoch niektorej zo štyroch kategórií súťaže z 276 škôl na Slovensku. Keďže hlavným cieľom súťaže je široko plošná propagácia a rozvoj informatického vzdelávania pre čo najvyšší počet žiakov, tento rok sme venovali získaným dátam veľkú pozornosť – vnímame ich ako mimoriadnu príležitosť (a) dozvedieť sa čo najviac o súťažiacich – o tom, ako riešili jednotlivé úlohy, ako na záver celú súťaž a jej úlohy hodnotili a pod., a tiež (b) zlepšiť nasledujúci ročník súťaže tak, aby bol pre žiakov atraktívny a aby nám zároveň čo najlepšie pomohol naplniť naše vzdelávacie a výskumné ciele.

Snažíme sa zdokonaľovať svoje vedomosti o možných kvantitatívnych výskumných postupoch pri objavovaní toho, čo je v pozbieraných dátach ukryté – podľa slov nášho kolegu E. Neuwirtha v Viedenskej univerzity sa *snažíme pozorne počúvať to, čo nám dáta veľmi potichu hovoria*. Zaujímajú nás rôzne javy, napr. úspešnosť žiakov v závislosti na veku a pohlaví, úspešnosť žiakov v jednotlivých úlohách, resp. v rôznych kategóriách úloh alebo analýza náročnosti úloh. Učíme sa používať **názorné vizualizácie**, ktoré vhodne charakterizujú určitý fenomén – a zároveň nás inšpirujú k tomu, aby sme (a) si kládli ďalšie otázky, idúce hlbšie v tomto jave, (b) zvažovali, čo môžeme korektne z týchto pozorovaní vyvodiť, aké hypotézy sformulovať a potvrdiť alebo vyvrátiť a (c) aké rozhodnutia prijať pri tvorbe úloh pre nasledujúci ročník súťaže, ktoré by nám mohli o rok priniesť kvalitné výpovedné dáta zamerané na tento jav.

Príkladom takejto užitočnej a v našej oblasti nie bežnej vizualizácie sú **kvartilové** či **box plot** grafy, pozri [6] a obr. **2**. Táto reprezentácia znázorňuje šírkou škatuľky (boxu) **početnosť vzorky**, strednou horizontálnou čiarou **medián** (teda nie priemer), dolnou a hornou horizontálnou úsečkou **minimálnu** a **maximálnu** nameranú hodnotu (v tomto prípade získané body) a napokon dolnou a hornou časťou



Obr. 2. Kvartilové grafy bodových ziskov v štyroch kategóriách (B – benjamíni, C – kadeti, J – juniori a S – seniori), zvlášť pre chlapcov a dievčatá

[7] najmä vďaka našim skúsenostiam vo vývoji moderných edukačných softvérových nástrojov pre konštruktivistický poznávací proces. Jednu výskumnú aktivitu sme koordinovali, v ďalších dvoch sme boli spoluriešiteľmi – išlo o sériu workshopov pre Virtual Doctoral School, pozri [25], a o spoluprácu na vývoji Digital Contents pre BBC spolu s Institute of Education v Londýne

[8] samozrejme, nie v zmysle osobných údajov

škatuľky hodnoty pre **druhý** a **tretí kvartil** vzorky, teda pre druhú a tretiu štvrtinu žiakov v skúmanej skupine. Skutočnosti, ktoré sme takto zobrazili, zatiaľ samy o sebe nič nevysvetľujú, iba zdôrazňujú isté javy a dávajú nám dôvod hľadať možné vysvetlenia, formulovať hypotézy a zvažovať, či vieme skúmať ich pravdivosť alebo nie, prípadne – a to je pre nás tiež mimoriadne zaujímavé – či ten istý jav ďalej neskúmať inými metódami a inými výskumnými metodológiami.

Jeden z týchto javov – náročnosť súťažných úloh – skúma P. Tomcsányi, pozri [17]. V svojom príspevku analyzuje, nakoľko sa líši hodnotenie náročnosti úloh podľa nás autorov a podľa výsledkov žiakov. M. Tomcsýnyiová sa v príspevku [16] zameriava na analýzu jedného typu zadaní iBobra – **interaktívne úlohy**, charakterizuje ich a skúma, aké námety sú pre takúto formu prezentácie vhodné.

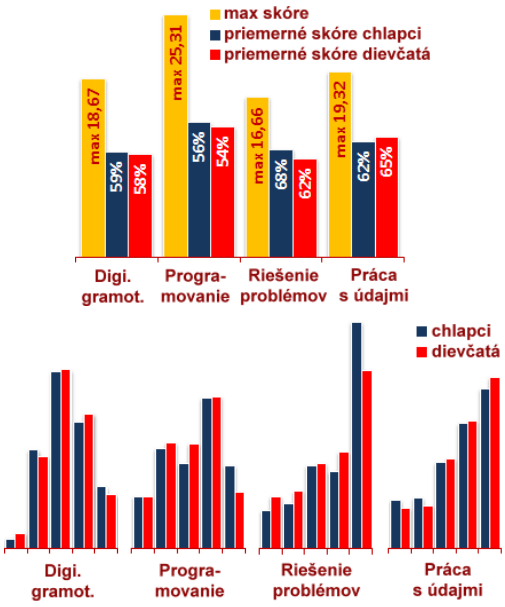
V [9] sme zasa pomocou kvantitatívnej stratégie skúmali, nakoľko sa nám na 2. stupni ZŠ (teda v kategóriách benjamín a kadet) darí využiť súťaž ako príležitosť na rozvinutie zručností žiakov pre programovanie[9] a posilnenie ich pozitívneho vzťahu k nemu. K tomu v našom článku špecifikujeme štyri komponenty informatického vzdelávania (**digitálna gramotnosť**, **programovanie**, **riešenie problémov** a **práca s údajmi**) a každú súťažnú úlohu priraďujeme k jednému, prípadne dvom z nich. Vďaka tomu môžeme porovnať, akí úspešní boli žiaci v jednotlivých komponentoch, a zvažovať, aké uzávery môžeme z toho vyvodiť, pozri obr. **3** a **4**. Uveďme niektoré z nich (podľa [9]):

- V kategórii benjamínov nás zaskočili relatívne nízke zisky v komponente digitálna gramotnosť. Domnievame sa, že tomuto komponentu zodpovedajúce témy zatiaľ nie sú dostatočne integrované do kurikula a reálneho, pravidelného vyučovania[10].
- Pomerne symetrické výsledky v oboch analyzovaných kategóriách v programovaní nás potešili, očakávali sme, že priemerné výsledky budú nižšie.

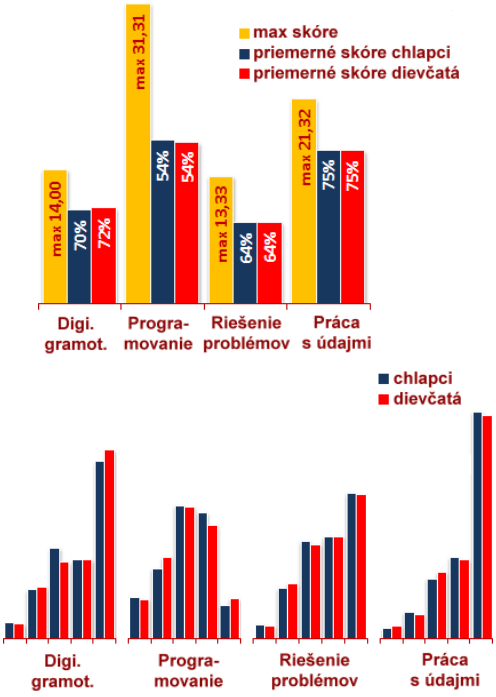
V našej analýze sme sa ďalej špeciálne zamerali na prieskum výsledkov v komponente **programovanie**. Navrhli sme kategorizáciu **kognitívnych operácií** (podobnú Bloomovej taxonómii), ktoré žiaci vykonávajú pri riešení týchto úloh, a zaujímalo nás, ktoré z nich sú pre nich menej náročné a ktoré viac. Rozhodli sme sa tiež využiť skutočnosť, že niektoré z programátorských úloh boli identické pre benjamínov aj kadetov, teda pre všetkých žiakov 2. stupňa.

[9] ide nám o rozvoj **moderného konštruktivistického programovania** – nedeformovaného rôznymi nesprávnymi koncepciami a zlými praktikami, nedegenerovaného na učenie programovacieho jazyka a nerealizovaného učiteľom, ktorý mu iba málo rozumie

[10] a možno i to, že žiaci tento komponent podceňujú a zároveň v ňom preceňujú úroveň svojich vedomostí



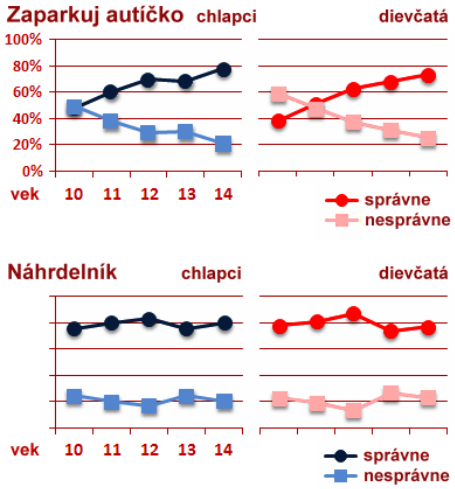
Obr. 3 Prvý graf: Maximálne a priemerné získané skóre v kategórii **benjamínov** v jednotlivých komponentoch informatického vzdelávania – u chlapcov a dievčat. Programovaniu sme v súťaži úmyselne venovali viac ako 25% bodov. **Druhý graf:** distribúcie získaných bodov u chlapcov a dievčat v štyroch komponentoch (horizontálna os predstavuje päť intervalov skóre: od 0 do 20%, od 20 do 40% atď.)



Obr. 4 Rovnako prezentované skóre podľa komponentov pre kategóriu **kadetov**

Nasledujúci obr. **5** ukazuje, ako závisela úspešnosť a neúspešnosť pri riešení spoločných programátorských úloh od veku žiakov. Horná časť obrázku potvrdzuje, že počet úspešných riešiteľov rastie

s vekom a že bodové zisky dievčat boli mierne nižšie (pri úlohe Zaparkuj autíčko).



Obr. 5 Akí úspešní boli chlapci a dievčatá v závislosti od veku pri riešení programátorských úloh rovnakých pre celý 2. stupeň ZŠ

Úloha Náhrdelník však bola trochu iná: Kadeti (t.j. vekové skupiny 13 a 14-ročných) mali riešiť *mierne náročnejší variant* zadania než benjamíni (t.j. vekové skupiny 10 až 12-ročných) – potvrdzujú to výsledky na obr. **5** dolu. Výsledky dievčat sú však v tejto úlohe **prakticky zhodné s výsledkami chlapcov**. To nezodpovedá výsledkom v úlohe Zaparkuj autíčko, preto sme prijali takúto hypotézu:

- Úspešnosť dievčat v programátorskej úlohe (vo veku benjamínov a kadetov) možno významne zvýšiť voľbou vhodnej „dievčenskej" motivácie (v úlohe Náhrdelník išlo o vyrábanie náhrdelníka z určitej krátkej opakujúcej sa vzorky z korálikov).

Na základe tohto pozorovania a tejto prijatej hypotézy sme sa rozhodli do nasledujúceho ročníka súťaže cielene zaradiť (aspoň) dve programátorské úlohy s rovnakou klasifikáciou kognitívnych operácií, ale rôznou motiváciou (pre chlapcov a pre dievčatá), a skúmať, či dokážeme potvrdiť vyššie vyslovenú hypotézu. To isté chceme urobiť aj v kategóriách pre stredoškolákov.

Uzavrime túto kapitolu dvoma poznámkami:

- **Kvantitatívna metodológia** je výborným prostriedkom pre edukačný výskum – v určitých situáciách. Musíme ju systematicky študovať a korektne používať. Musíme sa naučiť spoznať, kedy treba pri skúmaní daného javu siahnuť po inej, napr. **kvalitatívnej metodológii**.
- Príklad s úlohou Náhrdelník, pozorovanie a vyslovená hypotéza, zámer pripraviť pre ďalší ročník súťaže úlohy tak, aby sme tento jav mohli hlbšie skúmať... to všetko ilustruje, že našim postupom paralelne sledujeme dva ciele: porozumieť čo najlepšie reakciám žiakov v tejto informatickej súťaži, a zároveň iteratívne projektovať a implementovať informatickú súťaž ako atraktívnu inováciu pre moderné učenie sa. Keď čitateľ preštuduje nasledujúcu kapitolu, pochopí, prečo zdôrazňujeme tieto dve **prelínajúce** sa línie nášho výskumu súťaže iBobor.

5. VÝSKUM VÝVOJOM

Edukačný výskum má v školskom (či akomkoľvek učiacom sa) prostredí dva široké ciele: (1) **porozumieť**, ako sa žiaci učia, a (2) **navrhovať a vyvíjať** prostriedky (v literatúre sa im hovorí intervencie[11]), ktoré podporujú poznávací proces. Napĺňať oba tieto ciele – **zároveň a spolu** – je veľká výzva a príležitosť, ktorá môže znamenať významný posun pri dosahovaní oboch cieľov.

Práve na základe tohto prepojenia sa v ostatných rokoch vynára **nová paradigma pre edukačný výskum** v prostredí pre učenie sa, nazývaná *design-based research* alebo *výskum prostredníctvom vývoja* či ešte jednoduchšie *výskum vývojom* [12] (pozor, nie výskum vývoja). Je to interdisciplinárny prístup, v ktorom sa výskumníci spolu s praktickými edukátormi (učiteľmi) snažia vytvárať **presnejšie teórie učenia sa**, a to pomocou navrhovania, vytvárania, štúdia a iteratívneho vylepšovania teoreticky opodstatnených **intervencií pre učenie sa** v reálnej triede.

Východiská a koncepcia tejto paradigmy nás zaujali, pretože sa snažia priniesť riešenie tých istých metodologických problémov, ktoré konštatujeme aj my a ktoré sme už charakterizovali vyššie. Metodológia *výskumu vývojom* nás oslovuje preto, že:

- je to **empirický výskum** – teda korektnými vedeckými postupmi hľadá odpovede na otázky vyplývajúce z pozorovania reality,
- je to **edukačný** výskum – jeho cieľom je rozširovať naše porozumenie poznávacieho procesu,
- je to výskum **prostredníctvom vývoja**, teda metodológia, pri ktorej výskumník aktívne vstupuje do učiaceho sa procesu, snaží sa mu lepšie porozumieť a zároveň ho obohatiť či podporiť vývojom svojej intervencie.

Podľa [2] a [15] môžeme vysloviť **päť kľúčových princípov** *výskumu vývojom*:

- Vývoj prostredí pre učenie sa a vytváranie teórie o učení sa sa navzájom prelínajú.

---

[11] pod **intervenciou** budeme rozumieť ľubovoľný konštruktívny zásah do pedagogickej praxe (inováciu), napr. novú metodiku, študijný materiál, softvérový mikrosvet, príspevok do osnov alebo iný produkt, ktorý vzniká ako súčasť pedagogického výskumu

[12] medzi základné diela v tejto oblasti patrí napr. [2], [3] a [4], zaujímavé na čítanie sú tiež [11], [19], [21], [22] alebo [23]. Tento portál Virtuálnej doktorandskej školy siete Kaleidoscope vznikal ako spoločná aktivita kolegov z London Knowledge Lab a našej katedry

- Vývoj a výskum sa odohrávajú v opakujúcich sa cykloch vývoj ► nasadenie ► analýza ► vývoj ► ..., pozri obrázok **6**.
- Vývoj musí podporiť vznik teórie, ktorá bude užitočná pre praktikov i teoretikov.
- Výskum musí objasniť, ako fungujú vyvinuté intervencie v edukačnej praxi. Musí dokumentovať úspechy a neúspechy tohto vývoja, a tiež interakcie medzi aktérmi, ktoré zlepšujú naše chápanie poznávacieho procesu.
- Výskum musí používať metódy, ktoré vysvetľujú proces od nasadenia intervencie po pozorované javy.

Obr. 6 Schéma iterácií vo *výskume vývojom*, voľne podľa [23]

Povedzme celkom otvorene, že korene (krátkej) histórie *výskumu vývojom* siahajú do čias *pred-informatických* a *pred-IKT* [13]. Od 90-tych rokov však poznanie v oblasti vývoja softvéru obohatilo oblasť *výskumu vývojom* najmä svojou inovatívnou metodikou **iteratívneho dizajnu**. Prečo sú v tejto metodológii také dôležité práve iterácie? Dôvodov je viacero:

- dovoľujú výskumníkovi postupne spoznať, uvedomiť si a reagovať na množstvo rôznych javov (premenných), ktoré fungujú v bežnom komplexnom prostredí školskej triedy,
- dovoľujú výskumníkovi profitovať z vlastných skúseností, analýzy dát a výsledkov pozorovaní z predchádzajúcich iterácií,
- hrajú úlohy *systematických variácií* známu z klasického experimentálneho vývoja,
- dovoľujú výskumníkovi flexibilne korigovať svoje rozhodnutia v oblasti aplikovaných výskumných metód, teórií či vyvíjaných intervencií,
- dovoľujú kriticky analyzovať a hodnotiť roly ďalších zúčastnených aktérov, napr. ďalších výskumníkov, učiteľov či žiakov.

Výskum vývojom sa práve vďaka týmto vlastnostiam považuje za extrémny príklad *flexibilného edukačného výskumu*. V každej svojej iterácii obsahuje dve fázy (pozoruj obr. **6**): **návrh + vývoj + nasadenie** a **pozorovanie + analýzu**. Závery, ktoré získame v druhej fáze určitej iterácie, nám poslúžia ako program pre prvú fázu nasledujúcej iterácie. Takýto prístup odráža proces postupného vynárania sa toho, ako výskumník rozumie tomu, čo skúma a čo vyvíja.

Pri reálnych výskumných projektoch používajúcich paradigmu *výskumu vývojom* nie je ľahké plánovať, realizovať a analyzovať jednotlivé iterácie – často až dodatočne dokážeme presnejšie rozpoznať kroky a etapy, ktoré tvorili jednotlivé iterácie. Situácia sa ešte viac komplikuje tým, že často sa vo výskumnom projekte vyskytujú *iterácie rôznej mierky*: každú veľkú iteráciu tvorí niekoľko menších[14]. D. Pratt v [14] navrhuje schému dizertačného výskumu so štyrmi iteráciami s takýmto zameraním:

- **Iterácia 0 – orientačná**. Je príležitosťou na úvodný „nezáväzný“ prieskum zvolenej platformy a nástrojov, ktoré ponúka. Zvyčajne zahŕňa aj uvažovanie o predchádzajúcej práci a skúsenostiach, štúdium literatúry, myšlienkové experimenty s typickými aktérmi skúmanej oblasti (žiakmi, učiteľmi, ...), prezentovanie prototypov kolegom a diskusie o nich, a tiež pilotné pokusy s úvodnými vynárajúcimi sa predstavami o vyvíjanej intervencii s jedným či dvoma žiakmi.
- **Iterácia 1 – prieskumná**. Predstavuje iniciálny vývoj a overovanie nástrojov v zvolenej (softvérovej) platforme. Výskumník kladie dôraz na základné používanie vyvíjanej intervencie a sleduje intuitívne indikátory jej potenciálu na podporu poznávacieho procesu.
- **Iterácia 2 – vývojová**. Vyvíjaná intervencia dosiahla stav, ktorý už výskumníkovi dovoľuje zamerať svoju pozornosť na otázky učenia sa a konkrétne ciele a výskumné otázky v problémovej doméne.
- **Iterácia 3 – analytická**. Vo vývoji intervencie už výskumník robí iba minimálne úpravy, sústreďuje sa na otázky zberu dát, ich analýzy a dotvárania teórie, ktorú chce na záver svojho výskumu formulovať.

Iteratívny vývoj nemáva jasne rozpoznateľný stav ukončenia. Už z povahy skúmanej domény – teda poznávacieho procesu – vyplýva, že žiaden edukačný vývoj nemôže byť definitívny a perfektný. Na druhej strane sa však výskum musí z rôznych vonkajších dôvodov ukončiť (aspoň dočasne) a výsledky publikovať. Vo *výskume vývojom* sa niekedy používa pojem **konvergencie**, pozri [14], ktorý nám má pomôcť rozpoznať vhodný okamžik na uzavretie štvrtej iterácie. Konvergenciu v tomto kontexte chápeme ako proces zmenšovania a zjemňovania krokov vo vývoji, čoraz jemnejšie úpravy,

[13] medzi dôležité východiská sa zahŕňajú o.i. aj práce H.A.Simona, svetoznámeho ekonóma, informatika, odborníka v umelej inteligencii a kognitívneho psychológa

[14] o podobnom jave sme hovorili v roku 2008, v prvej časti tohto príspevku, v kapitole *9 Nová metodika vývoja edukačného výskumu*, pozri tiež [13] a v [10] aj poznámku pod čiarou číslo 21

ktoré vyplývajú z toho, že výskumník lepšie a hlbšie rozumie skúmaným javom (premenným), vzťahom a procesu učenia sa, ktorý sprevádza jeho prácu. Štvrtú iteráciu treba ukončiť vtedy, kedy kroky vo vývoji „konvergujú“ už len do drobných vylepšovaní (napr. aj vzhľadom na momentálny stav výskumníkovho poznania a jeho vývojárskych schopností).

Všetky iterácie – každú nasledujúcu vždy viac a viac – dopĺňa (i) **zber** a (ii) **analýza dát**. Starostlivý a precízny zber dát je pre výskumníka vždy veľkou výzvou. Rovnako, ako pri iných metodológiách edukačného výskumu, aj pri *výskume vývojom* musíme dbať o validitu a reliabilitu a využívať metódu triangulácie na neustálu kontrolu kvality výskumu. Medzi postupy na zber dát sa v tejto paradigme často používajú:

- písomné zhodnotenie stretnutia (pokusu, aktivity) samotnými aktérmi,
- rozhovory pred a po stretnutí,
- prípadové štúdie,
- video či audio nahrávky stretnutí,
- rozhovory *pri riešení problému*, kedy žiaka konfrontujeme s problémom a nabádame ho, aby *rozmýšľal nahlas*, teda aby komentoval svoj postup už počas riešenia problému,
- sondy do procesu konštruovania poznania – sú to krátke rozhovory, ktorými vstupujeme do procesu poznávania s cieľom dokumentovať jeho *mikro etapy*,
- zber produktov – textov, riešení či akýchkoľvek výtvorov, napr. aj žiackych programov v rôznych štádiách alebo ľubovoľných iných reprezentácií procesov, ktoré vytvárajú,
- terénne zápisky zo stretnutí, z diskusií, z riešenia problémov a pod.

Uvedomme si však, že vo *výskume vývojom* neexistuje tradičné pozorovanie – v tejto metodológii nepozorujeme realitu, ktorá existuje bez našej intervencie.

Získané dáta analyzujeme po každej iterácii (i počas nej), pretože práve takto získané pozorovania a poznatky nám slúžia pri projektovaní nasledujúcich krokov vo výskume a vývoji.

Aby sme zhrnuli úvodnú informáciu o novej paradigme edukačného výskumu, sumarizujme ešte raz päť základných charakteristík tejto metodológie Podľa autorov [19] je *výskum vývojom*:

- *pragmatický*, pretože jeho cieľom je riešiť reálne problémy, a to vývojom a nasadením intervencií pre lepšie učenie sa,
- *zameraný na teóriu*, pretože hľadá teóriu, ktorá lepšie ako doposiaľ vysvetlí toto učenie sa,
- *interaktívny, iteratívny a flexibilný*, pretože vyžaduje intenzívne interakcie medzi výskumníkmi a praktikmi, využíva iteratívny dizajn a svojim rekurzívnym výskumným procesom poskytuje vyššiu flexibilitu ako tradičné experimentálne prístupy,
- *integrujúci*, pretože integruje rôzne metódy kvantitatívneho i kvalitatívneho výskumu,
- *kontextový*, pretože jeho výsledky vyplývajú z kontextu, teda z prostredia, kde výskum vykonávame.

Na záver tejto časti by sme radi prezentovali ešte jednu – nie kľúčovú, ale pre našu výskumnú oblasť významnú informáciu. Medzi „otcami zakladateľmi“ tejto novej výskumnej paradigmy nachádzame o.i. aj významných didaktikov informatiky a expertov v oblasti TEL. Spomeňme napr. A. diSessu (z Berkeley), autora zásadnej monografie o korytnačej geometrii [1]; Y. Kafai (UCLA), autorku významných konštrukcionistických prác z oblasti vývoja hier pre deti samotnými deťmi, pozri napr. [7] a [8]; ďalej R. Nossa a D. Pratta (oboch z London Knowledge Lab), našich spoluriešiteľov z projektov MATCh, Playground a Kaleidoscope NoE, pozri [20], [24] a [25].

Aby sa *výskum vývojom* stal širšie akceptovanou výskumnou metodológiou, jeho používatelia sa musia v blízkej budúcnosti dopracovať k relatívne ustálenej (zavedenej) palete postupov a výskumných metód – iba tak budú môcť výskumníci zdieľať skúsenosti a kumulovať poznanie spôsobom, ktorý prospieva aj teórii, aj praxi.

6. MEMORANDUM VÝSKUMNÍKA (v TEÓRII VYUČOVANIA INFORMATIKY či TEL)

Prečo má edukačný výskum takú príšernú reputáciu?
C.F.Keastle, 1993 (Educational Researcher, 22 (1))

Ak usudzujeme na základe rôznych vedeckých aktivít, napr. aj podľa dizertačných prác obhajovaných v našej výskumnej oblasti na Slovensku v ostatných rokoch, môžeme konštatovať, že kvalita výskumu sa výrazne zlepšuje, zvyšuje sa záujem o korektné výskumné postupy, o aplikovanie korektných metodológií, o zber výpovedných dát a ich systematickú analýzu, pozri napr. [12].

Naša oblasť sa – síce pomaly, ale predsa – dostáva na úroveň, kedy dokáže kriticky reflektovať sama seba, hodnotiť korektnosť svojich postupov, uvažovať o vhodnosti rôznych metodológií. (Veď práve preto vytrvalo hľadáme nové výskumné paradigmy, ktoré by našej oblasti lepšie vyhovovali, a dúfame, že vo *výskume vývojom* sme nateraz takú našli.)

Nanešťastie musíme konštatovať aj to, že (i) na Slovensku je o systematický edukačný výskum zo strany štátu iba nepatrný záujem a skoro nikto sa mu profesionálne nevenuje, (ii) aj v zahraničí je terčom serióznej kritiky, a to predovšetkým z vlastných radov.

Podľa profesora Andersona z Athabasca University môžeme za dôvody zlej reputácie edukačného výskumu považovať:

- to, že sa mu venuje veľmi málo odborníkov,
- prakticky nulový záujem zo strany potenciálnych odberateľov, napr. štátu,
- prevládajúci nedostatok dôvery a úcty k jeho postupom a získaným výsledkom,
- žiadnu koncentráciu na **veľké témy** a problémy,
- teda sumárne: nedostatok kultúry a skúseností v tejto oblasti.

S týmto stavom sa nemôžeme zmieriť, musíme hľadať cesty na jeho nápravu – jednou z nich je **robiť seriózny výskum**, dať si za cieľ získavať seriózne, reálne použiteľné a pre našu prax dôležité poznatky a teórie. Jeden z možných námetov na takýto významný výskum, na takúto veľkú tému, sme spomenuli aj v [10] – výskum úlohy programovania vo všeobecnom vzdelaní. Ak sa chceme pokúsiť zvrátiť reálne nebezpečenstvo, že *programovanie ako kľúčová súčasť všeobecno-vzdelávacieho predmetu informatika zanikne*, musíme sa urýchlene posnažiť aj v rovine špičkového výskumu – musíme čo najskôr priniesť pozorovania, argumenty, závery a teórie, ktoré potvrdia[15] význam kompetencií súvisiacich s programovaním pre moderné všeobecné vzdelanie.

Preto sa chceme v tejto kapitole pokúsiť o tak trochu heraklovskú úlohu – načrtnúť niekoľko zásad, ktorými by sa mal výskumník (v oblasti TEL) riadiť.

Obr. 7 Herakles zápasí s hydrou o osud programovania vo všeobecnom vzdelaní. Akú výskumnú metodológiu použije pri hľadaní pádnych argumentov?

Uvedomujeme si, že niektoré z týchto zásad sú všeobecne známe a platné aj v iných oblastiach výskumu. Myslíme si však, že TEL (alebo zúžene teória vyučovania informatiky) má určité špecifiká, ktoré si musíme uvedomiť, naučiť sa s nimi existovať a reagovať na ne, prípadne z nich i profitovať.

Budeme sa snažiť vyhýbať úvahám, ktoré sú skôr *politické*, a sústredíme sa na úvahy *vecné a odborné* – ale hneď prvé dve zásady sú na hrane medzi odbornými úvahami a politikou:

▪ Viac odborníkov do našej oblasti

Azda najvážnejším problémom je pre nás výrazný a **dlhodobý nedostatok odborníkov**, ktorí sa profesionálne venujú výskumu vo vyučovaní informatiky a v informatizácii, napr. informatizácii regionálneho školstva. Keď porovnáme spoločenský význam tohto fenoménu pre vzdelávanie a počet pracovníkov/pracovísk, ktorí sa mu profesionálne a na plný úväzok venujú, uvedomíme si veľký nepomer a deficit[16].

Kto vlastne takýto výskum robí? Sú to skoro výlučne len doktorandi na katedrách, ktorí sa vlastne skúmať iba učia. Doktorandský výskum by mal zvyčajne byť súčasťou výskumu školiteľa či pracoviska.

▪ Viac peňazí, viac úcty, viac sebaúcty

Výskum sa nemôže skvalitňovať a rozrastať, ak nemá ako získať dostatok grantov a grantových prostriedkov. Príležitosti pre získanie výskumného grantu v oblasti TEL sú na Slovensku veľmi malé, len ťažko sa dá uveriť, že výsledky takéhoto výskumu niekto naozaj chce a **potrebuje**.

S tým však súvisí aj nedostatok našej vlastnej úcty ku kvalitnej výskumnej práci v TEL. Často počúvame ironické poznámky o dôveryhodnosti štatistických metód, hanlivé poznámky o kvalitatívnych metódach (a žiadne poznámky o iných výskumných metodológiách).

Ďalšie zásady už budú čisto odborné a vecné.

▪ Profitujme z iných odborov

S predmetom nášho výskumu sa nachádzame sme v strede medzi informatikou a spoločenskými vedami. To prináša rôzne problémy, ale aj veľké výhody a výzvy. TEL cielene prezentuje svoju multidisciplinárnu povahu. Potrebujeme spolupracovať s didaktikmi, metodikmi, štatistikmi, psychológmi, sociológmi, praktikmi zo škôl...

Odlišujeme sa niečím od didaktík iných predmetov? Domnievame sa, že áno. Naučili sme sa však už dosť od didaktikov matematiky? Domnievame sa, že nie.

[15] pozor: alebo NEPOTVRDIA! Ak hovoríme o serióznom výskume, nesmieme sa vopred naladiť na tú „správnu“ odpoveď. Obvinili by nás z predpojatosti a neobjektívnosti

[16] máme na mysli napr. aj minimálny počet odborníkov na informatiku, didaktiku informatiky a informatizáciu na ŠPÚ, na ŠIOV a na MŠ SR

Ešte raz spomeňme naliehavý problém významu programovania pre všeobecné vzdelanie. Ako pri takomto grandióznom (širokom, komplexnom a hlbokom) námete na výskum postupovať? Spomeňme dve riziká súvisiace s veľkosťou výskumného problému:

▪ Pozor na výskum širokých problémov

Skúmanie veľmi širokých tém je detská nemoc každej mladej disciplíny – takýmto prístupom sa o širokom probléme dozvieme veľmi málo (v lepšom prípade tak vzniká poznanie *typu Ementál*).

Na druhej strane však, rozumne úzke a kvalitné výskumné projekty nemôžu vznikať *na zelenej lúke*. Obhájené dizertačné práce na Slovensku potvrdzujú posun správnym smerom.

▪ Pozor na výskum triviálnych problémov

Opakom širokých problémov nemusia nutne byť iba vhodne úzke výskumné otázky, môžu nimi byť aj **otázky triviálne**, na ktoré vieme správne odpovede *uhádnuť intuíciou*.

Pri odhadovaní toho, čo je triviálne, však musíme byť opatrní. Ak na základe dát o súťaži iBobor môžeme usúdiť, že v programátorských úlohách skórujú lepšie chlapci než dievčatá, je to triviálny výsledok? Vieme ho vysvetliť? Potvrdia ho aj nasledujúce ročníky? Ale má tento pozorovaný jav iné príčiny? Ak je to pre niekoho očakávaný výsledok, prečo u benjamínov neplatí?

▪ Posúvajme (alebo prekladajme) hranicu poznania

Dobrým kritériom kvality výskumnej práce je, ak ju celá komunita odboru vníma ako posun svojho poznania. Výskum, ktorý tak nerobí, nemá význam.

▪ Buďme na seba nároční – každý ďalší výskumný projekt musí byť *lepší*

Okrem predchádzajúceho kritéria je dôležité aj to, aby sme každým ďalším výskumným projektom posúvali vpred aj sami seba, aby sme zlepšovali svoje postupy a obohacovali svoje skúsenosti.

▪ Učme sa

Najviac sa musí v danej oblasti vzdelávať ten, kto v nej vie najviac. **Čítajme** (a kriticky reflektujme) práce našich kolegov – blízkych i vzdialených.

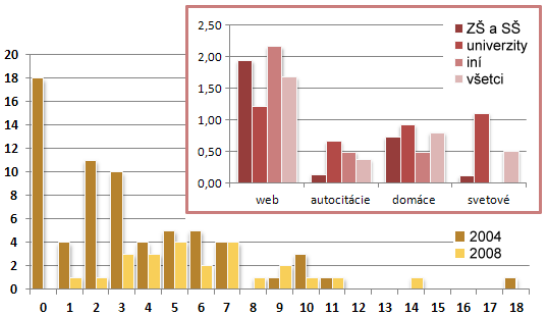
▪ Publikujme

Ak chceme dostávať, musíme dávať. Každý výsledok svojho nového poznania odovzdajme ďalším. Ak sa my chceme (a musíme) učiť od druhých, aj my ich musíme učiť, musíme ich obohacovať o naše nové poznanie – nech ide o akýkoľvek „malý“ nový poznatok.

▪ Citujme

Ak nečítame, nemôžeme robiť kvalitný výskum. Ak čítame, citujme[17]! Čítajme – a citujme kvalitné overené svetové zdroje, kvalitné domáce výsledky a práce svojich kolegov z odboru.

Obr. **8** ukazuje priemerné počty položiek v zozname použitej literatúry (aj podľa rôznych typov) z náhodne vybranej vzorky slovenských zborníkov z nášho odboru za obdobie 2004 až 2008. V dolnej časti vyjadruje horizontálna os počet položiek a vertikálna os počet príspevkov v našej vzorke s takýmto počtom položiek. Vidíme napr., že zo 64 príspevkov z roku 2004 až 18 neobsahuje jedinú referenciu. Typy a priemerné počty referencií v týchto 64 príspevkoch vidíme v hornej časti obrázka. Trend od roku 2004 po 2008 je však významne pozitívny a ukazuje na zlepšovanie citačnej kultúry.



Obr. 8 Malý prieskum počtu referencií v slovenských publikáciách

▪ (čiastočne) Odolávajme módnym trendom

Bohužiaľ, aj vo vede existujú „módne trendy“, podľa ktorých sa riadia napr. aj zahraničné grantové agentúry. Preto sa žiadatelia o granty často snažia vyrukovať s novými a honosnými pojmami – s nádejou, že tie predchádzajúce už nie sú dosť atraktívne. Žiadať napr. DG pre Informatiku (EC DIGIT) o grant, v názve ktorého sa vyskytne slovo e-Learning, je zbytočné. e-Learning je out.

▪ Skúmajme výskum

O výskumných stratégiách sa musíme učiť, musíme ich čoraz korektnejšie a kvalitnejšie používať – a v rámci našej komunity musíme o nich aj písať a bádať. Mali by sme dosiahnuť stav, kedy učebné texty či iné študijné zdroje o výskumných

[17] a nie nutne iba sami seba ☺

stratégiách pre budúcich vedcoch v našom odbore (napr. pre doktorandov) budú stáť na príkladoch našich vlastných realizovaných výskumov, s ich úspechmi i problémami.

7. ZÁVER

Cieľom druhej časti tejto mini série bolo prispieť k rastu úrovne vedeckej práce v oblasti TEL, posmeliť mladých odborníkov do ambicióznej výskumnej práce, ukázať niekoľko metodologických problémov, na ktoré v našej projektoch narážame, a predstaviť novú a pre nás mimoriadne zaujímavú metodológiu *výskumu vývojom*.

LITERATÚRA

[1] Abelson, H., diSessa, A.A.: Turtle Geometry. The Computer as a Medium for Exploring Mathematics. The MIT Press, 512 p., 1986

[2] Cobb, P., Confrey, J., diSessa, A., Lehrer, R. and Schaube, L.: Design Experiments in Education Research. *The Educational Researcher*, 32 (1), pp. 9-13, 2003

[3] diSessa, A. A.: Local sciences: Viewing the design of human-computer systems as cognitive science. In J. M. Carroll (Ed.): Designing Interaction: Psychology at the Human-Computer Interface. NY: Cambridge University Press, pp. 162-202, 1991

[4] diSessa, A. A., Cobb, P.: Ontological innovation and the role of theory in design experiments. *Journal of the Learning Sciences*, 13(1), pp. 77-103, 2004

[5] Hrušecká, A., Pekárová, J., Tomcsányi, P., Tomcsányiová, M.: Informatický bobor – nová súťaž v informačných technológiách pre žiakov základných a stredných škôl, zborník konferencie Didinfo, UMB Banská Bystrica, Fakulta prírodných vied, 2008

[6] Chráska, M.: Metody pedagogického výskumu. Grada 2007, 265 str.

[7] Kafai, Y. B. (1998). Video game designs by children: Consistency and variability of gender differences. In J. Cassell & H. Jenkins (Eds.), From Barbie to Mortal Kombat: Gender and computer games (pp. 90-114). Cambridge, MA: MIT Press.

[8] Kafai, Y. B. (2006). Playing and making games for learning: Instructionist and constructionist perspectives for game studies. *Games and Culture,* 1(1), 36-40.

[9] Kalaš, I., Tomcsányiová, M.: Students' Attitude to Programming in Modern Informatics, paper accepted for IFIP WCCE, Bento Gonçalves, 2009

[10] Kalaš, I.: Pedagogický výskum v informatike a informatizácii (1. časť), zborník konferencie Didinfo, UMB Banská Bystrica, Fakulta prírodných vied, 2008

[11] Mor, Y. and Winters, N.: Design approaches in technology enhanced learning. *Interactive Learning Environments 15(1)* , pp. 61-75, 2007

[12] Palmárová, V.: Tvorba a využívanie elektronických učebných pomôcok na podporu vyučovania programovania. Dizertačná práca, FPV UKF v Nitre, 2009

[13] Petráš, A.: Proces vývoja edukačného softvéru. Dizertačná práca, FMFI UK v Bratislave, 2008

[14] Pratt, D.: The Construction of Meanings in and for a Stochastic Domain of Abstraction. Dizertačná práca, University of London, Institute of Education, 1998 (prácu možno získať na www.ioe.ac.uk/gems/people/ dave_pratt)

[15] The Design-Based Research Collective: Design-Based Research: An Empirical Paradigm for Educational Inuiry. *The Educational Researcher*, 32 (1), pp. 5-8, 2003

[16] Tomcsányiová, M.: Interaktívne úlohy v súťaži informatický Bobor, zborník konferencie Didinfo, UMB Banská Bystrica, Fakulta prírodných vied, 2009

[17] Tomcsányi, P.: Náročnosť úloh v súťaži informatický Bobor, zborník konferencie Didinfo, UMB Banská Bystrica, Fakulta prírodných vied, 2009

[18] Zounek, J.: ICT v životě základní školy. Triton 2006, 152 str.

[19] Wang, F. and Hannafin, M. J.: Design-based research and technology-enhanced learning environments. *Educational Technology Research and Development*, 53 (4), pp. 5-23, 2005

ODKAZY NA WEBOVÉ STRÁNKY

[20] comlogo.web.elte.hu/team/match – stránka projektu MATCh na ELTE

[21] www.designbasedresearch.org – spoločnosť pre DBR na University of California Berkeley

[22] projects.coe.uga.edu/dbr – zaujímavý tutoriál o DBR, University of Georgia

[23] www.lkl.ac.uk/projects/designresearch – centrum VDS (Virtual Doctoral School) projektu Kaleidoscope o metodológii DBR

[24] www.ioe.ac.uk/playground – stránka projektu Playground

[25] www.noe-kaleidoscope.org – portál siete Kaleidoscope

[26] matchsz.inf.elte.hu/Colabs – portál projektu CoLabs

THE PEDAGOGY OF PROGRAM DESIGN

VIERA KRŇANOVÁ PROULX, PHD.

College of Computer and Information Science, Northeastern University,
360 Huntington Ave, Boston, MA, USA, tel. ++1-617-3732225, e-mail: vkp@ccs.neu.edu

ABSTRACT

We describe the pedagogy, the curriculum, and the software support for teaching systematic program design with emphasis on systematic testing and on understanding the connection between information and data. The curriculum begins with the BOOTSTRAP curriculum for children in grades 5-8, follows with the TeachScheme! segment for secondary schools and universities and extends to ReachJava, a full-scale object–oriented program design starting at the secondary level and extending to the software design at the university level.

The software support enables students to work with a language appropriate for their current knowledge of programming, it provides a framework for the design of interactive games that encourages creativity, and it supports the design and evaluation of tests appropriate for a novice programmer. The curriculum has been used in classrooms for a number of years and nearly all materials (software, the text, lab materials, worksheets, assignments) are available free on the web.

Keywords*: informatics, program design, object-oriented design, testing, games*

INTRODUCTION

Over the past fifteen years our team has been designing curriculum for teaching introductory programming and computing for students of all ages. The original curriculum labelled TeachScheme! has been used in secondary schools and universities throughout the world. Since I joined the team seven years ago I have worked on extending the curriculum to include program design in the object-oriented style, building a path to full scale software design curriculum. At the same time, other tem members designed a curriculum for children in grades 6–8 that combines the learning of computing and the basic algebra in the context of designing of a simple interactive game.

Our curriculum provides a path for learning that starts at the most introductory level, but leads to understanding deep concepts of program design at a very advanced level. The backbone of the curriculum at all levels is the pedagogy for describing the design process as a collection of **design recipes**.

This paper provides an overview of how the pedagogy of design recipes helps both the student and the teacher in all stages of learning.

1. DESIGN RECIPES

The *design recipe* is a series of steps a programmer should follow when designing a program, regardless of the expected complexity of the program. The recipe consists of several steps. At each step the programmer explores the next problem step by asking and answering a series of questions, producing an answer, and verifying the correctness of the answer. The pedagogical benefit to the learner is in the self-regulatory learning style the design recipe supports: the student is empowered to explore and solve the problem on her own, knowing how to proceed at each step. The benefit to the instructor is in the support for pedagogical intervention. When a student seeks help with a problem, the instructor can determine what step of the design recipe caused the difficulty, and lead the student through working out that part of the design recipe.

1.1. Design Recipe for Designing Functions

We start the programming instruction by designing functions. These are functions as seen in mathematics: each takes one or more arguments and produces a new value. Rather than changing the state of variables, our programs produce new values at each step. This type of programming (functional programming) has the advantage that the outcome of each function depends only on the values of the arguments, not its position within the program. It is easy to define the expected behavior of such functions. The *design recipe for functions* takes advantage of this functional behavior:

1. Analyze the problem, represent the relevant information as data.
2. Write down the purpose statement for the function, identifying clearly what data it consumes and what value it should produce. Write this information formally as a contract, then design the function header.
3. Make examples of the data the function will consume, write down the expected results. Do as many of these as needed to understand the problem.
4. Take an inventory of all available data and functions already defined that could be used in defining this function. This includes fields of complex data and functions that consume data of the type of data available to us as the function arguments.

5. Now design the body of the function. If the task looks to be too complex, divide it into smaller parts and make a *wish list* of functions you will need to solve the problem. (These functions will be designed later, following the same steps. For now we assume they have been defined already.)
6. Convert the examples from part 3 into formal tests and run them. Fix any errors and test again.

1.2. Design Recipe for Data Definitions

Without a solid understanding how the relevant information is represented as formal data students cannot design programs. A great deal of our curriculum focuses on understanding how information can be represented as data and how a piece of data can be interpreted to describe the information encoded within.

The *design recipe for defining data* guides the student through the process as follows:

1. Check if the information can be represented by one simple piece of data, such as a number, a symbol, or a String.
2. If several related pieces of information are needed to describe one object, combine them into a structure (later these become classes). Define a composite structure with one field per piece of information. (For example, a book with a title represented as a String, and a year of publication given as a number.)
3. If the information refers to another complex object, use containment. (This will be a structure or a class with a field that is an instance of another class: A book with a title, author (with name and year of birth), and a year of publication.)
4. If the information consists of several related object with some common features but with several variants that distinguish between them define a union (An shape is one of: a circle, a rectangle, a triangle). The define each member of the union separately. (In Java these turn into classes that implement a common interface or extend an abstract class.)
5. Connect fields with their data definitions as appropriate drawing containment and inheritance arrows.

1.3. Design Recipe for Designing Abstractions

Once the programs become sufficiently complex students begin to see that certain sections of code appear again and again, perhaps with a slight variation.

This motivates the design of abstractions. A list of books, or a list of persons is just a list of objects. When we sort data, we need to compare the two items and need a function that performs the comparison. Given the comparison function, the same sorting program works for any data.

The design recipe for abstractions provides the following guideline:

1. Compare two pieces of code that are similar, highlight their differences.
2. Replace each pair of differences with a parameter and rewrite the general solution.
3. Implement the solution to each of the original programs and run the tests.

2. COMPUTATION AND ALGEBRA

Our middle school curriculum is taught in the after-school programs by volunteer teachers. The class meets for 90 minutes for 10 weeks. Most of the students are weak in math. This is a very short time, yet the pupils design and implement an interactive game and on the final day not only show off their game, but talk about variables, conditionals, and function evaluation.

The goal is to design a game with one object moving as the time ticks and another object controlled through the arrow keys pressed by the game player. We restrict the movement of every object to one dimension (left-right or up-down). Collision of the two objects produces a special effect: the game may end, an explosion is shown, or the game score may be updated.

Children program the movement of the game components by defining functions that compute the position after one time tick, or the new position after a key has been pressed. They detect collision of two objects by computing the distance between them (we give them the formula for this). They learn what is a variable, how to evaluate expressions, learn a bit of geometry by defining the positions of their game components, learn about boolean values when detecting collisions or making sure the game pieces stay within the game canvas.

2.4. The Curriculum

The curriculum has been designed with two goals in mind. It has to speak to the child in his language, making sure the children are not distracted or overwhelmed by the programming language complexity. It also has to be divided into daily lessons that can be taught by volunteer teachers with a minimal training.

We use a special variant of the Scheme programming language. All values are color-coded, so that the children can easily distinguish between Strings, booleans, numbers, and symbols.

To explain the evaluation of expressions and functions we enclose each subexpression in a *CIRCLE OF EVALUATION*. The operation (plus sign, or a name of a function) is on the top and the values that are needed for the computation are

shown below in the appropriate order. This translates directly into Scheme functions. that are evaluated immediately in the *Interactions* window of the *DrScheme* programming environment:

> (- 9 5)

4

More complex expressions are drawn as circles within circles and children learn to evaluate the innermost circles first.

Children quickly learn to convert an expression such as (+ (* 2 3) (- 8 6)) to the collection of circles of evaluation and compute the value of the expression. The variables are then shown with the name above and the substituted value below a horizontal line.

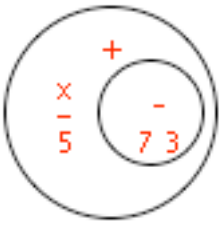


Fig. 1 Circles of Evaluation

Figure 1 shows the circle of evaluation for the expression *(x + (7 + 3))*, with *5* substituted for *x*. In Scheme this would be written as *(+ (x (- 7 3))*.

Students are motivated, as the goal is to design their own game. Children brainstorm about the game they want to design during the first lesson and the teacher finds appropriate images for every student's game by searching the clipart on the web. The supporting software library allows students to draw any image at the selected location. Children define the function onKeyEvent that produces the game scene after the player has hit the left or right arrow key, the function onTick that produces the game scene after one clock tick, and the function that detects the collision of the two objects.

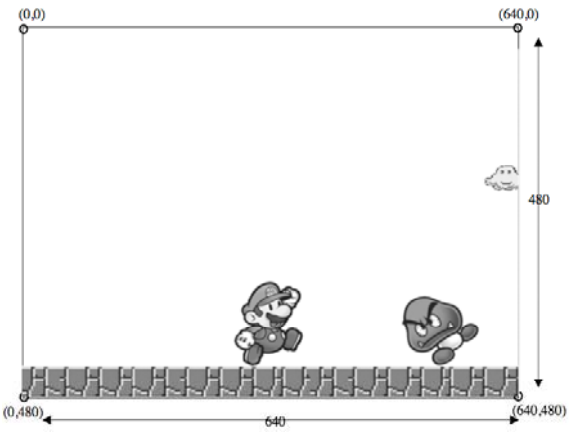


Fig. 2 A worksheet for Mario game

Children learn about conditionals, use the distance formula that we provide, and test their functions one by one as they design them. They follow the design recipe with worksheets where each step is carefully worked out and tested as a running program. The last day of the session they show their games to their parents, teachers, and friends, and answer questions about conditionals, functions, function evaluation, and their game design. We have seen some of these children succeed in math in their later years, but the number of students who have completed our program is too small to provide a statistically significant evidence of success.

3. UNDERSTANDING DATA

In the program design courses in the secondary schools and at the college level the first one or two weeks are similar to those of the younger children. However, to design programs of any complexity one must deal with sufficiently complex data. While the typical language-based programming instruction focuses on *algorithmics*, our curriculum is based on the premise that most of the production programs are driven by the structure of the data that they process. The art of defining data that represents the given information is the driving force behind program design.

3.5. Designing Data

The *design recipe* for data definitions allows us to define an ancestor tree as follows:

An Ancestor Tree (*AT*) is one of
- 'unknown
- Node : (make-node String *AT AT*)
(define-struct node (name mother father))

The *struct* definition allows us to build an instance of a complex data, provides a constructor (*make-node "Jan" mom dad)*, a predicate (*node? x)* to verify that a data item *x* is a *node* and a selector function for every field: *node-name, node-mother, node-father*.

For each new data definition we make sure students make examples of data and can interpret the data representation as information, as well as define new data items that represent the given information. For this example, students would convert their ancestor tree into data and read their friend's data and answer questions about the maternal grandfather, great grandmother, etc.

3.6. Data Driven Design of Functions

The first step in designing function is to identify the information that is available as well as the desired outcome, and understand how it will be represented as data.

In the second step students define the function header, a purpose statement that explains what the function will accomplish, and define the types of data the function consumes as arguments as well as the type of the result it will produce. We do not use assignment in the early weeks of the curriculum and every function produces a new value. This restriction greatly simplifies the program students write and allows us to reason about each function independently of its place in the program.

Before attempting to design the actual function, **the third step** of the *design recipe* asks that students make examples of the data that the function consumes, and explain the desired behavior by showing sample function invocations together with the expected outcomes.

To help define a function for more complex types of data **the fourth step** of the *design recipe* asks the students to decompose the problem into parts by designing an inventory. An inventory for the function that consumes complex arguments lists the fields of the function arguments as well as the functions that can be used with them (from earlier definitions). The inventory for a function that consumes an *AT* would be:

```
(define (fun at)
  (cond [(symbol? at) …]
        [(node? at)
            … (node-name at) …
            … (node-mother at) …
            … (node-father at) …
            … (fun (node-mother at))…
            … (fun(node-father at) )…  ]
```

If this function was counting the nodes in the tree (other than unknown) it would be clear that the two function applications to the mother and father subtrees will produce the count of known relatives in each ancestor subtree, and the rest of the computation trivially follows.

Only now, in **the fifth step** of the *design recipe*, do the students proceed with the design of the function body. If the task looks to be too complex, it is divided into smaller parts, with subtasks delegated to other functions, making up a *wish list*. Each function on the *wish list* must have the function header, contract, and purpose statement, so that it can be used in completing the original function design. Of course, later on we need to take care of the functions on the *wish list*.

Once the function has been designed, **the sixth step** of the *design recipe* directs the students to turn the examples defined earlier into test cases that are run and evaluated. Because every function produces a new value and has no side-effects, the tests only need to compare the expected value with the value produced by the function. The test design is simple. It is supported by a test harness that allows us to define the tests simply as:

```
(check-expect (count-nodes my-at)  5)
```

The test harness then reports which tests failed and shows the actual and expected values side-by-side.

In the early curriculum the structure of the functions students design follows exactly the structure of the data the function consumes. A large number of everyday programs are of this type: parsers, interpreters, programs that manipulate databases, GUI interactions programs, and more. We introduce algorithms that require some additional insight (e.g. quicksort) only after the students understand the program design driven by the structure of data.

3.7. Designing Abstractions

In order to build reusable programs we need to generalize: we need to abstract over the data type, over the functional behavior, the data structure manipulation, or the traversals over the data in a collection of data.

In both the early curriculum that uses *DrScheme* functional languages, and the early part of the class-based instruction we follow by introducing abstractions that simplify the program design and allow us to make the programs more general and reusable as libraries. In the Scheme-like languages we introduce the use of functions as arguments and the generalized Scheme loops such as *orMap, filter,* and *fold.*

We motivate abstractions by showing side-by-side two programs that are very similar, highlighting the places where they differ, and replacing the differences with parameters. So, for example a program that computes the sum of all numbers in a list and a program that computes the product look structurally the same, but differ in the base value and in the operator used to accumulate the resulting value. That means that the base value and the operator (a function of two arguments) become parameters for the generalized solution. The resulting general program can be then used to concatenate a list of Strings into one String, with the base value given as an empty String and the operator being the concatenation function.

To motivate the use of a function as an argument to another function we show how a sort function depends on the comparison function that determines the ordering of two items. Therefore, the comparison function becomes an argument to the sort function.

The design of abstractions depends in a substantial way on the language support for the abstractions we wish to discuss. On the other hand, the desire to design more general programs allows us to motivate specific language features that have been included in the language precisely to support such abstractions.

3.8. Curriculum Support

The *TeachScheme!* curriculum, supported by the *DrScheme* teaching languages and the text *How to Design Programs*, is used in over 700 high schools and universities both in the USA and in many countries throughout the world. The supporting software includes libraries for the design of interactive graphics-based games as well as the library for designing multi-player client games managed by a server and played over the internet.

Our students design several games during their first year. However, the functions that comprise the game are all designed according to the *design recipe* and are fully tested.

4. OBJECT-ORIENTED PROGRAM DESIGN

4.1. Understanding Data

The curriculum for our second course is known as *ReachJava*. A draft of the textbook *How to Design Classes* is nearly complete.

This course focuses on program design in a class-based language with object-oriented programming. Students already have a good understanding of basic functional program design. The first lectures of the course focus on designing classes. The *struct* definitions from the Scheme turn into class definitions. The unions of data defined in Scheme become classes that implement a common *interface*. The *struct* definitions that contain fields that are also *struct*s become classes with fields that are objects of another class.

We represent the class hierarchy as class diagrams using a simplified UML notation. Every class includes a full constructor. Every field is referenced as *this.xxx*. Students make examples of data and get to see that the data representation is in many ways similar to what they already know.

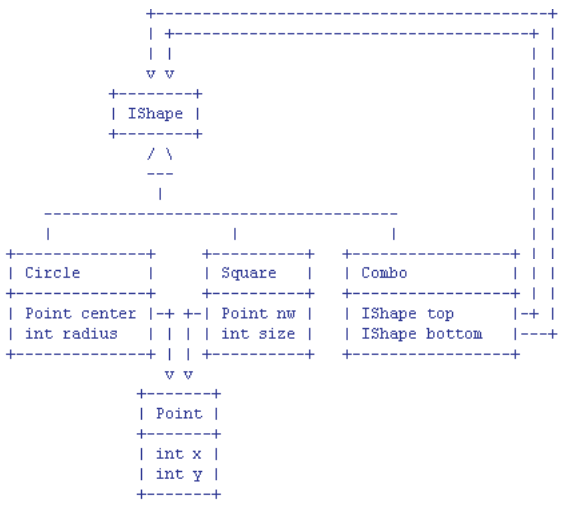


Fig. 3 Sample class diagram

Students define an *Examples* class (with the default constructor) and write all examples of data there. Before going any further, they must be able to translate the given information into its representation as data, and be able to interpret the data as the information the data represents. The *Examples* class acts as a client to student's code.

The main new idea in designing methods for the object-oriented languages is that each method belongs to a specific class and the object that invokes the method is somewhat *invisible* in the method definition. We require that students formulate the method purpose statement by referring to *this* object and the *given* arguments, so that the role of all data the method consumes becomes clear. Within the method definition we reference all fields of the current class with the *this.* prefix:

```
// compute the price of this book
// with the given discount
double salePrice(double discount){
  return this.price * discount;
}
```

Again, we follow the same *design recipe*. The examples of method invocation make it quite clear how the method invocation works and what is the role of the object that invokes the method. For the first couple of weeks we still follow the mutation-free style of programming where the outcome of the method is always a new object or a primitive value.

The inventory part of the design recipe now lists all the fields in *this* class as well as all arguments to the method, any methods already defined for this class, and any methods that can be invoked by any of the fields already listed:

So, in the class *Node* that implements the *AT* interface we would have:

```
// does this AT contain a person
// with the given name?
boolean contains(String name){

/* Inventory:
 Fields:
… this.name …     -- String
… this.mother …  -- AT
… this.father …    -- AT

Methods:
… this.count() …   -- int

Methods for fields:
… this.mother.count() …  -- int
… this.father.count() …    -- int

… this.name.equals(String) … -- boolean
*/
}
```

We can see that there is a great emphasis on making sure students understand very clearly the meaning of data and its structure. Combined with the examples of data they defined earlier the design of the actual method body becomes straightforward.

4.2. Test-First Design

The third step in the *design recipe* asks students to make examples of the method invocation with the expected results. In the absence of mutation this is quite easy. Students just have to provide as expected result the instance of an object with the expected values.

When the examples turn into tests in the last step of the *design recipe*, we just need to make sure that the resulting object matches the expected one, field by field. We would like the test to follow the earlier style:

```
Cat cat = new Cat("Boots", "Sam");
check-expect (
   cat.newOwner("Jan")
   new Cat("Boots", "Jan"));
```

The problem is that the comparison of objects by their value is not supported by the popular object-oriented languages. All equality comparisons besides the reference (identity) equality have to be defined by the programmer. This may not be too difficult for simple classes with just a few fields. However, defining equality of two *AncestorTree* objects is not a simple task for a novice programmer. The problem is much worse if the classes allow for circularly referential data (a *Book* with the *Author* field, while the *Author* class has a *Book* field).

To make the test-first design possible we must provide software support for the design and evaluation of tests that matches student's knowledge of the language and his program design maturity.

Our students start during the first couple of weeks using *ProfessorJ Beginner* and *ProfessorJ Intermediate* languages within the *DrScheme* IDE. The *Beginner* language is a Java-like language that allows for a class to implement a single interface, it has no loop constructs and no assignment statement. It also does not support any field or method modifiers. However, it provides support for the test design similar to that available for the Scheme programs. The test case for the *newOwner* method mentioned earlier would be written as:

```
boolean testNewOwner =
   check cat.newOwner("Jan")
   expect cat.newOwner("Jan");
```

The program definitions are written in the *Definitions* window of the *DrScheme* IDE. The results of running the program are shown in the *Interactions* window. For Java-like programs the Interactions window shows pretty-printed display of all fields defined in the *Examples* class. Any failed test cases are shown in a separate *TestCase* window and show not only the failed test but also the values that have been compared together with a link to the failed test. There is almost no new syntax needed to define the tests, there is no work involved in evaluating the test cases or understanding how to report the results.

After about three weeks we move on to standard Java language using one of the common IDEs. (We have chosen Eclipse, but other schools use NetBeans, or even BlueJ). Without a similar support for test design, evaluation, and reporting, it would be impossible to continue with the systematic use of the *design recipe*. To support this test-first design throughout the curriculum we have designed and implemented a *tester* library for standard Java that provides support for the design of tests appropriate for a novice programmer. The *tester* library examines the user's program through Java reflection support and evaluates every test method defined in the *Examples* class. The test cases shown above would be written as:

```
boolean testNewOwner(Tester t){
  return
  t.checkExpect(cat.newOwner("Jan"),
      cat.newOwner("Jan")) &&
  t.checkExpect(cat.newOwner("Martin"),
      cat.newOwner("Martin"));
}
```

The *tester* library compares the objects by their values, reports any failed test cases, pretty-prints both the actual and the expected value of the failed test and provides a link to the failed test. Optionally, it pretty-prints every field in the *Examples* class, or prints all test results (successes as well as failures).

4.3. Understanding Test Design

The industry has embraced Test Driven Design (TDD) as leading to a more granular, comprehensible, and testable design. The pedagogy of the *design recipe* is not TDD (we do not build stubs that are further refined until the method is designed) but designing tests first is the common basis for both, and the benefits are similar.

Teaching students to design tests before they design the body of a method helps students think about the problem and understand what data is involved in the computation. The examples help them understand what is the method expected to accomplish. The examples also point out the situations when the desired method is too complex and is trying to do too many tasks at once. This suggest to students to design helper methods that provide solution to some of the needed sub-tasks. Running the tests and observing the test reports helps students diagnose errors and design programs that behave as expected. We teach students good programming practice from the beginning.

Practicing test-first design in introductory courses is impossible without proper software support. There are three parts that have to work together when designing tests. First, the programmer

must define what actual and expected values should be compared. This part is a natural part of program design and with the proper support should be an inseparable component of program design for all students.

The second part concerns the test evaluation. This requires that the actual and expected values are compared in the manner that is consistent with the user's expectations. This part is very difficult and would be an undue burden for a beginner student. Novice programmers cannot define correctly the equality comparison methods until they understand very well the underlying object model, know how to compare different types of data, how to detect circularity, etc. In Java, overriding the *equals* method also requires that the programmer overrides the *hashCode* method, adding another layer of complexity. While learning about the different levels of equality should be a part of learning how to design programs, such instruction should proceed in a gradual manner as appropriate for the students' level of competence. Meanwhile, the test evaluation should be managed by a library that matches students' understanding of equality and their mastery of the test case design.

The third step in practicing test-first design is the reporting of the test results. Typical test libraries may report that certain tests failed, and possibly provide links to the failed tests. However, they do not show the actual and expected values that have caused the failed test. The programmer can provide methods that display the data in a readable format (*toString* method), but again, this adds another burden, another routine task that student must do just to get the program running. Without the library support, student would have to do this before writing the first method, when the syntax overhead of just defining one class with a constructor is huge. Additionally, when the data may be circularly referential, the student must worry about breaking the circularity.

Our *tester* library has been designed to provide robust support for test design for students at all stages of their programming instructions. As student learn how to override the *equals* method, and how to define their own *toString* method, the *tester* library allows them to use their methods when needed, while using the library support for classes without their customized methods. Over the past year we have added new types of test scenarios to the library, responding to the needs of students using the first versions of the *tester* library. There are test methods that check whether the actual value matches one of the several possible random outcomes, whether the actual value falls within the given range of values, whether the invocation of a method by a given object throws the expected exception, and more. The latest version of the library provides a test coverage tool. We are working on designing special support for tests that change the state of an object and tests that verify changes in the structure of data.

The library has been used by students in five different schools (secondary schools and universities) and by over 200 students in our classes. At this time we use the *tester* library for our own software development on a daily basis.

5. EXPANDING HORIZONS

Most of the programs our students write rely on the *tester* library to display the results of the test evaluation and the values of the objects defined in the program. We do not process user input, as that is a task for a seasoned programmer who understands the intricacies of parsing the input String and converting it to meaningful data. Additionally, programs with the user input are very difficult to test, and are typically first tested with the simulated inputs. We add some exercises that handle user inputs later in the course, but still require that the program be tested on sample data apart from the actual input.

5.1. Creative Design

To make sure students see programs with user interactions and practice their independent design skills on interesting problems, we again provide a library that supports the design of interactive games. Our students focus on the game engine and a simple display of the game components on a canvas. We provide three libraries for Java programs. The first one supports the functional style of programming where students design the classes that represent the game state and implement methods that produce the new game state after a tick of the clock, *onTick*, and in response the key press by the user, *onKeyEvent*. They define the *draw* method that shows the game components as simple graphics objects on the given *Canvas*. Their game world extends our abstract *World* class that provides a *Canvas* for drawing, abstract methods *onKeyEvent*, *onTick*, and *draw*, and a *bigBang* method that starts the game on a *Canvas* of the given size running the clock at the given speed. Students do not worry about display panels, layouts, listeners, actions, etc., but instead, they concentrate on the design of the game logistics.

The second, imperative library is very similar, but the state of the World changes *onTick* and *onKeyEvent*. The third library provides the same user interface as the imperative one, but allows the students to convert their games into Java Applets.

The game assignments give the students motivation and an opportunity to decide for themselves how to organize the game components into classes, decide which class should be responsible for each task, where the collisions should be detected, etc. We conduct individual project reviews with every student after they have completed their first game and comment on their

design flaws, the program style, and the test coverage of their programs.

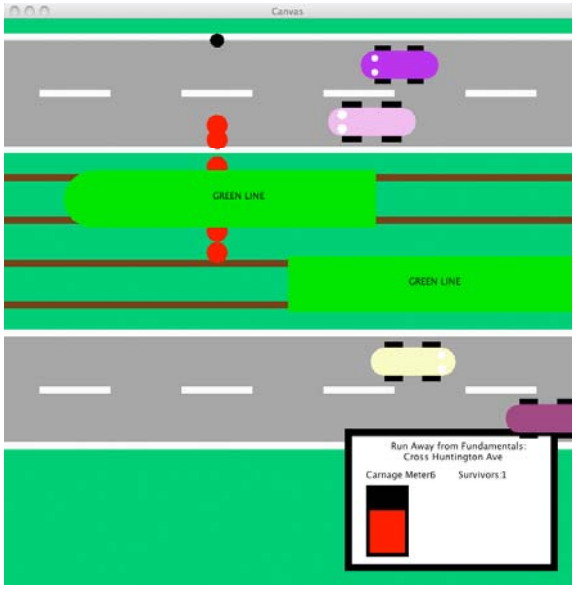


Fig. 4 *Crossing Huntington Avenue* game

5.2. Abstractions and Libraries: Reusable Code

Once the students understand the fundamentals of designing classes to represent data and designing methods for their class hierarchies, we use the repetition in their code as a motivation for introducing abstractions. Rather than starting with Java library classes, we first learn how to define abstract classes, interfaces that represent functional behaviour, interfaces that represent a traversal over a collection of data, classes that implement an interface that represents an abstract data type, and how to design generic algorithms that leverage the traversal and function objects.

Students who understand how to design programs with generalized behaviours learn quickly how to use libraries that have been designed on the same principles.

At this point we look at the cost of computation, learn about different data structures that support more elegant and efficient program design, learn about the complexity of algorithms, and run stress tests on large amounts of data to experience firsthand the differences between various algorithms and the underlying data structures they use.

5.3. User Interactions and GUIs

We believe that user interactions should be designed in a systematic way, just like the rest of model part of the program. At the end of the semester we introduce students to the JPT library that supports user interactions and GUI design through a cleanly designed abstraction layer. Students build an interactive GUI program with text fields, action buttons, sliders, colour choosers, radio buttons, graphics, key and mouse interactions, and a parsed console input after just one two hour lab session.

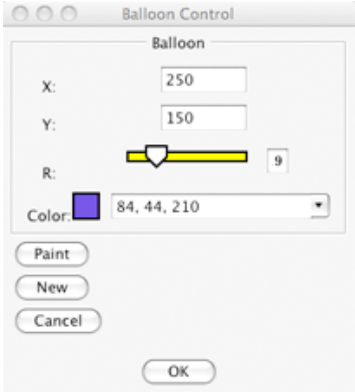


Fig. 5 A sample GUI panel

The *JPT* library includes *Java Power Framework (JPF)*. Any class that extends JPF automatically generates a *GUI* with a *Canvas* for drawing and a panel that contains one button for every method with no arguments that returns *void*. A button click runs the method in the context of the class where it has been defined. This provides some for some interaction and supports experimentation.

The simplicity of the level of abstraction presented by the JPT library and the JPF tools makes it possible for students to master quickly the basics of GUI design and to incorporate user interactions into their final project of the semester.

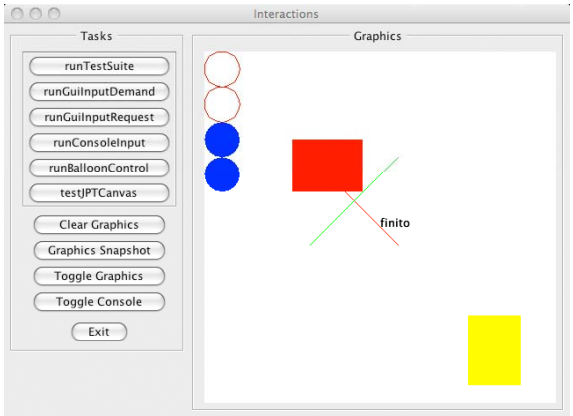


Fig. 6 *Java Power Framework* (*JPF*) window

6. CONCLUSION

The pedagogy of program design that drives our curriculum empowers the student and the instructor. In the early stages of learning to program the design recipes guide the students though the design process in clearly defined steps. Students learn to analyze the problem, define data, decompose complex problems into simpler ones, and they understand well the expected behaviour of every program they

write. When students encounter a problem the instructor knows what questions to ask and how to help the student discover the answer and the solution.

The same pedagogy, the same *design recipe* that works for young children and novice programmers, helps seasoned programmers to find a well-structured solution. By writing tests for every method all students acquire a lifelong design discipline that will make their programs better and safer.

7. ACKNOWLEDGEMENTS

The *TeachScheme!* curriculum has been designed by Matthias Felleisen, with the support of the coauthors of the text *How to Design Programs*: Robby Findler, Matthew Flatt, and Shriram Krishnamurthi. They are also the founders of the *PLT* team that continues to develop the *DrScheme* programming environment and languages.

The *ProfessorJ* languages and their support for testing have been implemented by Kathryn Gray and Matthew Flatt.

Matthias Felleisen designed the curriculum for the *ReachJava* component that Viera Proulx implemented in the classroom for the past seven years. Felleisen is also the lead author of the draft How to Design Classes. Proulx has developed a large collection of laboratory materials, programming assignments and projects and a collection of lecture notes and sample programs.

Viera Proulx designed and implemented the *tester* library. Weston Jossey has joined the *tester* library team last year and has since designed and implemented numerous enhancements to the *tester* library.

The *JPT (Java Power Tools)* has been developed jointly by Richard Rasala and Viera Proulx. Richard Rasala has been working on the library enhancement and maintenace for the past seven years.

Useful links:
How to Design Programs: http://htdp.org
TeachScheme/ReachJava: http://teach-scheme.org
Java Libraries: (tester, draw, idraw, adraw, geometry, colors): http://www.ccs.neu.edu/javalib
How to Design Classes materials: http://www.ccs.neu.edu/home/vkp/HtDC/
JPT and JPF: http://www.ccs.neu.edu/jpt

REFERENCES

[1] Felleisen, M., Findler, R.B., Flatt, M., Krishnamurthi, S.: TeachScheme! project: Computing and programming for every student. Computer Science Education, vol. 14, no. 1, 2004, pp. 55-77.

[2] Felleisen, M., Findler, R.B., Flatt, M., Gray, K.E., Krishnamurthi, S., Proulx, V. K.: How to Design Classes. in preparation.

[3] Felleisen, M., Findler, R.B., Flatt, M., Krishnamurthi, S.: How to Design Programs. MIT Press, Cambridge, MA, 2001.

[4] Findler, R.B., Flanagan, C., Flatt, M., Krishnamurthi, S., Felleisen, M.: DrScheme: A pedagogic programming environment for Scheme. In H. Glaser, P. Hartel, and H. Kuche, editors, Programming Languages: Implementations, Logics, and Programs, volume 1292 of LNCS, Southhampton, UK, September 1977, Springer, pp. 36-388.

[5] Gray, K.E., Flatt, M.: ProfessorJ: a gradual introduction to Java through language levels. In Companion of the 18th annual ACM-SIGPLAN Conference on Object Oriented Programming Systems, Languages, and Applications, Anaheim, CA, Oct. 2003, pp. 170-177.

[6] Gray, K.E., Felleisen, M.: Linguistic support for unit testing: Report uucs-07-013. http://www.cs.utah.edu/research/techreports.shtml

[7] Proulx, V.K.: Test-Driven Design for introductory OO programming. In SIGCSE Bulletin, vol. 41, no. 1, 2009.

[8] Proulx, V.K., Gray, K.E.: Design of class hierarchies: An introduction to OO program design. In SIGCSE Bulletin, vol. 38, no. 1, 2006, pp. 288-292.

[9] Proulx, V.K., Rasala, R.: Java IO and testing made simple. In SIGCSE Bulletin, vol. 36, no. 1, 2004, pp. 161-165.

[10] Rasala, R., Raab, J., Proulx, V.K.: Java Power Tools: Model software for teaching object-oriented design. In SIGCSE Bulletin, vol. 33, no. 1, 2001, pp. 297-301

APPENDIX: SAMPLE CODE WITH TESTS

```
import tester.*;

// to represent a traffic light color
class Light{
  String color;
  int time = 0;

  Light(String color){
    this.color = color; }

  Light(String color, int time){
    this.color = color;
    this.time = time; }

  void setTime(int time){
    this.time = time; }

  // reduce the time available by one
  void tick(){
    if (this.time > 0)
     this.time = this.time - 1; }
```

```
  boolean off(){
    return this.time == 0; }
}

// to represent a traffic light
class Traffic{
  Light current;
  int currIndex;
  int[] times;
  Light[] lights =
    new Light[]{new Light("Red"),
                new Light("Yellow"),
                new Light("Green")};

  // the constructor
  Traffic(int red, int yellow,
          int green){
    this.times =
      new int[]{red, yellow, green};
    this.currIndex = 0;
    this.current =
      this.lights[this.currIndex];
    this.current.setTime(
      this.times[this.currIndex]); }

  // change the light:
  // make the next light current
  void newCurrent(int index){
    this.currIndex = index;
    this.current =
       this.lights[this.currIndex];
    this.current.setTime(
      this.times[this.currIndex]); }

  // update the clock by one
  // change light if necessary
  void tick(){
    this.current.tick();
    if (this.current.off()){
      this.newCurrent(
        (this.currIndex + 1) % 3); }
  }
}


// a class for sample data and tests
class Examples{
  Examples(){}

  // test the method setTime
  // in the class Light
  void testSetTime(Tester t){

    Light r = new Light("Red");
    r.setTime(5);
    t.checkExpect(r,
        new Light("Red", 5)); }

  // test the method tick
  // in the class Light
  void testTick(Tester t){
    Light r5 = new Light("Red", 5);
    Light r0 = new Light("Red", 0);
    r5.tick();
    t.checkExpect(r5,
        new Light("Red", 4));

    r0.tick();
    t.checkExpect(r0,
        new Light("Red", -1)); }

  // test the method off
  // in the class Light
  void testOff(Tester t){

    Light r5 = new Light("Red", 5);
    Light r0 = new Light("Red", 0);
    t.checkExpect(r5.off(), false);
    t.checkExpect(r0.off(), true); }

  // test the method newCurrent
  // in the class Traffic
  void testNewCurrent(Tester t){

    Traffic tr = new Traffic(5, 2, 7);
    tr.newCurrent(2);
    t.checkExpect(tr.currIndex, 2);
    t.checkExpect(tr.current,
        new Light("Green", 7)); }

  // test the method tick
  // in the class Traffic
  void testTickTraffic(Tester t){

    Traffic tr = new Traffic(5, 2, 7);
    tr.tick();
    t.checkExpect(tr.currIndex, 0);
    t.checkExpect(tr.current,
        new Light("Red", 4));

    tr.tick();
    tr.tick();
    tr.tick();
    tr.tick();
    t.checkExpect(tr.currIndex, 1);
    t.checkExpect(tr.current,
        new Light("Yellow", 2));

    tr.newCurrent(2);
    tr.tick();
    tr.tick();
    tr.tick();
    tr.tick();
    tr.tick();
    tr.tick();
    tr.tick();
    t.checkExpect(tr.currIndex, 0);
    t.checkExpect(tr.current,
        new Light("Red", 5)); }

  // start to run program here
  public static void
      main(String[] argv){

    // make an instance of Examples
    Examples e = new Examples();

    // report all test results
    // show all data in Examples class
    Tester.runReport(e, true, true); }
}
```

PEDAGOGIKA A NÁSTROJE E-LEARNINGU

DOC. RNDR. JAROSLAVA MIKULECKÁ, CSC.

Katedra informatiky a kvantitativních metod, Fakulta informatiky a managementu, Univerzita Hradec Králové,
Rokitanského 62, Hradec Králové,
e-mail: Jaroslava.Mikulecka@uhk.cz

V počátcích zavádění e-learningu byl určujícím hlavně technologický aspekt, v posledním období se ale pozornost začíná soustřeďovat na rozvoj a ověřování vhodných pedagogických postupů. V příspěvku je prezentován stručný přehled výzkumných a vývojových aktivit různých subjektů, jejichž výstupy jsou využitelné pro zvýšení efektivnosti e-learningu.

Klíčová slova: *e-learning, vzdělávací objekty, digitální knihovny, návrh vyučování*

1. ÚVOD

E-learning má dva důležité aspekty – technologický a pedagogický. V počátcích zavádění e-learningu byl určujícím hlavně technologický aspekt, v posledním období se ale pozornost začíná soustřeďovat na rozvoj a ověřování vhodných pedagogických postupů. Technologická základna je v příspěvku chápána jako důležitý, nicméně pouze nutný předpoklad pro efektivní e-learning a pozornost je věnována hlavně způsobu, jakým je technologie používána v pedagogickém procesu.

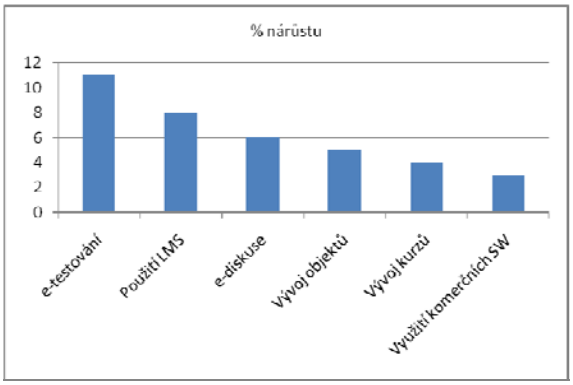
2. EFEKTIVNOST E-LEARNINGU

Termín efektivnost znamená realizovat činnost tak, aby vedla k zamýšlenému cíli aniž by docházelo k plýtvání s využívanými zdroji finančními, časovými či energetickými. V oblasti vzdělávání je cílem naučit studujícího novým dovednostem, rozšířit jeho přehled v určité oblasti či přivést ho k poznání nových zákonitostí tak, aby je dokázal tvůrčím způsobem dále využívat. Zdroje na zabezpečení tohoto procesu jsou především potřebný čas pedagoga, čas studujícího nutný k dosažení stanoveného cíle a samozřejmě studijní materiály, výukové prostory, informační technologie či laboratoře. Při posuzování efektivnosti vzdělávacího procesu je zásadní a netriviální zjistit, zda vzdělávání splnilo stanovený cíl – zda studující skutečně nabyl nové vědomosti a dovednosti v takové míře, že je bude schopen dále využívat a rozvíjet. Aby e-learning mohl být efektivní, je tedy nutné uváženě vybrat správnou metodologii výkladu, vhodný způsob ověřování vědomostí a správné technologie na podporu studentů při studiu i podporu učitelů při přípravě výuky i při samotné výuce.

Dnes již se informační a komunikační technologie ve školství zabydlely a prvotní nadšení z jejich přínosu leckde vyprchalo, aniž by došlo k radikální změně ve způsobu výuky. V této souvislosti jsou zajímavé výsledky projektu Weatherstation [15], jehož cílem bylo zjistit jaká je situace a hlavně jaké jsou změny v používaní informačních a komunikačních technologií na univerzitách, které v této oblasti patří v USA ke špičce. Ukázalo se, že při zavádění e-learningu se vycházelo z problematických předpokladů:

- Vybudujeme-li technologické zázemí e-learningu, zájemci přijdou.
- Studenti se budou s e-learningem cítit jako ryby ve vodě.
- E-learning si vynutí změny ve způsobu učení.

Studie potvrdila, že masívně se v tom čase používaly pouze elektronické systémy pro řízení studia (Learning Management Systems) a Powerpoint. Dále se ukázalo, že studenti chtějí být propojeni – ale hlavně mezi sebou, chtějí se bavit a chtějí prezentovat sami sebe. Postoj pedagogů k novým technologiím ukázal, že změny ve způsobu učení nebudou rychlé. Určitou naději vzbudila čísla, udávající nárůst používání jednotlivých prvků e-learningu – viz. Graf 1.



Graf 1 Nárůst používání jednotlivých prvků e-learningu. (Zdroj: [15])

Závěrem studie odpovídá na otázku, proč se rozplynul boom e-learningu a konstatuje, že e-learning se pokusil zkrátit obvyklý inovační proces. Nadšení některých firem vyprodukovalo příliš mnoho produktů, které byly uvedeny na trh aniž by prošly důkladným testováním a nesplnily očekávání zákazníků. E-learning vstoupil na scénu a uživatelé pořádně nevěděli, jak jej správně použít.

Podobný vývoj lze pozorovat i v České republice například i na vlastnostech e-kurzů či vzdělávacích objektů, přihlášených do soutěže eLearning, kterou pravidelně pořádá Univerzita Hradec Králové již téměř deset let. Pedagogové si rychle osvojili prezentační techniku; po prvotním váhání už dnes většina univerzit používá elektronický systém pro řízení studia. Některé univerzity se rozhodly implementovat svůj vlastní systém, těsně provázaný se studijní agendou, některé používají open source produkty – nejčastěji Moodle a některé zakoupily komerční systém. Další posun lze pozorovat ve využívání multimediálních materiálů – kurzy již kromě powerpointových prezentací a hypertextových materiálů většinou obsahují i animace a různé multimediální objekty; začínají se budovat digitální knihovny vzdělávacích objektů. Pro prověřování vědomostí mnozí pedagogové využívají různé typy testů, jež nabízejí příslušné LMS.

Univerzita Hradec Králové již od roku 2001 používá systém WebCT - nyní Blackboard Learning System CE6 (před dvěma roky došlo k fúzi dvou největších tvůrců a poskytovatelů LMS). V roce 2007 byl systém lokalizován do češtiny a proběhlo rozsáhlé školení uživatelů, financované ze Strukturálních fondů EU. Toto byl významný impulz ke zvýšení počtu pedagogů nabízejících v systému podporu studentům prezenční i kombinované formy studia. Stále větší oblibu si získává testování prostřednictvím Blackboard system CE6, který nabízí celkem devět typů otázek. Kromě standardního výběru z odpovědí a otázek typu Ano/Ne jsou k dispozici výpočet, kombinace správných odpovědí, přiřazování, přeházená věta, krátká odpověď, vyplnění prázdného pole, odstavec (volná odpověď). Kromě volné odpovědi, kterou musí pedagog ručně opravit, jsou všechny ostatní typy otázek vyhodnocované automaticky. Vzorové odpovědi lze tvořit s použitím standardních regulárních výrazů a tak lze dosáhnout velmi dobré flexibility v automatickém vyhodnocování odpovědí studentů. Přeházenou větu a vyplnění prázdného pole lze velmi dobře použít například i při výuce programovacích jazyků. Systém udržuje otázky v databázi otázek, z nichž lze poté vytvářet samotný test. Díky tomu, že otázky mohou být v databázi zařazovány do kategorií, lze velmi rychle vytvářet testy na prověření vědomostí libovolné oblasti. Testy samotné mají skupinu parametrů, jejichž nastavením se ovlivňuje například kdy, na jak dlouho a kterým studentům mají být testy k dispozici. Velmi užitečnou stránkou testů jsou statistiky, které systém vytváří – a to jak pro jednotlivé otázky v testu, tak pro jednotlivé studenty, skupiny studentů nebo všechny studenty. Lze tak velmi rychle porovnat například výkon různých skupin studentů.

Systém dále umožňuje odevzdávání a hodnocení studentských projektů a sledování aktivit studentů jako například, jak často a na jak dlouho jsou používané jednotlivé nástroje kurzu. Testy a úkoly mohou výrazně přispět k efektivnosti vzdělávání, základní podmínkou ale je, aby pedagogové kromě samotného nástroje dobře znali i metodologii tvorby testů.

3. PEDAGOGIKA E-LEARNINGU

Studenti nejsou homogenní skupina s identickými požadavky na vzdělávací proces a každý preferuje určitý styl učení. V kombinované formě studia se obvykle vyskytují navíc další rozdíly, dané úrovní vstupních vědomostí, různými zkušenostmi a také různou motivací ke studiu. Na tuto situaci je vhodné reagovat při přípravě vzdělávacích materiálů tak, aby se respektovaly hlavně různé styly učení. Problém je, že dopředu obvykle nevíme, jaký styl učení převažuje v dané skupině a proto je vhodná určitá redundance ve studijních materiálech a studenti by měli mít možnost výběru. Styly učení jsou předmětem mnoha pedagogických výzkumů a na jejich identifikaci existují standardní testy. Velmi známý je Kolbův model [9], který specifikuje čtyři základní *styly učení* na základě preferovaných vzdělávacích aktivit:

- aktivní experimentování (AE)
- reflektivní pozorování (RP)
- konkrétní zkušenost (KZ)
- abstraktní představa (AP)

Kombinace stylů učení je určující pro definování základních typů studujících:

Přemítavec (converger - AE, AP). Studenti tohoto typu rádi řeší problémy a umí logicky zdůvodňovat. Obvykle úspěšně řeší technické problémy a jsou spíše nesociální. Pro tento typ bude nejvhodnější použít simulace a úkoly k řešení. Dle Kolba, tento typ studentů převažuje na technických oborech.

Aktivista (Diverger - KZ, RP). Jedná se o imaginativní typ člověka, který se rád zapojuje do nových zkušeností. Je úspěšný ve vytváření alternativních hypotéz a je spíše sociální. Pro tento typ jsou vhodné skupinové aktivity a výzkumné aktivity.

Teoretik (Assimilator - AP, RP) – jedná se o induktivní typ studenta, který rád vytváří teoretické modely a abstraktní koncepty. Preferuje teorii před praktickými zkušenostmi a je spíše nesociální. Obvykle preferuje teoretický výklad a prezentaci příkladů.

Pragmatik (Accomodator - AE, KZ) – rád zkouší nové myšlenky v praxi, řád podstupuje riziko. Je přizpůsobivý, problémy často řeší metodou pokusů a

omylů a je spíše sociální. Pro tento typ studentů je nejvhodnější skupinové řešení problémů. Dle Kolba, tento typ studentů převažuje na ekonomických oborech.

Banham v [1] zkoumal efektivnost vzdělávacího procesu v závislosti na preferovaném stylu učení a na formě vzdělávání a dospěl k zajímavým výsledkům:

V on-line vzdělávání měli lepší výsledky studenti, preferující konkrétní zkušenost a studenti preferující reflektivní pozorování.

V tradičním vzdělávání naopak jsou úspěšnější studenti, preferující abstraktní představy a aktivní experimentování. Z toho lze odvodit, že „nejrizikovějším" typem studentů v případě on-line vzdělávání jsou přemítavci, nejlepší výsledky můžeme očekávat u aktivistů.

Teoretický rozvoj tvorby vyučování (instructional design), tj. detailní popis vyučovacích operací v prostředí e-learningu zpočátku zaostával za samotným rozvojem a nasazením moderních technologií do vzdělávání, v poslední době se v této oblasti objevilo několik zajímavých publikací například [4, 8, 12, 14] .

Motivace studentů je jedním ze základních faktorů ovlivňujícím efektivnost vzdělávání a proto by jí měla být věnována náležitá pozornost ve všech přístupech k tvorbě vyučování. Dobře známý Kellerův ARCS model (Attention, Relevance, Confidence, Satisfaction lze doporučit i v případě e-learningu. Základní zásady, které doporučuje ARCS model [8] lze stručně popsat následovně:

- Pozornost studentů lze upoutat osobní nebo emocionální informací, otázkami, mentální výzvou, příklady z reálného života.
- Relevantnost výuky lze docílit zjištěním potřeb a očekávání studentů a těmto očekáváním přizpůsobit obsah kurzu, příklady, úkoly i testy.
- Důvěru studentů lze vzbudit vhodně zvolenou obtížností problémů – na začátku musí být jasně dáno za jakých podmínek může student úspěšně zvládnout látku. Tyto podmínky musí být pro studenta splnitelné, ne však triviální.
- Uspokojení studentů se odvíjí od toho, zda mají příležitost použít nově získané znalosti či dovednosti, nebo jsou-li jejich nové znalosti na závěr prověřené – nejlépe na řešení reálných problémů.

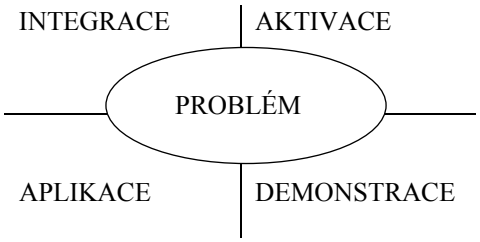
Organizace vzdělávání by se dle [4] měla řídit následující logikou:

- upoutat pozornost studenta,
- informovat studenta o cílech a stanovit úroveň očekávaných výsledků,
- připomenut již naučené znalosti,
- jasně představit nové učivo,
- řídit učení,
- požadovat viditelné projevy chování dokazující učení,
- zajišťovat zpětnou vazbu,
- hodnotit výkon studenta,
- podporovat přenos poznatků a jejich aplikaci v jiných souvislostech.

David Merill [12] analyzoval různé přístupy, používané při návrhu vzdělávacích kurzů a identifikoval základní principy, které jsou přítomné ve většině z nich. Konstatoval, že učení lze usnadnit, pokud:

- Studenti jsou zapojeni do řešení reálných problémů. (Nechte mě řešit celý problém!),
- Nové znalosti se budují na základě osvěžených předchozích znalostí. (Poraďte mi, kde mám začít!).
- Studentům jsou prezentovány a vysvětleny nové znalosti. (Ukažte mi, jak se to dělá!).
- Studenti aplikují nové znalosti. (Nechte mě to udělat!).
- Studenti nové znalosti použijí v reálném životě. (Sledujte mě!).

Fáze efektivního učení dle Merrilla jsou zobrazeny na obrázku 1. Základem tohoto přístupu je problém, k jehož pochopení a řešení směřují všechny další kroky.



Obr. 1 Fáze efektivního učení dle [12]

Moderní a pro e-learning vhodný je i holistický model 4D/ID navržený van Merriënboerem [14], jehož základem je vhodně volená hierarchie a provázanost problémů.

3. PEDAGOGICKÝ MODELOVACÍ JAZYK

Pedagogický modelovací jazyk (EML - Educational Modeling Languae) byl vytvořen na Open University of the Netherlands jako výsledek snahy o vytvoření modelů různých pedagogických postupů. Jazyk EML má vazbu na XML a lze jej použít na vyjádření celého spektra pedagogických procesů. Tento jazyk byl posléze použitý jako základ specifikace IMS Learning Design (IMS LD). Tato specifikace se týká modelování procesů v návrhu vyučování a vznikla částečně proto, aby se přesunula pozornost od obsahu vzdělávání na procesy

vzdělávání. IMS LD specifikuje jazyk na popis vzdělávacích aktivit a popisuje vazbu na XML. Předpokládá, že postup vzdělávání je organizován jako posloupnost vzdělávacích jednotek (vzdělávacích modulů). Vzdělávací jednotka (learning unit) je definována jako samostatná část procesu vzdělávání, která byla navržena tak, aby po jejím absolvování studenti dosáhli jeden nebo více souvisejících vzdělávacích cílů a nelze ji rozdělit na menší části, aniž by ztratila svůj sémantický význam. Cíl vzdělávání může být definován na různé úrovni podrobnosti.

IMS LD je vytvořen na základě metafory divadelního představení a sestává z různých prvků (elementů): rolí které herci hrají (kdo co dělá), aktivit (co dělá) a prostředí, které zahrnuje kde to dělá (služby) a s čím to dělá (vzdělávací objekty).

Obecně je scénář nebo návrh popsán jako metoda, která obsahuje představení, dějství a role. IMS LD umožňuje znovupoužití těchto prvků v jiných návrzích vzdělávání.

Pro zjednodušení implementace je specifikace IMS LD rozdělená do tří úrovní. Úroveň A obsahuje jádro návrhu: herce, aktivity, zdroje a jejich koordinaci prostřednictvím metody, dějství a rolí. Toto jednoduše poskytuje sérii v čase uspořádaných vzdělávacích aktivit, které mají realizovat studenti a učitelé použitím vzdělávacích objektů a/nebo služeb. Úroveň B přidá více řízení a složitosti prostřednictvím vlastností a podmínek. Úroveň C přidá zprávy (oznámení), tak aby učitelé, studenti a jiné vzdělávací systémy mohli dostat informace o postupu vzdělávacích aktivit.

Použití této specifikace vyžaduje od tvůrce vzdělávání aby se zamyslel nad pedagogikou a zvážil, s jakým cílem je vzdělávání navrhováno a jaké prerekvizity jsou potřebné, aby bylo možné identifikovat aktivity všech herců a identifikovat které nástroje se mají v procesu použít.

Typické kroky v procesu návrhu vzdělávání sestávají z analýzy konkrétního vzdělávacího problému, která vyústí do didaktického postupu který se zachytí ve formě slovního popisu - často jako výčet bodů. Na základě slovního popisu se vytvoří UML diagram aktivit. Tento diagram slouží jak základ pro XML dokument, který odpovídá IMS LD specifikaci.

IMS LD umožňuje modelovat nejen situaci jednotlivých studentů, ale také situace, na kterých se podílí více studentů nebo kolaborativní vzdělávací procesy. IMS LD vzbudil pozornost u mnoha výzkumníků, zabývajících se znovupoužitím scénářů vzdělávání. Tato specifikace je odpovědí na potřebu formalizace reprezentace scénářů vzdělávání. Na druhé straně ale tato notace není lehká na pochopení pro tvůrce vzdělávání, kteří potřebují nástroje, které ulehčí implementaci specifikace.

Většina nástrojů na modelování pedagogických postupů je ještě v začátcích. Dobře dokumentovaný a relativně lehce použitelný je open source grafický editor MOT Plus [13]. Tento produkt v sobě zahrnuje 4 typy editorů – jedním z nich je pedagogický editor (Educational Editor) a umožňuje vytváření vzdělávacích jednotek splňujících specifikaci IMS LD na úrovni A. Dalším open source nástrojem, který se velmi rychle rozvíjí je LAMS (Learning Activity Management System). LAMS [10] lze integrovat do systému Moodle, Blackboard i Microsoft Sharepoint.

4. DIGITÁLNÍ KNIHOVNY

Digitální knihovny mohou výrazně přispět ke zvýšení efektivnosti a kvality vzdělávání tím, že podporují znovupoužití vzdělávacích materiálů a často poskytují i platformu pro spolupráci a vytváření komunit. Digitální knihovnou se v kontextu e-learningu obvykle rozumí kolekce vzdělávacích materiálů, která obsahuje metadata těchto objektů a je přístupná prostřednictvím Internetu. Kromě samotných objektů může digitální knihovna obsahovat i cestu k objektům, které mohou být umístěné v jiné digitální knihovně nebo kdekoliv v síti. Materiály v digitální knihovně by měly splňovat základní technické standardy, aby je bylo možné použít v různých LMS systémech a měly by mít garantovanou kvalitu. Kvalita vzdělávacího objektu má několik aspektů a je dána kvalitou vzdělávacího obsahu, potenciálem svého použití a přívětivostí svého uživatelského rozhraní.

Digitální knihovna vzdělávacích materiálů by měla poskytovat následující služby:

- uložení vzdělávacích materiálů a/nebo uložení odkazů na vzdělávací materiály,
- náhled a stažení materiálů,
- uložení a prohlížení metadat, jejichž struktura by měla splňovat standard LOM,
- autorizaci na základě rolí,
- export obsahu ve formě IMS Content Package nebo SCORM,
- jednoduché a pokročilé vyhledávání v metadatech,
- klasifikaci zdrojů,
- import metadat,
- automatický sběr (harvesting) metadat.

V posledním období se do budování a provozu digitálních knihoven pro potřeby akademických institucí ve světě hodně investovalo - například projekt EduSource Canada nebo projekt JORUM ve Velké Británi. V České republice Ministerstvo školství, mládeže a tělovýchovy v rámci rozvojových programů finančně podpořilo projekt Spolupráce vysokých škol při tvorbě

multimediálních pomůcek jehož hlavními výstupy jsou vytvořené a v distribuované digitální knihovně [3] zpřístupněné multimediální vzdělávací materiály.

Co je důvodem tohoto zájmu o budování digitálních knihoven? Jeden z důvodů je ten, že digitální materiály vznikají a je velmi žádoucí, zejména pokud vznikají z veřejných zdrojů, aby byly dány k dispozici na znovupoužití. Dá se očekávat, že pokud bude ve veřejně přístupných knihovnách k dispozici určité kritické množství vzdělávacích objektů přiměřené kvality, jejich uživatelé začnou vytvářet a do knihovny ukládat další materiály, čím by se mohlo dosáhnout dlouhodobější udržitelnosti takové knihovny. Příkladem může být knihovna MERLOT [11].

V lednu 2006 byla publikovaná zajímavá studie [7], analyzující 19 aktuálně používaných digitálních knihoven vzdělávacích materiálů, přičemž se sledovaly různé aspekty, například jak knihovna vznikla, komu je určena, jaké služby poskytuje, kolik a jakých objektů (nebo odkazů na objekty) obsahuje, jakou má základní technickou architekturu.

Analýza dospěla k následujícím zásadním zjištěním trendu vývoje:

a) Většina digitálních knihoven představuje on-line katalog webovsky přístupných zdrojů, přičemž samotné zdroje neobsahuje.
b) Objevuje se federativní prohledávání mezi velkými digitálními knihovnami.
c) V digitálních knihovnách vytvořených a provozovaných v USA a v Kanadě se používá Creative Commons Licence, která umožňuje modifikovat objekt získaný z knihovny a uložit jej jako nový objekt za podmínky, že je uvedena citace původního objektu.
d) Digitální knihovny většinou mají zaveden určitý systém na zajištění kvality materiálů.
e) Stoupá počet knihoven, které využívají decentralizovanou peer-to-peer architekturu. Tato architektura umožňuje instituci administrovat vlastní instanci knihovny, přitom ale poskytuje možnost jednoduše prohledávat i jiné knihovny.
f) Knihovny začínají poskytovat různé podpůrné nástroje a to jednak nástroje pro vývojáře, různá fóra, školení, příklady dobré praxe a různé publikace.
g) Velké knihovny zavedly pobídky pro přispěvatele, aby se zvýšil počet nabízených materiálů.
h) Stoupá počet knihoven, které využívají open source řešení.

Obecně slouží digitální knihovny pro ukládání a zpřístupňování různých digitálních dokumentů – hlavním ukládaným artiklem tedy nemusí být vzdělávací materiály a vzdělávací objekty ale například diplomové práce, vědecké práce apod. Pro ukládaní tohoto typu digitálních objektů se s úspěchem používají open source produkty, jakým je DSpace, Fedora či EPrints. U těchto produktů lze do budoucna očekávat, že budou poskytovat i specifické služby, požadované v případě digitálních knihoven vzdělávacích objektů.

Odpovědnost za kvalitu vzdělávacích objektů, umístěných v digitálních knihovnách je v některých případech ponechána zcela na autorovi objektu. V některých případech jsou objekty před svým zveřejněním standardně recenzovány, v mnoha případech jsou objekty hodnoceny pouze svými uživateli. Podrobně rozpracovaný proces hodnocení kvality materiálů mají knihovny MERLOT [11] a CLOE [8].

Knihovna MERLOT (Multimedia Educational Resources for Learning and Online Teaching) je budovaná od roku 1997 na California State University a obsahuje katalog vzdělávacích zdrojů pro univerzity, tj. obsahuje pouze metadata s odkazy. Vzdělávací materiály jsou rozděleny do čtrnácti oborových oblastí z nichž každá má redakční radu, která řídí recenzní proces a garantuje i školení recenzentů. Všechny oborové rady mají jednotná základní kriteria a škálu hodnocení od 1 do 5. Hodnotí se kvalita obsahu, efektivnost objektu pro vzdělávání a jednoduchost použití.

Základní filozofie hodnocení kvality je založena na tom, že objekty mohou dostat "známku kvality". Metadata objektů jsou zveřejněna aniž byly objekty recenzovány a recenzní proces probíhá až následně, přitom ale cílem recenze není objekty vyřadit, ale označit daný objekt "známkou kvality".

Celý proces probíhá tak, že oborově příslušná ediční rada vybere materiály, vhodné pro recenzi na základě předběžného hodnocení. Požádá autory předběžně vybraných materiálů o souhlas s recenzí. Každý zdroj recenzují dva recenzenti a je-li mezi nimi nesoulad, tak se osloví ještě třetí recenzent. Recenze se pošle autorovi i s žádostí o povolení recenzi publikovat. Publikují se pouze recenze s hodnocením alespoň tři. Nemá-li materiál žádné hodnocení, tak buď nebyl vůbec recenzován, anebo dostal hodnocení horší než 3. Jako stimulaci přispěvatelů do knihovny ediční rady udělují každoročně Cenu Merlotu, která je částečně sponzorovaná nakladatelstvím McGraw-Hill.

CLOE (The Co-operative Learning Object Exchange) je knihovna, obsahující výlučně interaktivní vzdělávací objekty. Pojem vzdělávací objekt se ještě stále vyvíjí. Dle standardu IEEE je vzdělávací objekt definován jako jakákoliv entita, použitelná v procesu vzdělávání. Nyní se za vzdělávací objekt většinou považuje entita, která slouží určitému vzdělávacímu cíli, je opatřená vhodnou množinou metadat a je schopná standardně

komunikovat s LMS. Takto je chápán vzdělávací objekt i v knihovně CLOE.

Všechny vzdělávací objekty jsou před svým zveřejněním v knihovně recenzované, přičemž prvním krokem je ověření technické funkčnosti objektu. Pokud jsou s technickou funkčností problémy, tak objekt není akceptován a autor objektu obdrží zprávu. Pokud objekt splňuje technické požadavky, administrátor knihovny zabezpečí recenzní řízení (tři recenzenti – jeden pedagog a dva odborníci v daném oboru).

Škála hodnocení má pouze tři úrovně: vyhovuje, nevyhovuje, částečně vyhovuje.

Hodnotí se stejné aspekty jako v případě knihovny MERLOT, tj. kvalita obsahu, efektivnost objektu pro vzdělávání a jednoduchost použití. V případě hodnocení efektivnosti obsahu se hodnotí, zda

- je jasně specifikován cíl vzdělávání,
- vzdělávací objekt podporuje cíl,
- je jasně definovaná cílová skupina,
- je jasně popsán způsob použití objektu,
- technologie napomáhá vzdělávacímu cíli,
- studující má možnost získat zpětnou vazbu,
- autor má možnost získat zpětnou vazbu,
- jsou specifikovány požadavky na předchozí vědomosti,
- vzdělávací objekt může existovat samostatně.

Autor, jehož objekt úspěšně prošel recenzním řízením, dostane oficiální informaci o zveřejnění objektu a poté každý rok dostává informace, jak byl objekt využíván. Tato zpětná informace autorům objektů může být dobrým motivačním faktorem pro další příspěvky do knihovny.

Hlavním problémem mnoha digitálních knihoven, které vznikly jako výstupy různých projektů je jejich udržitelnost po ukončení financování.

5. ZÁVĚR

E-learning je oblast, která se stále rychle rozvíjí. V článku jsme stručně představili nové metodologie a nástroje, které mohou přispět ke zvýšení efektivnosti vzdělávacího procesu.

LITERATURA

[1] Banham, H.: Training Effectiveness, On-Line De-livery and the Influence of Learning Styles, ACM Press, New York, NY, USA, 2002

[2] CLOE: http://cloe.on.ca

[3] Edubrana: www.edubrana.cz

[4] Gagné, R. M., Wagner, W.W., Golas,C. ,Keller, J.M. (2005), Principles of instructional design, Thomson Waldsworth

[5] IMS Learning Design Specification http://www.imsglobal.org/learningdesign/index.html

[6] JISC Online Repository for learning and teaching Materials. JORUM Scoping and Appraisal Study, 2004, http://www.jorum.ac.uk/

[7] JOURUM Project: E-learning Repository http://www.jorum.ac.uk/

[8] Keller. J., M.: Motivation and Performance. . In: Trends and Issues in Instructional Design and Technology. Edited by R.A. Reiser and J.V. Dempsey, Pearson Education, 2nd. Edition, 2007

[9] Kolb, D.A.: Experiental learning: Experience as the source of learning and development. Engelwood Cliffs, NJ, Prentice Hall, 1984

[10] LAMS: http://www.lamsinternational.com/

[11] MERLOT: http://www.merlot.org

[12] Merrill, D.: First Principles of Instruction: A synnthesis. In: Trends and Issues in Instructional Design and Technology. Edited by R.A. Reiser and J.V. Dempsey, Pearson Education, 2nd. Edition, 2007

[13] MOT PLUS: http://www.unfold-project.net

[14] van Merriënboer, J., G.: Alternate Models of Instructional Design: Holistic Design Approaches and Complex Learning. In: Trends and Issues in Instructional Design and Technology. Edited by R.A. Reiser and J.V. Dempsey, Pearson Education, 2nd. Edition, 2007

[15] Zemsky, R., Massy, W.,F.:Thwarted Innovation. What Happened to e-learning and Why, A Final Report for The Weatherstation Project of The Learning Alliance at the University of Pennsylvania in cooperation with the Thomson Corporation, 2004. Přístupné na http://www.irhe.upenn.edu/Docs/Jun2004/ThwartedInnovation.pdf

ŽÁKOVSKÉ PROJEKTY VE VÝUCE INFORMAČNÍCH TECHNOLOGIÍ

PAEDDR. JIŘÍ VANÍČEK, PH.D.

Katedra informatiky, Pedagogická fakulta, Jihočeská univerzita v Českých Budějovicích, Jeronýmova 10, České Budějovice, e-mail: vanicek@pf.jcu.cz

Článek se zabývá projektovou výukou informačních technologií s cílem zpřesnit čtenářovu představu žákovského projektu, vysvětlit, čím se projekt liší od jiné práce na počítači a poradit učiteli, jak kvalitní projekt připravit, vést a hodnotit. Článek dále obsahuje typologii chyb, kterých se učitelé dopouštějí při přípravě projektu. Jednotlivé aspekty přípravy a vedení projektu učitelem, stejně jako hodnocení projektu jsou komentovány na základě autorových zkušeností s přípravou učitelů informatiky na pedagogické fakultě a s vlastní realizací a tvorbou projektů v roli učitele.

Klíčová slova: *didaktika informatiky, žákovské projekty, projektová výuka, příprava učitelů*

1. ÚVOD

Technologická podpora školní soutěže, CD jako vzpomínka na lyžařský výcvikový kurz, Webová galerie žákovských fotografií, Školní anketa o nejpopulárnějšího spolužáka, Počítačová hra Mravenec, Školní časopis, Kronika naší ulice, Moje firma, Poznáváme své kořeny – tak nějak mohou znít názvy nebo témata projektů, jejichž uskutečňování je v dnešní škole prosazováno [1]. Jde o trend jistě zdravý, práce na projektu rozvíjí řadu kompetencí, které žák bude potřebovat v budoucín povolání (schopnost organizovat věci a činnosti, spolupracovat v týmu, jít za cílem, řešit problémy a nacházet vlastní řešení, vytvářet a flexibilně měnit plán, přenášet poznatky z jednoho oboru do druhého). Ovšem stejně tak mohou výše uvedené názvy skrývat aktivity, které nejsou projekty. V českých školách je v poslední době poměrně hojná představa, že jakákoliv dlouhodobější, nejlépe mezipředmětová činnost, která není „podle učebnice“ a která je zastřešena nějakým společným netradičním názvem, je projekt (a protože projekty jsou vedením školy vyžadovány, je i neprojektová výuková činnost často jako projekt vykazována). Dokumentují to i některé náměty na projekty (a to nejen z oblasti informačních technologií) zveřejněné na metodickém portálu RVP – někdy se z takového zadání nedá poznat, zda žáci opravdu budou pracovat projektově.

V tomto článku se chceme projekty podrobněji zabývat s cílem zpřesnit čtenářovu představu žákovského projektu, poradit učiteli a vysvětlit

- čím se projekt liší od jiné práce na počítači
- jak projekt připravit, vést a hodnotit.

2. CO JE PROJEKT A CO NENÍ PROJEKT

Chceme-li zařadit projektovou výuku do stávající struktury vyučovacích forem, tak z hlediska sociálních forem ji lze umístit mezi práci v celé třídě a práci ve skupině, z hlediska povahy žákovských aktivit mezi tzv. související formu (žák se pohybuje v oblasti učitelem prezentované látky) a řízené objevování (znovuobjevování obecně známého poznatku „pro sebe“). Nejedná se tedy o individuální výuku a nejde o tzv. volný výzkum (s otevřeným koncem) [2].

Projekt má své charakteristické vlastnosti: samoorganizovanost a zodpovědnost, cílenost, orientace na produkt, důraz na praktickou činnost, orientace na zájmy zúčastněných, situační zřetel, sociální učení a v neposlední řadě mezipředmětovost [2].

Je třeba rozlišovat mezi učitelským projektem a žákovským projektem. Pokud si učitel připravuje svoji výuku netradičními formami, např. za pomoci experimentování, objevování, s aktivním zapojením žáka, zařazením nových témat a jejich provázáním apod., je to velice záslužná činnost. Zde se jedná o učitelův projekt jeho výuky, automaticky to však neznamená, že také žáci nutně budou pracovat projektovým způsobem.

Projektová výuka může kvalitně připravit žáka pro budoucí povolání takzvaného samostatného zaměstnance, který umí nejen poslouchat pokyny a rozumět instrukcím, ale dokáže se sám rozhodovat, přijímat opatření, plánovat svoji činnost i činnost jiných, být schopen komunikovat v týmu, jít za určitým cílem a řešit při tom vyvstávající problémy. Právě typ zaměstnance s takovýmito kompetencemi je zaměstnavateli velice žádaný, dá se říci, že je nedostatkovým zbožím.

Jak poznat, co je projektová výuka a co již ne? Žákovský projekt lze od neprojektové práce dobře rozeznat podle míry podílu řešitele na zadání úlohy a míry samostatného řízení pracovního procesu. Jestliže je žákovi přesně stanoveno, co má kdy dělat, jde pouze o sled (navazujících) úloh, nikoliv o projekt. V praxi (tedy v zaměstnání dospělých) se při zadávání projektu často vyžaduje podíl toho, kdo projekt realizuje, na upřesnění, specifikaci zadání nebo někdy i rozsáhlejší změnu zadání (po konzultaci se zadavatelem). Vychází se z toho, že zadavatel nemá tolik odborných zkušeností, nezná technologické postupy, novinky v oboru apod., aby problém byl schopen uchopit v plné šíři a zadání do

detailů specifikovat. Žákovské projekty se snaží právě toto simulovat [3].

Projektem **není** sled krátkých navazujících úloh, které má žák provést. Projektem **není** ani dlouhodobá činnost, pokud má žák přesně určeno, co má dělat, a nemůže se sám svojí aktivitou zapojit do jeho přípravy. Projekt mívá často ne zcela konkrétní zadání, nicméně má stanoven cíl (co má vzniknout) a požadavky (kritéria), které musí řešitel projektu splnit.

Názvy projektů z úvodu článku mohou být opravdu názvy žákovských projektů, stejně tak jako tématy pro tzv. integrovanou tematickou výuku [5], v níž některé téma dlouhodobě zastřešuje a překrývá řadu výukových aktivit ve třídě. Na první pohled jde o tentýž typ výuky: učitel připravuje často rozsáhlé přípravy, má originálně sestavené kurikulum, zařazuje netypické úlohy, výuka se může odehrávat ve více vyučovacích předmětech atd.

Tím nechceme říci, že takováto výuka je nevhodná, horší než projektová nebo že by neměla být vyučována. Naopak, takovéto aktivity jsou velmi často učitelem velmi kvalitně připraveny i odučeny. Pouze chceme sdělit, že v tomto případě nejde o žákovský projekt, a učitel, který takovou výuku připravuje, by měl mít na zřeteli, že takovou výukou se netrénují ty kompetence žáka, jejichž získání projektová výuka obecně deklaruje.

Obr. 1 Školní tablo, typický grafický projekt. Žáci se vyfotografují, upraví fotky a každý zvolí svoje téma výsledného tabla. Vytvořili žáci ZŠ Nerudova, Č. Budějovice

Jaká je vzájemná pozice zadavatele a řešitele při startu projektu? Žák může rozhodovat o rozsahu, formě výstupu, o osnově práce nebo o druhu použitých údajů po domluvě s učitelem nebo na základě omezujících kritérií, na kterých se třída dohodne nebo jež jsou dána striktně učitelem (v roli jakéhosi zákazníka). Učitel musí mít právo stanovit omezující kritéria (a stanovit je tak, aby práce na projektu rozvíjela ty žákovy dovednosti v práci s počítačem, které učitel potřebuje). Neobsahuje-li zadání konkrétní kritéria, je velice obtížné projekt hodnotit, protože téměř každá činnost se „vejde“ do široce pojatých mantinelů; každá práce je pak „správná“. Také pak není zajištěno, že se po obsahové stránce žáci učí to, co učitel očekává.

Příklady, v nichž jsou na konkrétních projektech uvedená kritéria rozebrána, najde zájemce v [4].

3. PŘÍPRAVA A VEDENÍ PROJEKTU

Navštíví-li projektové vyučování nějaká návštěva, s největší pravděpodobností přijde v době vlastní realizace projektu: forma výuky je uvolněná, žáci pracují sami, autorita učitele není vidět. Takové vyučování vypadá jako snadné. Je však třeba říci, že

- právě při takovéto formě je náročné udržet základní směr, tedy orientaci na produkt (žáci se často zaberou do práce, cíl ztrácí ze zřetele)
- za touto „pohodou“ se skrývá mnoho práce učitele při samotném zahájení projektu a také při jeho dokončování (odevzdávání prací, prezentace výsledků a oponentura).

Rozdělíme realizaci žákovského výukového projektu na několik etap:

1. před projektem (příprava projektu učitelem)
2. příprava projektu ve třídě
3. vlastní realizace
4. dokončení projektu

Tyto etapy budeme v následujících odstavcích komentovat.

Před projektem

Projekt musí být předem dobře promyšlen. V této fázi učitel připravuje (vybírá a přizpůsobuje) obsah projektu. Podstatné je jednotící téma projektu, které má význam pro motivaci, pro počáteční identifikaci žáků s projektem a jeho cíli. Do většiny projektů lze „naroubovat“ i velmi různorodé aktivity z naprosto odlišných oborů, spojovací prvek se najde snadno. Důležitější je stanovit si dovednosti a znalosti, které se žáci mají naučit, a především těm přizpůsobit obsah a také kritéria pro hodnocení výsledné práce.

Projekt je svojí povahou zaměřen na produkt, na výsledek (obrázek, prezentace, web, video, referát o provedeném zkoumání...). To svádí k představě, že je produkt považován za to nejdůležitější, co má vzniknout. Jestliže je pro žáky cílem projektu opravdu vytvoření nějakého produktu, učitel k takovému pohledu sklouznout nesmí. Pro učitele je cílem samotné vzdělávání, tedy jaké dovednosti a znalosti si žáci v průběhu realizace projektu osvojí, co nového budou na konci projektu umět. Pouhé vytváření žákovských výtvorů bez ohledu na to, že při jejich tvorbě se žák něco učí, je samoúčelné a v důsledku bere učiteli čas.

Učitel si potřebuje projekt dobře naplánovat. Protože se jedná o výukovou aktivitu s ne zcela specifikovaným výstupem, ale omezeným časem, je potřeba v plánu počítat s časovou rezervou, zvláště u začátečníků. Je totiž zásadně důležité, aby projekt

byl dokončen a také odpovídajícím způsobem zhodnocen; není možné, aby žáci udělali třeba 80 % práce a pak projekt skončil.

Příprava projektu ve třídě

Před žáky učitel předstupuje s jasnou vizí obsahu projektu, i když jeho zadání se bude ve třídě ještě upravovat. Příprava projektu společně s žáky spočívá jednak v počátečním přednesení tématu a zaměření projektu, jednak (a to především) v diskusi o obsahu a způsobu provedení projektu. Učitelovým zájmem je, aby žák projekt přijal za svůj, aby se s ním ztotožnil, což se nejlépe pozná právě mírou spolupráce žáka na utváření zadání (co by mohl výstup projektu obsahovat, jak jej realizovat atd.) Zde učitel nesmí litovat času, chvátat apod., protože v této fázi vzniká žákova představa výsledné práce. Povaha této činnosti je unikátní, je zde trénována schopnost porozumět, vcítit se do role zákazníka, promítnout si jeho zadání do své představy výsledného produktu a pracovního plánu a tuto představu korigovat.

Velmi vhodnou metodou je brainstorming. V první fázi diskuse je komukoliv dovoleno říkat jakýkoliv nápad, učitel povzbuzuje, cílem je aktivizovat. Teprve ve druhé části lze mezi nápady vybírat, třídit, některé zavrhovat - zde nastupuje kritické myšlení. Toto třídění provádějí žáci, učitel je v této činnosti pouze vede. Výsledkem této diskuse má být žákovo pochopení požadavků učitele na výslednou práci a ujasnění si cesty, jak projekt realizovat.

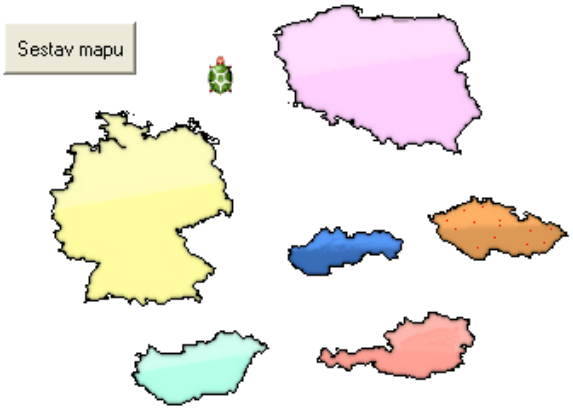
Na konci přípravy projektu ve třídě učitel jednoznačně stanoví kritéria (co má projekt obsahovat, kdy má být dokončen), klade podmínky podobně jako zákazník, který zadává zakázku. Jasné a stručné vyjádření přispívá k rychlému startu projektu (žáci si lépe naplánují svoji činnost) a pomáhá také při jeho vyhodnocení. Do kritérií lze přidat i některé velice konkrétní a specifické požadavky, aby bylo zaručeno, že žák se bude věnovat určité činnosti, použije vhodné nástroje a software, použije určité dovednosti; konkrétní kritéria by však neměla převládat, aby se zachoval charakter projektové činnosti.

Vlastní realizace projektu

V této části projektu učitel ustupuje do pozadí, v popředí je žák (nebo tým) a jeho práce na projektu. Učitel přejímá roli konzultanta a hlídače času, aby udržel směr a tempo (často se stane, že se žáci příliš pohrouží do dílčích činností a projekt by nestihli celý vypracovat). Je nutné ovšem vědět, že právě při těchto dílčích činnostech se žáci intenzívně učí na problémech, které přinášejí reálné situace.

V úvodu každé vyučovací hodiny je možné konkretizovat cíle (např. „nyní bychom měli dokončovat ... a je potřeba, abyste začali ...“). Po dokončení určitých etap je možno dosavadní práci vyhodnotit a stanovovat další dílčí cíle atd. Realizaci projektu je možno, je-li to potřebné, přerušit a vložit krátké pasáže nové látky, seznámení s novými pojmy nebo zopakování, a poté v projektové činnosti navázat. Výuka nové látky je často efektivnější standardními metodami.

Učitel musí být připraven, že projekt může přinášet „zapeklité“ problémy „jak něco udělat“, že bude žáky dotázán na velmi pokročilý nebo netradiční problém. Bude muset posoudit, zda si žáci nestanovili příliš vysoký úkol, zda k cíli nepovede jednodušší cesta. Nemusí tedy pouze radit, „jak na to“, ale také ukázat směr další práce.



Obr. 2 Mapa států střední Evropy, programovací projekt. Požadavek učitele může znít, že mapa se na „požádání“ sama zkontroluje nebo správně složí.

Dokončení projektu

Nesmírně důležitá část, ve které žáci dokončí projekt se všemi detaily, odevzdají a prezentují projekt. K úplnému dokončení je potřeba dostatek času - celý projekt by neměl smysl, kdyby nebyl zcela dokončen. Velmi důležitou součástí této fáze je zhodnocení projektu (publikování, předvedení před třídou, komentář učitele, hodnocení ostatních žáků či týmů, v případě mezipředmětových aplikací využití v jiném předmětu).

Velký význam má praktické uplatnění výsledků v životě školy či komunity, které může vést k aktivnějšímu zapojení se žáků do společenského života. Např. výsledky ankety o kvalitě stravy v jídelně nebo zpracování výzkumu míry ekologického chování občanů obce v nakládání s odpadky by neměly být „pouze vytvořeny, oznámkovány a zapomenuty“.

4. HODNOCENÍ PROJEKTU

Oproti jiným formám „zkoušení“ při výuce informatiky (vyhodnocení testu, kontrola samostatných prací žáků obodováním) je hodnocení projektu náročnější. Je sice pravda, že zaškrtáváním testových odpovědí žáci neprokazují počítačové dovednosti, je také pravda, že bodování řady krátkých jednoduchých úloh u samostatné práce neověří žákův přehledu v problematice a jeho

tvořivost, ovšem způsob hodnocení „sečtením bodů“ je rychlejší a pohodlnější. Jinými slovy, i když test ani obodovaná samostatná práce neprověří žákovu kvalitu (a je otázkou, zda zkouší opravdu reprezentativní dovednosti), nabízí snadnější formu hodnocení než projekt. Ovšem lapidárně řečeno, učitel by ovšem neměl hodnotit to, co je snadné hodnotit, ale to, co má smysl hodnotit [5].

Žákova práce na projektu, často dlouhodobá, musí být adekvátně ohodnocena, ovšem komplexnost projektu to neusnadňuje. Učitelé si nevědí rady především s klasifikací výsledné práce, proto často poskytují velmi povrchní nebo zcela subjektivní hodnocení („je to pěkné, mně se to líbilo“) bez výrazné zpětné vazby (žák se nedozví, co udělal dobře, v čem se má ještě zlepšit). Ani hodnocení, které některá vyjmenovaná kritéria převede na bodování, čímž převede hodnocení projektu na učiteli známé hodnocení samostatné práce, neumožňuje dostatečnou zpětnou vazbu; informace, že žák získal 22 ze 30 možných bodů, kvalitní reflexi neposkytuje.

Dalo by se říci, že projekt má několik „dimenzí“, jejichž hodnocení nelze jednoduše sčítat. Nelze do jedné škály promítnout žákovu znalost software, se kterým pracuje, odbornou správnost, originalitu řešení, pečlivost či schopnost reagovat na připomínky během realizace. Projekt má být zhodnocen komplexně, proto bodování ani pouhé hledání chyb není optimální metodou.

Pro hodnocení projektů můžeme použít metodu *tří S* [5]:

- *Splnění* říká, zda byl zadaný úkol dokončen, zkompletován a odevzdán (prezentován), zda splňuje zadání (tedy jsou-li splněna právě ta kritéria, vyjmenovaná na začátku projektu).
- *Správnost* hodnotí, zda práce obsahuje správné poznatky a úvahy, zda žák předvedl hodnocené dovednosti, zda je při práci čerpáno z více než jednoho zdroje, zda práce odpovídá normám pro tuto práci, zda žák nedělal věcné či další chyby, zda výsledky práce správně prezentoval.
- *Souhrnnost* říká, zda je na práci vidět pečlivost v myšlení a hledání řešení, přehled v oboru, zda jde rozeznat, zda žák rozumí pojmům, objevujícím se v práci, zda práce není jednostranná, nakolik je kreativní, nakolik se projevil vlastní přístup, iniciativa či netradiční řešení.

Hodnotí-li učitel třeba studii o zvycích žáků o přestávkách, kritérium splnění říká, že žák studii včas a řádně odevzdal a také před třídou s prezentací vystoupil, že splnil kritéria (např. použil tabulkový procesor ke zpracování a webový dotazník k získání dat, prezentace výsledků obsahuje rozbory a grafy). Kritérium správnosti hodnotí správné použití software, kvalitu otázek pro respondenty výzkumu, správné vytvoření osnovy závěrečné zprávy a její naplnění obsahem bez věcných, gramatických a typografických chyb, na její srozumitelnost, zda při závěrečné zprávě měl žák své vystoupení dobře připraveno (např. zda pouze nečetl promítaný text, zda věděl, co má k jednotlivým snímkům hovořit, zda animace a přechody snímků nepřekážely). Kritérium souhrnnosti hodnotí originální nápady použité ve výzkumu a jeho zpracování, ale také, s jakým zaujetím žák při projektu pracoval, jak spolupracoval s učitelem, zda prokázal přehled v problematice použití daného typu software, jaký byl celkový dojem z jeho vystoupení a jak kvalitně reagoval na připomínky a dotazy apod.

Sdělení učitelova hodnocení projektu žákovi podle takových kritérií má velký zpětnovazební efekt, větší než klasifikace. Nese totiž potenciál zkvalitnění žákovy práce na budoucích projektech, naučí jej projektově pracovat. Je však navrhované hodnocení projektu podle *tří S* objektivní? Odpověď zní: není, ovšem trápit to učitele nemusí. Ostatně žádné hodnocení není objektivní, protože i při bodování se učitel rozhoduje, kdy ve sporných situacích bod udělí a kdy nikoliv, rozhoduje, která dovednost bude hodnocena, která zaslouží vyšší bodové ohodnocení atd.

Jak ale klasifikovat projekt, jestliže není k dispozici žádná bodová škála? V řadě případů stačí rozhodnout (a zdůvodnit), zda je projekt splněný, tedy uznatelný, a rozeznat mimořádně zdařilý projekt. Na pomoc učiteli přicházejí právě kritéria, jmenovaná v metodě *tří S*. U skupinového projektu pak zbývá rozlišit podíl jednotlivých členů týmu na výsledném díle (zde je nutno, aby si členové týmu rozdělili nebo dostali přidělenu zodpovědnost za jednotlivé dílčí části).

Obr. 3 Ankety s hromadným zpracováním dat jsou vděčnými a žáky oblíbenými tématy, je ovšem se třeba citlivě ohlídat choulostivé náměty, tak aby výsledky nepřispívaly ke zhoršení klimatu ve škole.

5. SPECIÁLNÍ TYPY PROJEKTŮ

V této kapitole představíme několik typů projektů, které mají některá specifika. Jsou však již určeny pro učitele, kteří mají s vedením projektů nějaké zkušenosti.

Skupinové projekty

Každý projekt nutně nemusí být skupinový; pro učitele, který s projektovou výukou začíná, je mnohem snadnější zorganizovat projekt, v němž žák pracuje individuálně. Při týmové práci se navíc žáci musí naučit spolupracovat, tedy komunikovat, nést odpovědnost za celý tým, umět si uvnitř týmu rozdělit práci a zodpovědnost.

Skupinové projekty lze roztřídit podle způsobu vytvoření a organizace pracovních skupin na:

- Homogenní skupiny různé úrovně
- Heterogenní skupiny stejné úrovně

Homogenní skupiny jsou vnitřně neroztříděné, všichni jejich členové „dělají totéž“, ovšem každá z těchto skupin, na které je třída rozdělena, má odlišnou práci, funkci, úkol. Často každá ze skupin plní část rozsáhlejšího úkolu, skupiny tak spolu musí kooperovat (třeba prostřednictvím učitele).

Např. v projektu „Jak zelená je naše třída?“, v němž mají žáci zjistit řekněme „ekologickou kvalitu“ jejich učebny. Třída je pak rozdělena na týmy různé povahy: botaniky, zoology, geometry, inženýry. Zatímco botanici např. popisují počet a stav flóry v jejich třídě (i té nežádoucí), geometři mají za úkol změřit rozměry a zjistit množství vzduchu ve třídě, zda odpovídá normám, a spočtou zatížení atmosféry učebny v „člověkohodinách“ (tedy v počtech žáků během dne, kteří třídu využijí), a inženýři zjišťují třeba velikost oken a dostatečnost osvětlení a porovnávají s normami. Všichni společně pak sestavují zprávu (a případně analyzují a porovnávají se zprávou z jiné školy, jde-li o projekt meziškolní či mezinárodní).

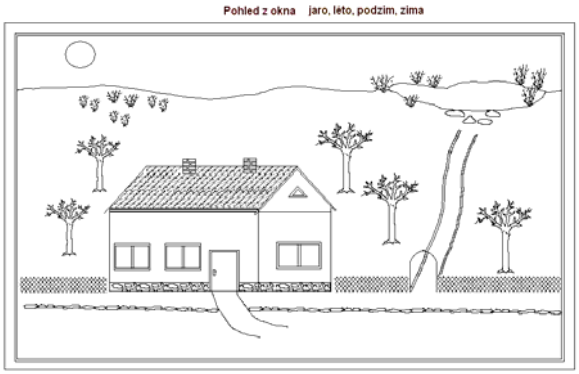
Heterogenní skupiny stejné úrovně jsou často skupinami, které spolu soupeří. Všechny týmy pracují na tomtéž úkolu a nevyžaduje se vzájemná kooperace. Jiné je to ale uvnitř týmů. Jednotliví členové skupin mají své funkce a stanovené kompetence.

Představme si projekt „Soutěž o veřejnou zakázku“, v níž jednotlivé žákovské týmy (firmy) připravují projekt třeba pro výstavbu sportovní haly; tyto týmy se utkají o „zakázku“, nejkvalitnější návrh vítězí. Týmy jsou vnitřně strukturované, každý člen týmu má určenu nějakou pozici (např. velitel, komunikátor, prezentátor, kreativec, zapisovatel a dokumentarista, logistik). *Velitel* je ten, kterého ostatní respektují, který rozděluje práci, v případě sporů rozhoduje a především hlídá čas. *Komunikátor* je jediný člen týmu, který se smí vzdálit, přijmout instrukce od učitele či se s ním jít poradit (tzn. že pokud některý člen týmu něčemu nerozumí a ostatní mu neporadí, musí svůj problém vysvětlit komunikátorovi, ten se zeptá učitele a poté musí odpověď srozumitelně tlumočit v týmu). *Prezentátor* je ten, který hovoří za tým veřejně, prezentuje výsledky a účastní se veřejných debat nebo společných aktivit všech týmů (třeba „dražby“ nebo „konference“). *Kreativci* se naproti tomu nemusí zabývat administrativou, pracují pouze na odborné části projektu. I nejslabší žák v týmu může mít velice důležitou roli *Logistika* (péče o materiál, jako je zálohování, příprava a rozdílení materiálů pro společnou práci, jejich schraňování před ztrátou nebo konkurencí). Taková role má pro slabšího žáka, který nemůže být „mozkem“ týmu, velký význam, protože nese zodpovědnost za tým a může být svou pečlivostí a rychlostí svému týmu velice užitečný.

Projekt etapový

Jde o projekt, skládající se z několika relativně samostatných částí (etap), které mohou být postupně obtížnější, zpřesňující nebo konkretizující. Např. u projektu „Redakce novin“ žák postupně vyzkouší praktický přijímací test do redakce, práci jazykového korektora, reportéra, sazeče, fotoreportéra, redaktora rubriky a na závěr projektu skupinově vytvoří výtisk novin (i s obsazenou pozicí šéfredaktora, případně s konkurzem na toto místo).

Jako příklad popíšeme organizaci skupinového projektu „Pohled z okna“, v němž žáci vytváří sadu obrázků téže krajiny v různých ročních obdobích. Žáci pracují v týmech po čtyřech žácích, projekt se skládá ze dvou etap. V první etapě členové týmu pracují individuálně, soutěží mezi sebou, kdo dokáže nakreslit imaginární krajinu tak, aby na ní bylo možno dodatečně dokreslit co nejvíce rozlišovacích znaků určitého ročního období. Žák např. nemůže do svého obrázku nakreslit strom s listy, protože takový obrázek by se nedal použít pro zimní období; strom bez listí je akceptovatelný – listy se mohou dokreslit později. Na konci 1. etapy členové týmu rozhodnou, který obrázek je nejlepší a bude reprezentovat tým; ten poté poslouží jako šablona pro vytvoření sady obrázků pro jaro, léto, podzim a zimu (každý člen týmu kreslí jedno období). Protože tým soutěží s ostatními, členové se scházejí k poradám a pomáhají si nápady. Výsledná sada obrázků se vytiskne nebo poslouží jako podklad pro čtvrtletní kalendář.



Obr. 4 Obrázek krajiny jako žákovská práce po 1. etapě projektu "Pohled z okna" slouží jako šablona pro dokreslení jednotlivými členy týmu pro různá roční období.

Projekt založený na komunikaci s okolím

Komunikace např. e-mailem, zvláště mezi školami v různých státech, poskytuje bezprostřední kontakt mezi žáky různých škol i kultur, potřebu domluvit se v některém světovém jazyce, nabízí ke zpracování data, která zjistili nebo naměřili žáci jinde (jsou to tedy "živá" údaje, nikoliv nějaká čísla z učebnicových příkladů). Projekty využívající komunikaci mezi kulturně odlišnými prostředími kromě společné práce nabízejí možnost přispívat k poznání světa i sebe sama, k překonávání xenofobie nebo kulturního etnocentrismu (kdy člověk povyšuje normy chování a kulturní zvyklosti své a svého okolí nad normy běžné v jiných zemích).

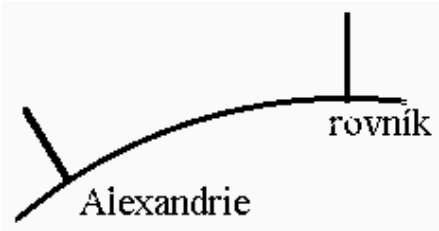
Například, při projektu "Weathernet" žáci dlouhodobě měřili teplotu, srážky a sílu větru a pravidelně si tyto údaje vyměňovali z desítkou jiných škol po celém světě, aby je poté mohli porovnat. Přitom získali nejen řadu fyzikálních a zeměpisných poznatků (např. že uprostřed Spojených států jsou vlivem kontinentálního počasí skutečně krutější zimy než na pobřeží Aljašky), ale z pravidelných týdenních zpráv se dozvídali pro ně někdy nesrozumitelné údaje, které museli nejprve dekódovat. Např. podivná zpráva, že na Floridě bylo minulý týden dost chladno, pouze 50 stupňů, byla natolik motivující, že vedla k pátrání a objevení Fahrenheitovy teplotní stupnice (a poznání, že se ve světě používají i jiné fyzikální jednotky než naše) podobně jako informace, že v Juneau na Aljašce napadly tři stopy sněhu (a kolik to je centimetrů).

K podobným poznáním vedou i jednoduše pojaté dopisovací projekty. Když si například před třinácti lety naši a kanadští šesťáci vyměňovali informace, co se právě učí v přírodopisu, moc si navzájem nerozuměli, co se ti druzí vlastně učí. V Kanadě právě probírali sociální chování a potravní řetězce zvířat v lese, u nás to bylo trávicí ústrojí kočky.

Mezi projekty, v nichž hraje komunikace hravní roli při získávání měřených dat, můžeme jmenovat např. projekt Eratosthenes, který se nechal inspirovat starořeckým matematikem, který z rozdílu délky stínu o rovnodennosti mezi Alexandrií a rovníkem jako první odhadl obvod zeměkoule. Děti měřily tzv. „stínový úhel“, tedy úhel mezi paprsky slunce a svislicí v poledne o rovnodennosti, kdy na rovníku svislá tyč nevrhá žádný stín. V závěru projektu diskutovaly, proč změřené hodnoty stínového úhlu odpovídají zeměpisné šířce školy.

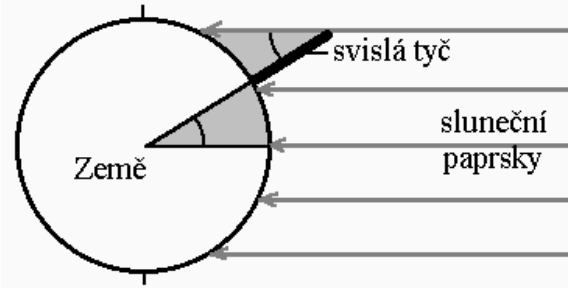
Význam takových projektů spočívá mj. v tom, že žáci pracují se skutečně změřenými údaji, s „živými“ hodnotami zjištěnými jinými dětmi v jiných zemích, a ne v počítání s tvůrci učebnic uměle vytvořenými čísly, sloužícími k procvičování probrané látky.

Při komunikaci s cizincem si člověk uvědomí svoji identitu, příslušnost k místu bydliště, k národu. Zajímavé bylo pozorovat, jak jsou žáci schopni prezentovat sebe sama. Když měli popsat svoji osobu, většinou psali: moji rodiče pracují tam a tam, mám dva sourozence, máme psa, rád poslouchám hudbu apod. Rozdíl v pojetí sebeprezentace mezi dětmi z Českých Budějovic a z Atlanty byl markantní: každý žák 5. třídy z atlantské Pepperell Elementary School byl něším vyjímečný: nejlepší matematik třídy, nejlepší házenkář třídy, největší fanda místního klubu amerického fotbalu apod., dokonce i žák, který má nejdelší chodidla v celé třídě. Našim žákům připadaly jejich výpovědi chudé, utěšovalo je aspoň vědomí, že američtí páťáci se teprve učili psát psacím písmem (a dopisy, které byly psané, psaly tužkou).



Worldwide Eratosthenes Experiment Results

School	Location	Lat.	Long.	Shadow Angle
Lakewood High School	Lakewood, Ohio	41-30'N	81-45'W	43
Balboa High School	Republic of Panama	8-57'N	79-33'W	8.3
International School Amsterdam	The Netherlands	52.6'N	4.8'E	55.2
Tom Duncan	Braintree, Massachusetts	42.14'N	71W	46
Moorhead Junior High School	Moorhead, Minnesota	47 N	97 W	50
Bob Powell-West Georgia College	Carrollton, Georgia	33-34'N	85-06'W	31.1
Gordon Central High School	Calhoun, Georgia	34-30'N	85 W	32.2
Alexander High School	Douglasville, Georgia	33-41'N	34-47'W	34
Central High School	Carrollton, Georgia	33-33'N	85-04'W	33
Lithia Springs High School	Lithia Springs, Georgia	33-45'N	84-47'W	32.7
Marietta High School	Marietta, Georgia	33-55'N	84-41'W	32.7
Heard County High School	Franklin, Georgia	33-17'N	85-06'W	31.3
Jordan High School	Los Angles, California	33-56'N	118-16'W	34
Woodbridge High School	Bridgeville, Delaware	38-45'N	75-36'W	34.4
Guy B. Phillips MIddle School	Chapel Hill,NC	35-55'N	79-05'W	34.5
Laura Mercer	Columbia, MO	38-56'N	92-20'W	50
Harrison County High School	Cynthiana, Kentucky	38.20N	84.20W	41.25
Parkview Arts/Science M.	Little, Arkansas	34.73N	99.33W	34.4



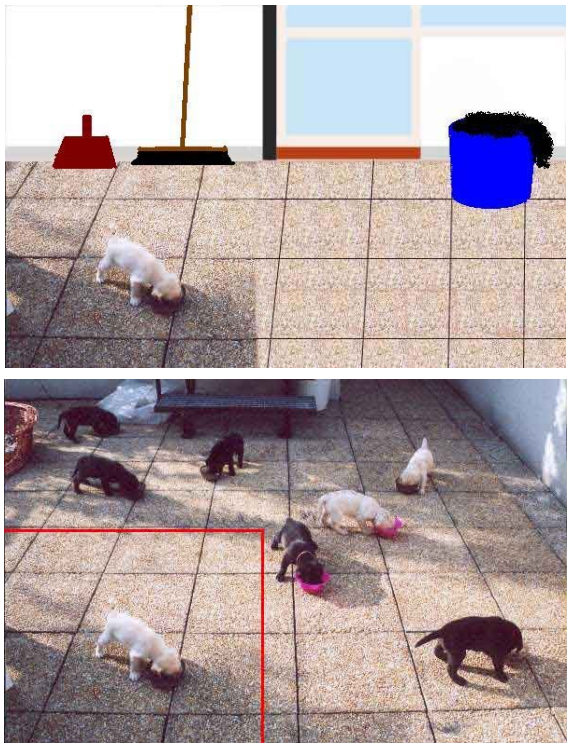
Obr. 6 Projekt Eratosthenes, porovnávání změřeného tzv. „stínového úhlu“ se zemepisnou šířkou ve stovkách škol po celém světě.

Programovací projekt

Vynikající a přirozenou formou vzdělávání žáků v oblasti informačních a komunikačních technologií jsou projekty programovací, protože dokážou obsáhnout a intenzívně využívat celé řady kompetencí získaných jak v předchozí výuce programování, tak technologických a matematických znalostí či dovedností v práci s grafikou a schopnosti plánovat, být důsledný a vytrvalý, pružně reagovat na situaci, řešit problémy a být vynalézavý.

Pro programovací projekty je potřeba mít pracoviště vybaveno kvalitním didaktickým programovacím prostředím, které je uživateli srozumitelné a dostatečně jednoduché, pomáhá mu učit se programovat správnými postupy a přitom jej motivuje možností, že výsledný programový projekt (např. jednoduchá počítačová hra) bude moci jeho tvůrce šířit jako jiný běžný program, který bude moci spustit na každém počítači (nebo jako část webové stránky).

Programovací projekty a jejich aspekty by byly tématem na celý další článek, zde chceme pouze upozornit na jejich důležité místo mezi žákovskými projekty ve školní informatice.



Obr. 6 Projekt Části fotografií, autor Zdeněk Karas. Z části původní fotografie (dole) se část „uřízne“ a žáci vyretušují obrázek (nahoře) s cílem, aby porovnání s původní fotografií způsobilo údiv.

6. ZKUŠENOSTI S PŘÍPRAVOU UČITELŮ

V této kapitole popíšeme naše zkušenosti s přípravou studentů učitelství informatiky v rámci běžného vysokoškolského kurzu a s učiteli z praxe, navštěvující kurzy dalšího vzdělávání učitelů, v oblasti jejich přípravy na vedení projektové výuky. Utřídíme a budeme komentovat chyby, kterých se při tvorbě příprav dopouštějí.

Studenti mají jako úkol vytvořit vlastní přípravu na několikahodinový žákovský projekt na vlastní téma, což představuje rozmyšlení obsahu a cílů výuky, časového rozvržení a organizace výuky, kritérií pro výsledný produkt a jeho hodnocení, přípravu učebních pomůcek a vytvoření vzorového žákovského řešení (jak si učitel představuje vzorově splněný úkol). Toto vzorové řešení je důležité z toho důvodu, že hodnotiteli projektu dává konkrétní představu o studentově záměru, neboť samotné zadání projektu bývá nepřesné, nesrozumitelné nebo příliš obecné, šablonovité. Teprve při tvorbě vzorového řešení si student někdy uvědomí určité nedostatky (nároky na dovednosti žáků i náročnost časová, organizační návaznost při skupinové práci) a může je včas korigovat.

Chyby, kterých se studenti při tvorbě zadání žákovských projektů dopouštějí, lze rozdělit následujím způsobem:

- **Výběr tématu**. Studenti mívají potíže vybrat téma projektu tak, aby v něm byla vyvážena složka technologická a netechnologická. Takové projekty, jinak zajímavé, často použijí počítač pouze jako psací stroj nebo zařízení pro tisk digitálních fotografií.
  Jiným extrémem je, když projekt se svým tématem nachází celý v oblasti informačních technologií, což potlačuje interdisciplinaritu (výroba prezentací o nějakém software, o nabídce komponent na trhu, tvorba webových stránek o tvorbě webových stránek apod.)
- **Není to projekt**. Studenti dělají chyby popsané v článku výše: projekt připomíná sled přesně popsaných úloh, jde o jakousi delší samostatnou práci podle návodu. Někdy studenti formálně přejímají znaky přípravy projektu (např. popisují, jak děti diskutují nad zadáním, ale kritéria na výsledný produkt spolu s pokyny pro práci jsou nastaveny tak, že žák nemá možnost si zadání přizpůsobit nebo jeho možnosti svoji práci sám organizovat jsou potlačeny).
  V extrémních případech „projekt“ připomíná učitelův plán výuky nějakého tématického celku.
- **Formální stránka**. Studenti mají problémy vytvořit správnou strukturu návrhu projektu. Velmi často chybí srozumitelná anotace, z níž by se dalo rychle zjistit, o jaký typ práce se jedná a co děti vytvoří (tyto informace je nutno hledat po celém dokumentu a nezřídka se dají zjistit až z náhledu vzorového žákovského řešení). Podobnou chybou je nekonkrétnost popisu projektu. Studenti vytvářejí jakési šablony příprav, do nichž by se daly doplnit konkrétní informace (ale právě nad těmito konkrétními údaji je třeba hluboce přemýšlet). Zadání projektu pak vypadá spíše jako souhrn metodických rad a připomínek než příprava konkrétní činnosti.
- **Špatně definované cíle výuky**. Studenti teoreticky vědí, že cílem výuky jsou kompetence, které žák získá během výuky, ovšem nedokáží je v daném konkrétním projektu odhalit. Vypadá to, že pro učitele je cílem projektu také vytvořený produkt, nikoliv ty kompetence, dovednosti a znalosti, které žák během tvorby projektu

trénuje.
Jiným problémem je, že studenti často pěkně definují cíle v oblastech mimo práci s počítačem (např. z oblasti vyučovacího předmětu, kterého se téma projektu týká, z oblasti tvořivosti, pochopení pojmů a faktů, organizace), zatímco popis „počítačových“ kompetencí je slabý, velice obecný, nekonkretizovaný (např. „žák si procvičí práci s MS Word“). Bez stanovení konkrétnějších cílů učitel na konci projektu může jen obtížně kontrolovat, zda jsou splněny či nikoliv.

- **Problémy s hodnocením** úzce souvisí se stanovením cílů. Studenti často nezahrnují do hodnocení podstatné složky, trénované právě při projektové výuce. Často se omezují na obecná kritéria („celkový dojem“, „úroveň zpracování“), která neusnadní učiteli vlastní hodnocení. Kritéria, podle kterých se projekt bude hodnotit a která musí být žákům řečena předem, pak nejsou žákům srozumitelná, nevedou je k cíli a neohraničují při práci pole jejich působnosti, které se tak může stát bezbřehé.
  Studenti také často přemýšlení o kritériích, podle kterých budou hodnotit, zaměňují za stanovování proporcionality jednotlivých kritérií (jakoby je více zajímá klasifikace, tedy jak stanovit výslednou známku, než podle čeho se projekt bude hodnotit).
- **Vzorové žákovské řešení**. Studenti učitelství často vytvářejí toto řešení jen „jako“, např. místo psaní textů, které po žácích v projektu požadují, je sami kopírují z Internetu apod. Pak nemají žádnou představu o náročnosti tohoto projektu a potřebné časové dotaci. Odbyté žákovské řešení se projevuje velkým výskytem gramatických a typografických chyb – učitelé mají dojem, že oni sami nemusí dodržovat pravidla, která vyžadují po svých svěřencích. Učitelé z praxe zase často místo svých vzorových řešení přinášejí projekty, které ve škole vytvořili jejich žáci. Tyto „reálné“ práce ovšem obsahují chyby nebo ne zcela odpovídají zadání, které učitel připravil, nemohou sloužit jako ideál, ke kterému má výsledný produkt projektu směřovat.
- **Reálnost realizace projektu**. Někteří studenti se nechají unést vizí vytvoření ambiciózního často skupinového nebo etapového projektu, se kterým nemají zkušenosti. Takový projekt je zpravidla nerealizovatelný (např. projekt redakce elektronického blogovacího časopisu a několika rolemi, v nichž se mají žáci vždy po měsíci vystřídat, takže celý projekt s tímtéž tématem trvá třeba půl roku bez významného progresu v učení).
  Problém organizace skupinového projektu často spočívá v nutnosti zorganizovat následnost činností (co budou dělat sazeči, než jim reportéři dodají texty a fotografie). Právě dobře provedené žákovské řešení usnadní sestavení časového harmonogramu projektu.

7. ZÁVĚR

Práce na projektech rozvíjí jiné stránky osobnosti žáka než tradiční výuka. Učitel informatiky může mít pocit, že ty „počítačové“ dovednosti, které má v žácích trénovat, se při práci na projektu netrénují tak intenzívně, jako při jiném typu výuky. Je třeba říci, že projektová výuka není primárně určena v nabytí nových dovedností z oblasti IKT, jako spíše k jejich aplikaci a k rozvíjení dalších kompetencí, přesahujících rámec této vzdělávací oblasti.

Je dobrou zprávou, že informatika jako školní předmět dokáže projektovou výuku přirozeně zahrnout do svého kurikula; v podstatě každý tématický celek může být zakončen projektem. Projekty ukáží dětem odlišný styl práce, při němž mohou více přejímat zodpovědnost za sebe i a za práci jiných. Projektová výuka má předpoklady stát se běžnou součástí školní výuky, bez nádechu něčeho zvláštního, netypického. Vyžaduje ovšem připraveného učitele.

LITERATURA

[1] ČERNOCHOVÁ, M., KOMRSKA, T., NOVÁK, J. *Využití počítače při vyučování. Náměty pro práci dětí s počítačem.* Praha: Portál, 1998. ISBN 80-7178-272-6.

[2] SCHUBERT, S., SCHWILL, A. *Didaktik der Informatik*. Heidelberg: Spektrum, 2004. ISBN 3-8274-1382-6

[3] VANÍČEK, J. *Metodická příručka Informatika pro základní školy a víceletá gymnázia, 2. díl*. Brno: Computer Press, 2005. ISBN 80-251-0631-4

[4] VANÍČEK, J. O projektové výuce ve školní informatice. *Matematika, informatika, fyzika,* roč. 17, č. 32, s. 15 – 22. ISSN 1335-7794

[5] KOVALIKOVÁ, S. *Integrovaná tematická výuka: model*. Kroměříž: Spirála, 1995. ISBN 80- 901873-0-7

Vybrané vedecké prednášky

Selected Research Papers

VYHODNOTENIE RELIABILITY HODNOTENIA OLYMPIÁDY V INFORMATIKE

RNDr. Michal Forišek
Katedra informatiky, Fakulta matematiky, fyziky a informatiky Univerzity Komenského, Mlynská Dolina, 842 48 Bratislava
e-mail: forisek@dcs.fmph.uniba.sk

Pomocou Item Response Theory navrhujeme teoretický model na vyhodnotenie reliability výsledkov Olympiády v informatike. Navrhnutý model používame na konkrétnych dátach pre krajské kolá 22. a 23. ročníka Olympiády v informatike.

Kľúčové slová: *item response theory, olympiáda v informatike, obtiažnosť úloh, reliabilita hodnotenia*

1. ÚVOD

Cieľom výskumu prezentovaného v tomto článku bolo analyzovať reliabilitu výsledkov žiakov v Olympiáde v informatike a formulovať závery, ktoré povedú k jej zlepšeniu.

V kapitolách 2, 3 a 4 prezentujeme nami použitý teoretický model, následne v kapitole 5 uvádzame analýzu konkrétnych dát pre školské roky 2006/07 a 2007/08.

2. MODEL OBTIAŽNOSTI ÚLOHY

Našu analýzu postavíme na modernej teórii testovania – Item Response Theory (IRT). V tejto kapitole uvedieme stručný prehľad použitých výsledkov, ako aj vlastné úpravy existujúcich modelov. Podrobnejší popis tejto teórie nájdete napr. v [1] a [5].

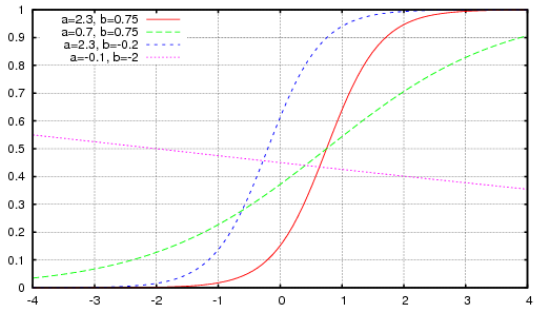
(Lingvistická poznámka: Stretli sme sa s viacerými rôzne zlými slovenskými prekladmi, asi najlepší je „teória odpovedí na položku“.)

Základným predpokladom, z ktorého budeme vychádzať, je, že každý z subjektov, v našom prípade teda súťažiacich žiakov, má určitú úroveň schopnosti riešiť algoritmické úlohy. Budeme predpokladať, že úroveň tejto schopnosti subjektu *s* sa dá popísať reálnym číslom, ktoré budeme označovať θ_s. Keďže ide o latentnú schopnosť, hodnoty θ_s nevieme merať priamo, ale vieme získavať ich odhady prostredníctvom testov.

Náš model obtiažnosti úlohy postavíme na 2-parametrovom logistickom (2PL) modeli. Klasický 2PL model je navrhovaný pre úlohy s binárnym (v teórii testovania sa niekedy používa pojem „dichotomickým“) hodnotením – subjekt buď úlohu vyriešil alebo nie. V klasickom 2PL modeli popisujeme obtiažnosť úlohy dvoma parametrami: koeficient obtiažnosti *b*, a koeficient diskriminácie *a*. Pre úlohu s parametrami *(a,b)* je pravdepodobnosť, že ju subjekt so schopnosťou θ vyrieši, daná logistickou funkciou:

$$Pr(\theta, a, b) = \frac{1}{1+e^{-a(\theta-b)}}$$

Príklady priebehu tejto funkcie pre rôzne dvojice *(a,b)* sú znázornené na obr. 1.



Obr. 1 Logistická funkcia pre 2PL model.

V našom prípade tento model ale nemôžeme priamo použiť, keďže Olympiáda v informatike používa iné ako binárne hodnotenie. Model, ktorý použijeme, by mal čo najpresnejšie odzrkadľovať použitý spôsob hodnotenia.

Úlohy v Olympiáde v informatike sú hodnotené na diskrétnej škále od 0 do 10 bodov, primárnym kritériom hodnotenia je efektívnosť nájdeného algoritmu. Pre veľkú väčšinu zadaných úloh (vrátane tých skúmaných v nasledujúcej kapitole) platia nasledovné hranice:

- každé korektné riešenie získa aspoň 4 body
- každé efektívne riešenie aspoň 7 bodov
- riešenia podobné vzorovému aspoň 9 bodov

Riešenie úlohy väčšinou vyžaduje viacero myšlienkových krokov, ktoré vedú k postupnému zlepšeniu nájdeného riešenia. Na základe tohto pozorovania sme sa rozhodli modelovať takúto úlohu ako tri samostatné binárne hodnotené úlohy. Následne vieme očakávaný počet bodov (na škále 0-3, nie 0-10) vyjadriť ako súčet pravdepodobností vyriešenia jednotlivých podúloh:

$$E(Body(\theta)) = \sum_{i=1}^{3} \frac{1}{1+e^{-a_i(\theta-b_i)}}$$

3. FISHEROVA INFORMÁCIA

Vždy, keď pozorujeme hodnotu náhodnej premennej, získavame tým informáciu. Pri testovaní

môžeme túto informáciu použiť na získanie lepšieho odhadu skrytých parametrov systému. Intuitívne môžeme mieru informácie definovať ako prevrátenú hodnotu presnosti, s akou vieme daný parameter odhadnúť. Formálne túto intuíciu zachytáva Fisherova definícia štatistickej informácie ako očakávanej hodnoty variancie skóre. Známym výsledkom je, že pre 2PL model je:

$$I(\theta)=a^2 Pr(\theta,a,b)(1-Pr(\theta,a,b))$$

Znamená to teda, že zaradenie úlohy s parametrami (a,b) do testu nám poskytne najviac informácie pre subjekty, pre ktoré je $Pr(\theta,a,b)=1/2$, a teda $\theta=b$.

Keďže predpokladáme, že výsledky subjektu pre rôzne položky sú navzájom nezávislé náhodné premenné, platí, že informácia získaná z rôznych položiek testu je aditívna.

Zamerajme sa teraz na krajské kolo Olympiády v informatike. Z tohto kola najlepších približne 30 riešiteľov z celého Slovenska postupuje na celoštátne kolo. Reliabilita krajského kola je teda schopnosť čo najpresnejšie identifikovať 30 súťažiacich, ktorí sú v riešení algoritmických úloh najlepší. Znamená to teda, že našim cieľom pri výbere súťažných úloh by malo byť primárne vytvoriť takú sadu úloh, ktorá nám umožní čo najpresnejšie určiť schopnosti subjektov v okolí postupovej hranice, t. j. 30-teho miesta.

4. ODHAD PARAMETROV

Pochopiteľne, aj keby náš teoretický model presne popisoval realitu (čo samozrejme nie je pravda), presné parametre popisujúce obtiažnosť úloh nám nik nedá. Najbežnejším spôsobom, ako približné hodnoty týchto parametrov získať pred samotným použitím testu je kalibrácia – dáme úlohy testovacej vzorke subjektov, ktorých úrovne schopnosti považujeme za známe, a na základe ich výsledkov vypočítame najpravdepodobnejšie hodnoty parametrov úloh. Tento postup však v našom článku nepotrebujeme použiť, preto sa ním nebudeme podrobnejšie zaoberať.

Pri zodpovedaní našej otázky si vystačíme s posteriórnym určením parametrov úloh. Použijeme metódu maximum likelihood estimate.

(Keďže aj „likelihood“ aj „probability“ sa do slovenčiny prekladajú ako „pravdepodobnosť“, kvôli potrebe odlíšiť tieto dva pojmy budeme používať anglický pojem „likelihood“.)

Základná myšlienka tohto prístupu je nasledovná: Majme náhodnú premennú X, ktorej rozloženie pravdepodobnosti fy závisí od parametra y. Namerali sme premennú X a dostali sme výsledok x. V tejto situácii môžeme likelihood $L(y)$ parametra y definovať ako podmienenú pravdepodobnosť $P(X=x \mid y)$. Zdôrazňujeme, že hodnota $L(y)$ nie je pravdepodobnosť, že y je skutočná hodnota skrytého parametra. Hodnota y, pre ktorú funkcia $L(y)$ nadobúda maximum, sa nazýva maximum likelihood estimate premennej y.

V našom prípade budeme chcieť zostrojiť maximum likelihood estimate pre vektory parametrov θ_i, a_i, a b_i za predpokladu, ktorým sú výsledky subjektov pri riešení dotyčných úloh.

5. ANALÝZA DÁT Z PRAXE

Analyzovali sme dáta zodpovedajúce dvom školským rokom. V každom školskom roku sme mali k dispozícii 44 úloh – 4 série Korešpondenčného seminára z programovania (KSP) po 10 úloh a 4 úlohy krajského kola Olympiády v informatike (OI), kategórie A. Počet subjektov, ktoré sa do súťaží zapojili, bol 177 v šk. r. 06/07, resp. 201 v šk. r. 07/08.

Na modelovanie obtiažnosti úloh sme použili model popísaný v kapitole 2, teda každú úlohu modelujeme ako tri nezávislé binárne hodnotené položky. Vzorky odpovedí (response patterns) pre tieto položky sme zostrojili tak, že za správne vyriešenú sme položku označili u tých riešiteľov, ktorí za ňu získali dostatočne veľa bodov.

Napr. pre úlohu z OI prvú položku zodpovedajúcu úlohe správne vyriešili riešitelia, ktorí získali aspoň 3, druhú tí, ktorí aspoň 6, a aj tretiu tí, ktorí získali aspoň 9 bodov z 10. Pre úlohy z KSP, ktoré používalo škálu od 0 do 15 bodov, sme použili hranice 4, 8, resp. 13 bodov.

Na základe týchto vzoriek odpovedí sme zostrojili maximum likelihood estimates pre schopnosť θ_i každého z účastníkov a pre parametre a_i a b_i každej z položiek. Použili sme na to metódu hill climbing numerickej optimalizácie, náš program začal s základným odhadom že všetky θ_i sú rovné 0 a všetky $(a_i,b_i)=(1,0)$. Následne až do okamihu dostatočnej konvergencie striedavo dopočítal maximum likelihood estimate pre schopnosti a pre parametre položiek na základe predchádzajúceho odhadu pre opačný typ parametrov.

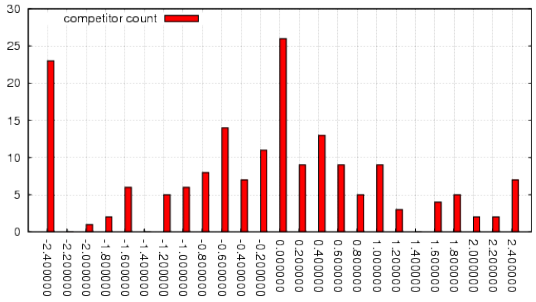
Použitá formula pre likelihood schopnosti θ konkrétneho subjektu na základe parametrov položiek a správnosti jeho odpovedí na ne:

$$L(\theta)=\Pi_{i=1}^{N} p_i^{s_i}(1-p_i)^{1-s_i}$$

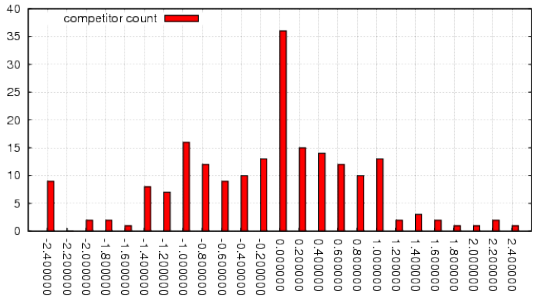
kde $p_i=Pr(\theta,a_i,b_i)$ a s_i je 0 ak položku i nevyriešil a 1 ak áno. Maximum likelihood estimate parametra θ sme počítali ako maximum tejto funkcie (presnejšie jej logaritmu, ktorý je aditívny, má maximum v rovnakom bode a výpočet je numericky stabilnejší.) Formula pre opačný krok vyzerá analogicky.

Na obr. 2a, 2b je vyššie uvedeným postupom získané rozloženie schopností riešiteľov pre jednotlivé školské roky. Všimnite si, že rozdelenie podľa očakávania nezodpovedá Gaussovej krivke: na ľavej strane je vidieť, že existuje skupina

riešiteľov, ktorí sa do súťaže zapoja, no úlohy sú nad ich sily. Hranica 30. miesta v celej vzorke subjektov zodpovedala v šk. r. 06/07 približne hodnote $\theta=0,95$ a v šk. r. 07/08 hodnote $\theta=0,8$. Hranica 30. miesta v množine účastníkov krajského kola bola v oboch prípadoch mierne nižšia.

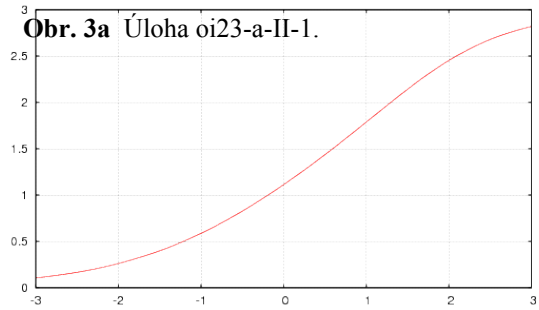


Obr. 2a Rozloženie schopností v šk. r. 06/07.

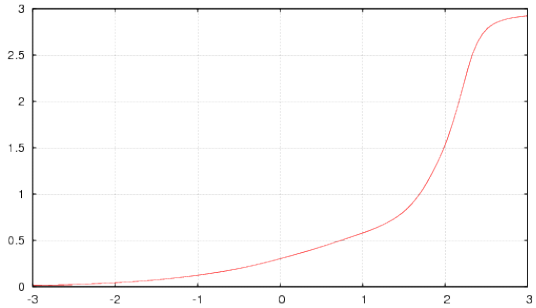


Obr. 2b Rozloženie schopností v šk. r. 07/08.

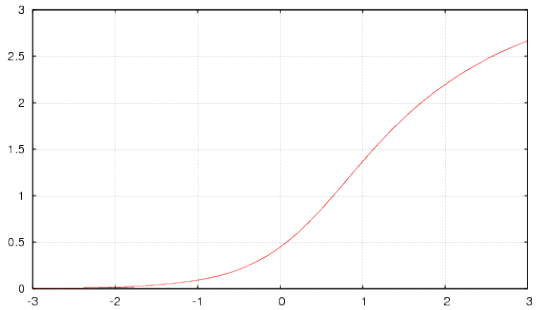
Na obr. 3a, 3b, 3c sú grafy znázorňujúce očakávaný počet bodov (na škále 0-3, nie 0-10) pre tri úlohy krajského kola OI v šk. r. 07/08. Vidíme, že priebehy týchto funkcií sú výrazne rozličné. V prvom prípade ide o relatívne ľahkú, vyváženú úlohu, v ktorej očakávaný počet bodov približne lineárne zodpovedá schopnosti riešiteľa. V druhom prípade ide o ťažkú úlohu, kde je prudko netriviálne nájsť ľubovoľné efektívne riešenie, v dôsledku čoho je očakávaný počet bodov vysoký len na samom konci spektra. V treťom prípade ide o teoretickú úlohu, ktorá vyžaduje určitú úroveň už na pochopenie zadania, preto očakávaný počet bodov začína byť nenulový až v okolí $\theta=0$.



Obr. 3a Úloha oi23-a-II-1.

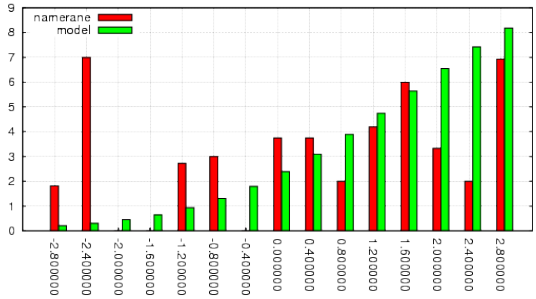


Obr. 3b Úloha oi23-a-II-3.

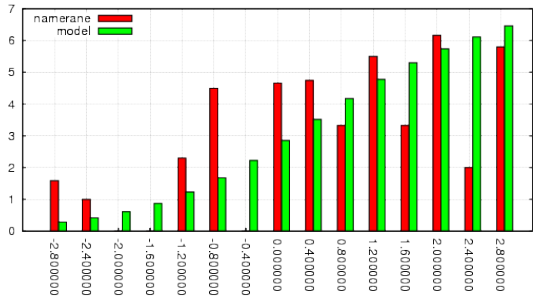


Obr. 3c Úloha oi23-a-II-4.

Na obrázkoch 4a, 4b je porovnanie skutočného priemerného počtu bodov a očakávaného počtu bodov pre dve vybrané úlohy. Riešiteľov sme rozdelili do skupín podľa ich odhadnutých schopností. Výrazné odchýlky reality od hodnôt predpovedaných modelom (pre $\theta=-2,4$ a $\theta=2,4$ na obr. 4a, resp. $\theta=2,4$ na obr. 4b) všetky zodpovedajú skupinám, do ktorých padol len jeden riešiteľ. Skupiny, ktoré v grafe nemajú vôbec znázornený priemerný počet bodov, sú prázdne.



Obr. 4a Úloha oi22-a-II-1.

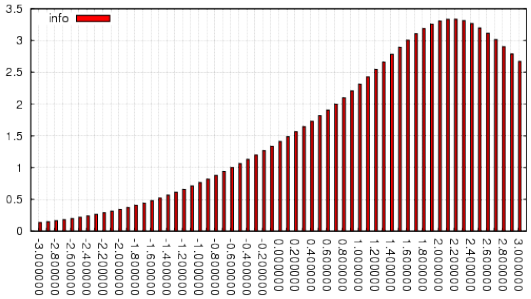


Obr. 4b Úloha oi22-a-II-3.

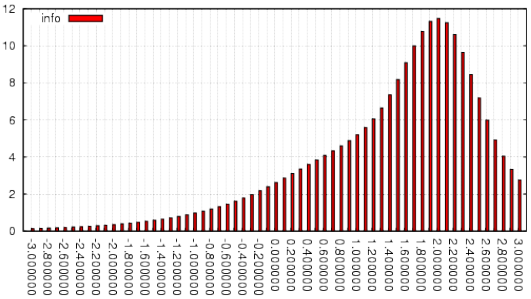
Hlavným záverom, ktorý nás pri našom výskume zaujíma, je funkcia udávajúca množstvo získanej informácie v závislosti od schopnosti riešiteľa pre celé krajské kolo dokopy (4 úlohy, t. j. 12 binárnych

položiek v našom modeli.)

Priebeh tejto funkcie pre krajské kolá v šk. rokoch 06/07 a 07/08 je znázornený na obr. 5a, 5b.



Obr. 5a Krajské kolo 22. ročníka OI, kat. A



Obr. 5b Krajské kolo 23. ročníka OI, kat. A

Vidíme, že v oboch prípadoch je maximum medzi $\theta=2,0$ a $\theta=2,2$. Teda z výsledkov uvedených krajských kôl vieme najpresnejšie odhadnúť schopnosti žiakov, ktorí patria do tohto intervalu. Tu však ide o absolútnu špičku súťažiacich, a to nie je to, čo by sme od krajského kola požadovali. Ako sme už uviedli vyššie, hlavným cieľom krajského kola by malo byť čo najpresnejšie identifikovať množinu najlepších, ktorí by mali postúpiť do celoštátneho kola. Na to by ale bolo omnoho vhodnejšie zvoliť také úlohy, pre ktoré by test information funkcia mala maximum v okolí očakávanej hranice schopností potrebných na postup. (Naopak, skúmané sady úloh by boli výborné pre použitie na celoštátnom kole, kde je hlavným cieľom identifikovať víťaza.)

Ďalej považujeme za potrebné uviesť ešte jedno súvisiace pozorovanie. Hlavným dôvodom, prečo Olympiáda v informatike používa 10-bodovú škálu, je prítomnosť sekundárnych kritérii hodnotenia. Okrem samotnej myšlienky riešenia sa hodnotí aj dôkaz jej správnosti, kvalita popisu riešenia, odhad časovej a pamäťovej zložitosti a prítomnosť drobných chýb. Tieto faktory nemajú na hodnotenie až taký výrazný vplyv ako samotná efektívnosť nájdeného riešenia.

V nami skúmaných krajských kolách však voľba súťažných úloh spôsobila, že medzi bodovým hodnotením efektívnosti algoritmov u súťažiacich, ktorí sa schopnosťou pohybovali okolo hranice postupu, boli len malé rozdiely. Dôsledkom toho mali na konečné umiestnenie v tejto časti výsledkovej listiny vplyv práve spomínané sekundárne faktory. Dovoľujeme si teda tvrdiť, že spomedzi súťažiacich, ktorých schopnosti sa pohybovali okolo hranice postupu, postúpili na celoštátne kolá tí, ktorí dokázali svoje riešenia poriadnejšie vypracovať. Rozhodnutie, či v tomto prípade ide o pozitívum alebo negatívum, prenechávame na čitateľa.

6. ZÁVER

Vypracovali sme teoretický model, pomocou ktorého sme úspešne dokázali modelovať zložitosť algoritmickej úlohy. Pomocou uvedeného modelu sme analyzovali dáta pochádzajúce z reálnych súťaží. Dospeli sme k záveru, že skladba úloh, ktoré sa v súčasnosti používajú na krajskom kole Olympiády v informatike, je vhodná na určenie najlepších riešiteľov. Bolo by však lepšie zmeniť náročnosť použitej sady tak, aby sme pomocou nej čo najlepšie dokázali odhadnúť schopnosti tých riešiteľov, ktorí sú v okolí hranice na postup. Odporúčame teda v budúcnosti použiť širšie spektrum úloh s nižším priemerom koeficientov náročnosti.

Ako možnú tému ďalšieho výskumu vidíme otázku súvisu bodového hodnotenia a motivácie riešiteľov. Ide o komplexnú problematiku: uvedieme napr. situáciu, kedy zaradenie ľahkej úlohy síce prospeje motivácii riešiteľov („aspoň niečo som vyriešil“), avšak zvyšuje volatilitu výsledného umiestnenia riešiteľov (chyba pri riešení ľahkej úlohy sa nemusí dať „dohnať“ bodmi za ostatné.)

LITERATÚRA

[1] BAKER, F. B. and KIM, S.: Item Response Theory: Parameter Estimation Techniques. CRC 2004. ISBN 978-0824758257.

[2] CORMACK, G.: Random Factors in IOI 2005 Test Case Scoring. Informatics in Education 5/2006. ISSN 0868-4952.

[3] KEMKES, G. et al.: Objective Scoring for Computing Competition Tasks. Information technologies at school, 2006.

[4] GLEASON, J.: An Evaluation of Mathematics Competitions Using Item Response Theory. Notices of the ACM, 2008.

[5] PARTCHEV, I.: A visual guide to item response theory. 2004. Dostupné na http://www.metheval.uni-jena.de/irt/VisualIRT.pdf

[6] Slovenská komisia OI: Organizačný poriadok Olympiády v informatike. 2006. Dostupné na http://www.olympiady.sk/index.php?www=sp_file&id_item=555

PROGRAMOVATEĽNÉ HRAČKY A ALGORITMICKÉ MYSLENIE PREDŠKOLÁKOV

MGR. JANKA PEKÁROVÁ

Katedra základov a vyučovania informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina, 84248 Bratislava
tel. 02/60295210, e-mail: pekarova@fmph.uniba.sk

Programovateľné hračky sú v našich podmienkach nezvyčajným nástrojom pre rozvoj plánovacích kompetencií a algoritmického myslenia detí predškolského veku a nižších ročníkov základnej školy. V článku analyzujeme skúsenosti s programovateľnou hračkou Bee-Bot. Metodika jej použitia, vychádzajúca z pozorovaní detských interakcií s hračkou a prieskumu literatúry, môže byť užitočným sprievodcom aj pre učiteľov informatiky na prvom stupni základnej školy. Popisujeme aktivity, ktoré môžu využiť na hodine informatiky, upozorňujeme na silné a slabé stránky práce s hračkou, uvádzame spôsob sledovania pokroku dieťaťa.

Kľúčové slová: *programovateľná hračka, algoritmické myslenie, predškolská príprava*

1. ÚVOD

V roku 2008 prebehla rozsiahla transformácia štátneho vzdelávacieho programu pre deti v materskej škole. V pôvodnom dokumente (Guziová et al., 1999) sa počítaču a digitálnym technológiám nepripisovala takmer žiadna pozornosť. Vo Veľkej Británii a ďalších krajinách sa pritom v rovnakom období v kurikulárnych dokumentoch objavujú digitálne technológie ako silný nástroj pre rozvíjanie porozumenia súčasnému svetu či prostriedok pre rozvoj matematických predstáv a pod. (DfEE, 2000).

Aktuálny program[1] pre predprimárne vzdelávanie viac reflektuje na prienik digitálnych technológií do každodenného života. Učiteľom ponúka priestor pre využitie edukačných softvérových prostredí pre rozvoj percepčno-motorických schopností.

Digitálne technológie však predstavujú **širší súbor nástrojov** než sú softvérové prostredia a počítač. V zahraničí sa veľkej obľube na rôznych predmetoch a stupňoch školského vzdelávania tešia **programovateľné hračky** (Bee-Bot, Roamer, Pixie a iné). V rámci širšieho výskumu integrácie digitálnych technológií do predškolskej prípravy sme s deťmi pracovali s hračkou Bee-Bot[2]. V článku bližšie analyzujeme prácu s hračkou. Hľadáme odpovede na nasledujúce problémy:

- Aký typ aktivít môže rozvíjať učiteľ so zreteľom na podporu algoritmického myslenia a plánovania detí?
- Aké aktivity sú príťažlivé pre deti predškolského veku, prípadne prvého ročníka základnej školy?
- Ako môže učiteľ monitorovať pokrok detí v algoritmickom myslení a plánovaní?

2. PROGRAMOVATEĽNÉ HRAČKY V ŠKÔLKE

Algoritmizácia tvorí veľký (pre učiteľa nezriedka náročný) tematický celok informatiky na základnej i strednej škole. Schopnosť ovládať robota či riadiť nejakú hračku však súčasné deti častokrát získavajú už pred vstupom na základnú školu. Mnohí chlapci sa hrávajú s autíčkami na ovládanie, dievčatá sa stretajú s interaktívnymi bábikami. Hračky na ovládanie tvoria akýsi predstupeň algoritmizácie. Deti sa učia bezprostredne ovládať jednoduché zariadenie a krátkodobo predvídať jeho správanie (prechod autíčka zákrutou).

Siraj-Blatchford a Whitebread (2004) za vhodný nástroj na hlbší rozvoj schopnosti plánovať pokladajú práve programovateľné, robotické hračky. Ak chce dieťa pohnúť robotom, prenáša do neho **vlastné porozumenie** tomu, ako sa pohybuje v priestore a ako môže merať svoj pohyb. Robot tak dáva deťom možnosť prechodu od **akcie** k jej **reprezentácii** (Fine a Thornbury in Hayes a Whitebread, 2006). Robotické hračky rozvíjajú pozičný jazyk (deti si osvojujú pojmy vzťahujúce sa k umiestneniu predmetu v priestore). Ich prostredníctvom sa deti učia pracovať s pojmami ako smer, vzdialenosť, počet. Príbehy dokumentujúce ich použitie poukazujú na rozvoj jazykových schopností i matematických predstáv detí (napr. Hayes a Whitebread, 2006; Stella et al., 2008).

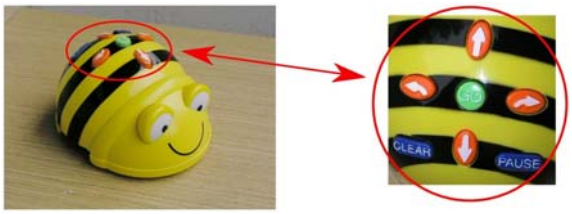
Predškolská učiteľka môže programovateľnú hračku systematicky používať aj ako pomôcku pre rozvoj tvorivého myslenia. Dieťa má totiž v rámci predškolskej prípravy objavovať **algoritmus riešenia úloh** pokusom a omylom alebo **podľa zadávaných inštrukcií**, či odstrániť prípadnú chybu.

[1] http://www.statpedu.sk/ [cit. 20.1.2009]
[2] www.bee-bot.co.uk [cit. 20.1.2009]

3. BEE-BOT

Programovateľná hračka Bee-Bot má tvar veselo pôsobiacej včely s veľmi jednoduchým rozhraním na ovládanie. Pomocou štyroch tlačidiel na chrbte včely (dopredu, dozadu, vpravo vbok a vľavo vbok) dieťa zadáva postupnosť krokov na prejdenie určitej dráhy v štvorcovej sieti. Po zadaní postupnosti krokov stlačením tlačidla GO spustí vykonávanie príkazov a hračka sa začne hýbať. Celá postupnosť môže obsahovať maximálne 40 príkazov. Tlačidlom CLEAR sa postupnosť príkazov vymaže z pamäte hračky. Tlačidlom PAUSE vyvolá používateľ pozastavenie hračky na jednu sekundu (detailnejšie sa používateľskému rozhraniu hračky venuje Pekárová, 2008).

Základnou funkcionalitou robota je **naplánovať cestu v štvorcovej sieti** a spustením postupnosti príkazov (t.j. ich vykonaním) overiť svoje riešenie. Hračku pritom nemožno pripojiť na počítač a rozšíriť jej príkazy pomocou programovacieho prostredia, teda využiť prístup známy zo stavebníc Lego Mindstorms či PicoCricket.



Obr. 1 Bee-Bot a jeho používateľské rozhranie

Stella et al. (2008) pripisuje hračke Bee-Bot potenciál pre rozvoj schopností ako počítanie a logické myslenie, riešenie topologických problémov a najmä skúseností s učením založeným na skúmaní (inquiry based learning), ktoré deti môžu využiť v rozličných situáciách aj mimo oblasti robotiky.

4. PLÁNOVANIE CESTY U PREDŠKOLÁKOV – ANALÝZA

Aké skúsenosti môže očakávať učiteľ pri prvých stretnutiach dieťaťa s hračkou? Záznamy z našej činnosti ilustrujú miskoncepcie ovládania hračky, ale i pokrok v plánovaní cesty pre hračku u jednotlivých detí:

Tlačidlo GO spustí postupnosť príkazov.

Danko, ktorý s hračkou nikdy nepracoval, zabúda vymazať hračke pamäť pred vykonaním ďalších príkazov. Zadá jednoduchú postupnosť KROK DOPREDU a stlačí GO. Ukážem mu, že keď znovu stlačí GO, hračka znovu spraví krok dopredu. Danko okamžite využije tento trik na ďalšie tri kroky dopredu. Hračka dôjde na okrajové políčko podložky, ďalej sa môže hýbať až po otočení sa. Tu Dankovi treba znovu pripomenúť nutnosť zmazania pamäte tlačidlom CLEAR.

Deti si pri použití včely častokrát ukazovali prstom, kde sa včela pri vykonávaní príkazu bude nachádzať. Úspešnosť ich riešenia niekedy ovplyvnilo aj to, či si ukazovali iba polohu alebo aj smer otočenia hračky.

Adam, ktorý úspešne používal včelu pri riešení zložitých úloh a sám si vyhľadával náročné problémy, si dokonca pri riešení ťažkej úlohy presúval/otáčal pri každom stlačení tlačidla včelu tak, aby jej poloha zodpovedala stlačeniu tlačidla v postupnosti príkazov. Po vytvorení celej postupnosti príkazov premiestnil včelu na pôvodnú pozíciu a víťazoslávne stlačil GO.

Pri zadávaní dlhšej postupnosti príkazov obsahujúcej rôzne otáčania sa deti častokrát pomýlia. Skúsenosť s Adamom naznačuje, že viacerým deťom by mohol pomôcť papierový „kurzor“ v tvare včely, ktorým by deti hýbali/otáčali pri zadávaní jednotlivých príkazov.

Po troch stretnutiach s programovateľnou hračkou, v ktorých deti pracovali rôznymi spôsobmi (polovica triedy, dvojica detí a malá skupina) sme sa snažili zmapovať získané zručnosti detí. Väčšina detí zadávala včele postupnosť niekoľkých krokov, deti sa snažili aj otáčať hračku programovo.

Marek si pri vytváraní príkazov pre obídenie obvodu štvorca ukazoval prstom, kde sa včela bude nachádzať. Po stlačení tlačidla na otočenie vpravo posunul prstom o jedno políčko doprava. Následne som ho upozornila na to, že tlačidlo VPRAVO znamená iba otočenie na mieste, bez zmeny polohy včielky. Pri ďalšej postupnosti príkazov si už Marek ukazoval správne a úspešne vytváral postupnosti príkazov obsahujúce otáčanie.

Väčšina detí podobne ako Marek použila tlačidlo pre otočenie v domnienke, že hračka zmení svoj smer a pohne sa potom o jeden krok. Deti, ktoré si počínali pri riešení úloh s menšou istotou, začali po komentári učiteľky zadávať otočenie vždy ako samostatný príkaz a sledovali, kam sa včela dostane.

Deti si samé stupňujú úlohy, ktoré vedia vyriešiť – uhlom otočenia alebo umiestnením hračky. Pokročilejší používatelia hračky si nechávajú hračku po zastavení otočenú ľubovoľným smerom. Iné deti si hračku pred štartom otočia rukou tak, aby sa v ďalšej úlohe hýbala dopredu. Učiteľ povzbudzuje menej sebavedomé deti, aby skúsili úlohu vyriešiť pokročilejším spôsobom (otočiť hračku programovo, naplánovať viac krokov). Deti, ktoré správaniu hračky rozumejú, si tieto rady rýchlo osvoja.

Tánička úspešne skladá kratšie postupnosti príkazov pre včelu, vrátane otáčania. Vie odhadnúť, kde včela skončí po zadaní takejto postupnosti. Niekedy však hračku premiestni rukou. Pri dlhšej postupnosti sa pomýli.

Učiteľ postupne stupňuje náročnosť riešených úloh. Deti neskôr dokážu navrhovať zaujímavé divergentné úlohy. Takýmto spôsobom si určujú

rozsah a hĺbku svojich vedomostí podľa momentálnej úrovne svojich schopností. Programovateľné hračky tak nadväzujú na Papertovu filozofiu konštrukcionizmu.

Učiteľ však niekedy musí dohliadnuť na férové striedanie hračky medzi deťmi.

5. HODNOTENIE POKROKU DETÍ S HRAČKOU

Lydon (2007) vo svojej sérii námetov pre použitie hračky Bee-Bot ponúka postupnosť ôsmich krokov na plánovanie práce zvyšujúcej sa náročnosti s Bee-Botom. Prvých päť krokov zvládla po štyroch stretnutiach aj väčšina škôlkarov z našej triedy:

1) Naprogramuj hračku Bee-Bot na pohyb o jeden krok dopredu.
2) Naprogramuj hračku Bee-Bot na pohyb niekoľko krokov dopredu v jednej postupnosti príkazov a stlač tlačidlo GO.
3) Naprogramuj hračku Bee-Bot na pohyb niekoľko krokov dopredu a dozadu v jednej postupnosti príkazov a stlač tlačidlo GO.
4) Naprogramuj hračku Bee-Bot tak, aby sa otočila vpravo alebo vľavo.
5) Naprogramuj hračku Bee-Bot na pohyb niekoľko krokov dopredu a dozadu, vrátane otáčania, a stlač tlačidlo GO.

Učiteľ informatiky na prvom stupni môže pozorovať pokrok dieťaťa aj v ďalších štádiách práce s hračkou:

6) Použi príkaz PAUSE v postupnosti príkazov.
7) Napíš celý program na papier. Potom ho zadaj hračke a stlač tlačidlo GO.
8) Uprav program a podľa potreby ho prepíš.

Popri algoritmickom myslení môže učiteľ tiež sledovať sociometrické vzťahy v triede, vzťahy medzi spolužiakmi, všímať si, kto je dominantným a suverénnym používateľom hračky, kto radí iným a kto je naopak individualista. Svoje pozorovania môže neskôr využiť pri zadávaní skupinových činností a výbere členov skupín.

Okrem samotného riadenia hračky ju učiteľ využije aj na rozvoj metakognitívnych schopností detí. Učiteľ bude ovládať včelu po určitej krajine. Deti majú pred sebou štvorčekový papier s mapou krajiny. Učiteľ zadá postupnosť príkazov, deti majú na svojej mapke zakresliť, kde včela zastane po vykonaní tejto postupnosti. Náročnosť úlohy možno stupňovať a učiteľ na základe odpovedí detí vyhodnotí, na akej úrovni učenia sa programovania s hračkou sa nachádzajú.

6. METODIKA POUŽITIA HRAČKY BEE-BOT

Zamerajme sa teraz na úlohu hračky pre rozvoj algoritmického myslenia dieťaťa, špeciálne na rozvoj schopnosti plánovať postupnosť krokov.

Učiteľ by mal do použitia programovateľnej hračky zahrnúť aj aktivity, v ktorých môžu deti využiť **zapamätanie postupnosti príkazov**. Scenár podobných aktivít možno popísať takto:

- Aktivita pozostáva z niekoľkých častí. Bee-Bot je na začiatku každej časti umiestnený na rovnakom mieste.
- Každá časť zahŕňa zopakovanie všetkých predošlých častí, a teda celej postupnosti príkazov.

Príkladom takejto aktivity je *Bee-Bot rytierom* (Lydon, 2007). Slovenský učiteľ môže pre uvedený typ aktivít použiť známe rozprávky *Ako išlo vajce na vandrovku* či *Sliepočka a kohútik*.

Úlohy podobného charakteru by mal učiteľ obmieňať s úlohami, v ktorých je potrebné **mazať pamäť** hračky. Deti si tak formujú prvé predstavy o role pamäte pre ukladanie údajov.

Pre úvodné aktivity zahŕňajúce pohyb dopredu a dozadu je vhodná štvorcová sieť napr. rozmeru 9 x 3 štvorce. Včela sa pohybuje v prostrednom páse, v krajných štvorcoch môže učiteľ umiestniť tematické obrázky, plastelínové figúrky či iné výtvory detí. Na začiatku školského roka sa nám osvedčila aktivita *Panelák.* Použili sme výkres s nakresleným obrysom paneláku (rozmer 9 x 3 štvorce), stredný pás pre pohyb včely predstavoval výťah, deti do krajných políčok umiestnili obrázky – okienka so svojou fotografiou. Včela sa akoby zoznamovala s deťmi, pohybovala sa iba niekoľko krokov dopredu (prípadne dozadu) medzi poschodiami.

Obr. 2 Aktivita Zelovoc

Nadväzujúcou aktivitou potom, čo si deti osvoja základy ovládania hračky, je *Zelovoc*. Opäť sa použije rovnaká štvorcová sieť, umiestnia sa na ňu však obrázky ovocia a zeleniny. Na koncové políčka sme umiestnili obrázky zajaca a ježka. K zajacovi deti dopravujú všetky typy zeleniny, k ježkovi naopak ovocie. V aktivite možno použiť i prídavný vozík, ktorý sa pripája na hračku (stráca sa však možnosť cúvania hračky). Na tento vozík si deti budú ukladať nazbieraný tovar. Pri zbieraní do vozíka môžu deti v rámci jednej postupnosti príkazov viac ráz využiť tlačidlo PAUSE, zastať pri danom políčku, naložiť tovar do vozíka a pokračovať v ceste. Pri tejto aktivite sa začali objavovať pokusy otáčať hračku na mieste smerom

k tovaru. Spojenie otáčania a pohybu je záverečnou fázou využitia hračky pre rozvoj algoritmického myslenia detí.

Škála problémov, ktoré deti budú riešiť, je pritom obmedzená viacerými faktormi:

- hračka dokáže vykonať najviac 40 inštrukcií v jednej postupnosti príkazov,
- dĺžka kroku hračky a rýchlosť pohybu hračky je konštantná.

Kopelke (dátum neuvedený) ponúka učiteľom informatiky i ďalších predmetov široké spektrum nápadov pre integráciu hračky Bee-Bot do vyučovania. So staršími deťmi na pokročilej úrovni práce s hračkou môže učiteľ informatiky iniciovať diskusiu na tému programovacieho jazyka pre hračku, brain-storming na tému, aké príkazy by mala poznať budúca verzia hračky, ak ju budeme môcť ovládať slovnými pokynmi? Učiteľ môže počas diskusie vytvárať v špecializovanom softvéri myšlienkovú mapu nápadov detí.

Individuálna práca jedného dieťaťa so včelou umožňuje učiteľovi hlbšie porozumenie spôsobu myslenia konkrétneho dieťaťa. Uvedomujeme si však, že v našich podmienkach je individuálna práca veľmi časovo aj organizačne náročná. Učiteľ na prvom stupni základnej školy však môže pracovať s deťmi rozdelenými do menších skupín (3 – 5 detí). Jedna skupina detí sa pritom hrá so včelou na podložke – tematickej mape (čarovný ostrov, krajina okolo rozprávkového skleneného vrchu) a ďalšie skupiny môžu na počítačoch pripravovať svoje vlastné mapy krajiny ako kombináciu kreslenia v grafickom editore, tlače ako aj použitia výtvarných pomôcok (vodové farby, pastelky).

Alternatívu predstavuje softvérové prostredie *Focus on Bee-Bot*[3], ktoré simuluje správanie skutočnej hračky na obrazovke. Časť detí môže pracovať s reálnou hračkou, ďalšia časť riešiť pripravené úlohy na počítači. Učiteľ však musí pamätať na uzavretosť softvéru – pri tvorbe nových zadaní pracuje s fixnou množinou objektov a máp, na ktorých sa virtuálna včela pohybuje. Aj preto sa nazdávame sa, že pre deti je atraktívnejšia práca s reálnou hračkou a vytváranie prostredia pre jej použitie.

Obr. 3 Softvérové prostredie Focus on Bee-Bot

[3] http://www.focuseducational.com/[cit. 20.1.2009]

7. ZÁVER

Programovateľné hračky prinášajú do vyučovania informatiky možnosť rozvíjať kritické a tvorivé myslenie, navrhovať a riešiť problémy rôznej náročnosti, systematicky naplánovať a testovať svoje riešenie aj bez použitia počítača. Pre učiteľa vytvárajú priestor na skĺbenie informatických znalostí s inými oblasťami (tvorba príbehov, orientácia na mape a pod.). V slovenských základných a materských školách sa doposiaľ takéto hračky vyskytujú zriedka. Veríme však, že v budúcnosti ich možnosti pre bohatšie učenie a učenie sa objavia mnohí tvoriví učitelia informatiky.

LITERATÚRA

[1] DfEE (2000) *Curriculum Guidance for the Foundation Stage.* London: Qualifications and Curriculum Authority Publications, 2000.

[2] Guziová, K. et al.: *Program výchovy a vzdelávania detí v materských školách.* Bratislava: Štátny pedagogický ústav, 1999. ISBN 80- 967721-1-2

[3] Hayes, M., Whitebread. D. (ed.): *ICT in the Early Years.* Maidenhead: Open University Press, Mc. Graw-Hill Education, 2006. ISBN 0-335-20808-8

[4] Kopelke, K. (dátum neuvedený): *Making your classroom buzz with Bee-Bots: Ideas and Activities for the Early Phase.* [online, cit. 15.7.2008] http://www.earlyphaseicts.com/

[5] Lydon, A.: *Let's Go with Bee-Bot. Using your Bee-Bot across the curriculum.* Kirkby-In-Ashfield: TTS Group, Ltd., 2007. ISBN 978-1-906213-78-7

[6] Pekárová, J.: Using a Programmable Toy at Preschool Age: Why and How? In: *Workshop Proceedings of SIMPAR 2008 Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS.* Venice: University of Padova, pp. 112 – 121, 2008. ISBN 978-88-95872-01-8

[7] Siraj-Blatchford, J., Whitebread, D.: *Supporting Information and Communications Technology in the Early Years*. Open University Press, McGraw-Hill Education, 2007, 4. vydanie. ISBN 0-335-20942-4

[8] Stella de Michelle, M. et al.: A Piedmont SchoolNet for a K-12 Mini-Robots Programming Project: Experiences in Primary Schools. In: *Workshop Proceedings of SIMPAR 2008 Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS.* Venice: University of Padova, pp. 90 – 99, 2008. ISBN 978-88-95872-01-8

O DIDAKTIKE PROGRAMOVANIA

RNDR. ĽUBOMÍR SALANCI, PHD.

Katedra základov a vyučovania informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Bratislava
+421 2 602 95 284, e-mail: salanci@fmph.unbia.sk

V didaktike programovania sa snažíme pozerať na vyučovanie algoritmov a programovania ako na proces, ktorý má svoje zákonitosti a pravidlá. Niektoré z nich ilustrujeme a na vybraných situáciách ukazujeme, prečo sú pre učiteľa dôležité. Zamýšľame sa aj nad tým, prečo je náročné kvalitne učiť algoritmy a programovanie.

Kľúčové slová: *didaktika, vzdelávanie, informatika, programovanie, poznávací proces*

1. ÚVOD

Postupy, riešenie problémov a algoritmické myslenie je súčasťou vzdelávacieho obsahu informatiky súčasného Štátneho vzdelávacieho programu [1], [2]. Znamená to, že s algoritmami a programovaním sa nebudú zoznamovať iba vybraní žiaci, ktorí sa zaujímajú o informatiku a ktorých informatika „baví“. Informatika, algoritmické myslenie a v primeranej forme aj jednoduché programovanie budú súčasťou všeobecného vzdelania. Preto si uvedomujeme a pripúšťame, že učitelia informatiky budú učiť aj takých žiakov, ktorých „počítače nebavia“.

Učiteľ informatiky teda na jednej strane musí byť kompetentný vyučovať tematický okruh „Postupy, riešenie problémov, algoritmické myslenie“ a na druhej strane musí brať do úvahy rozmanitosť kolektívu žiakov.

Úloha učiteľa vo vzdelávaní je nesmierne dôležitá. Kladieme si preto prirodzenú, ale ťažkú otázku: „Ako vzdelávať (budúcich) učiteľov informatiky tak, aby vhodným spôsobom rozvíjali algoritmické myslenie žiakov?“

2. VZDELANIE UČITEĽA INFORMATIKY

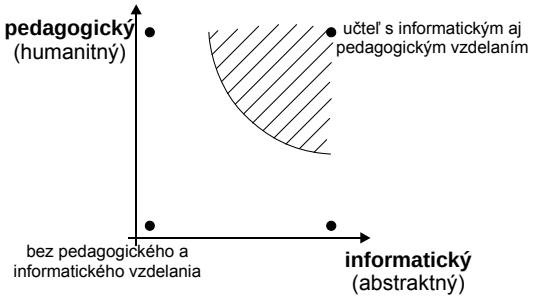
Vyučovanie algoritmov a programovania v sebe spája dva rozmery – pedagogický a informatický. Pre informatiku je charakteristické precízne a abstraktné uvažovanie alebo dokazovanie. Naopak, vo svete pedagogiky pracujeme s istou mierou neurčitosti, s nejasnými, nepresnými hodnotami alebo s pozorovaniami, ktoré sa dajú viac či menej potvrdiť, takže výsledky sú platné s určitou pravdepodobnosťou [3]. Nazdávame sa, že učiteľ informatiky by mal byť vzdelaný v oboch oblastiach.

Učiteľ ako odborník by mal mať praktické skúsenosti a teoretické základy z informatiky:

- pozná nástroje informatiky, základné algoritmy, údajové štruktúry a dokáže ich efektívne používať, navrhovať, analyzovať,
- má prehľad a nadhľad o rôznych oblastiach informatiky a dokáže sa ďalej vzdelávať v oblasti informatiky.

Ako pedagóg by mal mať praktické skúsenosti a teoretické základy z vyučovania informatiky:

- dokáže rozmýšľať o vyučovacom procese, vie analyzovať témy a aplikovať teóriou (teórie) poznávacieho procesu [4], [5] pri vyučovaní informatiky,
- v odborných témach, ktoré často vyžadujú rozvinuté abstraktné myslenie, dokáže rozpoznať dôležité poznatky a primeraným spôsobom ich odovzdať svojim študentom.



Obr. 1 Ak si informatický a pedagogický rozmer vzdelania znázorníme ako graf s osami *x* a *y*, nazdávame sa, že učiteľ informatiky by mal ležať niekde v pravom hornom rohu grafu

Študenti, ktorí na našej fakulte úspešne absolvovali štúdium informatiky (vedeckej alebo učiteľskej) a uplatnili sa ako učitelia informatiky, získali počas univerzitného štúdia dostatočné odborné informatické vzdelanie. Preto v praxi nemávajú problémy s odbornou informatikou. Súčasťou pedagogického vzdelávania budúcich učiteľov informatiky na našej fakulte je niekoľko predmetov. Jedným z nich je didaktika programovania v prvom ročníku magisterského štúdia. Hoci sa obsah tohto predmetu formuje a rozvíja, domnievame sa, že viaceré postrehy, ktoré sme pri jeho vedení získali, sú zaujímavé. Preto sa na didaktiku programovania ďalej zameriame.

3. ILUSTRÁCIA VYBRANÝCH PROBLÉMOV

Prečo potrebujeme učiteľov informatiky vzdelávať v didaktike? Nestačí, aby bol učiteľ iba dobrý odborník v informatike? Nestačí, ak začínajúcemu učiteľovi informatiky vložíme do ruky

učebnice a povieme: „Učte podľa nich, učte tak, ako viete, a pritom používajte zdravý rozum“?

Odpovede na položené otázky môžu naznačiť nasledujúce vybrané ukážky. V nich analyzujeme rozmýšľanie našich študentov, ktoré sme zaznamenali pri rôznych aktivitách v rámci predmetu didaktika programovania. Tento predmet navštevujú predovšetkým študenti informatických učiteľských študijných odborov, niekedy sa na ňom zúčastňujú aj učitelia z praxe, ktorí si dopĺňajú aprobáciu z informatiky alebo aj študenti vedeckej informatiky, ktorí si vybrali doplnkové pedagogické štúdium.

Situácia 1. Študentom, budúcim učiteľom informatiky, sme rozdali rôzne učebnice, ktoré sú zamerané na algoritmy a programovanie (napr. [6], [7], [8]). Študentom sme zadali úlohu: „Ako budúci učitelia si pozrite učebnicu a porozprávajte o nej svojim kolegom.“ Nevraveli sme, ako majú učebnice analyzovať. Zaujímalo nás, čo si na učebnici budú všímať, keďže doposiaľ nepočuli žiadne teoretické základy o vyučovaní algoritmov a programovania. Študenti mali na analýzu 20 minút, čo sa ukázalo ako dostatočné, pretože viacerí skončili v priebehu dvoch až troch minút. Na ilustráciu uvádzame hodnotenia dvoch študentov:

- Prvý študent referoval: „Je to učebnica o Logu.“ S ironickým hlasom ďalej pokračoval: „Na učebnici vidno, v akom roku bola vydaná, a je teda taká, čiernobiela.“
- Druhá študentka o inej učebnici povedala: „Táto učebnica je taká pekná farebná, sú v nej pekné farebné obrázky a príklady. Škoda, že sme my takú peknú farebnú učebnicu nemali.“

Môžeme pozorovať, že obaja študenti takmer úplne ignorovali obsah učebnice. Upútal ich predovšetkým vzhľad učebnice a podľa neho učebnicu posúdili ako zlú alebo dobrú. Znamená to, že ich momentálne vedomosti v oblasti vyučovania informatiky boli také, že učebnice nedokázali posúdiť, analyzovať. Nevedeli, čo si majú pri analýze všímať.

S prvoplánovým hodnotením sa stretneme aj v bežnom živote. Ľudia, často aj veľmi múdri, rýchlo vyhodnotia učebný text. Napríklad, rodič môže zhodnotiť učebnicu takto: „Čo je to za programovanie, veď tam iba kreslíte obrázky.“

Schopnosť hodnotiť pedagogické dokumenty je pre učiteľa dôležitá. V budúcnosti sa bude musieť rozhodnúť, ktorú knihu si kúpi, ktorú učebnicu použije alebo, ktorú web stránku odporučí svojim študentom.

Situácia 2. Každý študent, budúci učiteľ si zvolil jednu tému z algoritmov a programovania (napríklad: korytnačia geometria, premenná, cyklus atď.). Študentom sme opäť rozdali učebnice. Úlohou každého bolo pohľadať v učebnici svoju tému a referovať ostatným spolužiakom to, ako autori učebnice robia úvod do programovania k zvolenej téme. Poznamenajme, že teóriu o poznávacom procese sme ešte neprezentovali ani netrénovali. Na ilustráciu uvádzame vystúpenie študenta, ktorý si zvolil úvod do korytnačej geometrie [6]:

- Študent: „Takže, úvod do korytnačej geometrie. Začína sa kreslením štvorca, príkaz `opakuj 4 [dopredu 100 vpravo 90]` ...“
- Na tomto mieste sme študenta zastavili a opýtali sa: „Naozaj učebnica začína týmto príkladom? Nenašli ste predtým ešte niečo?“
- Študent: „Niečo tam ešte je, ale to sú len také ‚rozprávky‘ o korytnačke ... príkazy `vlavo` a `vpravo`.“
- Pýtame sa: „A nie je práve toto úvod do korytnačej geometrie?“
- Odpoveď študenta: „Toto? Ale, veď je to hrozne jednoduché a samozrejmé...“

Študent zatiaľ nerozumie postupom, ktoré autori učebnice používajú a ani tomu, prečo ich používajú. Je to preto, lebo sa na vyučovanie pozeral očami odborníka, skúseného informatika, ktorý základy dávno zvládol. Teória poznávacieho procesu však hovorí, že pre žiaka, ktorý sa učí nové poznatky, sú „rozprávky o korytnačke“ a elementárne príklady nesmierne dôležité.

Rozprávka o korytnačke v sebe obsahuje určitú motiváciu a približuje metaforu grafického pera. Elementárne ukážky, ako napríklad `dopredu 100`, slúžia na to, aby žiaci zbierali skúsenosti: kam sa príkazy zapisujú, akú majú syntax, ako fungujú, že program je tvorený postupnosťou príkazov alebo, ako bude vyzerať výsledok vykonávania programu.

Všimnime si, že informatika obsahuje veľa abstraktných pojmov: grafické pero; premenná; cyklus; vetvenie; podprogram a ďalšie, ktorým musia žiaci porozumieť. Ukazuje sa nám, že žiak porozumie novým abstraktným pojmom a nadobudne nové poznatky až potom, keď nazbiera dostatok konkrétnych skúseností.

Situácia 3. Študenti dostali úlohu predviesť, ako by učili zadanú tému. Jeden z nich by učil tému pole týmto spôsobom:

- Študent začal: „Takto sa deklaruje pole.“ a na tabuľu napísal:
  `var A: array[1..100] of Integer;`
- Pokračoval: „A je pole. Prvky poľa budú... “ a na tabuľu nakreslil niekoľko obdĺžnikov, do ktorých vpísal číselné hodnoty:

7	4	33	12	10	...

- Ďalej hovoril: „Teraz chceme prvky poľa usporiadať podľa veľkosti. Vezmeme prvok s indexom 1 ....“ a začal vysvetľovať algoritmus bublinkového triedenia.

Vidíme, že počas výkladu sa dopustil množstva chýb. Nebudeme teraz študentov výklad kritizovať, ale budeme ho analyzovať a odnesieme si ponaučenie, prečo chyby nastali:

- Cieľ vyučovania: učiteľ by mal vedieť rozpoznať, „čo“ a „prečo“ je na téme dôležité. Náš učiteľ to ešte nevedel, a preto skombinoval dve náročné témy – polia a triedenie. Spájanie náročných tém považujeme za nesprávne. V literatúre aj v učebniciach [8] však nájdeme spojenie tém, ako napríklad: „rekruzia + backracking“, „smerník + dynamické premenné“, “dynamické premenné + dynamické údajové typy“, „binárne súbory + hašovanie“. O cieli vyučovania určitej témy môžeme vždy diskutovať. Napríklad, v knihe pre skúseného programátora môže byť cieľom naučiť syntaktické pravidlá a všetky možné varianty deklarácie polí a indexovania prvkov. V škole môžeme mať odlišný cieľ – naučiť žiakov pamätať si veľké množstvo údajov pomocou poľa, pracovať s prvkami poľa, pričom syntaktickým pravidlám dáme menšiu váhu a priestor. Triedenie môžeme učiť oveľa neskôr, keď ho žiaci budú potrebovať.
- Motivácia: pred tým, ako začneme učiť tému pole, mali by sme v žiakoch vyvolať pocit, že pole potrebujú používať. V ukážke motivácia chýbala. Motivácia je dôležitou súčasťou vyučovania a tvorí úvod k téme. Žiaci by mali už na začiatku rozumieť tomu, prečo sa novú tému učia, že sa potrebujú naučiť niečo navyše („Prečo sa chceme naučiť pole?“, „Načo nám bude pole užitočné?“). Vymýšľať dobré motivácie je veľmi ťažké, pretože motivácia musí znieť jednoducho, jasne a zmysluplne.
- Analýza témy: videli sme, že v ukážke budúci učiteľ používal nové pojmy a poznatky bez toho, aby ich vysvetlil (pole, index, hranice poľa, prvok poľa, triedenie, syntax). Žiaci im ešte nerozumejú, pretože ich vo svojom slovníku nemajú, resp. im prisudzujú iný význam, ako učiteľ. Pre učiteľa je potrebné, aby dokázal v konkrétnej téme identifikovať nové pojmy, poznatky a problémy.
- Redukovanie: mladý, zanietený, nadšený a entuziastický učiteľ má sklon „učiť všetko, čo sám vie“. Potom, ako učiteľ tému analyzuje a identifikuje v nej nové pojmy, poznatky, problémy môže prísť k záveru, že ich je priveľa a žiaci by im na jednej vyučovacej hodine neboli schopní porozumieť. Preto by z nich mal zvoliť tie najdôležitejšie, na vyučovacej hodine ich vysvetliť a žiaci by si mali ich použitie natrénovať. Napríklad, máme odpozorované, že už len práca s prvkami poľa pomocou indexov je pre žiakov nový a ťažký princíp.
- Aplikovanie teórie o poznávacom procese: z teórie o poznávacom procese už vieme, že pre začiatočníkov je dôležité zbieranie skúseností. Dá so to dosiahnuť tým, že žiaci budú vidieť, trasovať a uvažovať o elementárnych príkladoch (index ako konštanta `A[1]:=100`, index ako konštantný výraz, nepriamy index – index ako premenná). Pozorujeme, že mnohí, nie len budúci, učitelia nevedia vymyslieť elementárny a názorný príklad, ktorý by priamo a zrozumiteľne ilustroval alebo trénoval nový problém alebo poznatok.
- Rozmýšľať o vyučovaní: skôr ako vstúpi učiteľ do triedy musí mať premyslený priebeh vyučovacej hodiny – ukážkové príklady, ktoré bude predvádzať; čo bude kresliť na tabuľu; úlohy pre žiakov atď. Napríklad, náš učiteľ si zvolil pole s veľkým počtom prvkov, takže sa mu údajovú štruktúru nepodarilo vhodne znázorniť. Tri bodky, ktoré sú použité v zmysle „a tak ďalej“ budú pre žiakov metúce. Pri kreslení na tabuli tiež nenaznačil, čo je index.

Na predchádzajúcich situáciách sme videli, že vyučovanie algoritmov a programovania skrýva z pohľadu didaktiky veľa pascí, ale aj dôležitých prvkov. Mnohé z nich si začínajúci učiteľ neuvedomuje alebo im neprikladá dostatočnú váhu. Preto žiaci jeho vysvetľovanie môžu zhodnotiť ako chaotické, ťažké alebo nudné. Myslíme si, že je potrebné, aby učiteľ porozumel cieľu vyučovania, nezabúdal na motiváciu, bol schopný analyzovať tému, rozpoznať v nej nové a dôležité pojmy, poznatky, problémy a mal predstavu o poznávacom procese, aby si dokázal zorganizovať vyučovanie a bol kompetentný hľadať, vymýšľať a posudzovať nové témy, príklady a úlohy.

Ukazuje sa, že nestačí poskytnúť učiteľovi pedagogické materiály a nestačí ani to, aby učiteľ videl iné pedagogické vzory a napodobňoval ich. Je to preto, lebo cudzím postupom neporozumie – nevie prečo ich autor učebnice alebo iný učiteľ použil. Potom ani nechápe, prečo žiaci inému učiteľovi rozumejú, ale jemu nie – hoci učí veľmi podobne, ale „nestráca čas s triviálnosťami“.

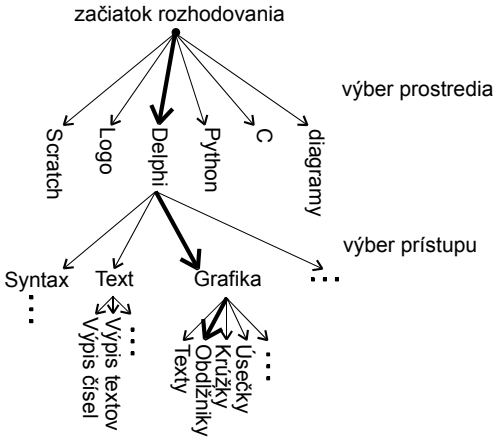
4. KRITICKÉ MYSLENIE VO VYUČOVANÍ ALGORITMOV A PROGRAMOVANIA

Na vyučovanie algoritmov a programovania sa môžeme pozerať ako na proces, pri ktorom sa neustále rozhodujeme a zvažujeme rôzne varianty: prístupy, metódy, témy, poradie tém. Napríklad sa môžeme zamýšľať nad tým, ktorý programovací jazyk použijeme. Do úvahy zvykneme brať výhody a nevýhody programovacieho prostredia, či učíme malých žiakov alebo stredoškolákov, aké ciele pri vyučovaní sledujeme, atď.

Napríklad, pri voľbe prostredia Delphi zvažujeme, či použijeme textovo orientovaný (konzolový) prístup alebo už od začiatku budeme vytvárať grafické a motivujúce programy. Takéto rozhodnutie následne ovplyvní voľbu toho, aké príkazy pre vstup a výstup použijeme, aké typy úloh budeme riešiť atď. Ak by sme uvažovali o jazyku Logo, naše uvažovanie by sa od predchádzajúceho pravdepodobne líšilo, pretože by sme sa rozhodovali

medzi tým, ktoré príkazy korytnačej geometrie naučíme ako prvé.

Takéto procesy rozhodovania môžeme znázorniť ako strom. Jeho vrcholy sú momenty, v ktorých sme prijímali rozhodnutia. Cesta od koreňa k listu predstavuje jednu konkrétnu alternatívu, ktorú sme si zvolili, poradie vrcholov na tejto ceste predstavuje postupnosť krokov, ktoré sme pri vyučovaní vykonali alebo poradie tém, ktoré sme učili.



Obr. 2 Časť stromu s jednou zvýraznenou cestou

Takýto strom môže byť nesmierne rozsiahly. Je to preto, lebo informatika poskytuje množstvo tém a námetov, ktoré sa dajú rôznym spôsobom kombinovať.

Predstavme si množinu, ktorej prvkami sú jednotlivé témy alebo námety: {*prostredie*, *základné príkazy*, *premenné*, *cyklus*, *vetvenie*, *syntax jazyka*, ..., *záznamy*, *polia*, *objekty*, *grafy*, ... *triedenia*, *backtracking*, ..., *UML*, *automaty*, ..., *zložitosť algoritmov*, ...}. Z prvkov takejto množiny môžeme zostavovať rôzne permutácie. Každá permutácia potom predstavuje poradie, v akom by sme (teoreticky) jednotlivé témy postupne učili. A to bez ohľadu na to, či nám poradie tém v permutácii momentálne dáva zmysel alebo nie. Medzi vygenerovanými permutáciami nájdeme aj také, ktoré poznáme z učebníc alebo odbornej literatúry:

- *prostredie* → *základné príkazy* → *cyklus* ...
- *syntax jazyka* → *premenné* → *vetvenie* → ...

Ich počet však bude v porovnaní so všetkými permutáciami zanedbateľný. Medzi permutáciami nájdeme aj varianty, ktoré môžu vyzerať nelogicky:

- *UML* → *objekty* → *grafika* → ...
- *backtracking* → *grafy* → ...

Sú však ľudia, ktorí ich skúmajú – zisťujú, či sa takýmto spôsobom dá učiť algoritmizácia a programovanie. Svoje názory, skúsenosti alebo výsledky potom prezentujú na konferenciách.

A čo si máme myslieť o ostatných permutáciách? Mnohé z nich vôbec nie sú preskúmané alebo sú preskúmané iba veľmi málo. Môžeme ich analyzovať, overovať, hodnotiť a kriticky sa nad nimi zamýšľať, teda: či vedú k naplneniu obsahových cieľov na strane učiteľa a porozumenia poznatkov na strane žiaka alebo koľko energie vynaloží učiteľ na vysvetlenie a žiak na pochopenie pojmov a poznatkov. Je dokonca možné, že sa medzi nepreskúmanými permutáciami nachádzajú aj také, ktoré sú efektívnejšie, ako doposiaľ objavené a zaužívané postupy...

V didaktike programovania kriticky zvažujeme rôzne možnosti vyučovania algoritmov a programovania. Rozmýšľame aj o problémoch, ktoré majú žiaci s pochopením pojmov a poznatkov, snažíme sa ich, pomenovať [9], diagnostikovať a korigovať. Vďaka takýmto poznatkom sa dokážeme ľahšie vyhnúť nesprávnym rozhodnutiam, nesprávnym cestám. Učiteľom informatiky často odovzdávame až výsledky výskumu – formulujeme pravidlá, píšeme učebnice alebo metodiky.

5. ZÁVER

Mohlo by sa zdať, že pri vyučovaní algoritmov a programovania nie sú žiadne problémy. Ako sme však ukázali, problémov spojených s vyučovaním algoritmov a programovania je veľa. Za dôležité považujeme to, aby sa učiteľ kriticky zamýšľal nad vyučovaním a pri vysvetľovaní nových problémov používal elementárne ukážky, úlohy alebo príklady. Až tie vedú k pochopeniu a osvojeniu poznatkov a k riešeniu náročnejších úloh.

LITERATÚRA

[1] Osnovy, obsah a štandardy: www.statpedu.sk

[2] Štátny vzdelávací program: www.minedu.sk

[3] Kalaš, I.: Pedagogický výskum v informatike a informatizácii (1. časť) – DidInfo 2008, ISBN 978–80–8083–367–1, s. 50.

[4] Hejny, M.: Understanding and structure, Proc. of CERME 3, Bellaria (2003) www.dm.unipi.it/~didattica/CERME3/proceedings/Groups/TG3/TG3_Hejny_cerme3.pdf

[5] Salanci, Ľ.: Understanding algorithms and programming. ISSEP 2008 : Informatics Education – Contribution Across the Curriculum. Torun, Poland : ISBN 978–83–60425–31–2, s. 10–19

[6] Varga, M., Blaho, A., Zimanová, R.: Informatika pre gymnáziá – Algoritmy s Logom. SPN 1999, ISBN 80–08–02965–X

[7] Blaho, A.: Programovanie v Delphi. Bratislava : SPN – Mladé letá, 2006. ISBN 80-10-00421-9.

[8] Langfield, S.: Learning to Program in Pascal & Delphi. Payne–Gallway Publishers Ltd., 2003. ISBN 1-904467-29-6.

[9] Paz, T.: What Can We Learn from Educationally Disadvantaged High School Students about Variable? ISSEP 2006, Vilnius : s. 29–39

ÚVAHA O DIMENZIÁCH ALGORITMICKÉHO MYSLENIA A JEHO ROZVÍJANÍ

RNDR. ĽUBOMÍR ŠNAJDER, PHD., MGR. JÁN GUNIŠ

Univerzita Pavla Jozefa Šafárika v Košiciach, Prírodovedecká fakulta, Ústav informatiky, Jesenná 5, 041 54 Košice, tel. 055 23 461 39, e-mail: lubomir.snajder@upjs.sk, jan.gunis@upjs.sk

V príspevku sa zaoberáme vzťahom algoritmického myslenia ku kľúčovým kompetenciám a potrebe jeho rozvíjania u žiakov základnej a strednej školy. Na základe štúdia a našich skúseností predstavujeme model popisu algoritmického myslenia podľa troch dimenzií – univerzálnosť riešenia, formálnosť vyjadrenia riešenia, úroveň myslenia využitá pri riešení problému. Stručne uvádzame spôsoby rozvíjania algoritmického myslenia v rámci vyučovania informatiky a iných predmetoch. V druhej časti príspevku uvádzame výsledky nášho výskumu v oblasti rozvíjania algoritmického myslenia prostredníctvom zážitkového vyučovania na príklade objavovania algoritmov usporadúvania žiakmi základnej školy.

Kľúčové slová: *algoritmické myslenie, kľúčové kompetencie, výučba, informatika, zážitkové metódy*

1. ÚVOD

Výsledky medzinárodných výskumov [1] v oblasti pripravenosti mladých ľudí vo veku 15 rokov a na konci povinnej školskej dochádzky vyrovnať sa s požiadavkami súčasnej informačnej spoločnosti ukazujú, že naša populácia 15-ročných žiakov vykazuje podpriemerné výsledky.

V oblasti matematickej gramotnosti (použitie matematických vedomostí v množstve rozličných situácií rozličnými spôsobmi) dosahuje najvyššiu úroveň 5-6 len 12,7 % našich žiakov (priemer OECD je 14,6 %, Česká republika 18,2 %). Naši žiaci dosahujú priemerne úroveň 3 (zo škály úrovní 0 - 6).

V oblasti riešenia problémov (individuálna schopnosť použiť kognitívne procesy na riešenie reálnych medzipredmetových problémov, keď cesta k riešeniu nie je okamžite viditeľná a predmet a obsah oblasti, ktoré sú v úlohe použité, nie sú na prvý pohľad zreteľné) dosahujú naši žiaci úroveň 1 (zo škály úrovní 0 - 3). Naši žiaci sú schopní riešiť len základné problémy, nevedia riešiť viacdimenzionálne problémy, ktoré majú viac dátových zdrojov, alebo vyžadujú zdôvodnenie na základe poskytnutých informácií. Naše skúsenosti so študentmi prvého ročníka vysokej školy, z výučby na strednej škole a z programátorskej súťaže PALMA junior [2] tieto zistenia potvrdzujú.

Schopnosť riešiť problémy je významnou kompetenciou každého človeka. Štátny vzdelávací program [3] túto schopnosť uvádza ako jednu z kľúčových kompetencií (kompetencia riešiť problémy). Uvedená schopnosť vyžaduje logické, algoritmické a abstraktné myslenie, ktorých rozvoj je a mal by ostať súčasťou informatického vzdelávania žiakov.

2. ALGORITMICKÉ MYSLENIE

S pojmom algoritmické myslenie sa môžeme stretnúť v dokumentoch definujúcich Štátny vzdelávací program [3]. Algoritmické myslenie je jednou zo zložiek kľúčovej kompetencie v oblasti informačných a komunikačných technológií. Kým jedným z cieľov [3] primárneho vzdelávania je, aby žiaci získali základy algoritmického myslenia, výsledkom vyššieho sekundárneho vzdelávania je, že žiak vie algoritmicky myslieť a využívať tieto schopnosti v reálnom živote.

Podľa [3] má vyučovanie matematiky napomôcť rozvoju algoritmického myslenia žiakov, schopnosti pracovať s návodmi a tvoriť ich. Informatika si v tejto oblasti kladie vyššie ciele. Vo výkonovom štandarde oblasti „Postupy, riešenie problémov, algoritmické myslenie“ je uvedené, že žiak vie „Analyzovať problém, navrhnúť algoritmus riešenia problému, zapísať algoritmus v zrozumiteľnej formálnej podobe, overiť správnosť algoritmu“ a „Riešiť problémy pomocou algoritmov, vedieť ich zapísať do programovacieho jazyka, hľadať a opravovať chyby.“

Vidíme, že algoritmizácia je činnosťou pozostávajúcou z viacerých zložiek. Otázkou ostáva, čo je algoritmické myslenie, ako ho merať a ako ho rozvíjať.

Podľa [4] je algoritmické myslenie súbor schopností umožňujúci vytváranie algoritmov a porozumenie algoritmom. Patrí sem schopnosť analyzovať daný problém, schopnosť extrahovať podstatu problému, schopnosť nájsť a určiť základné kroky (činnosti) adekvátne riešeniu daného problému, schopnosť zostrojiť zo základných krokov správny algoritmus vedúci k riešeniu daného problému, schopnosť uvažovať o všetkých možných (aj špeciálnych) prípadoch problému a schopnosť vylepšovať zložitosť nájdeného algoritmu.

Na základe štúdia odborných zdrojov [5, 6], pozorovaní a skúseností z vlastnej výučby na SŠ a VŠ a programátorskej súťaže PALMA junior pre žiakov ZŠ a SŠ [2] rozlišujeme niekoľko dimenzií algoritmického myslenia.

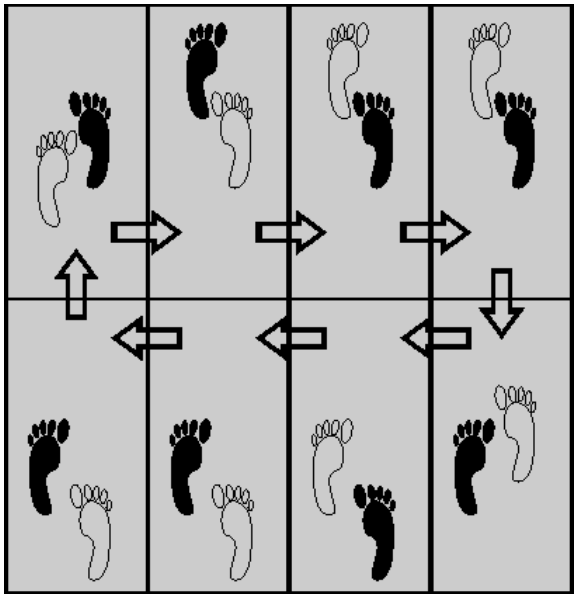
Dimenzia univerzálnosti riešenia:

- riešenie pokrývajúce jeden konkrétny prípad (vyriešenie ilustratívneho prípadu zo zadania, napr. nájdenie cesty z konkrétneho labyrintu),
- riešenie pokrývajúce viacero prípadov, nie však všetky možné prípady (riešenie rovnice $ax+b=0$ len pre prípad $a\neq0$),
- všeobecné riešenie pokrývajúce všetky prípady (riešenie rovnice $ax^2+bx+c=0$ pre $\forall a, b, c \in R$).

Miera univerzálnosti riešenia je relatívna, lebo je závislá od úrovne vedomostí žiakov. Rovnicu $ax^2+bx+c=0$ vieme riešiť na vyššom stupni aj v obore komplexných čísel.

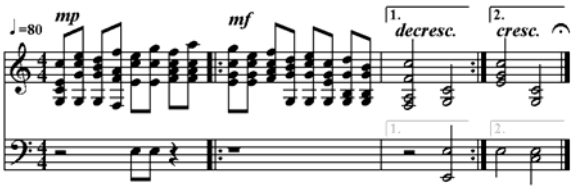
Dimenzia formálnosti vyjadrenia riešenia:

- neverbálne vyjadrenie algoritmu riešenia priamou akciou, činnosťou, napr. pohybom (usporiadanie ceruziek podľa dĺžky deťmi v materskej škole, algoritmický pochod filipínskych väzňov [7]),
- vyjadrenie algoritmu riešenia v prirodzenom jazyku, voľne, bez obmedzení, žiak nielen vykonáva, ale aj popisuje svoje riešenie (rodič učí dieťa, ako bezpečne prejsť cez križovatku),
- vyjadrenie algoritmu riešenia v prirodzenom jazyku s prvkami symboliky danej oblasti (postup výroby kyseliny sírovej, popis krokov tanca),



Obr. 1 Časť krokového algoritmu salsy

- vyjadrenie pomocou formálneho jazyka (zápis Euklidovho algoritmu v programovacom jazyku, grafický zápis algoritmu rozkladu čísla na prvočinitele, matematický zápis konštrukcie stredu úsečky, notový zápis skladby); úroveň vyjadrenia riešenia pomocou formálneho jazyka má viacero podúrovní odlišujúcich sa silou výrazových prostriedkov.



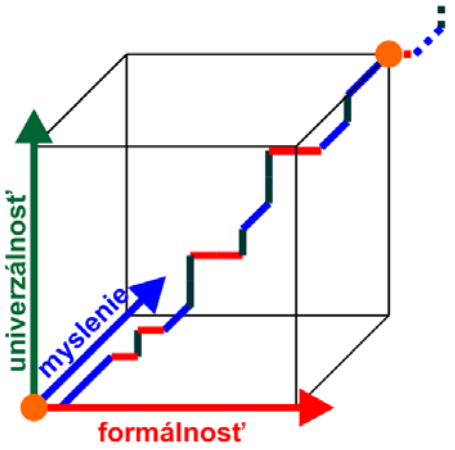
Obr. 2 Zápis skladby obsahujúci značky opakovania, vetvenia a paralelných procesov

Dimenzia úrovne myslenia:

- reprodukcia zapamätaného, už existujúceho algoritmu bez porozumenia (žiak vie napísať algoritmus usporadúvania, ale nerozumie mu, nevie ho vysvetliť),
- reprodukcia zapamätaného, už existujúceho algoritmu s porozumením (žiak vie popísať a vysvetliť algoritmus usporadúvania, rozumie mu),
- aplikácia hotového (cudzieho) algoritmu v konkrétnej situácii (žiak rozpozná situáciu, problém, na riešenie ktorého vie aplikovať algoritmus usporadúvania),
- analyzovanie a porovnanie existujúcich algoritmov (žiak rozumie cudziemu, hotovému algoritmu, vie porovnať, ktorý algoritmus usporadúvania je vhodnejší pre riešenie danej situácie),
- vytvorenie vlastného algoritmu (žiak samostatne objaví algoritmus binárneho vyhľadávania pri tvorbe hry na uhádnutie čísla),
- vytvorenie vlastného algoritmu s vyššou mierou zovšeobecnenia (parametrizácie) resp. lepšou efektivitou, vypočítanie jeho výpočtovej zložitosti, zdôvodnenie správnosti riešenia (žiak vie porovnať svoje riešenie s inými známymi algoritmami a argumentovať pre použitie svojho riešenia, riešenie náročnej úlohy z programátorskej súťaže).

V oblastiach ako programovanie, inžinierstvo, umenie či dizajn očakávame, že žiak nezostane na šiestej úrovni revidovanej Bloomovej taxonómie [5]. Dlhodobým cieľom informatického vzdelávania je, aby bol žiak schopný využiť nadobudnuté vedomosti, nadhľad a skúsenosti pri vytváraní kvalitatívne nových aplikácií (Higher Application) [6].

Pohľad na algoritmické myslenie cez tieto dimenzie nám pomáha identifikovať a pochopiť dôvody niektorých problémov pri výučbe algoritmizácie a programovania. Uvedomenie si týchto dimenzií nám pomôže pri hľadaní spôsobov rozvíjania algoritmického myslenia žiakov.



Obr. 3 Trojdimenzionálny pohľad na rozvoj algoritmického myslenia

Na základe navrhnutého modelu a uvedených príkladov sme presvedčení, že algoritmické myslenie je možné rozvíjať v celej populácii žiakov, pričom nepokladáme za neúspech, keď sa každý žiak nedostane do pomyselného bodu [1, 1, 1]. Ďalej sme presvedčení, že pri rozvíjaní algoritmického myslenia nemôžeme robiť veľké skoky v jednej dimenzii na úkor ostatných.

3. SPÔSOBY ROZVÍJANIA ALGORITMICKÉHO MYSLENIA

Rozvíjať algoritmické myslenie žiakov je možné a potrebné nielen vo výučbe informatiky a matematiky. Považujeme za potrebné venovať sa tejto problematike aj vo výučbe ostatných predmetov a v mimoškolskej činnosti žiakov.

V rámci informatiky existuje viacero spôsobov rozvíjania algoritmického myslenia prostredníctvom tvorby projektov (napr. hier), programovaním robotických stavebníc, modelovaním javov a závislostí v prostredí tabuľkového kalkulátora, prostredníctvom programátorských súťaží, pomocou zážitkového vyučovania [8] atď.

Pri výučbe algoritmizácie a programovania odporúčame venovať dostatočný priestor samotnej algoritmizácii. Pre lepšie pochopenie sú vhodné metódy vizualizácie [4] a osobného zážitku. Takýto spôsob je pre žiakov prirodzenejší a neznechutí ich. Je prijateľnejší hlavne pre tých žiakov, ktorí majú slabšie abstraktné myslenie a matematický aparát.

Už u žiakov prvého stupňa základnej školy môžeme rozvíjať algoritmické myslenie. Samozrejme na veku primeranej adekvátnej úrovni myslenia a akceptujúc ich výrazové prostriedky. Môžeme s nimi realizovať aktivity spojené s postupmi, napr. skladaním origami, tvorbou príbehov, scénok, tancom, pohybovými hrami, hraním logických hier (logik, sudoku, hádaj číslo, lodičky), kúziel atď.

V ďalšej časti príspevku popíšeme naše skúsenosti a odporúčania pri objavovaní algoritmov usporadúvania žiakmi základných škôl.

4. ALGORITMY USPORADÚVANIA

Popis aktivity:

Skupinu žiakov vyzveme, aby sa vzájomne usporiadali. Po usporiadaní ich požiadame, aby svoj postup zapísali tak, aby ktokoľvek (aj žiaci samotní) jeho vykonaním dokázal skupinu žiakov opakovane usporiadať. Vykonávateľom algoritmu môže byť súčasne každý žiak (porovnáva sa so susedmi). Výsledkom je paralelné usporadúvanie. Vykonávateľom môže byť len jeden žiak. Ostatní sú pasívni. Výsledkom je usporadúvanie sekvenčné.

Ciele:

- Pochopiť potrebu usporiadania dát ako predpoklad dobrej organizácie dát a ich rýchleho prehľadávania.
- Pochopiť význam kritéria porovnania (primárneho a sekundárneho kľúča) v procese usporadúvania.
- Pochopiť význam pojmov vzostupné a zostupné usporiadanie.
- Objaviť základné algoritmy paralelného usporadúvania, objaviť základné algoritmy sekvenčného usporadúvania.
- Vedieť nájdený postup usporadúvania formálne zapísať.
- Vedieť realizovať formálne zapísaný postup (algoritmus).
- Uvedomiť si dôsledky vyplývajúce z obmedzenia pamäťovej kapacity a času.
- Intuitívne chápať pamäťovú a časovú zložitosť algoritmov.
- Chápať rozdiel medzi algoritmom (postupom), vykonávateľom a prvkami (dátami).
- Pochopiť rozdiel medzi vykonávateľom-človekom a vykonávateľom-strojom.

Komentáre a odporúčania:

Veľké skupiny (rádovo desiatky) žiakov sa ťažko manažujú, komunikácia medzi učiteľom a žiakmi je komplikovaná.

Počas vyjasňovania si úlohy „usporiadania sa“ sa žiaci celkom správne zaujímajú podľa čoho sa majú usporiadať (čo je kľúč) a v akom poradí (vzostupne alebo zostupne).

Nie je vhodné, ak hodnota kľúča je priamo viditeľná (napr. výška, veľkosť topánky). Ak hodnota kľúča nie je priamo viditeľná (napr. deň narodenia, počet súrodencov), žiakov „donútime“ k vzájomnej komunikácii, k priamemu porovnávaniu dvoch hodnôt.

Počas samotného usporadúvania môže nastať problém ako postupovať v prípade rovnosti hodnoty kľúčov. Ponúkajú sa dve riešenia. Použiť sekundárny kľúč, alebo v prípade rovnosti považovať pozíciu za vyhovujúcu (žiaci s rovnakou hodnotou sa môžu postaviť napr. za seba).

Ak sa žiaci zoskupia do tvaru kružnice (rozmiestnenie v kruhu je celkom prirodzené, pretože nikto nie je na okraji), algoritmus usporadúvania môže generovať nekonečný proces. Tento problém si žiaci počas usporadúvania uvedomia a preskupia sa do lineárnej štruktúry. Nesnažme sa ich preto vhodne rozmiestniť my už na začiatku.

Diskutujme so žiakmi o „vylepšeniach“, ktoré použili.

Napr. ak je kľúčom deň narodenia, vieme podľa jeho hodnoty odhadnúť približnú výslednú pozíciu. Hrubé priblíženie k výslednej pozícii je teda veľmi rýchle. Spresnenie pozície sa už deje v relatívne malom množstve dát. Položme žiakom otázku: Čo ak pozíciu nevieme odhadnúť?

Žiaci si pamätajú niektoré hodnoty, takže niektoré porovnania nerobia, pretože poznajú výsledok. Položme žiakom otázku: Čo ak si odpovede na otázky nepamätáme a musíme sa opätovne spytovať?

Diskutujme o možných vylepšeniach algoritmu, napr. ako redukovať počet opýtaní a výmen. Pomaly zavádzajme obmedzujúce podmienky tak, aby sme sa čo najviac priblížili podmienkam typickým pre počítač.

Žiaci tvoria relatívne malú skupinu dát. Vykonávateľ (človek) má nadhľad nad situáciou, napr. pohľadom dokáže vybrať najvyššieho v skupine, ľahko rozpozná, že časť množiny je už usporiadaná a pod. Svoj postup dokáže človek v priebehu vykonávania algoritmu operatívne modifikovať. Bol by postup žiakov rovnako dobre realizovateľný, ak by ich bolo 100, 1000, 1000000? Vyzveme žiakov, aby svoj postup prípadne skorigovali.

Celý proces smeruje k tomu, aby žiaci získali predstavu o tom, ako usporadúva počítač. Niektoré výhody je preto potrebné „umelo“ redukovať. Napr. pohľadom na skupinu žiakov zistím, kto je najvyšší (toto počítač spraviť nevie).

Žiaci sú schopní objaviť a formálne pomerne presne zapísať niektorý z algoritmov paralelného usporadúvania, napr. „Porovnaj sa s pravým susedom a ak si väčší, vymeňte sa!“.

Vyzvime žiakov, aby navrhli postup pre usporiadanie čísel, ktoré sú napísané na kartičkách. Kartičky sú číslami otočené smerom dole a v jednom čase vieme porovnať čísla len na dvoch kartičkách. Je pravdepodobné, že žiaci objavia algoritmus usporadúvania prebublávaním. Sú schopní ho formálne zapísať. Uvedomujú si niektoré invarianty, napr. že po i-tom prebublaní je i najväčších prvkov už na svojom mieste a nie je potrebné ich opätovne kontrolovať.

Záver:

Žiaci sú schopní objaviť a formálne popísať niektoré základné algoritmy usporadúvania. Navrhovaná aktivita im umožňuje proces usporadúvania zažiť, a tak si ho dôslednejšie uvedomiť. Naučia sa chápať obmedzenia počítača. Sme presvedčení, že takýto spôsob im umožní rýchlejšie pochopiť aj ďalšie algoritmy, ktoré už nebudú schopní objaviť a budú im predostreté ako sčasti hotové.

5. ZÁVER

Algoritmické myslenie je v úzkom vzťahu s viacerými kľúčovými kompetenciami žiakov. V príspevku sme predstavili trojdimenzionálny model popisu algoritmického myslenia, pomocou ktorého sme chceli poukázať na jeho rôzne medziúrovne. Takého nazeranie na algoritmické myslenie pomôže hľadať spôsoby jeho rozvíjania už u žiakov na prvom stupni základných škôl. Výsledky publikované v príspevku boli vytvorené v rámci grantov LPP-0131-06 „Zvyšovanie vedomostného potenciálu“ a LPP-0170-07 „Talent v akcii“.

LITERATÚRA

[1] PISA SK 2003 - Národná správa < http://www.statpedu.sk/buxus/generate_page.php_page_id=973.html >

[2] PALMA junior < http://di.upjs.sk/palmaj/ >

[3] Štátny vzdelávací program < http://www.statpedu.sk/buxus/generate_page.php_page_id=1221.html >

[4] FUTSCHEK, G.: Algorithmic Thinking: The Key for Understanding Computer Science. In: Informatics Education – The Bridge between Using and Understanding Computers. Springer Berlin/Heidelberg, 2006. pp.159-168, ISBN 978-3-540-48218-5

[5] FOREHAND, M.: Bloom's Taxonomy <http://projects.coe.uga.edu/epltt/index.php?title=Bloom%27s_Taxonomy>

[6] JOHNSON, C. G., FULLER, U.: Is Bloom’s Taxonomy Appropriate for Computer Science? <http://www.cs.kent.ac.uk/pubs/2007/2552/content.pdf>

[7] Algorithm March with Prisoners <http://www.youtube.com/watch?v=WjMd2Vabcv8>

[8] ŠNAJDER, Ľ., GUNIŠ, J., GUNIŠOVÁ, V.: Aktivizujúce metódy v školskej informatike. In: Zborník konferencie DidInfo 2008 Banská Bystrica [CD ROM]. Univerzita Mateja Bela, Fakulta prírodných vied, Banská Bystrica. 2008. ISBN 978-80-8083-556-9.

Krátke prednášky

Short Papers

DRUHÁ VZDELÁVACIA AKTIVITA V NÁRODNOM PROJEKTE ĎALŠIEHO VZDELÁVANIA UČITEĽOV V PREDMETE INFORMATIKA
verzus
ROZŠIRUJÚCE ŠTÚDIUM INFORMATIKA

DOC. RNDR. GABRIELA ANDREJKOVÁ, CSC.

Ústav informatiky, Prírodovedecká fakulta, Univerzita P. J. Šafárika, Jesenná 5, Košice,
tel. 055/2346228, e-mail: Gabriela. Andrejkova@upjs.sk

Existujúce rozširujúce štúdium Informatika (RŠI) je zamerané na získanie kvalifikácie pre ďalší aprobačný predmet pre učiteľov, ktorí učia alebo predpokladajú, že budú učiť informatiku. Národný program ďalšieho vzdelávania učiteľov základných škôl a stredných škôl v predmete informatika (DVUi) je zameraný na vzdelávanie v troch vytypovaných skupinách učiteľov [4, 5], pričom druhá skupina je orientovaná tiež na získanie kvalifikácie v predmete Informatika.

Príspevok je zameraný na informáciu o plánovanej druhej vzdelávacej aktivite v rámci národného projektu DVUI a na spracovanie skúseností s výučbou na RŠI, špeciálne na skúsenosti s prípravou zadaní záverečných prác, klasifikáciu riešených prác podľa spracovávanej problematiky. Postup aktivít pri vedení poslucháčov pracujúcich na záverečných prácach je vnorený do kolesa aktivít. V závere sú uvedené odporúčania pre postup pri príprave zadaní záverečných prác a kontrolu postupov pri ich spracovávaní.

Kľúčové slová: *ďalšie vzdelávanie učiteľov, didaktika informatiky, záverečná práca z informatiky, koleso aktivít*

1. ÚVOD

MŠ SR v rámci operačného programu ***Vzdelávanie*** priamym zadaním požiadalo Štátny pedagogický ústav o vypracovanie národného projektu ***Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika***. Projekt bol vypracovaný [5] a v októbri aj 2008 schválený. Prvé vzdelávanie učiteľov by sa malo začať v máji 2009 v jednej z vytypovaných skupín učiteľov (viac informácií o vzdelávaných skupinách je v [4]), s síce v druhej skupine, ktorá je zameraná na učiteľov, ktorí učia informatiku, ale doposiaľ nezískali kvalifikáciu pre tento predmet. Projekt má pre túto skupinu nasledujúce ciele [5]:

- navrhnúť, pripraviť a realizovať intenzívne ucelené vzdelávanie pre učiteľov 2. stupňa ZŠ a SŠ, ktorí učia a/alebo budú učiť informatiku, ale zatiaľ pre tento predmet nemajú kvalifikáciu,
- podporiť takéto vzdelávanie ucelenou sériou študijných materiálov (tlačených i elektronických),
- prostredníctvom elektronických študijných materiálov a na projektovom portáli publikovaných záverečných projektov účastníkov prispieť k rozvoju virtuálneho prostredia pre ďalšie vzdelávanie učiteľov informatiky na ZŠ a SŠ.

Ucelené vzdelanie a získanie kvalifikácie pre aprobačný predmet Informatika v súčasnosti je pre učiteľov možné v rámci rozširujúceho štúdia, ktoré je realizované na viacerých univerzitách (UMB, UPJŠ, UKF). Študijné programy sa snažia dodržať požiadavky akreditačnej komisie pre študijný odbor: učiteľstvo akademických predmetov – informatika. Analýza študijných programov ukazuje, že toto naozaj dodržujú, avšak málo zohľadňujú výučbu predmetov pripravujúcich žiakov SŠ na maturitu.

V príspevku sa budeme najprv venovať popisu druhej vzdelávacej aktivity týkajúcej sa druhej cieľovej skupiny v rámci projektu DVUi. V ďalšej kapitole popíšeme skúsenosti z realizácie rozširujúceho štúdia Informatika na PF UPJŠ, pričom sa budeme zameriavať na záverečné práce, analýze ktorých budeme venovať nasledujúcu kapitolu. Modifikovaný model kolesa bádania – koleso aktivít použijeme na spracovanie postupnosti aktivít pri zadávaní a spracovávaní záverečných prác. V závere uvedieme odporúčania týkajúce sa zadávania záverečných prác a kontroly ich riešenia.

2. DRUHÁ VZDELÁVACIA AKTIVITA DVUI

Druhá vzdelávacia aktivita v rámci národného projektu pre ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika (DVUi) je zameraná na vzdelávanie učiteľov 2. stupňa ZŠ a SŠ, ktorí vyučujú predmet informatika (v rôznej forme a s rôznym názvom), ale nemajú na jeho vyučovanie kvalifikáciu. Vzdelávanie učiteľov potrvá päť semestrov (v semestri plánujeme 12 pracovných týždňov) a bude pozostávať zo 480 vyučovacích hodín. Obsah výučby pokryje akreditované vzdelávanie rozširujúceho štúdia informatiky, čím učitelia dostanú príležitosť získať kvalifikáciu pre ďalší aprobačný predmet (po absolvovaní štátnych skúšok na niektorej z fakúlt, ktorá túto akreditáciu má). Koncepciu tejto aktivity staviame okrem iného na nasledujúcich východiskách [4, 5]:

- o vyučovanie predmetu informatika je veľký záujem aj zo strany rodičov a žiakov, a tiež učiteľov, vedenia škôl a zamestnávateľov,

- v súčasnosti tento predmet vyučuje veľa nekvalifikovaných učiteľov, u ktorých môžeme očakávať relatívne dobrú úroveň základnej digitálnej gramotnosti, ktorí však nikdy neštudovali odbor informatika. Predpokladáme teda, že nemajú vybudované hlbšie porozumenie základov informatiky, programovania a ďalších kľúčových zložiek takejto kvalifikácie. Ak chceme, aby sa na školách rozvíjala moderná informatika a napr. aj žiacke programátorské či robotické súťaže, musíme venovať veľkú pozornosť príprave učiteľov informatiky (učiteľov nekvalifikovaných, rekvalifikovaných aj kvalifikovaných),

- informatika – a ešte viac jej didaktika – je mladý vedný odbor, ktorý sa rýchlo vyvíja. Len postupne na didakticko-informatických katedrách (v celej Európe) budujeme koncepciu moderného vyučovania tohto predmetu, ktorá musí nahradiť niektoré neaktuálne a neatraktívne prístupy (napr. prístupy neprimerane technologické, ktoré chybne redukujú informatiku na náuku o počítači alebo na prácu s počítačom).

Vzdelávanie účastníkov sa bude realizovať v jednotlivých študijných skupinách pod vedením lektorov, s využitím pripravených študijných materiálov, s aktívnym využitím moderných digitálnych technológií, v kvalitne vybavených učebniach s PC a video dáta technikou – a to na fakultách slovenských vysokých škôl, ktoré pripravujú budúcich učiteľov informatiky. Pri vzdelávaní využijeme rôzne formy vzdelávania (prezenčná forma štúdia a e-learning) a viaceré moderné metódy vzdelávania: prednášku, prezentáciu, seminár, praktické cvičenia, diskusie, samostatné štúdium, vývoj projektov, praktické ukážky, otvorené hodiny a ďalšie.

Účastník tohto vzdelávania úspešne ukončí svoje štúdium, ak:

- úspešne absolvuje všetky povinné a vybrané voliteľné študijné moduly a splní všetky predpísané študijné povinnosti,
- vypracuje a pred skúšobnou komisiou obháji svoju záverečnú prácu,
- úspešne absolvuje odbornú rozpravu pred skúšobnou komisiou, počas ktorej komisia preverí, či zvládol obsah vzdelávania a splnil jeho ciele, t.j. či dosiahol takú úroveň, aby sa mohol stať kvalifikovaným učiteľom predmetu informatika

Po úspešnom ukončení vzdelávania účastník dostane zodpovedajúci certifikát.

Cieľom projektu je teda pripraviť vzdelávanie tejto skupiny učiteľov na takej úrovni, aby dostali možnosť stať sa kvalifikovanými učiteľmi. Kvalifikovaný učiteľ informatiky je veľkým prínosom pre školu, ale tiež pre celú spoločnosť.

3. ROZŠIRUJÚCE ŠTÚDIUM INFORMATIKA

Rozširujúce štúdium Informatika je zamerané na získanie kvalifikácie pre ďalší aprobačný predmet pre učiteľov, ktorí učia alebo predpokladajú, že budú učiť informatiku. Štúdium trvá päť semestrov a je ukončené štátnou skúškou a obhajobou záverečnej práce. V súlade s korpusom pre *študijný odbor: učiteľstvo akademických predmetov – informatika* [6] sa očakáva, že absolventi okrem všeobecných pedagogických poznatkov a zručností získajú (vyberieme najdôležitejšie):

A. Odborné poznatky, t.j. budú

- poznať a rozumieť základom vedeckého odboru informatika, získať pevný základ, ktorý dovoľuje celoživotne si obnovovať znalosti súbežne s rozvojom oblasti,
- vedieť klasifikovať a riešiť problémy súvisiace s informatikou, poznať vhodné nástroje na ich riešenie. Vedieť kriticky vyhodnocovať riešenia, testovať ich, porovnávať a prezentovať,
- poznať rôzne stupne informatizácie spoločnosti, chápať ekonomické, sociálne, etické a právne aspekty používania informačných a komunikačných technológií (ďalej IKT) a informatiky v spoločnosti, v škole, v rodine a v regióne.

B. Pedagogické poznatky a zručnosti – špecifické pre informatiku, t.j. budú

- mať IKT zručnosti na vyššej úrovni,
- mať hlboké znalosti z didaktiky vyučovania informatiky (pre druhý stupeň ZŠ, pre SŠ),
- vedieť riadiť projekty žiakov a študentov využívajúce IKT – pre informatiku a pre všetky predmety,
- sledovať vývoj technológií, sledovať stav informatizácie spoločnosti a sprostredkovávať ho žiakom.

Študijné plány pre RŠI splňujú požiadavky akreditačnej komisie, majú zaradené odborné predmety, ktoré sledujú vzdelávací korpus. Na UPJŠ má toto štúdium dlhodobú tradíciu (približne 15 rokov) a v študijných skupinách sa nájdu učitelia, ktorí sa sťažujú, že študijný program málo sleduje tie informatické predmety, ktoré oni vyučujú, resp. sa pripravujú na ich výučbu. Máme zaradený predmet Seminár zo stredoškolskej informatiky, ktorý orientujeme na prácu s talentovanou mládežou. Na tento seminár zaradíme aj podrobnejší prehľad požiadaviek na maturitných skúškach z informatiky.

Záverečné práce na rozširujúcom štúdiu Informatika sú dôležitým výsledkom štúdia, v ktorom absolventi preukážu samostatné zvládnutie prednášaných predmetov a technológií. Celé štúdium trvá päť semestrov, v priebehu štvrtého semestra poslucháči si vyberajú problematiku, ktorú budú spracovávať

v záverečnej práci, a koncom štvrtého semestra je vytvorené jej zadanie.

4. ANALÝZA TÉM ZÁVEREČNÝCH PRÁC

Príprava témy záverečnej práce sa spravidla začína vo štvrtom semestri, začiatkom semestra poslucháči sa zúčastňujú na obhajobách záverečných prác predchádzajúceho ročníka, kedy si vytvoria predstavu o riešených problémoch a obhajovaných prácach. Poslucháči tak dostanú prvý silnejší impulz na zamyslenie sa nad záverečnou prácou, ktorá ich čaká. V literatúre a v študijných programoch sa stretneme s nasledujúcou charakteristikou záverečnej práce [5, 6]:

Záverečnou prácou poslucháč prezentuje svoju pripravenosť vyučovať informatiku na 2. stupni ZŠ a na SŠ, v zmysle aktuálnych pedagogických dokumentov a v zhode s modernou didaktikou informatiky. Poslucháč musí preukázať primeranú znalosť informatiky a jej didaktiky pre sekundárne vzdelávanie. Práca musí byt originálna, vytvorená autorom pri dodržaní pravidiel práce s informačnými zdrojmi, nesmie mať charakter plagiátorstva, nesmie narušať autorské práva iných autorov. Súčasťou záverečnej práce je aj písomná dokumentácia.

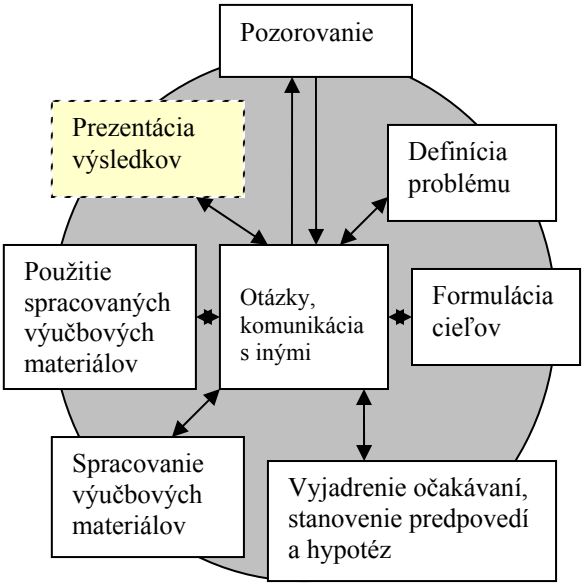
Rozširujúce štúdium je nadštandardné vysokoškolské štúdium, poslucháči už absolvovali jednu vysokú školu, už spracovali jednu diplomovú prácu. Záverečná práca na RŠI sa robí jeden semester, vlastne až po absolvovaní podstatnej časti informatických predmetov na tom to štúdiu, pretože sa predpokladá, že poslucháči uplatnia v nej získané vedomosti a zručnosti. Vzhľadom na krátky čas (jeden semester) je veľmi dôležité stanoviť vhodný rozsah spracovávanej problematiky, aby boli v práci preukázané didaktické princípy, aby poslucháči stihli pripraviť materiály pre výučbu, aby stihli absolvovať pedagogický experiment s vytvoreným výučbovým materiálom a spracovať skúsenosti z experimentu.

Analýza prác predchádzajúcich dvoch ročníkov, v ktorých sme mali 38 absolventov tohto štúdia, hovorí o nasledujúcom zložení prác podľa zamerania:

- **Práce zamerané na spracovanie výučby určitej témy v predmete inom než informatika.** Napríklad, CabriJava applety vo výučbe významných prvkov trojuholníka, Edukačný hypertext zameraný na problematiku druhov drevín, Výučba tém násobenie a delenie využitím IKT, atď.
- **Práce zamerané na výučbu niektorých oblastí informatiky.** Napríklad, Výučba tvorby bitmapových animácii v prostrediach Logomotion a Imagine Logo, Výučba témy spracovanie zvukov na počítači na základnej škole, Algoritmizácia a programovanie úloh z logiky, Inscenačná metóda vo výučbe programovania na základnej škole, atď.
- **Práce zamerané na podporu výučby vo všeobecnosti**. Napríklad, Generovanie písomných testov zo základoškolskej matematiky, atď.
- **Práce zamerané na výučbu špeciálnych predmetov na odborných stredných školách.** Napríklad, Podporný softvér pre programovanie NC strojov, atď.

5. UPRAVENÝ MODEL KOLESA AKTIVÍT

Model kolesa bádania, ktorý sme sa rozhodli použiť a modifikovať podľa našich názorov, vychádza z modelu uvedeného v [1, 2, 3].



Obr. 1 Upravený model kolesa aktivít

- **Pozorovanie.** Je tu myslené také pozorovanie, ktoré prináša množstvo otázok, na ktoré sú hľadané odpovede. Tieto pozorovania môžu byť získané rôznym spôsobom (čítaním odbornej alebo vedeckej literatúry, našimi zmyslami, v laboratóriách, atď.). V prípade záverečných prác sa toto pozorovanie týka napríklad hľadania témy, ktorá sa vyučuje v škole na 2. stupni ZŠ alebo na SŠ, ku ktorej riešiteľ má určitý vzťah a predpokladá výučbu tejto témy použitím informačných technológií získaných didaktických poznatkov. Je dobré, keď poslucháči hľadajú námety na témy vo vlastnej výučbe, pretože dosiahnuté výsledky vedia využiť vo svojej učiteľskej praxi.
  Napríklad, poslucháč RŠI pozoruje u žiakov na základnej škole, že pri delení a násobení žiaci robia nejaké chyby. Napadne mu, že by sa mohol

touto problematikou viac zaoberať. Začne sa radiť s vyučujúcimi a kolegami.

- **Definícia problému**. Vychádzajúc z pozorovaní, poznatkov z literatúry, komunikácie s učiteľmi, poslucháči identifikujú problémy, ktoré by sa pri hľadaní riešenia mohli vyskytnúť. Na základe svojich skúseností, na základe skúsenosti komunity, v ktorej pracujú, na základe odporúčaní vyučujúcich sa musia rozhodnúť, akú problematiku sú schopní didakticky spracovať a pripraviť pre výučbu.
  Napríklad, učiteľ po diskusiách navrhne tému svojej práce Výučba tém násobenie a delenie využitím IKT.
- **Formulácia cieľov.** Mnohokrát sa stane, že ciele práce sú sformulované pomerne všeobecne a pomocou otázok, ktoré sme si položili. Tieto otázky sú zo začiatku možno neúplné, možno redundantné. Je tu celý proces, kým sa sformulujú výsledné ciele pre prácu a úlohy, ktoré majú byť splnené. Učebnice, ale obsahujú už ten výsledok a nesledujú ten proces jeho tvorby.
  Napríklad, pre vyššie uvedenú tému spolu s konzultantom práce boli stanovené ciele:
  - *Vymedziť základné učivo tematického celku násobenie a delenie v predmete matematika štvrtého ročníka ZŠ.*
  - *Navrhnúť a vytvoriť učebný hypertext s prvkami aktívneho čítania, so zbierkou úloh a návodmi na ich riešenia s dôrazom na motiváciu žiakov.*
  - *Navrhnúť metodiku vyučovania tohto tematického celku pomocou vytvoreného hypertextu.*
  - *Overiť využiteľnosť hypertextu a navrhnutú metodiku v praxi.*
- **Vyjadrenie očakávaní, stanovenie hypotéz.** Poznatky o doterajšej výučbe a pochopenie problémov, ktoré pri nej vznikajú, vedú poslucháča k tomu, že formuluje predbežné hypotézy o zlepšení vedomostí žiakov. Tieto predbežné hypotézy sa nakoniec buď potvrdia alebo vyvrátia. Bolo by dobre tu vyžadovať vysvetlenie vlastného didaktického prínosu k spracovávanej téme.
- **Spracovanie výučbových materiálov.** V tomto bode poslucháč už musí mať vymedzený rozsah výučby (ktorý nesmie byť prehnane široký), musí byť rozhodnutý, akým spôsobom výučba bude prebiehať, už musí mať premyslené použitie didaktických princípov. Spracovaniu materiálov predchádza rozsiahla niekoľko kroková diskusia.
- **Použitie spracovaných výučbových materiálov a ich vyhodnotenie.** Tento krok je závislý od témy, pre ktorú majú byť učebné materiály pripravené a od spôsobu spracovania materiálov. Môžu to byť doplnkové študijné materiály, môže to byť pomôcka pre učiteľa na hodine, a pod. Od poslucháčov očakávame, že vytvorené materiály vo svojej výučbe využijú a popíšu skúsenosti s touto výučbou a odporúčania na použitie materiálov. Mnohí dokladajú analýzu dotazníkov, ktoré k danej výučbe vytvorili.
- **Prezentácia výsledkov.** Tento krok môže byť rozdelený do viacerých častí. Dôležité sú aj prezentácie čiastočných výsledkov. Umožňujú poslucháčovi získať ucelenejší pohľad na dosiahnuté výsledky. Mala by sa uskutočniť aspoň jedna verejná prezentácia dosiahnutých výsledkov, aby poslucháči mohli skorigovať prípadne postupy nesprávnym smerom.
- **Centrum deja - Otázky, komunikácia s inými.** V priebehu celého postupu je dôležité komunikovať s okolím, so spolupracovníkmi. Otázky a odpovede na tieto otázky sú jadrom celého procesu v rámci kolesa aktivít. Poslucháči sa navzájom ovplyvňujú a navzájom motivujú. Je celkom zaujímavé sledovať ich spoluprácu.

.

6. ZÁVER

V príspevku sme poskytli informácie o dvoch vzdelávacích aktivitách zameraných na rovnakú skupinu učiteľov. Mnohé dobré skúsenosti zo vzdelávania v jednej sú použiteľné aj v tej druhej. Pretože očakávame v jednom roku (2011) až 200 záverečných prác, bolo by dobre mať premyslený systém ich zadávania, aby ich výsledky boli prínosom vo výučbe a aby boli použiteľné aj inými vyučujúcimi. Model kolesa aktivity nepovažujem za dokonalý, ale môže poslúžiť ako východisko pri hľadaní systému na zadávanie záverečných prác. Tento model môže pomôcť pri tom, aby na práce boli kladené rovnaké nároky a aby zadania prác boli premyslené a splniteľné.

LITERATÚRA

[1] W. Harwood: A New Model for Inquiry, *Journal of College Science Teaching,* Vol. 33, No. 7, July/August 2004

[2] W. Harwood: An Activity Model for Scientific Inquiry. *The Science Teacher,* Vol. 71, No. 1, 2004.

[3] G. Andrejková: Model aktivity založený na príncipe vedeckého bádania, Zborník prednášok *DidInfo 2007, UMB,* B. Bystrica

[4] Z. Kubincová, M. Chalachanová: DVUi – národný projekt ďalšieho vzdelávania učiteľov základných škôl a stredných škôl v predmete informatika, Zborník prednášok *DidInfo 2009,* B. Bystrica

[5] Národný projekt: Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika, Podrobný opis projektu, Interný materiál ŠPU, Bratislava, 2008

[6] Korpus študijného odboru – učiteľstvo akademických predmetov – informatika, 2003

MODELOVANIE A SIMULÁCIA V ŠKOLSKEJ MATEMATIKE A FYZIKE S EXCELOM

PAEDDR. JÁN BEŇAČKA, PHD.

Katedra informatiky, Fakulta prírodných vied, Univerzita Konštantína Filozofa,
Tr. A. Hlinku 1, Nitra, e-mail: jbenacka@ukf.sk

ABSTRAKT

Článok prináša niekoľko príkladov použitia Excelu ako efektívneho nástroja na modelovanie a simuláciu v rámci štúdia matematiky a fyziky na strednej škole a to z pohľadu výučby príslušných zručností v narábaní s týmto nástrojom.

Kľúčové slová: *Excel, modelovanie a simulácia, výučba matematiky, fyziky a informatiky*

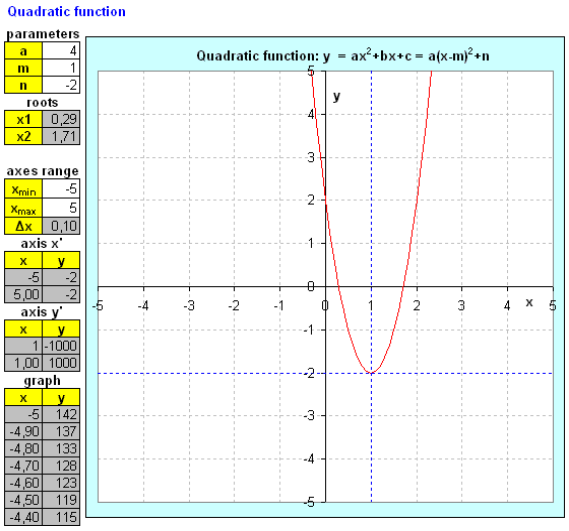
1. ÚVOD

Tradičným spôsobom výučby práce s tabuľkovým kalkulátorom je vytváranie rôznych štatistických tabuliek, výpočtov nad nimi a ich grafická interpretácia [1]. Študenti ďalej využívajú tieto vedomosti a zručnosti pri vyhodnocovaní fyzikálnych, chemických a biologických experimentov v rámci laboratórnych cvičení alebo pri analyzovaní geografických a ekonomických dát v rámci príslušných predmetov. Iným, menej známym spôsobom využitia tabuľkového kalkulátora je tvorba interaktívnych modelov parametrických systémov pre štúdium ich správania sa pri meniacich sa hodnotách vstupov [2]. Takéto modely sú ideálne pre výučbu a štúdium matematiky a fyziky (pozri [5]), a sú mimoriadne účinné pri budovaní vizuálnej pamäte. Pri vytváraní modelu študenti získavajú a osvojujú si nové vedomosti a zručnosti v duchu konštruktivizmu [3] a objaviteľskej matematiky [4]. Článok prináša niekoľko príkladov použitia tabuľkového kalkulátora MS Excel™ ako efektívneho nástroja na modelovanie a simuláciu v rámci výučby a štúdia matematiky a fyziky na strednej škole a to z pohľadu výučby príslušných zručností v narábaní s týmto nástrojom. Niektoré z uvedených aplikácií boli vytvorené v rámci výučby predmetov Vedecké výpočty v tabuľkových kalkulátoroch a Didaktika informatiky. Modely neobsahujú makrá. Čitateľ nájde ďalšie zaujímavé príklady modelovania s Excelom v [2], [6].

2. EXCEL A MATEMATIKA

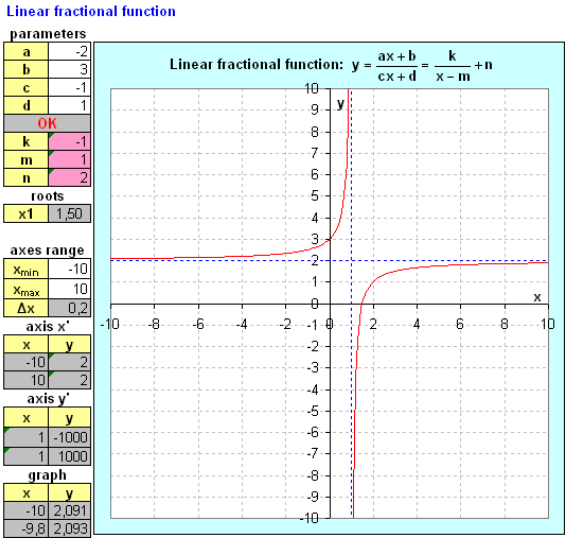
Už počas mojej učiteľskej praxe na gymnáziu, kde som učil matematiku, fyziku a informatiku, som začal vyvíjať v Exceli aplikácie na vizualizáciu matematických vzťahov [7], [8]. Overoval som ich na viacerých školách s pozitívnymi odozvami zo strany študentov i učiteľov. Niektorí boli nadšení možnosťami Excelu a jednoduchosťou požívania aplikácií. Uvádzam niekoľko ukážok aplikácií, ktorých vývoj trvá rádovo minúty, ak je tvorca dostatočne zručný v práci s Excelom.

Na Obr. 1 je aplikácia pre kreslenie grafu kvadratickej funkcie. Biele bunky sú vstupy. Korene sú počítané automaticky. Graf je typu xy cez 100 bodov. Posunuté osi x' a y' sú dvojbodové xy grafy.



Obr. 1 Kvadratická funkcia

Graf interaktívne reaguje na zmeny parametrov *a*, *m*, *n*, čo umožňuje študentom vyšetrovať ich vplyv na tvar grafu a tieto súvislosti si vizuálne pamätať.

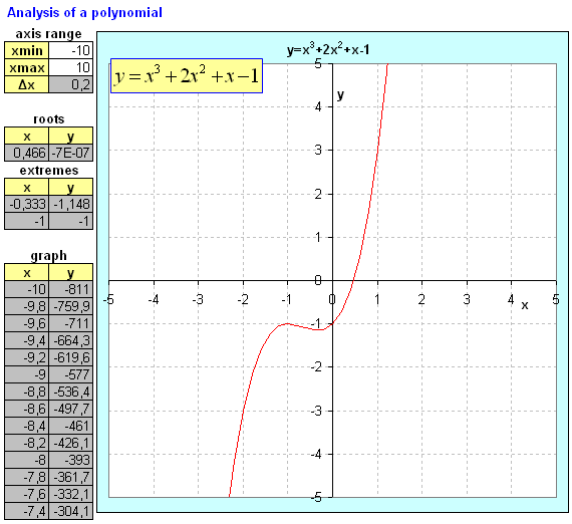


Obr. 2 Lineárna lomená funkcia

Poznámka: Rozsah osí grafu treba meniť ručne.

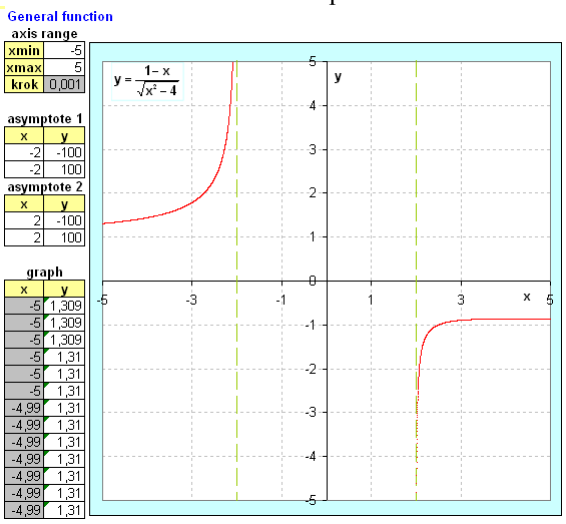
In Obr. 2 je aplikácia pre kreslenie grafu lineárnej lomenej funkcie. Parametre a, b, c, d sú kontrolované funkciou IF či $ad \neq bc$. Ak áno, tak je vypísané „OK“, inák je vypísané „no“. Koreň je počítaný automaticky. Graf pozostáva z dvoch vetiev, ktoré sú počítané v skrytých stĺpcoch. Ľavá vetva ide od $x_{\min}$ po m, pravá od m to $x_{\max}$. Oblasť “graph” je len „didaktickou tabuľkou na konštrukciu grafu”. Tento trik je užitočný, pretože údaje v jednotlivých vetvách sú pre užívateľa často mätúce. Je ho vhodné použiť pri konštrukcii grafov všetkých elementárnych funkcií, ktorých definičný obor nie je R, ako sú mocninná funkcia, logaritmická funkcia, tangens, atď.

Na Obr. 3 je aplikácia pre analýzu polynómu. Korene a extrémy sú počítané pomocou nástroja Riešiteľ (Solver).



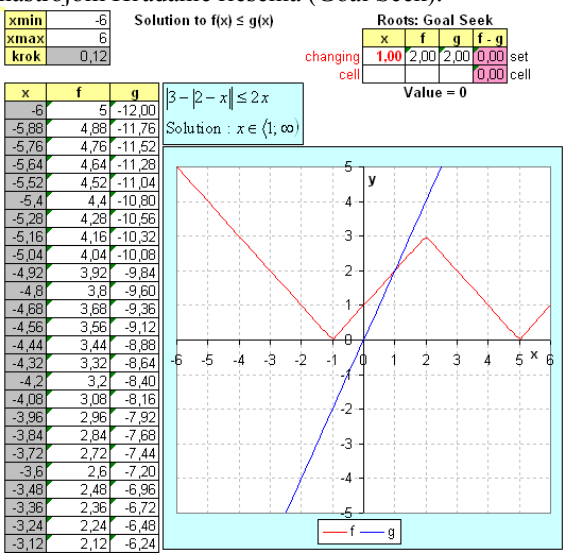
Obr. 3 Polynóm

Na Obr. 4 je aplikácia pre analýzu všeobecnej funkcie. Keďže definičný obor nie je R, je použitý bodový xy graf cez veľa bodov ošetrených funkciou ISERROR. Aplikácie na Obr. 3 a 4 študenti využijú v rámci štúdia diferenciálneho počtu.



Obr. 4 Všeobecná funkcia

Na Obr. 5 je aplikácia pre riešenie rovníc a nerovníc. Riešením je interval, v ktorom je červený graf pod modrým. Priesečník je počítaný nástrojom Hľadanie riešenia (Goal Seek).

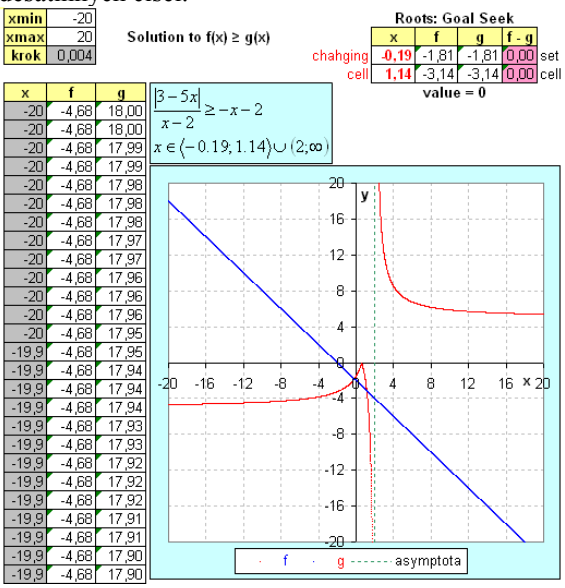


Obr. 5 Riešenie rovníc a nerovníc

Na Obr. 6 je riešenie omnoho komplikovanejšej úlohy. Jej analytické riešenie je

$$x \in \left\langle \frac{5-\sqrt{29}}{2}; \frac{-5+\sqrt{53}}{2} \right\rangle \cup (2;\infty),$$

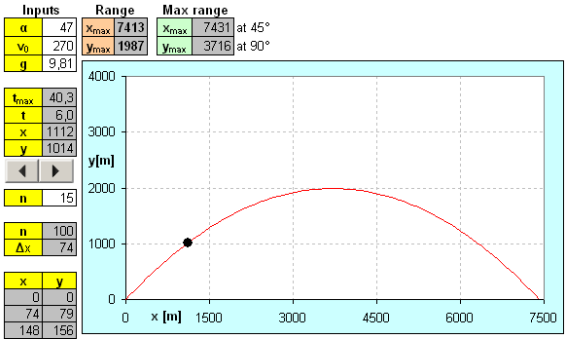
a dospieť k tomuto výsledku si vyžaduje zhruba pätnásť minút sústredenej práce a dve – tri strany A4. Riešenie Excelom je hotové omnoho rýchlejšie, je prehľadné a vierohodné. “Presné” analytické riešenie bude nakoniec aplikované aj tak len v tvare desatinných čísel.



Obr. 6 Riešenie rovníc a nerovníc

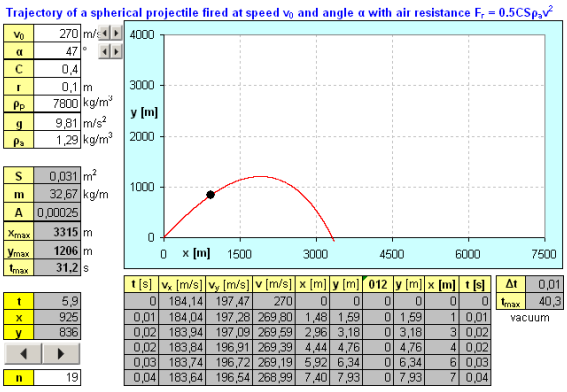
3. EXCEL A FYZIKA

Na Obr. 7 je aplikácia pre znázornenie šikmého vrhu vo vákuu. (pozri [5]). Trajektoriálna parabola je xy čiarový graf počítaný cez 100 bodov. Najprv je podľa známeho vzorca [9] vypočítaný horizontálny dostrel. Ten je rozdelený na 100 bodov x. Súradnice y sú počítané podľa známeho vzorca. Vstupy sú v bielych bunkách a graf reaguje interaktívne na ich zmeny. Známy fakt, že dostrel je maximálny pri uhle 45° je možné ihneď overiť. Pre simuláciu pohybu projektilu je najprv vypočítaná doba letu t_{max}. Tento čas je rozdelený na 100 bodov, v ktorých sú počítané súradnice xy projektilu. Projektil je jednobodový xy graf. Aktuálny čas projektilu je počítaný ako prírastkový diel času t_{max} a je ovládaný spriahnutým číselníkom (spinbutton). Kliknutie na číselník predstavuje posun na ďalší časový bod. Model verne kopíruje realitu – smerom nahor sa pohyb spomaľuje a smerom nadol sa zrýchľuje.



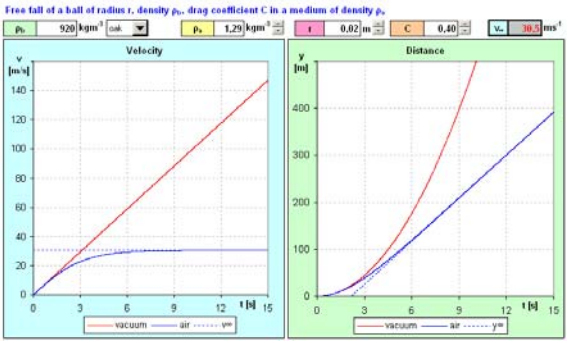
Obr. 7 Šikmý vrh vo vákuu

Na Obr. 8 je aplikácia na znázornenie pohybu guľového projektilu v atmosfére pri rýchlostiach do 300 m/s, keď odpor je úmerným kvadrátu rýchlosti (viď. Newtonov zákon [9]). Ide o ukážku trajektórie gúľ vrhnutých z prakov a balíst. Integrálne riešenie problému je známe [10], ale je príliš komplikované pre použitie na strednej škole. Problém je riešený numericky pomocou Eulerovej metódy, ktorá je dostatočne jednoduchá a zrozumiteľná pre žiakov štvrtého ročníka, ktorí sú oboznámení so základmi diferenciálneho počtu. Trajektória je xy čiarový graf počítaný cez 5000 bodov. Doba letu je zhora odhadnutá dobou letu vo vákuu. Dostrel je počítaný pomocou stĺpca 012, kde je zapísané číslo 1, ak súradnice y zmení znamienko. Bod dopadu a doba letu sú potom počítané lineárnou interpoláciou medzi poslednou kladnou a prvou zápornou hodnotou súradnice y, ktoré sú vrátené funkciou VLOOKUP. Simulácia je urobená rovnako ako v predchádzajúcom príklade. Dá sa jednoducho ukázať, že v tomto prípade je dostrel maximálny pri uhle vždy menšom ako 45° – v prípade vstupov podľa Obr. 8 je to 39°. Presnosť modelu je výborná – ak je hustota prostredia nulová, t. j. pohyb sa koná vo vákuu, dostávame dostrel $x_{max} = 7412$ m, čo je len o 1 m menej ako je presná hodnota podľa modelu na Obr. 7 (relatívna chyba 0.0135 %).



Obr. 8 Pohyb guľového projektilu v atmosfére

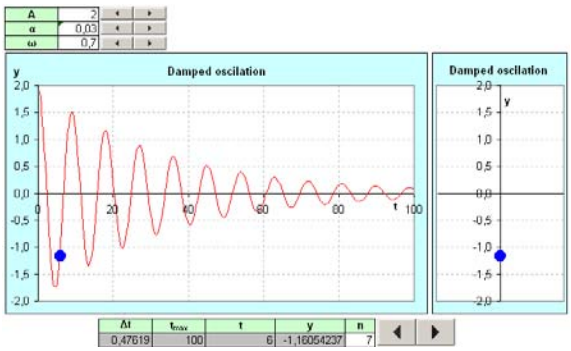
Na Obr. 9 je aplikácia znázorňujúca graf rýchlosti a dráhy voľne padajúcej gule v atmosfére (kvadratický odpor). Analytické riešenie je známe (viď. napr. [11]). Obsahuje hyperbolické funkcie, ktoré sú dnes už štandardnou súčasťou kalkulačiek, (napríklad tej vo Windows), a študenti sa na ne často pýtajú. Použité sú xy čiarové grafy cez 100 bodov. Asymptoty sú dvojbodové xy grafy a znázorňujú fakt, že po uplynutí dostatočne dlhého času sa pohyb telesa stáva rovnomerným, t. j. rýchlosť bude konštantná funkcia a dráha bude lineárna funkcia. Číselníky umožňujú pohodlne a promptne meniť hodnoty vstupov, a rozbaľovacie pole (checkbox) umožňuje meniť hustotu látky, z ktorej je guľa zhotovená. Aplikácia umožňuje demonštrovať vplyv vstupov na výslednú rýchlosť telesa – táto rýchlosť je väčšia, ak je guľa väčšia pri tej istej hustote, ak je jej hustota väčšia pri tej istej veľkosti, ak je koeficient odporu C menší (hladšia guľa), alebo ak je hustota atmosféry menšia.



Obr. 9 Voľný pád gule v atmosfére

Na aplikácii je tiež možné demonštrovať (a následne matematicky dokázať) päť limitných prípadov: 1) ak sa koeficient C blíži k nule, tak grafy pre pohyb vo vzduchu splynú s grafmi pre pohyb vo vákuu (2 limity); 2) ak sa čas t blíži k nekonečnu, tak grafy splynú s asymptotami (3 limity, z toho 2 pre asymptotu dráhy). Podotýkam, že v učebniciach matematiky sa vyskytuje množstvo náročných príkladov na výpočet limit, ktoré nemajú reálnu podstatu. Preto považujem túto úlohu a model za veľmi užitočné.

Na Obr. 10 je aplikácia pre znázornenie pohybu tlmeného mechanického oscilátora. Graf je kreslený cez 210 bodov, pričom tri možné riešenia [12] pohybovej rovnice sú oddelené funkciou IF v závislosti od hodnôt vstupov. Model umožňuje ukázať i anharmonický pohyb, keď tlmenie je také veľké, že kmitanie nenastane, napr. ak je oscilátor ponorený do medu.



Obr. 10 Tlmené kmitanie

Záujemca môže nájsť numerické modely voľného pádu v atmosfére a tlmeného kmitania v [13]. V [14] sú tri rôzne modely matematického kyvadla.

4. ZÁVER

Cieľom článku je ukázať čitateľovi možnosti, ktorá ponúka Excel v oblasti modelovania v školskej matematike a fyzike. Ako bolo uvedené v úvode, prezentované aplikácie využívajú len štandardné funkcie Excelu, t. j. bez programovania. Ide o písanie vzorcov s relatívnymi a absolútnymi adresami, plnenie smerom nadol, použitie xy grafov, nástrojov Hľadanie riešenia a Riešiteľ, funkcií IF, ISERROR, a VLOOKUP, komponentov spinbutton, scrollbar (nie je uvedený), checkbox, a zmena rozsahov osí na grafe (ak je nutné). Výsledok stojí za námahu. Nadobudnuté zručnosti otvárajú pred študentmi so záujmom o matematiku a fyziku nový horizont. Považujem tento aspekt za dobrý dôvod na to, aby uvedený spôsob práce s tabuľkovým kalkulátorom bol zahrnutý do učebných plánov informatiky na strednej škole.

REFERENCES

[1] Lukáč, S., Šnajder, Ľ.: Práca s tabuľkami: Tematický zošit pre stredné školy a ZŠ s triedami so zameraním na matematiku, Bratislava : SPN, 2001. ISBN 80-08-02987-0

[2] Neuwirth, E., Arganbright, D.: The Active Modeler: Mathematical Modeling with Microsoft Excel. Belmont (CA): Brooks/Cole, 2004. ISBN 0-534-42085-0

[3] Lovászová, G., Turčáni, M.: Konštruktivizmus vo vyučovaní informatiky. In: *Technológia vzdelávania*. **12**, 2004, pp. 2- 4. ISSN 1335-003X

[4] Kalaš, I.: Vizuálne interaktívne objekty ako nová forma v poznávacom procese. In: *Poškole 2004, Sborník národní konference o počítačích ve škole*. Liberec : Mezinárodní organizační výbor Poškole, Jednota školských informatiků, 2004. ISBN 80-239-2598-9

[5] Kajetanowicz, P.: Motivating interactions - teaching informatics by math and physics models. In: *Didinfo 2007*, zborník príspevkov z medzinárodnej konferencie. Banská Bystrica : Fakulta prírodných vied UMB, 2007. ISBN 978-80-8083-367-1

[6] http://epublications.bond.edu.au/ejsie/

[7] Beňačka, J.: Excel v gymnaziálnej matematike. Nitra : FPV UKF, Prírodovedec č. 173, 2005, ISBN 80-8050-853-4

[8] Beňačka, J.: Stredoškolská stereometria v Exceli. Nitra : FPV UKF. Prírodovedec č. 219, 2006, ISBN 80-8094-040-1

[9] Vachek, J. a kol.: Fyzika pre 1.ročník gymnázia. Bratislava, SPN, 1984

[10] Hayen, J. C.: Projectile motion in a resistant medium. In. *Int. J. Non-Linear Mech*. **38** (3), 2003, pp. 357-369. ISSN 0020-7462

[11] Beňačka, J., Štubňa I.: Accuracy in computing acceleration of free fall in the air. In: *The Physics Teacher*, **43**, 2005, pp.432 – 433. ISSN 0031-921x

[12] http://sk.wikipedia.org/wiki/Kmitanie

[13] Beňačka, J.: Spreadsheet numerical modelling in secondary school physics and biology. In: *Spreadsheets in Education*, **2**, 2008, pp. 289-298, ISSN 1448-6156

[14] Beňačka, J.: Three spreadsheet models of a simple pendulum. In: *Spreadsheets in Education*, **3**, 2008, pp. 11-13, ISSN 1448-6156

OPENING THE GATE TO OBJECT ORIENTED APPROACH IN TEACHING PROGRAMMING

PAEDDR. IVAN BRODENEC

Katedra informatiky, Fakulta prírodných vied, Univerzita Mateja Bela, Tajovského 40, Banská Bystrica, tel. 048/446 7128, e-mail: brodenec@fpv.umb.sk

This article gives some information and results of our research in the area of object-oriented programming and its implementation at the secondary schools. We show some models which were used as part of special seminar in the „Science Gate Open“ project which is being held at our university.

Keywords*: object-oriented approach,teaching of informatics, project*

1. INTRODUCTION

One would maybe say that it is not necessary to make informatics more popular among students at the secondary schools. Informatics is almost in any part of our life and we definitely do not have to say that informatics is unpopular. Many students who are interested in this subject search for news in every area of this science. They ask for information which they don't get at school. It's usually not covered by curriculum either because of small number of informatics lessons at the school or because of not enough documents which support these new ways of teaching informatics.

The aim of the project “Science gate open” is to prepare activities which would achieve this spreading of science at the schools. This is done at special seminars taking place at the school, one-time lectures on given topic and students' grants which allow the students to do little science on their own.

2. THE SEMINAR ABOUT OOP

This activity which has taken place at the Secondary Grammar School of Andrej Sladkovic was not only opening of door to object-oriented programming (hence the title of this article) but also a possibility to present and verify some of our topics which are done as part of our research in this field.

The main idea of this model of teaching programming is to show a paradigm which works in practice many years. Object oriented programming does not seem to be just fashion wave like other approaches in history of programming.

If we take the history into consideration, these approaches started from computer-oriented ways of programming like assembler, through structured and modular programming, to the more human-oriented way of programming. To resolve this term we can use many objects from the real life and the way we work with many of them. It is our opinion that if we would like to teach programming, we should go in the opposite direction to that historical one. It means that one can better understand all the topics in programming if starting with object-oriented approach.

If we talk about topics of this seminar, the main objective was to learn the principles of OOP in the development environment Delphi. The skeleton of the seminar is formed by these topics:

- Basic prepared components
- Own basic object, encapsulation
- Use of basic object in more complex object
- Inheritance, hierarchy of objects
- Polymorphism

The idea of objects and their behavior is very close to the man. Around us there are many objects that have their basic properties, can respond to different events, and send us or to each other different messages. Doing so, in many cases we do not need to know how exactly the object works, how it's implemented. For us it's enough to know what options we have to control the object. Remaining parts can work for us as a black box. This is the first of the object-oriented programming properties (encapsulation), which is to be learned in this approach.

In our model we look at this concept from two aspects when teaching programming. First is the object manipulation, use of their properties and execution of methods which will solve the problem we are given. Second one is the design of simple object in the real world which will be later extended with other OOP properties.

We use integrated development environments for the first aspect. In this way we have two possibilities to work with it. We can have a set of objects which is given to us by this environment, or the teacher can create his own microworld, which is a interesting motivation when used with a theme which students like. Regardless of which programming language we use and what kind of supporting software is used, it is the way to learn the basics of object creation and manipulation. Because the students which were part of this seminar were interested in understanding the Delphi environment, we used this one for it. As we

already mentioned, we combined the two aspects. First we used the components which are available through visual components library where we use basic objects which support easy input and output of values and graphical representation (TButton, TEdit, TCanvas, TLabel). We need to understand just behavior of them, knowing the properties and methods which are available. One of the most important objects from these is in our opinion the canvas which gives the motivation that we see what we create graphically.

Object that we started to design after getting the experience with object manipulation was a simple traffic light at the junction, which consists of three lights. The first object was therefore TLight, with basic properties and methods which were discussed with students. Implementation of them is of course the way to familiarize with the basic rules of language syntax, which we started to use. It's useful to say that learning the syntax should not overshadow the object design. It is the same as the object manipulation in the previous step. We do not need to know the background of the object, we need to know just how it works. The same idea works also with design of object, so we will discover the properties and rules of syntax gradually.

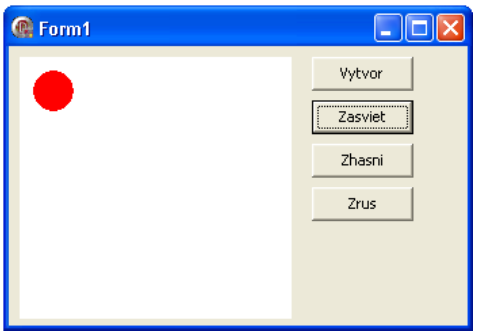


Fig.1

Students can manipulate with this object, create more instances of the class with different colors and prepare themselves for creating a streetlight. They of course don't just design this more complex object but they also implement it.

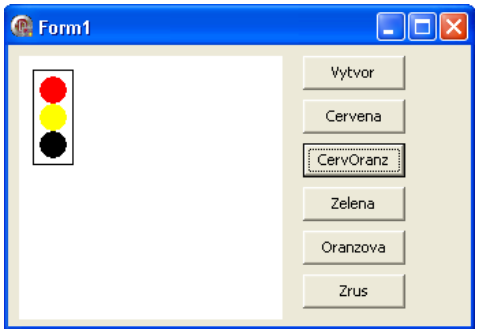


Fig. 2

The same application is done in two ways. A sequence of commands is performed at first by pressing the buttons. These actions work with original TLight objects. After transforming these commands into a new object methods of TStreetLight, buttons serve only to carry out a specific method. I can say that this form of design worked well at the seminar. Student has the option to directly manipulate objects and solve the specific problem. The problem is then transformed into methods of the new object without the need to deal with complex issue from beginning, which could be difficult for him to handle.

Inheritance is the next important property in object-oriented programming. Again there are two ways of implementing it into the informatics lessons, each of which may focus on specific skills we want the student to learn. This handles either creating a hierarchy of objects starting with our base object which was already defined or extending the functionality of prepared object which can be used for learning some other techniques and algorithms.

Polymorphism is a property which is very closely related with inheritance. You can really quickly find it's functionality in the presented streetlight example. If the students creates a new light with different shape for example, he can use it in the same TStreetLight object and it will work with no problem. The concept of polymorphism differs a little bit depending on language we use. Again there is idea not to “roomwash”. It means not to concentrate on specific restrictions which are in the language but use generally used properties which are applicable anywhere.

3. BACKTRACKING AND OOP

Another activity covers more advanced topic in the teaching of programming. It is again our opinion that we can start this topic using a prepared object which will first act as a black box and can be used to improve skills in manipulating objects and using the properties of object-oriented programming to extend the functionality of the object. After the first use we can look inside the object and learn the fundamentals of backtracking as one of programming methods.

```
type
  TKon = class
  private
    p: integer;
    terminated: boolean;
    procedure backtrack(x,y:integer);
  public
    sachovnica: array[1..8,1..8] of integer;
    priNajdeniRiesenia: TRiesenieUlohy;
    predZaznacenimTahu: TZaznacenieTahu;
    priZaznaceniTahu: TZaznacenieTahu;
    priOdznaceniTahu: TZaznacenieTahu;
    poOdznaceniTahu: TZaznacenieTahu;
    constructor Create;
    procedure Ries(x,y:integer);
    procedure Terminate;
  end;
```

Fig. 3

Activity is based on understanding the principles of backtracking on a given problem which can be solved this way. Student will just extend the functionality of the object either by assigning some event handler of the object or by using inheritance and creating ancestor of the given object.

At the end he can see some result of his work, understands the basic concept of backtracking and those students who like to know more have possibility to look into the base object and try to understand how it works.

4. DATA STRUCTURES

Similar idea as in the previous chapter works also in this model. One of the aims of teaching informatics at secondary schools is to teach algorithms and the way they can be solved. Of course many of these algorithms are solved using different data structures but in many problems it is not necessary the first information the student should get when solving some problem.

In many situations we just use some synonym for these data structures. In our model we use a crane which works with belt conveyor and boxes on it. It can take the first box, put it on surface to form a pile of boxes or it can put the box on the belt conveyor again as the last one there. Using these two structures we can model many simple and also more advanced algorithms like choosing only boxes with given property, sorting the boxes using some property and many more if we add some other functionality like rewriting the label on box, creating a box with same label as the one we hold in crane etc.

5. SUMMARY

Given activities and the results from them make me believe that this opening of door to object-oriented programming might be successful when used in school informatics at secondary schools.

LITERATURE

[1] KALAŠ, I.: Pedagogický výskum v informatike a informatizácii (1.časť) In Didinfo 2008, Banská Bystrica, 2008, s.50.
ISBN 978-80-8083-556-9

[2] BRINDA, T.: Didaktisches System für objektorientiertes Modellieren im Informatikunterricht der Sekundarstufe II., Siegen, 2004

[3] GABAĽOVÁ, V.: Mikrosvet – vhodný prostriedok na vyučovanie základov programovania In Didinfo 2008, Banská Bystrica, 2008, s.50.
ISBN 978-80-8083-556-9

[4] BRODENEC, I.: Problem solving in OOP, In XXI. Didmattech, Eger 2008

K SENZOROM V NAŠICH ZÁKLADNÝCH ŠKOLACH

PETER CVIK.

Katedra aplikovanej informatiky FMFI Univerzita Komenského Bratislava,
EDUXE, Bratislava
tel. 0905/580 657, e-mail: eduxe@internet.sk

K mapovaniu procesov prebiehajúcich v technickom, prírodovednom a biologickom prostredí sa používajú senzory. Ich výhodou je okrem iných aj možnosť opätovnej prezentácie získaných dát a následná práca s týmito dátami. Učiteľom a praxi sú bežne známe senzory na fyziku, chémiu, geometriu a ďalšie prírodné vedy. V súčasnosti sú u nás v oblasti vedeckého záujmu aktuálne senzory na biológiu a senzory uplatňované v zdravotníctve. V príspevku sú uvádzané senzory k mapovaniu základných fenoménov z oblasti fyziky, chémie, biológie a ekológie dostupné na našich základných a stredných školách. V druhej časti sú prezentované skúsenosti získané pri práci s týmito senzormi prezentované na školeniach EDUXE k RoboLabu.

Kľúčové slová: *didaktika informatiky, typy fyzikálnych senzorov, konferencia DidInfo, Banská Bystrica*

1. ÚVOD

Senzory sú zariadenia, ktoré umožňujú mapovať okolitý svet pomocou prevodov na elektrickú energiu a následne ju evidovať a predpokladá sa, že informačne spracovávať.

Slovo senzor je možné nájsť v anglicko-českom technickom slovníku z roku 1969 a prekladá sa ako *snímač,* alebo *čidlo.* Názov senzor pravdepodobne pochádza z anglického slova *sensor* a toto z latinského slova *sensus,* čo znamená schopnosť registrácie. Najskôr sa uplatňovalo vo väzbe na zrak a až následne na ostatné zmyslové vnemy.
Vzhľadom na to, že elektrická energia bola z technických energií systematicky spracovaná ako prvá sa aj slovo senzor najskôr objavuje v súvislosti s ňou a až následne ako sa odraz informačného diania dostáva do informatiky.

Senzory sa do našich škôl začali výraznejšie dostávať najmä vďaka aktivitám Infoveku. Cieľom ich uplatňovania bola orientovaná diagnostika najmä technického prostredia, schopnosť mapovania a evidencie sledovaných hodnôt, následná práca s týmito hodnotami až po celkové ovplyvňovanie technickej situácie tak, aby sa dosiahli požadované výsledky. Ďalším výrazným zdrojom uplatnenia bolo stavanie robotov podľa istých požiadaviek a následne riadenie robotov k dosahovaniu stanovených cieľov.

Vývoj senzorov neustále prebieha a dnes sa ich vyrába na stovky a je ich možné členiť podľa viacerých kritérií tak, ako to prax potrebuje. Vzhľadom k zameraniu konferencie a skúsenostiam viacerých z nás uvediem technickú inžiniersku klasifikáciu podľa počtu bitov, ktorá odráža aj informačný potenciál uplatnenia. Senzory sa v súčasnosti delia na kvalitatívne (1. školenie) a kvantitatívne (2. školenie). Kvalitatívne senzory sú schopné analyzovať viac fenoménov jedného prostredia napríklad voda, vzduch, svetlo, plyn a podobne. V našich školách a aj vo svete sa začína s kvantitatívnymi senzormi, ktoré obyčajne merajú jeden fenomén obsiahnutý v názve senzoru. Začalo sa s 8 bitovými senzormi schopnými merať jasne určené veličiny v obmedzenom rozsahu. 16 bitové senzory už aj pri častom uplatňovaní boli schopné presne merať jeden typ hodnôt, prípadne na 4 bitoch ponúkať istý výber a následne ho merať (cena a kvalita). Senzory na 32 bitov sú schopné pokrývať biologické aktivity a ku mapovaniu činností v oblasti vedomia (psychiky) sú potrebné 64 a viac bitové senzory

Začínalo sa ako obyčajne v technických oblastiach, lebo senzor bol predovšetkým technické zariadenie. Podstatne neskôr pribudli senzory na meranie prírodných a hneď na to biologických kvalít a súčasný kozmický priemysel si vyžaduje silnejšie ako 64 bitové senzory.

Príspevok je venovaný fyzikálnym senzorom 16 a 32 bitovým, k použitiu ktorých sú učitelia školení a tiež ku senzorom, ktoré sa vyskytujú na informatických a robotických súťažiach. Dnes sú vo svete rozšírené tri súťaže a rad by som uviedol aj tu, že Slovensko ja na nich pravidelne zastúpené. Do našich škôl sa postupne dostáva do 15 typov senzorov, ktoré sú na viac ako 25 základných školách na Slovensku. Z uvádzaných senzorov nie sú v našich školách zastúpené iba dva, ktoré sú na trhu od 1. januára tohto roku a sú vhodné pre žiakov 1. stupňa ZŠ.

2. SENZORY OD LEGO EDUCATION

Počas ostatných viac ako 20. rokov vyučovania informatiky na slovenských školách prešiel tento predmet rozsiahlym vývojom, či už po obsahovej, alebo po organizačnej stránke a stále je vo vývoji.

Najmä vďaka záujmu učiteľov sa do našich základných škôl začiatkom deväťdesiatich rokov začali dostávať prvé osem bitové senzory. Neskôr ich vystriedali 16 bitové a dnes sa ťažiskovo orientujeme na 32 bitové senzory. Úvodom budem prezentovať základné senzory od LEGO, nie všetky vyrábané priamo v LEGO:

9846 Ultrasonic Senzor. Ultrasonický senzor. Meria presne vzdialenosť buď v cm, alebo v palcoch a je vhodný pre deti od 8 rokov.

9845 Sound Sensor. Zvukový senzor. Meria intenzitu zvuku v dB a výšku tónu v dBA, v rozsahu 3-6 kHz, kde je zvuk najrozpoznateľnejší pre človeka a senzor je vhodný pre deti od 8 rokov.

9843 Touch sensor. Dotykový senzor. Eviduje stlačenie a meria počet stlačení. Často predstavuje prsty robota. Pre deti je použiteľný od 8 rokov.

9844 Light sensor. Svetelný senzor. Obyčajne predstavuje oči robota a dobre rozpoznáva medzi odtieňmi šedej od bielej po čiernu, tak isto podľa množstva odrazu svetla mapuje aj farby, nerozlišuje ich však. S jeho uplatňovaním vznikajú občas problémy, nakoľko je nastavený na farby umelej hmoty. Možné použiť už pre deti od 8 rokov, ale lepší na 2. stupeň.

MS1038 Colour Sensor for MINDSTORMS NXT. Farebný senzor pre Mindstorms NXT. Farebný senzor rozlišuje rad svetlých a pastelových farieb. Je vhodný pre deti od 10 rokov.

MS1042 Infrared Seeker for MINDSTORMS NXT. Infračervený vyhľadávač pre Mindstorms NXT. Umožňuje robotu detekovať zdroj infračerveného svetla a určiť jeho smerovanie za použitia 5 infračervených detektorov rozložených po 60 stupňoch do kruhu. Je vhodný pre deti od 10 rokov a vyžaduje vhodné IR zariadenie.

MS1034 Compass Sensor for MINDSTORMS NXT. Kompas senzor pre Mindstorms NXT. Kompas je schopný merať magnetické polia a určiť smer svojho otočenia (obyčajne smer otočenia robota). Kompas má zabudovanú ochranu a kalibráciu umožňujúcu redukovať magnetické vplyvy iných zdrojov. Vhodný pre deti nad 10 rokov.

MS1040 Accelerometer Sensor for MINDSTORMS NXT. Senzor merania zrýchlenia a smerovania pre Mindstorms NXT. Senzor umožňuje robotu, aby evidoval svoje smerovanie hore a dole, napravo a naľavo, alebo zo strany na stranu. Tento trojosí akcelerometrický senzor tiež meria zrýchlenie a tak umožňuje meranie intenzity g, ako u pilotov, alebo astronautov.

MS1044 Gyroscopic Sensor for MINDSTORMS NXT. Gyroskopický senzor pre Mindstorms NXT. Používa sa pri stavbe modelov robotov Mindstorms NXT, ktoré sú schopné pracovať a merať gyro hodnoty ako pri otáčaní, tak aj pri kvartzovom rezonátore. Presné odčítanie hodnôt vyžaduje NXT-G blok, alebo iné programovacie prostredie umožňujúce odčítanie hodnôt zo senzora. Po inštrukciách učiteľa je vhodné pre žiakov od 10 rokov.

MS1046 Infrared Link Sensor. Spojovací infra senzor. Umožňuje zvýšenie efektívnosti komunikácie medzi Mindstorms RCX, LEGO IR riadenými vlakmi a LEGO riadenými motormi ako u nového modelu 9842 interaktívny servo motor. Teda spája doposiaľ vytvorené inteligentné LEGO modely pod jednu informačne pokrytú komunikačnú strechu.

3. UČITEĽSKÉ IMPULZY

Po druhom školení sa učitelia začali samostatne aktivizovať a hľadali ďalšie senzory, ktoré by im mohli pomôcť v školách. Oslovili sme vo svete silné spoločnosti a boli nám odporúčané senzory rozšírené vo svete. Na žiadosť učiteľov sa do škôl začali dostávať aj citlivé senzory, ktoré sa uplatňujú na hodinách fyziky, chémie a informatiky. Z týchto by som uviedol:

General Air Pressore Sensor – Senzor tlaku vzduchu. Požiteľný v rozsahu 0 - 200 kPa s presnosťou 9%. Širšie použiteľný senzor na meranie tlaku vzduchu vrátane barometrického. Štandardný pre SI v kPa (kilopascal). Aplikácie: Pochopenie zákonov o plynoch najmä o tlaku a teplote, kvasenie, rýchlosť reakcií pri určitých tlakoch plynu.

Barometric Air Pressure Sensor – Senzor (barometrického) tlaku vzduchu. Meria tlak vzduchu v rozsahu 950 – 1050 hPa s presnosťou 0.1Pa. Vhodný k mapovaniu meraniu aktuálneho tlaku vzduchu.

HumiPro Humidity Sensor – Senzor vlhkosti. Meria v rozsahu 0 – 100% relatívnej vlhkosti vzduchu s presnosťou 4% do 80% vlhkosti. Uplatnenie je v škole a doma, pre dažďom a po daždi, všestranné.

Adjustable pH Aplifier – Nastaviteľný pH zosilňovač. PH zosilňovač umožňuje veľmi presné meranie vhodné aj ku normovaniu pH sond a ku ich prepojeniu ako do LogIT, tak aj BNC systémov. Učitelia si ho vyžiadali vzhľadom k jednej úlohe v populárnej odbornej súťaži.

pH Probe – pH sonda. Vysoko kvalitná pH sonda určená pre študentov. Má robustné telo a plastovú ochranu na konci so senzorom. Uplatňuje sa pri kvasení ako pri jogurte, tak aj pri víne, monitorovaní prostredia, kyslých titráciach a reakciách medzi kyslými a zásaditými zmesami.

ProTemp – Temperature sensor (teplotný senzor). Precízne vyrobený senzor k meraniu teploty tekutiny, alebo plynu so 180 mm dlhou sondou z kvalitnej ocele. Presnosť je 1° C v teplotnom

rozsahu -10°C až 60°C. Použitie ako u bežných teplomerov : teplota tela, krivka chladenia, izolačné vlastnosti materiálov, prežitie zvierat v rozličných klimatických podmienkach, premena energií v zvláštnych prípadoch a pod.

Position Sensor – Pozičný senzor – Senzor otáčok a polohy. Pozičný a pohybový senzor je rotačný senzor s extrémne malým mŕtvym bodom – iba 20°. Meria ako uhol vychýlenia v rozsahu (0°, 340°), tak aj počet otočení okolo osi a teda aj vzdialenosti. Má dobré výkonové charakteristiky a môže byť najviac zaťažený 50 gramami. Umožňuje analýzu jednoduchých harmonických pohybov, rast rastlín, mapovanie pohybov kyvadla a podobne.

Sound Sensor - Zvukový senzor. Senzor je určený k meraniu hluku a zvuku. Má podobné vlastnosti ako ľudské ucho to jest eviduje zvuky v rozsahu od 400Hz do 4 kHz pri hlasitosti od 50 do 100 dBA s oneskorením 0,5 sekundy. Je použiteľný pri analýze absorbcie zvuku cez jednotlivé materiály, odraze zvuku, šírení hluku a podobne.

Voltage Measurement Set – Zostava k meraniu voltov. Zostava je určená k meraniu napätia v elektrických okruhoch alebo kolísania napätia pri práci so senzormi. Pracuje v rozsahu (-25 V, 25 V), s presnosťou 1% meranej hodnoty. Uplatňuje sa pri štúdiu efektívnosti solárnych batérii, elektrických motorov, indukcii, potenciálnych rozdielov prechodu cez jednotlivé komponenty elektrického okruhu a podobne.

Current Mesurement Set – Ampérmeter. Umožňuje meranie intenzity prúdu v elektrickom okruhu do 25 V. Pri maximálnom zaťažení 1 A s presnosťou 1% je použiteľný pri meraní intenzity prúdu v žiarovkách, Ohmovom zákone, tepelných efektoch a podobne

Linear Accelerometer – Lineárny akcelerometer. Lineárrny merač zrýchlenia je malý senzor k meraniu akcelerácie, vrátane zrýchlenia voľného pádu a tiež ho je možné použiť ako senzor na meranie uhlu naklonenia. Zrýchlenie meria v rozsahu d0m/s/s. Uplatňuje sa pri meraní zrýchlenia počas voľného pádu, zrýchlenia v dynamických systémoch, meranie zrýchlenia v rozličných typoch experimentov, pri meraní efektov naklonenia a pri transformácii delikátnych objektov.

LUX Light Sensor – Svetelný senzor meria intenzitu svetla v LUXoch. Senzor je určený k meraniu intenzity bežného denného svetla v LUXoch pri maximálne 565 nanometroch, presnosti 9%, veľkosti citlivej plochy 7,5 mm.. Používa sa pri experimentoch vyžadujúcich kalibráciu merania a tiež pri experimentoch so slabými lasermi a vo veľa aplikáciách sa uplatňuje ako bežný svetelný senzor.

4. PRÁCA UČITEĽOV

Je potešením, že naši učitelia majú stále dosť záujmu o prácu žiakov so zreteľom na ich podporu pri porozumení súčasnej technike. Uvádzam aktivity Dr. Ondreja Boreka, Dr. Miroslava Vojteka, Dr. Ľudovíta Mačora, Mgr. Petra Cehelského a Dr. Karola Pauchlyho, Mgr. Jozefa Mača a RNDr. Slávky Blichovej, korí sa výraznou mierou podieľali na zbere prvých skúseností na našich základných školách a spracovali nasledovné témy k jednotlivým hodinám

Fyzika: Hydrostatický tlak v vode z vodovodu a v morskej vode (7. tr). Tlak plynu v uzavretej nádobe a jeho zmeny (7. tr.). Meranie hluku počas veľkej prestávky (9.tr). Senzory napätia a prúdu (6. - 9. ročník): Elektrický prúd v kovoch, Tepelné elektrické spotrebiče, Magnetické pole elektrického prúdu, Rozvetvený elektrický obvod, Prúd v kvapalinách a v plynoch, Zákony elektrického prúdu v obvodoch, Elektromagnetické javy, Striedavý prúd, Vedenie elektrického prúdu v polovodičoch, Elektrická energia, jednoduché elektrické obvody, Redoxné (redukcia a oxidácia) reakcie a korózia kovov, Zahrievanie vodiča pri prechode elektrického prúdu, Funkcia poistky v elektrickom obvode, Elektrický zvonček, Vedenie elektrického prúdu v niektorých roztokoch. Vodiče ako ľudské telo a vlhký povrch, Elektromagnetická indukcia – vznik indukovaného prúdu, Polovodičová dióda – vysvetlenie činnosti, Tlak vzduchu na rozličných miestach v TANAPe, Závislosť hlučnosti od typu zdroja, Frekvenčná citlivosť snímača, Zmena vlhkosti ovzdušia v závislosti od počtu hodín po daždi, Vach dióda a ampérmeter a voltmeter,

Práca s počítačom: Vytvorenie programu pre práce s DCP, Redoxné reakcie a korózia kovov – zostavenie výpisov, Grafy meraní, Spätná väzba pri ovplyvňovaní elektrickej sústavy, RoboLab vo výučbe – uplatnenie,

Chémia: Uplatnenie pH sondy pri chemických pokusoch, Chémia cez VT, pH a zdravie, Vodivosť vody a iných tekutín v chémii,

Ekológia: Meranie hluku počas veľkej prestávky (9.tr), Vlhkosť v škole (8.tr), Znečistenie vzduchu pri ceste Slobody, Meranie pH u dažďovej vody, pitnej vody, vody v potoku a u pokazeného vína, Kde a kedy je čisté ovzdušie – Ako ho odmeriame?

Uplatňovanie senzorov pri rozličných aktivitách v školách sa odráža aj na záujme našich žiakov v účasti na rozličných medzinárodných súťažiach. Práca s počítačom je ťažisková v troch svetových súťažiach jedna je v EU (LEGO v pozadí) jedna v

USA a jedna v Ázii. Cieľom týchto súťaží nie je iba aktivita žiakov, ale aj záujem organizátorov poskytnúť súťažiacim vhodné podmienky,aby následne študovali v rozumnom prostredí.

5. ZÁVER

Cieľom príspevku bolo poukázať na aktivity našich vzácnych učiteľov a poznamenať, že sme v počte umiestnených na troch medzinárodných súťažiach (viac ich nie je z počtom nad desať štátov) skutočne pre túto časť sveta nadpriemerní. Bol by som rád, ak by si túto skutočnosť všimli aj nadriadené orgány.

Ďakujem za pozornosť

LITERATÚRA

[1] Educational Research, Methodology, and Measurement, J. P. Keeves, Pergamon Press 1988

[2] RoboLab Getting Started 1, Martha N. Cyr, LEGO 1999

[3] RoboLab Getting Started 2, Martha N. Cyr, LEGO 1999

[4] RoboLab Getting Started 3, Martha N. Cyr, LEGO 1999

[5] http: www.eduxe.sk

[6] http: www.eduxe.cz

[7] http: www.lego.com/education

Kontaktná adresa
Peter Cvik, EDUXE, M.C.Sklodowskej 2, 85104 Bratislava
tel: 02/62522562, fax: 02/62522563
Email: eduxe@internet.sk

INFORMATIKA A MULTIMÉDIÁ V ŠKOLE A MIMOŠKOLSKEJ ČINNOSTI.

MGR. VLADIMÍR DRÁB

Gabor, spol. s r.o., J. Gabora 1, 957 01 Bánovce nad Bebravou, e-mail: vladimir_drab(at)hotmail.com

Multimédiá ako motivačný faktor výučby informatiky v organizovanej aj spontánnej činnosti. Príspevok pojednáva o zamestnaniach s využitím multimédií. Prezentuje praktické skúsenosti s aplikáciou multimédií, kedy je intencionálny prístup ukrytý za tvorivou prácou a hrou. Tvorivé zamestnania s dôrazom na rozvoj kompetencií učiteľov informatiky a žiakov, praktické využitie v bežnom živote. Práca s atraktívnym aktuálnym obsahom je výhodou, ktorú by bolo škoda nevyužiť.

Kľúčové slová: *výučba informatiky, multimédiá, alternatívne postupy.*

1. ÚVOD

Vo svojom vstupe by som rád odprezentoval niektoré tézy zo svojej, nedávno obhájenej, diplomovej práce pod vedením PaedDr Miloslavy Súdolskej, PhD. Pedagogická prax, ktorú som absolvoval, ma priviedla k výučbe informatiky formou rôzne zameraných zamestnaní, v ktorých sa žiaci učili prostredníctvom kreatívnych úloh (učenie hrou a prácou). Vekovú rôznorodosť členov krúžkov v Centre voľného času (CVČ) som necítil ako nevýhodu, ani absenciu takých silných učiteľských zbraní ako je klasifikácia žiackeho výkonu a správania sa (okrem neformálneho slovného hodnotenia). Mojou jedinou šancou, ako sa realizovať vo výchovno – vyučovacom procese, bolo zaujať a motivovať žiakov. Táto skutočnosť ma nútila neustále vymýšľať nové zamestnania a postupy, pretože na podobnú činnosť neexistovali metodické materiály a odporúčania. Práci s deťmi a mládežou som sa, ako vychovávateľ, vedúci oddelenia techniky venoval 17 rokov.

2. PREČO MULTIMÉDIÁ?

Multimédiá sú moderným prostriedkom aj predmetom výučby a mimoškolského života. Obklopujú nielen ľudí od počítačov, ale aj laickú verejnosť a našich žiakov ako pasívnych konzumentov. Vymenovanie všetkých oblastí výskytu multimediálnych prvkov by vystačilo na niekoľko takýchto príspevkov. Pre ilustráciu uveďme, prečo sú také rozšírené a atraktívne. Sú interaktívne, názorné, pestré, pôsobivé, oslovujú viacero zmyslov, ponúkajú priestor na kreativitu, praktické a opakovateľné využitie. Ich tvorba učí žiakov rešpektovať autorské právo, pretože sa stávajú autormi, hrdými na svoje výkony. Vyučovanie cez multimédiá prekonáva technokratický prístup k informatickému vzdelávaniu, približuje a popularizuje IKT aj žiakom, ktorí by sa v našom predmete inak nerealizovali. V každodennom živote každý žiak prichádza do styku s internetom, hudbou, aplikáciami, hrami, či mobilmi prepchatými multimediálnymi funkciami a schopnosťami. Multimédiami mládež žije a nám preto nerobí problém pomocou nich si ľahko získať jej záujem. Správne motivovaný žiak je predpokladom úspešného naplnenia našich cieľov. Tým nechcem podsúvať interpretáciu, že obsah učiva by mal byť poplatný nejakej módnej vlne. Iba by som rád naznačil, ako sa osvedčili postupy výučby multimédií a cez multimédiá.

V kontexte s inými predmetmi vidím v tejto téme výhodu vďaka aktuálnosti a atraktivite obsahu. Využitím multimédií posúvame, z výchovno – vzdelávacieho hľadiska, seba a svojich percipientov z interpretačnej roviny do kreatívnej. Poskytujú okrem informatických kompetencií aj medzipredmetové presahy. Vyžaduje sa estetické cítenie a výtvarné vlohy. Na tomto mieste by som rád poukázal na svoju skúsenosť, že deti s potešením kreslia na počítači, dokonca aj tie menej nadané. Majú eidetické vlohy a fantáziu, ktorých uplatnenie tvorba počítačovej grafiky evokuje. Navyše má grafický editor pomerne jednoduchú korekciu chýb gumovaním a viackrokovú funkciu „Späť“. Okrem týchto predností multimédiá ponúkajú integráciu textu, zvukov a hudby, fotografií, animácií ale aj videotvorbu, simuláciu dejov a úkazov, tvorbu interaktívnych aplikácií a prezentácií využiteľných v iných predmetoch a oblastiach života. [1] Multimédiá sú multiplikátorom svojou podstatou, zložením aj pôsobením.

3. PREKÁŽKY SKUTOČNÉ A ZDANLIVÉ

Výučba multimédií je náročná na prípravu a čas pedagóga. Študenti ho častokrát „tromfnú“. V takejto situácii je vhodný rovnocenný vzťah so žiakmi, spolupracujeme s nimi ako s kolegami. Autoritu si dobrý pedagóg získa svojimi vedomosťami, vystupovaním, schopnosťou zaujať, prípravou zamestnaní ktoré sú pútavé, nie sú samoúčelné, majú využitie v bežnom živote. Multimédiá a ich elementy sú síce súčasťou učebných osnov, ale učitelia často prednášajú tieto a iné témy bez zreteľu na to, že pracujú s elementmi multimédií, z ktorých môžu skomponovať v spolupráci so žiakmi hodnotné diela. Niektorí z nich majú málo skúseností a zručností v multimediálnej tvorbe.

Ďalším argumentom, ktorý sa používa proti rozšíreniu multimédií je výhrada o hardvérovej náročnosti multimédií, slabej technickej vybavenosti

škôl a školských zariadení. Je pravdou, že čím novší softvér, tým väčšie nároky na systémové zdroje. Čo nám ale bráni používať staršie verzie programov, ktoré nemajú taký hlad po výkone? V skutočnosti na všetky oblasti multimediálnej tvorby existuje na školách vhodný softvér, počnúc výbavou dominantného OS Windows a kancelárskych balíkov. V špeciálnych prípadoch si môžeme pomôcť bezplatnými aplikáciami typu open-source alebo freeware a shareware aplikáciami z odborných časopisov, kde redakcie pribaľujú do elektronickej prílohy vhodný komerčný program na voľné, alebo obmedzené použitie. Námietky o chýbajúcom softvéri neobstoja.

4. ZOPÁR PRÍKLADOV

V aktuálnych učebných osnovách vieme multimédiá vyučovať ako súčasť takmer všetkých tematických celkov. Princíp je jednoduchý. Dekompozícia úlohy na jednotlivé zložky, ich tvorba, získavanie, spracovanie, úprava a prezentácia. Potom nasleduje skladanie rôznorodých multimediálnych prvkov do kompozície, pridávanie interaktivity.

Čo máme odskúšané vo výučbe multimédií v podmienkach základných a stredných škôl?

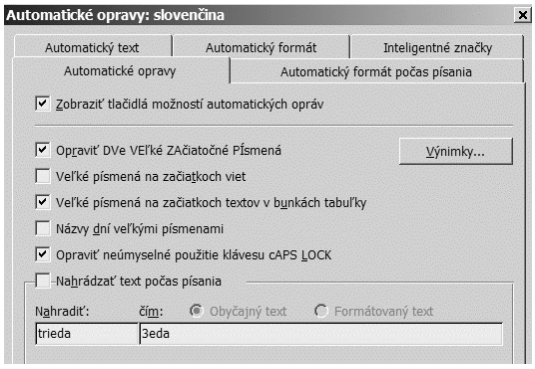
Spracovanie textu

- ASCII ART – píšeme obrázky v textovom editore; písmenkové emotikony 8-)

\/ |_ /\ |> <>, /\/\ | |_ /\ |\|, \/ | |< |,

```
        __oo
 - - - - - (___:>
        "  "
```

- pokročilé techniky spracovania textu v textovom procesore, základy DTP kombináciou textu, tabuliek, obrázkov a obtekania; tabulátory, štýly a makrá [2]; žartíky vo Worde – využitie v školskom časopise, seminárne a laboratórne práce, nástenky a oznamy → medzipredmetové vzťahy

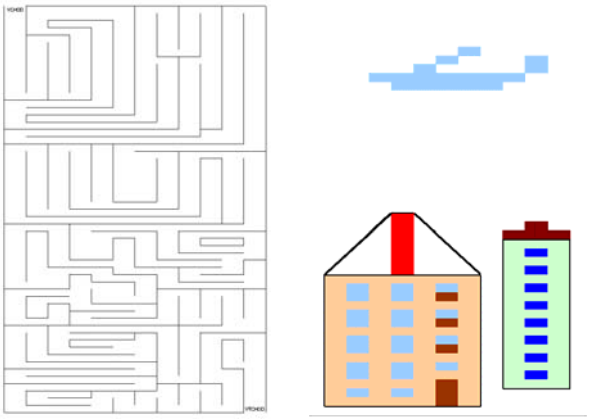


Obr. 1 – Žartovné opravy v MS Word

- Kreslenie a animácia v textovom režime Pascalu – tvorba statického obrázku, animácie z písmen a znakov, využitie cyklov a procedúr

Statická grafika

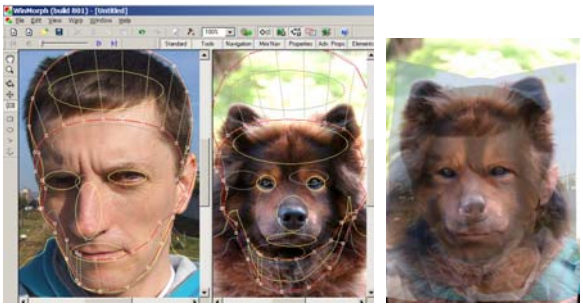
- Kreslenie v tabuľkovom kalkulátore



Obr. 2 – Bludisko a domy v MS Excel

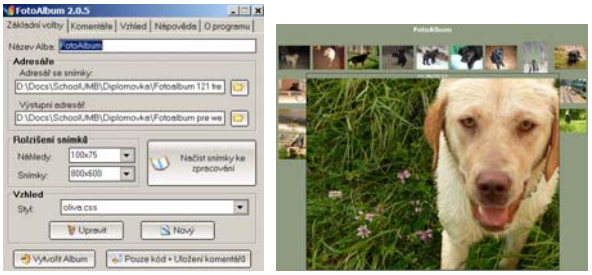
- Bitmapy – kreslenie, fotohádanky a fototriky, výrezy, morphing

Obr. 3 – Kreslené fonty



Obr. 4 - Morphing

- Kreslenie vo vyššom programovacom jazyku
- Galérie a fotoalbumy – lokálne aj internetové (moderná kronika školských podujatí, osobné, rodinné, klubové využitie)



Obr. 5 – Fotoalbum

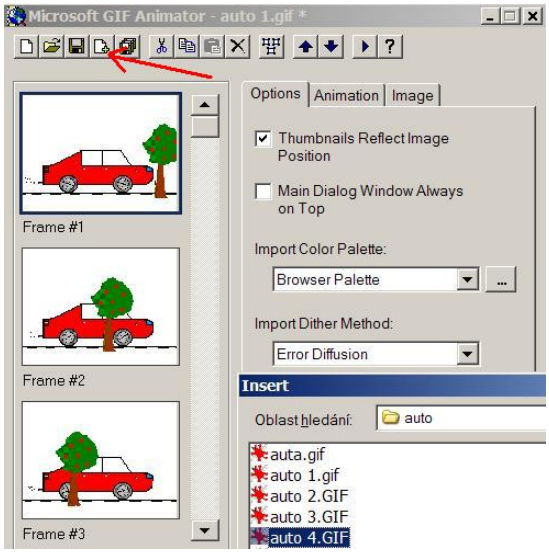
- Spracovanie a úpravy obrázkov v GIMP-e, filtre, efekty, orez

Obr. 6 – Spracovanie fotografií v GIMP

- Aplikácie na prezeranie, tvorbu, úpravu rastrovej grafiky
- Kryptovacia metóda steganografia – ukrytie posolstva do obrázku (modif. LSB)
- Vektorová grafika – InkScape – transformácie objektov, plagátová tvorba
- OCR

Animácie

- Animovanie bitmapových obrázkov v programoch MS Gif Animator a Logo Motion



Obr. 7 – Animácia

- Animácie vo vyšších programovacích jazykoch – príprava na interaktívny projekt
- Animácie v prezentačných programoch typu Powerpoint

Interaktívne multimédiá

- Programovanie hier, programov na podporu vyučovania – vo vyšších, ale aj detských programovacích jazykoch

Obr. 8 – Hra Worms v Imagine

Obr. 9 – Hra delo vytvorená v prostredí Pascal

Zvuk

- Záznam zvuku
- Audacity – spoty, zvonenia do MT, nahrávanie a archivácia starých analógových nahrávok, šumové fitre, úpravy zvukov
- CDex – vyrábame mp3
- Podcasty – rádiovysielanie po internete
- Vzorkovanie a kvantovanie zvuku
- Programovanie zvukov vo vyššom programovacom jazyku
- Prehrávanie a konverzia formátov

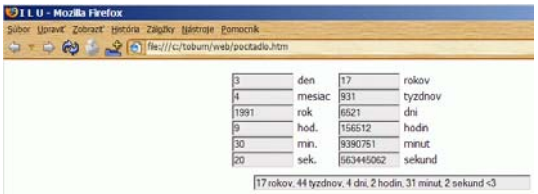
Video

- Virtuálne video – videonávody z obrazovky, záznam hry
- Film z fotografií
- Krátke videožánre – videoklipy, dokumenty, reklamné šoty, ankety
- Školská televízia AMOS
- Kontajnery a videoformáty
- Konverzie a kompresia videa

- Webkamery, IP kamery a ich využitie – videochat, videokonferencia, ostraha objektov, návšteva virtuálnych miest, galérií, múzeí, vesmírnych misií NASA

Multimédiá a internet.

- Tvorba stránok



Obr. 10 – Žiacka stránka - počítadlo

- Audio a video stream

5. ZÁVER

Prečo organizujem prácu s počítačom cez multimédiá? Žiaci sa na zamestnaniach evidentne bavia a učia sa mimovoľne. Výchova a vzdelávanie nie sú prvoplánové z pohľadu žiakov, pôsobenie je rozvrstvené a vyvážené. Tento postup pôsobí aj na problémových žiakov, pretože z vyučovania nevyčnieva mentorovanie, konečne sú rovnocenní a chcú to dokázať. Je to pre nich parketa ako vyjadriť svoje pocity a postoje cez dielo a tvorbu. Nie vždy sa s ich prejavom dá súhlasiť, ale dostali možnosť sa slobodne vyjadriť cez tvorivú činnosť a my vzácnu príležitosť ich lepšie poznať, o názoroch diskutovať. Tak sa priťahujú opačné póly magnetu, keď medzi nimi nie je prekážka alebo priepastná vzdialenosť.

Použitá literatúra:

[1] Sokolowsky, P. – Šedivá, Z., Multimédia, současnost budoucnosti. 1. vyd. Praha: Grada Publishing s.r.o., 1994. 208s. ISBN 80-7169-081-3

[2] Martínek, Z., Počítačová typografie srozumi-telně ... s Wordem. 1. vyd. České Budějovice : KOPP, 1996. 207 s. ISBN 80-85828-75-8

DISTANČNÍ KURZ INTERNET JAKO PROSTŘEDEK PRO PODPORU VÝUKY

MGR. ZBYNĚK FILIPI, MGR. TOMÁŠ PŘIBÁŇ, DOC. ING. VÁCLAV VRBÍK, CSC.

Katedra výpočetní a didaktické techniky, Fakulta pedagogická, Západočeská univerzita v Plzni, Univerzitní 8, 306 14 Plzeň, Česká republika,
tel. +420 377 636 440, e-mail: filipiz@kvd.zcu.cz, tpriban@kvd.zcu.cz, vrbik@kvd.zcu.cz

Příspěvek popisuje multimediální e-kurz Internet vytvořený na Fakultě pedagogické Západočeské univerzity v Plzni. V textu je rovněž uveden odkaz na plnou off-line verzi kurzu (bez možnosti práce s úkoly a testy). Závěrem jsou vysloveny některé obecnější závěry získané na základě zkušeností s distančním způsobem výuky.

Klíčová slova: *distanční vzdělávání, e-learning, vizualizace, internet*

1. ÚVOD

Před několika roky jsme se rozhodli vzhledem k zaměření našeho pracoviště zabývat rychle se rozvíjející oblastí vzdělávání, do níž patří distanční výuka. Nové e-learningové multimediální výukové aplikace se zdály být vhodným nástrojem, který podpoří kvalitu zmíněného způsobu výuky.

Po různých počátečních pokusech s tvorbou a nasazením vlastních e-kurzů do výuky jsme měli možnost podílet se na řešení projektu Leonardo da Vinci II BATCOS (Development & Piloting of Basic On-Line Training Courses). Partnery projektu byly relevantní organizace z České republiky, Německa, Itálie a Velké Británie. V projektu byly vyvinuty vzdělávací materiály pro 33 kurzů. Vyvinuté vzdělávací materiály jsou v angličtině, češtině, němčině a italštině. Obsah vytvořených kurzů a ukázky z CD verzí lze nalézt na adrese http://athena.zcu.cz/batcos pod odkazem DEMO. Při řešení projektu jsme nabyli řadu zkušeností, které jsme dále rozvíjeli.

V dalším textu budou uvedeny konkrétní příklady kurzu Internet, který je zařazen jako výběrový předmět ve všech formách bakalářského studia na Fakultě pedagogické Západočeské univerzity v Plzni a jako povinný v certifikátovém programu Informační výchova pro ZŠ a pro SŠ.

2. STRUKTURA A FORMÁLNÍ USPOŘÁDÁNÍ KURZU

Základ každého e-kurzu je tvořen plánem kurzu. Plán kurzu je časově orientovanou sekvencí jednotlivých na sebe navazujících *studijních kapitol,* sdružujících v sobě různé typy vzdělávacích aktivit. Každá studijní kapitola začíná stručným úvodem, ve kterém je student seznámen s anotací kapitoly, studijním cílem, časovou náročností a kritérii hodnocení. Základní vzdělávací aktivity jsou *studijní článek* (prezentace nové látky), *úkol* (jeho řešení se odesílá lektorovi kurzu k hodnocení), *cvičení* (jeho řešení se neodesílá lektorovi kurzu k hodnocení, slouží tedy pouze studentovi pro procvičení látky), *diskuse*, *test* (jeho řešení se odesílá lektorovi kurzu k hodnocení), *autotest* (jeho řešení se neodesílá lektorovi kurzu k hodnocení, slouží tedy pouze studentovi pro procvičení látky) a *anketa*.

Začlenění vhodných vizualizací k jednotlivým částem prezentované látky posiluje jejich srozumitelnost a názornost. Tímto v sobě obsahuje také výrazný motivační potenciál studia. Uvedením ilustračních vizualizací má student možnost přímo sledovat aplikaci prezentované látky na praktické úrovni. Studijní články kurzů jsou z formálního hlediska zobrazeny ve dvou svislých rámcích v okně prohlížeče (doporučen je Microsoft Internet Explorer). V pravém rámci je stručný vysvětlující text. V levém rámci se nachází vizualizace probírané látky (obrázky, animace, simulace…).

3. KURZ INTERNET

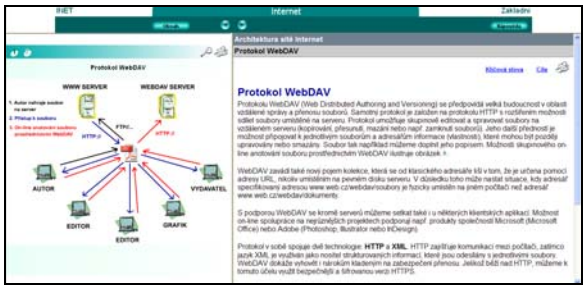
V této části je stručně uveden obsah a charakteristika kurzu. Celou off-line verzi (bez možnosti práce s úkoly a testy) je možné si prohlédnout na stránkách pracoviště autorů http://www.kvd.zcu.cz/cz/materialy/inet/index.html (heslo: inet) [1], [2].

Obsah kurzu (tučně jsou názvy kapitol, v závorkách jsou názvy studijních článků): **Úvod** (Co je to Internet, Historie sítě Internet, Internet2). **Architektura sítě Internet** (Architektura sítě Internet, Rodina protokolů TCP/IP, Protokoly HTTP a HTTPS, Protokol FTP, Protokol WebDAV, Protokoly Telnet a SSH, Protokoly elektronické pošty). **Adresování v síti Internet** (Adresy v síti Internet, IP adresa, Doménová adresa, URL adresa). **Služby Internetu** (Přehled služeb, Služby WWW a Gopher, Terminálové služby, Přenos souborů). **Možnosti připojení do sítě Internet** (Přehled způsobů připojení, Dial-up a ISDN, DSL linky a pevné připojení, Kabelové připojení, Bezdrátové připojení, Připojení z mobilního telefonu). **Prohlížeče webových stránek** (Přehled základních webových prohlížečů, Pracovní prostředí v MS Internet Explorer, Nastavení prohlížeče a základy práce, Oblíbené položky a Historie, Uložení a tisk stránek). **Off-line komunikace** (Off-line komunikace na Internetu, Elektronická pošta, E-pošta prostřednictvím poštovního klienta,

Nastavení a správa MS Outlook Express, Práce se zprávami v MS Outlook Express, E-pošta přes webové rozhraní, Nastavení a správa poštovní schránky, Práce se zprávami v poštovní schránce, Elektronické konference, Diskusní skupiny). **On-line komunikace** (On-line komunikace na Internetu, Diskusní systémy, Web chat, Videokonference, IP telefonie). **Vyhledávání na Internetu** (Jak hledat informace na Internetu, Webové portály, Katalogy, Vyhledávací stroje, Princip fulltextového vyhledávání, Metavyhledávače, FTP archivy, Vyhledávání diskusních skupin a příspěvků, Vyhledávání osob a adres e-pošty, Vyhledávání firem, Slovníky a ostatní možnosti vyhledávání). **Multimediální obsah Internetu** (Multimediální web, Obrazové formáty, Zvukové formáty, Streamované audio, Video formáty, Streamované video, Přehrávače, Animace, Elektronické publikování). **Internet a bezpečnost** (Internetová kriminalita, Počítačová infiltrace, Dialery, Spyware, Spam, Hacking a phishing, Cookies). **Elektronické obchodování** (E-komerce na Internetu, Platby na Internetu, Zabezpečení v systému e-komerce, Elektronický podpis a digitální certifikáty, Komunikace se státní správou). **Elektronické bankovnictví** (Přímé bankovnictví, Ochrana bankovních dat, Home banking, Internet banking, GSM banking – Mobil banking, Mail banking, Phone banking – Telebanking, WAP banking). **Základy tvorby webových stránek** (XHTML versus HTML, Editory, Struktura XHTML dokumentu, Formátování textu, Obrázky, Odkazy, Seznamy, Tabulky, Rozšířené možnosti tvorby stránek, Validátory, Publikace webových stránek).

4. POUŽITÉ DRUHY VIZUALIZACÍ

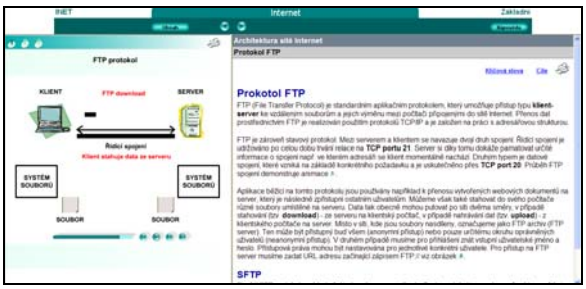
V kurzu Internet je použito několik odlišných způsobů vizualizací probírané látky [3]. Např. v článku **Protokol WebDAV** je látka ilustrována dvěma obrázky ve formátu *swf* (viz obr. 1).



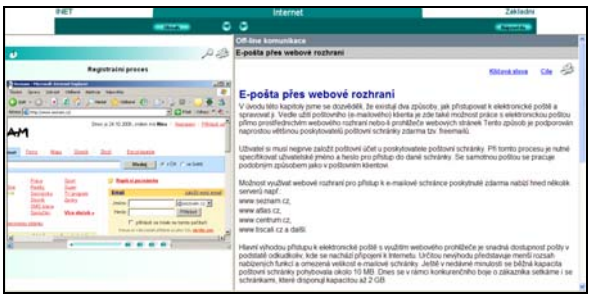
Obr. 1 Vizualizace pomocí obrázků.

Ve studijním článku **Protokol FTP** jsou použity animace vytvořené v *Macromedia Flash MX 2004*. Jednotlivé fáze komunikace mezi klientem a serverem jsou podrobně ilustrovány třemi animacemi. Animace lze zastavovat a opakovaně spouštět (viz obr. 2).

Ve studijním článku **E-pošta přes webové rozhraní** jsou použity animace vytvořené v *Macromedia Captivate 2*. Výhodou použití *Macromedia Captivate* je skutečnost, že umožňuje „nahrávání“ práce s vybraným programem anebo s celým prostředím počítače. Díky snadnému ovládání a přehlednému uspořádání celé aplikace je následná editace vytvořených animací jednoduchá a není problém vytvořit zajímavou animaci práce v určitém prostředí. Animace lze zastavovat a opakovaně spouštět (viz obr. 3).

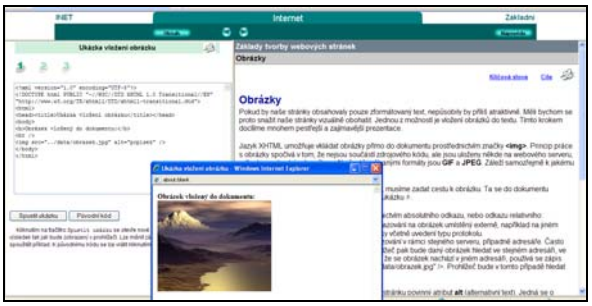


Obr. 2 Vizualizace pomocí animace v *Macromedia Flash*.



Obr. 3 Vizualizace pomocí animace v *Macromedia Captivate*.

Ve studijním článku **Obrázky** je použit simulátor XHTML kódu. Příklady uvedené v levém rámci lze spouštět a samozřejmě také modifikovat. Spouštění příkladů je naprogramováno pomocí *JavaScriptu* (viz obr. 4) a zobrazuje se v novém okně prohlížeče.



Obr. 4 Vizualizace pomocí simulátoru.

5. AUTORSKÝ SYSTÉM PROAUTHOR

Pro snazší tvorbu a správu e-kurzů je na ZČU v Plzni vyvíjen autorský systém s názvem *ProAuthor* (podrobnější informace včetně možnosti stažení demo verze autorského systému je na http://athena.zcu.cz/ProAuthor/). Tento systém je orientovaný především na podporu tvorby obsahových zdrojů on-line kurzů a off-line

multimediálních učebnic. Umožňuje generování dat autorských zdrojů ve formátu odpovídajícímu standardu SCORM. Výše uvedený kurz byl vytvořen právě s podporou tohoto nástroje.

ProAuthor pomáhá udržovat strukturu kurzu tak, aby odpovídala možnostem výukového prostředí. Kromě formátu pro distanční výukové prostředí (na ZČU je jím LMS Eden – viz obr. 5) lze samozřejmě kurz exportovat i v jiných formátech – např. E-book (elektronická kniha) – jde o formát, který obsahuje kapitoly, články, příklady, cvičení a úkoly. Je však ochuzen o testy a autotesty. Lze jej využít např. k domácímu studiu bez nutnosti připojení k Internetu.

Obr. 5 LMS Eden.

Z kurzu připraveného v *ProAuthoru* se vytváří databáze a samotný HTML kód s texty a ostatními prvky kurzu. Databáze slouží pro přípravu všech prvků kurzu, jako například testů a autotestů ve výukovém prostředí.

Základní prvky kurzu, které lze v *ProAuthoru* využít jsou *kapitola*, *článek*, *cvičení*, *úkol*, *test*, *autotest*, *anketa* a *diskuse*. Hlavním stavebním prvkem každého kurzu jsou kapitoly, které v sobě sdružují hlavně studijní články a samozřejmě i ostatní výše zmiňované prvky. Dalšími podpůrnými objekty, které lze při tvorbě kurzu v *ProAuthoru* využít, jsou takzvané *MM&S komponenty*, *anketní otázky* a *testové otázky*.

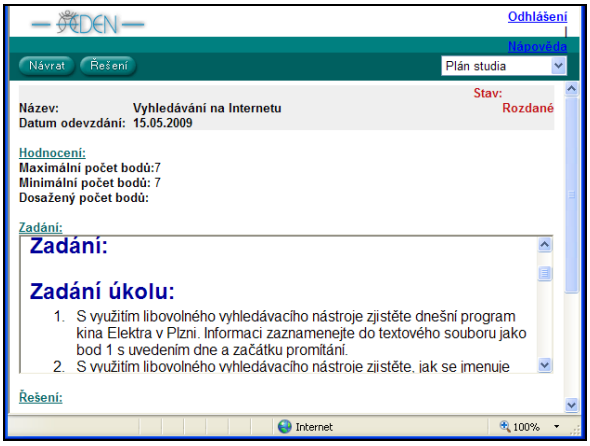
Mezi *základní MM&S komponenty* patří například obrázky, audio a video soubory, animace, ostatní podpůrné soubory a HTML soubory. Většina z nich je použita pro zobrazení v levých rámcích výsledného kurzu. Z textu kurzu na ně lze odkazovat.

Samotný kurz obsahuje úvodní slovo pro studenta, cíle kurzu, softwarové požadavky, časovou náročnost a kritéria hodnocení. Časová náročnost je údaj, který účastníka kurzu informuje o celkovém čase nutném pro absolvování kurzu. Tento údaj je dán součtem doporučených časů trvání všech studijních článků. Kritéria hodnocení obsahují minimální počet bodů, který je nutný pro splnění celého kurzu. Samozřejmě nechybí ani maximální počet bodů, který je možné získat splněním všech úkolů a testů.

Kapitola je hlavním stavebním prvkem kurzu, který má své vlastnosti. Tyto vlastnosti jsou strukturovány ve stejném duchu jako celý kurz. To znamená, že každá kapitola začíná úvodním článkem, který seznamuje studenty s krátkou anotací kapitoly, *studijním cílem*, časovou náročností a kritérii hodnocení. Na úvodní článek navazují studijní články a ostatní vzdělávací aktivity (cvičení, úkol, test, autotest, diskuze, anketa). *Studijní článek* má podobné vlastnosti jako kapitola. Opět nechybí cíle, úvodní slovo, klíčová slova a také *doporučený čas* pro jeho zvládnutí.

Cvičení je nepovinné, student jej dělá dobrovolně, což je jeho hlavní rozdíl oproti úkolu. Cvičení se jeví jako kapitola, ale obsahuje jen zadání a tipy pro řešení. Dále obsahuje návrh řešení, který vidí pouze lektor kurzu. Opět je zde udán doporučený čas.

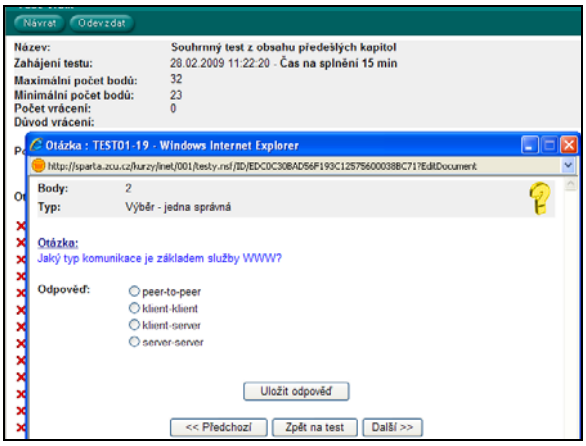
Úkol (viz obr. 6) podobně jako cvičení má své



Obr. 6 Příklad úkolu.

zadání, tipy pro řešení a správné řešení, které vidí pouze lektor kurzu. Navíc jsou zde udána *hodnotící kritéria*. Jeho *splnění je povinné*. Opět nechybí údaj o doporučeném času řešení.

Test (viz obr. 7) je složen z otázek, které jsou



Obr. 7 Příklad testu.

k danému testu přiřazeny ze seznamu otázek. Opět je zde možnost definovat doporučený čas, případně časový limit. Důležitý je samozřejmě též počet bodů.

Každá otázka má váhu jednoho bodu. Definujeme tedy minimální počet bodů ke splnění testu.

Autotest je obdoba testu, avšak student není povinen tento test splnit. Otázky se opět vybírají ze seznamu otázek. Ostatní volby jsou obdobné jako u testu.

6. NĚKTERÉ ZKUŠENOSTI A ZOBECNĚNÍ

Na základě našich dosavadních zkušeností (i z aplikace uvedeného e-kurzu Internet) lze vyslovit následující obecnější závěry. Klasická (kontaktní) forma výuky je charakterizována permanentní osobní komunikací mezi učitelem a studenty. U distančního studia plní důležitou roli především multimediální forma prezentace. Stručný text je doplněn vhodnými vizualizacemi, které napomáhají lepšímu pochopení probíraného tématu. Děje se tak díky zapojení různých smyslů a možnostem aktivního zapojení studenta. Při využívání e-learningu je třeba správně chápat roli technické i pedagogické složky. Volba vhodného autorského systému a prostředí LMS nesmí zastínit potřebu tvorby dostatečného počtu didaktických aktivizujících prvků v e-kurzu. Obsah e-kurzu je též třeba rozdělit do dostatečného počtu modulů, aby bylo snažší provádět jeho nutnou aktualizaci. Celý vzdělávací proces je při distančním pojetí založen na přístupu k virtuálnímu vzdělávacímu systému a fyzickém oddělení učitele a studentů.

Důležitou vlastností distanční výuky je její značná individualizace vedoucí k menší vazbě studenta a učitele. Studenti ke svému samostatnému získávání znalostí využívají připravené multimediální studijní materiály a s učitelem i mezi sebou navzájem komunikují prostřednictvím virtuálního studijního prostředí.

Distanční forma studia zásadním způsobem změnila úroveň participace učitele ve vzdělávacím procesu, a to tím způsobem, že do určité míry minimalizovala jeho přítomnost v tomto procesu. Jeho úkolem je zejména vystupovat jako metodický zprostředkovatel a také komunikovat se studenty, zodpovídat jejich dotazy, ať už přímo při prezenčních setkáních, nebo nepřímým způsobem prostřednictvím virtuální učebny. V neposlední řadě také musí v průběhu studia prověřovat a hodnotit znalosti studentů a poskytovat odbornou pomoc při řešení studijních obtíží.

Výhodou distančního vzdělávání je, že klade velmi malé nároky na studijní prostory (učebny) a s tím související provozní náklady (světlo, teplo). Předností pro studenta (pro některé studenty je však toto diskutabilní) je individualizace studia na základě jeho potřeb a schopností. Každý student má tak při dostatečné modularitě e-kurzů možnost efektivně si rozvrhnout tempo studia a přizpůsobit si jej svým časovým možnostem.

Nemůžeme však opomenout relativně vysokou míru izolovanosti studenta v průběhu studia, která může vést k nižšímu stupni sebemotivace. Učitel nemusí být vždy automaticky přístupný on-line a jeho kontaktování může vyžadovat určitou míru trpělivosti, a to zejména v situacích, kdy studující může mít potřebu konzultovat dotaz zásadního charakteru.

Někteří studenti výrazně preferují osobní kontakt s učitelem a v distančním způsobu vzdělávání založeném na občasném a vzdáleném kontaktu se mohou potýkat s určitou mírou ztráty motivace, která se u nich projevuje nižší zodpovědností za dosažené studijní výsledky a špatným rozvržením času při studiu. Řešením tohoto problému se jeví být blended learning, což se na našem pracovišti odráží v tom, že předměty využívající e-kurzy začínají vždy úvodním společným seminářem či přednáškou. Studenti mají v průběhu semestru k vyučujícím přímý přístup i v konzultačních hodinách.

7. ZÁVĚR

Zavedení distančních e-kurzů do výuky na Fakultě pedagogické Západočeské univerzity v Plzni změnilo podmínky pro orientovanou a samostatnou práci studentů v rámci výrazných změn odborného kurikula i organizace studia s uplatněním nejprogresivnějších profesionálních nástrojů informačních a komunikačních technologií. Jejich aplikace přispěla k zefektivnění výuky (ekonomické hledisko) při současném zvýšení kvality výuky (individualizace a názornost). Rovněž se opět potvrdila vhodnost využití jednoho ze základních didaktických principů používaných v kurzech vytvářených na pracovišti autorů a to maximální použití vizualizací probírané látky. Velmi se osvědčily interaktivní vizualizace (simulátor XHTML kódu).

LITERATURA

[1] KOZMÉR, T. Distanční výuka předmětu Internet. Plzeň, 2007. 64 s. , 1 CD-ROM. Fakulta pedagogická. Západočeská univerzita v Plzni. Katedra výpočetní a didaktické techniky. Vedoucí diplomové práce Doc. Ing. Václav Vrbík, CSc.

[2] VRBÍK, V., KOZMÉR, T. Internet : Multimediální kurz pro distanční studium [online]. 2007 [cit. 2009-01-05]. Dostupný z WWW: <http://sparta.zcu.cz>.

[3] VRBÍK, V. Internet a tvorba jednoduchých WWW stránek. Plzeň : Západočeská univerzita v Plzni, 2001. 80 s. ISBN 80-7082-724-6.

EDITOR LAMBDA – NÁSTROJ NA SPRÍSTUPNENIE MATEMATIKY NEVIDIACIM A JEHO ROZŠIROVANIE NA SLOVENSKU

ING. MARTIN HORŇANSKÝ, MGR. PETER LECKÝ

Podporné centrum pre zrakovo postihnutých študentov
Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina, 842 48 Bratislava 4
tel. 02/602 95 515, e-mail: martin.hornansky@gmail.com, peter.lecky@fmph.uniba.sk

PAEDDR. IVETA KOHANOVÁ, PHD.

Katedra algebry, geometrie a didaktiky matematiky
Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina, 842 48 Bratislava 4
tel. 02/602 95 490, e-mail: kohanova@fmph.uniba.sk

Abstrakt: *Príspevok podáva stručnú správu o práci nevidiacich žiakov na Slovensku s matematikou, charakteristikách matematického editora pre nevidiacich Lambda, jeho lokalizácii do slovenského jazyka a o kurzoch organizovaných v Podpornom centre pre zrakovo postihnutýh študentov s cieľom oboznámiť účastníkov (nevidiacich žiakov, učiteľov nevidiacich žiakov) s týmto editorom a zefektívniť ich prácu s matematikou.*

Kľúčové slová: *Prístupnosť matematiky pre nevidiacich, Editor Lambda*

1. ÚVOD

Na jar 2006 vznikla v *Podpornom centre pre zrakovo postihnutých študentov na Katedre aplikovanej informatiky, Fakulty matematiky, fyziky a informatiky UK v Bratislave* (ďalej iba v *Podpornom centre*), ako odozva na potrebu matematického editora pre nevidiacich študentov integrovaných na bežných základných a stredných školách po celom Slovensku, iniciatíva na zorganizovanie kurzu zameraného na vtedy ešte len vyvíjajúci sa špeciálny matematický editor s názvom *Lambda.* Počas posledných troch rokov sa editor *Lambda* vyvinul do plnohodnotného nástroja poskytujúceho pomoc nevidiacim pri práci s matematikou a zvyšujúceho efektivitu takejto práce. V *Podpornom centre* sme ho za tento čas lokalizovali do slovenčiny, pričom sme testovali jeho funkčnosť na jednom dlhšom a piatich krátkych intenzívnych kurzoch pre nevidiacich študentov. Finálnu verziu editora *Lambda* sme vo februári 2008 predstavili na seminári učiteľom matematiky, ktorí majú v triedach nevidiacich študentov.

2. SÚČASNÁ SITUÁCIA VZDELÁVANIA NEVIDIACICH NA SLOVENSKU

Väčšina slovenských nevidiacich žiakov vo veku povinnej školskej dochádzky navštevuje špeciálne základné školy. Na hodinách matematiky používajú knihy v Braillovom písme s grafickými ilustráciami v reliéfnej forme. Triedy v takýchto školách sú málopočetné a vyučujú tu spravidla špeciálni pedagógovia. Pri písomnej práci môžu žiaci používať mechanické písacie stroje pre nevidiacich, akými sú napríklad *Pichtove stroje* (viď obr. 1), ktoré im umožňujú vytvárať texty v bodovom písme. Špecifikom mechanických písacích strojov a bodového písma je, že pre matematiku existuje špeciálny zápis, ktorý vyžaduje oveľa viac priestoru ako výpočty zapísané bežným spôsobom, a laikovi sa môže zdať tento spôsob práce pomalý a neprehľadný. Skúsení nevidiaci žiaci sa však vedia v texte orientovať bez problémov, zápis si môžu kontrolovať hmatom a rýchlosť ich práce je individuálna. Časť žiakov si tiež pri výpočtoch pomáha počítaním z pamäti.

Obr. 1 *Pichtov písací stroj*

Po absolvovaní základnej školy môžu nevidiaci žiaci pokračovať v štúdiu na špeciálnej strednej škole (pre zrakovo postihnutých existuje na Slovensku len jedna takáto škola - Stredné odborné učilište internátne pre zrakovo postihnutú mládež v Levoči) alebo na bežných stredných školách. V súčasnosti stále viac nevidiacich začína študovať integrovane na bežných základných a stredných školách. Učitelia bežných škôl však spravidla nie sú špeciálnymi pedagógmi a bez intenzívnej spolupráce so špecializovanými pracoviskami sa len ťažko zaobídu. Používanie *Pichtových písacích strojov* kvôli ich hlučnej prevádzke, a teda rušeniu ostatných žiakov v triedach (ktoré sú omnoho početnejšie ako

na špeciálnych školách), je problematické. Napriek tomu ho časť integrovaných nevidiacich žiakov pri práci s matematickými textami využíva - ak nie priamo počas hodiny, tak na individuálnom vyučovaní, resp. na konzultáciách s učiteľom po vyučovaní.

3. POTREBA MATEMATICKÉHO EDITORA PRE NEVIDIACICH NA SLOVENSKU

Nielen v zahraničí sú mechanické písacie stroje čoraz častejšie nahrádzané elektronickým vybavením. Najčastejšie sa ako náhrada používa prenosný počítač s čítačom obrazovky. Čítač obrazovky je softvér, ktorý umožňuje nevidiacemu používateľovi pracovať s počítačom. Ide o program, ktorý používateľovi sprostredkováva obsah obrazovky prostredníctvom hlasového výstupu alebo braillovského riadka (špeciálne zariadenie zobrazujúce jeden riadok obrazovky v Braillovom písme). Takéto vybavenie umožňuje nevidiacemu používateľovi relatívne komfortnú a efektívnu prácu s bežným formátovaným textom, na prácu s komplexnými matematickými výrazmi však z nasledujúcich dôvodov nepostačuje:

- Nevidiaci používateľ môže v dôsledku obmedzení súčasných čítačov obrazovky pracovať len s textovou informáciou. Grafické informácie (napríklad netextové matematické symboly) sú používateľom čítačov obrazovky z technických dôvodov neprístupné. Tento fakt je možné kompenzovať používaním špeciálneho zápisu, ktorý umožňuje zapisovať aj grafické informácie sémanticky ekvivalentnými textovými symbolmi. Špeciálnym zápisom však väčšina vidiacich kolegov (učiteľov, spolužiakov,...) nerozumie, a preto sa tradičné techniky zápisov matematických symbolov nahrádzajú technikami podporovanými informačnými technológiami.

- Nevidiaci používatelia strácajú možnosť efektívne pracovať s dvojrozmernou informáciou (bežne zapisované zlomky, indexy,...). Čítač obrazovky dokáže používateľovi efektívne sprostredkovať najviac jednorozmernú informáciu (na braillovskom riadku možno "vidieť" celý riadok naraz), s hlasovým výstupom má dokonca používateľ na dáta len "bodový" pohľad (nevidiaci používateľ vie len o tom, čo hlasový výstup práve hovorí alebo dávnejšie povedal). To teda znamená, že pri práci s matematikou je strata možnosti pracovať s dvojrozmernou informáciou jedným z naváznejších problémov a je ju potrebné určitým spôsobom kompenzovať. Jednou možnou takouto kompenzáciou sú takzvané lineárne kódy. Tie nahrádzajú všetky dvojrozmerné štruktúry jednorozmernými alternatívami, napríklad zlomky sa zapisujú na jeden riadok a pod. Takéto zápisy nevidiacim vyhovujú a používajú ich aj pri písaní na mechanických písacích strojoch, no pre vidiacich kolegov sú opäť pomerne ťažko čitateľné. Je dôležité si uvedomiť, že mechanický písací stroj umožňuje nevidiacemu používateľovi pracovať aj s dvojrozmernými štruktúrami. Ich zápisy však zaberajú na papieri priveľa miesta, čo je hlavným dôvodom, prečo sa aj pri práci s mechanickými písacími strojmi dáva prednosť lineárnym kódom.

- Zložitejšie matematické výrazy zapísané v lineárnej forme (napríklad zložené zlomky s exponentom v čitateli aj menovateli) sú (hlavne v prípade, že svojou dĺžkou presahujú jeden riadok) veľmi neprehľadné. Toto už môže vyriešiť len špeciálny editor, ktorý poskytne funkcie na prezeranie komplexných štruktúr (napr. skryje práve nepodstatné časti výrazu, a pod.).

Z vyššie uvedených skutočností vyplýva, že ak chceme uvažovať o nahradení mechanických písacích strojov modernou technikou, musíme zaviesť do praxe používanie vhodného nástroja, ktorý poskytne nasledujúce funkcie:

- Umožní jednoduché vkladanie aj takých matematických symbolov, ktoré sa nenachádzajú na bežnej počítačovej klávesnici.

- Poskytne dostatočne prehľadný výstup prostredníctvom hlasového výstupu resp. braillovského riadka (tu je dôležité uvedomiť si, že forma dát sprostredkovaných používatelovi prostredníctvom hlasového výstupu sa líši od formy sprostredkovanej braillovským riadkom), ale aj štandardného výstupu čitateľného pre vidiacich kolegov.

- Poskytne nevidiacemu používateľovi funkcie, ktoré mu umožnia rýchlo si prezerať štruktúru komplexných výrazov, ale aj efektívne s takýmito výrazmi pracovať.

- V súčasnosti je možné dostať sa k čoraz väčšiemu množstvu matematických textov aj v elektronickej forme. Primeranou požiadavkou na opisovaný matematický nástroj je teda aj schopnosť pracovať s najčastejšie používanými elektronickými formátmi, ktoré umožňujú reprezentovať matematické texty (*MathML*, *LaTeX*,...).

Hlavne kvôli vyššie menovaným dôvodom vznikla na Slovensku potreba nájsť taký matematický editor, ktorý by umožňoval nevidiacemu používateľovi a jeho vidiacemu učiteľovi, resp. spolužiakom, jednoduché vkladanie, čítanie a orientáciu v zložitejších matematických výrazoch. Takým editorom sa nám v *Podpornom centre* javil práve editor *Lambda*.

4. MATEMATICKÝ EDITOR LAMBDA

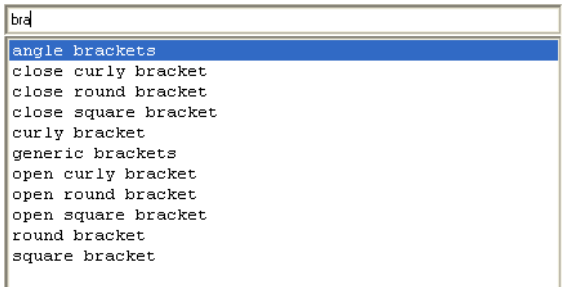
Niet pochýb o tom, že informačné technológie môžu byť pri práci s matematikou pre nevidiacich študentov veľmi užitočnými. Veľmi dôležitou požiadavkou na takéto technológie je, aby umožňovali nevidiacim študentom pracovať

s matematikou porovnateľne rýchlo a efektívne ako s ňou dokážu pracovať ich vidiaci spolužiaci.

Editor *Lambda* (**L**inear **A**ccess to **M**athematics for **B**raille **D**evices and **A**udio-synthesis) bol vytvorený a testovaný v rámci medzinárodného projektu, na ktorom sa zúčastnilo spolu 14 partnerov zo siedmich európskych krajín v rokoch 2002 až 2005. Zámerom projektu bolo vytvoriť integrovaný systém na vyriešenie vyššie menovaných problémov nevidiacich pri práci s matematikou, t.j. vyvinúť editor, ktorý by umožňoval zápis matematických výrazov lineárnym spôsobom a ich aktívnu úpravu (vkladanie, čítanie, počítanie, orientáciu...) pomocou čítača obrazovky a braillovského riadka, resp. hlasového výstupu.

Medzi hlavné charakteristiky editora *Lambda* patria:

1. Viaceré alternatívy vkladať znaky a matematické symboly (Bernareggi, 2006):
   - Vkladanie pomocou menu
   Jednotlivé symboly sú zoskupené v rámci skupín, ktoré sú prístupné cez rozbaľovacie menu, a teda používateľ môže nájsť požadovaný symbol prehľadaním položiek menu pomocou klávesnice alebo myši (v prípade vidiaceho používateľa). Nie je to veľmi rýchla stratégia vkladania symbolov, ale je veľmi jednoduchá pre začínajúceho používateľa.
   - Vkladanie symbolov pomocou dynamického zoznamu
   Všetky symboly sú abecedne zoradené. Stačí vložiť iba začiatočné písmená a zoznam sa redukuje iba na relevantné položky, v rámci ktorých sa už používateľ ľahko orientuje pomocou klávesnice a braillovského riadka, resp. hlasového výstupu.



Obr. 2 *Dynamický zoznam* umožňujúci vkladanie symbolov

   - Vkladanie pomocou klávesových skratiek
   Táto možnosť umožňuje používateľovi vkladať desiatky symbolov. Množina klávesových skratiek môže byť používateľom rozširovaná a upravovaná.
   - Vkladanie symbolov závislých od kontextu
   Za účelom dodržania štruktúry výrazu oddeľovač, resp. ukončovacia značka akéhokoľvek konštruktu môžu byť automaticky vložené pomocou dvoch klávesových skratiek.

2. Môžnosť rýchleho pochopenia globálnej informácie o štruktúre výrazu, ktorá je v mnohých situáciach dôležitá, aby bolo možné zvoliť najvhodnejšiu metódu na riešenie daného výrazu. Napríklad pri nasledujúcom výraze:

$$\frac{x^2-4}{x^2+4x+4}+\frac{4}{x+2}$$

Jeho lineárna reprezentácia v editore *Lambda* je:

//x^2-4/x^2+4x+4\\+//4/x+2\\

Je evidentné, že pracujúc v lineárnej reprezentácii môže byť ťažšie nájsť špecifické časti výrazu a rýchlo porozumieť vzťahom medzi jeho štruktúrami, čo je bezprostredná operácia vidiacich ľudí, ktorí výraz vnímajú priamo globálne a dvojrozmerne. *Lambda* uľahčuje orientáciu vo výrazoch poskytnutím možnosti premiestnenia sa na nasledujúci, resp. predchádzajúci korešpondujúci symbol, akými sú: čitateľ, menovateľ, oddeľovač, ukončovacia značka a pod. Ďalšou možnosťou na zlepšenie orientácie v príklade je viacúrovňové zobrazenie štruktúry príkladu, ktoré dokáže skryť v danom momente nepodstatné časti výrazu. Ukážka najkomprimovanejšieho zobrazenia štruktúry vyššie uvedeného výrazu:

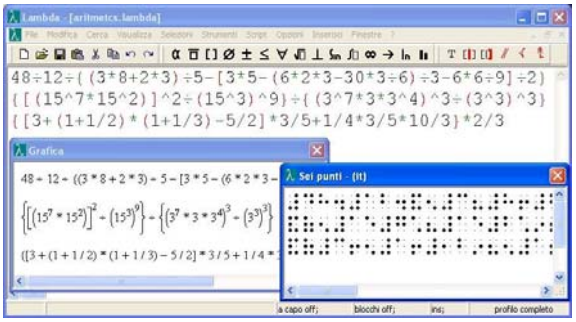
//∅\\+//∅\\

O úroveň miernejšia komprimácia štruktúry by sa používateľovi zobrazila nasledovne:

//( )∅( )\\+//4/x+2\\

Tento spôsob zobrazenia výrazu, ktorý nahrádza obsah skrytých blokov medzerami, je užitočný v prípade, že používateľ chce získať informáciu o veľkosti skrytých blokov.

3. Ďalším užitočným nástrojom editora *Lambda* je vedecká kalkulačka, ktorá používateľovi umožní počítať numerické výrazy a priamo vkladať ich výsledky do dokumentu.
4. Hoci je lineárny zápis, využívaný v editore *Lambda,* relatívne intuitívny, učitelia, rodičia a vidiaci spolužiaci môžu pri spolupráci s nevidiacim oceniť aj tzv. *Grafické zobrazenie* výrazov, ktoré zobrazí lineárne zapísaný výraz pre vidiaceho používateľa štandardným dvojrozmerným spôsobom.



Obr. 3 Okno editora *Lambda* s grafickým a braillovským zobrazením počítaného výrazu

Bližšie informácie o *Lambda Projekte* v rámci ktorého bol editor *Lambda* vyvíjaný spolu s časovo obmedzenou verziou editora prístupnou na stiahnutie môže nájsť čitateľ na oficiálnej stránke *Lambda Project* [3].

Najväčšou prekážkou v počiatočnom používaní editora *Lambda* na Slovensku bola absencia slovenskej jazykovej verzie, tak sme sa po viacerých diskusiách v *Podpornom centre* rozhodli, že sa pokúsime editor *Lambda* do slovenského jazyka lokalizovať.

Pred začiatkom práce získalo *Podporné centrum* od tímu vývojárov z Talianska súhlas s lokalizáciou. Veľmi dôležité bolo postupne posilňovať spoluprácu s týmto tímom, napríklad pretože nie vždy bolo jasné, ktoré časti zdrojového kódu boli zmenené pri tom, ako editor *Lambda* postupne získaval na funkcionalite.

5. KURZY S EDITOROM LAMBDA

Počas uplynulých troch rokov sme v *Podpornom centre* zorganizovali jeden dlohodobý a päť intenzívnych kurzov pre nevidiacich žiakov základných a stredných škôl s hlavným cieľom oboznámiť ich s editorom *Lambda* a demonštrovať, ako by im mohol zefektívniť prácu s matematikou.

Hoci kurzy mali rovnaký hlavný cieľ, ich krátkodobé ciele sa vyvíjali priebežne. Počiatočné kurzy mali za úlohu otestovať prototyp tohto editora v slovenskom prostredí, neskôr sme na kurzoch testovali aj jeho vyvíjajúcu sa slovenskú verziu a posledné kurzy mali za úlohu okrem zdokonalenia už nadobudnutých zručností študentov tiež precvičenie si práce v editore *Lambda* s využitím braillovských riadkov. V medziobdobí sme komunikovali s jeho používateľmi na Slovensku prostredníctvom *Mailing-listu.*

Okrem práce s nevidiacimi žiakmi sme pokladali za potrebné rozvinúť spoluprácu aj s ich učiteľmi. Prax ukazuje, že pedagógovia bežných škôl sa necítia byť na prácu s nevidiacimi žiakmi dostatočne pripravení a záujem učiť sa pracovať s pomôckami svojich nevidiacich žiakov je z rôznych dôvodov nízky. Preto sme sa rozhodli zorganizovať v *Podpornom centre* dva semináre určené pre učiteľov (prioritne učitelia bežných škôl, ktorí majú v triedach nevidiacich žiakov). Prvý seminár s podrobnou prezentáciou editora *Lambda* sa konal vo februári 2008, druhý, zameraný na metodiku sprístupňovania vybraných častí stredoškolskej matematiky nevidiacim, v septembri 2008.

Keďže na Slovensku je komunita pedagógov - matematikov pracujúcich s nevidiacimi žiakmi počtom relatívne malá, no geograficky značne roztrúsená, považovali sme za užitočné umožniť účastníkom seminárov vymeniť si ich základné údaje a zahrnúť ich do *Mailing-listu*, ktorý ponúka tak žiakovi, ako aj jeho učiteľovi okrem možnosti obrátiť sa v prípade problémov na iných používateľov, tiež možnosť kontaktovať odborníkov a nepriamo sa tak tiež čiastočne zúčastňovali na úpravách editora *Lambda*.

6. ZÁVER

Môžeme konštatovať, že vyššie popísané kurzy a semináre organizované v *Podpornom centre* splnili svoj účel a majú dôležitý podiel na úspešnom vytvorení slovenskej verzie editora *Lambda*, ktorý predstavuje významný prínos smerom k uľahčeniu práce s matematikou nevidiacim používateľom na Slovensku, a teda má potenciál dopomôcť integrácii nevidiacich žiakov na bežných základných a stredných školách.

V súčasnosti v *Podpornom centre* pracujeme na metodickej príručke k používaniu editora *Lambda*, ktorá by mohla byť vhodnou pomôckou učiteľa matematiky nevidiaceho žiaka používajúceho tento editor. *Podporné* centrum tiež plánuje tiež pokračovať v tradícii organizovania kurzov tak pre nevidiacich žiakov, ako i pre ich učiteľov.

LITERATÚRA

[1] BERNAREGGI C.: A framework for vocal and tactile access to science in an educational environment, Doctoral thesis, Facoltà di scienze matematiche, fisiche e naturali, Università degli studi di Milano, Facoltà di scienze matematiche, fisiche e naturali, Milano, 2006

[2] HORŇANSKÝ, M., LECKÝ, P.: First experiences with using LAMBDA editor in Slovak environment. In *Making Science Accessible: emerging opportunities for blind and partially sighted*, Miláno, Taliansko, 20. 10. 2008. Dostupné na: < http://river-valley.tv/conferences/math_for_blind_2008/> (8. 1. 2009)

[3] KOHANOVÁ I.: The ways of teaching mathematics to visually impaired students, International Congress on Mathematical Education, Monterrey, Mexico, 6. – 13. 7. 2008.

[4] LAMBDA PROJECT: Oficiálna stránka projektu zaoberajúceho sa vývojom editora *Lambda*. Dostupné na: <http://www.lambdaproject.org/> (8. 1. 2009)

[5] MIKOLAJOVÁ, K., HORŇANSKÝ, M.: Matematika s Lambdou. Študentská vedecká konferencia, Fakulta matematiky, fyziky a informatiky UK, Bratislava, 30. 4. 2007. Dostupné na: <http://www.fmph.uniba.sk/fileadmin/user_upload/editors/studium/svk/2007/DID/mikolajova.pdf> (8. 1. 2009)

IMPLEMENTÁCIA PROGRAMÁTORSKÉHO ROZHRANIA PRE TVORBU INTERAKTÍVNYCH VZDELÁVACÍCH MATERIÁLOV

ING. MGR. ROMAN HORVÁTH

Stredisko pre celoživotné vzdelávanie, Pedagogická fakulta, Trnavská univerzita v Trnave, Priemyselná 4, 918 43 Trnava, tel. +421 33 5514 618, e-mail: rhorvath@truni.sk

Článok sa zaoberá opisom programátorského rozhrania pre tvorbu interaktívnych vzdelávacích materiálov vytváraného na Pedagogickej fakulte Trnavskej univerzity. Rozhranie bude pilotne použité pri tvorbe materiálu pre vybraný predmet odboru Informatika, konkrétne predmetu s tematikou počítačových sietí. V rámci pilotného použitia bolo vypísaných niekoľko tém pre bakalárske práce, počas vypracovania ktorých môže byť rozhranie lepšie otestované a operatívne rozširované.

Kľúčové slová: *interaktívny vzdelávací materiál, vyučovanie informatiky, elektronické vzdelávanie, podpora vyučovania informatiky, IKT prostriedky*

1. ÚVOD

S prípravou elektronických vzdelávacích materiálov (inde označovaných pojmom „kurz“, v rámci tohto článku označených skratkou EVM, v súlade so zaužívanou terminológiou na TvU), sú na Pedagogickej fakulte dlhoročné skúsenosti. Príkladom sú materiály pripravené kolegami autora článku [1], [2] a aj autorom samotným, napr. [3]. Skúsenosti získané pri príprave a používaní EVM [4], [5], [6], [7], sú prínosom pri tvorbe ďalších vzdelávacích materiálov, ktoré sú vďaka tomu lepšie koncipované.

Prirodzeným smerovaním rozvíjajúceho sa systému je, okrem iného, hľadanie alternatívnych spôsobov implementácie vzdelávacích materiálov. Opis jednej z aktivít tohto druhu opisuje tento článok. Ide o tvorbu programátorského rozhrania (Programming Interface, v zmysle sústavy „knižníc“, ak vôbec pri použití JavaScriptu/JScriptu môžeme hovoriť o knižniciach) v jazyku JavaScript/JScript. Aktivita zahŕňa, v rámci testovania vyvíjaného rozhrania, prípravu materiálu pre predmet Správa IKT prostriedkov na školách (SIKT). Tento predmet má jeden zo vzdelávacích cieľov prípravu budúcich učiteľov na správu lokálnej počítačovej siete v mieste svojho pôsobiska.

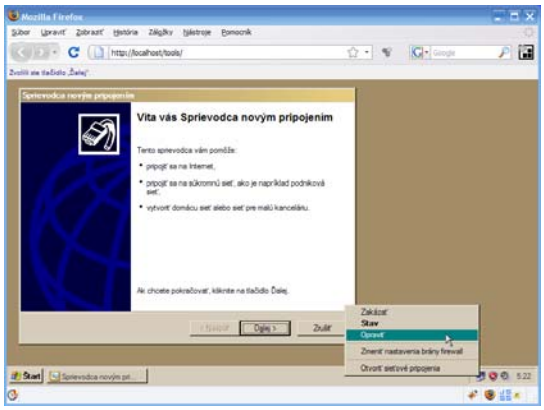
Splnenie spomenutého vzdelávacieho cieľa predmetu SIKT by bolo v podmienkach Pedagogickej fakulty jednoduchšie, ak bola k dispozícii ďalšia počítačová miestnosť, ktorá by mohla byť využitá ako „dielňa“ pre rôzne študentské konfiguračné pokusy. To nie je v možnostiach fakulty a súčasné obmedzenia zavedené v súlade so sieťovou politikou univerzity, nedovoľujú študentom vykonávať akékoľvek konfiguračné nastavenia a už vôbec nie nastavenia týkajúce sa lokálnej počítačovej siete. Preto je vhodné hľadať, v záujme zlepšenia kvality výučby, alternatívne riešenia. Jednou z možností je inštalácia virtuálnych počítačov, ktorú realizoval kolega. Toto riešenie sa ukázalo byť relatívne náročné na výkon aktuálne používaných staníc v učebni, preto nasledovalo hľadanie ďalších riešení.

Pre potreby výučby spomínaného predmetu (SIKT), je ideálnym riešením výroba interaktívneho EVM, ktorý by pokrýval vopred vyselektované oblasti z jeho obsahovej náplne. EVM je v súčasnosti v štádiu vývoja a je jedným z nástrojov testovania opisovaného rozhrania. S používaním interaktívnych EVM vo vyučovacom procese sú na fakulte isté skúsenosti, porovnaním výsledkov študentov používajúcich interaktívny EVM sa zaoberá napríklad publikácia [8]. Využitím interaktivity sa zaoberá EVM opísaný v [3].

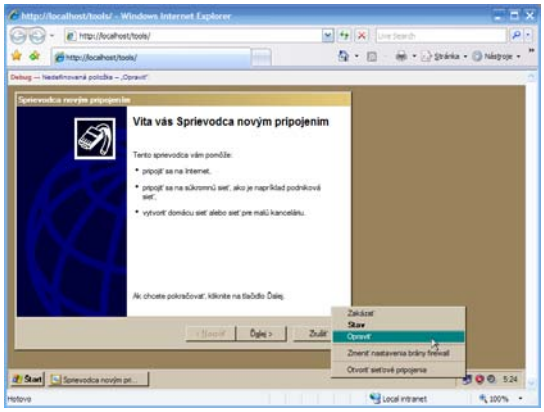
2. OPIS ROZHRANIA

Nadväzujúc na skúsenosti získané pri tvorbe materiálu [3], bola zahájená implementácia programátorského rozhrania pre tvorbu interaktívnej simulácie GUI OS Windows. Podobne ako pri [3], bol zvolený jazyk JavaScript/JScript. Pre účely vývoja bolo vytvorené malé testovacie prostredie s obmedzenými možnosťami. Testovacie prostredie rozhrania je možné vidieť na obrázkoch obr. 1 až obr. 3 (viac o testovaní a o testovaných prehliadačoch v ďalšej kapitole Problémy pri implementácii). Na obr. 3 je možné spozorovať jeden neskôr diskutovaný problém.

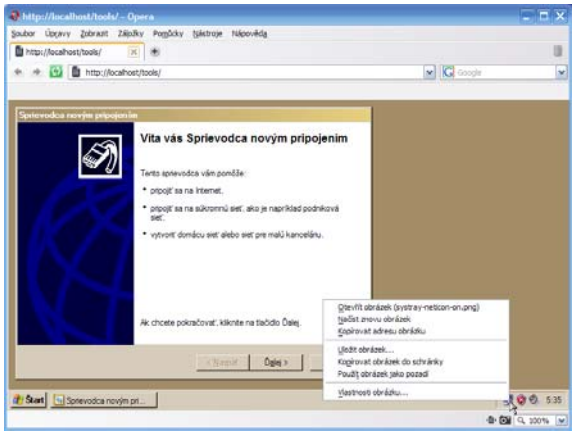
Samotné rozhranie definuje triedy pre vytváranie inštancií: kaskádových ponúk, okien, ovládacích prvkov a ďalších komponentov reagujúcich na udalosti klávesnice a myši tak, aby bolo ich správanie (pokiaľ možno) na nerozoznanie identické s prvkami GUI Windows.



Obr. 1 Ukážka testovacieho prostredia v prehliadači Mozilla Firefox

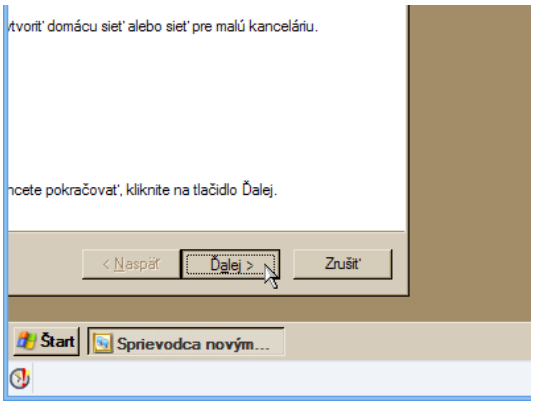


Obr. 2 Ukážka testovacieho prostredia v prehliadači Microsoft Internet Explorer



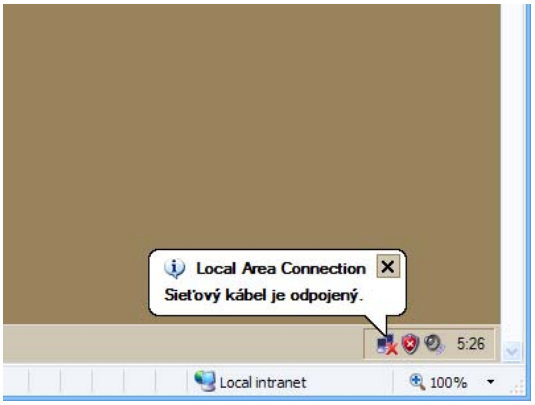
Obr. 3 Ukážka testovacieho prostredia v prehliadači Opera

Obrazové snímky okien, komponentov a definície ponúk v kombinácii s DOM internetového prehliadača, sa ukázali byť dostatočne silným nástrojom pre vytvorenie simulovaného prostredia. Jazyk JavaScript/JScript opäť ukázal svoje možnosti.

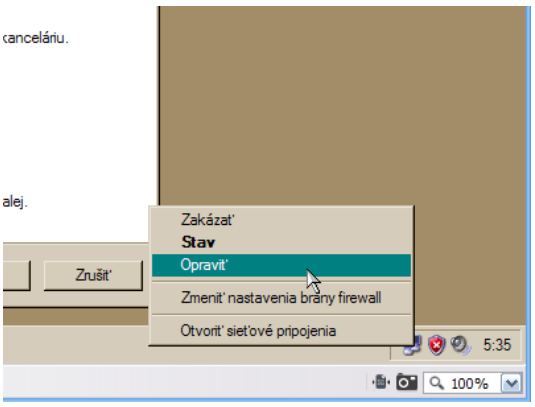


Obr. 4 Detail ovládacích prvkov – tlačidlá

Okrem vizuálnych prvkov, sú v rozhraní definované pomocné prvky nevizuálneho charakteru. Napríklad programová obálka pre zachytávanie udalostí myši, ktorá je svojím spôsobom „rodičovskou triedou“ pre ďalšie definície (ak vôbec v rámci JavaScriptu/JScriptu môžeme hovoriť o dedičnosti na tej kvalitatívnej úrovni ako ju poznáme z iných jazykov).



Obr. 5 Detail ovládacích prvkov – oznamovacia oblasť s bublinou



Obr. 6 Detail ovládacích prvkov – kontextová ponuka

Vývoj rozhrania nie je ukončený. Bude vyvíjané aj v priebehu implementácie EVM pre predmet SIKT, ktorý je (ako bolo spomenuté) ďalším nástrojom testovania. Počas procesu implementácie EVM budú odhalené také nedostatky, na detekciu ktorých rámcové testovacie prostredie nestačí. Taktiež môžu vzniknúť požiadavky na implementáciu ďalších tried, na ktoré v priebehu doterajšieho procesu vývoja buď nebolo myslené, alebo ktorých implementácia bola z rôznych dôvodov odložená.

Komponenty rozhrania viditeľné na obrázkoch obr. 4 až obr. 6, sú zostavené použitím dvoch mierne odlišných prístupov:

(1) úplné zostavenie vizuálnej časti komponentu pomocou DHTML/CSS elementov a následné „oživenie“ pomocou JavaScriptu/JScriptu – takto sú tvorené kaskádové ponuky, zadávacie polia apod.

(2) „oživenie“ sústavy výstrižkov obrazovkových snímok komponentu (screenshotov; vyhotovovaných v rôznych stavoch) pomocou JavaScriptu/JScriptu, čo zahŕňa aj vzájomnú hierarchizáciu jednotlivých objektov – takto sú implementované okná s tlačidlami, ikony v oznamovacej oblasti apod.

Použitie programátorského rozhrania pozostáva zo série operácií: zo zostavenie základného dokumentu (ako vzor môže poslúžiť dokument rámcového testovacieho prostredia), vytvárania inštancií definovaných tried, volania ich metód na správnych miestach v rámci hlavného DHTML dokumentu a obsluhy udalostí vstupu od používateľa pomocou spätných volaní (callback) funkcií.

Prvým krokom je vytvorenie hlavného dokumentu, pretože niektoré časti simulácie, ako pracovná plocha Windows, panel úloh s tlačidlom Štart a oznamovacou oblasťou, musia mať explicitné zastúpenie vo forme sústavy HTML elementov. Potom nasleduje tvorba inštancií objektov programátorského rozhrania. Podrobnosti sú nad rámec tohto dokumentu, uveďme jednoduchý príklad – dvojica volaní:

```
oWindowWizard1 =
   new ClassWindow
   ('oWindowWizard1');

oWindowWizard1.Create
   ('window-wizard1-enabled.png',
   'window-wizard1-disabled.png',
    null);
```

vytvorí základ okna sprievodcu viditeľného na obrázkoch obr. 1 až obr. 3. Volaním ďalších metód sú do okna pridávané tlačidlá a ďalšie komponenty, ktoré majú preddefinované správanie v súlade s funkcionalitou GUI Windows, ale podrobnosti správania sa simulácie určuje autor EVM. Pri tom sú aktívne využívané spomenuté callback funkcie. Ich názvy sú zadávané ako argumenty metód inštancií rozhrania. Dokumentácia k rozhraniu je vytváraná paralelne s ním.

3. PROBLÉMY PRI IMPLEMENTÁCII

Základným problémom pri webovom programovaní je rôznorodosť webových prehliadačov a z toho vyplývajúca nekompatibilita. Pri CSS sú to nekonzistencie zobrazení jednotlivých elementov a pri JavaScripte vs. JScripte je to odlišná podpora vlastností preddefinovaných objektov a rozdielne správanie vyplývajúce z odlišnej interpretácie toho istého skriptu rôznymi prehliadačmi.

Väčšinu problémov vyplývajúcich z nekompatibilít bolo možné vyriešiť. Ako vidno na obrázkoch obr. 1 až obr. 3, rozhranie je schopné korektne simulovať GUI prostredie prinajmenšom v troch najpoužívanejších prehliadačoch – Mozilla Firefox 3.0 (ďalej MozFF3), Microsoft Internet Explorer 7.0 (ďalej MSIE7) a Opera 9.63 (ďalej Opera9). Okrem nich bolo prostredie testované aj v prehliadačoch Safari 3.1.2 (vo verzii pre Windows) a Google Chrome 1.0.

Dva problémy nebolo možné vyriešiť ani po dlhodobom pátraní na internetových fórach riešiacich podobné otázky. Prvý sa týka prehliadača MozFF3. V prehliadači dochádza ku „zadrhávaniu“ simulovaných okien počas presúvania metódou Drag&Drop. Problém nastane vždy po opakovanom pokuse o presun okna. Keďže v starších verziách problém nenastával, bol nahlásený pomocou služby Bugzilla. Odozva komunity bola pozitívna s prísľubom nápravy v najbližšej hlavnej verzii Firefoxu.

Druhý sa týka prehliadača Opera9. Prejav je viditeľný na obr. 3. Napriek všetkej snahe sa, ani po dodržaní všetkých inštrukcií publikovaných na Internete, nepodarilo v tomto prehliadači potlačiť štandardnú reakciu na stlačenie sekundárneho tlačidla myši, ktorou je vyvolanie štandardnej kontextovej ponuky prehliadača. Tá pri tejto simulácii prekáža. Ako si môžete všimnúť na obr. 6, ktorý bol zosnímaný chvíľu po obr. 3 – hneď po stlačení klávesu ESC – kontextová ponuka simulácie je korektne zobrazená aj v prehliadači Opera. Nachádza sa však pod štandardnou ponukou prehliadača. Keďže primárna pozornosť bola venovaná prehliadačom MozFF3 a MSIE7, nebol zatiaľ tento problém konzultovaný si strediskom Opera. Tento postup bude zvážený neskôr.

4. PLÁNOVANÉ VYUŽITIE ROZHRANIA

Vďaka použitej technológii, má akýkoľvek materiál zhotovený s využitím tohto programátorského rozhrania široké možnosti využitia. Materiál môže byť prakticky bez výrazných obmedzujúcich kritérií zverejnený on-line, dodávaný v off-line verzii alebo zverejnený v elektronickom vzdelávacom portáli LMS v pasívnom režime (bez sledovania času a postupu študentov). V aktívnom režime jedine po dopracovaní a rozšírení o príkazy rozhrania SCORM.

V prípade schválenia zaradenia do elektronického vzdelávacieho systému, sa rozhranie ako časť EVM stane súčasťou programu rozvoja elektronického vzdelávania a vzdelávania s podporou IKT prostriedkov, ktorý je u nás neustále rozširovaný, ako vidno z [10], [11], [12], [13] a ďalších.

5. ZÁVER

V článku sme mali možnosť oboznámiť sa s programátorským rozhraním pre tvorbu interaktívnych vzdelávacích materiálov, ktoré vyžadujú simuláciu grafického používateľského rozhrania proprietárneho OS. Boli sme informovaní o problémoch pri implementácii a o pilotnom využití rozhrania pri tvorbe interaktívneho vzdelávacieho materiálu pre podporu vyučovania predmetu SIKT. Ten slúži na testovanie, odlaďovanie rozhrania a identifikáciu ďalších potrieb na základe ktorých môže byť rozširované.

LITERATÚRA

[1] HÍC, P. – POKORNÝ, M.: *Grafové algoritmy v školskej praxi.* On-line kurz. Trnava : 2004. ISBN 80-8082-023-6.

[2] HÍC, P. – POKORNÝ, M.: *Binárne relácie.* On-line kurz. Trnava : 2005. ISBN 80-8082-023-6.

[3] HORVÁTH, R.: *Programovanie a algoritmy.* On-line kurz. Trnava. ISBN 978-80-8082-211-8.

[4] GAZDÍKOVÁ, V.: Príprava kurzov pre EKP. In: *Odborný seminár o využití LMS EKP v akademickom prostredí.* Nová Dubnica : 2006. ISBN 80-969555-5-1.

[5] GAZDÍKOVÁ, V.: *Základy dištančného elektronického vzdelávania, študijné texty.* Trnava : PdF TU, 2003.

[6] GAZDÍKOVÁ, V. – ŠKOLKOVÁ, K.: Skúsenosti so štúdiom z elektronického študijného materiálu. In: *Acta Facultatis Paedagogicae Universitatis Tyrnaviensis, Ser. D.* Trnava : 2003. ISBN 80-89074-84-7. s. 53-57.

[7] HÍC, P – POKORNÝ, M.: Computer Supported Teaching of Binary Operations. In: *Kultúra – priestor interdisciplinárneho myslenia 1.* Nitra : UKF, 2005. ISBN 80-8050-837-2. s. 87-91.

[8] HORVÁTH, R.: Vyučovanie informatiky na SŠ s využitím elektronického kurzu. In: *Didinfo 2007.* Banská Bystrica : FPV UMB, 2007. ISBN 978-80-8083-367-1. pp. 46.

[9] ŠKOLKOVÁ, K. – HORVÁTH, R.: Porovnanie úrovne IKT zručností študentov na PdF TU. In: *DidMatTech 2008.* Komárno : 2008. (in press)

[10] GAZDÍKOVÁ, V. – MIŠÚT, M. – ŠKOLKOVÁ, K. – GABAĽOVÁ, V.: Postavenie elektronického vzdelávania na TU v Trnave. In: *Sborník příspěvků z mezinárodní konference e-Learning ve vysokoškolském vzdělávání 2004.* Zlín : Univerzita Tomáše Bati ve Zlíně, 2004. ISBN 80-7318-190-8. s. 50-54.

[11] HÍC, P – POKORNÝ, M.: E-learning vo vyučovaní matematiky. In: *Acta Mathematica 8.* Nitra : UKF, 2005. ISBN 80-8050-896-8. s. 113-118.

[12] GAZDÍKOVÁ, V. – ŠKOLKOVÁ, K. – MIŠÚT, M.: Integration of Information and Communication Technology into Education. In: *Sborník z mezinárodní konference Information & Comunication Technology in Education 2004.* Rožnov pod Radhoštěm : 2004. ISBN 80-7042-993-3. pp. 165-168.

[13] ŠKOLKOVÁ, K. – GAZDÍKOVÁ, V.: Príprava učiteľov na výučbu s podporou IKT. In: *Zborník z konferencie DIVAI 2004.* Nitra : 2004. ISBN 80-8050-691-4. s. 69.

INOVÁCIE KURIKULA A VÝBER VÝUČBOVÝCH METÓD

ING. JANA JACKOVÁ

Katedra informatiky, Fakulta riadenia a informatiky, Žilinská univerzita v Žiline, Univerzitná 8215/1, Žilina, tel. 041/513 4170, e-mail: Jana.Jackova@fri.uniza.sk

Článok sa venuje problematike rozhodovania pri inováciách kurikula bakalárskeho študijného programu Informatika. Nastoľuje otázku argumentácie pri výbere výučbovej metódy a spôsobu presadzovania výsledkov pedagogického výskumu. Uvádza prípadovú štúdiu, v ktorej sa čitateľ dozvie, ako sa konkretizovali závery širokej diskusie na fakulte pripravujúcej budúcich inžinierov do inovovaného kurikula základného kurzu programovania. Článok poskytuje základnú orientáciu v možnostiach pri výbere výučbových metód uvedením ich klasifikácie podľa novších prameňov. Všeobecný prístup je konkretizovaný na ukážke zaradenia didaktického modelu „mastery learning", efektívnosť ktorého autorka článku experimentálne overuje vo svojej výučbe.

Kľúčové slová: *kurikulum bakalárskej informatiky, programovanie, Java, vyučovacia metóda, mastery learning*

1. PRÍPADOVÁ ŠTÚDIA

Na fakulte technického zamerania prebehla medzi učiteľmi počas akademického roka 2007/2008 široká diskusia k reštrukturalizácii bakalárskeho študijného programu Informatika. Dôvodom bolo prispôsobenie výučby súčasným požiadavkám praxe zohľadňujúc pri tom medzinárodné kurikulárne dokumenty i vstupnú úroveň a očakávania prichádzajúcich študentov [1].

Medzi nosné predmety prvého ročníka [2] patrí základný kurz programovania v predmetoch Základy informatiky I, II (rozsah 2-1-1), ktorý je prerekvizitou takmer všetkých následných informatických predmetov. Študenti v ňom prejdú algoritmizáciou, základmi softvérového inžinierstva a programovacích techník, oboznámia sa so štruktúrovaným i objektovým prístupom a zvládnu modulárne programovanie. V rámci vyššie uvedenej diskusie bol živo diskutovaný prvý programovací jazyk (doteraz Pascal), ale tiež obsah predmetu Základy informatiky, medzipredmetové vzťahy, týždenný počet hodín a ich štruktúra, i spôsob výučby.

Zainteresovaní pedagogickí pracovníci sa stretávali na špeciálnom seminári, mali prizvaných odborníkov z praxe i z iných škôl podobného zamerania, preštudovali relevantné zahraničné pramene, viedli rozhovory aj mimo seminára na jednotlivých katedrách a podávali návrhy. K výberu prvého programovacieho jazyka prebehla anketa i hlasovanie (odskúšanú procedúru na výber vhodného prvého programovacieho jazyka sme uviedli v [1]). Čo sa týka metodiky výučby programovania, boli riešené najmä otázky:

- Čím začať ?
  - Kedy objekty: najprv / čo najskôr / až ku koncu?
  - Čo potom s algoritmizáciou?
  - Kedy začať s podprogramami?
- Aká učebnica?
- Prístup k študentovi: diferencovať študijné skupiny podľa vstupnej úrovne?, čo s talentovanými študentami?

Prvým programovacím jazykom sa od akademického roka 2009/2010 stáva Java (doteraz sa ho mali možnosť naučiť študenti vyšších ročníkov vo voliteľnom predmete). Prijaté nové kurikulum študijného programu zahŕňa zmeny v predmetoch i v ich následnosti.

Predmet Základy informatiky bude posilnený o dve vyučovacie hodiny týždenne (rozsah 2-2-2). Učitelia sa intenzívne pripravujú na inovovanú výučbu, na katedrovom seminári konfrontujú doterajšiu výučbu s plánovanou. Pri príprave jednotlivých tém sa oi. oboznamujú aj s prístupom C. Horstmanna [3] podľa jeho učebnice Big Java [4], ktorá má aj webovú podporu [5]. Táto kniha je určená pre prvý informatický kurz (v odbore "computer science"), lebo sa popri oboznamovaní so syntaxou jazyka zameriava na koncepty informatiky. Objektovo-orientovaný prístup je v nej uvedený postupne, čo je pružnejší prístup oproti "objects-early" (objekty zo začiatku) alebo "objects-late" (objekty ku koncu). V prvom semestri budú študenti pracovať vo vývojovom prostredí DrJava.

2. ÚLOHA DIDAKTIKY VO VÝBERE VYUČOVACEJ METÓDY

Motivácia:
Stretnete sa s kolegom, ktorý vás bude presviedčať o tom, že metóda vyučovania, s ktorou sa nedávno zoznámil, je zaručene najsprávnejšia. Máte o tom trochu pochybnosti. Aké otázky mu položíte, aby ste mohli posúdiť jeho tvrdenie? [6]

Pri argumentácii a presadzovaní názorov k riešeným problémom v uvedenej prípadovej štúdii boli niekedy účastníci diskusií vystavení „silovému" prístupu niektorých aktérov, prípadne sa stávalo, že niektoré vystúpenia boli príliš emocionálne. Najčastejšie sa tak stávalo v súvislosti s výberom prvého programovacieho

jazyka a pri diskusiách o použití výučbových metód. V takýchto prípadoch mohli byť pravdepodobne diskutované témy zvládnuté konštruktívnejšie, ak by sa oi. brali viac do úvahy výsledky výskumov z oblasti didaktiky informatiky a poznatky pedagogických vied (učitelia technických odborov vysokých škôl vo všeobecnosti nemusia mať pedagogické vzdelanie). Pozrime sa preto v tejto kapitole bližšie na problematiku výberu výučbovej metódy z pohľadu klasifikácie didaktických teórií a modelov. Tému sme spracovali podľa [6],[7].

Výučbová metóda je koordinovaný systém vyučovacích činností učiteľa a učebných aktivít žiaka, ktorý je zameraný na dosiahnutie učiteľom stanovených a žiakmi akceptovaných výučbových cieľov. Ak má byť zvolená výučbová metóda didakticky účinná, je žiadúce, aby bola:

1. informatívne nosná
2. formatívne účinná
3. racionálne a emotívne pôsobivá
4. rešpektuje systém vedy a poznania
5. výchovná
6. prirodzená v svojom priebehu a dôsledkoch
7. použiteľná v praxi, v skutočnom živote, približuje školu životu
8. adekvátna žiakom
9. adekvátna učiteľovi
10. didakticky ekonomická
11. hygienická

Znalosť problematiky **klasifikácie didaktických teórií a modelov** nám umožňuje poznať, že „nová“ metóda je obvykle len variantom inej, už predtým rozpracovanej metódy. Zaradenie určitej metódy do všeobecnejšej skupiny umožní tiež vopred odhadnúť, v čom asi budú jej silné a slabé stránky. Niekedy rôzne didaktioké teórie hovoria o tom istom, iba používajú rôznu terminológiu (duplicita), inokedy pre jednu vec odporúčajú ako optimálne rôzne postupy (rozpor), niekedy sa zamerajú na rôzne aspekty problému a ponúkajú pre ne rôzne riešenia (vzájomne sa dopĺňajú a dá sa na ich základe vytvoriť aj všeobecnejšia teória pokrývajúca širší okruh aspektov, integrovať ich).

Pri popise nejakej časti sociálnej reality sa často stretávane s pojmami makroúroveň a mikroúroveň. **Makrodidaktické teórie** sa týkajú zásad usporiadania vzdelávacieho systému, kurikula na národnej úrovni apod., alebo sa skôr zameriavajú na to, ako vybrať a usporiadať prvky učiva, zatiaľ čo **mikrodidaktické teórie** študujú dianie v triede, medzi učiteľom a žiakmi, samotnú výučbu, prípadne tiež špecifikujú, ako optimálne vyučovať konkrétny prvok učiva.

Učenie je ústredným pojmom psychológie aj didaktiky. Spôsob, akým vyučujeme, vychádza z našich predstáv o tom, ako sa ľudia učia. Preto **väčšina didaktických teórií vychádza z určitých teórií učenia**. V triede ale nevystačíme len s jednoduchou aplikáciou psychologických poznatkov o učení, musíme si tiež všímať špecifickú štruktúru učiva, ktoré má žiak zvládnuť. Interakciu medzi vyučovaním a procesom učenia žiaka, a tiež vplyv sociálneho kontextu (spolužiaci, ďalší ľudia, komunita, jeho kultúra) na poznávacie procesy žiaka, na jeho učenie, študuje psychodidaktika. Zaoberá sa skúmaním edukačných procesov ako z hľadiska obsahového (t.j. v pohľade didaktiky), tak z hľadiska procesuálneho (t.j. v pohľade psychológie) [7].

Model vyučovania resp. didaktický model by mal definovať, ako má vyzerať:

1. učivo
2. metódy (stratégie) učenia
3. sociálne prostredie, v ktorom prebieha učenie

	Teória	**Model**
Cieľ	popis procesov	optimalizácia výsledkov
Typická situácia	laboratórna	školská, redukcia počtu premenných

Tab. 1 Vzťah medzi teóriami a modelmi učenia

Pohľady na učenie a vyučovanie

- **biologické a evolučné pohľady na učenie a vyučovanie:**

  Štúdium ľudského genómu a mozgu prináša lepšie porozumenie človeku vôbec a tiež predpokladom, ktorými je vybavený k učeniu.

- **podnety sociálnych a kultúrnych vied**

  Teórie školského vzdelávania v súčasnosti významne ovplyvňuje kultúrna psychológia. Z jej viacerých prúdov pedagogiku výrazne ovplyvnila činnostná teória (teória aktivity, praxe). Nový spôsob myslenia je dôsledkom nových úloh. Rizikom pri aplikácii prístupu činnostných psychológov je niekedy paradoxne zanedbanie analýzy konkrétnej činnosti, čo môže viesť k zavádzaniu radu mechanických „hier“ do vyučovania, ktoré v skutočnosti nijako nestimulujú rozvoj psychických štruktúr žiaka. Vnútorný svet kultúry jedinca nie je totožný s vonkajším svetom kultúry spoločnosti. Jedinec (jeho duševný svet) nie je pasívnym produktom sociálneho sveta, ktorý ho obklopuje, ale si aktívne vyberá svoje kultúrne prostredie. Štúdie individuálneho prežívania kultúrnych a sociálnych podmienok ukazujú, že jedinec môže presiahnuť svoje podmienky (transcendovať ich). Praktickým dôsledkom pre školu môže byť poznanie, že žiaci môžu prekonať zdanlivo determinujúce vplyvy ich sociokultúrneho prostredia.

- **podnety z technických odborov: systémová analýza**

  Systémová analýza (systémové inžinierstvo) je metodickým postupom, ktorý má zaručiť, že pri riešení zložitej úlohy sa nezabudne na žiadne dôležité hľadisko. Keďže tendencia k systémovej

analýze úloh a ich plánovaniu zvýšila účinnosť práce v mnohých odboroch, je snaha zaviesť ich aj do oblasti teórie a praxe vzdelávania. Systémový prístup je cenným nástrojom pre poznanie fungovania školy a pre riadenie výučby. V didaktike viedol k vytvoreniu modelu označovanému ako „instructional design“. Pri štúdiu otázok vzdelávania uvažujeme systémovo (holisticky) vtedy, keď si napr. uvedomujeme, že každá zmena, ktorá postihuje určitý prvok systému, bude mať vplyv na celý systém a jeho ostatné prvky a naopak. Je ale potrebné si uvedomiť, že aj v rámci systému sú niektoré veci dôležitejšie než iné. Preto je možné – a niekedy je aj výhodnejšie – študovať prvky systému samostatne. Zložité systémy ani nie sme schopní plne obsiahnuť. Vedomie previazanosti všetkých zložiek systému, ktorý sa dá len ťažko zmeniť, môže brať chuť a odvahu k reformám.

Systémové myslenie neoddeľuje otázky čo učiť (kurikulum) od otázok ako učiť a ako hodnotiť (formy a metódy vyučovania a evaluácia).
Kroky systémového prístupu k výučbe:
1. zistenie potrieb a súčasnej úrovne žiaka
2. formulovanie cieľov
3. návrh, realizácia a hodnotenie vyučovania

Dôležitá je (spoľahlivá) spätna väzba, nielen čo sa týka hodnotenia žiakovho výkonu, ale aj školy (organizácia musí tiež byť schopná učenia, reflexie), bez ktorej nemôže byť výučba efektívna.

Systémový prístup k výučbe si po stanovení cieľov predmetu, učiva príslušného ročníka alebo cieľov tematického celku v rámci návrhu výučby vyžaduje mikrodidaktickú analýzu učiva [6]. Analýza učiva na úrovni jednotlivých hodín alebo tém je podrobne rozpracovaná v preskriptívnych mikrodidaktických modeloch. Teoretickým východiskom týchto modelov je psychologická teória učenia, ktorá predpokladá, že:
1. je možné rozdeliť (didakticky analyzovať) učivo do niekoľkých málo typov alebo kategórií úloh (napr. naučiť sa fakt, naučiť sa postup, naučiť sa zákonitosť),
2. ku každému z týchto typov (kategórií učiva) je možné stanoviť optimálne podmienky pre učenie alebo postup, ako žiakov daný typ poznatkov naučiť.

Pritom je dôležité žiakom explicitne ukazovať, zviditeľňovať a pripomínať štruktúru učiva. Učiteľ musí mať tiež na zreteli, že je rozdiel medzi **informovaním** a **vyučovaním** (inštruovaním). Oznámenie informácie, ktorá nezahŕňa predvedenie, príležitosť k precvičeniu a pomoc pri samostatnom používaní je len informovaním, ale nie vyučovaním ani navodzovaním riadeného učenia.

Konštruktivizmus nie je teória vzdelávania, ale umožňuje formulovať významné závery pre vzdelávanie, napr. všeobecne prijímané tvrdenie, že učenie je aktívny proces. Učenie znamená zmenu učiaceho sa subjektu, žiaka, a môže sa uskutočniť len prostredníctvom toho, čo žiak robí. Činnosť učiteľa je dôležitá natoľko, nakoľko vedie žiaka k tomu, aby sa zapojil do určitej činnosti, do ktorej by sa inak nezapojil. Deje sa tak predovšetkým prostredníctvom učebných úloh. Nejde pritom iba o činnosť vonkajšiu, ale predovšetkým o činnosť vnútornú, o „prácu na sebe, na vlastných poňatiach sveta“, o aktívnu zmenu mentálnych štruktúr. Odovzdávanie dôležitých tradičných kultúrnych poznatkov by sa nemalo diať spôsobom, ktorý by nielen nepodporoval rozvoj žiaka, ale dokonca mu prekážal. Konštruktivistické vyučovanie je každá zámerná, reflektovaná vzdelávacia činnosť, ktorá je zameraná na podporu žiakovho aktívneho porozumenia.

Rôzne formy a metódy vyučovania môžeme klasifikovať aj podľa toho, z akej predstavy o podstate a podmienkach učenia vychádzajú [6]:
- **behaviorálne modely,**
- **pesonálne (humanistické modely),**
- **sociálne modely,**
- **skupina kognitívnych modelov (modelov spracovania informácie)**.

Text tejto kapitoly je východiskom k pochopeniu širších súvislostí vedúcich k výberu vhodného modelu vyučovania. Neexistuje žiadna jednotná, všetkými prijímaná didaktická teória ani optimálny model vyučovania, ale niekoľko výrazných prúdov, v rámci ktorých existujú desiatky teórií a modelov vyučovania. Výskumy v tejto oblasti ukazujú, že najlepším postupom je zodpovedné kombinovanie rôznych teoretických prístupov podľa kontextu a potrieb žiaka.

3. MASTERY LEARNING (ML)

Ako ilustračný príklad k predchádzajúcej kapitole uvádzame psychodidaktickú koncepciu „mastery learning“ (ďalej ML), ktorej efektívnosť overujeme vo výučbe kurzu programovania z prvej kapitoly [8], [9]. ML nájdeme v tabuľke didaktických modelov [6, s. 318] zaradenú takto:

Model	programované učenie, individualizované učenie, učenie riadené počítačom
Varianty	***mastery learning***, Kellerov plán, CAL (Computer Assisted Learning), NLP (Neuro Linguistic Programming),
Didaktický princíp	zvládanie cieľov, malé úseky učiva, postup podľa potrieb žiaka, individuálny prístup

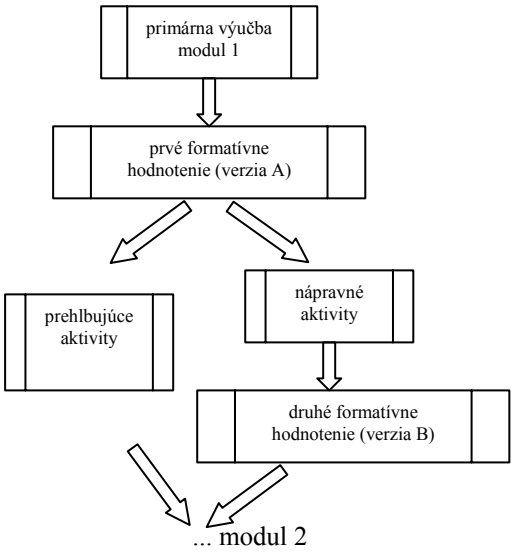
Tab. 2 Zaradenie ML ako didaktického modelu

V slovenskej odbornej literatúre sa popri anglickom termíne používajú ekvivalenty *učenie sa zvládnutím* (Gavora, Mareš), *systém dokonalého osvojenia učiva* (Turek), *zvládacie učenie sa* (Švec), *majstrovské učenie sa* (Zelina) [10].

Bloomova psychodidaktická teória ML je najvýznamnejším príkladom **humanistických teórií** týkajúcich sa školskej edukácie [7]. Táto teória pracuje s predpokladom, že zvládnutie nejakého súboru poznatkov alebo zručností je teoreticky možné u všetkých žiakov, ak majú vhodné podmienky, a to

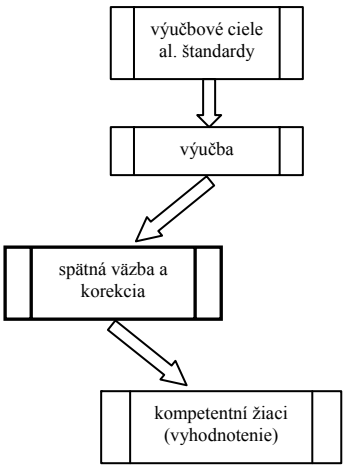
- spôsob vyučovania optimálne prispôsobený žiakovi a tiež
- množstvo času, ktoré žiak k učeniu potrebuje.

Základnou ideou ML je, že žiak sa učí preto, aby vedel, a tiež snaha dokázať, že nie je opodstatnené tvrdenie, že existujú na jednej strane "dobrí" žiaci (t.j. s vyhovujúcim prospechom, úspešní) a na druhej strane „zlí" žiaci (t.j. neprospievajúci, s nízkou školskou úspešnosťou).



Obr. 1 Výučbový proces v ML

Preštudovaním problematiky ML podľa [6], [7], [11], [12], [13] dôjdeme k záveru, že ML integruje tie prvky z rôznych didaktických teórií uvedených v druhej kapitole, ktoré by mali zabezpečiť jeho úspešnosť v praxi.



Obr. 2 Hlavné zložky vzdelávacieho procesu v ML

ML môže byť aplikovaný pri hromadnom vyučovaní, ale i pri individuálnej forme štúdia. Táto modifikácia je známa ako **Kellerov plán** (viď varianty v Tab. 2), resp. pod termínom „Personalized System of Instruction" (systém individualizovaného vyučovania) al. PSI.

Správne implementovaný ML by mal v podmienkach kurzu programovania z prvej kapitoly zvýšiť počet študentov, ktorí ho úspešne absolvujú. To by mohlo prispieť k riešeniu častého problému študentov prvého ročníka vysokých škôl, ktorým je odchod zo štúdia z dôvodu neprospechu.

LITERATÚRA

[1] Jacková, J.: Výučbové ciele v kontexte informaticky zameraných odborov technických fakúlt. In: *SCHOLA 2007 – Inžinierska pedagogika, zborník z 8. medzinárodnej vedeckej konferencie.* [CD-ROM]. Trnava: AlumniPress, 2007, 7 strán. ISBN 978-80-8096-038-4

[2] Fakulta riadenia a informatiky ŽU, študijný program Informatika http://www.fri.uniza.sk/index.php?page=1&categID=162

[3] Cay Horstmann's Home Page http://www.horstmann.com/

[4] Horstmann, C.: Big Java, 3rd Edition. Hoboken: Wiley, 2007. ISBN: 978-0-470-10554-2

[5] Big Java / Java Concepts http://www.horstmann.com/bigjava.html

[6] Kalhous, Z., Obst, O. a kol.: Školní didaktika. Praha: Portál, 2002. ISBN 80-7178-253-X

[7] Průcha, J.: Moderní pedagogika. Praha: Portál, 2002. ISBN 80-7178-631-4

[8] Jacková, J.: Systém dokonalého osvojenia učiva vo výučbe programovania. In: *Konferencia DidInfo 2008.* Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied, 2008, s. 18, [CD-ROM] 2 strany. ISBN 978-80-8083-556-9

[9] Jacková, J.: Learning for Mastery in an Introductory Programming Course. In: *ACM SIGCSE Bulletin*. Volume 40 , Issue 3 (September 2008). New York, NY, USA: ACM Association for Computing Machinery, 2008. P. 352. ISSN 0097-8418. [on-line] http://portal.acm.org/citation.cfm?doid=1384271.1384394

[10] Grofčíková, S., Tirpáková, A.: Vplyv koncepcie mastery learning na úroveň vzdelávacích výsledkov žiakov ZŠ. In: *Výchova, škola, spoločnosť - minulosť a súčasnosť.* Bratislava: Univerzita Komenského, 2007, s. 155 – 161. ISBN 978-80-223-2295-9

[11] Guskey, T. R.: Formative Classroom Assessment and Benjamin S.Bloom: Theory, Research, and Implications. Paper presented at the Annual Meeting of the American Educational Research Association, Montreal, Canada. April 2005. [on-line] http://www.eric.ed.gov/ERICDocs/data/ericdocs2sql/content_storage_01/0000019b/80/1b/c1/35.pdf

[12] Petty, G.: Teaching Today. Cheltenham: Nelson Thornes, 2004. ISBN 0-7487-8525-6

[13] Turek, I.: Didaktika. Bratislava: Iura Edition, 2008. ISBN 978-80-8078-198-9

PRÍPRAVA BUDÚCEHO UČITEĽA V REÁLNOM PROSTREDÍ ŠKOLY

MGR. BRONISLAVA JAKUBÍKOVÁ, MGR. ZUZANA JUROŠKOVÁ, MGR. LENKA LÍŠKOVÁ

Katedra knižničných a informačných vied a didaktiky informatiky, Fakulta prírodných vied, Žilinská univerzita v Žiline, Hurbanova 15, 010 26 Žilina
tel. 041/513 6338, e-mail: bronislava.jakubikova@fpv.uniza.sk

Príspevok sa zaoberá výsledkami aktivity pod názvom Pilotný projekt, ktorá bola realizovaná v rámci projektu ESF „Adaptácia prípravy študentov učiteľstva na informačnú spoločnosť" riešenom na Fakulte prírodných vied Žilinskej univerzity v Žiline v rokoch 2007-2008. V článku sú naznačené možnosti praktickej prípravy študentov učiteľstva akademických predmetov v podmienkach praxe reálnej školy.

Kľúčové slová: *didaktika, IKT v predmetoch, príprava učiteľov, spolupráca s praxou*

1. ÚVOD

V podmienkach nastupujúcej znalostnej spoločnosti je potrebná nielen reforma školstva v zmysle reformy foriem práce a obsahu vzdelávania, ale aj v zmysle adaptácie prípravy budúcich učiteľov na nové podmienky súčasnej školy. Na prípravu budúcich učiteľov sú kladené obrovské nároky v rámci nadobúdania kľúčových kompetencií v oblasti odbornej prípravy, ale aj zvládania výchovných problémov dnešnej doby.

2. PILOTNÝ PROJEKT PROJEKTU ESF

Začínajúci učiteľ je konfrontovaný v praxi nielen s učivom, ale aj so žiakom a rodičom. Absolvent pedagogického vzdelávania je pripravený ako fundovaný odborník v aprobačných predmetoch, pozná didaktické zásady, metódy a formy práce so žiakom, ktoré v umelom prostredí seminára dokáže kvalifikovane uplatniť. Vysokoškolské vzdelávanie pripravuje učiteľa teoretika, ktorý v praxi stráca pevnú pôdu pod nohami. Prvý kontakt s reálnym žiakom a rodičom býva krutou ukážkou toho, že len dokonale zvládnutá teória nestačí. Prax absolventa získavaná počas priebežnej a súvislej praxe je absolútne nedostatočná.

V rámci realizácie aktivity pilotný projekt, ktorá bola súčasťou ESF projektu pod názvom *Adaptácia prípravy študentov učiteľstva na informačnú spoločnosť*, riešeného na pôde Fakulty prírodných vied ŽU v Žiline v rokoch 2007-2008, sa nám naskytla možnosť realizovať časť prípravy študentov učiteľstva v praxi základnej školy. Študentky piateho ročníka učiteľstva akademických predmetov v kombinácii matematika - informatika a anglický jazyk - informatika odučili v podmienkach základnej školy Vendelína Javorku 32 v Žiline blok vyučovacích hodín svojho prvého aprobačného predmetu, t.j. matematika a anglický jazyk, s využitím informačno-komunikačných technológií. Pilotný projekt tohto vyučovania sa realizoval v dvoch fázach, pričom v prvej fáze sa študentky zoznámili formou náčuvov s triedami, v ktorých budú vyučovať i s vyučujúcimi zabezpečujúcimi vybrané predmety (matematika, anglický jazyk). Druhú fázu tvorilo samostatné vyučovanie študentiek FPV na pôde ZŠ. Výber ročníka, v ktorom bolo aplikované vyučovanie bol podmienený prebiehajúcou reformou školstva a preto bol zámerne vybratý piaty ročník ZŠ, aby sa podporilo vnímanie zmien v rámci reformy u budúcich učiteľov. Pri príprave na vyučovanie si študentky museli podrobne naštudovať štátny vzdelávací program, ako i oboznámiť sa s konkrétnym školským vzdelávacím programom a v diskusii s učiteľmi z praxe rozobrať podstatu zmien, ktoré priniesla reforma do podmienok ZŠ. K faktu, že vyučovanie matematiky a anglického jazyka bude študentkami realizované s využitím IKT sa aj zabezpečujúci učitelia stavali vysoko pozitívne, nakoľko práve vhodná kombinácia aprobačných predmetov našich študentiek s informatikou dávala vysoký predpoklad dobre zvládnutej aplikácie IKT do oboch vybraných predmetov.

Obr. 1 Náčuv na hodine anglického jazyka

Samotná realizácia vyučovacích hodín bola hodnotená všetkými zúčastnenými stranami vysoko pozitívne. Študentky ocenili možnosť takejto formy praxe pod supervíziou didaktika informatiky v spolupráci s učiteľom z praxe, pretože cvičné hodiny realizované na seminároch v rámci jednotlivých didaktík zďaleka nedosiahli takú sebareflexiu a neumožnili takú spätnú väzbu, ako

hodiny odučené na základnej škole. Mikrovýstup realizovaný v umelom prostredí seminára nikdy neobsiahne podmienky hodiny v skutočnej triede, s reálnymi žiakmi, ich charaktermi, správaním, rozmýšľaním a prípadnými hendikepmi. Atmosféru, ktorá vládne v reálnej triede sa nikdy nepodarí umelo vytvoriť v prostredí vysokej školy, nech by sme sa snažili akokoľvek. Učiteľky základnej školy sa zoznámili s novými softvérmi vhodnými na výučbu ich predmetov, respektíve jednotlivých častí tematických celkov, ale hlavne získali „bezpracne" od mladých študentiek know - how a budú schopné didaktický softvér použitý na ukážkovej hodine ďalej použiť.

Na hodinách matematiky bol používaný program Mafix na precvičovanie delenia viacciferným číslom a na hodinách anglického jazyka to bolo praktické využitie www stránok, vypĺňanie oblastí v grafickej predlohe a kreslenie v skicári využité pri utvrdzovaní slovnej zásoby v rámci tém čísla – farby - moja rodina. Obidva odučené predmety anglický jazyk aj matematika boli pripravené len s minimálnymi požiadavkami na prípravu učiteľa, čo učiteľky z praxe výrazne ocenili. V rámci hodín bolo využité delenie detí do pracovných skupín. Obe skupiny pracovali súbežne, jedna skupina s pracovnými listami a druhá skupina s počítačom. Skupiny sa v aktivitách striedali tak, aby deti boli motivované, zaujaté a povzbudené novou činnosťou. Pracovné listy využité pri vyučovaní si študentky učiteľstva pripravili samostatne, obsahovali zaujímavé, hravé typy úloh, zohľadňovali zásadu primeranosti. Študentky na hodinách matematiky predviedli učiteľkám z praxe, ako je možné počítač využiť tak, aby umožnil učiteľovi individuálnu prácu so žiakom. Skupina, ktorá pracovala s počítačom bola kontrolovaná a vyhodnocovaná softvérom, zásah učiteľa bol minimálny – hlavne v závere práce, kde bol priestor pre pozitívnu motiváciu detí formou pochvaly za peknú prácu s počítačom a matematikou zároveň. Takto vznikol učiteľovi priestor na venovanie sa druhej skupine a riešenie a kontrolovanie výsledkov práce žiakov v pracovných listoch.

Obr. 2 Hodina matematiky

Spätná väzba od žiakov po realizovaní takéhoto typu vyučovania bola jednoznačne pozitívna. Aj integrovaní žiaci vykazovali výbornú úroveň zvládnutia témy, čo potvrdilo aj zabezpečujúcim učiteľkám ZŠ zmysluplnosť vyučovania s využitím IKT v predmete. Učiteľky základnej školy, ktoré sa doteraz obávali zaradiť IKT do svojho predmetu, videli jeho výhody (pri správnom aplikovaní) a ocenili hlavne možnosť rôzneho tempa práce zohľadňujúceho individuálne schopnosti žiaka. Tento individuálny prístup sa pozitívne prejavil najmä v triedach s integrovanými žiakmi.

Základná škola Vendelína Javorku umožnila vo svojom prostredí realizáciu vyučovania nielen vo výberových triedach so zameraním na cudzí jazyk, ale aj v klasických triedach, ktoré v rámci podmienok školy sú naplnené slabšie prospievajúcimi a integrovanými žiakmi. V praxi to znamená aj spoluprácu učiteľa s asistentom žiaka a špeciálnym pedagógom školy pracujúcim v tejto škole na plný úväzok. Takáto skúsenosť mladého učiteľa je na nezaplatenie. Žiaci výberovej triedy boli naozaj aktívni, prácu s počítačom brali skôr ako zábavu a tak sa nenásilnou a hravou formou vzdelávali. Práca so žiakmi nevýberových tried však bola naozaj náročná. Bolo treba rešpektovať hlavne integrované deti a celú hodinu prispôsobiť tak, aby ani jedna skupina nebola „ochudobnená". Študentky tak nadobudli úplne iný pohľad na vyučovanie, získali veľa cenných skúseností, ktoré určite budú môcť využiť v budúcom povolaní. Je predpoklad, že do učiteľskej praxe budú nastupovať s reálnymi predstavami a očakávaniami.

Významnú úlohu v celom pilotnom projekte zohrala aj záverečná fáza realizácie tohto vyučovania, kde sa študentky, učiteľky a didaktik stretli v diskusii a v rozbore vyučovaných hodín. Došlo k výmene skúseností, definovaniu kladov, záporov a nedostatkov s prínosom pre všetky zainteresované strany.

3. SPOLUPRÁCA SO ZŠ

V rámci realizácie projektu boli nadviazané medzi FPV ŽU a manažmentom základnej školy nadštandardné vzťahy, ktoré sú základom ďalšej spolupráce. Ešte v prípravnej fáze realizácie vyučovania pri zisťovaní vybavenosti školy IKT sme narazili na fakt, že škola je vybavená aj e-beam tabuľou, ktorá nebola dostatočne využívaná. Práve táto skutočnosť nás viedla k záväzku, že po odučení hodín pilotného projektu, z našej strany (didaktika a študentov FPV) pripravíme ukážku využitia tabule e-beam vo vyučovaní predmetov základnej školy. Seminár pre učiteľov základnej školy viedol didaktik informatiky a študenti prezentovali pripravené príspevky pod supervíziou didaktika. Aj táto forma spolupráce bola súčasťou hodín didaktiky informatiky, kde študenti nemali mikrovýstupy

v umelom prostredí len pred didaktikom predmetu, ale boli konfrontovaní s učiteľmi z praxe, ich otázkami, či pripomienkami.

Pre didaktiku informatiky zabezpečovanú na Fakulte prírodných vied ŽU nezostávajú výsledky projektu *Adaptácia prípravy študentov učiteľstva na informačnú spoločnosť* mementom, ale výzvou a záväzkom do budúcnosti. Po dohode s manažmentom Základnej školy Vendelína Javorku 32 v Žiline budú semináre didaktiky informatiky v letnom semestri akademického roka 2008/09 realizované vo vyučovaní tejto školy. Musíme priznať fakt, že didaktik informatiky má šesťhodinový úväzok na tejto škole, ktorý významnou mierou prispeje k takejto realizácii seminára z didaktiky. Vedenie seminára v procese vyučovania základnej školy je náročné na prípravu samotného didaktika tak, aby boli zabezpečené všetky didaktické zásady a nebol negatívne narušený vyučovací proces ako taký. Študenti seminára Didaktika informatiky po úvodnom stretnutí absolvujú ďalšie vyučovacie hodiny seminára v dvoch šesťhodinových blokoch v skupinách. Pripravené mikrovýstupy zrealizujú v reálnej triede s reálnym žiakom v rámci tematických celkov plánovaných v tematickom pláne predmetu technická výchova - práca s počítačom.

4. VÍZIE DO BUDÚCNOSTI

Veríme, že takto realizovaná príprava budúcich učiteľov informatiky má význam a zmysel a dôkladnejšie pripraví absolventa učiteľského štúdia na výkon jeho budúceho povolania v súčasnej spoločnosti. Nechceme zostať len pri vyučovaní didaktiky informatiky v prostredí tejto základnej školy, ale aj ďalej rozvíjať naznačené formy spolupráce. Na jednej strane realizovať ukážky možnosti vyučovania druhého aprobačného predmetu našich študentov s využívaním IKT formou otvorených hodín pre pedagógov ZŠ, tak i na druhej strane prípravou pracovných seminárov s aktuálnymi témami, ktoré trápia učiteľa z praxe a študent učiteľstva môže pomôcť teoretickými vedomosťami. Budúcu spoluprácu vidíme aj v oblasti vedenia bakalárskych a diplomových prác, ktoré budú riešiť problémy výchovno-vzdelávacej praxe.

5. ZÁVER

Na základe skúseností z realizácie pilotného projektu je našou ambíciou vybudovať kvalitnú cvičnú školu pre prípravu študentov v prvom aj druhom stupni vysokoškolského štúdia učiteľstva akademických predmetov v kombinácii. Takéto prepojenie teórie a praxe považujeme za kľúčové v oblasti efektívnej a kvalitnej prípravy moderného učiteľa na požiadavky nastupujúcej znalostnej spoločnosti.

LITERATÚRA

[1] KYRIACOU, G.: Klíčové dovednosti učitele, Portál, Praha, 1996. ISBN 80-7178-022-7

[2] PETTY, G.: Moderní vyučování, Portál, Praha, 1996. ISBN 80-7178-070-7

SÚ TESTY ECDL VHODNÉ NA EVALUÁCIU POČÍTAČOVEJ GRAMOTNOSTI BUDÚCICH UČITEĽOV?

RNDR. ĽUDMILA JAŠKOVÁ PHD., MGR. JANKA PEKÁROVÁ

Katedra základov a vyučovania informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina, Bratislava,
e-mail: jaskova@fmph.uniba.sk, pekarova@fmph.uniba.sk

V článku popisujeme naše skúsenosti s využívaním systému ECDL na verifikáciu počítačovej gramotnosti študentov rôznych aprobácií učiteľského štúdia. Analyzujeme tu výsledky, ktoré v testoch študenti dosiahli. Zamýšľame sa nad vhodnosťou tohoto certifikačného systému pre učiteľov a porovnávame ho s certifikačným systémom EPICT, ktorý je určený pre pedagogických pracovníkov.

Kľúčové slová: *počítačová gramotnosť, testy ECDL, testy EPICT*

1. ÚVOD

Súčasťou procesu rozvoja digitálnej gramotnosti (ďalej už len DG) budúcich učiteľov by mal byť aj mechanizmus merania a evaluácie, resp. jej certifikácia. Jedným zo štandardných spôsobov certifikácie zručností pracovať s počítačom je v súčasnosti ECDL (European Computer Driving Licence). Aj keď táto certifikačná procedúra nevznikla pre potreby učiteľov a vzdelávacieho systému, momentálne na Slovensku nie je dostupný a akceptovaný žiaden vhodnejší variant, ktorý by certifikoval DG s ohľadom na potreby učiteľov, žiakov a moderného poznávacieho procesu.

Pripomeňme si, že ECDL [3] je štandardný spôsob merania zručností potrebných pre prácu s počítačom a využívanie jeho základných aplikácií (základy IT, používanie počítača a správa súborov, textový editor, tabuľkový kalkulátor, databázový systém, elektronická prezentácia, informácie a komunikácia). Na našej fakulte študenti učiteľského štúdia dosiahnu zručnosti na úrovni ECDL (s výnimkou databázových systémov) po úspešnom absolvovaní všetkých študijných jednotiek prvého ročníka štúdia (presnejšie, najmä vďaka predmetom IKT1 a IKT2) [1].

Zručnosti a vedomosti na úrovni ECDL považujeme za minimálne požiadavky DG, ktoré musia budúci učitelia spĺňať. Komplexná DG budúcich učiteľov totiž musí byť na kvalitatívne inej úrovni. Budúci učitelia musia získať IKT zručnosti, ale okrem toho musia byť schopní digitálne technológie využívať pre študentov atraktívnym, bezpečným a efektívnym spôsobom vo vyučovaní predmetov podľa svojej aprobácie.

Myslíme si, že na meranie schopností budúcich učiteľov v oblasti DG sa v budúcnosti uplatnia upravené ECDL testy (určené špecificky pre takúto cieľovú skupinu), alebo sa okrem testov ECDL budú schopnosti, špecifické pre DG učiteľov merať inými prostriedkami. Rozumnou alternatívou sa zdá byť systém vzdelávania EPICT (European Pedagogical ICT Licence) [4], ale s ním zatiaľ na Slovensku nemáme skúsenosti.

V nasledujúcich častiach popíšeme naše skúsenosti s prípravou študentov pre certifikáciu ECDL a v závere sa zmienime o počítačovej pedagogickej licencii EPICT.

2. POPIS PREDMETU IKT1

Predmet IKT1 [1] (Informačné a komunikačné technológie 1) majú študenti v prvom semestri štúdia. Obsahovo nadväzuje na skúsenosti a vedomosti, ktoré získali počas štúdia na strednej škole. Zameriava sa na rozvoj základnej DG, na používanie všeobecných softvérových nástrojov.

Absolvent predmetu IKT1 má byť schopný:

- efektívne pracovať so súbormi a priečinkami v počítači,
- používať internet na vyhľadávanie a prezentovanie informácií,
- používať rôzne nástroje na interaktívnu a neinteraktívnu komunikáciu,
- pracovať s informáciami rôznych typov, teda:
  - využívať nástroje textového editora na spracovanie textu,
  - vytvárať tabuľky a pracovať s údajmi v tabuľkách,
  - reprezentovať informácie pomocou tabuľky a grafu,
  - vytvárať a upravovať obrázky pomocou rastrového grafického editora;
- prezentovať informácie:
  - s využitím nástrojov prezentačného softvéru,
  - s využitím internetu – na vlastnej webovej stránke.

Predmet IKT1 absolvujú študenti učiteľstva všeobecnovzdelávacích predmetov týchto aprobácií:

- matematika – informatika,
- fyzika – informatika,
- geografia – informatika,
- biológia – informatika,
- chémia – informatika,

- matematika – fyzika,
- matematika – deskriptívna geometria,
- matematika – telesná výchova.

Počas nášho výskumu sme sledovali schopnosť študentov zvládnuť učivo na takej úrovni, aby boli schopní úspešne vykonať ECDL testy zo 4 modulov:

- Práca s počítačom a správa súborov (modul 2)
- Textový editor (modul 3)
- Tabuľkový editor (modul 4)
- Služby informačných sietí (modul 7)

Testy, ktoré študenti vykonávali počas nášho kurzu neboli reálne ECDL testy. Boli to testy s rovnakou formou, avšak obsahovo plne nepokrývali problematiku testovanú na reálnych testoch ECDL. Podľa plánu mali študenti v dostatočnej miere zvládnuť problematiku požadovanú na testoch ECDL až po absolvovaní predmetu IKT2.

Za úspešné riešenie sme považovali, ak študent získal aspoň 75% z celkového počtu bodov.

Na začiatku semestra študenti vykonali vstupný test a na konci semestra záverečný test.

Vstupný test absolvovalo 35 študentov.

Tabuľka 1 znázorňuje počty študentov v percentách, ktorí jednotlivé moduly zvládli aspoň na 75%.

Výsledky vstupného testu sú uvedené v druhom stĺpci tabuľky 1. Vidíme, že najmenšie problémy študentom robil modul 2 a najviac problémov moduly 4 a 7. Neúspešnosť pri riešení úloh z modulu 7 súvisela s technickými problémami v učebni a nie s neznalosťami.

Záverečný test absolvovalo 26 študentov.

Výsledky záverečného testu sú uvedené v treťom stĺpci tabuľky 1. Vidíme, že najmenšie problémy študentom robil modul 7 a najobťažnejšie boli moduly 3 a 4. V porovnaní so vstupným testom však vidíme výrazné zlepšenie v troch moduloch. Pri riešení úloh modulu 2 zase vidíme dramatické zhoršenie. Je to spôsobené tým, že záverečný test obsahoval rozsiahlejší test z teórie, ktorý robil študentom problémy.

Modul	Úspešnosť Vstupný test	Úspešnosť Záverečný test	Zlepšenie (+) Zhoršenie (-)
Modul 2	91,4%	73%	-18,4%
Modul 3	57,1%	62%	+4,9%
Modul 4	20,0%	65,4%	+45,4%
Modul 7	20,0%	92,3%	+72,3%

Tab. 1 Výsledky testov vykonaných v rámci predmetu IKT1

3. POPIS PREDMETU IKT2

Ciele predmetu IKT2 sú nasledujúce [5]:

- aby študenti získali prehľad o softvéri a hardvéri pre aktuálne informačné a komunikačné technológie, naučili sa inštalovať a spravovať počítač, pripojiť ho k sieti, spoznali princípy bezpečnosti prevádzky v sieti,
- prehĺbenie zručností v oblasti práce s textovým editorom, tabuľkovým kalkulátorom prezentačným softvérom a grafickým editorom,
- príprava študentov na získavanie certifikátu ECDL Štart (úspešné zvládnutie ľubovoľných štyroch modulov).

V závere kurzu študenti riešili 4 zvolené testy z modulov ECDL číslo 2, 3, 4, 7 a z modulu číslo 6 – Elektronická prezentácia.

Záverečného testovania sa zúčastnilo 32 študentov. Tabuľka 2 ilustruje počet absolventov jednotlivých testov a ich úspešnosť:

Modul	Počet študentov	Úspešnosť	Zlepšenie (+) Zhoršenie (-)
Modul 2	100%	91%	´+19%
Modul 3	97%	84%	+22%
Modul 4	97%	94%	+30,6%
Modul 6	75%	92%	-
Modul 7	31%	70%	-22,3%

Tab. 2 Výsledky testov vykonaných v rámci predmetu IKT2

Úspešnosť v moduloch 2, 4 a 6 presiahla 90%. Úspešnosť pri module č. 7 je zdanlivo nízka – len 70%. Tento modul však absolvoval pomerne malý počet študentov. Okrem toho sa pri realizácii testu vyskytli technické problémy s e-mailovými schránkami.

Ako vidíme vo štvrtom stĺpci tabuľky 2, v porovnaní so záverečným testom z IKT1 nastalo výrazné zlepšenie pri troch moduloch (2, 3 a 4). Pri riešení úloh modulu 7 zase vidíme zhoršenie. Ako sme spomenuli vyššie, nízka úspešnosť pri riešení tohto modulu však nebola spôsobená neznalosťami, ale technickými problémami.

4. NAJČASTEJŠIE PROBLÉMY PRI RIEŠENÍ TESTOV ECDL

Počas nášho výskumu sme zisťovali aj to, akých chýb sa študenti najčastejšie dopúšťali [1, 5] a čo boli najčastejšie príčiny ich neúspechu. Tu sme okrem podrobnej analýzy riešení testov vychádzali aj z analýzy príspevkov študentov do diskusných fór a z analýzy dotazníkov, ktoré študenti vypĺňali na konci semestra.

- **Spracovanie textu.** Najväčšie problémy robilo mäkké zalomenie, automatické rozdeľovanie slov, orámovanie bloku riadkov, hromadná korešpondencia, zmena písma na kapitálky, nastavovanie medzier pred a za odsekom, vloženie symbolu a zarážky tabulátora. Problémy boli aj s hlavičkou a pätičkou dokumentu, pozadím tabuľky, tvarom odrážok, nastaveniami dokumentu a odsadením prvého riadku v odseku.
- **Tabuľkový kalkulátor.** Najväčšie problémy robila funkcia IF, hlavička a pätička,

ukotvenie priečok, nastavenia dokumentu, formát dátumu a tlač grafu.

- **Informácie a komunikácia**. Najväčšie problémy robil príznak pripomienky, nastavenie priority správy, nastavenie webového prehliadača, nastavenie vlastností pre tlač webovej stránky a usporiadanie správ podľa nejakého kritéria. Menšie problémy robilo aj nastavenie záhlavia zoznamu správ a vyhľadanie správ podľa kľúčového slova. Problémy robili aj otázky z teórie .
- **Správa súborov.** Najväčšie problémy robilo zistenie počtu priečinkov, podpriečinkov a súborov (podľa rôznych kritérií), práca s pomocníkom pre danú aplikáciu, zistenie veľkosti RAM a zmeny atribútov pre súbory.
- **Elektronická prezentácia.** Najväčšie problémy robilo použite šablóny návrhu, práca s predlohou, zmena farby výplne pre automatické útvary, práca s organizačnou schémou, automatické číslovanie snímok a tlač osnovy. Menšie problémy robili aj farby v grafe.

Neúspešnosť pri riešení testov bola často spôsobená aj nepozornosťou pri čítaní zadaní úloh a nedôslednosťou pri riešení úloh. Študenti tiež podcenili teoretickú oblasť.

5. SKÚSENOSTI S VHODNOSŤOU POUŽITIA TESTOV ECDL

Pri hodnotení vhodnosti testov ECDL pre študentov učiteľstva vychádzame najmä z analýzy záverečných dotazníkov, ktoré študenti vypĺňali a z rozhovorov lektorov, ktorí semináre v rámci predmetov IKT1 a IKT2 viedli.

Za veľké výhody použitia systému ECDL v rámci našich predmetov považujeme:

- Je to medzinárodne uznávaný štandard.
- Študenti mali pocit, že získali v rámci štúdia znalosti, ktoré sú cenné aj v prípade, že neukončia vysokoškolské štúdium, prípadne dosiahnu iba titul bakalára.
- Je to veľmi dobrá metóda na získanie kvantitatívnych informácií o vedomostiach študentov. Pri našej malej vzorke študentov však potrebujeme aj ďalšie metódy (skôr kvalitatívne orientované) na zistenie príčin daného stavu a získanie spätnej väzby.

Za nevýhody použitia systému ECDL v rámci prípravy budúcich pedagógov považujeme:

- Nie sú tu zahrnuté poznatky z nasledujúcich oblastí:
  - Grafika – rastrová a vektorová.
  - Tvorba webových dokumentov.
  - Multimédiá (práca so zvukom, práca s videom, videokonferencie).
- V rámci oblastí, ktoré testy ECDL pokrývajú, nie sú zahrnuté nasledujúce poznatky, ktoré by študenti mali ovládať:
  - V rámci oblasti Spracovanie textu postrádame poznatky o vytváraní obsahu, indexu, zoznamu obrázkov, o editovaní matematických textov, diagramov, vektorových obrázkov, o práci s revíziami.
  - V rámci oblasti Tabuľkový kalkulátor, otázky zamerané na používanie vzorcov nevyžadujú hlbšiu myšlienkovú analýzu problému, ale iba formálne vykonanie naučenej postupnosti úkonov.
  - V časti Informácie a komunikácia nie je zahrnutá problematika interaktívnej komunikácie (čet, videokonferencie), v rámci neinteraktívnej komunikácie je potrebné sa venovať aj problematike tvorby a používania knihy adries a filtrov a tiež používaniu diskusných fór. Dôležité je, aby sa študenti zoznámili aj so základnými funkciami e-learningového prostredia.
- Po formálnej stránke sme sa stretli s nasledujúcimi obmedzeniami:
  - Zadania úloh sú príliš tematicky viazané na jednu aplikáciu a postrádame úlohy, ktoré by využívali viaceré aplikácie súčasne (napr. "Nájdite údaje na webe a použite ich v textovom editore alebo tabuľkovom kalkulátore"), čo je v praxi bežná činnosť.
  - Niektoré teoretické otázky sú formulované nejednoznačne a možnosti odpovedí na ne sú diskutabilné (ide pravdepodobne o nevhodnú formuláciu alebo nepresný preklad z originálu).
  - Zadania úloh sú formulované tak, aby sa ľahko hodnotili, ale sú nevhodné pre použitie v školstve, kde je vhodné zadanie doplniť obrázkom výsledného produktu činnosti, prípadne nechať študentom priestor na rozvinutie ich vlastnej tvorivosti. Mnoho chýb študenti robili práve kvôli tomu, že nemali v zadaní obrázkovú predlohu, prípadne chceli pristupovať k zadaniu tvorivým spôsobom.
  - Zadania sú miestami málo názorné a príliš náročné na pozorné čítanie. Mnohých chýb sa študenti dopúšťali z nepozornosti a únavy, čo pôsobilo demotivujúco.

6. CERTIFIKÁCIA INFORMAČNEJ GRAMOTNOSTI PRE UČITEĽOV EPICT

Systém vzdelávania EPICT, ktorý prezentujeme v tejto časti (najmä na základe textov z ich webového sídla), je zrejme jedným z najprepracovanejších a je priamo určený učiteľom. Preto je potrebné sa ním zaoberať, aj keď v súčasnosti neexistuje jeho slovenská lokalizácia. Koncepciu EPICT prijali za svoju stratégiu

profesionálneho rozvoja učiteľov v mnohých krajinách Európy, Afriky a Austrálie – napr. Dánsko, Nórsko, Grécko, Taliansko, Island, Maďarsko, Tasmánia, Anglicko, Ghana, Uganda či Kamerun.

EPICT (po slovensky by sme mohli povedať Európska pedagogická licencia pre informačnú gramotnosť) je alternatívny vzdelávací a certifikačný program, podobný ECDL, špeciálne navrhnutý pre potreby učiteľov a ďalších odborníkov pôsobiacich vo vzdelávaní.

Tým, že tento prístup kladie dôraz na pedagogiku, zasahuje priamo do centra učenia a učenia sa pomocou IKT. Podobne ako v ECDL, aj v rámci tohto systému je problematika rozdelená do modulov, a to do štyroch základných (povinných) a dvanástich voliteľných. Základné moduly pokrývajú kľúčové oblasti IKT, voliteľné moduly zasa dovoľujú používateľovi špecializovať sa a zamerať svoje učenie sa na oblasti ako sú hry, žiaci so špeciálnymi potrebami, interaktívne tabule, či virtuálne prostredia na vzdelávanie.

Základné moduly:

- Vyhľadávanie a využívanie on-line zdrojov.
- Používanie a tvorba interaktívnych materiálov pre učenie sa.
- Elektronická komunikácia a kooperácia.
- IKT a strategické inovácie .

Voliteľné moduly:

- Gramotnosť a IKT.
- Matematická gramotnosť a IKT.
- Prezentačné technológie, interaktívne tabule a interaktívnosť.
- Publikovanie na webe.
- IKT a vzdelávanie žiakov so zdravotným postihnutím.
- Efektívna práca s virtuálnymi prostrediami pre učenie sa.
- Digitálne obrázky.
- Hry, zábava a vzdelávanie.
- Modelovanie pomocou tabuliek.
- IKT a nové metódy práce.
- IKT a tvorivosť.
- Hodnotenie v kontexte TEL.

Na získanie pedagogickej počítačovej licencie potrebuje uchádzač úspešne absolvovať osem modulov – všetky 4 základné. a 4 voliteľné podľa vlastného výberu.

7. ZÁVER

Záverom si dovolíme porovnať počítačovú licenciu ECDL a pedagogickú počítačovú licenciu EPICT.

Počítačová licencia ECDL je v súčasnosti na Slovensku a v rámci Európy pomerne rozšírený a uznávaný certifikačný štandard. Pre učiteľov však požadované znalosti v rámci nej nie sú dostatočné, pretože sa nevenuje pozornosť využitiu informačných technológií vo vzdelávaní. Z tohto hľadiska je pedagogická licencia EPICT oveľa vhodnejšia. Na druhej strane však certifikácia ECDL má na Slovensku vybudovanú infraštruktúru vzdelávacích a testovacích centier a je uznávaná u zamestnávateľov. Pedagogická licencia EPICT nie je na Slovensku lokalizovaná a zatiaľ nie je v sieti zamestnávateľov pedagogických pracovníkov známa a požadovaná.

	Výhody	Nevýhody
ECDL	Je to univerzálne uznávaný štandard pre certifikáciu počítačovej gramotnosti. V rámci Európy a na Slovensku má vytvorenú sieť testovacích centier.	Certifikovaná počítačová gramotnosť nie je pre učiteľov dostatočná.
EPICT	Okrem základnej počítačovej gramotnosti, pokrýva aj zručnosti potrebné pre využitie IKT vo vzdelávaní – pre učiteľov predstavuje komplexnejšiu kvalifikáciu.	Je určený iba pre učiteľov. Nie je natoľko známy a uznávaný medzi zamestnávateľmi. V rámci Európy nemá v mnohých krajinách (ani na Slovensku) vytvorenú sieť vzdelávacích a testovacích centier.

Tab. 3 Porovnanie počítačovej licencie ECDL a pedagogickej počítačovej licencie EPICT

Z vyššie uvedeného vyplýva, že pre študentov učiteľského štúdia a učiteľov z praxe je v súčasnosti výhodnejšie uchádzať sa o počítačovú licenciu ECDL. Pre inštitúcie vzdelávajúce pedagogických pracovníkov je však nevyhnutné, aby sa inšpirovali pri príprave učiteľov aj požiadavkami kladenými na absolventov kurzov EPICT.

LITERATÚRA

[1] HRUŠECKÁ, A et. al..: Vhodnosť použitia testov ECDL v príprave učiteľov ZŠ a SŠ. In Štúdia vypracovaná v rámci aktivity A5 projektu ESF 13120120275 (JPD 3 2005/NP1-045) MOVYP. Bratislava, FMFI UK, 2008, 43 s.

[2] KALAŠ, I. et al.: Rozvoj komplexnej digitálnej gramotnosti budúcich učiteľov. Záverečná správa aktivity 2.1 projektu ESF 13120120161 (JPD 3 2005/1-012): MIŠŠ 21 - Moderná Informatizácia Štúdia a Školy pre 21. storočie, KZVI FMFI UK, Bratislava, január 2007, 88 s.

[3] ECDL Foundation website <www.ecdl.com>

[4] eEPICT – The European Pedagogical ICT Licence < www.epict.org/>

[5] PEKÁROVÁ, J.: Pilotáž modifikovaného kurzu IKT2. vypracovaná v rámci aktivity A5 projektu ESF 13120120275 (JPD 3 2005/NP1-045) MOVYP. Bratislava, FMFI UK, 2008, 6 s.

KOGNITÍVNY TÚTOR PRE ZAČIATOČNÍKOV V PROGRAMOVANÍ

MGR. MARTINA KABÁTOVÁ

Katedra základov a vyučovania informatiky, Fakulta matematiky, fyziky a informatiky,
Univerzita Komenského, Mlynská dolina, 842 48 Bratislava,
tel. 0907/963 946, e-mail: kabatova@fmph.uniba.sk

Tvorba kognitívnych tútorov pre vyučovanie stavia na poznatkoch kognitívnej vedy a využíva moderné digitálne technológie. Tútoringový program predstavuje edukačnú pomôcku s potenciálom individualizovať a zefektívniť vyučovanie. Zároveň dokáže získavať cenné dáta o postupe študentov a ich spôsobe práce s výučbovým programom. Pomocou nástrojov CTAT (Cognitive Tutor Authoring Tools) sme vytvorili sériu tútorovaných úloh pre začiatočníkov v programovaní a skúmali sme, ako budú študenti na takýto nástroj reagovať, ako ovplyvní vyučovanie a zvažujeme, ako do budúcnosti tútoring ďalej využívať.

Kľúčové slová: *didaktika programovania, digitálne technológie, kognitívny tútor, ACT-R, CTAT*

1. ÚVOD – KOGNITÍVNY TÚTOR

Kognitívny tútor je moderná interaktívna digitálna pomôcka pre podporu procesu učenia sa. Využíva poznatky z kognitívnej vedy zastúpenej najmä kognitívnou psychológiou a modelovaním mysle pomocou počítačov. Je to počítačová aplikácia, ktorá pri riešení zadávaných úloh poskytuje študentovi okamžitú odozvu – ohlasuje chyby a vysvetľuje ich, poskytuje na vyžiadanie nápovede a kontroluje správnosť riešenia.

Prvé tútoringové programy vznikli ako pomocné aplikácie teórie ACT-R (Adaptive Control of Thaught - Rational), aby lepšie pomohli pochopiť niektoré aspekty ľudského myslenia a učenia sa (v zmysle nadobúdania procedurálnych zručností). Prvou oblasťou, ktorá bola pre kognitívneho tútora spracovaná, bola matematika – tútor riešil lineárne rovnice. ACT-R je teória, ktorá ašpiruje na vysvetlenie ľudskej kognície pomocou modelovania umelého systému, ktorý bude vykazovať rovnaké schopnosti ako ľudská myseľ. Základné ***princípy teórie ACT-R*** sú [3]:

- existujú dva typy pamäti – deklaratívna a procedurálna, v deklaratívnej pamäti sú uložené vedomosti (jednotlivé fakty, napríklad – pes je cicavec), procedurálna pamäť obsahuje postupy ako niečo urobiť (napríklad ako sčítať dve veľké čísla),
- popis interakcie medzi oboma typmi pamäti a popis sprístupnenia deklaratívnych poznatkov procedurálnym zručnostiam sú kľúčové pre vysvetlenie kognitívnych prejavov a myslenia,
- existujú mechanizmy, ktoré umožňujú vznik nových procedurálnych zručností (pomocou imitácie, analógie, interpretácie) ,
- stratégie riešenia problémov sú zakódované ako pravidlá (typu „AK stav_y, TAK akcia_x“),
- procedurálne zručnosti a deklaratívne poznatky sa upevňujú používaním.

Z týchto princípov priamo vychádza koncepcia učenia sa, ktorá stavia na tom, že viaceré domény (matematika, programovanie aj iné), sa dajú popísať ako súbor pravidiel. Napríklad úspešné vyriešenie lineárnej rovnice predstavuje použitie viacerých pravidiel („AK sa neznáma vyskytuje v tvare n.x, TAK vynásob obe strany rovnice číslom 1/n“ a veľa iných). ***Učenie sa*** potom spočíva

- v zapamätaní si faktov a princípov, ktoré figurujú v pravidlách (tvorba a upevňovanie obsahu deklaratívnej pamäti),
- v trénovaní použitia týchto pravidiel (upevňovanie obsahu procedurálnej pamäti) a
- v tvorbe nových súvisiacich pravidiel (tvorba obsahu procedurálnej pamäti).

Kognitívne tútoringové programy sa zameriavajú na upevňovanie a tvorbu obsahu procedurálnej pamäti, pretože predpokladajú, že deklaratívne poznatky sú získané inou cestou a v ich zápise do pamäti nie je žiaden principiálny problém. Naproti tomu procedurálne zručnosti je nutné prakticky precvičovať, nestačí o nich hovoriť [1]. V tomto bode sa prístup kognitívneho tútoringu zhoduje s papertovským konštrukcionizmom.

Pre ideálny postup je dobré dodržať tieto dve pravidlá: 1. dôkladne analyzovať a rozčleniť danú oblasť na jednotlivé komponenty (pravidlá), ktoré reprezentujú zručnosti, 2. zabezpečiť, aby študent dobre zvládol jednoduchšie zručnosti pred tým, než prejde k zložitejším [2].

Častá námietka znie, že učenie sa nejde rozdeliť na takto izolované komponenty, pretože je to zložitý proces, kde sa nedá celok brať ako súčet častí. Ak by sa takto postupovalo, učili by sme izolované zručnosti, čo nezodpovedá princípu učenia sa v kontexte. Kognitívny tútoring však nemá za cieľ vysvetľovať celý kontext vzdelávania v danej oblasti, poskytuje iba prostriedky pre tréning dobre definovaných malých úloh. Učenie sa využitiu týchto zručností v širšom kontexte by nemalo byť zrejme realizované pomocou tútora, ale inou formou

vyučovania. Treba priznať, že niektoré domény môžu byť svojou podstatou nevhodné pre akékoľvek využitie tútorov.

2. CTAT TÚTOR PRE DELPHI

Pre tvorbu tútora pre začiatočníkov v programovaní sme sa rozhodli využiť nástroje CTAT* (Cognitive Tutor Authoring Tools) vyvinuté na pracovisku LearnLab na univerzite Carnegie Mellon v USA. Hlavnou motiváciou pre vytvorenie takýchto nástrojov bola požiadavka na zníženie časovej náročnosti tvorby tútora. Zároveň sa otvorila možnosť pripraviť tútora pre vyučovanie aj bez hlbších znalostí programovania, čo umožňuje vytvárať tútoringové aplikácie širšej skupine ľudí (v ideálnom prípade celému spektru učiteľov na ZŠ, SŠ a VŠ).

Pomocou CTAT sa dajú vytvoriť ***dva typy tútorov***:

- tútor trasujúci príklady (example tracing tutor) – jeho príprava je jednoduchšia, ale treba definovať všetky správne aj nesprávne postupy pri riešení pre každú úlohu zvlášť,
- pravý kognitívny tútor – vyžaduje programovanie v jazyku JESS, kde je nutné popísať všeobecné pravidlá riešenia úloh jedného typu.

Vytvorili sme tútorov (trasujúcich príklady) pre sériu deviatich úloh zameraných na deklaráciu premennej a priradenie do premennej v jazyku pascal. Tvorba tútora pozostávala z troch krokov:

- ***Príprava užívateľského rozhrania***

Užívateľské rozhranie sme vytvorili v programe Flash. Využili sme predpripravené komponenty nevyhnutné pre fungovanie tútora, ktoré komunikujú s jeho skrytými mechanizmami. Okrem týchto komponentov sme považovali za vhodné doplniť signalizáciu splnenia úlohy a prehľad o vyriešených úlohách (Obr. 1).

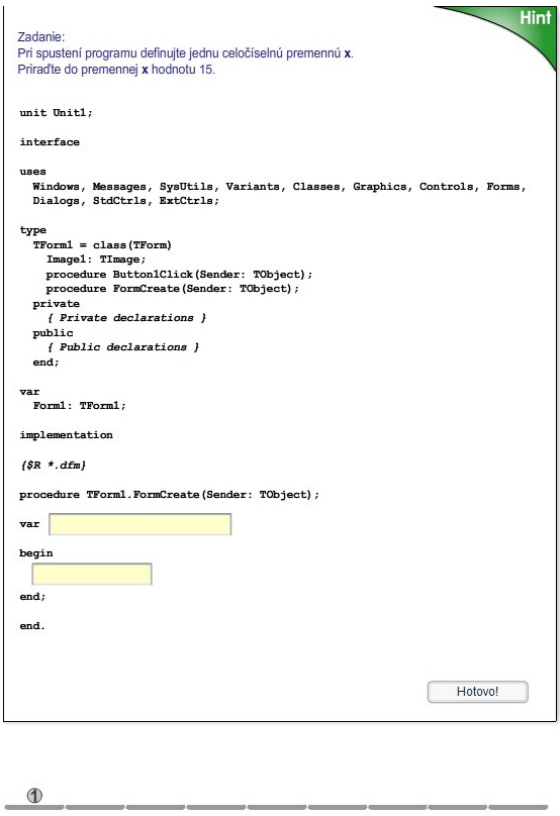
- ***Naplnenie grafu riešení***

Graf riešení sa vytvára demonštráciou pomocou editora grafov, ktorý je súčasťou CTAT. V hotovom užívateľskom rozhraní (ktoré sa dá spustiť v ľubovoľnom internetovom prehliadači) sa predvádzajú správne aj nesprávne riešenia, ktoré sa zaznamenávajú do grafu. Predvádzať riešenie v tomto prípade znamená vpisovať do políčok vstupy, vyberať z ponuky, stláčať tlačidlá a podobne. Priamo v zobrazenom grafe sa potom dopĺňajú aj texty nápovedí a texty chybových hlásení. Okrem toho je možné pomenovať jednotlivé kroky riešenia pre sprehľadnenie záznamu o interakcii študenta s tútorom

- ***Sprístupnenie hotového tútora***

Pre zobrazovanie tútora cez internet a jeho funkčnosť treba umiestniť na server súbory Flash, ktoré predstavujú užívateľské rozhranie, HTML súbory, ktoré zobrazia Flash súbory, textové súbory, v ktorých je zakódovaný graf riešení a podpornú aplikáciu, ktorá bude po celý čas na serveri spustená a bude zabezpečovať spoluprácu všetkých ostatných súborov. Potom je možné spúšťať tútora z akéhokoľvek počítača pripojeného na internet.

V našej prvej sérií sme pripravili dva typy úloh: 1. študent má vpisovať do prázdnych políčok časti programu, 2. študent vyberá z niekoľkých možností správnu odpoveď.



Obr. 1 Užívateľské rozhranie pre úlohu č.1

Treba poznamenať, že vlastnosťou tútora vytvoreného pomocou CTAT je to, že políčka sa nedajú vypĺňať inak než postupne. Tútor nedokáže skontrolovať iba čiastočne správne riešenie, pretože riešenie je zakódované v stromovom grafe, v ktorom sa dá pohybovať po uzloch iba zhora nadol - tútor nepodporuje čiastočnú kontrolu niektorého podgrafu. Táto vlastnosť nie je ideálna z hľadiska didaktiky programovania, pretože niektoré typy úloh sa lepšie riešia nie riadok po riadku ale postupným vylepšovaním spočiatku „nesprávneho“ riešenia. Toto spolu s inými faktormi znemožňuje použiť takýto typ tútora pre zložitejšie úlohy z programovania. Treba si však uvedomiť, že to nie je všeobecná vlastnosť všetkých kognitívnych tútorov, naopak príklady z praxe dokazujú, že je možné vytvoriť veľmi kvalitný tútoringový program, ktorý zohľadní všetky požiadavky didaktiky programovania. CTAT predstavujú zjednodušenie

* Podrobnosti nájdete na stránke: http://ctat.pact.cs.cmu.edu

pre menej náročného tvorcu, ktoré ide na úkor možností výsledného produktu, ale má iné výhody.

3. DÁTA ZÍSKANÉ PRI TESTOVANÍ TÚTORA

Túto vytvorený pomocou CTAT dokáže zaznamenať všetky vstupy od študentov a všetky interakcie s prvkami ovládania. Tieto údaje spolu s časovou stopou a inými dátami sa ukladajú do textových súborov. Pre vytváranie korektného záznamu o práci študenta s túto rom je potrebné zabezpečiť, aby sa študenti mohli pred spustením identifikovať (potom je však nutné riešiť otázku spracovania osobných údajov), tým vieme priradiť jednému užívateľovi záznamy z jednotlivých úloh, ktoré riešil, a spracovať tak údaje o jeho postupe v celej sérii. Ak nepotrebujeme vedieť, ktoré dáta patria ktorému konkrétnemu študentovi, stačí, aby nejaký skript generoval unikátne identifikačné číslo pre každého, kto túto spustí. Takto získané záznamy predstavujú cenný zdroj informácií o tom, ako študenti riešia zadané úlohy, ako interagujú s prvkami túto ra, kedy otvárajú nápovede a podobne. Pre analýzu a spracovanie dát z túto rov existuje aplikácia DataShop, ktorú plánujeme v budúcnosti využiť.

Prvú sériu úloh zameranú na deklaráciu a priradenie do premennej sme testovali so skupinou študentov v prvom ročníku bakalárskeho štúdia učiteľstva všeobecnovzdelávacích predmetov v kombinácií s informatikou. Máme skúsenosti, že študenti týchto odborov prichádzajú často na univerzitu bez dostatočných skúsenosti s programovaním, veľa z nich sa s programovaním nikdy predtým nestretlo. S túto rom sme pracovali na nepovinnom cvičení za počítačmi, ktoré prebehlo v druhom týždni semestra a študenti teda mali za sebou minimálne jednu prednášku z tohto predmetu.

Počas pozorovania práce študentov s túto rom spolu s ďalšími troma vyučujúcimi sme si všimli niekoľko javov:

- Pre prvákov je mätúce ďalšie nové prostredie, ktoré sa musia naučiť ovládať a využívať. Ostáva na zváženie, či bude prínos túto ra taký, aby túto nevýhodu vyvážil.
- Pre vyučujúceho je to ďalšia záťaž - tiež sa musí naučiť s túto rom pracovať, poznať jeho potenciál a obmedzenia.
- Niektoré technické problémy (túto r prestane kontrolovať správnosť vstupov, nutnosť vypĺňať bunky postupne) a problémy s ovládaním programu (nemožnosť presúvať sa medzi políčkami iba pomocou klávesnice) znemožňujú jeho efektívne použitie.
- Študenti akoby nechceli veriť, že môžu používať nápovede v rámci túto ra aj napriek tomu, že na možnosť nápovede bolo upozornené vopred . Radšej sa pýtajú vyučujúcich, spolusediacich alebo študujú svoje poznámky z prednášky.
- Veľa študentov vyjadrilo spokojnosť nad vyriešenou úlohou (dobré riešenie a splnenie úlohy je signalizované). Zrejme je dôležité, aby študenti vedeli, že majú správne riešenie a v akom štádiu riešenia celej série úloh sa nachádzajú. Považujeme to za dôležitý motivačný faktor zodpovedajúci filozofii „mastery learning“.

Ako študenti pracovali na úlohách zadávaných túto rom, všetky vstupy a akcie sa zaznamenávali do anonymných súborov (v tomto prípade sme nepoužili žiadnu identifikáciu študentov v túto rovi). Zo záznamových súborov sme zistili množinu častých nesprávnych riešení pre jednotlivé úlohy, tieto zapracujeme do novej verzie túto ra a doplníme o príslušné chybové hlásenia. Okrem toho sme mali možnosť všímať si niektoré špecifické prístupy k riešeniu úloh v túto rovi a rôzne vnímanie ovládacích prvkov. Postupy pri riešení úloh by sa dali rozdeliť na „dobré“ a „zlé“ - dobré vedú pomerne rýchlo k splneniu celej úlohy, zlé priveľmi zvyšujú počet krokov, alebo nevedú k vyriešeniu úlohy (keďže všetci študenti prešli celou sériou úloh, je zrejmé, že niektoré úlohy boli spustené jedným študentom viackrát, vo finále však úloha bola vždy vyriešená). Za „dobré“ postupy považujeme:

- **metóda „pokus-omyl“** - študent väčšinou nevie ihneď správne riešenie (typicky chýba bodkočiarka za príkazom), ale na niekoľko málo pokusov sa k nemu dopracuje, jeho chyba je veľakrát taká, že ju túto r rozozná a poskytne chybové hlásenie,
- **metóda „najprv nápoveď“** - bez toho, aby sa študent pokúšal o vlastné riešenie, vyžiada si hneď nápoveď, podľa ktorej spraví väčšinou správne riešenie, táto metóda je použitá iba zriedkavo,
- **metóda „nápoveď neskôr“** - po niekoľkých nesprávnych pokusoch si študent vyžiada nápoveď a podľa nej napíše správne riešenie, môže ísť o situáciu, keď študent vzdá metódu „pokus-omyl“ bez nápovede,

Použitie „zlého“ postupu môže byť zapríčinené nesprávnym pochopením ovládacích prvkov túto ra, nepozornosťou alebo ignorovaním chybových hlásení. Za „zlé“ prístupy k riešeniu považujeme:

- **používanie tlačidla „HOTOVO“ ako kontroly správnosti** – veľa študentov stláčalo tlačidlo (ktoré slúži na ukončenie úlohy a presun na ďalšiu úlohu) napriek tomu, že túto r im v chybovej správe oznamoval, že ešte nemajú vyplnené všetky bunky, alebo že majú chybu v niektorom z políčok. Tlačidlo „HOTOVO“ takúto funkciu nemá a túto r v skutočnosti funguje tak, že kontroluje obsah daného políčka hneď ako sa dopíše vstup a označí sa myšou iné políčko. Je vhodné zvážiť, či by nemalo byť doplnené nejaké ďalšie tlačidlo (možno aj pre každú bunku zvlášť), ktoré by spĺňalo takúto funkciu. Iné vysvetlenie je, že chybné vstupy sú nedostatočne

označené, alebo že chybové hlásenia sú málo výrazné či nejasné.

- **Preskakovanie políčok** - niekoľko študentov sa snažilo vypĺňať prázdne políčka v inom poradí, než bolo určené. Tútor nedokáže takéto riešenie vyhodnotiť, aj keby bolo čiastočne správne. Ak to bude technicky možné, považujeme za dôležité umožniť študentom vypĺňať políčka riešenia v ľubovoľnom poradí.
- **Ignorovanie chyby vo vyššom políčku** – veľa nesprávnych riešení prebiehalo tak, že študent vyplnil všetky políčka, pričom niektoré z nich nesprávne, a označil úlohu za hotovú. Tútor označil prvé nesprávne políčko a požadoval od študenta, aby sa vrátil a opravil riešenie. Mnoho študentov nevedelo, ako pokračovať a často v takejto situácii opakovane žiadali nápoveď, z ktorej však nevyčítali, kde sa stala chyba. Takýto postup je v rámci práce s tútorom (prihliadnuc na jeho obmedzenia) veľmi nevhodný, pokúsime sa to vyriešiť lepšou signalizáciou správnosti/nesprávnosti vstupu. Možno by bolo dobré, aby neboli aktívne tie políčka, ktoré tútor aktuálne nedokáže skontrolovať. Tým by sme však prišli o záznamy študentov, ktorí majú snahu takto úlohu riešiť, čo zatiaľ považujeme za hodnotnú informáciu.

Veríme, že viacero problémov by sa odstránilo, keby mali študenti možnosť pracovať s tútorom opakovane a zvyknúť si na niektoré špecifiká ovládania a funkčnosti. Našim cieľom však je, aby ovládanie bolo čo najintuitívnejšie a funkčnosť pochopiteľná pre začiatočníkov v programovaní.

4. NAVRHOVANÉ ZMENY A VYLEPŠENIA

Prednášky k predmetu - úprava a nový koncept

Pre efektívne využitie tútorov v kurze programovania by bolo vhodné upraviť texty prednášok a zadania povinných a nepovinných cvičení, tak aby spĺňali požiadavky prístupu k učeniu sa podľa teórie ACT-R. Predovšetkým by sa malo zabezpečiť, aby materiál k prednáške pokrýval na 100 % obsah cvičení a domácich úloh. Ďalším z najdôležitejších bodov bude upraviť cvičenia tak, aby boli úlohy vždy stupňované od najjednoduchších po najzložitejšie.

Úvodný kurz k programovaniu

Po zvážení situácie s novými študentami a v rámci tvorby tútorov pre vyučovanie programovania sme sa rozhodli pripraviť úvodný kurz programovania pre úplných začiatočníkov, ktorý by bol mimo semestra. Domnievame sa, že je to najvhodnejšie miesto pre využitie tútorov. Takýto kurz by pomohol zjednotiť úroveň študentov v prvom ročníku a tým umožnil ich kvalitnejšie vzdelávanie počas semestra.

Identifikácia študentov pre zber dát z tútora
Aby sme dokázali vyťažiť maximum z dát, ktoré tútor zbiera a následne ich dokázali analyzovať pomocou aplikácie na to určenej, potrebujeme zabezpečiť identifikáciu jednotlivých študentov pred spustením tútora (pomocou zadania mena alebo kódu).

Úpravy prvej série úloh:

- doplnenie rozpoznávaných chýb a príslušných chybových hlásení podľa chybných vstupov získaných z prvého testovania,
- zvýraznenie správnosti/nesprávnosti vstupu,
- preklad chybových hlásení, ktoré sú v anglickom jazyku,
- oprava ovládania pomocou klávesnice,
- kontrola čiastočných riešení – ak bude možné upraviť nástroje na tvorbu tútorov, tak aby to podporovali.

5. ZÁVER

Vyskúšali sme si tvorbu série tútoringových programov pre vyučovanie. Aj keď nástroje na tvorbu tútorov sú pomerne obmedzené a vyžadujú od tvorcu viacero kompromisov, domnievame sa, že majú veľký potenciál v mnohých doménach vzdelávania. Digitálne technológie, ktoré môžu obohatiť vyučovanie je potrebné naďalej skúmať, pretože dávajú možnosť individualizovať vyučovanie a zároveň skúmať pravidelnosti v javoch, ktoré by nám inak ostali ukryté.

LITERATÚRA

[1] Anderson, J.R., Corbett, A.T., Koedinger, K.R., Pelletier, R.: Cognitive Tutors: Lessons Learned. In *The Journal of the Learning Sciences.* 1995, 4(2), s. 167 - 207.

[2] Corbett, A.T., Anderson, J.R.: Knowlege Tracing: Modeling the Aquisition of Procedural Knowledge. In *User Modeling and User-Adapted Interaction.* 1995, 4, s. 253 - 278.

[3] Anderson, J.R.: How Can the Human Mind Occur in the Physical Universe? New York : Oxford University Press, 2007. ISBN 978-0-19-532425-9

[4] Anderson, J.R., Gluck, K.: What role do cognitive architectures play in intelligent tutoring systems? In Klahr, D., Carver, S.M. (eds.): Cognition & Instruction: Twenty-five years of progress. Mahwah, NJ : Erlbaum, 2001, s. 227 – 262. ISBN 0-89859-689-0

[5] Koedinger, K.R., Aleven, V., Heffernan, N., McLaren, B., Hockenberry, M.: Opening the Door to Non-programmers: Authoring Intelligent Tutor Behavior by Demonstration. In *Proceedings of 7th Annual Intelligent Tutoring Systems Conference*. Maceio, Brazil, 2004.

MOŽNOSTI VYUŽITIA ALTERNATÍVNYCH NÁSTROJOV GIS PRI VYUČOVANÍ GEOGRAFIE A EKOLÓGIE

RNDR. JOZEF KRNÁČ

Katedra informatiky, Fakulta prírodných vied, Universita Mateja Bela,
Tajovského 40, Banská Bystrica, Slovensko, tel. ++421 48/602 3175, e-mail: krnac@fpv.umb.sk

ABSTRACT

Možnosti využitia alternatívnych nástrojov geoinformatiky, ako sú QGIS, Grass pri vyučovaní a modelovaní priestorových procesov a javov v odbore geografia a ekológia. Jeho možnosti sú ohraničené výkonom hardvérovej časti a vedomostnými schopnosťami používateľa. Reálne interpretujú priestorové javy a simulujú procesy v krajinne. Umožňujú učiteľovi na praktických príkladoch ukázať študentovi preberanú metodiku a priestorový celok so všetkými aktivitami ktoré v ňom v časovom horizonte aktuálne na základe vstupných údajov prebiehajú.

Keywords: *Vyučovanie, Geografia, Ekológia, GIS, Priestor, Jav, modelovanie, nástroje,*

1. ÚVOD

Geografické informačné systémy čoraz viac prenikajú do teoretických i aplikačných vedných disciplín. Z nich sú najdôležitejšie tie, ktoré riešia priestorové modelovanie v prepojení na databázové informačné systémy.

Na fakulte prírodných vied Univerzity Mateja Bela v Banskej Bystrici sa podporné nástroje geoinformatiky vyučujú na piatich katedrách (Katedra informatiky, Katedra biológie a ekológie, Katedra geografie a krajinnej ekológie, Katedra chémie a Katedra environmentálneho manažérstva). Vo väčšine prípadov sa na výučbu používajú komerčné nástroje ako sú Idrisi od Clark Labs a GeoMedia od Intergrph. Už určitú dobu sa pokúšame vo vyučovacom procese otvoriť možnosť využitia aj otvorených softvérových produktov, ktoré by ponúkali študentom aj prednášajúcim rovnaké možnosti a nástroje na spracovanie zadanej problematiky a zároveň aplikáciu pri vyučovaní ako komerčné nástroje.

2. NÁSTROJE GIS

Široké spektrum možností využitia GIS viedlo ku vzniku viacerých rozdielnych prístupov v jeho chápaní a nejednoznačnosti jeho definície. I keď existuje viacero definícií pre pojem GIS, všetky sa zhodujú v tom, že je to *informačný systém, ktorý pracuje s geografickými informáciami.*

Existuje však i množstvo ďalších definícií pre GIS:
ARONOFF (1989) nazýva GIS-om akýkoľvek súbor manuálnych, alebo počítačových procedúr používaných na uskladnenie a manipuláciu geograficky definovaných údajov.
BURROUGH (1986) vníma GIS ako súbor prostriedkov pre zber, ukladanie, vyhľadávanie, transformáciu, analyzovanie a zobrazovanie priestorových údajov reálneho sveta z hľadiska:

- ich polohy vzhľadom k definovanému súradnicovému systému,
- ich popisných (atribútových) vlastností,
- ich priestorových vzťahov k iným objektom,
- ich topológie.

Z pohľadu teórie systémov a systémovej analýzy geografickej sféry, resp. krajiny ako jej výrezu znie definícia podľa (KRCHO, MIČIETOVÁ, 1989): *Geografický informačný systém predstavuje technicko-programový aparát na zber, uchovávanie údajov o krajine a odvodenie informácií o nej tak, aby boli zabezpečené riadiace funkcie v systéme riadenia výskumu a využitia krajiny.*

GIS je podľa (Mičietová, 1999) funkčným prvkom, ktorý zabezpečuje:

- údaje o objekte a subjekte riadenia,
- technológie na spracovanie týchto údajov,
- metodiky na spracovanie základných údajov a odvodenie nových informácií,
- modelovacie nástroje, ktoré vyžadujú metodiky na odvodenie informácií,
- distribúciu informácií v procese riadenia.

V dôsledku toho sa pri budovaní geografických informačných systémov aj v odbore systémová ekológia venuje veľká pozornosť analýze vlastností geografickej krajiny ako objektu riadenia, ale aj riešeným úlohám, ktoré generuje subsystém riadiaci na základe zvolených stratégií riadenia. Koncipovanie geografických informačných systémov vychádza preto z analýzy vlastností geografickej krajiny ako objektu riadenia.

Informačný systém zohľadňujúci väzby (vzťahy, závislosti, súvislosti,..) medzi jednotlivými

prvkami plní požiadavky geoinformačného systému geografického typu, systém s riadenou bázou dát (SRBD), ktorá uchováva a spracúva údaje tak o jednotlivých zložkách krajiny, ako aj o krajine ako priestorovo vysoko organizovanom celku. Zároveň dbá na priestorové vzťahy, ako z horizontálneho, tak z vertikálneho pohľadu (Krcho, Mičietová, 1989).

V informačných systémoch vytvárame väzby definované ako informácie a prvky ako miesta transformácie (spracovania) informácie. Tieto miesta transformácie možno v zmysle (Mičietová, 2001) rozdeliť podľa typu do štyroch úrovní:

- úroveň zberu, prenosu a prvotného spracovania údajov, generovanie a formálne zjednotenie základných informácií,
- úroveň jednotného uloženia, logického a technického zjednotenia základných informácií a zabezpečenie komplexného a integrovaného prístupu k nim,
- úroveň spracovania základných a generovanie nových (odvodených) informácií, ako podklad k rozhodovaniu v procese riadenia,
- úroveň distribúcie informácií v systéme riadenia.

Na exaktné vyjadrenie georeliéfu, resp. jeho geometrických vlastností sa v technických, ale aj iných vedných disciplínách používame digitálne modely, pre ktoré sa všeobecne vžil názov digitálne modely terénu (Digital Terrain Models – DTM). Komplexný digitálny model reliéfu je modelujúci systém krajinného reliéfu, uvažovaného ako subsystému. Pomocou KDMR možno však modelovať nielen krajinný reliéf, ale akékoľvek druhy abstraktných priestorových plôch. Morfometrická analýza, realizovaná prostredníctvom KDMR nadobúda tak širší všeobecný význam.

2.1. Oblasť využitia nástrojov GIS

V geografii a podobne aj v ekológii sa študenti pod vedením učiteľov zameriavajú na priestorovú orientáciu v rámci určitého priestorového celku v závislosti na jeho hraniciach (prírodných, socioekonomických, ekonomických atď.). Nástroj GIS im má pomôcť a umožniť sledovať priestorové javy vyvíjajúce sa na základe procesov prebiehajúcich v priestore a čase na vybranom územnom celku, poprípade sledovať určitú dynamickú zmenu vývoja skúmanej veličiny a jej vplyv na krajinu ako komplex. Jedným z takýchto nástrojov je aj nástroj GRASS, ktorý je plnohodnotným modulovým systémom na tvorbu a spracovanie údajov v určitých modulových hladinách v závislosti od použitej údajovej hladiny. Jeho využitie z hľadiska riešenia projektov, semestrálnych prác, bakalárskych prác a diplomových prác je na Fakulte prírodných vied zatiaľ dosť obmedzené. Vyučuje sa ako voliteľný predmet pre študentov Katedry biológie a ekológie v odbore systémová ekológia ako podporný nástroj pre študentov profilujúcich sa v oblasti GIS a Tvorby a monitorovania informačných systémov.

Študenti modelujú pomocou neho morfometrické ukazovatele reliéfu (topické, chorické a topicko - chorické) a ich vplyv na hydrologické, klimatické, populačné procesy a zmeny v priestore. Vytvárajú rôzne analytické mapové diela vstupujúce do zložitejších syntetických vrstiev a následných interpretácií priestoru. Študenti sa zaoberajú v menšej mierke skúmaním javov prebiehajúcich v krajine ako systéme.

Krajina ktorá "je dynamický priestorový systém javov prírodnej a sociálno – ekonomickej povahy, viažuca sa k zemskému povrchu. Má synergické (súčinné), chorické (priestorové) a chronologické (v čase sa vyvíjajúce) aspekty". (MAZÚR, DRDOŠ, URBÁNEK, 1980). Význam pojmu krajina nie je v jej vonkajšom vzhľade, ale v hľadaní vzájomných vzťahov medzi jednotlivými komponentmi a javmi krajiny. *"Geografická krajina je konkrétna časť zemského povrchu vymedzená na základe zvoleného kritéria"*.

3. ALTERNATÍVNE NÁSTROJE GIS

Na rozdiel od známych komerčných GIS nástrojov existujú a v praxi sa využívajú voľne dostupné nástroje na modelovanie geografických, ekologických situácií, ktoré umožňujú používateľovi s rovnakým vstupom dosiahnuť pomerne vysoko profesionálne riešenia bez bolestivých vstupných finančných investícií do operačného systéme a GIS softvérového produktu. Jedným z najkomplexnejších nástrojov je modulový nástroj GRASS.

Grass je geografický informačný systém pre podporu analýz geografických zdrojov. Tento produkt je voľne šíriteľný podľa pravidiel GNU GPL licencie, navrhnutej pre platformu operačného systému UNIX/Linux.

Fig. 1 Logo GRASS

Grass nie je len jednostranne zameraný program, je to komplexný systém pre spracovanie analýzu a vizualizáciu rastrových a vektorových údajov a nástroj pre priestorové modelovanie. Vďaka schopnosti pracovať s rastrovými údajmi sa dá GRASS využiť ako priestorový modelovací systém. Obsahuje viac ako 100 multifunkčných modulov pre

analýzu a sprístupnenie rastrových dát. Modely zrážok a povrchových odtokov, počítania dráh povrchových odtokov a klimatických analýz. Keďže je veľa modulov otvorených, môžu ich používatelia editovať na podľa svojej potreby, podobne môžu vytvárať (doprogramovávať v jazyku C) vlastné špecifické moduly a pomocou nich generovať mapy s požadovanými výstupnými údajmi uloženými v databanke GRASSu.

Okrem štandardnej dvojrozmernej analýzy dovoľuje GRASS spracovať údaje aj vo väčších dimenziách. Rastrové, vektorové a bodové údaje sa dajú použiť pri vizualizácii. Príkladom toho použitia môže byť návrh letiska, analýza krajiny a priestorových náchylností. Vizuálne prostriedky umožňujú animáciu priestorových dát. 3D pohľady takto môžu byť prezentované, ako jednotlivé obrázky alebo ako video MPEG uložené pre ďalšiu prácu.

4. PRAKTICKÉ POUŽITIE

Stručne by som sa chcel dotknúť aspoň základných a najčastejšie využívaných nástrojov, ako sú nástroje na spracovanie vektorových, rastrových a databázových údajov.

4.2. Interpolácia

Jedným z najčastejších použití (pravidelne, alebo nepravidelne rozložených) bodových údajov je interpolácia súvislého povrchu na ohraničenom území. V GIS sú potom tieto povrchy často používané v rôznych modeloch a simuláciách. Asi najznámejším prípadom interpolácie je generovanie digitálneho výškového modelu (DEM) z bodových výškových údajov.

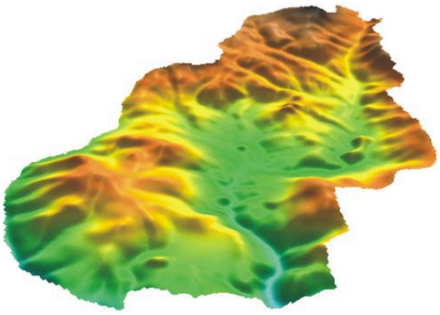


Fig. 2 3D model – s povrchom DEM

4.3. Energia dopadajúceho slnečného žiarenia

Potenciálna dynamika oslnenia reliéfu je podmienená :

- polohou Slnka na nebeskej sfére,
- sklonom reliéfu a orientáciou reliéfu voči svetovým stranám.

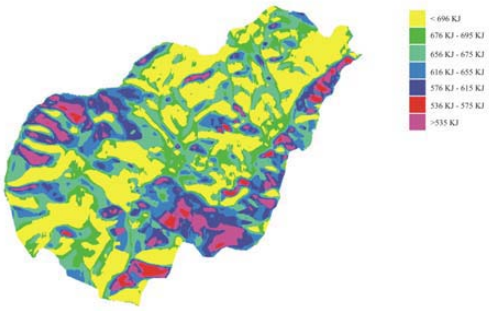


Fig. 3 Množstvo priameho slnečného žiarenia

4.4. Krivosť reliéfu

Priestorovou syntézou horizontálnej a normálovej krivosti boli vytvorené formy reliéfu, ktoré v rozhodujúcej miere ovplyvňujú svahové procesy, ktoré vyjadrujú ako sa na svahu materiál zhromažďuje, rozptyľuje, spomaľuje alebo zrýchľuje.

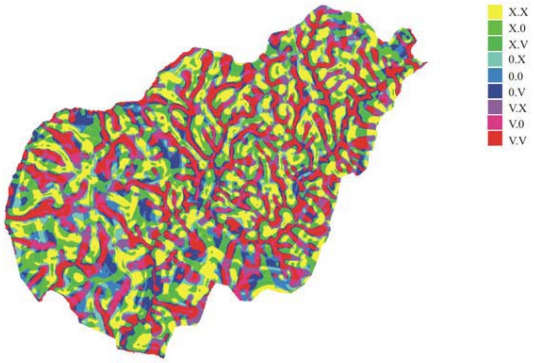


Fig. 4 Krivosť reliéfu – syntéza H. a N.krivosti

4.5. Odtokové pomery

Základnou jednotkou pre odtok vody v koryte vodného toku je prietok – Q, čiže množstvo vody, ktoré pretečie za jednu sekundu priečnym profilom toku prietok sa vyjadruje m3.s-1.

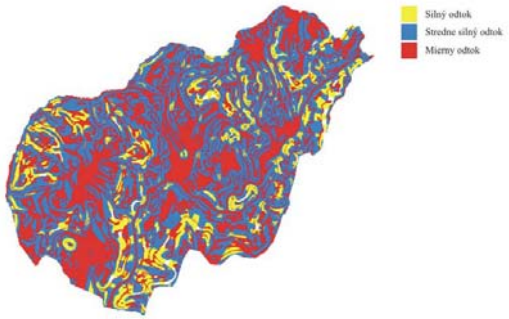


Fig. 5 Povrchový odtok

4.6. Potenciálna erózia

Pre potreby simulácie scenárov riešenia erózie sa vyžaduje poznanie vlastností krajiny, ktoré ovplyvňujú časopriestorové zmeny povrchového

toku a teda aj vstupných parametrov modelu. Na naplnenie databázy ***vstupných parametrov eróznych modelov*** sú potrebné nasledovné skupiny údajov: morfometrické parametre reliéfu, vlastnosti krajinnej pokrývky (využitie krajiny, vlastnosti vegetácie), fyzikálne vlastnosti pôdy, priebeh zrážkovej udalosti.

Podľa rovnice:

erózia= R*K*ls_factor ..... potenciálna erózia

Kde:
R – faktor dažďa [MJ.ha-1.cm.h-1]
K – pôdny faktor
L – faktor dĺžky svahu
S – faktor sklonu svahu

Fig. 6 Potenciálna erózia modelového územia

4.7. Tvorba morfoklimatických pomerov

(vplývajúcich na zmenu rýchlosti vetra v prízemnej vrstve atmosféry)

Prúdenie v prízemnej vrstve atmosféry ovplyvňujú hlavne orografické systémy, polohové vzťahy na reliéfe a rôzne typy prekážok na dráhe vetra, až v detaile poľnohospodárske plodiny a drsnosť pôdy.

S postupom:

- Výber základných polohových vlastností na reliéfe, ktoré výrazne vplývajú na zmenu prúdenia vzduchu.
- Zostavenie mapy typov polohových vzťahov na reliéfe,
- Interpretácia polohových vlastností na reliéfe.

Polohové vzťahy na reliéfe sa teda prejavujú ako rôzne rýchlostné kategórie vetra. Pri vytváraní vychádzame z experimentálnych poznatkov a prehodnotenia ďalších známych zákonitostí , zmeny rýchlosti prúdenia vzduchu v členitom teréne (KURPELOVÁ, KRNÁČ, 1983).

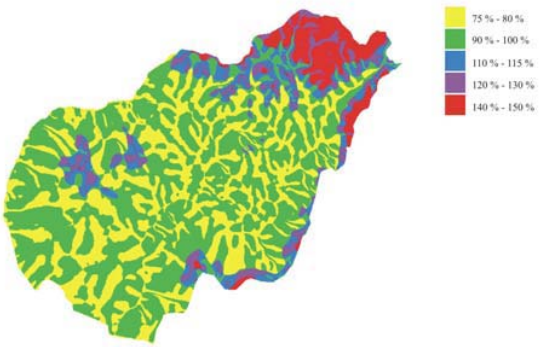


Fig. 7 Rýchlostné kategórie vetra

5. ZÁVER

Predkladaným príspevkom som mal záujem upozorniť a poukázať na potrebnosť vychovávať používateľov nástrojov geoinformatiky. Je to nástroj využiteľný v akomkoľvek vednom odbore na prezentovanie údajov vzťahujúcich sa k zemskému povrchu čo je v súčasnej dobe obrovská výhoda pri plánovaní a využívaní akýchkoľvek procesov v priestore. Je dostupný aj v alternatívne podobe ako open source a keďže okrem toho veľmi názorne prezentuje procesy a javy v priestore preto je nanajvýš vhodný na výučbové účely.

LITERATÚRA

[1] ARONOFF, S.: Geographical Information Systems - A Management Perspective. WDL Publications, Ottawa, 1989
[2] P. A. BURROUGH: Principles of Geographical Information Systems for Land Resources Assessment. 1986, ISBN-10: 0198545924
[3] KRCHO, J.; MIČIETOVÁ, E. (1989): Geoinformačný systém o geografickej sfére a komplexný digitálny model priestorovej štruktúry ako jeho integrálna súčasť. Geografický časopis č.3, s. 249-274.
[4] MIČIETOVÁ, E. (2001): Geografický informačný systém (GIS) – štruktúra, integrita, interoperabilita, implementácia. In: Sborník 14. Kartografická konference, Plzeň.
[5] MIČIETOVÁ, E. (1999): Kvalita, funkcie a operačné možnosti databázy geografického informačného systému. Geografický časopis č.3, s.297-312
[6] KRCHO, J.; MIČIETOVÁ, E. (1989): Geoinformačný systém o geografickej sfére a komplexný digitálny model priestorovej štruktúry ako jeho integrálna súčasť. Geografický časopis č.3, s. 249-274.
[7] MAZÚR, DRDOŠ, URBÁNEK: Krajinná syntéza oblasti Tatranskej Lomnice. VEDA Bratislava 1998

VESELÁ ŠKÔLKA LIENKA.
DVA POHĽADY NA EDUKAČNÝ SOFTVÉR PRE PREDŠKOLÁKOV

ANITA KROMMEROVÁ, JANKA PEKÁROVÁ

Katedra základov a vyučovania informatiky, Fakulta matematiky, fyziky a informatiky Univerzity Komenského, Mlynská dolina, 842 48 Bratislava
e-mail: anita.krommerova@gmail.com, jana.pekarova@fmph.uniba.sk

Rodičia a vychovávatelia predškolských detí sú častokrát konfrontovaní s otázkou prístupu k počítačom a počítačovým hrám. Riešenia tejto otázky sa rôznia, no s počítačom a digitálnymi technológiami sa v súčasnosti stretá veľké množstvo detí. Vzrastá potreba kvalitného pedagogického softvéru, ktorý by deti motivoval k objavovaniu a skúmaniu namiesto konzumácie pripravených scenárov. Aké aspekty by mal spĺňať softvér, ktorý vytvárame pre takúto vekovú kategóriu? Ako zodpovedá kvalita pedagogického softvéru požiadavkám predškolského kurikula?

Kľúčové slová: *edukačný softvér, predškolská príprava, rozpoznávanie geometrických tvarov*

1. ÚVOD

Aj v materských školách sa pomaly objavujú počítače a ďalšie digitálne technológie. V rámci programu KidSmart Early Learning[1] napríklad dodala spoločnosť IBM špeciálny model počítača pre detí do 30 bratislavských škôlok a do 24 škôlok v ďalších oblastiach Slovenska. K používaniu edukačných softvérov vyzýva i štátny vzdelávací program pre predškolskú prípravu [7]. Učiteľky materských škôl riešia otázku kvalitného pedagogického softvéru, ktorý bude napĺňať ciele predškolskej prípravy a zároveň bude prispôsobený potrebám malých používateľov. Materská škola Švantnerova v Bratislave predstavuje deťom program Skicár. Základom ovládania tohto programu sú totiž grafické symboly, ikony jednotlivých nástrojov, takže si ich môže osvojiť aj dieťa, ktoré ešte nevie čítať.

V materskej škole si dieťa rozvíja mnohé kompetencie potrebné pre jeho budúci úspešný život. Formuje si prvé matematické predstavy, porozumenie číslam a tvarom. Podporiť rozlišovanie a chápanie vlastností základných geometrických útvarov sme sa rozhodli aj my – tvorbou edukačného softvéru pre predškolákov. Uskutočnili sme i mikroprieskum skúseností predškolákov s počítačom.

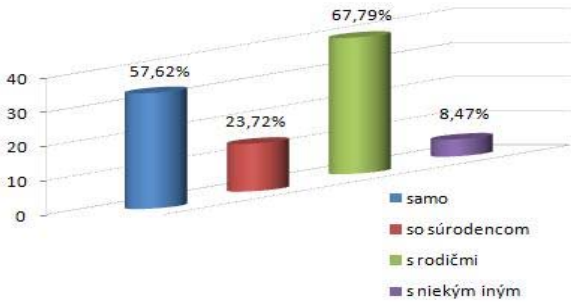
2. SKÚSENOSTI S POČÍTAČOM U PREDŠKOLÁKOV

Akým spôsobom používajú deti počítač v domácom prostredí? S akými softvérovými prostrediami sa stretajú? Na tieto otázky sme hľadali odpoveď pomocou dotazníka v dvoch partnerských materských školách. Prieskumu sa zúčastnilo 59 rodičov (30,6% celkového počtu), z toho 14 rodičov trojročných, 17 štvorročných, 23 päťročných a 5 šesťročných detí.

Väčšina detí (81%) má možnosť tráviť doma čas pri počítači. Čo robia deti na počítači? Mnohé na počítači kreslia (52,5%) a hrajú sa náučné (50,8%), prípadne športové (35,6%) a logické hry (28,8%). Pomerne veľa detí navyše sleduje, čo robia ich rodičia na počítači (39%). Ďalšie aktivity, ktoré rodičia uviedli, sú: písanie písmen, pozeranie fotografií, videa a rozprávok (aj v cudzom jazyku).

Deti sa na počítači hrajú nepretržite približne polhodinu (90%). Podobný výsledok zaznamenal rozsiahly americký prieskum spôsobov práce s počítačom [3] u 2 – 5 ročných detí (priemerne 27 minút denne) v roku 2000.

Zaujímalo nás tiež, kto sprevádza deti pri aktivitách s počítačom (viď Obr. 1).



Obr. 1 Odpovede na otázku
"S kým trávia škôlkari voľný čas pri počítači"

Najviac času sa hrajú s rodičmi (67,8%) alebo samé (57,6%). Zriedkavejšie nastane možnosť, že

1 http://www-05.ibm.com/sk/kidsmart/ [cit. 10.1.2009]

škôlkar „pracuje“ na počítači so súrodencom (23,7%), alebo s niekým iným (8,5%). Mnohé deti teda alebo spolupracujú s dospelým alebo naopak – hrajú sa samé. Pre tvorbu pedagogického softvéru z tohto vyplýva požiadavka **škálovateľnosti aktivít**, ale aj **jednoduchého ovládania** a **názornosti**.

Aké ďalšie nároky by mal spĺňať edukačný softvér pre predškolákov? V nasledujúcej časti si priblížime konkrétne softvérové prostredie a poukážeme na jeho silné a slabé stránky pre použitie v materskej škole. Ilustrujeme dve hľadiská – pohľad vývoja pedagogického softvéru a pohľad didaktiky jeho používania v škôlkach.

3. CHARAKTERISTIKA EDUKAČNÉHO PROSTREDIA VESELÁ ŠKÔLKA LIENKA. POHĽAD VÝVOJÁRA

Softvér „Veselá Škôlka Lienka“ je určený pre deti vo veku 5 – 6 rokov. Skladá sa zo šiestich rôznorodých aktivít, v ktorých dieťa predovšetkým usporadúva, triedi, dopĺňa a skladá základné geometrické útvary. Päť aktivít (Hra s kartičkami, Tangram, Vymaľuj obrázok podľa návodu, Rad, Had) tvoria uzatvorené úlohy, aktivita Stavebnica je divergentná a tvorivá. Aktivity nadväzujú na pracovné zošity v predškolskej príprave. Sú výsledkom viacerých konzultácií s predškolskými učiteľkami. Vývojárka následne testovala softvér s dvadsiatimi deťmi predškolského veku.

Obr. 2 Úvodná obrazovka Veselej škôlky Lienka

Po spustení programu sa deti ocitnú v rozprávkovom prostredí. Každú aktivitu symbolizuje lienka s náčrtom aktivity. Symboly aktivít si deti pri testovaní rýchlo zapamätali a dokázali si samostatne zvoliť aktivitu, ktorá ich najviac zaujala.

Ovládanie softvéru je **jednoduché** – väčšina akcií sa uskutočňuje klikaním na ľavé tlačidlo myši, prípadne ťahaním objektov.

Jediné textové informácie ukrýva dvojjazyčný (slovensko – maďarský) pomocník pre učiteľku. Ostatné informácie vrátane spätnej väzby sú reprezentované **výlučne grafickými prvkami**. Rôzne navigačné prvky sa vývojárka snažila navrhnúť tak, aby boli pochopiteľné i pre dieťa, ktoré ešte nevie čítať (keďže pomerne veľké percento detí z nášho prieskumu sa hrá na počítači aj bez pomoci dospelých).

Program poskytuje formou animácie pasívnu **spätnú väzbu pre dieťa**. V každej aktivite sa objavuje lienka Poradca, ktorá potvrdzuje splnenie úlohy. V prípade, že riešenie nie je správne, lienka Poradca niečo čmára do svojho poznámkového bloku. Ak je úloha správne vyriešená, lienka si od radosti zatancuje. Pri testovaní softvéru v predškolskej triede reagovali mnohé deti na výkon lienky smiechom.

Aktivity riadi dieťa, má možnosť samo si **určiť tempo**. Na úvodnej stránke si vyberá lienku zodpovedajúcu aktivite. Deti si počas testovania jednotlivé ikonky rýchlo osvojili a používali na prepínanie sa medzi aktivitami.

Počas návrhu a programovania všetkých aktivít sa vývojársky tím orientoval na poskytnutie vhodnej úrovne náročnosti pre každú aktivitu. Obtiažnosť sa zvyšuje po vyriešení zopár elementárnych úloh. V aktivite Tangram dieťa môže listovať medzi rôzne zložitými úlohami a zvoliť si obrázok zodpovedajúci vlastným schopnostiam či záujmom.

4. AKTIVITY SOFTVÉRU

Aktivita *Doplň rad* rozvíja u detí schopnosť logického myslenia, riešenia problémov a trénuje ich pozornosť, učí ich trpezlivosti, koncentrácii.

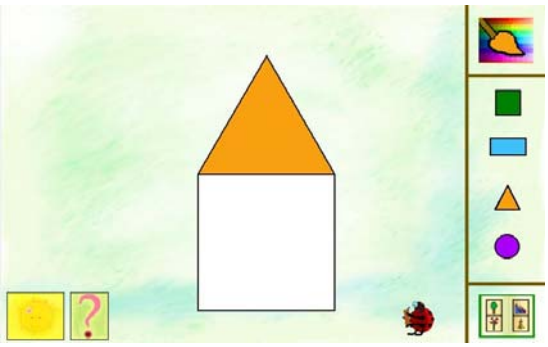
Obr. 3 Aktivita Doplň rad

Škôlkar má za úlohu doplniť chýbajúci tvar v rade striedajúcich sa tvarov. Najprv musí zistiť pravidlo, podľa ktorého sa tvary opakujú. V pravej časti obrazovky si zvolí daný tvar z viacerých možností. Po kliknutí na tvar ťahá dielik na prázdne políčko, ktoré treba doplniť. Pri správnom umiestnení v dolnej časti obrazovky vyletí malá lienka na steblo trávy. V opačnom prípade jedna lienka zo stebla odletí. Deti počas pozorovania práce so softvérom táto udalosť nesmierne motivovala, snažili sa rozmýšľať, aby čo najskôr pozbierali maximálny počet lienok.

Predškolská učiteľka pôsobila ako odborný poradca pre cieľovú skupinu softvéru. Vývojárka podľa jej požiadaviek prispôsobila softvér tak, aby sa prázdne políčko nikdy negenerovalo na prvých štyroch pozíciách radu. Hľadanie chýbajúceho tvaru na začiatku postupnosti si vyžaduje analýzu radu

v smere odzadu – dopredu, čo podľa predškolskej učiteľky presahuje schopnosti predškolského dieťaťa.

V pracovných zošitoch zameraných na rozvoj matematických predstáv sa veľmi často vyskytuje aktivita *Vymaľuj obrázok podľa návodu*.



Obr. 4 Aktivita Vymaľuj obrázok podľa návodu

V tejto úlohe je potrebné rozpoznať tvary a vyfarbiť ich v obrázku danou farbou.

Vývojárka vytvorila na základe pracovných zošitov dvanásť rôznych maľovaniek. Knižnica maľovaniek sa však dá podľa potreby rozširovať. Náročnosť úloh sa zvyšuje; začína dvoma tvarmi štvorec a trojuholník – teda domček. V ďalších zadaniach tvary postupne pribúdajú a začínajú sa prekrývať.

Štetec v pravom hornom rohu ukazuje, akú farbu má dieťa zvolenú. Keď spozná daný tvar v obrázku, klikne na tvar v pravej časti obrazovky – štetec vtedy zmení svoju farbu – a potom klikne na tvar v obrázku. Môže sa stať, že miesto obdĺžnika dieťa označí v ponuke tvarov štvorec. V takomto prípade sa tvar v maľovanke zafarbí a odfarbí trikrát zvolenou farbou (štvorca), no nakoniec zostane nezafarbený (biely).

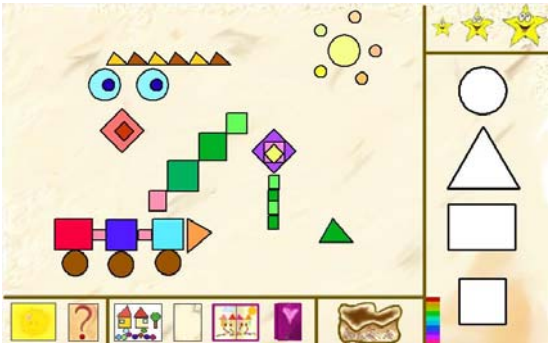
Aj v tejto aktivite lienka Poradca sleduje postup riešenia. Ak je každý tvar v obrázku správne zafarbený, lienka začne tancovať.

Pri overovaní tejto aktivity sa deti tešili hlavne z rôznych obrázkov – predlôh maľovaniek.

Základnými kameňmi aktivity *Stavebnica* sú opäť základné geometrické tvary. Dieťa tu voľne narába s tvarmi a ukladá ich do pracovnej plochy. V pracovnej ploche môže tvar presúvať – ťahaním – i vyfarbovať – klikaním. V pravom paneli nástrojov nájde paletu pre výber farby ako aj nástroj pre zmenu veľkosti tvaru. Dieťa si môže vybrať z troch veľkosti tvarov.

Obzvlášť peknú stavebnicu môže učiteľka alebo rodič uložiť kliknutím na ikonku zatvorenej knihy a napísaním mena súboru. Načítanie stavebnice ikonkou otvorenej knihy si vyžaduje zvoliť meno súboru.

Po načítaní stavebnice môže dieťa pokračovať v jej vytváraní. Teda presúvať, prefarbovať, mazať tvary, odfarbovať, pridávať ďalšie tvary a znovu ukladať. Objekty sa ukladajú do zoznamu v textovom súbore.



Obr. 5 Aktivita Stavebnica

Aktivitu Stavebnica môžu využiť učiteľky na prípravu nových zadaní pre aktivitu Vymaľuj obrázok podľa návodu.

Deti pozitívne reagovali na všetky testované aktivity. Zaujali ich nielen zadaniami, ale aj grafickým spracovaním[2]. Pri prvom testovaní vývojárka objavila rôzne nejasnosti, ktoré sa úspešne opravili (napr. aktivita Stavebnica ponúkala zo začiatku veľa zbytočných nástrojov, bolo potrebné zjednodušiť ovládanie). O projekt prejavili intenzívny záujem nielen deti, ale aj učiteľky, ktoré v ňom vidia cenný nástroj na precvičovanie geometrických pojmov.

5. Z POHĽADU DIDAKTIKY

Edukačné prostredie Veselá škôlka Lienka zodpovedá mnohým charakteristikám kvalitného pedagogického softvéru [4]:

- prostredie zaujme príťažlivou grafikou a obrázkami zrozumiteľnými pre cieľovú skupinu používateľov,
- väčšina aktivít má jednoduché a intuitívne ovládanie,
- má jasný didaktický cieľ,
- softvér je spracovaný formou hry,
- poskytuje rôzne úrovne náročnosti,
- je interaktívny, väčšina aktivít je rozšíriteľná,
- dáva deťom spätnú väzbu v podobe lienky Poradcu,
- deti si môžu samy voliť aktivity a postupovať vlastným tempom.

Softvér má podobu nástenky, jednotlivé aktivity sú navzájom poprepájané motívom lienok, nie dejovou líniou medzi aktivitami. Vo veľkej miere

2 Grafickú podobu programu vytvorila Daniela Onačilová.

rozvíja vizuálnu gramotnosť detí, ďalšie multimediálne prvky v ňom nie sú zastúpené.

Každej z aktivít na počítači by mala predchádzať priama manipulácia s kockami, stavebnicou či prinajmenšom pracovnými listami. Aktivita *Had*, v ktorej deti vyberajú jeden tvar tak, aby zodpovedal postupnosti tvarov – zhodoval sa vo farbe alebo tvare – sa obzvlášť hodí na hru s kartičkami v kruhu detí celej triedy. Deti si volia tvar a vysvetľujú, prečo si zvolili práve takýto tvar. Schopnosť rozpoznať daný tvar sa tak posúva až do úrovne hodnotenia a kritického posúdenia. Takýto typ aktivity je pre deti predškolského veku neobvyklý až náročný. Konzultujúca učiteľka takýto typ aktivity neocenila, i keď sa vyskytuje v slovenskej odbornej literatúre pre predškolákov [8]. Vznikla v konštruktivisticky zameranej brazílskej Escola Parque. Napriek kultúrnym odlišnostiam deti brazílskej škôlky trávili čas podobnými aktivitami, aké poznáme zo slovenských vzdelávacích inštitúcií pre predškolákov. Predškolská učiteľka hodnotila softvér Veselá škôlka Lienka kriticky – páčil sa jej, no zdôrazňovala, že je dobrý najmä kvôli aktivite Stavebnica, ktorá svojou otvorenosťou vedie deti k tvorivým riešeniam.

[5] stavia do kontrastu potrebu detského softvéru atraktívneho pre učenie sa s realitou: množstvo edukačného softvéru pre predškolákov nedovoľuje deťom experimentovať, testovať hypotézy a voľne manipulovať s prostredím. Na túto črtu by mali pamätať nielen vývojári detských softvérov, ale i učitelia zvažujúci výber softvérových produktov pre podporu učenia sa v škôlke.

Softvér zameraný na rozpoznávanie základných geometrických útvarov sa používa aj v slovenských materských školách. Napovedajú tomu výsledky školskej inšpekcie [1] už z roku 2007. Takáto situácia je bežná aj v iných krajinách. [2] ukazuje v prieskume 28 britských materských škôl, že väčšina softvérových aktivít v oblasti predškolskej matematiky je inštruktívna až drilovacia. Predškolskí pedagógovia však nie vždy pripisujú počítaču potenciál pre rozvoj tvorivého myslenia [6]. Práve aktivity ako Stavebnica dávajú téme rozpoznávania tvarov tvorivý rozmer. Na rozdiel od priamej manipulácie s objektmi môže softvér deťom ponúknuť veľkú flexibilitu vo voľbe tvarov, veľkostí, farieb. Pýtajme sa však na dôvod použitia uzatvorených softvérov: vyplýva z preferovaného štýlu aktivít v škôlkach alebo naopak, z preferovaného typu aktivít vyvíjaných pre predškolákov?

6. ZÁVER

Edukačné prostredie Veselá škôlka Lienka je príkladom softvéru pre rozpoznávanie tvarov, ktorý sa pohybuje od uzavretých aktivít s jedným riešením po jednu otvorenú aktivitu. Deti tak trénujú nielen špecifickú matematickú kompetenciu, no rozvíjajú i svoju fantáziu i prvotné kritické myslenie.

Nadšenie predškolských učiteľov pre integráciu rôznych softvérových prostredí do práce s deťmi by malo prameniť zo silnej reflexie ich kladov a záporov pre konkrétnu oblasť rozvoja osobnosti dieťaťa. To je podmienka pre to, aby sa počítače stali dobrým kamarátom predškolských detí, nikdy však nie náhradou rodičov či učiteľov. Vývojári edukačných prostredí ich môžu výrazne podporovať. Vyžaduje si to však neustály dialóg medzi pedagogickou verejnosťou a úzkym okruhom tvorcov pedagogického softvéru.

LITERATÚRA

[1] Hajduková, V.: Ako hodnotí Štátna školská inšpekcia materské školy v SR v školskom roku 2006/2007. In: *Predškolská výchova*, ročník 62 (1), 2007, pp. 11 – 18. ISSN 0032-7220.

[2] Hayes, M., Whitebread, D.: *ICT in the Early Years.* Maidenhead: Open University Press, 2006. ISBN 0335208088

[3] Children and Computer Technology: Analysis and Recommendations. In: *The Future of Children*, vol 10 (2), 2000, pp. 4 – 30. ISSN 1054-8289

[4] Moravčík, M.: *Edukačný softvér pre deti predškolského veku.* Písomná práca k dizertačnej skúške. Bratislava: FMFI UK, 2008.

[5] Torres, A. Creating games in Imagine for pre school. In: *EuroLogo 2007 – 40 Years of Influence of Education: Proceedings*. Bratislava: Knižničné a edičné centrum FMFI UK, 2007, CD ROM. ISBN 978-80-89186-20-4

[6] Stephen, C., Plowman, L.: *'Come back in two years!', a study of the use of ICT in pre-school settings.* Dundee: Learning and Teaching Scotland, 2003. ISBN 1-85955-779-1

[7] Štátny pedagogický ústav: Štátny vzdelávací program ISCED 0 – predprimárne vzdelávanie. [cit. 20.1.2009] http://www.statpedu.sk/buxus/docs/kurikularna_transformacia/isced0predpr.pdf

[8] Uherčíková, V. - Haverlík, I.: *Didaktika rozvíjania základných matematických predstáv.* Bratislava : DONY, 2007. ISBN 978-80-968087-4-8

DVUI – NÁRODNÝ PROJEKT ĎALŠIEHO VZDELÁVANIA UČITEĽOV ZÁKLADNÝCH ŠKÔL A STREDNÝCH ŠKÔL V PREDMETE INFORMATIKA.

RNDR. ZUZANA KUBINCOVÁ

Katedra základov a vyučovania informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina, Bratislava,
tel. 02/602 95 236, e-mail: kubincova@fmph.uniba.sk

PAEDDR. MARTINA CHALACHÁNOVÁ

Základná škola, Stredisková 2735/5, Lieskovec
tel. 045-5370227, e-mail: zwkladns@zslieskovec.edu.sk

Pre modernizáciu a zefektívnenie poznávacieho procesu má veľký význam aj komplexné vzdelávanie učiteľov. V súčasnom období je ešte oveľa dôležitejšie, keďže učitelia by mali byť hybnou pákou práve prebiehajúcej reformy školského systému. Aj zavedenie nového predmetu informatická výchova v primárnom vzdelávaní, či posilnenie predmetu informatika v štátnom vzdelávacom programe nižšieho i vyššieho sekundárneho vzdelávania, si vyžadujú venovať vzdelávaniu učiteľov týchto predmetov náležitú pozornosť.

Národný program ďalšieho vzdelávania učiteľov základných škôl a stredných škôl v predmete informatika ponúka vybranej skupine 1500 učiteľov možnosť moderného vzdelávania podporovaného digitálnymi technológiami. V našom príspevku prinášame informácie o tomto práve sa rozbiehajúcom projekte.

Kľúčové slová: *ďalšie vzdelávanie učiteľov, informatika, informatická výchova*

1. ÚVOD

V súčasnosti prebiehajúca reforma školstva prináša so sebou mnoho výrazných zmien. Jednou z nich – takou, ktorú naša komunita vníma mimoriadne pozitívne – je zavedenie nového predmetu Informatická výchova do štátneho vzdelávacieho programu na prvý stupeň základných škôl a posilnenie hodinovej dotácie predmetu Informatika na druhom stupni základných škôl a na stredných školách. Táto zmena však predpokladá dostatočný počet kvalifikovaných učiteľov informatiky a informatickej výchovy na všetkých typoch našich základných a stredných škôl. Bohužiaľ, súčasný stav je trochu odlišný.

Túto situáciu si zrejme uvedomilo aj MŠ SR, keď v rámci operačného programu Vzdelávanie vypísalo priame zadanie národného projektu Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika. Projekt bol vypracovaný (viď [1]) a 1.októbra 2008 ho minister školstva schválil.

K ďalším východiskám pre projekt patria fakty, že o vyučovanie predmetu informatika je veľký záujem aj zo strany rodičov a žiakov, a tiež učiteľov, vedenia škôl a zamestnávateľov, ako aj to, že predmet informatika má mnoho predpokladov vytvoriť produktívny priestor pre modernizáciu školy a vzdelávania, ako atraktívny a efektívny prostriedok pre komplexný rozvoj žiakov na oboch stupňoch ZŠ a na SŠ v zmysle konštruktivizmu, pre rozvoj ich digitálnej gramotnosti a informatického vzdelania, na rozvoj ich kreativity, stratégií učenia sa, schopností spolupracovať, komunikovať, učiť sa objavovaním a skúmaním.

Strategický cieľ projektu – navrhnúť, vyvinúť a realizovať moderné ďalšie vzdelávanie učiteľov informatiky a informatickej výchovy na ZŠ a SŠ, podporované digitálnymi technológiami – sa má dosiahnuť prostredníctvom nasledujúcich dvoch špecifických cieľov:

1. Vytvoriť a realizovať akreditované vzdelávacie moduly pre tri cieľové skupiny učiteľov informatiky a informatickej výchovy na ZŠ a SŠ.
2. Vybaviť účastníkov vzdelávania digitálnymi technológiami potrebnými k ich efektívnemu vzdelávaniu, ktoré následne využijú vo vyučovacom procese.

Prvý z nich sa bude realizovať prostredníctvom troch vzdelávacích aktivít, pričom každá bude zameraná na inú cieľovú skupinu. Ich cieľom je poskytnúť spolu 1500 učiteľom informatiky a informatickej výchovy na základných a stredných školách všetkých typov kvalitné vzdelanie potrebné pre realizáciu informatiky ako moderného všeobecnovzdelávacieho predmetu. Toto vzdelávanie nie bude klásť dôraz na komplexný rozvoj vlastnej digitálnej gramotnosti účastníkov, na rozvoj ich odbornej pripravenosti v oblasti informatiky, a tiež na rozvoj ich poznania v oblasti didaktiky informatiky a v oblasti moderných koncepcií a tendencií vo vzdelávaní, v hodnotení, v organizovaní poznávacieho procesu a v rozvíjaní vyšších poznávacích funkcií žiakov.

Obsah vzdelávania vo všetkých vzdelávacích aktivitách projektu sa bude realizovať v štyroch líniách:

- Línia 1: Digitálna gramotnosť učiteľa. Cieľom vzdelávania podľa tejto línie bude získanie či

rozvoj primeranej komplexnej digitálnej gramotnosti, ktorej súčasťou je napr. aj práca s modernými digitálnymi technológiami a softvérovými nástrojmi (edukačné portály, e-learning, virtuálne laboratóriá, robotické stavebnice, teleprojekty, pátracie aktivity, modelovanie, myšlienkové mapy, interaktívne tabule, m-learning, a iné). Ďalej tiež rozvoj kompetencií učiteľa pre celoživotné vzdelávanie, spoznanie otázok bezpečnosti a rizík pre žiakov vo virtuálnom priestore, ďalej sociálnych a právnych aspektov informatizácie spoločnosti a pod.

- Línia 2: Moderná škola. V nej sa účastník kurzu oboznámi s modernými teóriami poznávania a s modernými pohľadmi na školu ako priestor na premýšľanie, skúmanie, komunikáciu a kooperatívne učenie sa, spozná nové formy organizovania poznávacieho procesu, moderné formy motivovania a hodnotenia žiakov, alternatívne vzdelávacie systémy, a tiež úlohu digitálnych technológií v prekonávaní rôznych foriem nerovnosti. Súčasťou tejto línie bude aj úloha digitálnych technológií pre rozvoj vyšších poznávacích funkcií žiaka (hľadáme významy, robíme kritické úsudky, sme tvoriví, vyjadrujeme sa, prejavujeme sa, prezentujeme sa, premýšľame a riešime problémy, robíme rozhodnutia, učíme sa učiť, robíme výskum a podobne).
- Línia 3: Vlastný odborový kontext informatiky a informatickej výchovy – predstavuje vlastné jadro odbornej prípravy účastníkov vzdelávania. Obsah tejto línie bude primerane a vhodne zostavený pre potreby jednotlivých cieľových skupín.
- Línia 4: Didaktika informatiky a informatickej výchovy, v ktorej sa účastník oboznámi s cieľmi predmetu, osnovami a jeho formami, s didaktikou práce so štandardným softvérom, s grafikou, s textom, s číslami, grafmi a údajmi, s multimédiami, zvukom, fotografiou a hudbou, s didaktikou práce s internetom, s didaktikou algoritmizácie, programovania a riadenia digitálnych zariadení (robotov a iných).

Druhý špecifický cieľ sa naplní prostredníctvom aktivity, ktorá zabezpečí pre každého účastníka vzdelávania 1 notebook, 1 dataprojektor a 1 USB kľúč. Účastníci ich dostanú na začiatku vzdelávania a budú ich aktívne využívať počas celého svojho vzdelávania na získavanie a ukladanie materiálov, na vypracovávanie zadaných úloh i záverečného projektu, na komunikáciu s lektormi a kolegami, ako aj na implementáciu priebežných výsledkov svojho vzdelávania do vyučovania.

Projekt sa bude realizovať v troch rokoch, pričom v prvom by sa mal pripraviť samotný konkrétny obsah vzdelávania, zabezpečiť materiálne vybavenie pre účastníkov a začať vzdelávanie v jednej z cieľových skupín. V druhom a treťom roku by malo prebehnúť v dvoch etapách vzdelávanie aj v ďalších dvoch cieľových skupinách.

Vzdelávanie v rámci projektu bude prebiehať na celom Slovensku, výber účastníkov sa bude realizovať na základe vopred oznámených kritérií tak, aby ich počty boli rovnomerne rozdelené podľa počtu žiakov na jednotlivých typoch škôl v jednotlivých samosprávnych krajoch a súčasne zabezpečené rovnomerné zastúpenie učiteľov z rôznych škôl.

Obr. 1 Logo projektu

2. VZDELÁVANIE UČITEĽOV INFORMATICKEJ VÝCHOVY

Prvou vzdelávacou aktivitou projektu je vzdelávanie učiteľov informatickej výchovy, ktorá sa v štátnom vzdelávacom programe zavádza v primárnom vzdelávaní od druhého ročníka, v dotácii jednej hodiny týždenne.

Vzdelávanie žiakov mladšieho školského veku je natoľko špecifické, že je potrebné, aby tento predmet vyučovali učitelia s aprobáciou učiteľstva pre prvý stupeň ZŠ ale s doplneným vzdelávaním o aprobáciu pre vyučovanie informatickej výchovy.

Vzdelávanie učiteľov 1. stupňa ZŠ na informatickú výchovu je na Slovensku historicky prvým systémovým krokom k vytvoreniu priestoru a príležitosti pre moderný konštruktívny poznávací proces, v ktorom žiaci rozvíjajú svoju digitálnu gramotnosť, a tiež samostatné, kritické a zodpovedné rozhodovanie, svoju tvorivosť a ďalšie vyššie poznávacie funkcie.

Cieľom tejto aktivity je poskytnúť 700 učiteľom a učiteľkám 1. stupňa ZŠ moderné a kvalitné vzdelanie potrebné pre realizáciu nového predmetu v primárnom vzdelávaní – informatickej výchovy. Navrhnutý a vytvorený vzdelávací kurz bude akreditovaný na Akreditačnej komisii pre ďalšie vzdelávanie.

Vzdelávanie učiteľov potrvá jeden rok a bude pozostávať zo 130 vyučovacích hodín, pričom v úvode absolvuje každý pedagóg 4-dňové sústredenie s rozsahom 32 hodín, následne 10 stretnutí s odstupom 3 až 4 týždne, každé po 6 hodín. Na vypracovanie záverečnej práce budú mať 18 hodín a záverečné 2,5-dňové sústredenie spojené s obhajobou záverečnej práce a odbornou skúšobnou rozpravou bude trvať 20 hodín. Počas behu projektu sa uskutočnia dva takéto ročné vzdelávacie cykly.

Základnou časovou jednotkou vzdelávania bude modul so 6 vyučovacími hodinami. Oblasti vzdelávania sa budú organizovať do celkov pozostávajúcich z jedného, dvoch či viacerých modulov.

Obsah vzdelávania v tejto aktivite bude zodpovedať vyššie spomenutým štyrom líniám: *Digitálna gramotnosť učiteľa* – línia bude vo vzdelávaní zastúpená min. 24 vyučovacími hodinami (štyrmi modulmi), *Moderná škola* – bude vo vzdelávaní zastúpená min. 18 vyučovacími hodinami (tromi modulmi), *Vlastný odborový kontext informatiky a informatickej výchovy* – min. 36 hodinami (šiestimi modulmi) a *Didaktika informatiky a informatickej výchovy* – zastúpená min. 30 hodinami (piatimi modulmi).

Účastník úspešne ukončí svoje vzdelávanie, ak úspešne absolvuje všetky predpísané (alebo zvolené) študijné moduly a splní všetky predpísané študijné povinnosti, vypracuje a pred skúšobnou komisiou obháji svoju záverečnú prácu, úspešne absolvuje odbornú rozpravu pred skúšobnou komisiou, v ktorej komisia preverí, či zvládol obsah vzdelávania a splnil jeho ciele.

Po úspešnom ukončení vzdelávania účastník dostane zodpovedajúce osvedčenie.

3. VZDELÁVANIE UČITEĽOV, KTORÍ NEMAJÚ KVALIFIKÁCIU PRE PREDMET INFORMATIKA

Cieľom druhej vzdelávacej aktivity je poskytnúť 200 učiteľom 2. stupňa základných a stredných škôl všetkých typov – ktorí nemajú formálne požadovanú kvalifikáciu pre predmet informatika, ale učia ju, alebo ju budú učiť – kvalitné vzdelanie potrebné pre realizáciu informatiky ako moderného všeobecnovzdelávacieho predmetu.

Toto vzdelávanie bude prebiehať výlučne na fakultách slovenských vysokých škôl, ktoré v súčasnosti realizujú podľa akreditovaných študijných programov 1. a 2. stupňa prípravu budúcich učiteľov informatiky pre sekundárne vzdelávanie.

Navrhnutý a vytvorený vzdelávací kurz bude akreditovaný na Akreditačnej komisii pre ďalšie vzdelávanie. Učitelia získajú Osvedčenie o získanom vzdelaní s celoštátnou platnosťou vydané podľa platných zákonov o ďalšom vzdelávaní.

Po ukončení vzdelávania príslušné fakulty umožnia jeho účastníkom vykonať štátne skúšky a získať vysvedčenie o rozšírení pedagogickej spôsobilosti pre predmet informatika.

Viac informácií o tejto vzdelávacej aktivite možno nájsť v [2].

4. VZDELÁVANIE KVALIFIKOVANÝCH UČITEĽOV INFORMATIKY

Tretiu vzdelávaciu aktivitu projektu tvorí vzdelávanie kvalifikovaných učiteľov informatiky. Títo učitelia patria medzi tých, ktorí sa sami najviac vzdelávajú, majú snahu sledovať aktuálny vývoj technológií i vedeckých poznatkov, vedia sa prispôsobiť a používať nové technológie pri vyučovaní i príprave naň. Napriek tomu (alebo možno práve preto) sa organizovanému vzdelávaniu tejto skupiny učiteľov doteraz nevenovala taká pozornosť akú by si zasluhovalo. Preto sa nám táto aktivita javí ako dôležitý systémový krok.

Jej cieľom je navrhnúť, vytvoriť a zrealizovať kurz, ktorý bude akreditovaný na Akreditačnej komisii pre ďalšie vzdelávanie.

Účastníkmi vzdelávania budú kvalifikovaní učitelia informatiky na 2. stupni ZŠ alebo na SŠ, ktorí vyštudovali učiteľstvo informatiky ako svoju aprobáciu na univerzite alebo si ju dorobili formou ŠKŠ na MPC. Podmienkou je, aby pracovali na svojej škole na 100% pracovný úväzok, mali minimálne dvojročnú pedagogickú prax a počítalo sa s nimi v nasledujúcich školských rokoch pre výučbu informatiky na 2. stupni ZŠ, resp. na SŠ. V cieľovej skupine by malo byť 600 účastníkov, z toho 80 z Bratislavského kraja a zvyšok z ostatných krajov.

Vzdelávanie tejto skupiny učiteľov potrvá jeden rok a bude pozostávať zo 160 vyučovacích hodín v nasledujúcej štruktúre:

- dva semestre priameho aj dištančného vyučovania (e-learning), v každom semestri 10 týždňov po 4 hodiny výučby – spolu 80 hodín,
- po každom semestri 4-dňové sústredenie – spolu 64 hodín,
- vypracovanie záverečnej práce – 16 hodín,
- v rámci druhého sústredenia sa uskutoční obhajoba záverečnej práce a odborná skúšobná rozprava.

Základnou jednotkou vzdelávania bude modul s 8 vyučovacími hodinami. Oblasti vzdelávania sa budú organizovať do celkov pozostávajúcich z jedného, dvoch či viacerých modulov. Moduly budú navrhnuté takým spôsobom, aby pokrývali oblasti vzdelávania z jednotlivých línií tak, ako je to pre túto cieľovú skupinu najvhodnejšie – línia *Digitálna gramotnosť učiteľa* bude zastúpená minimálne 16 vyučovacími hodinami, línia *Moderná škola* minimálne 24 vyučovacími hodinami, línia *Odborový kontext* minimálne 64 vyučovacími hodinami a línia *Didaktika informatiky a informatickej výchovy* minimálne 40 hodinami.

Obsah vzdelávania bude rozdelený do povinných a voliteľných modulov realizovaných prezenčnou, dištančnou (e-learning) aj kombinovanou formou. Povinné moduly budú zrejme pripravené v dvoch

alternatívach – pre 2.stupeň ZŠ a pre SŠ. Podľa predbežného návrhu budú všetci účastníci povinní absolvovať moduly Didaktika informatiky a Moderné prostriedky vo vzdelávaní. Učitelia ZŠ a gymnázií budú musieť povinne absolvovať aj moduly Programovanie a Didaktika programovania. Okrem nich si ešte vyberú z voliteľných modulov tak, aby pokryli zvyšné hodiny. Učitelia SOŠ budú pokrývať voliteľnými modulmi väčší počet hodín.

Súčasťou každého študijného modulu bude špecifikácia jeho úspešného absolvovania – štandard vedomostí a zručností, ktoré musí dosiahnuť každý účastník, ktorý ho absolvuje. Po ukončení každého študijného modulu bude úlohou lektora vyhodnotiť a potvrdiť jeho úspešné absolvovanie každým účastníkom.

Prvý rok behu projektu bude v tejto aktivite vyhradený na detailnú tvorbu vzdelávacieho programu, modulov, študijných i výučbových materiálov, a pod. Samotné vzdelávanie učiteľov sa uskutoční v dvoch etapách v druhom a treťom roku behu projektu. V každom z nich sa ho zúčastní 300 učiteľov zorganizovaných do študijných skupín približne po 20 účastníkov. Študijné skupiny sa budú organizovať podľa bydliska tak, aby účastníci vzdelávania nemuseli dochádzať do veľkej vzdialenosti.

Výber účastníkov vzdelávania bude prebiehať na základe splnenia určitých kritérií, ktoré budú bodovo ohodnotené a na základe posúdenia motivačného listu, ktorý záujemcovia o vzdelávanie napíšu ako prejav záujmu o svoj odborný rast.

Vzdelávanie bude ukončené obhajobou záverečnej práce a vedeckou rozpravou na niektorú z tém z absolvovaných modulov, blízkych téme práce. Ako záverečnú prácu absolvent môže pripraviť napr. vzorovú vyučovaciu hodinu. Kompletne ju popíše v písomnom projekte a fragment z nej predvedie počas obhajoby.

Úspešní absolventi získajú Osvedčenie o získanom vzdelaní s celoštátnou platnosťou vydané podľa platných zákonov o vzdelávaní.

5. ZÁVER

Projekt umožní vzdelávať sa skupine 1500 učiteľov predmetu informatika a informatická výchova na oboch stupňoch ZŠ a na SŠ. Títo absolventi budú okrem odborovej a odborovo-didaktickej zložky poznať aj moderné koncepcie vzdelávania v zmysle konštruktivizmu, moderné formy organizovania poznávacieho procesu a hodnotenia žiakov, projektové vyučovanie, aktivity podporujúce komunikáciu a kooperáciu žiakov a rozvoj ich vyšších poznávacích funkcií. Počas vzdelávania tiež každý účastník významne rozvinie svoju digitálnu gramotnosť. Absolventi vzdelávania budú potom na svojich školách nepriamo sprostredkovávať svoje nové znalosti a zručnosti nielen žiakom, ale aj svojim kolegom a vedeniu školy.

Výstupom z projektu budú aj akreditované kurzy (ich koncepcia, štruktúra, rozsah, forma a obsah) pre tri typy ďalšieho vzdelávania učiteľov, súbor overených a expertnou skupinou recenzovaných metodických a študijných materiálov (v rôznych formách, v odôvodnených prípadoch napr. ako e-learningové kurzy), ako aj overená infraštruktúra takéhoto plošného vzdelávania.

Získané vzdelanie umožní absolventom humanizovať poznávací proces, zvyšovať efektívnosť učebných schopností a stratégií ich žiakov, a tiež vytvárať priestor pre konštruktívny poznávací proces, v ktorom žiaci rozvíjajú svoje samostatné, kritické a zodpovedné rozhodovanie, schopnosť komunikovať a kooperovať, vyjadriť sa primeranými formami a prostriedkami.

LITERATÚRA

[1] Národný projekt: Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika, Podrobný opis projektu, Interný materiál ŠPU, Bratislava, 2008.

[2] Andrejková, G: Druhá vzdelávacia aktivita v národnom projekte ďalšieho vzdelávania učiteľov v predmete informatika verzus Rozširujúce štúdium Informatika, podané na DIDINFO 2009, UMB Banská Bystrica, 2009

DIDAKTICKÝ TEST AKO SPÄTNOVÄZOBNÝ PRVOK PRE FORMU E-LEARNINGOVEJ EDUKÁCIE

ING. ANNA KÚTNA

Katedra informatiky, Pedagogická fakulta, Katolícka univerzita v Ružomberku, Nám. A. Hlinku 56/1, 034 01 Ružomberok
e-mail: kutna@fedu.ku.sk

ABSTRAKT

Testy a testovanie sú v súčasnosti veľmi rozšíreným spôsobom hodnotenia vo vyspelých krajinách sveta. Didaktické testy sú moderným prostriedkom preverovania a hodnotenia výsledkov procesu učenia sa jednotlivca. Zisťuje sa nimi kvantita a kvalita vedomostí a zručností študentov. Didaktický test je jeden zo spôsobov zisťovania vedomostnej úrovne študentov. Môžeme ho zaradiť medzi spätnoväzobný prvok pre elektronickú aj klasickú formu výučby.

Kľúčové slová: *didaktické testy, Moodle, pedagogický experiment, štatistická analýza*

1. ÚVOD

Spoločenské zmeny majú vplyv na výchovno-vzdelávací systém. Očakáva sa vysoko kvalitná výučba podľa reálnych požiadaviek praxe s dobrými podpornými službami. Pedagógovia musia vytvárať vysoko účinné výučbové systémy, ktoré uspokoja požiadavky spoločnosti. Úspešný e-learningový vzdelávací systém si vyžaduje systematické plánovanie, tvorbu návrhov, hodnotenie a uvedenie do praxe on-line prostredia, v ktorom je učenie aktívne podporované a stimulované.

2. DIDAKTICKÝ TEST

Didaktický test je jedným zo spôsobov zisťovania vedomostnej úrovne študenta. Testovanie má presné pravidlá, ale aj svoje zvláštnosti i riziká. Pre vytvorenie komplexu testov je potrebný tím špeciálnych odborníkov. Aj učiteľ sa v pedagogickej praxi stretáva s tvorbou testu, pre ktorý musia platiť aspoň základné kritériá kvality didaktického testu.

Katedra informatiky Pedagogickej fakulty Katolíckej univerzity realizovala pedagogický experiment, ktorého cieľom bolo preukázať, že vyučovanie prostredníctvom e-learningu je efektívnejšie ako klasická forma výučby. Išlo o prvý experiment tohto typu na Pedagogickej fakulte. V rámci pedagogického experimentu sme v predmete Informačné technológie pre prvý ročník odboru Špeciálnej pedagogiky zaradili tému „Sluchovo postihnutí a využitie informačných a komunikačných technológií“. Učivo bolo realizované dvoma spôsobmi. Prvý spôsob bol realizovaný s využitím elektronickej formy spracovaného učiva v LMS Moodle. Druhý spôsob bola výučba s využitím prezentácie v programe PowerPoint klasickou formou výučby čiže prednáškou. Obidve formy výučby boli prezentované jedným pedagógom.

Test pozostával z desiatich otázok z toho dve boli otvorené s možnosťou stručnej doplňujúcej odpovede. Osem otázok bolo zatvorených s možnosťou výberu odpovede z ponúkaných alternatív.

3. PEDAGOGICKÝ EXPERIMENT

Pre pedagogický experiment sme si stanovili hypotézu, že výučba prostredníctvom e-learningu zvýši vedomostnú úroveň študentov v porovnaní s výučbou realizovanou klasickou prezenčnou formou.

Úroveň osvojenia si učiva sme preverovali pomocou didaktického testu. Didaktické testy sú moderným prostriedkom preverovania a hodnotenia výsledkov procesu učenia sa jednotlivca. Zisťuje sa nimi kvantita i kvalita vedomostí a zručností učiacich sa subjektov.

Validitu didaktických testov je možné posudzovať podľa rôznych kritérií a metód, napríklad formou predvýskumu, ale aj pomocou rozhovoru s pedagógmi. V danom prípade bol didaktický test na preverenie znalostí z obsahu učiva predložený na posúdenie vyučujúcim v odbore špeciálna pedagogika. Z výpovedí učiteľov vyplynulo, že ani jedna úloha zostaveného testu nebola vyhodnotená ako nevhodná. Na základe hodnotenia pedagógov sme mohli konštatovať, že didaktický test pre zvolený experiment je dostatočne validný.

Celkový počet študentov zúčastnených na pedagogickom experimente bol 61. Prvá forma didaktického testu bola realizovaná u študentov špeciálnej pedagogiky, kde bolo učivo prezentované v elektronickej podobe a úroveň osvojených poznatkov bola preverená didaktickým testom v systéme Moodle (27 študentov). Druhú (kontrolnú) skupinu tvorilo 34 študentov, u ktorých bolo učivo prezentované klasickou formou a test mal písomnú podobu. Obidve skupiny študentov mali na vypracovanie test rovnaký čas a rovnaké podmienky. Experiment prebiehal v počítačovej miestnosti, študenti, ktorí boli v kontrolnej skupine písali na papier. Obe skupiny dostali na vypracovanie rovnaké testy. Pri teste v systéme Moodle nebolo možné používať internet ani iný zdroj informácií.

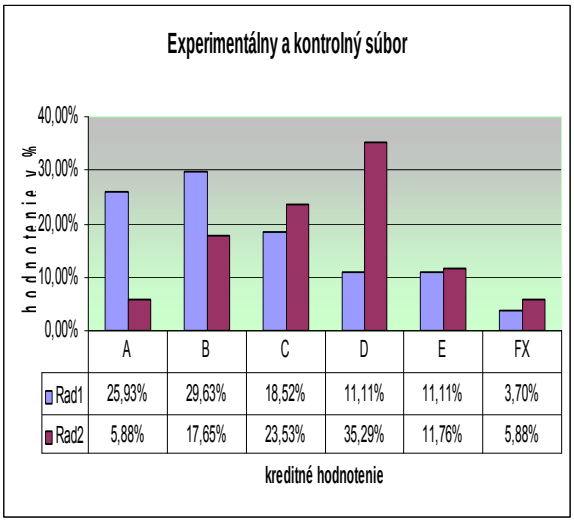
Študenti oboch skupín študujú v prvom ročníku odboru Špeciálna pedagogika na PF KU v Ružomberku.

Pedagogický experiment prebiehal najskôr v experimentálnej skupine, ktorej bolo učivo

prezentované v elektronickej forme s využitím LMS Moodle. Potom nasledovala výučba klasickou formou v kontrolnej skupine a rovnaký postup prebiehal aj pri overovaní osvojených poznatkov didaktickým testom. Výsledky získané pedagogickým experimentom boli spracované do kontingentných tabuliek absolútnych a relatívnych početností a následne boli aplikované pre verifikáciu stanovenej hypotézy príslušné štatistické operácie.

Pri porovnávaní úrovne úspešnosti osvojovaného učiva študentov experimentálneho a kontrolného súboru podľa grafu 1 je zrejmé, že hodnotenie na úrovni A v experimentálnej skupine je 25,93 % čo je vyššie o 20,05 % v porovnaní s kontrolnou skupinou, hodnotenie B je 29,63 % a je vyššie o 11,98 %, C je 18,52 % čo je nižšie o 5,01 % v porovnaní s kontrolnou skupinou, D je 11,11 % čo je nižšie o 24,18 % v porovnaní s kontrolnou skupinou, E je 11,11 % čo je nižšie o 0,65 % a FX je 3,70 % čo je nižšie o 2,10 % v porovnaní s kontrolnou skupinou.

Z uvedeného porovnania vyplýva, že študenti experimentálnej skupiny boli úspešnejší v úrovni osvojovaného učiva ako študenti kontrolnej skupiny. Toto zistenie sme ešte overili aplikáciou štatistických procedúr.



Graf 1 Porovnávací graf hodnotenia výsledkov didaktického testu experimentálneho a kontrolného súboru

Popisná štatistika		
	ES	KS
Stredná. hodnota	7,655556	6,988235
Chyba str. hodnoty	0,27829	0,226725
Medián	8	6,9
Modus	9	6,4
Smer. odchýlka	1,446038	1,32202
Rozptyl výberu	2,091026	1,747736
Špicatosť	-0,09503	0,062841
Šikmosť	-0,68434	-0,25579
Rozdiel max-min	5,7	5,6
Minimum	4	4
Maximum	9,7	9,6
Súčet	206,7	237,6
Počet	27	34
Hladina spoľahlivosti (95,0%)	0,572034	0,461275

Tabuľka 1 Popisná štatistika

V popisnej štatistike bol stanovený priemer súborov a chyba strednej hodnoty. Pracovali sme len s bodovou hodnotou, nakoľko relácie v % sú identické a prevod na kódovanie známky tiež. Z tabuľky popisnej štatistiky vyplýva, že všetky tri charakteristiky polohy (stredná hodnota, modus a medián) sú pri experimentálnom súbore vyššie ako pri kontrolnom súbore. Pretože parametre súborov sa vizuálne odlišujú, otestovali sme ich ďalej testom významnosti rozdielu medzi strednými hodnotami. Šikmosť a špicatosť nevylučujú možnosť normálneho rozdelenia pri obidvoch skupinách, čo je predpoklad pre jeho použitie. Tento odhad sme overili na histogramoch.

Histogram vyjadruje individuálne frekvencie pre údaje v rozsahu buniek a blokoch údajov. V tabuľke histogramu sa zobrazia hranice známok a počet výskytov dosiahnutých známok medzi dolnou a aktuálnou hranicou. Najčastejší výskyt známky predstavuje modus. Z tabuľky 2 vyplýva, že rozsah bodov v triede 2-4 a 4-6 je približne rovnaký, v triede 6-8 je najväčší pre kontrolný súbor, v 8-10 je najväčší pre experimentálny súbor.

Trieda	ES	KS
0-2	0,00	0,00
2-4	0,04	0,06
4-6	0,15	0,15
6-8	0,41	0,56
8-10	0,41	0,24

Tabuľka 2 Spoločný histogram výkonu obidvoch súborov

Pedagogický experiment ukázal skutočnosť, že študenti sú schopní si samostatne s minimálnou podporou učiteľa úspešne osvojiť určité učivo, ktoré je im prezentované prostredníctvom elektronických výučbových prostriedkov. Aplikácia elektronickej výučby môže viesť k ovplyvneniu vzťahu študentov k študovanému odboru, rovnako ako aj k ovplyvneniu motivácie študentov k ich štúdiu.

Elektronická výučba má potenciál pomáhať študentom pochopiť učivo efektívnejšie tým, že im ponúka alternatívne metódy jeho prezentácie, ktoré rešpektujú ich potreby a preferencie v štýle učenia sa, s čím súvisí aj ich zvýšená motivácia pri štúdiu samotnom.

Realizácia elektronickej výučby nielen pomáha zvyšovať motiváciu študentov k štúdiu, ale vedie k zlepšovaniu porozumenia a zapamätávania učiva,

umožňuje aj jeho prispôsobenie ich schopnostiam. Názornosť a interaktivita pomáha študentom si ľahšie predstaviť a zapamätať súvislosti. Je potrebná aj včasná spätná väzba zo strany učiteľa, aby študenti vedeli, že je sústredený na ich štúdium, čo môže tiež slúžiť ako motivačný faktor štúdia.

4. VÝSLEDKY PEDAGOGICKÉHO EXPERIMENTU

Výsledky pedagogického experimentu potvrdili predpoklad, že elektronická forma výučby je efektívnejšia ako klasická prezenčná forma. Na základe tohto zistenia sme mohli formulovať odporúčania pre pedagogickú prax aj na iných vysokých školách:

- je potrebné zabezpečiť, aby každá vysoká škola mala vlastný LMS, zodpovedajúci jej možnostiam a odbornej orientácii študijných programov,
- na vysokej škole by mala byť vybudovaná virtuálna učebňa dištančného vzdelávania, ktorá by zabezpečovala tvorbu e-learningových výučbových materiálov a ich distribúciu,
- všetci pedagógovia by mali postupne vypracovať prednášky jednotlivých disciplín v elektronickej podobe pre daný LMS systém vysokej školy,
- je potrebné včas pripraviť študentov pre prácu v LMS formou vhodného kurzu,
- pedagógovia by mali aplikovať v elektronickej forme didaktické testy ako jeden zo spôsobov spätnej väzby a na zistenie úrovne osvojených poznatkov študentov,
- konštrukcia didaktických testov musí zodpovedať jednotným kritériám s požadovanou validitou a reliabilitou,
- tvorba prednášky v e-learningovej forme by mala rešpektovať štandardné pravidlá (Computer Based Training) kurzov.

Elektronická forma vyučovania by mala byť aplikovaná nie iba vo výučbe odborných disciplín, ale aj v iných odboroch humanitného, prírodovedného a technického zamerania.

Odporúčame, aby výučba bola cieľavedome organizačne pripravená, dostatočne flexibilná a didaktické prostredie výučby by malo poskytovať študentom maximálnu pomoc.

Pre pedagógov je podstatné, aby elektronická výučba ich disciplín bola systematicky plánovaná a efektívnosť výučby aj pravidelne overovaná. Vysoká škola by mala mať prehľad o návratnosti investícií do systému elektronickej výučby aj o úrovni spokojnosti študentov s kvalitou výučby jednotlivých disciplín.

Prioritou by mal byť vývoj a zlepšovanie procesov riadeného samoučenia a efektívny proces motivujúceho vzdelávania.

Pedagógovia by mali pri tvorbe elektronickej výučby spolupracovať aj s odborníkmi z odboru pedagogiky, psychológie a informatiky, aby sa tak eliminovali chyby v obsahu učiva, ale aj v jeho didaktickej transformácii. Pretože populácia študentov je rôznorodá, mali by sa používať rozličné stratégie e-learningovej výučby a aktivity, ktoré vyhovujú rôznym štýlom učenia sa.

Zrejme nie je možné vytvoriť také prostredie elektronickej výučby, ktoré by vyhovovalo každému štýlu učenia sa, ale odporúčame, aby vysoké školy presne stanovili pravidlá a postupy pre využívanie elektronickej výučby vo všetkých formách vzdelávania (najmä dištančnej kombinovanej), ktoré by mali dodržiavať všetci účastníci e-learningu, pedagógovia, študenti aj administratívny aparát vysokoškolskej inštitúcie.

5. ZÁVER

Výsledky štatistickej analýzy údajov získaných pedagogickým experimentom ukázali, že existuje štatistický významný rozdiel v hodnotení výsledkov didaktického testu medzi experimentálnym a kontrolným súborom v prospech experimentálneho súboru. Toto zistenie potvrdilo stanovenú hypotézu, v ktorej sme predpokladali, že výučba prostredníctvom e-learningu zvýši vedomostnú úroveň študentov oproti výučbe realizovanej klasickou prezenčnou formou.

Tieto formálne odporúčania pre pedagogickú prax sú asi dosť odvážne, pretože vyplývajú z jedného pedagogického experimentu, ale na našej škole išlo o premiéru (prvotinu). Nemali sme k dispozícii výsledky z podobných experimentov na iných vysokých školách na základe, ktorých by bolo možné spraviť porovnanie. Navyše experiment prebiehal vo výučbe odboru Špeciálna pedagogika u prvého ročníka, kde študenti ešte nemajú veľa skúseností s LMS Moodle.

Elektronická výučba môže vo výchovno-vzdelávacej situácii priťahovať pozornosť študentov svojou atraktívnosťou a novosťou, čím môže navodiť a udržiavať ich vnútornú motiváciu k štúdiu v domácich podmienkach pri samoštúdiu aj v rámci vyučovacej hodiny v škole, čo má v konečnom dôsledku vplyv na ľahší a rýchlejší proces učenia.

LITERATÚRA

[1] ČERNÁK, I., MAŠEK, E.: Základy elektronického vzdelávania. Pedagogická fakulta KU v Ružomberku, 2007, s. 343. ISBN 978-80-8084-1713

[2] CHRÁSKA, M.: Didaktické testy. Paido. Brno 1999, s. 20 - 42. ISBN 80-85931-68-0

[3] GAVORA, P.: Výzkumné metody v pedagogice. Paido. Brno, 1996, s.87 - 96. ISBN 80-85931-15-X

[4] KÚTNA, A. Prostriedky hodnotenia výsledkov výučby v elektronickom kurze. In: Informatika v škole a praxi. 4. ročník konferencie

s medzinárodnou účasťou, 16.-18.9.2008, Ružomberok, s.166 – 170. ISBN 978 – 80 – 8084 – 362 – 5

[5] LEZO, M.: Štatistická analýza výskumných dát. In.: Základy metodológie pedagogického výskumu. Študijný text pre špecializačné štúdium učiteľov výskumníkov. Bratislava, SPN 1985, s. 79 - 128.

[6] LINQUIST, E.F.:Statistická aanlýza v pedagogickém výzkumu. PRAHA, SPN 1967

[7] TUREK, I.: Kapitoly z didaktiky vysokých škôl. Košice, KIP TU 1998

INFORMATICKÉ SÚŤAŽE NA SLOVENSKU

MGR. JULIANA LIPKOVÁ

Katedra základov a vyučovania informatiky, Fakulta matematiky, fyziky a informatiky Univerzity Komenského, Mlynská dolina, 842 48 Bratislava
e-mail: lipkova@fmph.uniba.sk

Pozeráme sa na informatické súťaže na Slovensku. Uvádzame príklady súťaží z rôznych oblastí informatiky. Spomíname ich ciele a ako sa môže študent či učiteľ dostať k informáciám o súťažiach, prípadne k nadstavbovým vedomostiam.Článok je určený hlavne učiteľom, ktorí by radi priviedli svojich študentov k súťažiam, či už preto, že vidia v študentovi nadanie, alebo preto, že chcú, aby si ich študenti rozširovali a trénovali kompetencie.

Kľúčové slová: *informatické súťaže, programovanie, olympiáda v informatike*

1. ÚVOD

Súťaže sú jeden z prostriedkov, ako pomáhať študentom vzdelávať sa vo voľnom čase. Na Slovensku beží niekoľko súťaží z rôznych oblastí informatiky.

Cieľom týchto súťaží nie je iba to, aby si študenti znalosti natrénovali. Ciele sú ďaleko rozsiahlejšie. Tie si popíšeme v kapitole 2. Typy súťaží uvedieme v kapitole 3 a bližšie popíšeme prototypy súťaží v kapitole 4. Kapitola 5 bude o možnosti neformálneho vzdelávania a o spôsobe ako získať informácie o súťažiach. Tieto dve témy uvádzam v jednej kapitole, lebo spolu úzko súvisia. V kapitole 6 sa pozrieme na informatické súťaže v zahraničí.

2. CIELE SÚŤAŽÍ

„Cieľa si všímaj pozornejšie, ako prostriedky. Prostriedky sú totiž pre cieľ, vôbec nie pre seba samé, a k prostriedkom sa spravidla pripája to, čo nesmeruje k cieľu a nemôžeme to ľahšie rozoznať ako tak, že prihliadneme na cieľ. Ak sa neustále hľadí k cieľu, je možné sa vyhnúť zákrutám.“

(J. A. Komenský)

Asi je všetkým jasné, že súťažami sa snažíme trénovať myseľ študentov a rozširovať ich vedomosti v oblasti informatiky. Nie je to však všetko.

Na súťažiach a sústredeniach sa stretávajú študenti s podobnými záujmami. Sú v rôznych vekových a odborných kategóriách, takže starší učia mladších a nastáva sociálna interakcia, čo sedí s Vygotského teóriou učenia sa [1].

Na sústredeniach a okolitých akciách sa buduje vzťah organizátor-riešiteľ, takže študent nemá zábrany sa opýtať a učiť sa od organizátora, pretože ho považuje za staršieho priateľa [14].

Študenti získavajú svoje vzory a to ich poháňa v ďalšom štúdiu, lebo vidia, čo môžu dosiahnuť. Motivujú ich tiež výhry, možnosť ísť na sústredenie, prípadne medzinárodné súťaže.

Príklady na súťažiach bývajú častokrát zo skutočného života. Zadania by mali byť napísané zábavnou formou. Je málo ľudí, ktorých nadchne úloha, ktorá je formulovaná tak, ako v súčasnosti bežne býva v škole: "Tu je vzorec, vyrátaj!". Zadania by mali byť formulované nenásilne, aby žiaka príklad zaujal a rozhodol sa nad ním porozmýšľať. [2]

Napríklad súťaž iBobor chce v žiakoch vyvolať potrebu spolupráce medzi sebou navzájom a tiež ich posmeliť v odovzdávaní svojich skúseností. Symbolom súťaže sa stal usilovný, inteligentný a čulý bobor. [3]

Obr. 1: Logo súťaže iBobor

Súťaže sa snažia podchytávať talenty. Častokrát majú viac kategórií. To znamená, že študent začne riešiť najľahšiu kategóriu a postupne prechádza k ťažším a ťažším príkladom. Medzi kategóriami nebýva veľký skok, takže študent na súťaž nezabudne, lebo stále má čo riešiť. Tým sa zo začiatočníka môže stať úspešný človek, ktorý sa nestratí ani vo vedeckej informatike.

Na sústredeniach sa snažíme ukázať študentom, že byť informatikom neznamená celý deň sedieť za počítačom, alebo riešiť teoretické úlohy. Treba aj športovať, trénovať myseľ inými typmi úloh a získavať rôzne typy vnemov. Pozri [2].

Taktiež sa študenti učia dodržiavať termíny, opisovanie a podvádzanie je na súťažiach neprijateľné a moja skúsenosť hovorí, že študenti sa ani nesnažia podvádzať. A samozrejme súťažiam sa venujú vo voľnom čase, namiesto toho, aby robili pre spoločnosť neprijateľné veci.

3. TYPY SÚŤAŽÍ

V minulosti bolo súťaží v informatike iba pár. Aj vďaka atraktívnosti informatiky na trhu vznikajú nové a nové súťaže. Väčšina je na podporu talentov, iba málo je pre širokú verejnosť. Existujú súťaže pre študentov základných a aj stredných škôl.

Najviac súťaží je algoritmických a robotických. Sú však aj súťaže rozvíjajúce znalosti práce s IKT, spoločenské aspekty, či logické myslenie.

Niektoré súťaže sú krátkodobé, niektoré dlhodobé. Niektoré treba riešiť na mieste, pri iných stačí posielať riešenia cez Internet.

4. PROTOTYPY

Je veľký rozdiel medzi súťažami určenými pre základoškolákov a stredoškolákov. Preto si tieto kategórie uvedieme samostatne.

Algoritmické súťaže pre študentov základných škôl bývajú väčšinou v detských programovacích jazykoch, ako je Imagine, Logo, Baltík. Patria sem súťaže ako Palma Junior, Baltík a Imagine Logo Cup. Úlohou v týchto súťažiach býva buď vykresliť niečím špecifikovaný obrázok (niekedy bývajú parametre zadávané počas behu programu, nie v zadaní), alebo nejakú animáciu, ktoré môže mať aj interakciu s užívateľom.

MiniProg je algoritmická súťaž pre základoškolákov. Líši sa od predchádzajúcich hlavne tým, že sa programuje v jazykoch ako Delphi, Pascal, Basic, Visual C++. Na rozdiel od stredoškolských súťaží sú však príklady viac zamerané na praktickosť, ako na algoritmy.

Ďalšie súťaže sú v robotike. Ich výhoda je taká, že sú veľmi atraktívne pre diváka. Na Slovensku tento rok odštartovala súťaž FIRST Lego League a už dlhšie roky beží RoboCup Junior. Žiaci tu konštruujú a programujú robotov, ktorí musia zvládnuť vopred zadanú úlohu, prípadne robia projekty a učia sa tímovej práci.

Pre žiakov, ktorí riešia radi logické, matematické a informatické hádanky, existuje portál Bitland (pozri [4]). V nedávnej dobe sa mohli tiež zabaviť na súťaži Sherlock Holmes. V tej si mohli trénovať a zväčšovať výber vhodných počítačových softvérov, naučili sa pracovať v skupine a vyhľadávať informácie na Internete [5].

Tento rok, už ako druhý rok, prebehla súťaž iBobor [3]. Tá sa snaží pokryť veľkú oblasť informatiky, takže úlohy sú rôznorodé aj svojim obsahom, aj svojou náročnosťou. Súťaž je určená pre študentov základných a aj stredných škôl. Jej zaujímavosťou je tiež to, že ju rieši množstvo žiakov z celého Slovenska a umožňuje im porovnať sa s ostatnými rovesníkmi.

Na stredných školách je najprestížnejšou súťažou Olympiáda v Informatike (ďalej OI). Má dve kategórie. B kategóriu riešia spravidla študenti 1. a 2. ročníka SŠ a A kategóriu 3. a 4. ročníka SŠ. Samozrejme mladší študenti môžu riešiť aj vyššiu kategóriu. V najvyššej kategórii sa dá postúpiť až na Medzinárodnú olympiádu v informatike. Táto súťaž je algoritmická, nestačí však vymyslieť a naprogramovať ľubovoľný algoritmus, mal by byť aj efektívny.

Korešpondenčný seminár z programovania (ďalej KSP) pomáha pripravovať študentov na OI, avšak je to samostatná súťaž. Príklady sú rovnakého rázu ako na OI. Zaujímavosťou KSP je však týždňové sústredenie, ktoré je odmenou pre najlepších riešiteľov, a to vo všetkých kategóriách, aby sa učili aj pravidelní a aj začínajúci študenti.

Ďalšími algoritmickými súťažiami sú Palma, ProFiit, Zenit a IT-Liga. V týchto treba úlohu naprogramovať a poslať do systému na otestovanie. Zaujímavosťou súťaže IT-Liga boli aj pravidelné hlavolamy.

IPSC (Internet Problem Solving Contest) je súťaž, ktorá je na Slovensku unikátna (a bola prvá svojho druhu aj na svete). Zaujímavá je tým, že zadanie síce vyzerá podobne ako zadanie algoritmických súťaží, má definovaný problém a vstup, avšak ako riešenie sa posiela výstup úlohy, nie program, ktorý úlohu rieši. To znamená, že riešiteľ môže použiť ľubovoľný nástroj, napríklad aj pero a papier, aby úlohu vyriešil. Táto súťaž je tímová, medzinárodná a v samostatnej kategórii ju môže riešiť široká verejnosť, vysokoškoláci, čí ľudia pohybujúci sa v IT priemysle..

Okrem algoritmických súťaží beží na Slovensku veľa robotických súťaží. Napríklad je to Istrobot a RoboCup. Líšia sa hlavne témami. Súťaží sa vo futbale, sume, či v chodení po bludisku. Úlohou je aj naprogramovať a aj zostrojiť robota.

5. NEFORMÁLNE VZDELÁVANIE

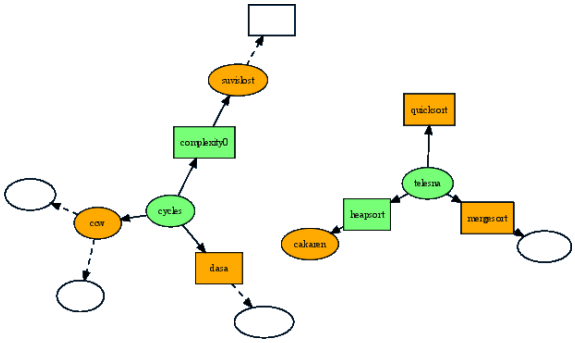
Na Slovensku je viac miest, kde si môžu študenti rozširovať znalosti z informatiky. Popíšeme si niektoré akcie, ktoré sú zamerané na vzdelávanie vo vedeckej informatike.

Združenie Trojsten [6] organizuje pravidelne v Bratislave dve akcie: Klub Trojstenu a Akadémia Trojstenu. Na týchto akciách, okrem zábavného programu, bežia prednášky a diskusie z matematiky, fyziky a informatiky. Na Klube prednášajú ľudia organizujúci korešpondenčné semináre a na Akadémii to sú osobnosti z Fakulty matematiky, fyziky a informatiky, ktorí sa snažia vedu priblížiť mladým ľuďom.

V Košiciach sa uskutočňujú stretnutia s názvom Popoludnie s informatikou. Ako sa píše na webstránke [7], cieľom je sprostredkovať najnovšie poznatky a trendy v oblasti informatiky, riešiť a analyzovať úlohy z programátorských súťaží a olympiád a neformálne diskutovať o všetkom, čo súvisí s informatikou a štúdiom na vysokej škole.

Ďalšia možnosť vzdelávania je na sústredeniach. Napríklad na sústredení KSP študent absolvuje 2 prednášky denne, seminár (čo je prednáška na pokračovanie) a odborne orientované hry. To je dokopy asi 5 hodín informatiky za deň.

Na Internete funguje aj e-learningový portál Liaheň [8]. Študent tam nájde vzdelávacie texty a príklady. Príklady sú usporiadané do stromovej štruktúry, pričom keď študent vyrieši úlohu, otvorí sa mu možnosť riešiť ďalšie úlohy, ktoré predpokladajú znalosť použitého algoritmu v predchádzajúcej úlohe.



Obr. 2: Graf úloh v Liahni

Aby študent začal riešiť súťaže, musí dostať prvotný impulz. To môže byť napríklad od učiteľa. Na to však samotný učiteľ musí vedieť niečo o súťažiach.

Zoznam súťaží so stručnými popismi a hypertextovými linkami môžeme nájsť na [9] a [10].

Zadania a letáky súťaží sa posielajú na školy. Tam sa posielajú tiež informácie o prípadnom školení pre učiteľov, kde sa dozvedia a naučia, ako s deťmi pracovať a pomôcť im s prípravou na súťaž.

Taktiež je dobré sledovať webstránku [11]. Môžeme si tam pozrieť rôzne metodické texty a občas nezisková organizácia P-MAT organizuje školenia.

Školenia pripravujúce učiteľov na prácu so žiakmi, ktorí majú záujem o súťaže, sa objavujú sporadicky. Jedným z takýchto školení bol Učiteľský seminár k Olympiáde v informatike [15].

6. SÚŤAŽE V ZAHRANIČÍ

V zahraničí existujú rôzne súťaže. Mnoho krajín vysiela svojich reprezentantov na Medzinárodnú olympiádu v informatike. Väčšinou tomu, rovnako ako u nás, predchádza národná olympiáda.

Olympiáda zahŕňa aj témy, ktoré nie sú súčasťou štandardných učebných osnov. V zahraničí sa im venujú väčšinou na sústredeniach, ktoré sa uskutočňujú pred olympiádou. Zaujímavé je, že korešpondenčné semináre, tak ako sa organizujú u nás, sú iba v strednej Európe.

Častokrát sa študenti pripravujú na rôznych internetových portáloch. Príkladom je Chorvátska olympiáda [12] a USACO [13].

V zahraničí samozrejme nebežia len súťaže, ktoré sú zamerané na prípravu na olympiádu (a s tým spojené aj algoritmické vzdelávanie). Napríklad v Litve vznikla súťaž Beaver, ktorá beží teraz aj na Slovensku pod názvom iBobor. Robotické súťaže zväčša pochádzajú z Ázie.

7. ZÁVER

Nie je to náhoda, že práve riešitelia súťaží dosahujú vynikajúce výsledky na vysokej škole a neskôr v živote. Súťaže ich naučili, že vzdelanie majú vo svojich rukách, a že veda je radosť. Preto sa nebráňme súťažiam a pomáhajme študentom nájsť si k nim cestu.

LITERATÚRA

[1] BENSON, P. - GRABE, W. - STOLLER, F. L.: Teaching and Researching Autonomy in Language Learning. Pearson Education, 2001, s. 39, ISBN 0582368162

[2] LIPKOVÁ, J. : Korešpondenčné semináre ako súčasť mimoškolského vzdelávania, Bratislava, 2007, seminárna práca z predmetu Pedagogika, dostupné na Internete [http://people.ksp.sk/~julka/skola/pedagogika/ped.pdf]

[3] HRUŠECKÁ, A. - PEKÁROVÁ, J. - TOMCSÁNYI, P. - TOMCSÁNYIOVÁ, M.: Informatický bobor – nová súťaž v informačných technológiách pre žiakov základných a stredných škôl, DidInfo 2008, Univerzita Mateja Bela, Fakulta prírodných vied, Banská Bystrica, 2008, ISBN 978-80-8083-556-9

[4] http://www.bitland.sk/

[5] ONDRIŠKOVÁ, J. - GÁLIK, Z. : Miesto činu – Internet, DidInfo 2007, Univerzita Mateja Bela, Fakulta prírodných vied, Banská Bystrica, 2007, ISBN 978-80-8083-367-1

[6] http://www.trojsten.sk/

[7] http://ics.upjs.sk/psinf/

[8] http://www.ksp.sk/~liahen/

[9] http://people.ksp.sk/~julka/sutaze/

[10] http://www.infovek.sk/predmety/inform/index.php?k=6

[11] http://www.ucmeradi.sk/

[12] http://www.hsin.hr/coci/

[13] http://train.usaco.org/usacogate

[14] FORIŠEK, M. - WINCZER, M. : Non-formal Activities as Scaffolding to Informatics Achievement, Information Technologies at School, Valentina Dagiene and Roland Mittermeir, eds., 2006, s. 529-534

[15] http://ics.upjs.sk/pub/usoi.php

INFORMATICKÁ VÝCHOVA: ELEKTRONICKÉ PRACOVNÉ LISTY

RNDR. GABRIELA LOVÁSZOVÁ, PHD.

Katedra informatiky, Fakulta prírodných vied, Univerzita Konštantína Filozofa, Tr. A. Hlinku 1, 949 74 Nitra,
e-mail: glovaszova@ukf.sk

Informatická výchova je od školského roku 2008/2009 povinným predmetom v 2. až v 4. ročníku základnej školy. V článku sa zaoberáme konkretizáciou učebných osnov do podrobnejších tematických plánov a návrhom vhodných foriem a metód práce. Uvádzame príklady elektronických pracovných listov a návrhy projektov, v ktorých sa e-pracovné listy použijú.

Kľúčové slová: *informatická výchova, e-pracovné listy, projekty*

1. ÚVOD

Informatická výchova je novým povinným predmetom v učebnom pláne 1. stupňa základných škôl (v medzinárodnej klasifikácii ISCED1). Štátny vzdelávací program mu vyčleňuje po jednej hodine týždenne v 2., 3. a 4. ročníku.

Zavedeniu informatickej výchovy do učebného plánu 1. stupňa základnej školy predchádzalo experimentálne vyučovanie v rámci projektu Informatizácia v ZŠ (od roku 2001) na 19 základných školách pod vedením Štátneho pedagogického ústavu [1], [2]. Overovali sa tri varianty informatizácie vyučovania: informatická výchova ako samostatný predmet; využívanie informačno-komunikačných technológií (IKT) na ostatných predmetoch počas pevne stanovenej vyučovacej hodiny nad rámec učebného plánu a integrovanie IKT do ostatných predmetov. Okrem tohto experimentu sa na mnohých základných školách vyučovala informatika v rámci nepovinného predmetu Práca s počítačom alebo v rámci krúžkovej činnosti. Tieto skúsenosti, vybavenie základných škôl počítačovými učebňami a spoločenská potreba digitálnej gramotnosti vyústili do zaradenia informatickej výchovy medzi povinné predmety na 1. stupni základných škôl.

2. OD UČEBNÝCH OSNOV K VYUČOVACEJ HODINE

Pre nový vyučovací predmet boli vypracované učebné osnovy. Ich zavedenie do praxe si však vyžaduje tvorivý prístup učiteľa, ktorého úlohou je vypracovať konkrétny časovo-tematický plán, zvoliť vhodné vyučovacie metódy, pripraviť aktivity a učebné pomôcky na jednotlivé vyučovacie hodiny. Inšpiráciou, ako postupovať, môže byť článok [3].

Práca učiteľa je v prípade informatickej výchovy o to ťažšia, že:

1. Mnohí učitelia nemajú dostatočnú kvalifikáciu na vyučovanie tohto predmetu. Tento problém bude čiastočne riešiť projekt Ďalšie vzdelávanie učiteľov v informatike, ktorého cieľom je poskytnúť potrebnú kvalifikáciu 600 učiteľom 1. stupňa ZŠ.
2. Neexistujú učebnice pre tento predmet. Učiteľ musí čerpať z rôznych zdrojov (napr. [4]). Učebnice by sa mali vydávať postupne. Za užitočné považujeme aj existenciu viacerých alternatívnych učebníc.

V tabuľke 1 uvádzame návrh na konkretizáciu obsahu (Čo učiť?) pre jednotlivé ročníky. Vychádzali sme z učebných osnov, ktoré definujú obsah a štandardy pre celý stupeň vzdelávania.

Tab. 1 Tematický plán vyučovania informatickej výchovy pre 2. až 4. ročník ZŠ

Téma	2. ročník	3. ročník	4. ročník
Oblasť Informácie okolo nás (ION): **Údaj, informácia** **Text** **Obrázok** **Zvuk** **Animácia**	Práca s myšou (pohyb, klik, dvojklik, ťahanie) Kreslenie voľnou rukou (ťahanie) Pečiatkovanie, výplň (klikanie) Písanie jednoriadkového textu Písanie čísiel Mazanie textu, pohyb po texte šípkami Vyhľadávanie údajov v databáze (kalendár, slovník, rozvrh hodín, žiacka knižka)	Textový editor (editovanie textu, formátovanie písma) Grafický editor (používanie nástrojov štetec, výplň, čiara, elipsa, obdĺžnik) Práca s menu a tlačidlami Práca s výberom Práca so schránkou (klávesové skratky, kontextové menu) Animácia (dvojfázová) na papieri a v počítači Vytvorenie databázy (rodostrom)	Textový editor (formátovanie odsekov) Grafický editor (používanie nástrojov, vkladanie textu) Tabuľkový kalkulátor (zapísať údaje do tabuľky, formátovať tabuľku) Zvuk (nahrávanie a prehrávanie) Video (nahrávanie a prehrávanie) Multimediálny projekt (návrh a príprava materiálu)

			Zber údajov do tabuľky (anketa)
Oblasť Komunikácia prostredníctvom IKT (IKT): **e-mail** **sms** **www**	Sms	e-mail (písanie) Prehliadanie webových stránok (adresa, odkaz)	Vyhľadávanie informácií na webe, sťahovanie súborov, posielanie súborov e-mailom
Oblasť Postupy, riešenie problémov, algoritmické myslenie (ALG): **Algoritmus** **Kód** **Vykonávanie** **Zápis**	Obrázkové návody (vykonávanie, zoraďovanie) Slovné návody (vykonávanie, zostavovanie) Symbolické návody (vykonávanie, zostavovanie)	Ikonografický programovací jazyk (Baltík) Kódovanie a šifrovanie	Programovací jazyk (Baltík alebo Logo) Postupnosť príkazov, cyklus Číselné sústavy (grafický prevod)
Oblasť Princípy fungovania IKT (PP): **Počítač** **Periférie** **Súbory a priečinky**	Hardvér: Klávesnica, myš, monitor, reproduktor, slúchadlá (aktívne) tlačiareň, skener, dataprojektor (pasívne) Softvér: Spustenie programu (dvojklik) Ukončenie programu	Pamäť (operačná, pevný disk, CD, DVD, USB kľúč) Súbor (otvorenie, uloženie) Priečinok (prehľadávanie, vytvorenie, kontextové menu, pravé tlačidlo myši)	Mikrofón Diktafón Webová kamera Digitálny fotoaparát GPS
Oblasť Informačná spoločnosť (IS): **Výučbový softvér** **Hry** **Multimédiá** **Počítačová bezpečnosť** **Počítačová etika**	Výučbový softvér Hry Medzipredmetový projekt	e-mailová etika Bezpečnosť (meno a heslo) Prierezový projekt (dopravná výchova, zdravie a zdravá výživa, ochrana životného prostredia, región)	Počítačový vírus a ochrana Etika kopírovania, typy softvéru z hľadiska autorských práv (voľný, voľný na vyskúšanie, platený) Teleprojekt

Ďalšou otázkou, ktorú si musí učiteľ pri zostavovaní podrobného časovo-tematického plánu položiť, je: Ako učiť? Voľba metód a celkovej koncepcie vyučovania zároveň ovplyvňuje aj obsah, výber konkrétnych ativít.

Za najvhodnejšiu metódu vyučovania hlavne v 2. ročníku považujeme riešenie úloh vo forme pracovných listov. Táto forma je deťom blízka a je využívaná aj v iných predmetoch. V informatickej výchove sú to elektronické pracovné listy, ktoré sa vypĺňajú na počítači s cieľom nacvičiť prácu s myšou a s klávesnicou. Medzipredmetové vzťahy zas využijeme pri tvorbe projektov, ktoré sú viac zamerané na rozvoj tvorivosti.

Treťou otázkou je: Prečo? Pri návrhu aktivity a jej zaradení do určitého tematického okruhu je potrebné si uvedomiť, aké primárne ciele chceme dosiahnuť, aké ďalšie ciele môžeme dosiahnuť a či je daná aktivita na dosiahnutie týchto cieľov efektívna.

3. ELEKTRONICKÉ PRACOVNÉ LISTY

Elektronický pracovný list je malá aplikácia s didaktickým obsahom a s veľmi špecifickým vzdelávacím cieľom. Jeho výhodou je, že dieťa môže okamžite pracovať na konkrétnom zadaní bez nutnosti vykonávať iné činnosti, ktoré nesúvisia s primárnym cieľom vyučovania.

Práci s elektronickým pracovným listom by mal predchádzať motivačný rozhovor o téme pracovného listu. Pri náročnejších zadaniach odporúčame vyplnenie pracovného listu aj v papierovej podobe, aby dieťa pochopilo, čo sa od neho bude očakávať a mohlo sa sústrediť na obsluhu počítača.

Pracovné listy môžu obsahovať drilové úlohy, ktorých cieľom je rozvíjať motorické zručnosti pri práci s počítačom, ale aj úlohy problémové a tvorivé. Takéto pracovné listy je možné potom použiť pri tvorbe projektu. S pracovnými listami pracujú žiaci väčšinou samostatne. Pri tvorbe komplexnejších projektov spolupracujú s učiteľom a so spolužiakmi.

Uvedieme dva príklady elektronických pracovných listov: Kalendár a Škola robotkov. Oba námety sú vhodné aj na projektové vyučovanie.

4. KALENDÁR

Séria pracovných listov je zameraná na vyhľadávanie informácií v databáze (konkrétne v kalendári) a na nácvik používania klávesnice a myši. Cieľom je, aby žiak vedel:

- doplniť údaj do postupnosti údajov (dátumy, dni),
- vyhľadať údaj v databáze (meno v kalendári podľa dátumu, dátum podľa mena a mesiaca, mesiac podľa dátumu a mena).

Ďalšími cieľmi je upevňovať si vedomosti a zručnosti osvojené predtým:

- písať čísla na numerickej klávesnici,
- písať slová s diakritikou a s veľkými písmenami,
- opraviť chybu pri písaní pomocou klávesu BackSpace.
- dvojkliknať myšou pri aktivovaní editovacieho riadka v pracovnom liste.

Obr. 1 Pracovný list k projektu Kalendár

Žiaci sa stretli s témou plynutia času na hodinách prvouky a prírodovedy. Motivačný rozhovor sa vedie o týchto poznatkoch, o narodeninách, meninách a iných sviatkoch a podobne. Ako pomôcku si z domu žiaci prinesú papierové kalendáre, v ktorých sa nachádzajú aj mená. Po vyplnení jedného papierového pracovného listu vypĺňajú niekoľko e-pracovných listov. Pracovné listy obsahujú vždy jeden týždeň s údajmi: rok, mesiac, dátum, deň v týždni a meno (obr. 1).

Prácu možno rozvinúť do projektu, keď deti nakreslia obrázok vystihujúci niektoré ročné obdobie a vyplnia pracovný list s týždňom z daného ročného obdobia. S pomocou učiteľa obrázok zoskenujú. Učiteľ prenesie obrázok do pracovného listu a spolu so žiakmi vytlačí. Podľa počtu a usilovnosti žiakov možno vytlačiť celý rok alebo určitú časť roka.

Po doplnení pracovných listov o obrázky budú žiaci vedieť:

- pomenovať skener a vysvetliť, na čo slúži,
- pomenovať tlačiareň a vysvetliť, na čo slúži.

5. ŠKOLA ROBOTKOV

Námet na túto aktivitu sme čerpali z [4]. Pracovný list je súčasťou projektu, v ktorom si deti vytvoria kartičky na didaktické hry zamerané na rozvoj abstraktného a algoritmického myslenia. Z počítačových zručností sa precvičuje práca s myšou. Cieľom projektu je, aby žiak vedel:

- pracovať podľa návodu,
- triediť a vyhľadávať údaje podľa rôznych kritérií.

Ďalšími cieľmi je upevňovať si vedomosti a zručnosti osvojené predtým:

- klikať myšou,
- vedieť pomenovať tlačiareň a vysvetliť, na čo slúži.

Práca na projekte pozostáva z dvoch etáp. V prvej etape pracujú žiaci samostatne s e-pracovným listom, pomocou ktorého si pripravia kartičky na didaktickú hru. V druhej etape sa spolu hrajú didaktcké hry s pripravenými kartičkami.

Pracovný list obsahuje 16 kartičiek s obrázkom robotka zoradených do matice 4 krát 4 kartičky. Na obrázku robotka nie sú vyfarbené štyri atribúty: anténky, oči, uši a krk. Úlohou je vyfarbiť tieto časti robotkov (klikaním myšou) tak, aby každý zo šestnástich robotkov bol iný. Úloha je vzhľadom na vysoký počet kartičiek netriviálna, preto budú deti postupovať podľa návodu (algoritmu):

1. Vyfarbi ľavú anténku všetkých robotkov v prvom a v druhom riadku. Potom vyfarbi pravú anténku všetkých robotkov v treťom a v štvrtom riadku.
2. Robotkom v prvom a v druhom stĺpci vyfarbi oči tak, aby pozerali doprava, v treťom a v štvrtom stĺpci majú pozerať doľava.
3. Robotkom v prvom a v treťom riadku vyfarbi uši, robotkom v druhom a v štvrtom riadku nechaj uši nevyfarbené.
4. Robotkom v prvom a v treťom stĺpci vyfarbi krk, robotkom v druhom a v štvrtom stĺpci nechaj krk nevyfarbený.

Po vyfarbení si deti s pomocou učiteľa kartičky vytlačia a samy rozstrihajú. Kartičky sa použijú na

didaktické hry, ktoré rozvíjajú abstraktné a algoritmické myslenie.

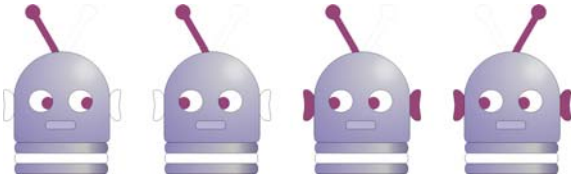
Hra „Pošta pre robotka“:

Do školy robotkov prišla pošta. Robotka, ktorý dostal poštu, učiteľ oznámi vymenovaním jeho štyroch vlastností (anténka, oči, uši, krk). Deti majú identifikovať robotka, ktorému patrí pošta (zdvihnú kartičku). Pri určovaní robotka manipulujú s kartičkami, triedia údaje podľa znakov vždy do dvoch skupín.

Hra „Nástup robotkov“:

Robotkovia sa v škole naučili nastupovať. Na povel Nástup! sa zoradia tak, že vedľa seba stoja robotkovia, ktorí sa líšia len v jednej vlastnosti.

Hra je určená pre skupiny žiakov alebo pre celú triedu. Každý pracuje so svojimi kartičkami. Po nástupe sa spočíta od začiatku dĺžka radu, pokiaľ sú robotkovia zoradení správne. Za každého robotka v dobre zostavenom rade získa žiak jeden bod. Na obrázku 2 sú správne nastúpení štyria robotkovia.



Obr. 2 Nástup robotkov

Hra „Posádka rakety“:

Posádku rakety tvoria štyria robotkovia. Do posádok sú zadeľovaní podľa nejakého pravidla. Deti dopĺňajú troch členov posádky štvrtým, musia odhaliť pravidlo.

Neúplné posádky rakety má učiteľ pripravené a premieta ich na tabuľu. Deti samostatne manipulujú s kartičkami. Úlohu je možné zjednodušiť zmenšením súboru kartičiek, z ktorých sa vyberá štvrtý člen posádky.

6. ZÁVER

Používanie e-pracovných listov považujeme za najvhodnejšiu formu vyučovania na hodinách informatickej výchovy hlavne u detí, ktoré sa s počítačom stretávajú poprvýkrát. Vhodne premyslenými zadaniami sa trénuje zručnosť v používaní myši (klikanie, dvojklikanie, ťahanie) a v používaní klávesnice (písanie oboma rukami).

Vývoj poznávacích schopností detí v tejto vekovej kategórii sa nachádza vo fáze konkrétnych operácií. Poznávací proces sa orientuje hlavne na získavanie skúseností a tvorbu separovaných (konkrétnych) modelov. Preto sme toho názoru, že vyučovanie by sa nemalo sústrediť len na prácu s počítačom, ale okrem počítačových aktivít by mali deti pracovať aj s hmotnými učebnými pomôckami (v aktivite Kalendár to boli papierové kalendáre, v Škole robotkov papierové kartičky). Manipuláciou s hmotnými objektmi deti získavajú autentickejšie skúsenosti potrebné pre rozvoj abstraktného myslenia.

V tvorivých činnostiach odporúčame vždy venovať veľkú pozornosť estetickej stránke žiackych produktov. Preto sme napríklad v projekte Kalendár navrhli nakresliť obrázok ručne na papier a zoskenovať. Úlohy by sa mali voliť tak, aby boli primerané schopnostiam detí a zároveň aby sa nimi dal dosiahnuť čo najestetickejší výsledok.

Tento príspevok vznikol v rámci projektu Spoločné integrované laboratórium didaktiky prírodovedných predmetov DIDLAB a bol podporený Agentúrou na podporu výskumu a vývoja na základe zmluvy č. LPP-0486-07.

POĎAKOVANIE

Ďakujem Mgr. Ladislavovi Székesimu a jeho žiakom 4. ročníka ZŠ Ferenca Kazinczyho v Tornali, ktorí uvedené aktivity zrealizovali a poskytli mi spätnú väzbu.

LITERATÚRA

[1] BLAHOVÁ, V. : Informatická výchova v 1. stupni základnej školy. In *DidInfo 2004*. Banská Bystrica : FPV UMB, 2004. ISBN 80-8055-786-1

[2] CHALACHÁNOVÁ, M.: Vyučovanie informatickej výchovy na 1. stupni ZŠ s podtitulom Nebojte sa začať čo najskôr. In *DidInfo 2006.* Banská Bystrica : FPV UMB, 2006, s. 95-98. ISBN 80-8083-202-1

[3] TOMCSÁNYIOVÁ, M.: Od osnov k vyučovacej hodine informatiky. In *DidInfo 2005*. Banská Bystrica : FPV UMB, 2005, s. 131-136. ISBN 80-8083-090-8

[4] CHALACHÁNOVÁ, M.: *V kráľovstve počítačov*. MAPA Slovakia Plus, 2005, 38 s. ISBN 80-8067-142-7

[5] PÁLI, J. – FORGÁCH, B.: *Jó játék a kép, a kártya.* Budapest : Móra Kiadó, 1987, 54 s. ISBN 963 11 5293 6

TVORBA VZDELÁVACÍCH CIEĽOV VO VYUČOVANÍ INFORMATIKY

ING. JANKA MAJHEROVÁ

Katedra informatiky, Pedagogická fakulta, Katolícka univerzita v Ružomberku, Nám. A. Hlinku 56/1, Ružomberok
e-mail: majherova@fedu.ku.sk

V príspevku sa venujeme určovaniu vzdelávacích cieľov vo vyučovaní informatiky v praxi. Na základe poznatkov upravenej Bloomovej taxonómie cieľov sme realizovali prípravu a hodnotenie experimentálnej výučby zameranej na modelovanie v predmete informatika na gymnáziu.

In this paper we address a specification of the educational goals in the teaching of informatics. Based on the modified Bloom's taxonomy of targets, we have implemented the development and evaluation of the experimental teaching, which is focused on modeling at the high school informatics.

Kľúčové slová: *vzdelávacie ciele, upravená Bloomova taxonómia, vyučovanie informatiky, modelovanie rastlín*

1. ÚVOD

Pri stanovení obsahu a cieľov výučby informatiky na základnej a strednej škole môžeme použiť známu Bloomovu taxonómiu výchovno-vzdelávacích cieľov (BT), ktorá bola vytvorená v 50. rokoch 20. storočia [1]. Bloomova taxonómia je jednodimenzionálna a zahŕňa šesť hierarchicky zoradených kategórií: znalosť (vedomosť), porozumenie, aplikácia, analýza, syntéza a hodnotenie. Jednotlivé kategórie kognitívneho procesu vyjadrené slovami poznať, pochopiť, používať, analyzovať, vytvoriť alebo navrhnúť v spojení s konkrétnym obsahom učiva nájdeme aj vo vzdelávacích štandardoch predmetu [2].

V súčasnosti sa môžeme v oblasti taxonómie cieľov inšpirovať upravenou Bloomovou taxonómiou, ktorá bola publikovaná v USA v r. 2001 [3].

2. UPRAVENÁ BLOOMOVA TAXONÓMIA VZDELÁVACÍCH CIEĽOV

Upravená Bloomova taxonómia vzdelávacích cieľov (UBT) vznikla doplnením pôvodnej BT pod vplyvom rozvoja kognitívnej psychológie a požiadaviek praxe. Zameriava sa na kognitívnu doménu, v ktorej je podľa autorov ťažisko výučby.
Nová taxonómia je dvojdimenzionálna (tab. 1). Najpočetnejšiu kategóriu pôvodnej Bloomovej taxonómie **vedomosti** autori vyčlenili a nahradili novou dimenziou – **znalostnou**, ktorá má štyri kategórie:

- faktické poznatky
- konceptuálne poznatky
- procedurálne poznatky
- metakognitívne poznatky.

Znalosť faktov predstavuje základné prvky, ktoré musia žiaci poznať, aby boli oboznámení s predmetom a boli schopní riešiť jeho problémy. Znalosť konceptov zahŕňa vzájomné vzťahy medzi základnými prvkami vo vnútri väčších štruktúr, ktoré umožňujú ich vzájomné fungovanie. Procedurálne poznatky sú o tom, ako niečo robiť, patria medzi ne odborné techniky, metódy a zručnosti. Metakognitívne znalosti predstavujú stratégiu myslenia a riešenia problémov.
Druhú dimenziu upravenej Bloomovej taxonómie – dimenziu **kognitívneho procesu** tvorí šesť kategórií: zapamätať si, porozumieť, aplikovať, analyzovať, hodnotiť, tvoriť (tab. 1). V prípade kategórií *syntéza* a *hodnotenie* bolo vymenené poradie. Syntéza bola nahradená kategóriou „tvoriť“, ktorá nie je chápaná ako opätovné zostavenie jednotlivých prvkov, ale zahŕňa tvorivý prvok ako aj hodnotenie. Kategória *pochopenie* bola premenovaná na *porozumenie*. Kognitívne dimenzie sú uvádzané v tvare slovesa (verbum), kým znalostné dimenzie sú v tvare podstatného mena (substantívum). Pri upravenej Bloomovej taxonómii sa ustupuje od hierarchického usporiadania kategórií od najnižšej po najvyššiu. Môže dochádzať k prelínaniu najmä na aktivitách vo vyučovaní, napr. žiak niekedy hodnotí, hoci ešte neanalyzoval alebo tvorí na základe čiastočných vedomostí. V celku sa však majú prejaviť všetky dimenzie. V tabuľke 1 sú uvedené príklady možných činností žiakov pre jednotlivé zložky znalostnej a kognitívnej dimenzie [podľa 4].

Ak téma vyučovacej jednotky je napríklad **Program, zložený príkaz, opakovanie bloku príkazov,** vzdelávacie ciele môže učiteľ stanoviť nasledovne:

Žiak dokáže

1. opísať pojmy príkaz, zložený príkaz, blok príkazov, cyklus
2. porozumieť pojem cyklus
3. použiť jednoduchý cyklus v programe.

Tieto ciele výučby podľa upravenej Bloomovej taxonómie pokrývajú kognitívne dimenzie zapamätania, porozumenia a aplikovania a zároveň sú zamerané na získanie faktických,

konceptuálnych a procedurálnych poznatkov z programovania. Vzdelávací cieľ „žiak dokáže použiť cyklus v programe“ môžeme v tabuľke 1 zaradiť ako napr. „aplikovať konceptuálnu a procedurálnu znalosť“ (**B3, C3**).

Tabuľka 1 Dimenzie upravenej Bloomovej taxonómie

	DIMENZIA KOGNITÍVNEHO PROCESU					
ZNALOSTNÁ DIMENZIA	**1. Zapamätať si**	**2. Porozumieť**	**3. Aplikovať**	**4. Analyzovať**	**5. Hodnotiť**	**6.Tvoriť**
A. faktické poznatky	vymenovať, uviesť	stručne vyjadriť, zhrnúť	roztriediť, klasifikovať	usporiadať	zatriediť vybrať	kombinovať
B. konceptuálne poznatky	opísať	interpretovať rozoznať	experimentovať	vysvetliť porovnať	odhadnúť určiť	plánovať načrtnúť
C. procedurálne poznatky	usporiadať	predpokladať	vypočítať riešiť	rozlišovať znázorniť	vyvodiť, usúdiť	vytvoriť poskladať navrhnúť
D. metakognitívne poznatky	použiť	spracovať	skonštruovať	vytvoriť	vykonať vyjadriť	aktualizovať zdokonaliť

Zaradenie cieľa do tabuľky nie je jednoduché a vždy jednoznačné. Samotná formulácia vzdelávacích cieľov a ich taxonomické zaradenie pomáha učiteľovi pri voľbe vyučovacích prostriedkov (metód a foriem výučby) ako aj spôsobe hodnotenia primeranom povahe cieľov. Zvolenému vzdelávaciemu cieľu má zodpovedať činnosť žiaka i učiteľa. Niektoré vyučovacie metódy a roly učiteľa a žiaka vhodné pre jednotlivé kategórie kognitívneho procesu sú uvedené v tabuľke 2 [podľa 5].

Poznatky upravenej Bloomovej taxonómie sú aplikované i v novom Rámcovom vzdelávacom programe pre stredné školy ako aj v iných medzinárodných pedagogických dokumentoch (napr. PISA 2006) [6,7].

Tabuľka 2 Vyučovacie metódy podľa kategórií kognitívneho procesu

	1. Zapamätať si	**2.Porozumieť**	**3. Aplikovať**	**4.Analyzovať**	**5. Hodnotiť**	**6.Tvoriť**
Metódy	výklad, problémový výklad, test, kvíz	riešenie úloh, vysvetlenie, príklad, kvíz, rozhovor	samostatné riešenie úloh, skupinová práca	riešenie úloh riadené učiteľom, brainstorming	problémový výklad, diskusia, riešenie úloh	riešenie úloh, kooperatívne učenie, práca s modelmi
Rola učiteľa	riadi, rozpráva, preveruje, pýta sa, hodnotí	demonštruje, porovnáva, počúva, pýta sa, hodnotí	ukazuje, pomáha, pýta sa, organizuje, hodnotí, sleduje	rozdeľuje, riadi, pýta sa, hodnotí	objasňuje schvaľuje usmerňuje	pomáha, vyjadruje sa analyzuje, hodnotí
Rola žiaka	odpovedá, definuje, zapamätá si	vysvetľuje, opisuje, demonštruje interpretuje	rieši, predvádza použitie vedomostí, vypočíta	aktívne diskutuje, argumentuje testuje, skúša, skúma	aktívne diskutuje, porovnáva pýta sa, vyberá	aktívne tvorí, formuluje, plánuje, navrhuje, modifikuje

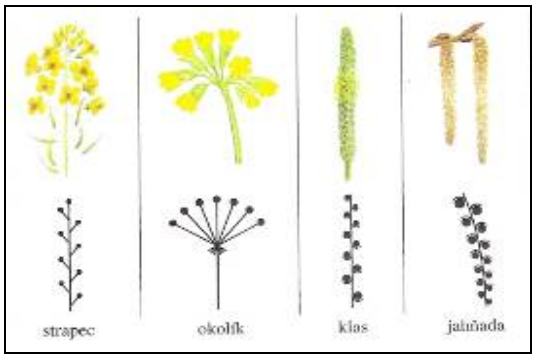
3. POUŽITIE UPRAVENEJ BLOOMOVEJ TAXONÓMIE PRI MODELOVANÍ NA INFORMATIKE

Upravená Bloomova taxonómia nás motivovala pri príprave a realizácii experimentálnej výučby tematického celku Model a modelovanie v informatike na strednej škole. Jeho cieľom bolo vytvorenie modelu rastliny v programovacom prostredí pomocou korytnačej grafiky [8,9].
Výučba prebiehala v 1. ročníku gymnázia v dvoch skupinách žiakov počas ôsmich týždňov. Pri tvorbe modelu rastliny sme ako vyučovacie formy použili individuálnu a skupinovú prácu v dvojiciach a projektové vyučovanie. Pomôckou bolo programovacie prostredie Baltie 4 C# a obrázky jednoduchých súkvetí rastlín (obr. 1). Pri výučbe sme použili vyučovacie metódy ako problémový výklad, riešenie úloh, tvorbu projektu a iné.

Po absolvovaní tematického celku žiak mal získať poznatky vo všetkých znalostných kategóriách upravenej Bloomovej taxonómie:

– faktické poznatky: znalosť, čo je to model, algoritmus, program, príkaz, cyklus a pod.

– konceptuálne: poznať súvislosti medzi pojmami (uhol - otočenie korytnačky a pod.)
– procedurálne: zapísať a upraviť program, použiť cyklus, premennú, procedúru
– metakognitívne poznatky: prepojiť vedomosti z predmetov informatika, matematika a biológia.



Obr. 1 Modely súkvetí

Pre vyučovaciu jednotku Tvorba modelu rastliny sme stanovili vzdelávacie ciele a ku každému sme priradili kód podľa jeho umiestnenia v hierarchii upravenej Bloomovej taxonómie. Žiaci mali nadobudnúť nasledovné vedomosti a zručnosti:

- navrhnúť grafický model rastliny podľa originálu – aplikovať poznatky faktické a konceptuálne **(A3, B3),**
- použiť príkazy korytnačej grafiky na realizáciu grafického modelu rastliny – zapamätať, porozumieť a aplikovať konceptuálne a procedurálne poznatky **(B1, B2**, **B3, C3)**
- použiť prvok náhodnosti pri modelovaní – aplikovať poznatky (**B3, C3)**
- realizovať grafický model rastliny na počítači – tvoriť (**C6)**
- zmeniť vlastnosti modelu – analyzovať a hodnotiť (**C4, C5)**
- použiť v programe premenné, cykly, procedúry – porozumieť a aplikovať procedurálne poznatky **(C2, C3, C4)**
- zhotoviť projekt Modelovanie v informatike – tvoriť (**C6)**
- aplikovať medzipredmetové vzťahy predmetov matematika - biológia - informatika **(D2, D3)**

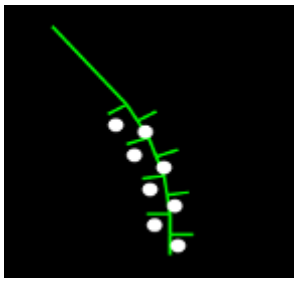
Dimenzie uvedených vzdelávacích cieľov sme prehľadne znázornili aj v tabuľke 3.

Tabuľka 3 Dimenzie vzdelávacích cieľov vyučovacej jednotky Tvorba modelu rastliny

	1	2	3	4	5	6
A	X	X	X			
B	X	X	X			
C		X	X	X	X	X
D		X	X			

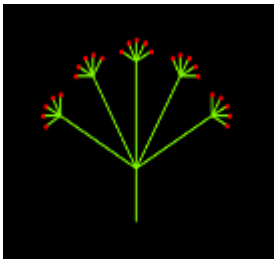
Pri realizácii výučby sme stanovené vzdelávacie ciele použili aj na hodnotenie žiakov. V rámci kognitívneho procesu žiak mohol získať kompetencie vo všetkých oblastiach od zapamätania, porozumenia, cez aplikovanie po hodnotenie a tvorbu nových produktov.

Samozrejme všetci žiaci nedosiahli rovnakú úroveň vedomostí a zručností. Na ilustráciu porovnáme výstupy dvoch žiakov Lukáša a Juraja. Pri analýze ich programov a projektov sme zistili, že obaja žiaci zvládli použitie cyklu a parametrov príkazov. Rozdiely boli pri použití premenných a procedúr. Juraj samostatne upravil príkazy a premenné v procedúre, aby dosiahol zmenu modelu rastliny (obr. 2).



Obr.2 Jurajov model jahňada okolík

Lukáš vedel upraviť parametre príkazov, ale nevedel zmeniť premennú alebo príkaz v procedúre. a na modeli rastliny vedel zmeniť iba farbu (obr. 3).



Obr. 3 Lukášov model okolík

Pri tvorbe záverečného projektu mali žiaci samostatne hotový model upraviť, napr. použiť prvok náhodnosti. Juraj pristúpil k projektu tvorivo, použil nové informácie z internetu o modeloch a samostatne upravil grafický model rastliny v programovacom prostredí. Lukáš v projekte použil len informácie získané počas výučby. Jednoduchý model rastliny zostavil, ale nesprávne priradil typ súkvetia originálneho kvetu. Lukáš dosiahol úroveň zapamätania, porozumenia, aplikovania a čiastočne analýzy získaných vedomostí, Juraj prejavil aj úroveň samostatnej analýzy a hodnotenia.

Úroveň dosiahnutých kategórií kognitívneho procesu sme obom žiakom zaznačili do taxonomickej tabuľky. V nasledujúcich dvoch tabuľkách vidíme rozdiel medzi dosiahnutou úrovňou u Lukáša (tab. 4) a Juraja (tab. 5). Stĺpce tabuliek označujú dimenzie kognitívneho procesu, riadky označujú typ nadobudnutých poznatkov.

Tabuľka 4 Dimenzie kategórií Lukáša

	1	2	3	4	5	6
A	X	X	X			
B	X	X	X			
C		X	X	X		
D						

Tabuľka 5 Dimenzie kategórií Juraja

	1	2	3	4	5	6
A	X	X	X			
B	X	X	X			
C		X	X	X	X	X
D		X	X			

Práca s modelmi predstavuje vyššiu úroveň kognitívneho procesu vo vyučovaní informatiky. Cieľom našej výučby bolo, aby modelovanie zároveň podporilo algoritmické myslenie a programovacie zručnosti žiakov. Tvorba grafického modelu žiakom sprostredkovala názorne použitie cyklu, procedúry, premenných ako aj vizuálne názorné testovanie a ladenie programu.

4. ZÁVER

Naším príspevkom chceme prispieť k metodike správneho formulovania vzdelávacích cieľov a k posúdeniu ich kognitívnej náročnosti. Určenie vzdelávacieho cieľa a didaktických aktivít podľa upravenej Bloomovej taxonómie pomôže pri zefektívnení výučby informatiky. Vedie k výberu vhodných aktivít na vyučovacej hodine, k rozvoju kompetencií žiakov a následne umožní objektívne hodnotenie žiakov. Žiak neostáva len pri úrovni zapamätania a porozumenia, ale má priestor použiť svoje vedomosti v praxi a získať kľúčové kompetencie potrebné pre jeho ďalšie štúdium a odbornosť. Predmetom ďalšieho výskumu môže byť overenie, aké typy modelov a ich tvorba podporí u žiakov vyššie úrovne kognitívneho procesu a získanie metakognitívnych kompetencií.

Literatúra:

[1] Bloom B. et al. 1956. *The Taxonomy of Educational Objectives. The Classification of Educational Goals, Handbook I: Cognitive Domain*. New York: Longman Green, 1956. Podľa Turek, I.: *Inovácie v didaktike*. 2005. s. 14-16. Bratislava: MPC, 2005.

[2] Blahová, V. 2001. *Vzdelávací štandard z informatiky pre gymnázium štvorročné štúdium. ŠPÚ 2001.* Dostupné na <http://www.statpedu.sk/buxus/generate_page.php_page_id=343.html>

[3] Anderson, L. et al. 2001. *A Taxonomy for Learning, Teaching a Assessing of Educational Objectives*. New York: Longman, 2001. 352 s. ISBN 0-321-08405-5. Podľa Hudecová, D. 2007. *Revize Bloomovy taxonomie edukačních cílů.* Dostupné na <http://www.msmt.cz/Files/DOC/NHRevizeBloomovytaxonomieedukace.doc> [6.5.2008]

[4] Overbaugh, R. – Schultz, L. Bloom's Taxonomy. Dostupné na <http://www.odu.edu/educ/roverbau/Bloom/blooms_taxonomy.htm> [6.5.2008]

[5] Bloom´s Revised Taxonomy. Dostupné na <http://www.kurwongbss.eq.edu.au/thinking/Bloom/blooms.htm>[6.5.2008]

[6] Rámcový vzdelávací program pre stredné školy. Obsah vzdelávania Vzdelávacia oblasť Matematické myslenie a práca s informáciami. ŠPÚ. 2007. Dostupné na <http://www.statpedu.sk/buxus/docs//RVP/rvp_ss_0203_matematika.pdf> [10.7.2008]

[7] *PISA Slovensko národná správa 2006.* Dostupné na <http://www.statpedu.sk/buxus/docs//projekty/PISA/pisa2006nsprava.pdf> [10.7.2008]

[8] Kelemenová, A., Majherová, J. 2006. *L systémy v stredoškolskej informatike.* In: Černák, I. (ed.): Informatika v škole a v praxi. Ružomberok: PF KU 2006. s. 193-197. ISBN 80-8084-112-8.

[9] Majherová, J. 2008. *Modelovanie rastlín na informatike.* [disk] In: Huraj, J. (ed.): Konferencia DIDINFO 2008. Banská Bystrica: UMB Banská Bystrica 2008. s. 41. ISBN 978-80-8083-556-9

AKTIVITY VO VIDEOEDITORE PRE II. STUPEŇ ZŠ

MGR. ART. KATARÍNA MIKOLAJOVÁ

Katedra základov a vyučovania informatiky, Fakulta matematiky, fyziky a informatiky Univerzity Komenského, Bratislava
e-mail: mikolajova@fmph.uniba.sk

Dnešní žiaci žijú v multimediálnom svete, a rovnako by takým malo byť i prostredie, v ktorom študujú. Rozvíjanie multimediálnej gramotnosti v škole sa stáva nevyhnutnou súčasťou vzdelávacieho procesu.

Prostredníctvom programu na spracovanie a editovanie videa a sme schopní vytvárať obrazové prezentácie, rozhlasové relácie, dokumenty, reklamy, edukačné videá, filmy alebo rozprávky... Žiaci pritom získavajú nové, príp. zdokonaľujú už nadobudnuté informatické zručnosti, a tiež rozvíjajú svoju kreativitu, komunikačné a kooperatívne schopnosti.

Kľúčové slová: *didaktika informatiky, digitálne médiá, digitálna gramotnosť, program na spracovanie videa*

1. ÚVOD

Byť gramotným v 21. storočí už neznamená len poznať abecedu, vedieť čítať a písať, gramotnosť by mala v sebe zahŕňať aj schopnosť porozumieť rôznym formám audiovizuálnej komunikácie, ovládať nové spôsoby sprístupnenia, spracovania a používania informácií [1].

Efektívna digitálna komunikácia je schopnosť vytvoriť presvedčivý odkaz použitím rôznych digitálnych médií (aj vizuálnych a iných multimediálnych komponentov) [2]. Použitie multimédií ponúka študentom cestu hľadania vlastného spôsobu sebavyjadrenia, ďaleko za hranicou len písaného textu [3].

Práve informatika by mala k rozvoju digitálnej gramotnosti výrazne prispieť. Jednou z možností (a ukazuje sa, že efektívnou) je tvorba a editovanie videa. Práca s videom rozvíja u žiaka viacero informatických zručností, jeho jazykové schopnosti, okrem toho dáva žiakovi i priestor slobodne a tvorivo sa vyjadriť.

V článku bližšie analyzujeme možnosti využitia digitálnych technológií, popisujeme dva programy na spracovanie videa, a niekoľko aktivít zameraných na tvorbu a editovanie videa. Niektoré z aktivít boli realizované na Gymnázium Grosslingová v Bratislave.

V príspevku sa snažíme nájsť odpovede na nasledujúce problémy:

- Prečo by mali žiaci študovať v multimediálnom prostredí a naučiť sa s multimédiami pracovať?
- Dokážu žiaci použiť digitálne technológie na to, aby vytvorili presvedčivý odkaz pre poslucháčov (nielen svojich spolužiakov)?
- Vedia myslieť kriticky, aby vedeli zvoliť tie správne prostriedky na komunikáciu so špecifickým publikom?
- Čo všetko nám práca vo videoeditore z hľadiska rozvíjania žiakových osobnostných kvalít a informatických zručností umožňuje?

2. DIGITÁLNE MÉDIA

Dnešní žiaci žijú v multimediálnom svete, a rovnako by takým malo byť i prostredie, v ktorom študujú. Rozvíjanie multimediálnej gramotnosti v škole sa stáva nevyhnutnou súčasťou vzdelávacieho procesu. Sprostredkúvanie informácií audiovizuálnou formou (súčasne použitý text, obraz, zvuk, video...) je pre žiakov atraktívnejšie ako klasický spôsob prezentovania informácií (tabuľa a krieda).

V súčasnosti je tvorba digitálnych médií omnoho jednoduchšia, ako bola pred niekoľkými rokmi. Hardvérové vybavenie k tomu potrebné je cenovo dostupné pre čoraz väčšiu skupinu ľudí, a taktiež na mnohých základných a stredných školách na Slovensku. Program na spracovanie videa – *Revelation Sight & Sound* – bol v rámci projektu *Infovek* dodaný na všetky základné školy zadarmo. Navyše, *Windows Movie Maker* je súčasťou rozšíreného OS Windows XP, tiež Windows Vista. Mnoho základných i stredných škôl už disponuje technickými prostriedkami na prácu s videoeditorom (slúchadlá, mikrofón, digitálny fotoaparát, alebo kamera).

Prostredníctvom videoeditora sme schopní vytvárať obrazové prezentácie, rozhlasové relácie, dokumenty, reklamy, edukačné videá, filmy alebo rozprávky... Žiaci si popritom zdokonaľujú svoje informatické zručnosti: správa súborov (práca so súbormi a priečinkami), spracovanie grafickej informácie a zvuku, práca s Internetom, vytváranie animácií, práca s textom, typy a veľkosti súborov, konvertovanie a kompresia súborov a iné. Za použitia multimediálnych prvkov môžu starostlivo vybrané príklady videoklipov, zvukov a obrázkov pomáhať žiakom rozvíjať ich *estetické cítenie*

3. RÔZNE PROGRAMY NA ÚPRAVU VIDEA

V tejto časti sa zameriame na stručné oboznámenie sa s dvoma vybranými programami na spracovanie videa. Realizácia aktivít prebehla na II.

stupni ZŠ, tomu aj zodpovedá charakter zvoleného softvéru.

Oba videoeditory disponujú približne rovnakou funkcionalitou:

- Nahrávanie – zachytávanie videa z externých zdrojov, analógových alebo digitálnych
- Editovanie – strih videa, vkladanie prechodov medzi klipy, pridávanie efektov a titulkov, zvuku, grafiky
- Export – vygenerovanie videa vo zvolenom formáte a na zvolené médium

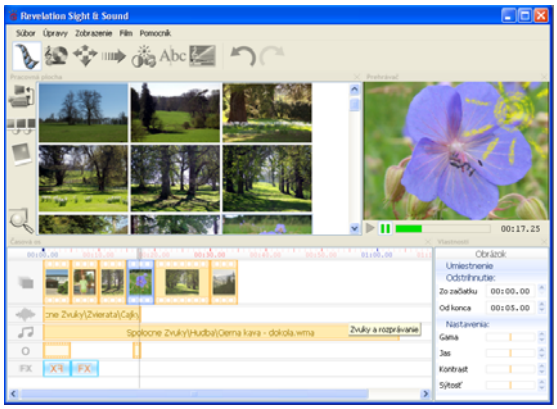
RELEVATION SIGHT&SOUND

Program na spracovanie videa – *Revelation Sight & Sound* – bol v rámci projektu *Infovek* dodaný na všetky základné školy zadarmo. Súčasťou inštalácie tohto programu sú dve príručky v elektronickej, jedna aj v tlačenej podobe, a editor Revelation Sight & Sound Obrázkový scenár na tvorbu obrázkových scenárov pre film. Stránka k programu v slovenskom jazyku http://www.logo.com/rss/sk/ obsahuje nápady pre tvorbu digitálnych filmov a zdroje, ktoré je možné využiť pri práci s Revelation Sight & Sound alebo iným videoeditorom.

Spôsob aplikovania rôznych efektov a prechodov funguje na princípe „ťahaj a polož“, čo žiakom umožňuje rýchlo a jednoducho modifikovať vzhľad a výsledný obraz videa. Použitím panorámy či priblíženia (zoom) môžu žiaci zo statických obrázkov a objektov vytvárať pohybujúce sa (efekt rostrum kamery).

Veľkou výhodou je, že súčasťou programu je množstvo obrázkov (aj transparentných), videí, zvukov, ktoré žiaci pri tvorbe videa môžu využiť. Súčasťou programu sú aj projekty, ktoré je možné ďalej editovať použitím vlastných nápadov a myšlienok.

Ako jeden z mála, poskytuje tento editor možnosť vytvárania obrázkových scenárov v aplikácií *Revelation Sight and Sound Obrázkový scenár*. O i. sa pri tvorení scenárov žiaci učia aj súslednosti a systematickosti. Súčasťou tohto editora je knižnica pozadí, postáv, predmetov, ktoré môžeme použiť, a tiež kresliaci panel na dotvorenie náčrtu scény. Okrem toho program umožňuje zmeniť prekrývanie jednotlivých objektov na scéne podľa toho, ktorý z nich je bližšie k divákovi. Jednotlivé predmety alebo postavy sa dajú zmenšovať alebo zväčšovať.

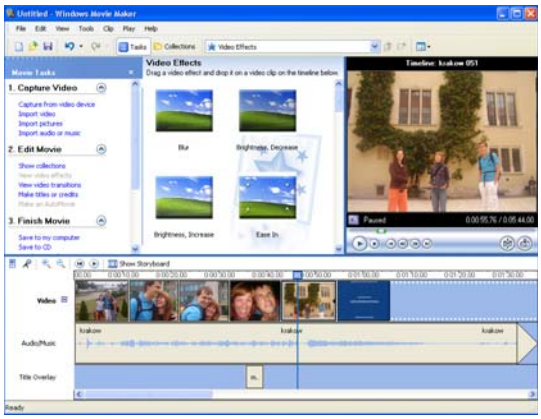


Obr. 1 Pracovné okno Revelation Sight&Sound

WINDOWS MOVIE MAKER

Windows Movie Maker je súčasťou rozšíreného OS Windows XP, tiež Windows Vista.

Z hľadiska vhodnosti tohto programu pre použitie v škole sa javí tento softvér ako prijateľný, pokiaľ je na škole inštalovaný OS Windows XP/Vista v slovenskom (prípadne v anglickom) jazyku. Neposkytuje však žiakom toľko možností na realizáciu, ako ponúka program Revelation Sight & Sound (vytváranie animácie za pomoci kamery, možnosť vytvárania obrázkového scenáru, použitie priesvitnej vrstvy, transparentný efekt, použitie panorámy alebo rôznych zvukových predlôh a fotografií, ktoré sú súčasťou programu...). Takisto aj dizajn je v porovnaní so softvérom Revelation Sight & Sound pre žiaka menej vhodný. Stále však patrí k jednoduchým programom určeným pre bežného užívateľa, preto ho hodnotíme ako vhodného kandidáta na prvé kroky pri spracovaní videa na II. stupni ZŠ.



Obr.2 Pracovné okno programu Windows Movie Maker

4. AKO PLÁNOVAŤ AKTIVITY

Na úvod by sme mali zvážiť doterajšie informatické zručnosti žiakov, zobrať do úvahy stanovené časové limity a podľa toho začať aktivity plánovať. Veľmi prospešné pre vyučovací proces je

poznať žiakov osobne, podľa toho ich potom vieme vhodnejšie zaradiť do rôznych skupín počas jednotlivých aktivít a stanoviť si reálnejší časový plán hodiny.

Mladší žiaci budú potrebovať presnejšie štruktúrovaný postup, zatiaľ čo starším študentom môže byť ponechaná väčšia „voľnosť“ a flexibilita. V oboch prípadoch platí, že čím viac kompetencií ponecháme žiakovi, tým viac času mu zaberie realizácia aktivity, ale tým rôznorodejšie a pestrejšie budú pravdepodobne výsledné projekty. Divergentné úlohy nabádajú žiakov k hľadaniu/vytvoreniu vlastnej štruktúry práce (čo im však zaberie určitý čas), zároveň sa väčší dôraz kladie na vlastnú kreativitu, ale i samostatný manažment práce, čo ich učí zodpovednosti. Naproti tomu aktivity, v ktorých učiteľ zadá tému s konkrétnym postupom aj návrhom, ako by mal výsledný projekt vyzerať, sú svojím spôsobom menej náročné, a preto pre mladších žiakov vhodnejšie, i keď neposkytujú taký priestor pre kreativitu a voľné experimentovanie [4].

Hľadanie vhodných a zaujímavých aktivít pre žiakov na prácu s videom je bezprostredne späté s rozvíjaním medzipredmetových vzťahov. Aktivity sa môžu dotýkať širokého okruhu vyučovacích predmetov, napríklad fyziky, chémie, cudzích jazykov, telesnej výchovy, slovenského jazyka a literatúry atď. Rovnako to môžu byť aktivity s voľnejšou témou – natočenie zábavnej reklamy, sprievodcu po škole, strašidelného príbehu...

5. NIEKOĽKO VYBRANÝCH AKTIVÍT

Prvé tri nasledujúce aktivity sú voľne preložené z angličtiny a boli dostupné na webovej stránke firmy Adobe Company [4].

VEDOMOSTNÝ KVÍZ

Vytvoríme si vlastnú vedomostnú súťaž. Divákovi položíme otázky a ponúkneme niekoľko odpovedí. Potom označíme správnu/nesprávnu odpoveď špeciálnym efektom – napríklad efektom zrnitého obrazu. Úlohou divákov je hádať, ktorá odpoveď bude označená za správnu/nesprávnu.

Do natáčania kvízu môžeme zapojiť dvoch alebo viacerých študentov. Každý študent bude ponúkať odlišnú odpoveď. Jeden zo študentov môže otázku čítať, alebo ju ukáže použitím titulkov. Otázka môže znieť napríklad takto: "Aká je normálna teplota ľudského tela?". Odpovede môžu byť ľubovoľné, napríklad:"37 °C" alebo "39°C". Takýmto spôsobom vytvoríme niekoľko otázok a k tomu prislúchajúce správne aj nesprávne odpovede.

MOJE STRAŠIDLO

Našou úlohou bude najprv natočiť, a potom použitím špeciálnych efektov vytvoriť krátky „hororový“ film. Strašidlom môže byť akýkoľvek objekt (napríklad chutné gumené cukríky), dôležité je vytvoriť správnu atmosféru za pomoci špeciálnych efektov a hudby.

Strašidlo: Pre tento film nemusí byť strašidlo samo o sebe hrôzostrašné – my z neho príšeru spravíme. Pokiaľ chceme, aby náš príbeh skončil šťastne, vyberieme strašidlo, ktoré môže byť porazené zaujímavým spôsobom. Futbalovú loptu napríklad sfúkneme.
Príbeh: Najprv predstavíme hlavnú postavu, s ktorou sa môžu diváci stotožniť, potom predstavíme strašidlo. To atakuje hlavnú postavu. Hlavná postava poráža strašidlo. Každý žije šťastne až do konca... okrem strašidla. Samozrejme je mnoho variácií na túto tému. Niekedy je strašidlo na konci hrdinom.
Natáčame video: Príbeh môžeme natáčať z uhla pohľadu strašidla. Záber hrdinu (alebo obete) je zaujímavejší, keď je snímaný zhora – to robí charakter viac zraniteľným. Naopak, záber zdola dodáva charakteru viac sily.
Editujeme video: V tomto bode svoj projekt naozaj ozvláštnime. Zábery z pohľadu strašidla môžu byť pôsobivejšie/strašidelnejšie pridaním červenej farby do obrazu. Takisto sa dá záber nakloniť alebo zdeformovať. Pridanie červenej farby do obrazu a vhodného zvuku (dýchanie, alebo tlkot srdca) je jednoduché a zároveň veľmi efektívne. Spomínané zvuky môžeme voľne stiahnuť z http://www.findsounds.com/ (niektoré sú súčasťou programu Revelation Sight & Sound).

ZA DVERAMI

Jedným zo silných nástrojov pri natáčaní videa je schopnosť vedieť, čo divák očakáva. To môžeme využiť tak, že diváka prekvapíme, vyľakáme alebo zmätieme. V tejto aktivite prevedieme diváka cez dvere, aby za nimi v zapätí objavil čosi neočakávané. Budeme natáčať na dvoch odlišných miestach, ktoré pri spracovaní vo videoeditore upravíme tak, aby sa divákovi zdalo, že sme natáčali na tom istom mieste

Najprv natočíme nášho herca pred dverami ľubovoľnej miestnosti. Študent môže stáť pred školskou knižnicou a rozprávať divákovi o jej obrovskej zásobe kníh. Ďalší záber na herca bude z knižnice – dvere sa otvárajú a užasnutý študent vstupuje do vnútra. Nasledujúci záber na miestnosť knižnice môže byť zamenený záberom z obrovskej univerzitnej knižnice, divák ostane rovnako prekvapený ako herec, ktorý sa ocitol v knižnici.

UČÍME SA ANGLICKY

V tejto aktivite majú žiaci za úlohu vytvoriť krátke edukačné video pre anglický (španielsky,

ruský...) jazyk. Premyslia si jednotlivé scenáre/situácie a pomocou programu *Revelation Sight & Sound Obrázkový scenár* vytvoria obrázky, v ktorých v cudzom jazyku pomenujú predmety na scéne, prípadne naznačia kreslený dialóg medzi postavami. Obrázky z programu pomocou klávesu *PrintScreen* zachytia, uložia, upravia (napr. pripíšu text do bublín), a importujú do videa. Program disponuje pomerne bohatou zásobou vlastných pozadí, predmetov, postáv – umožňuje vytvárať naozaj rôznorodé situácie/ scény. Okrem toho môžu žiaci video doplniť „živým" dialógom v cudzom jazyku – natočia video v triede (modelová situácia – zoznámenie sa). K jednotlivým scénam žiaci nahrajú komentár v cudzom jazyku.

Obr. 3 Ukážka práce žiaka – novinový titulok na klipe

6. ZÁVER

V súčasnosti nie je práca s programom na spracovanie videa na základných školách na Slovensku veľmi rozšírená. Aj keď sú takéto aktivity zväčša časovo i technicky pomerne náročné, zanietenie žiakov a výsledky ich práce svedčia o tom, že aktivity vo videoeditore by si počas vyučovania informatiky mohli, či odvážnejšie povedané mali nájsť svoje miesto (napríklad raz mesačne, štvrťročne...).

Vychádzajúc zo skúseností nadobudnutých počas realizácie niektorých aktivít na základných a stredných školách a osemročnom gymnáziu v Bratislave sa nazdávam, že aktivity využívajúce videoeditor sú pre deti veľmi inšpiratívne a obohacujúce. Počas ich realizácie si žiaci osvojujú nové, prípadne upevňujú už získané informatické zručnosti, ako je práca s textom (editovanie titulkov), práca sa súbormi (importovanie súborov, ich konverzia), nahrávanie komentára, vkladanie rôznych efektov na klipy, či prechodov medzi ne, práca s grafickým editorom, vyhľadávanie informácií na Internete, sťahovanie zvukových a obrázkových súborov z Internetu a iné.

Verím, že práca s videom osloví čoraz väčší počet učiteľov informatiky, ktorí sa takýmto spôsobom rozhodnú prispieť k rozvoju kreativity, počítačových zručností a multimediálnej gramotnosti svojich študentov. Vhodne zvolené aktivity dokážu žiakov *motivovať*, *inšpirovať*, postaviť ich pred *nové výzvy*, *uľahčiť* im *chápanie vzťahov a javov*, pomôcť im *hľadať* a *rozvíjať* v sebe *nové danosti*, ale aj učiť ich *tvorivo* nazerať na svet, ktorého sú sami súčasťou.

LITERATÚRA

[1] MULTIMEDIA LITERACY, dostupné na Internete 23. apríla 2008 <http://education.stateuniversity.com/pages/2186/Literacy-MULTIMEDIA-LITERACY.html>

[2] KRISTI AHO: *Digital communication for all students*, 2005, dostupné na Internete 20. januára 2009: <http://www.adobe.com/education/solutions/pdfs/dc_for_all_students.pdf>

[3] THE REPORT OF 21st CENTURY LITERACY SUMMIT, dostupné na internete 20. januára 2009: <http://www.adobe.com/education/digkids/lessons>

[4] CRAFT, A.: *An analysis of research and literature on Creativity in Education*, Report prepared for the Qualifications and Curriculum Authority, 2001, dostupné na Internete 11. februára 2008: <http://www.ncaction.org.uk/creativity/creativity_report.pdf>

[4] ŠNAJDER, Ľ., KIREŠ, M.: *Práca s multimédiami* (Informatika pre stredné školy), SPN – Mladé letá, s.r.o., Bratislava, 2005

[5] Webová stránka ADOBE COMPANY, Education, Digital Kids Club, *Digital video lessons*, dostupné na interenete 21. novembra 2007: <http://www.adobe.com/education/digkids/lessons>

[6] ŠNAJDER, Ľ., KIREŠ, M., KALAKAY, R.: *Multimédia pre učiteľa*, Asociácia projektu Infovek, 2002

[7] ROBINSON, K.: *Do schools kill creativity?,* TED Ideas Worth spreading – TEDBlogs, transkripcia prezentácie dostupná na Internete 17. decembra 2008: <http://blog.ted.com/2006/06/sir_ken_robinso.php>

OD ELEKTRONICKEJ UČEBNICE K UČENIU SA.

MGR. MILAN MORAVČÍK, MGR. JANKA PEKÁROVÁ

Katedra základov a vyučovania informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina, 84248 Bratislava
tel. 02/60295210, e-mail: moravcik@fmph.uniba.sk, pekarova@fmph.uniba.sk

Pedagogický výskum pomáha učiteľom odhaliť nové zákonitosti, súvislosti a dôležité faktory ovplyvňujúce kvalitu detského učenia sa, interakcií medzi žiakmi a učiteľom ako i vzťahu medzi obsahom učiva a formou, akou je prezentovaný. Pri tvorbe on-line učebnice ľubovoľnej znalostnej domény vyvstáva otázka: aký najvhodnejší spôsob reprezentácie informácií použiť? Čím uľahčíme študujúcemu pochopenie učiva a jeho neskorší transfer do praktických životných situácií? Implementácia výslednej podoby on-line učebnice ďalej stavia pred vývojára technologické otázky. Použité technológie môžu spätne vplývať na formu učebnice, preto je pre tvorcu užitočné ich poznať a vedieť sa v nich orientovať.

Kľúčové slová: *elektronická učebnica, vývojárske nástroje, učebnica a učenie sa*

1. ÚVOD

Takmer každý súčasný pedagóg vo väčšej či menšej miere používa elektronické výučbové materiály pre podporu vlastného vyučovania. Učitelia vytvárajú nové materiály, publikujú ich na internete a komunikujú o ich obsahu s ďalšími odborníkmi v oblasti vzdelávania (viď portály ako infovek.sk, modernaskola.sk, modernyucitel.net). Mnohí však pritom pristupujú k internetu - médiu založenému na rozvetvenosti a nelineárnej štruktúre, ako keby bol knihou so stránkami, odsekmi a vetami, ktoré plynú jedným smerom a vytvárajú tak celok [8]. Internet však ponúka oveľa širší potenciál pre poskytovanie učebného obsahu i vlastné učenie sa [2].

V článku preto

- hľadáme vodítka pre formu výučbového materiálu tak, aby prezentácia na webe vytvárala hodnotný a atraktívny priestor pre učenie sa,
- analyzujeme implementáciu elektronickej učebnice z hľadiska dostupných technológií a nástrojov.

2. DIGITÁLNA VERZUS KLASICKÁ UČEBNICA

Elektronická učebnica disponuje viacerými výhodami v porovnaní s klasickou, papierovou učebnicou v rovnakej znalostnej doméne. Klasická učebnica je [1]

- NErozširovateľná – informácie, ktoré obsahuje, sú v čase jej vydania uzatvoreným celkom,
- NEmodifikovateľná – po publikovaní sa stráca možnosť úprav a opráv, prípadne aktualizácie uvedených poznatkov,
- NEmultimediálna – priestor pre zahrnutie rôznych typov médií je značne obmedzený,
- NEadaptívna – obyčajne ponúka len jeden postup, pohľad na vec. Študent sa prispôsobuje učebnici, a nie naopak.

Navyše je častokrát izolovaná od ďalších zdrojov informácií [8], ktoré by mohli významne prehĺbiť porozumenie študenta a poskytnúť alternatívne pohľady na rovnakú problematiku, teda rozvinúť učenie sa do šírky i hĺbky.

Niektoré výhody elektronickej učebnice oproti klasickej sú zjavné. Český projekt Elektronická učebnice [5] napríklad zdôrazňuje nasledujúce vlastnosti:

- obrovský **rozsah**, **odkazy** na kompletné texty,
- **atraktívna forma** (usporiadanie, obrázky),
- **didaktické** spracovanie.

Pri adaptácii klasickej učebnice na digitálnu sme ako pedagógovia nútení zmeniť zaužívané postupy tvorby vlastných vzdelávacích materiálov. [3] však upozorňuje na časté riziko, že obsah nového prostredia je na počiatku vždy postavený na staršom prostredí a prístupe.

Ako by sme teda mali transformovať obsah klasickej učebnice tak, aby zodpovedal moderným koncepciám učenia sa?

3. DIGITÁLNA UČEBNICA A UČENIE SA

Naše vedomosti zodpovedajú asociatívnej štruktúre, v ktorej sú jednotlivé poznatky navzájom poprepájané. Digitálna učebnica nám umožní lepšie prezentovať takúto štruktúru poznatkov – typickou ukážkou je *hypertext*.

Nelineárnej štruktúre zodpovedá aj **myšlienková mapa**, ktorá zachytáva hĺbku, asociácie, vzťahy medzi jednotlivými poznatkami; demonštruje aktuálne smerovanie vednej disciplíny a ukazuje širšie súvislosti [8].

Vyučovanie sa v súčasnosti orientuje na žiaka. Webová aplikácia môže rešpektovať túto orientáciu, poskytnúť študentovi možnosť adaptácie učebného prostredia ([7], [8]) podľa jeho preferencií. Dizajnér by mal implementovať **sériu stratégií a zdrojov**, ktoré môže využiť študent preferujúci istý **štýl učenia sa** [7]. V ideálnom prípade je digitálny obsah ušitý na mieru pre každého študujúceho, zároveň má obrovský potenciál rozvíjať rôzne typy inteligencie (viď Obr.1) [7].

Obr. 1 Digitálny obsah by mal rozvíjať viaceré typy inteligencie (podľa [6])

V prípade digitálnej učebnice môžeme prinajmenšom zahrnúť do jej spracovania **multimediálny obsah**:

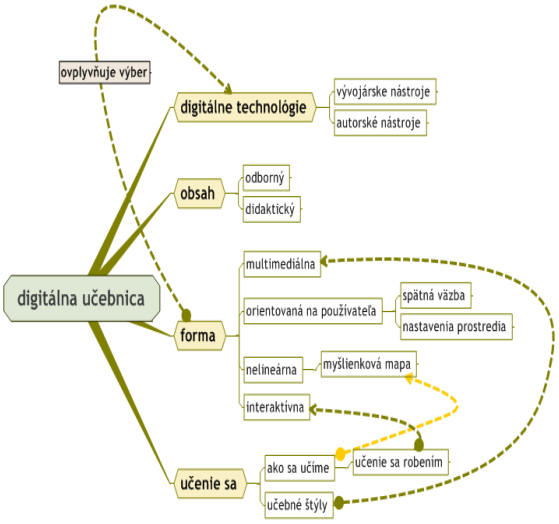
- vizuálne (obrázky, diagramy, schémy, interaktívne animácie),
- audio-vizuálne (autentické nahrávky, videá).

Hoci používatelia elektronickej učebnice ostávajú pre jej tvorcu niekedy anonymnými, môžeme myslieť na rozvoj ich intrapersonálnej aj interpersonálnej inteligencie. Posúdiť svoje porozumenie učebného obsahu môžeme krátkym testom, ktorý uzaviera tematický celok. Takýto prístup využívajú aj mnohé on-line kurzy. Študent by mal hneď po vyplnení testu získať **spätnú väzbu** k svojmu riešeniu. V prípade on-line kurzov si môže vystačiť s dvojškálovým hodnotením správne – nesprávne. Nejasnosti mu pomáha riešiť komunita učiacich sa a tútor. V prípade elektronickej učebnice by ohodnotenie odpovedí malo obsahovať hodnotenie s dôkladne vysvetľujúcou spätnou väzbou (nakoľko odborník na vzdelávanie v danej oblasti nemusí byť vždy dostupný). Automatické vyhodnocovanie odpovedí však v sebe skrýva pascu pre samotného tvorcu obsahu učebnice. Otázky by mali postupovať od jednoduchých úrovní učenia sa (znalosť, porozumenie) až po hlboké pochopenie (analýza, syntéza, hodnotenie). Vyššie úrovne učenia sa je však náročné, niekedy prakticky nemožné hodnotiť automatickými nástrojmi.

Obsah každej kapitoly by sa mal výrazne odlišovať od obsahu rovnomennej kapitoly v pôvodnej, neelektronickej učebnici. V priebehu celej kapitoly by sa mali vyskytovať aktivizujúce prvky – **motivačné otázky**, úlohy na doplnenie určitej informácie, prípadne opravu chýb v texte. [2] ponúka široké spektrum nápadov pre podporu interakcií medzi členmi e-learningového prostredia. Viaceré z nich možno modifikovať a použiť aj pre prípravu elektronickej učebnice:

- myšlienkové mapy,
- časové osi,
- demonštrácie a simulácie,
- mnemotechnické pomôcky,
- prípadové štúdie.

Časť príkladov v učebnici môže viesť ku komunikácii s odborníkmi v danej oblasti, k virtuálnym diskusiám so zahraničnými študentmi preberajúcimi rovnakú oblasť.



Obr. 2 Otázky vznikajúce pri tvorbe digitálnej učebnice

4. TECHNOLÓGIE PRE TVORBU DIGITÁLNYCH UČEBNÍC

Aké technológie ponúka súčasnosť pre tvorcov digitálnych učebníc? Odpoveď na takúto otázku je inšpiráciou pre mnohé záujmové skupiny. My sa pokúsime našimi odpoveďami zaujať bežného učiteľa, ktorý má záujem vytvoriť digitálnu učebnicu a zároveň mu pomôcť zorientovať sa v súčasnej širokej ponuke technológií. Naše odpovede sme rozdelili do štyroch skupín. Každá zo skupín vlastne predstavuje určitú úroveň technologickej náročnosti. Teda prvá skupina popisuje technologicky najnáročnejší spôsob tvorby

digitálnej učebnice. Posledná úroveň zas ten najľahší.

1.) vytvorenie učebnice "vo vlastnej réžii"

V tomto prípade musí byť učiteľ zároveň dobrým programátorom. Je to technicky najnáročnejší spôsob tvorby učebnice, na druhej strane je najflexibilnejší a dokáže sa prispôsobiť akýmkoľvek nápadom autorov. Vyžaduje si však znalosť webových programovacích a skriptovacích jazykov, knižníc, štandardov a poznanie základných princípov tvorby internetových stránok. Jeho nasadenie v praxi prináša niekoľko rizikových faktorov. Keďže ide o internetovú aplikáciu, najvážnejším rizikom je bezpečnosť. Tým, že programátor buduje čoraz viac interaktívnejší portál, poskytujúci dostatočnú spätnú väzbu, hodnotiace formuláre a ďalšie interaktívne prvky, riziko bezpečnosti exponenciálne rastie, je často podceňované a môže spôsobiť problémy nielen autorovi stránky, ale aj jej návštevníkom.

Takýto druh tvorby je určený predovšetkým skúseným programátorom, ktorí ovládajú:

- *značkovací jazyk* – umožňuje sémantický (HTML) a prezentačný (CSS) zápis textových informácií, ale aj iných častí (obrázkov, odkazov, kódu skriptovacích jazykov a pod.) jednotlivých stránok portálu.
- *skriptovací jazyk* – umožňuje tvoriť dynamické stránky, ktorých obsah je spracovaný na strane klienta (napr. JavaScript) alebo servera (napr. PHP, Java).
- *prácu s databázami* – umožňuje efektívnym a zmysluplným spôsobom uchovávať a spracúvať veľké množstvo údajov (jednoduché riešenia sú napr. MySql, SqlLite).
- *základné štandardy a zásady* – umožňujú navrhnúť modulárnu a bezpečnú webovú aplikáciu.

2.) vytvorenie učebnice s využitím systému WAF

Programovanie "vo vlastnej réžii" sa po čase stáva neprehľadným a zdrojový kód narastá do obrovského svojvoľne sa vyvíjajúceho klbka. Programátori webových aplikácií bežne predchádzajú takémuto stavu použitím rôznych tzv. **WAF[1] systémov**, ktoré im umožnia lepšie manažovať programátorskú prácu a spravovať časti programátorského kódu. Takéto systémy obsahujú viaceré návrhové vzory (tzv. design patterns).

[1] V našom kontexte považujeme **WAF (Web Application Framework) systém** za softvérovú štruktúru, ktorá slúži ako podpora pri vývoji webových aplikácií. Môže obsahovať podporné programy, knižnicu API, návrhové vzory alebo odporúčané postupy pri vývoji. [cit. on-line http://cs.wikipedia.org/wiki/Framework]

Medzi často používané návrhové vzory patrí model MVC (Model-View-Controller), ktorý organizuje architektúru aplikácie. V princípe spočíva vo vytvorení troch navzájom nezávislých komponentov, z ktorých obmena jedného má len minimálny vplyv na zvyšné komponenty. Programátor tak môže v aplikácii izolovane modifikovať prezentačnú vrstvu (View), vrstvu pre spracovanie údajov (Model) a vrstvu, ktorá organizuje udalosti (vstupy od užívateľa) a ovplyvňuje zmeny v prezentačnej a údajovej vrstve (Controller). Návrhový vzor MVC sa stal nosným pilierom viacerých WAF systémov, pre krátkosť spomenieme len niektoré z nich:

WAF	Jazyk
CodeIgniter, CakePHP, Symfony	PHP
Swing, Cocoon	Java
TurboGears, Web2py	Python
Ruby on Rails	Ruby
PureMVC, WaveMaker	JavaScript

Mnohé z týchto systémov používajú rôzne iné návrhové vzory na riešenie často vyskytujúcich sa technických problémov programovania. Programátor tak získa väčší časový priestor na tvorbu jadra svojej práce.

Použitie WAF systémov vyžaduje znalosť:

- základných programátorských webových techník,
- návrhových vzorov a princípov, s ktorými WAF systém pracuje,
- základných štandardov a princípov fungovania internetových aplikácií.

3.) redakčný systém

Ak učiteľ nie je programátorom, nemá čas a záujem sa ním stať, môže použiť na tvorbu internetového portálu **redakčný systém**. Sám redakčný systém je vlastne internetová aplikácia, ktorá v najjednoduchšom prípade slúži na tvorbu internetových prezentácií. Užívateľ taktiež veľmi rýchlo vytvorí jednoduchý internetový portál, s množstvom textov, obrázkov, bez znalosti programovania.

V súčasnosti existujú redakčné systémy rôznych úrovní. Hoci už v najjednoduchších z nich vytvorí užívateľ celý portál, väčšinou ide o statické stránky, ktoré majú skôr podobu webových prezentácií. Najnáročnejšie redakčné systémy dokážu produkovať rozsiahle webové portály, ktorých dynamika a forma prakticky nemá hraníc. Dokážu spracovať veľké množstvo údajov, výsledky spracovania zobraziť v rôznych podobách, organizovať práva a roly registrovaných užívateľov, a pod. Odradzujúcim faktorom je čas, ktorý treba vynaložiť na naučenie sa takto zložitého systému.

(medzi najznámejšie voľne šíriteľné redakčné systémy patria: Joomla, Drupal, Typo3, Plone)

Špeciálnu kategóriu tvoria tzv. **wiki** systémy. Slovíčkom wiki možno označiť každý webový portál, ktorý umožňuje návštevníkom upravovať jeho obsah. Toto isté slovíčko však možno použiť na pomenovanie softvéru, ktorý produkuje takéto – „návštevníkmi modifikované“ portály. Veľakrát sú wiki systémy kritizované pre nízku relevantnosť informácií. Ktorýkoľvek záškodník môže na stránky vložiť zavádzajúce informácie resp. vymazať už ich existujúci obsah. Tento nedostatok sa súčasné wiki systémy snažia riešiť registrovaním užívateľov, odobrením nových článkov moderátormi (typickou ukážkou je napr. Wikipédia). Najväčší význam má nasadenie wiki v úzkej komunite ľudí, ktorá dozerá na jej obsah, prípadne ho koriguje a dopĺňa.

Wiki-softvér	Jazyk implementácie
JAMWiki, SnipSnap	Java
DotWiki, FlexWiki	.NET/Mono
GetWiki, PhpWiki	PHP
WyPy, PikiPiki, Zwiki	Python
Hiki	Ruby

Z pohľadu tvorcu webového portálu použitie redakčného systému vyžaduje:

- investovať čas do nainštalovania systému,
- naučiť sa so systémom pracovať,
- poznať najzákladnejšie zásady tvorby webových stránok.

Čo nemusí tvorca vedieť:

- ovládať žiaden skriptovací či značkovací jazyk,
- pracovať s databázami pomocou databázových programovacích jazykov,
- riešiť otázky zabezpečenia aplikácie pred hackerskými útokmi.

5. ZÁVER

Digitálna učebnica rozširuje potenciál klasickej učebnice. Pri hľadaní transformácie pôvodného vzdelávacieho materiálu do webovej formy riešime konceptuálne i technologické otázky, ktoré vplývajú na kvalitu výsledného vzdelávacieho produktu. Prezentovali sme niekoľko postupov, ktoré pomôžu učiteľovi – vývojárovi digitálneho obsahu – robiť lepšie, kvalifikovanejšie rozhodnutia pri výbere technológie pre vývoj digitálnych vzdelávacích materiálov a ich adaptácii pre webové prostredie.

LITERATÚRA

[1] Agh, P.: Efektívne využívanie počítačových systémov a Internetu vďaka znalosti princípov. In: Hrušecká, A., Mandíková, K. a Kalaš, I. (Eds.) Zborník príspevkov 1. celoštátnej konferencie. Bratislava: UIPŠ MŠ SR, 2001, pp. 128 – 135, ISBN 80-7098-265-5

[2] Fisher, M.: Designing Courses and Teaching on the Web. Oxford: Scarecrow Education, 2003. ISBN 1-57886-052-0

[3] Hrušecký, R.: E-learning vo vzdelávaní budúcich učiteľov informatiky. Písomná práca k dizertačnej skúške. Bratislava: FMFI UK, 2006.

[4] Kadlec, V.: Agilní programování. Brno: Computer Press, 2004. ISBN: 80-251-0342-0

[5] Projekt Elektronická učebnice http://www.eucebnice.cz/literatura/egypt.html [on-line, cit. 27.1.2009]

[6] Reid, D.: Learning Styles and Inclusion. London: Paul Chapman Publishing, 2005. ISBN 1-4129-1064-1

[7] Sims, R.: Beyond instructional design: Making learning design a reality. *Journal of Learning Design*, 1(2), 2006, pp. 1-7. ISSN: 1832-8342

[8] Turner, M.: Contemporary Approach to Writing Non-Linear Online Learning Resources. Journal of Learning Design, Vol. 2 (2), 2007, pp. 56 - 69. ISSN: 1832-8342

Tento príspevok vznikol v rámci riešenia projektu KEGA číslo 3/6093/08 "Elektronická učebnica metodológie pedagogického výskumu".

POČÍTAČOVÁ GRAFIKA A JEJ EDUKÁCIA

MGR. DANIELA ONAČILOVÁ

Katedra základov a vyučovania informatiky, Fakulta matematiky, fyziky a informatiky,
Univerzita Komenského, Mlynská Dolina, Bratislava,
tel. 02/602 95 611, e-mail: onacilova@fmph.uniba.sk

Vyučovanie počítačovej grafiky sa stále viac stáva témou, ktorú nemožno prehliadnuť. Konferencie ako EUROGRAPHICS alebo SIGGRAPH pre počítačovú grafiku, podporujú príspevky so zameraním na vyučovanie. Mnohé svetové univerzity vyučujú tento predmet z pohľadu informatiky alebo umenia. Počítačová grafika môže byť aj prostriedkom ako učiť efektívnejšie.

Tento článok sa venuje stručnému prehľadu vyučovania počítačovej grafiky. Pozrieme sa na sylaby ACM SIGGRAPH, podmienky vyučovania v našom školskom systéme a na FMFI UK a nakoniec sa pozrieme na svetové univerzity a ich kurzy „Introduction to Computer Graphics“.

Kľúčové slová: *počítačová grafika, vyučovanie počítačovej grafiky*

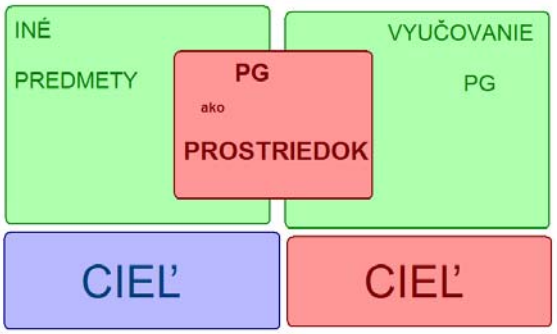
1. ÚVOD

V informatickej oblasti a vo vyučovaní informatiky sa nedá úplne vyhnúť oblasti počítačovej grafiky. Už len z toho dôvodu, že zobrazovanie obrazovej informácie v digitálnej podobe je súčasťou každodenného používania počítača. Táto oblasť nadobúda veľký význam. Venujú sa jej mnohé svetové univerzity, organizácie a firmy. Neustále sa vyvíja a nachádza si nové a nové uplatnenie. S počítačovou grafikou sa môžeme stretnúť vo filmovom priemysle, v medicíne, vo výrobe, v kultúrno-štátnom sektore (tvorba virtuálnych miest) v umení a vo vyučovaní. Také široké zastúpenie potrebuje nutne svojich odborníkov. A tých treba vyškoliť.

Predkladáme dve definície, z ktorých prvá je podľa organizácie ISO. **Počítačová grafika** (PG/CG) ako časť informatickej vedy sa zaoberá metódami a technikami konštrukcie, manipulácie, ukladania a zobrazovania obrazov pomocou počítača [1]. Podľa svetovej organizácie ACM a jej komisie nazývanej SIGGRAPH Education Committee's Curriculum Working Group je počítačová grafika spojením umeleckých a technických teórií a zručností [2].

Počítačová grafika, ako sme už spomenuli má široké uplatnenie. Pomocou nej možno riešiť problémy, ako vytváranie virtuálneho mesta, múzeí, simulovanie fyzikálnych javov, modelovanie živých organizmov a diagnostikovanie chorôb. Vyučovanie počítačovej grafiky je momentálne horúcim problémom, ktorému sa venujeme aj my. Význam PG nespočíva len v tom, aby sme vyškolili "dobrých grafikov". Počítačová grafika nemusí byť len **cieľom** výučby, ale aj **prostriedkom**.

V nasledujúcom sa budeme venovať sylabám ACM SIGGRAPH, vyučovaniu PG na stredných školách a na FMFI UK a nakoniec sa pozrieme na dvanásť kurzov svetových univerzít z USA a Kanady.



Obr. 1 Grafické znázornenie využitia PG z pohľadu vyučovania.

2. VYUČOVANIE POČÍTAČOVEJ GRAFIKY

Najväčšia svetová spoločnosť pre vzdelávanie a vedu v oblasti informatiky – ACM "Association for Computing Machinery" – obsahuje špeciálnu záujmovú skupinu počítačovej grafiky SIGGRAPH "Special Interest Groups Computer Graphics and Interactive Techniques" (Špecializované skupiny – počítačová grafika a interaktívne techniky) [3]. V časti "Education" sa dočítame o takzvanej "Education Committee", ktorá má za cieľ podporiť vyučovanie počítačovej grafiky a využívanie počítačovej grafiky pri vyučovaní. Jedným z jej počinov je aj server **CGEMS** [4] určený na zhromažďovanie elektronických pomôcok. Tam môžu prispievať učitelia, študenti a nadšenci v tejto oblasti. Ďalším výstupom sú dvoje syláb počítačovej grafiky. Jedny sú určené umeleckým školám [5] a druhé všeobecne počítačovej grafike ako spojeniu umenia a informatiky [2]. Nás zaujímajú tie druhé "**A Knowledge Base for the Emerging Discipline of Computer Graphics**", pretože sa venujú

počítačovej grafike z informatického hľadiska. Oblasti, ktoré patria do kurikula, sú nasledovné:

CGF: Fundamentals – Základy – všeobecne o počítačovej grafike, jej oblastiach, histórii, využití

PI: Professional Issues – Otázka profesionality – duševné právo, rešpektovanie kultúry, etické problémy...

CGP: Physical Sciences – Fyzikálna veda – dôkladné poznanie fyzikálnej stránky sveta pre správnu reprezentáciu javov ako napr. svetelné efekty, pohyby, fyzikálne procesy...

CGM: Math – Matematika – neoddeliteľná súčasť, vedomosti nevyhnutné pri používaní softvéru, porozumení a používaní matematických štruktúr a algoritmov. Nevyhnutné pre ďalší výskum

PC: Perception and Cognition – Vnímanie a poznávanie – porozumenie ľudskému vnímaniu – prispôsobenie PG algoritmov pre lepšie využívanie

HCI: Human Computer Interaction – Interakcia človek–počítač – uľahčenie interakcie s počítačom, porozumenie HCI - jej základné problémy, metodológie a jazyk (dorozumievanie sa s počítačom)

Programming & Scripting – Programovanie a tvorba ovládačov – programovanie knižníc, grafických softvérov, ovládačov grafických kariet – nutnosť ovládať programovací jazyk, možnosť používateľa upravovať si jednotlivé funkcie programu

CGA: Animation – Animácia – animovaný film, špeciálne efekty pre kameru, počítačové hry... vytváranie, porozumenie technikám vzniku

CGR: Rendering – Vykreslenie – proces vytvárania obrázka z 3D popisu scény

CGM: Modeling – Modelovanie – matematický opis scény, matematické operácie pri vytváraní objektov

CGH: Graphics Hardware – Grafický hardvér – grafické systémy, základné zariadenia, periférne zariadenia, efektívne využitie dostupných systémov a vytváranie budúcich

DI: Digital Images – Digitálny obraz – výstupom rôznych grafických systémov je obrazová informácia, porozumenie teórii obrazu pre základnú analýzu a manipuláciu s obrazovou informáciou

COM: Communications – Komunikácia – sprostredkovanie príbehu, komunikácia vizuálnou informáciou

CUL: Cultural Perspectives – Kultúrna perspektíva

ART: Art and Design Foundation – Základ umenia a dizajnu – význam vývoja digitálnych médií pre umenie a dizajn, tradícia a nové trendy IT, prínosy a význam IT pre umenie a dizajn

RTG: Real-Time Graphics – Grafika v reálnom čase – snaha o vytvorenie efektívnych prostriedkov pre zobrazovanie v reálnom čase, hardvér – ovládače – softvér (algoritmy)

CGA: Advanced Topics – Pokročilé témy

3. NAŠE PODMIENKY

Žiaci sa v našom školskom systéme **prvýkrát stretnú** s počítačovou grafikou na vyučovaní informatiky, na príbuznom krúžku alebo na základnej umeleckej škole. Venujú sa softvéru na kreslenie, upravujú fotografie alebo inak vytvorené obrazové informácie [6][7]. V matematike sa môžu stretnúť s rysovacím softvérom pre geometriu [8]. Niektorí žiaci programujú grafický 2D a 3D výstup v podobe hier dokonca pomocou profesionálnej grafickej knižnice [9] alebo pomocou povelov korytnačky v Imagine [10].

Na **vysokej škole** sa odbory priamo zameriavajú na počítačovú grafiku alebo predmety, ktoré sa venujú len nejakej oblasti počítačovej grafiky. Napríklad kreslene technických výkresov, urbánne modelovanie existujúcich alebo budúcich budov a simulácie [12], počítačom podporované konštruovanie mechanických a elektronických zariadení [11][13], konzervatóriá s kurzami pre vytváranie počítačovej 2D/3D animácie [14] a iné.

V úvode sme ukázali, že na počítačovú grafiku a s ňou spojené vyučovanie sa môžeme pozerať z dvoch pohľadov. Ako na prostriedok a ako na cieľ. Na to, **aby sa stala prostriedkom** pri vyučovaní, musia byť učitelia oboznámení s týmto odborom a jeho možnosťami. Na našej fakulte vyučujeme dva predmety "**Aplikácie počítačovej grafiky**" a "**Počítačová grafika**" ako povinne voliteľné v odbore učiteľstva informatiky. Cieľom prvého je „autorská tvorba a publikovanie digitálneho obsahu pomocou voľne dostupných nástrojov so zameraním na všeobecný prehľad a rozvoj zručností“ a cieľom druhého „poskytnúť základné vedomosti o štruktúre, postupoch a možnostiach počítačovej grafiky s dôrazom na algoritmické myslenie a praktické aplikácie“.

Počítačová grafika ako cieľ vystupuje na UK v troch študijných odboroch, matematike, informatike a aplikovanej informatike. Matematický odbor sa na PG pozerá hlavne z pohľadu matematiky a geometrie. Informatika a aplikovaná informatika z pohľadu implementácie algoritmov. Matematika a informatika sú v oboch odboroch nevyhnutné na porozumenie.

Teraz si bližšie rozoberieme jeden z úvodných kurzov pre matematikov s názvom "**Úvod do počítačovej grafiky**". Kurzom chceme „poskytnúť úvodnú predstavu o motivácii a štruktúre počítačovej grafiky a jej najdôležitejších aplikácií s dôrazom na algoritmický prístup k riešeniu potrebných problémov“ [15]. **Obsahom** kurzu sú nasledovné témy.

1. Počítačová grafika, definícia, konceptuálny rámec, história, základné problémy a ich riešenie.

2. Transformácie v rovine.
3. Základy špecifikácie grafických systémov a noriem.
4. Grafické objekty, ich vytváranie, kódovanie, modifikácia.
5. Rasterizácia.
6. Orezávanie a prieniky.
7. Základné algoritmy spracovania obrazu.

Kurz a prednášky sú vedené skúseným vyučujúcim počítačovej grafiky doc. RNDr. Andrejom Ferkom, PhD. Ku kurzu prebiehajú cvičenia za počítačom. Vzhľadom na počet študentov vzniká niekoľko skupín.

Hodnotenie predmetu delíme na niekoľko častí. Projekt, midterm a záverečný test. Najväčšiu váhu má projekt 85% a oba testy spolu 15% hodnotia len teoretické znalosti. Študenti si môžu vybrať zo štyroch rôznych zadaní projektu.

1. Vytvorenie a publikovanie digitálneho obsahu na internete - 3D model budovy, panoráma, prepojenie s google-earth a wikipediou.
2. Textúrovanie budov vo Virtuálnej Bratislave.
3. Vypracovanie šiestich programátorských zadaní.
4. Vlastné programátorské zadanie na špecifickú tému.

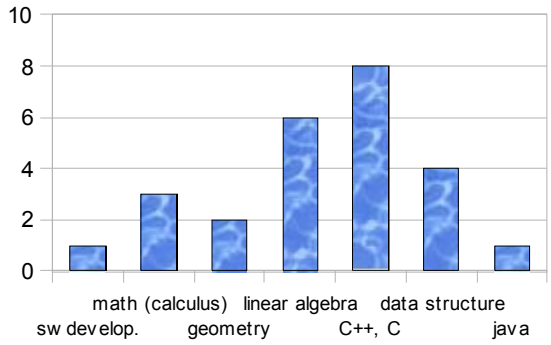
Najpopulárnejšie je prvé zo zadaní. Témy sa konzultujú a odovzdávajú cvičiacim. Praktické skúsenosti na vytvorenie projektu nadobúdajú študenti počas cvičení a teoretické počas prednášok.

Tento predmet prebieha v druhom semestri prvého ročníka, a preto si musíme uvedomiť, že niektorí študenti neovládajú programátorskú zručnosť. Prehlbovanie ich programátorskej skúsenosti z PG sa zabezpečuje počas magisterského štúdia.

4. SVETOVÉ UNIVERZITY

Niektoré zo svetových univerzít zverejňujú svoje kurzy na internete. Vybrali sme si 12 z nich a urobili sme krátky prehľad. Ide výslovne o kurzy „**Introduction to Computer Graphics**“ teda úvod do počítačovej grafiky. Sú to kurzy škôl Boston, Harvard, Oregon State, Stanford University a University of Virginia, Washington, Waterloo, Utah, British Columbia, California a Penn Engineering, Baskin Engineering.

Prvý údaj, ktorý nás bude zaujímať je, aké vstupné vedomosti by mali študenti ovládať Obr. 2. Tieto podmienky označujeme **prerekvizity**.



Obr. 2 Početnosti jednotlivých prerekvizít ku kurzom. Čísla označujú počet kurzov, ktoré danú prerekvizitu vyžadujú.

Je jasne vidieť, že dominuje znalosť programovacieho jazyka 9 z 12, a za ňou znalosť lineárnej algebry 6 z 12. Množiny „C++, C“ a „java“ sú vzhľadom na školy disjunktné.

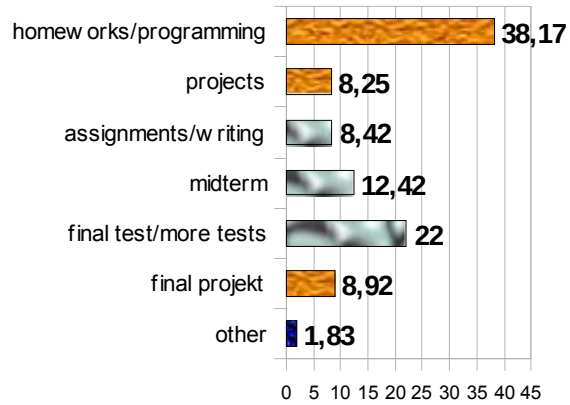
Obsahy jednotlivých kurzov sú zväčša veľmi podobné, zamerané na algoritmickú a teoretickú časť kam patria metódy zobrazovania a tiež geometria a matematika. Niektoré kurzy sa venujú aj úvodu do programovania.

Z nasledujúceho grafu vidieť priemerné **bodové ohodnotenie** práce a vedomostí študentov. Polovica škôl ohodnocuje programátorskú prácu počas semestra 50-70% bodov kurzu (homeworks/programming). Z celkového pohľadu je ohodnotenie programovania 55,33% a písomného prejavu 42,84% len mierne prevážené.

programátorské úlohy

písomné úlohy

Obr. 3 Percentuálne hodnotenie práce a vedomostí študentov v USA/CA.



Aj keď nie všetky kurzy hodnotili každú z uvedených kategórií, zvolili sme priemerovanie počtom všetkých škôl. Výhodou je celkový súčet 100% a tým možnosť komplexnejšieho porovnania.

5. CIELE

Za sebou máme stručný prehľad o vyučovaní počítačovej grafiky vo svete a na našej fakulte (FMFI). Pred nami je **výskum** zameraný na spomenutý kurz „Úvod do počítačovej grafiky“. Našim cieľom je začleniť sa do kurzu ako výskumník a zozbierať dostatok informácií. Tie by nám mali pomôcť identifikovať nedostatky vo výučbe a ukázať pohľad začiatočníka na PG a jeho vyučovanie. Je nutné najprv vytvoriť profil vedomostí študenta a na jeho teoretickom základe stavať ďalšie vedomosti. Kurz už prebieha niekoľko rokov, takže budeme môcť porovnať a zistiť či naše návrhy na vylepšenie sú efektívne. Po ukončení výskumu sa pokúsime o zovšeobecnenie postupu pri skvalitňovaní podobných kurzov PG pre začiatočníkov.

6. ZÁVER

V úvode článku sme si povedali, čo je to počítačová grafika a o motivácii k jej výučbe. Pozreli sme sa na vyučovanie počítačovej grafiky z pohľadu skúsenej a uznávanej organizácie ACM. Ukázali a stručne sme popísali sylaby navrhované SIGGRAPH Education Committee's Curriculum Working Group. Ukázali sme, ako sa snažíme napĺňať funkciu PG ako prostriedku a cieľa. Popísali sme si jeden z kurzov pre začiatočníkov v odbore matematika. Náš posledný pohľad bol upriamený na svetové univerzity a ich kurzy. Pred nami je výskum, ktorý by mal zlepšiť kvalitu jedného z kurzov PG pre začiatočníkov. Na jeho konci na pokúsime zovšeobecniť postup pri skvalitňovaní podobných kurzov PG pre začiatočníkov.

Tento výskum je čiastočne podporovaný z grantov VEGA 1/0763/09 Geometry Processing for Virtual Reality, UK Didaktika vyučovania počítačovej grafiky. Ďakujem recenzentom za podnetné pripomienky.

LITERATÚRA

[1] Ružický, Eugen a kol. Počítačová grafika a spracovanie obrazu. Sapientia, 1995. s. 3. ISBN 80-967180-2-9

[2] Orr, Genevieve a kol. A Knowledge Base for the Emerging Discipline of Computer Graphics. 9/7/2007. <http://cgems.inesc.pt/ModuleInfo.aspx?id=41>

[3] ACM. SIGGRAPH. [31.01.2009] <http://www.siggraph.org/about >

[4] SIGGRAPH Educational Committee, Eurographics Education Board. CGEMS. Mar 07/03. <http://cgems.inesc.pt/> <http://education.siggraph.org/resources/cgems/best-2008 >

[5] Tanzillo, Michael a kol. Digital Art Curriculum Framework v0.2. Last modified 2008-08-05. <http://education.siggraph.org/resources/artscurric/framework/view>

[6] Blahová, Viera. Vzdelávací štandard z informatiky pre gymnázium, štvorročné štúdium. 5. aplikácia grafický editor. ŠPU, 2002. <http://www.statpedu.sk/buxus/generate_page.php_page_id=633.html>

[7] Základná umelecká škola Imra Weinera Kráľa. Výtvarný odbor – počítačová grafika, počítačová grafika v kombinácii s klasickými výtvarnými technikami. [31.01.2009] <http://www.zusiwk.sk/vytvarny-odbor/ >

[8] Žilková, Katarína. Cabri geometria vo vyučovaní matematiky na strednej škole. [31.01.2009] [on–line] <http://www.matika.indatex.sk/index.htm>

[9] Belan, Anino. Knižnica OpenGL, učebný text pre oktávu osemročného gymnázia. Bratislava, 2007.<http://www.smnd.sk/main/vyucovanie/informatika>

[10] Blaho, Andrej, Salanci, Ľubomír. Tvorivá informatika – 1. zošity o obrázkoch a o práci s textom. Banská Bystrica, DidInfo 2008. s. 14. ISBN 978-80-8083-367-1

[11] SPU v Nitre, Mechanizačná fakulta. Informačná a automatizačná technika v kvalite produkcie. 523M402 Počítačová simulácia elektrických obvodov. [2009/10] <http://www.mf.uniag.sk/study/a_rok_09_10/Studijne_programy_Ing.pdf>

[12] TU v Košiciach, Stavebná fakulta, Ústav budov a prostredia. Informačný list predmetu I-502210 Simulácie v budovách. [31.01.2009] <http://www.svf.tuke.sk/images/stories/dokumenty/IL/IL_ING/IL_ING_TBP/II%20r/ZS/I-502210%20Simulacie%20v%20budovach.doc>

[13] TnUAD v Trenčíne, Fakulta špeciálnej techniky. Výrobné technológie – I. stupeň. 1-3510 Počítačom podporované konštruovanie I. [31.01.2009] <http://fst.tnuni.sk/files/image/fst/studijne_programy/vyrobne_technologie.pdf>

[14] Filmová a televízna fakulta VŠMU. Rozvrh hodín - ateliér animovanej tvorby – Bc. stupeň. Počítačové 2D animácia, Počítačové 3D animácia. Bratislava, 2008. <http://www.ftf.vsmu.sk/files/Animovan%C3%A1%20tvorba.doc>

[15] FMFI UK Bratislava. Matematika, 1-MAT-180 Úvod do počítačovej grafiky. Bratislava, 25.02.2008. <http://www.fmph.uniba.sk/fileadmin/user_upload/editors/studium/bc_mgr/IL/1-MAT-180.html>

VYUŽITIE POČÍTAČOVÝCH HIER VO VÝUČBE PROGRAMOVACIEHO JAZYKA IMAGINE.

ING. HEDVIGA PALÁSTHY, DOC. ING. IGOR ČERNÁK, PHD.

Katedra informatiky, Pedagogická fakulta, Katolícka Univerzita v Ružomberku, Nám. A. Hlinku 56/1, Ružomberok, tel. 044/432 09 60, e-mail: hp@pf.ku.sk, Igor.Cernak@pf.ku.sk

Abstrakt: Hra neodmysliteľne patrí do života detí, ale aj dospelých a je realitou, že prvé stretnutia detí s počítačom sú vždy spojené s hrou. Jej zásluhou strácajú zábrany a obavy z práce s neznámym prístrojom, postupne sa počítač stáva súčasťou detského sveta. Počítačové hry sú atraktívne, často veľmi náročné a zložité, budujú sebavedomie, dávajú pocit ovládania určitého priestoru. Učiteľ informatiky môže túto výhodu využiť vo vyučovaní a správnym výberom vhodných počítačových hier motivuje žiakov, zvyšuje ich záujem o predmet. Autori v článku uvádzajú niekoľko námetov pre využitie počítačových hier vo výučbe programovacích jazykov.

Kľúčové slová: *hra, didaktická hra, projekt, programovanie*

1. ÚVOD

Didaktický projekt je silný nástroj na zatraktívnenie a zefektívnenie vyučovania a myšlienka implementácie IKT do vyučovania nie je učiteľom cudzia. Niekedy učiteľov odradí technická stránka, komplikovaná inštalácia, spustenie alebo ukončenie programu. Spustiť program by mala byť zručnosť na úrovni „zapnúť rádio“. Inštrukcie typu „program je zazipovaný“, „stiahnite z internetu, „k spusteniu potrebujete...“. sú z pohľadu neinformatikov často nevykonateľné.

To, že program spĺňa kritéria kvality nie je zárukou, že učiteľ ho bude vedieť a chcieť používať. Schopnosť posúdiť kvalitu softvéru a vybrať vhodný softvér pre dané učivo je podmienené odbornou a metodickou prípravou učiteľov. Cieľom informatizácie by mal byť podporovať využívanie softvéru aj iné zložky IKT v jednotlivých predmetoch.

2. POČÍTAČOVÁ HRA A ALGORITMIZÁCIA

Algoritmizácia je schopnosť samostatne vytvoriť algoritmy. Zvládnutie algoritmizácie znamená poznať prostriedky algoritmického jazyka, schopnosti realizátora algoritmu a vedieť ich uplatniť pri tvorbe algoritmov. Nie je orientovaná len na vzťah k výpočtovej technike, ale ku komunikácii s prostredím vôbec. Rozvoj schopností algoritmizovať daný problém je jednou z najdôležitejších úloh výučby programovania. Algoritmizácia aj programovanie je tvorivá činnosť a záleží len na intelekte, skúsenostiach, zručnostiach, znalostiach a niekedy aj na intuícii.

Tvorba algoritmov je považovaná za veľmi dôležitú tvorivú činnosť, základy ktorej je potrebné rozvíjať u žiakov už na 1. a 2. stupni ZŠ. Môže to byť urobené rôznymi formami: tvorbou projektov, učením pomocou rozličných didaktických hier, zápisom do myšlienkových máp, výkladom, použitím situačných a inscenačných metód alebo pomocou nejakého programovacieho jazyka. Tematický celok algoritmy sme navrhli vyučovať pomocou rozličných logických hier.

Mnohé deti (žiaci) trávia neúmerne veľké množstvo svojho voľného času pri rôznych počítačových hrách. Predpokladáme, že počítačové hry majú v sebe niečo, čím priťahujú veľa detí a mladistvých. Zvyčajne sa však ich motivačná sila len v malej miere využíva na edukačné účely. Čoraz väčší počet výskumníkov sa zhoduje na tom, že hranie počítačových hier rozvíja myslenie detí v takej forme, ktorej musíme venovať významnú pozornosť. Dokonca sa v niektorých knihách objavujú tvrdenia, že myseľ detí je v súčasnej dobe preprogramovaná vďaka veľkému vplyvu počítačových hier. Podľa významných výskumníkov ako D. Tapscott a P. M. Greenfield, ktorí sa venujú edukačnej psychológii [7], sa v deťoch hrajúcich počítačové hry rozvíjajú celkom nové množiny kognitívnych zručností.

Tieto zručnosti môžeme zhrnúť ako paralelné spracovávanie informácií, lepšie spracovanie grafických informácií, experimentálny prístup k riešeniu problému, aktívne zapojenie žiaka do riešenia, rozvíjanie fantázie a tvorivosti žiaka a chápanie technológií ako priateľa.

Mladí ľudia neuveriteľne rýchlo rozvíjajú svoje schopnosti spracúvania informácií z rôznych zdrojov – rozlišujúc zdroje podľa toho, čo je pre nich dôležité a čo nie. S tým je spojené:

- rozvíjanie schopnosti zbierať údaje a informácie paralelne z niekoľkých informačných zdrojov,
- prieskum a zber dát nelineárnym spôsobom, keď si deti vytvárajú linky medzi zdanlivo
- nesúvisiacimi údajmi namiesto sledovania a vytvárania si lineárneho príbehu,
- posilňovanie tendencie rozvíjať myšlienky a hypotézy cez obrazy a piktogramy, a až následne
- potvrdzovať svoje hypotézy z prečítaného textu,
- prijímanie vytvoreného virtuálneho sveta ako priestoru na rozvíjanie fantázie a tvorivosti,

- prijímanie nových technológií ako pomocníka ba dokonca priateľa pri riešení problémov.

Za posledných 20 rokov, kedy sa počítačové hry stali takými populárnymi, sa odborníci zamýšľali nad tým, prečo sú tieto hry natoľko obľúbené, prečo im venuje toľko mladých ľudí takmer všetok voľný čas. T. Malone už v roku 1980 tvrdil, že hlavnými motivačnými faktormi pri hraní sa počítačových hier sú výzva, fantázia a zvedavosť. M. Prensky na základe Malonových tvrdení [7,8] zostavil niekoľko kľúčových charakteristík hier ako riadenie sa určitými pravidlami, definovanie cieľov a námetov, dosiahnutie výsledku a spätná väzba pre hráča, reakcie na konflikty/súťaženie/výzvy/protivníkov, interakcia s inými hráčmi resp. počítačom a reprezentácie príbehu.
Jones tieto charakteristiky ďalej rozvinul a vytvoril skupinu iných charakteristík, ktoré definujú zaujímavé hry:

- úlohy, ktoré sa dajú zvládnuť,
- schopnosť sústredenia sa na rozpracovanú úlohu,
- úlohy s jasnými cieľmi,
- riešenia poskytujúce okamžitú spätnú väzbu,
- hlboké, zároveň však pritom nie veľmi náročné zapojenie sa do hry,
- nacvičovanie si schopnosti ovládania svojich činností.

Niekoľko štúdií, venujúcich sa problematike hier a učenia sa, [7,8], tvrdí, že kľúčovým faktorom pri učení sa pomocou hier je rozvíjanie zručností. Medzi najdôležitejšie zručnosti tu zahŕňajú komunikácia a práca v tíme, riešenie problémov a rozvíjanie logického myslenia a matematických zručností. Počas hier sa deti neustále dostávajú do situácií, kedy musia riešiť problémy a úlohy. Správne riešenie od nich vyžaduje rýchle a efektívne naštudovanie aktuálnej pozície hráča, zorientovanie sa v súčasnom dianí v hre a uskutočnenie správneho rozhodnutia. Pritom si v zápale hry ani neuvedomujú, koľko nových vecí sa naučili pri hľadaní správnej odpovede na dôležitú otázku. Pri mnohých hrách sa hráči stretávajú s úlohami, kde musia preukázať svoje schopnosti pri jednoduchých výpočtoch.
Počítačové hry sme tiež využili pri výučbe programovacieho jazyka Imagine v pedagogickom experiment, ktorí prebiehal na Gymnáziu sv. Andreja v Ružomberku v období február – máj 2008. Trieda kvinta s 28 žiakmi bola rozdelená na štyri skupiny po 7 žiakov. Každý žiak mal k dispozícii svoj počítač, výučba prebiehala v rozsahu dva krát 45 minút za týždeň. Programovaciemu jazyku Imagine sme sa venovali celkom 9 vyučovacích hodín, z toho 4 hodiny samotnému jazyku a zvyšných 5 hodín tvorbe projektov.
Počas týchto zvyšných hodín žiaci pracovali v dvojiciach na projektoch, ktorých výstupom bol program v Imagine. Projekty si žiaci sami vybrali. Hoci sme im zadali niekoľko nápadov pre projekty (živý obraz, človeče, preteky, bludisko, mapy miest), väčšina si vybrala vlastnú tému. Pri vyberaní tém sme sa snažili, aby v projekte bolo obsiahnuté vlastné procedúry s podmienkami prípadne s cyklom, teda so štruktúrami, s ktorými sa počas hodín programovania v Imagine mohli stretnúť. Tiež sme sa snažili, aby projekty boli graficky príťažlivé a obohatené zvukom.

Témy vybraných projektov:

1. kuchárska kniha (jedlá sú rozdelené na raňajky, obedy a večere a ilustrované, úvodné obrázky boli použité z filmu Karcoolka),
2. návšteva ZOO (po vstupe máme pred sebou mapu ZOO, keď klikneme na klietku, objaví sa obrázok zvieratka, text o ňom a jeho zvuk),
3. divy sveta (lietadlom letíme po svete a keď priletíme nad div sveta, objavia sa o ňom informácie),

Obr. 1 Ukážka projektu Divy sveta

4. mapa Európy (po otvorení sa objaví mapa Európy, kde pri 10 štátoch je tlačidlo, po kliknutí sa objaví vlajka štátu, informácie o štáte a po kliknutí na vlajku sa spustí hymna štátu),
5. mapa dediny Černová (po otvorení sa ukáže mapa Černovej, kde sú červené bodky, po kliknutí na bodku sa objaví informácia o príslušnej lokalite),
6. domáce zvieratká (po kliknutí na zvieratko sa objaví o ňom text),
7. lietadlo (po obrazovke letí lietadlo, po kliknutí na panáčika skočí z lietadla a my ho máme navigovať na loď),
8. hra Zbieraj hubky (trojúrovňová hra – panáčik zbiera objekty, ak zoberie zlý objekt, tak prehrá, keď pozbiera všetky, prejde na ďalšiu úroveň, po tretej úrovni hráč hru vyhral),

Obr. 2 Ukážka projektu Zbieraj hubky

9. Írsko (autíčkom prechádzame po mape, keď sa dostaneme na bodku, objavia sa informácie o lokalite, okrem toho sú k dispozícii ukážky írskych tancov),

10. staroveký Egypt (tancujúcou múmiou chodíme po mape starovekého Egypta a keď prídeme na objekt, objavia sa informácie napr. o pyramídach, hieroglyfoch a pod.),
11. hra Rybka (žltá rybka zbiera 50 halierové mince a snaží sa vyhýbať červeným pohybujúcim sa rybkám),
12. sprievodca po Ružomberku (animovaný panáčik chodí po mape Ružomberka, keď narazí na objekt ukážu sa o ňom informácie, panáčik je rozanimovaný na 8 fáz, navrhli a rozanimovali si ho samy žiačky).

Vyučujúci pôsobil ako konzultant pri problémoch, ktoré sa počas programovania vyskytli. Niektoré projekty boli jednoduchšie (kuchárska kniha, návšteva ZOO, divy sveta, mapa Európy, mapa dediny Černová, domáce zvieratká, lietadlo), iné zložitejšie (Írsko, staroveký Egypt, sprievodca po Ružomberku, hra Zbieraj hubky, hra Rybka). Jednoduchšie projekty obsahujú väčšinou niekoľko navzájom prepojených stránok, ktoré obsahujú text, jednoduchú podmienku, obrázky a zvukovú kulisu. Boli viac zamerané na vyhľadávanie informácií. Tieto projekty zvládli žiaci väčšinou bez pomoci. Zložitejšie projekty v sebe obsahujú vlastné riadiace procedúry, ktoré si navrhli žiaci sami, prípadne sme im s nimi pomohli. Takisto sú viacstránkové. Dievčatá boli pri tvorbe projektov tvorivejšie, snažili sa, aby ich projekty boli zaujímavé graficky aj zvukovo, väčšinou viacstránkové s rozličnými informáciami o danej téme. Výnimkou bola dvojica dievčat, ktoré naprogramovali trojúrovňovú hru. Chlapci sa snažili programovať hlavne hry. Výnimky v tvorbe chlapcov sú mapa Európy a domáce zvieratká.
Programovanie v Imagine žiakov zaujímalo, aj tých, čo o celok algoritmizácia veľký záujem neprejavili. V štvrtej a druhej skupine žiaci prejavili mimoriadny záujem o programovanie v Imagine. Okrem hodín v škole sa Imaginu venovali aj doma a pýtali si ďalšie materiály. Poskytli sme im materiály prístupné na internetovej stránke súťaže Palma junior [9] a materiály pre výučbu detských programovacích jazykov od prof. Kalaša [Kalaš, I., Hrušecká, A., 2006]. Každý žiak mal k dispozícii tiež učebnicu Tvorivá informatika - 1. zošit z programovania [Kalaš, I., Blaho, A., 2005] a tiež CD s programovacím jazykom Imagine, aby mohol na projekte pracovať aj doma.
Tvorbou projektov sme rozvíjali nielen algoritmické myslenie, ale aj fantáziu, priestorové myslenie, rovinu predstavivosť, počítačové zručnosti a pod. Tiež sme u žiakov podporili kolaboratívne učenie. Na projektoch pracovali vo dvojiciach a úlohy si museli rozdeliť.
Výučba pomocou počítačových hier zlepšila počítačové zručnosti žiakov, ich strategické myslenie, schopnosť riešiť problémy (problémové situácie), zvýšila ich motiváciu a zlepšila ich pracovitosť na hodinách informatiky.

3. ZÁVER

Počítače a internet rozvíjajú u detí a mládeže duševný obzor, ich osobnosť, prípadne im pomáhajú v plnení školských povinností. Na druhej strane si však uvedomujú (i keď to menej zdôrazňujú) ich škodlivosť – nedostatok pohybu, športu, odvádzanie pozornosti od školských povinností, oslabovanie schopnosti bežne komunikovať, škodlivý vplyv počítačových hier a obsahu na internete a pod. V reálnom živote a v spoločnosti je obľúbenejší a viac cenený človek, ktorý vie komunikovať, ktorý vystupuje bez nervozity a má dobrú slovnú zásobu.
V prípade, že je dieťa odkázané prevažne na učenie sa pomocou počítača, je ochudobnené o niektoré životne dôležité skúsenosti založené na komunikácii s ľuďmi v jeho okolí.
Negatívny vplyv počítačových hier na psychický a sociálny vývoj detí sa doposiaľ v žiadnom vedeckom výskume nepotvrdil. A zrejme nie je celkom spravodlivé viniť IKT viac než všetky ďalšie „nástrahy a riziká“ dnešného života. Dobrá prevencia pred možnými zlými vplyvmi nových technológií a počítačových hier nespočíva vnútri IKT, ale v budovaní zdravého systému hodnôt.

LITERATÚRA

[1] ČERNÁK, I. – POLČIN, D.: Didaktika informatiky 2. Multimediálna učebnica na DVD. Ružomberok: Pedagogická fakulta KU 2008. ISBN: 978-80-8084-278-9
[2] ČERNÁK, I. – POLČIN, D.: Didaktika informatiky 1. Multimediálna učebnica na DVD. Ružomberok: Pedagogická fakulta KU 2007. ISBN: 978-80-8084-174-4
[3] KALAŠ, I.: Čo ponúkajú informačné a komunikačné technológie iným predmetom, Zborník konferencie Infovek 2000, Bratislava 2001, ISBN 80 – 7098 – 265 – 5.
[4] KALAŠ, I., BLAHO, A.: Tvorivá informatika - 1. zošit z programovania, SPN, Bratislava, 2005 ISBN 80-10-00019-1
[5] KALAŠ, I., HRUŠECKÁ, A.: Programovanie v prostredí Imagine, Metodické centrum 2006 Bratislava ISBN 80-8052-260-X
[6] ORTANČÍKOVÁ, H.: Didaktické projekty v informatike, DIDINFO 2005, UMB Banská Bystrica : 2005. ISBN 80-8083-090-8.:
[7] FACER, K. Computer Games and Learning. A discussion paper, FutureLab, UK, 2004
[8] GÁLIK, Z.: Pátracie aktivity a rozvoj moderných kompetencií žiakov. Bratislava, dizertačná práca, 2008
[9] RITNOŠÍKOVÁ, Z.: Interaktívne vyučovanie, http://www.mpctn.sk/konferencia/prispevky/Ritnosikova/Interaktivne_vyucovanie.doc
[10] http://edi.fmph.uniba.sk/~tomcsanyiova/
[11] http://di.ics.upjs.sk/palma/

VÝUČBA TVORBY WEBOVÝCH STRÁNOK NA STREDNEJ ŠKOLE

RNDR. DANIEL POLČIN, CSC. - LUCIA BIELÁKOVÁ

Katedra informatiky, Pedagogická fakulta, Katolícka univerzita, Námestie A. Hlinku 56/1, Ružomberok,
tel. 0908343155, e-mail: lucia.bielakova@gmail.com

Témou príspevku je poukázať na aktuálnu potrebu venovania sa výučbe tvorby internetových stránok na strednej škole. Popisuje časovú dotáciu, účel výučby tvorby webových stránok na strednej škole, spôsob i didaktické prostriedky potrebné vo výučbe a vhodný didaktický softvér na vytváranie internetových stránok. Obsah príspevku tvoria závery zo spracovanej diplomovej práce publikovanej a sprístupnenej na internete, v ktorej je navrhnutý i metodický postup výučby pre nižší a vyšší ročník strednej školy, ako aj overenie metodického postupu počas súvislej pedagogickej praxe, jej zhodnotenie a závery. Diplomová práca podáva nielen výsledky z dotazníkového prieskumu učiteľov stredných škôl, ale aj výsledky z dotazníka zadaného študentom na konci absolvovanej výučby.

Kľúčové slová: *časová dotácia, metodický postup, WYSIWYG editor KompoZer, overenie, závery*

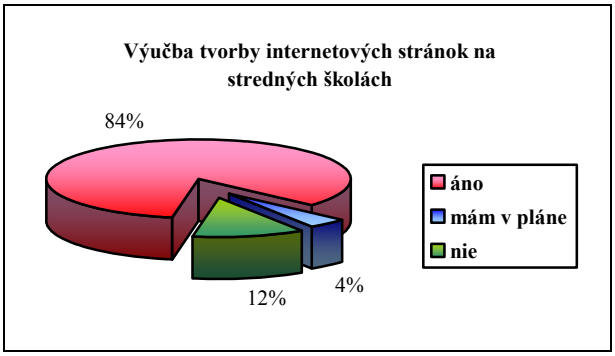
1. ÚVOD

Kto by to kedy povedal, že svet zaplaví vlna počítačov, ktorých vlastnenie sa ešte pred pár rokmi považovalo za luxus. Vďaka technickému pokroku, ktorý ovplyvnil celý svet, sa počítač stal neoddeliteľnou súčasťou takmer každej domácnosti. Postupom času už ľuďom nestačilo len vlastnenie tejto „hranatej krabice s káblami“ a tak sa snažili svoje obzory rozširovať aj ďalej. Pred nimi sa otvoril obrovský svet preplnený informáciami - svet internetu. V dnešnej dobe si už ľudia život bez internetu nevedia ani predstaviť. Prostredníctvom neho sa učia spoznávať ešte nespoznané, „nakuknú“ do všetkých kútov sveta, uzatvárajú obchody alebo udržiavajú kontakt s blízkymi osobami. Celý svet zaplnilo more serverov a domén internetových stránok. Internetové stránky sa tak stali nielen súčasťou reklamnej stratégie rôznych firiem, ale aj súčasťou výučby študentov v rámci informatiky hlavne na stredných a vysokých školách. Práve tvorba internetových stránok si čoraz viac nachádza obľubu aj u mladšej generácie. Tak ako technický pokrok nepozná hraníc, tak ani ľudia nepoznajú hranice pri rozširovaní svojich vedomostí a zručností. V diplomovej práci sa snažíme podať pomocnú ruku učiteľom informatiky stredných škôl v podobe metodickej príručky na výučbu tvorby webových stránok na strednej škole. Príručiek na výučbu tvorby internetových stránok je na trhu veľmi málo, a preto práve táto diplomová práca by mala pomocou jej jednotlivých kapitol učiteľom informatiky stredných škôl napomôcť pri tejto výučbe.

2. PREČO UČIŤ TVORBU WEBORÝCH STRÁNOK NA STREDNEJ ŠKOLE?

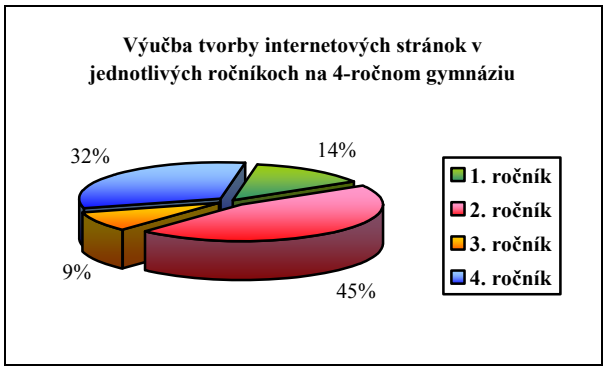
Internetové stránky vytvorené pomocou niektorého z HTML editorov vo veľkej miere nahradzujú prezentácie projektov vytvorených v špeciálne na to určených programoch (napr. Microsoft PowerPoint), ktorých nevýhodou je, že na ich spustenie je potrebný prezentačný softvér a projekt je prístupný len obmedzenému okruhu používateľov. Študenti si môžu jednoduchou a zábavnou formou vytvoriť vlastné internetové stránky, pomocou FTP ich sprístupniť na web a tým poskytnúť informácie nielen svojim spolužiakom, ale aj širokej verejnosti. Aj keď tvorba internetových stránok u študentov stredných škôl nie je až tak rozšírená, vďaka neustálemu technickému pokroku sa čoskoro dostane do popredia. To je dôvod, prečo by sa mala tvorba internetových stránok zaradiť aj na strednú školu.

Dotazníkový prieskum učiteľov informatiky rôznych typov stredných škôl len potvrdil veľký záujem stredných škôl o výučbu tvorby internetových stránok na strednej škole. Respondenti odpovedali na otázku: „Vyučujete na hodinách informatiky tvorbu internetových stránok?“, kde sa ukázalo, že 84 % učiteľov stredných škôl vyučuje na hodinách informatiky tvorbu webových stránok, 12 % učiteľov sa výučbou nezaoberá kvôli tomu, že tvorba internetových stránok nie je zaradená do učebných osnov jednotlivých typov stredných škôl a 4% učiteľov majú len v pláne výučbu tvorby internetových stránok zaradiť do učebných plánov strednej školy.



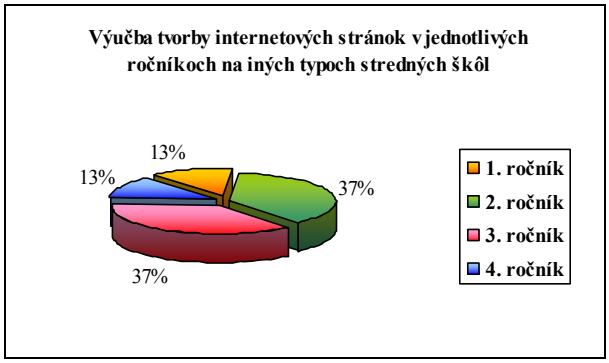
Obr. 1 Výučba tvorby internetových stránok na stredných školách

Podľa dotazníkového prieskumu sa tvorba internetových stránok vyučuje na gymnáziách a iných stredných školách rozdielne. Na 4-ročných gymnáziách výučbu preferuje až 45% učiteľov v 2. ročníku, 32% učiteľov v 4.ročníku, 14% učiteľov v 1. ročníku a len 9% učiteľov v 3. ročníku.



Obr. 2 Výučba tvorby internetových stránok v jednotlivých ročníkov na 4-ročnom gymnáziu

Tvorbu webových stránok na iných typoch stredných škôl vyučuje 37% učiteľov v 3. a 4. ročníku a 13% učiteľov v 1. a 4. ročníku.



Obr. 3 Výučby tvorby internetových stránok v jednotlivých ročníkoch na iných typoch stredných škôl

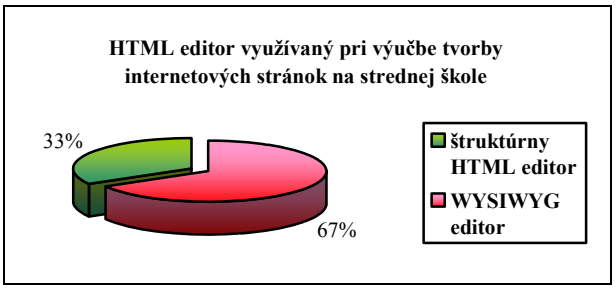
3. ČASOVÁ DOTÁCIA POTREBNÁ NA VÝUČBU

Na základe učebných osnov a štandardov pre jednotlivé typy stredných škôl odporúčame využiť nasledovnú časovú dotáciu. Výučbu tvorby internetových stránok na gymnáziách je možné rozdeliť do dvoch ročníkov. V 2. ročníku 4-ročného gymnázia odporúčame využiť časovú dotáciu 6 vyučovacích hodín a vo 4. ročníku vo voliteľnom predmete informatika alebo cvičenie z informatiky 4. vyučovacie hodiny. Na 8- ročných gymnáziách odporúčame zaradiť výučbu tvorby internetových stránok do 6. ročníka s časovou dotáciou 6 vyučovacích hodín a do 8. ročníka vo voliteľnom predmete informatika alebo cvičenie z informatiky s časovou dotáciou 4 vyučovacie hodiny. Na iných typoch stredných škôl odporúčame využiť časovú dotáciu do 10 vyučovacích hodín v 2. alebo v 3. ročníku.

4. AKÝ EDITOR ZARADIŤ DO VÝUČBY?

Kvôli zúženému finančnému rozpočtu pre stredné školy, nie je možné, aby sa na výučbu tvorby internetových stránok používali drahé HTML editory, ale editory, ktoré sú voľne dostupné na internete ako freeware. Jedno zo základných kritérií, ktoré by mali spĺňať, je jednoduché a prehľadné ovládanie, podobné s inými textovými editormi, na ktoré sú študenti zvyknutí a tým im vo veľkej miere uľahčiť ich prácu. Taktiež by to mal byť editor, ktorý je v praxi najpoužívanejší a ktorý je neustále aktualizovaný. Na základe týchto kritérií sú vhodnými editormi na výučbu tvorby internetových stránok editory KompoZer a Easy Editor, ktoré sa vyznačujú tým, že patria medzi WYSIWYG editory, za pomoci ktorých vytvoríme internetovú stránku aj bez toho, aby sme museli poznať HTML kód. Aj keď WYSIWYG editory majú niektoré nevýhody, na výučbu tvorby internetových stránok na stredných školách sú najvhodnejšie, nakoľko nie každého študenta baví učenie sa jazyka HTML a radšej si vytvorí peknú a jednoduchú stránku za krátky čas hravou formou v prostredí WYSIWYG editora.

Dotazníkový prieskum približuje druhy HTML editorov využívaných pri výučbe internetových stránok. Respondenti odpovedali na otázku „Ktorý HTML editor využívate na výučbu internetových stránok?“, z ktorých až 67% učiteľov preferuje práve WYSIWYG editor pre jeho jednoduchosť a 33% učiteľov štruktúrny HTML editor kvôli výučbe HTML kódu.



Obr. 4 HTML editor využívaný pri výučbe tvorby internetových stránok na strednej škole

5. SPÔSOB VÝUČBY A DIDAKTICKÉ PROSTRIEDKY POTREBNÉ NA VÝUČBU

Kvôli nízkej časovej dotácii na výučbu tvorby internetových stránok, nie je možné, aby sa učiteľ zameral len na HTML kód, ktorý je pre študentov dosť komplikovaný a nie tak zaujímavý, ako grafický mód WYSIWYG editora. Ak sa však učiteľ orientuje len na grafický mód WYSIWYG editora, dosiahne s ním na jednej strane rýchlu a jednoduchú tvorbu internetovej stránky, no na druhej strane nezáujem študentov o HTML kód. Pri chybách, ktoré môžu pri tvorbe internetovej stránky vzniknúť, si študent nebude vedieť danú chybu opraviť práve vďaka neznalosti HTML kódu. Preto odporúčame kombinovať výučbu tvorby internetových stránok medzi virtuálnym módom WYSIWYG editora a HTML kódom. Tým sa študenti nielen naučia pracovať s WYSIWYG editorom, ale aj pochopia základnú syntax jazyka HTML, čím sa budú vedieť orientovať v HTML kóde vlastnej internetovej stránky.

Tým, že internetové stránky sú v súčasnosti tvorené prevažne pomocou hlavnej tabuľky, do ktorej sa vkladá text, obrázky a hypertext a pomocou kaskádových štýlov, odporúčame takýto spôsob výučby využiť aj na strednej škole a rozdeliť ho do jej dvoch ročníkov.

V nižšom ročníku sa žiaci oboznámia s WYSIWYG editorom, získajú vedomosti a zručnosti potrebné na vytvorenie internetovej stránky tvorenej pomocou jednej hlavnej tabuľky a vo vyššou ročníku strednej školy si študenti vytvoria internetovú stránku pomocou kaskádových štýlov a interného CSS WYSIWYG editora.

Na dosiahnutie stanoveného vyučovacieho cieľa a zrozumiteľnejší výklad, odporúčame učiteľovi pri výučbe tvorby internetových stránok používať okrem počítača aj dataprojektor. Na rýchlejšie a jednoduchšie osvojenie si učebnej látky sú študentom k dispozícii študijné materiály v tlačenej ale aj elektronickej forme, ako aj upravené obrázky a šablóny poskytnuté učiteľom. Učiteľovi môže pri výučbe napomôcť kompletná multimediálna metodická príručka**, ktorá je učiteľovi k dispozícii v podobe internetovej stránky na adrese** www.diplomovka.comze.com.

6. METODICKÝ POSTUP PRE VÝUČBU TVORBY WEBOVÝCH STRÁNOK

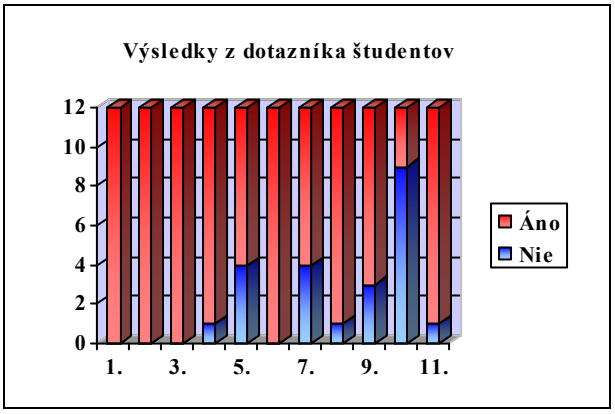
Metodický postup výučby tvorby webových stránok na strednej škole je rozdelený na metodický postup výučby pre nižší a vyšší ročník strednej školy. Každá vyučovacia hodina obsahuje okrem jej kompletného rozpracovania aj základné informácie potrebné pre učiteľa – časová dotácia, typ vyučovacej hodiny, metódy, formy a didaktické prostriedky, predpoklad očakávaný od študentov, stanovené ciele, nové pojmy alebo pojmy na zopakovanie a postup vyučovacej hodiny. V oboch metodických postupoch sú rozpracované jednotlivé vyučovacie hodiny výučby, ktoré sú okrem teoretickej časti doplnené pomocou rôznych ikon o rady učiteľom, poznámky týkajúce sa programu, v ktorom sa výučba uskutočňuje, o poznámky týkajúce sa výučby, základy HTML kódu, o úlohy a otázky zadávané študentom určené na upevnenie prebratého učiva a o úlohy určené na vytvorenie internetovej stránky spolu s ich správnym riešením.

7. ZÁVER

Po odpraxovaní navrhnutého metodického postupu na Gymnáziu Štefana Moyzesa v Ružomberku hodnotíme navrhnutý metodický postup a celkovú výučbu kladne. O správnosti navrhnutého metodického postupu a o výsledku pedagogickej praxe svedčí aj dotazník, ktorý bol vyplnený 12 študentmi prítomnými na poslednom stretnutí. Jeho cieľom bolo zistiť celkový pohľad študentov na problematiku výučby tvorby internetových stránok na strednej škole.

Celá výučba tvorby internetových stránok prebiehala v pokojnej atmosfére. Učivo, ktoré im bolo na jednotlivých stretnutiach vysvetlené jednoduchou a zrozumiteľnou formou, si študenti osvojili na výbornej úrovni (otázka z dotazníka č. 4). V dotazníku sa študenti zhodli, že obsah prebratého učiva na jednotlivých vyučovacích hodinách im bude postačovať na vytvorenie jednoduchej internetovej stránky (otázka z dotazníka č. 3). K lepšiemu pochopeniu učiva napomohli aj ukážky teoretickej alebo praktickej časti pomocou dataprojektora. WYSIWYG editor KompoZer študentov zaujal nielen základnými funkciami potrebnými na vytvorenie internetovej stránky (otázka z dotazníka č. 6), ale aj prehľadným grafickým rozhraním (otázka z dotazníka č. 5). Aj keď práca s ním neprebiehala vždy bez problémov (otázka z dotazníka č. 7), študenti si dokázali vytvoriť vlastné internetové stránky a tým sa dá predpokladať, že vďaka WYSIWYG editoru KompoZer študenti aj v budúcnosti siahnu po podobnom nástroji a budú ho využívať aj v praxi (otázka z dotazníka č. 9). Na základe práce s WYSIWYG editorom KompoZer môžeme skonštatovať, že kvôli jeho funkciám a dostupnosti zdarma sa dá považovať za vhodný nástroj na výučbu tvorby internetových stránok na strednej škole. Skoro všetci respondenti by WYSIWYG editor odporučili svojim spolužiakom a známym(otázka z dotazníka č. 8). Len málo zo študentov by na výučbu zvolilo iný WYSIWYG editor, napr. Microsoft FrontPage a Nvu. (otázka z dotazníka č. 10).

Všetci študenti hodnotili pozitívne, že mali na hodinách informatiky možnosť naučiť sa vytvárať jednoduché internetové stránky (otázka z dotazníka č. 2). Z tohto hľadiska sa zhodli, že by danú výučbu odporučili zaradiť do povinného predmetu informatika (otázka z dotazníka č. 11), keďže je v dnešnej dobe pre nich výhodou vedieť tvoriť internetové stránky (otázka z dotazníka č. 1). Práve tie môžu vo veľkej miere nahradiť prezentácie vytvorené pomocou programu Microsoft PowerPoint a môžu sa využívať aj na iných predmetoch ako len v predmete informatika.



Obr. 5 Výsledky z dotazníka zadaného študentom

LITERATÚRA

[1] MAŠEK, E. - LAJČIAK, P.: *Prezentačný softvér I.* Ružomberok: Edičné stredisko Pedagogickej fakulty. 2005. 152 s. ISBN 80-8084-049-0

[2] KLOHSE, J. 2005. *Anleitung: Für Nvu-Composer*

Version 1.0. [online]. Aktualizácia 26. Júl 2005. [citované 19. Jún 2008]. Dostupné na: <www.nvu-composer.de/doku/nvuanl101.pdf>

[3] ŠTÁTNY PEDAGOGICKÝ ÚSTAV. 2008. *Štátny vzdelávací program pre gymnáziá v Slovenskej republike. ISCED 3A – Vyššie sekundárne vzdelávanie.* [online]. [citované 1. Január 2009]. Dostupné na: <http://www.statpedu.sk/buxus/docs/kurikularna_transformacia/isced3a_jun30.pdf>

ORIENTED GRAPHS IN THE INFORMATICS

DOC. RNDR. BOHUSLAV SIVÁK, CSC., PAEDDR. MILOSLAVA SUDOLSKÁ, PHD.

Department of Computer Science, Faculty of Natural Science, University of Matej Bel,
Tajovského 40, Banská Bystrica, Slovakia, tel. ++421 48/602 3175,
e-mail: sivak@fpv.umb.sk, sudolska@fpv.umb.sk

ABSTRACT

Oriented graphs (mainly the finite graphs) can be used for the description of certain objects used in the informatics and in its applications, too. We are going to give some typical examples: starting form the binary relations to the formal languages and the geographical information systems.

Keywords: *vertex, edge, oriented graph, grammar, formal language, computer science education*

1. INTRODUCTION

We shall study finite and sometimes infinite oriented graphs having certain applications to some problems in the mathematics, the theory of formal languages and the geographical informatics systems. We assume that the readers all familiar the basic notions of the graph theory (vertex, edge, …).

2. BINARY RELATIONS

The binary relation on a set A is defined as any subset of the set

$AxA = \{(x,y) \mid x \in A , y \in A\}$.

Example 2.1. Let A = {1, 2, 3} and choose the binary relation

$R = \{(1,3), (2,1), (3,1)\}$.

Such a record of the binary relation is not used in practice. We can use the graphical method and draw the following oriented graph:

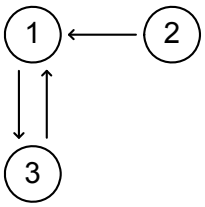


Fig. 2.1

The same binary relation can be defined (and this way is more suitable for a computer) by the following boolean matrix:

$$R = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} .$$

Using the boolean matrices, our students are able to compute, for example, the transitive closures of binary relations and the oriented graphs are a good method for the verification of these "matrix results".

3. NERODE´S AUTOMATA

In the theory of formal languages, we shall use some finite alphabet (a set of symbols) of the type

$A = \{a_1, a_2, \ldots, a_n\}$,

There are some operations that are often used in the theory of formal languages. Here are some of them:

- set-theoretical union: $L = L_1 \cup L_2$,
- concatenation: $L = L_1.L_2 = \{uv \mid u \in L_1 , v \in L_2\}$,
- powers: $L^n = \{u_1u_2...u_n \mid u_i \in L\}$,
- iteration: $L^* = \{\varepsilon\} \cup L \cup L^2 \cup L^3 \cup \ldots$,
- positive iteration: $L^+ = L \cup L^2 \cup L^3 \cup \ldots$.

Starting from finite sets of strings and applying these operations, we obtain a lot of formal languages in the (so called) Kleene´s form. For example, the language

$L = \{a\}^*.\{ba\}^+$

contains the strings

ba, aba, baba, aaba, ababa, bababa

and infinitely many other strings.

The formal languages can be distinguished by many types of automata. More than 50 years ago, Nerode defined a special type of automata that could be applied to every formal language. However, for some languages he obtained infinitely many states in these automata. The idea of Nerode was the following.

We have a finite alphabet A and a language $L \subseteq A^*$ .For every string $u \in A^*$ we construct a new language

$L_u = \{v \mid v \in A^* , uv \in L\}$.

Every language of the form L_u we shall use in the role of a state of the Nerode´s automaton. This automaton will be interpreted as an oriented graph. For every $t \in A$, there will be the oriented edge (labeled by t) from the vertex (= state) L_u to the

vertex L_{ut} . The state L_u will be accepting iff it contains the empty string ε . Usually, the empty state Ø is omitted in the table or the graph describing the automaton.

Example 3.1. Put $A = \{a,b\}$ and $L = (a.\{b^*)^+$. We can successively compute the states of the Nerode´s automaton:

	state of the Nerode´s aut.	accept.?
1	$L_\varepsilon = (a.\{b^*)^+$	–
2+	$L_a = \{b^*\}.\ (a.\{b\}^*)^*$	+
Ø	$L_b = \emptyset$	–
–	$L_{aa} = L_a$	–
–	$L_{ab} = L_a$	–

(The state 1 is not accepting, the state 2+ is accepting.) There are two methods how to describe the (Nerode´s) automaton: either by the table

	a	b
→ 1	2+	–
2+	2+	2+

or by the oriented graph (Fig. 3.1).

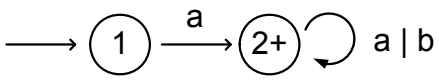


Fig. 3.1

Example 3.2. Put $A = \{a,b\}$ and

$L = \{a^x b^x \mid x \geq 1\}$.

Now, we obtain infinitely many states:

	a	b
→ 1	X1	–
X1	X2	Y1+
X2	X3	Y2
X3	X4	Y3
.....	.....	.....
X37	X38	Y37
.....	.....	.....
Y1+	–	–
Y2	–	Y1+
Y3	–	Y2
.....	.....	.....
Y37	–	Y36
.....	.....	.....

It is known that the language can be written in the Kleene´s form iff it is regular (= it can be generated by some regular grammar) and iff the Nerode´s automaton for this language has only a finite number of states. Consequently, the language studied in the Example 3.1 is not regular. On the other hand, this language is context-free because it can be generated by the following context-free grammar:

$$S \rightarrow ab \mid aSb\ .$$

4. GRAPH THEORY IN EDUCATION

The graph theory, as presented in the first part of article, supports the development of mathematical competence of a student. Applying the thus acquired knowledge in everyday life we develop his competence with ICT and the ability to problem solving, create the preconditions for the development of managerial skills and his ability to make the right choices in a given situation. There are the many ways how to teach basics of graph theory in a interesting way. We want to show the possibility of using geoinformatics, especially geographic information systems (GIS). Our goal is to create a set of interesting exercises, in which the students learn to apply theoretical knowledge of selected chapters of graph theory in situations they encounter in everyday life. Unlike traditional exercises, we want that the student to learn to specify the problem, to decide on the manner in which it is appropriate to address this problem, or to prepare and/or to retrieve the necessary data and find solutions. The benefit of the GIS technology use is that the student can verify the results of its solution (for example the way home from school) or he can display a solution on the map.

There are many tasks in geoinformatics that can be easily addressed through modeling and graphs, on which it is possible to apply theoretical knowledge of graph theory. The best known is the issue of coloring maps using the smallest number of colors. Less well known, but equally used, the geographical analysis of network (river, road, rail, utilities, etc.). The Voronoi diagrams are being used in the investigation of coherence of two phenomena.

We propose, for graphs theory education, to use the problems associated with finding the optimal connections between two locations or to create the path plan, or the optimal deployment of some objects in the region, and so on. In these problems students will learn how to analyze the problem, to abstract it correctly and to solve it in a simpler form - through the vertices and edges. They will learn to specify the quality and quantity of obtained data and how to order the source data by appropriate way with the goal to find the easiest way to solve the problem. A non-oriented graph is a special case of oriented one (where each edge can move in both directions). We will show the content of such at course in the following section.

4.1. Finding the shortest (best) route

Activity: The student has to find the shortest path to visit his 6 friends, each to be visited only once.

Proposed procedure: The student will mark places on the map which he wants to visit. He will assign paths and shortcuts, which he may use and he will compute their lengths (Fig. 4.1). Because each student has different friends, then solves this problem with different input data. Students think about the best solution after modeling of the problem. If necessary, the teacher will help students to create the graph model and show them the optimal

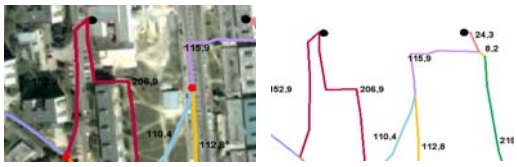


Fig. 4.1 Maps with assigned paths

way to find solutions (Fig.4.2). At this point, the teacher will continue with a plan of teaching, depending on the method to scan graph.

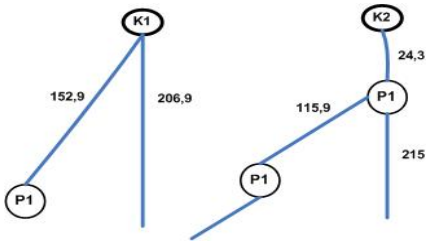


Fig. 4.2 Part of graph model

Alternative activities: The student goes on a bike. In this case he must follow the rules of the road (e.g., one-way streets). The conditions for problem solving are depended on the type of the bike (mountain, road). If a student knows the environment in which solve the problem, he may decide about the way to evaluate individual paths e.g. by the coefficient of distance and effort that must be made to overcome it (if the path is uphill, whether the "shortcut", some part of the way may be blocked, etc.).

Teaching aids: Which teacher's aids are needed for this education program? It is necessary to have the map of the locality, in which the student solves problem. The student should be able to obtain data on the distances. It is very good to use the map on the internet or take use of the program Google Earth, which has a measurement of distance function.

If the teacher chooses to combine this program with the teaching of programming, he must use the programming environment appropriate to the age of the student. He can use children's program Imagine, Microsoft Excel, Delphi or other programming environment. Technologies supporting geographic information systems, are the best, but the activity is made possible also through programs such as Microsoft Excel in conjunction with the Datamap which is complementary to the older versions, or a modern Microsoft MapPoint. Easier tasks can be addressed using any drawing program. The exercises, however, can be resolved without the Internet, even without a computer. The student used the classic maps and using map scales for the distance calculations.

It is not important, which technology we will choose. It is important to ensure that the students learn to work with data to analyze or to reorder data and to obtain the necessary information.

Suitable student age: What student age is appropriate for this program teaching? If a teacher elects correct approach may be easier tasks to solve in primary school. In our view, the age 8-10 years is suitable, given that in this age of the child is already able to generalize and to abstract. We know that children of this age are accessible to discovery and adventure. Teacher can use these traits.

5. CONCLUSION

We want to emphasize that these exercises are appropriate for integrated learning, students use knowledge from mathematics, geography and informatics. If formulated appropriately, one can also include topics from ecology and environmental education. It is possible to simplify the problems solution of everyday live by using graph theory, whether it is writing program, planning activities or optimize solutions. Therefore, we consider it appropriate for the pupil (the appropriate form) in the early school age basic algorithms of classical problems; so that they will know active use this knowledge.

Following organized learning positively affects the development of student's personality. Teacher teaches students to orient in large amount of information, to solve problems creatively and critically assess the results of their work.

REFERENCES

[1] Hopcroft, J. E., Ullman, J. D. : Formal languages and their relation to automata. Slovak translation: Alfa, Bratilsava 1978

[2] Chytil, M.: Automaty a gramatiky. SNTL, Praha 1984

[3] Molnár, Ľ: Češka, M., Melichar, B.: Gramatiky a jazyky. SNTL/Alfa, Bratislava 1987

[4] Bell, T., Wiktten, I. H., Fellows M.: Computer Science unplugged… 1998, available on internet: http://csunplugged.org, (January 2009)

[5] Sudolská, M.:. *Možnosti aplikovania GIS vo vzdelávacom procese*. Bratislava 2008: MPC, 2008,ISBN 80-8052-267-7

METODICKÉ POZNÁMKY K VÝUČBE OBLASTI KOMUNIKÁCIA PROSTREDNÍCTVOM IKT INTERNET = WEB + E-MAIL?

RNDR. ĽUBOMÍR ŠNAJDER, PHD. [1], MGR. JÁN GUNIŠ[1], MGR. VALENTÍNA GUNIŠOVÁ[1], RNDR. ĽUDMILA JAŠKOVÁ, PHD.[2]

[1]Prírodovedecká fakulta, Univerzita Pavla Jozefa Šafárika, Jesenná 5, Košice,
tel. 055 23 461 39, e-mail: lubomir.snajder@upjs.sk, jan.gunis@upjs.sk, valentina.gunisova@upjs.sk,
[2]Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina, Bratislava
tel. 02 602 95 396, ludmila.jaskova@fmph.uniba.sk

Čo učiť v stredoškolskej informatike v rámci oblasti Komunikácia prostredníctvom IKT? Stačí sa zaoberať len používaním webu a elektronickej pošty? Mali by sme sa venovať aj problematike zdravotných, sociálnych, etických a právnych aspektov využívania internetu? Klásť dôraz na fakty a konkrétne postupy, alebo na pojmy a princípy? Aký softvér používame na prácu s internetom vo výučbe? Ako často inovujeme svoje prípravy na výučbu? Aké pomôcky používame pri výučbe tejto oblasti? Ktoré kľúčové kompetencie žiakov a akým spôsobom chceme rozvíjať? K týmto a ďalším metodickým otázkam sa vyjadrujeme v príspevku a zároveň predstavujeme koncepciu a obsah nového tematického zošita Práca s internetom pre stredné školy.

Kľúčové slová: *internet, komunikácia, výučba, stredná škola, metodika, tematický zošit*

1. ÚVOD

Oblasť Komunikácia prostredníctvom IKT patrí k najnovším, najširším a najdynamickejšie sa rozvíjajúcim oblastiam školskej informatiky. Od roku 1994 sa začal objavovať internet na vybraných stredných a základných školách. Učitelia a žiaci sa vzájomne učili používať viaceré textovo orientované klientske programy. Postupne ich nahradili graficky orientované prostredia. Služba WWW nahradila viaceré zaniknuté služby, a to až do takej miery, že ju niektorí používatelia nesprávne stotožňujú s internetom. Vďaka čoraz rozsiahlejšiemu využívaniu internetu na komerčné účely sa jeho vývoj zdynamizoval. V súčasnosti je u nás internet prístupný prakticky pre každého. Na internet sú pripojené nielen počítače, ale aj mobilné telefóny, herné konzoly, dokonca aj domáce spotrebiče. Žiaci používajú internet nielen v škole, ale aj mimo nej. Internet prenikol do všetkých sfér života spoločnosti. Ukázal však aj svoje negatívne stránky – chorobnú závislosť, propagovanie nežiaducich ideológií, šírenie nevyžiadaných, či poplašných správ, kriminalitu, ale aj často podceňovanú problematiku dôveryhodnosti informácií prístupných prostredníctvom internetu.

Doterajšie vzdelávacie štandardy a učebnice [1, 2, 3, 4, 5] boli zamerané skôr na vysvetlenie problematiky základných pojmov a postupov práce s rôznymi internetovými klientmi. Súčasný internet však ponúka širokú škálu rôznych nástrojov pre získavanie, vyhľadávanie, výmenu, spracovanie, ukladanie, či prezentovanie informácií. S koľkými klientmi e-mailu sme sa doposiaľ stretli? Využívajú naši žiaci rovnakých klientov e-mailu? Koľkí z nás už zažili nejakú nepríjemnosť pri práci s internetom? Koľkí z nás sa už stali obeťou nejakého podvodníka? Koľko času trávime filtrovaním nevyžiadanej e-mailovej korešpondencie? Máme v triede nejakého, od internetu závislého, študenta? A vedeli by sme ho odhaliť? Do akej miery si uvedomujeme riziká pri využívaní internetu? Patrí problematika netikety, zdravotných a sociálnych aspektov využívania internetu do obsahu školskej informatiky, alebo patrí do obsahu iného predmetu, napr. do náuky o spoločnosti, etickej výchovy?

2. ČO JE OBSAHOM OBLASTI KOMUNIKÁCIA PROSTREDNÍCTVOM IKT

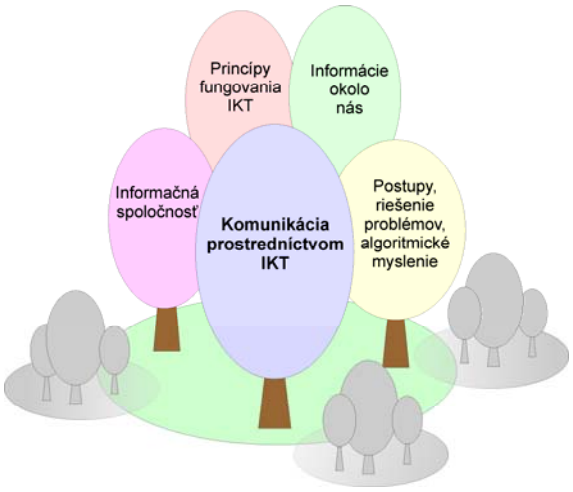
Aktuálne vzdelávacie štandardy z roku 2008 [6] v porovnaní s predchádzajúcimi viac zohľadňujú súčasné možnosti využívania internetu. Viac zdôrazňujú bezpečnostné, zdravotné, sociálne, etické, právne a iné aspekty. Dôležitosť problematiky internetu zdôrazňuje nové vymedzenie piatich kľúčových oblasti školskej informatiky. Vymedzila sa nová oblasť Komunikácia prostredníctvom IKT, ktorá dominantne pokrýva problematiku internetu. Pre jej úspešnú implementáciu do obsahu vyučovania je potrebné detailne rozpracovať nové vzdelávacie štandardy.

Pri vymedzovaní obsahu učiva by sme mali brať do úvahy viaceré aspekty.

Obsah problematiky internetu môžeme vnímať cez štyri pohľady: princípy fungovania internetu (infraštruktúra, protokoly), internetové služby (možnosti internetových klientov), riešenie reálnych problémov a bezpečnostné, etické, právne, zdravotné, sociálne a iné aspekty. Všetky tieto pohľady by mali byť zastúpené v obsahu stredoškolskej výučby. Každú redukciu považujeme za zúženie danej problematiky.

Množstvo poznatkov v tejto oblasti je veľmi veľké. Nové rýchlo pribúdajú, viaceré zastarávajú,

stávajú sa neaktuálnymi. Pri výbere obsahu učiva by sme mali byť otvorení novým poznatkom a ochotní vyradiť zastaralé a neaktuálne, orientovať sa na poznatky využiteľné v živote a ďalšom vzdelávaní.



Obr. 1 Obsah oblasti „Komunikácia prostredníctvom IKT“ sa prekrýva nielen s ostatnými oblasťami školskej informatiky, ale aj s obsahom ďalších školských predmetov.

Odporúčame klásť dôraz na pojmy a princípy, nie na fakty a konkrétne postupy, t.j. zamerať sa na vysvetľovanie významu pojmov, fakty uvádzať len na ilustráciu a spresnenie učiva. Zamerať sa na princíp fungovania triedy nástrojov s podobným zameraním, nie na ovládanie konkrétneho softvéru.

Vybraný obsah učiva by mal napomáhať rozvoju vyšších úrovní myslenia žiakov (analytickému mysleniu, hodnotiacemu mysleniu, tvorivosti) [7,8] a kľúčových kompetencií (kompetencia k celoživotnému učeniu sa, sociálne, komunikačné kompetencie, kompetencie uplatňovať matematické myslenie a poznávanie v oblasti vedy a techniky, kompetencie v oblasti IKT, kompetencia riešiť problémy, kompetencie občianske, kompetencie sociálne a personálne, kompetencie pracovné, kompetencie smerujúce k iniciatívnosti a podnikavosti, kompetencie vnímať a chápať kultúru a vyjadrovať sa nástrojmi kultúry) [9].

3. AKO UČIŤ OBLASŤ KOMUNIKÁCIA PROSTREDNÍCTVOM IKT

Po vymedzení obsahu učiva je na učiteľovi, aby ho vhodným spôsobom žiakom sprostredkoval. Boli by sme radi, keby okrem učebnice učiteľ mal k dispozícii a používal aj ďalšie pomôcky: metodickú príručku, pracovné listy, otázky a odpovede k riešeniu problémov, pracovné súbory, ukážky žiackych projektov atď.

Existuje pestrá škála didaktických metód a foriem výučby. Odporúčame čo najčastejšie využívať motivačné, konštruktivistické a aktivizujúce metódy, ktoré žiaka viac vtiahnu do aktívneho a zainteresovaného vzdelávania sa. Na ilustráciu uvádzame:

- myšlienková mapa (napr. pri vysvetľovaní princípov fungovania vybranej služby, na zobrazenie súvislostí a vzťahov medzi preberanými pojmami),
- brainstorming (napr. pri návrhu obsahu webovej stránky),
- hranie rolí (napr. zákaznícka podpora pri riešení problému internetového pripojenia),
- inscenačná metóda (napr. putovanie balíčkov dát medzi počítačmi cez sústavu smerovačov),
- didaktická hra (napr. skupinový vedomostný kvíz k základným pojmom internetu),
- prípadová štúdia (napr. riešenie bezpečnostného problému s internetbankingom),
- projektová metóda (napr. profesijná webová stránka žiaka) atď.

4. KONCEPCIA A OBSAH TEMATICKÉHO ZOŠITA PRÁCA S INTERNETOM

Pri tvorbe nového tematického zošita pre stredné školy [10] a definovaní jeho obsahu sme brali do úvahy nasledovné aspekty.

Tematický zošit je určený žiakom 1. - 3. ročníka stredných škôl. Podľa našich skúseností, žiaci prichádzajúci na strednú školu už majú skúsenosti s využívaním služieb internetu. Chýba im však systém poznatkov, internet využívajú intuitívne, živelne, často neefektívne. Upustili sme preto od uvádzania triviálnych postupov ako pracovať s vybranými nástrojmi. Namiesto konkrétnych faktov vysvetľujeme pojmy, namiesto konkrétnych postupov sa venujeme princípom. Na základe skúseností s výučbou sa ukazuje, že žiaci tento prístup akceptujú.

Tematický zošit chápeme ako učebnicu, a nie ako pracovný zošit alebo tutoriál s množstvom snímok obrazovky s konkrétnym programom. Nadväzujeme na základnú učebnicu stredoškolskej informatiky [1], ideme viac do hĺbky a do obsahu sme zaradili aktuálnu problematiku využívania internetu. Našou snahou bolo, aby zošit žiaka motivoval a zároveň rozšíril a systematizoval jeho vedomosti.

V zošite sa venujeme aj aspektom, ktoré na prvý pohľad s využívaním internetu nesúvisia, alebo súvisia len okrajovo. Problematiku využívania internetu však chápeme v širšom kontexte. Do zošita sme preto zaradili časti týkajúce sa netikety, počítačovej etiky, bezpečnosti, ale aj problematiku prístupnosti informácií pre osoby so špeciálnymi potrebami.

Aby sa žiakom so zošitom ľahšie pracovalo, všetky kapitoly majú rovnakú štruktúru. Pojmová

mapa na začiatku každej kapitoly prehľadne znázorňuje obsah kapitoly. Zároveň definuje ústredný pojem kapitoly a hierarchiu vzťahov medzi jednotlivými pojmami kapitoly.

Cieľom motivačného úvodu je ilustrovať, že preberaná problematika kapitoly sa žiaka môže osobne týkať. Popisuje situáciu, ktorej účastníkom môže byť reálne on sám. Zároveň sme týmto úvodom chceli žiaka „naštartovať“ pre štúdium kapitoly.

Samotný výkladový text je doplnený obrázkami a schémami. V texte sa nezameriavame len na vymenovanie obsahu samotného, popisujeme vlastnosti jednotlivých riešení, ukazujeme ich výhody a nevýhody. Žiak by mal mať prehľad o viacerých možnostiach riešenia daného problému a zároveň by si mal vedieť v konkrétnej situácii vybrať to najvhodnejšie.

Doplňujúce učivo, zaujímavosti z danej oblasti a odkazy na zaujímavé internetové zdroje je zobrazené formou nálepiek (post-it), čo je pre žiaka v texte ľahko identifikovateľné.

V texte sú priebežne umiestnené otázky, úlohy, námety na aktivity. Otázky sme formulovali tak, aby odpoveďou nebola len reprodukcia časti výkladového textu. Žiak by sa mal zamyslieť, pozrieť sa na danú problematiku v širšom kontexte a vziať do úvahy viacero aspektov. Pri riešení úloh, resp. hľadaní odpovedí na otázky, by mali žiaci diskutovať, spolupracovať a aktívne využívať internet. Komplexnejšie a náročnejšie úlohy, námety na projekty sú zaradené na konci každej kapitoly pod názvom „Porozmýšľajte a vyskúšajte si“. Niektoré z úloh sú formulované tak, aby sa žiak pri ich riešení niečo nové naučil. Či takéto úlohy budú zaradené do procesu výučby je na posúdení učiteľa.

Ďalším cieľom zošita je, aby sa žiaci vedeli správne vyjadrovať, používať vhodnú terminológiu. Anglická terminológia sa často neprekladá alebo prekladá nepresne. Kde je to možné, tam uvádzame slovenské pojmy a k nim aj anglické ekvivalenty. Myslíme si, že dnešný žiak by mal poznať základnú terminológiu predmetu aj v niektorom zo svetových jazykov.

Na konci zošita je uvedený register pojmov aj s referenciou na miesto v zošite, kde je pojem vysvetlený, resp. je použitý v kontexte. Samozrejmosťou je zoznam použitej literatúry a relevantných internetových zdrojov.

Do zošita sme zaradili aj ukážky menej tradičného spôsobu zobrazenia informácií. Schémou na titulnej strane sme chceli zdôrazniť, že dnešný internet je o ľuďoch a ich sociálnom správaní. Na vnútornej strane prednej obálky je uvedený jeden zo spôsobov zobrazenia štruktúry webovej stránky. Na vnútornej strane zadnej obálky je mrak slov znázorňujúci obsah problematiky spracovanej v celom zošite.

Obr. 2 Mrak slov reprezentujúci obsah tematického zošita.

Obmedzenia vyplývajúce z rozsahu zošita (48 strán) spôsobili, že jeho výsledná podoba je do istej miery kompromisom. Témy ako e-spoločnosť, počítačová etika a netiketa sú zaradené len v obmedzujúcom rozsahu. Dnes toľko diskutovanú problematiku sociálnych a zdravotných aspektov využívania internetu sme z vyššie uvedených dôvodov do zošita nakoniec nezaradili.

Výsledná podoba zošita obsahuje 8 kapitol.

V časti „Internet a počítačové siete“ stručne popisujeme dôvody vzniku internetu a jeho vlastnosti z toho vyplývajúce. Venujeme sa aj technickým aspektom počítačových sietí a princípom komunikácie zariadení v počítačových sieťach.

Kapitola „Prehľad internetových služieb, služba WWW“ popisuje internetové služby a možnosti ich využitia pri vyhľadávaní informácií a vzájomnej komunikácii. Nosnou časťou je služba WWW aj s dnes tak často používaným prívlastkom web 2.0.

Komunikácii v širšom kontexte sa venuje časť „Komunikácia na internete“. Zaoberá sa nielen vzájomnou komunikáciou ľudí, ale aj komunikáciou človeka so strojom využívajúcim prvky umelej inteligencie. Popisujeme nielen princíp a využitie jednotlivých typov komunikácie, uvádzame aj pravidlá komunikácie. Žiak sa oboznámi s modernými spôsobmi komunikácie, ako napr. IP telefónia alebo wiki.

V časti „Vyhľadávanie informácií na internete“ sa zaoberáme princípom jednotlivých typov vyhľadávacích nástrojov. Napriek sile týchto nástrojov však existuje väčšia časť obsahu webu, ktorá ostáva pre nich nedostupná, tzv. neviditeľný web. Uvádzame pravidlá ako tvoriť vyhľadávacie dopyty a ako postupovať pri vyhľadávaní, aby výsledok obsahoval čo najväčší podiel relevantných výsledkov. Venujeme sa aj tzv. sociálnemu vyhľadávaniu, ako dôsledku vyplývajúcemu zo sociálneho správania sa používateľov internetu.

Donedávna často podceňovanou oblasťou bola „Bezpečnosť na internete“. Narastajúci počet bezpečnostných incidentov a zraniteľnosť používateľov (najmä detí) na internete nás presviedča o dôležitosti tejto problematiky. Venujeme sa nielen zabezpečeniu počítača používateľa, ale aj ochrane používateľa samotného. Uvádzame pravidlá bezpečného správania sa v sieti internet. Zaoberáme sa problematikou pravdivosti a dôveryhodnosti informácií a možnosťami jej riešenia prostredníctvom šifrovania.

Časť „Využitie internetu vo vzdelávaní“ predstavuje internet ako priestor pre vzdelávanie, spoluprácu a vzájomnú výmenu poznatkov. Uvádzame množstvo námetov a konkrétnych príkladov využitia internetu vo vzdelávaní (blogy, teleprojekty, on-line aktivity a pod.). Venujeme sa aj problematike on-line a dištančného vzdelávania.

Cieľom časti „Internet pre všetkých“ je, aby si žiaci uvedomili, že internet a jeho zdroje by mali byť prístupné každému, bez ohľadu na špeciálne potreby alebo obmedzenia jeho používateľov. Ukazujeme, s akými problémami sa môžu takíto používatelia stretnúť a aké dôsledky to môže mať. Zároveň naznačujeme niektoré možnosti riešení. Táto časť je aj propedeutikou poslednej kapitoly „Profesijné webové stránky“.

Zavŕšením procesu spracovania informácií je ich samotná prezentácia. V prostredí internetu sú touto možnosťou najmä webové stránky. Aby sme sa vyhli obsahovo chudobným, resp. ťažko identifikovateľným webom upriamujeme pozornosť žiaka na osobné stránky resp. ich formálnejšie verzie profesijné stránky. Cieľom nie je zvládnutie jazyka HTML, resp. niektorého zo skriptovacích jazykov. Táto problematika by si zaslúžila samostatnú publikáciu. Naším cieľom je, aby žiak dokázal vytvoriť zmysluplnú, prístupnú a použiteľnú webovú prezentáciu. Žiak by nemal byť „rýchlym murárom“ skladajúcim jednotlivé tehličky webu do nič nehovoriaceho výsledku. Chceme, aby sa stal architektom vidiacim použiteľný celok, architektom, ktorý pomocou dostupných nástrojov vie tento celok vytvoriť.

5. ZÁVER

V našom príspevku sme chceli priblížiť učiteľskej verejnosti svoje predstavy o metodike vyučovania oblasti Komunikácia prostredníctvom IKT na strednej škole. Predstavili sme koncepciu a obsah nového tematického zošita Práca s internetom.

Výsledky publikované v príspevku boli vytvorené v rámci APVV grantu LPP-0131-06 „Zvyšovanie vedomostného potenciálu“

LITERATÚRA

[1] KALAŠ, I. a kol.: Informatika pre stredné školy. SPN - Media Trade, s. r. o., Bratislava,, 2005, 112 s., ISBN 80-10-00762-5

[2] JAŠKOVÁ, Ľ., ŠNAJDER, Ľ., BARANOVIČ, R.: Práca s Internetom - tematický zošit pre 1.ročník gymnázií. SPN - Media Trade, s. r. o., Bratislava, 2000, 48 s., ISBN 80-08-02986-2

[3] BARANOVIČ, R. , JAŠKOVÁ, Ľ., ŠNAJDER, Ľ.: Internet pre stredné školy - učebnica pre stredné školy. Computer Press : Brno, 2003, 275 s., ISBN 80-251-0088-X

[4] Vzdelávací štandard pre gymnáziá <http://www.statpedu.sk/buxus/docs/Pedagogicke_dokumenty/Gymnazia/4roc/standardy/Standard_z_informatiky.doc>

[5] VARGA, M., HRUŠECKÁ, A.: Tvorivá informatika - 1. zošit s internetom. SPN - Mladé letá, s. r. o., Bratislava, 2006, 48 s., ISBN 80-10-00648-3

[6] Štátny vzdelávací program - príloha 3 - ISCED 3 <http://www.statpedu.sk/buxus/docs/kurikularna_transformacia/isced3.rar>

[7] KALHOUS, Z., OBST, O. a kol.: Školní didaktika. 1. vyd. Praha : Portál, 2002, 448 s., ISBN 80-7178-253-X

[8] BAJTOŠ, J.: Teória a prax didaktiky. 1. vyd. Žilina : Žilinská univerzita, 2003, 384 s., ISBN 80-8070-130-X

[9] Štátny vzdelávací program pre gymnáziá v Slovenskej republike ISCED 3A – Vyššie sekundárne vzdelávanie. Štátny vzdelávací program <http://www.statpedu.sk/buxus/docs/kurikularna_transformacia/isced3a_jun30.pdf>

[10] ŠNAJDER, Ľ., GUNIŠ, J., GUNIŠOVÁ, V., JAŠKOVÁ, Ľ.: Informatika pre stredné školy - Práca s internetom. 1. vydanie. SPN - Media Trade, s. r. o., Bratislava, 2008. ISBN 978-80-10-01518-4

MOŽNOSTI NASADENIA OTVORENÉHO SOFTVÉRU NA ŠKOLÁCH.

DOC. ING. MILOŠ ŠRÁMEK, PHD.

Commission for Scientific Visualization, Austrian Academy of Sciences, Vienna, Austria

Katedra aplikovanej informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina Bratislava

e-mail: milos.sramek@oeaw.ac.at

Jednou z najvýznamnejších čŕt otvoreného a slobodného softvéru je skutočnosť, že je dostupný zdarma pre jednotlivcov, ako aj organizácie. Na prvý pohľad by sa teda zdalo, že nič nebráni jeho prieniku do škôl a do pedagogického procesu. Opak je však pravdou, miera využitia otvoreného a slobodného softvéru zaostáva za možnosťami, ktoré poskytuje. V príspevku sa pokúsime o analýzu dôvodov tohto stavu, podáme prehľad možností, ktoré otvorený a slobodný softvér poskytuje a uvedieme niekoľko stratégií, ako možno postupovať pri jeho nasadení v škole.

Kľúčové slová: *otvorený a slobodný softvér, programové vybavenie pre školy*

1. ÚVOD

Pojmom počítačová gramotnosť sa označuje súbor schopností efektívne využívať informačné technológie. Získať ju bezpochyby možno rôznym spôsobom, je však isté, že školy v tomto procese hrajú významnú úlohu. Dnes zrejme každá základná alebo stredná škola na Slovensku vlastní základné počítačové vybavenie, ktorého hlavným zdrojom bol projekt Infovek [1]. Školy takto dostali istý počet počítačov a softvér vo forme tzv. Edubalíka. Súčasťou dodaného softvéru bola infraštruktúra, orientovaná najmä na voľne dostupný Linux a používateľské programy, orientované na MS Windows. Dodatočné počítačové vybavenie mimo týchto zdrojov školy získali na základe iniciatívy a zápalu vedenia školy a najmä samotných učiteľov [2].

Takto získaná úroveň počítačového vybavenia škôl však už nevyhovuje dnešným požiadavkám. Ministerstvo školstva SR vypracovalo Stratégiu informatizácie regionálneho školstva v SR[1], ktorá stanovuje cieľ vybaviť do r. 2011 základné školy jedným počítačom na 10 žiakov a stredné školy jedným počítačom na 5 žiakov[2], čo predstavuje približne 114.000 počítačov. Súčasne sa uvažuje aj s vybavením učiteľov jedným notebookom na dvoch učiteľov, čo predstavuje asi 32.000 kusov.

Stratégia informatizácie nešpecifikuje postupy, ktorými sa má tento stav dosiahnuť. Školy, prípadne ich zriaďovatelia počítače nakupujú z vlastných prostriedkov a ministerstvo nemá oprávnenie do tohto procesu vstupovať [3]. V súvislosti so softvérovým zabezpečením ministerstvo podpísalo so spoločnosťou Microsoft zmluvu School Agreement, na základe ktorej vzdelávacie inštitúcie môžu získať licencie na softvérové produkty prostredníctvom predplatného na jednoročné alebo trojročné obdobie. Po uplynutí zmluvného obdobia je potrebné buď zmluvy predĺžiť alebo treba softvér z počítačov odstrániť. Bohužiaľ, cenová ponuka nie je zverejnená. Ak by sme však predpokladali podobnú úroveň, aká sa uvádza v návrhu zmluvy Microsoft Enterprise Agreement[3] na roky 2009 – 2011 (238 €/rok bez DPH pre najnižší produkt Desktop Standard), dospeli by sme k sume 42 mil. € ročne. Získavanie týchto prostriedkov však môže byť pre školy problémom. Pritom tu ide len o základné softvérové vybavenie (operačný systém MS Windows a balík MS Office), bez zarátania ďalších nevyhnutných nákladov a edukačného softvéru

Jednostranná orientácia na produkty spoločnosti Microsoft, prípadne iných komerčných dodávateľov, však nie je jedinou voľbou. V r. 2004 – 2005 bola na objednávku MŠ SR v rámci úlohy výskumu a vývoja „Open Source infraštruktúra“ (OSIN), vypracovaná štúdia o možnostiach nasadenia otvoreného softvéru, ktorý je dostupný zdarma, v štátnej správe a školstve [4]. Jej hlavný prínos spočíva v návrhu riešení zameraných na základné modely hardvérovej a softvérovej infraštruktúry, kde boli porovnávané ekvivalenty na proprietárnej a otvorenej báze. Záver bol: pri konzervatívnom odhade by sa použitím otvorených prostriedkov celoštátne v jednom obnovovacom cykle (6 – 8 rokov) dalo ušetriť až 18 miliárd korún. Táto zmena by mala, samozrejme, aj nepeňažné dôsledky, spočívajúce najmä v zlepšení dostupnosti moderných technológií.

Napriek týmto záverom však výsledok úlohy zostal prakticky nepovšimnutý. Cieľom tohto príspevku je preto bližšie upozorniť na možnosti, ktoré ponúka otvorený softvér pre školy a poukázať na možné stratégie jeho nasadenia.

1 www.minedu.sk/data/USERDATA/ISKOL/ISDOC/20080220_strategia_informatizacie_RS.rtf

2 http://www.minedu.sk/data/USERDATA/ISKOL/2008/20080627_Akcny_%20plan_%20strategie_%20informatizacie.pdf

3 https://lt.justice.gov.sk/Material/MaterialHome.aspx?instEID=-1&matEID=826&langEID=1

2. OTVORENÝ A SLOBODNÝ SOFTVÉR

V prvopočiatkoch výpočtovej techniky sa softvér nepovažoval za tovar, bol len doplnkom hardvéru. Používatelia počítačov vtedy boli aj programátormi a programy si mohli sami kompilovať, upravovať a dokonca aj vymieňať. Podľa dnešných kritérií by sme takýto softvér nazvali slobodným. Tento pojem však vznikol až neskôr, keď sa softvér stal tovarom a jeho používanie začali dodávatelia upravovať licenčnými zmluvami. Podstatou takýchto komerčných licenčných zmlúv býva najmä zákaz ďalšieho šírenia a obmedzenie prístupu k zdrojovému kódu programu, a tak aj znemožnenie jeho úpravy. Používateľ takéhoto komerčného softvéru je závislý od dodávateľa – v prípade problémov alebo potreby úprav je na neho odkázaný a je len vecou vôle dodávateľa, akú funkcionalitu v programe poskytne a či používateľovým potrebám vyhovie.

Tento, pre používateľov nepriaznivý stav, však nie všetci akceptovali. Jeden z nich, Richard M. Stallman si uvedomil [5], že práve licenčná zmluva, ktorá je nástrojom na obmedzovanie práv používateľov, môže byť aj nástrojom na ich ochranu. Vytvoril nový typ licencie, General Public License (GPL), ktorá je opakom komerčných licencií: namiesto ochrany záujmov dodávateľa ochraňuje práva používateľa. R. M. Stallman tým formuloval princípy tzv. slobodného softvéru, ktoré spočívajú v štyroch právach:

1. právo na použitie programu na akýkoľvek účel,
2. právo študovať ako program pracuje,
3. právo na ďalšie šírenie programu,
4. právo na úpravu programu podľa svojich potrieb a na šírenie upravenej verzie programu.

Podmienkou práv 2 a 4 je prístup k zdrojovému textu programu. Zdá sa, že so slobodným softvérom možno robiť čokoľvek. Nie je tomu tak. Nato, aby tieto slobody mohli byť v plnej miere naozaj využiteľné, GPL ich zakazuje obmedzovať. To znamená, že všetky ďalšie verzie programu s licenciou GPL, ako aj všetky od neho odvodené programy, musia mať licenciu GPL.

V horeuvedených podmienkach sa nehovorí nič o cene alebo komerčnom využití slobodného softvéru. Tu nie sú žiadne obmedzenia, slobodný softvér sa môže rozdávať zdarma, ale môže sa aj predávať. Samozrejme, práva 1. – 4. však musia byť zaručené. V prípade predaja je teda zrejmé, že nepôjde o klasický predaj softvéru, ale že skôr pôjde o služby s ním súvisiace. Takto vznikli viaceré obchodné modely, ktoré napríklad vychádzajú zo zostavovania kolekcií otvoreného softvéru – tzv. distribúcií (Red Hat, Mandrake, Ubuntu) alebo spočívajú v poskytovaní dodatočných služieb (IBM).

Idey R. M. Stallmana mali širokú odozvu a postupne bolo s využitím jeho licencie vytvorené bezpočetné množstvo programov. Dva najznámejšie z nich sú jadro operačného systému Linux a kancelársky balík OpenOffice.org. Autor Linuxu, Linus Torvalds, dokonca o licencii GPL povedal, že jej voľba bola najlepším rozhodnutím, aké kedy spravil, lebo práve vďaka nej mohol Linux v spolupráci so stovkami prispievateľov vyvíjať tak rýchlo.

Licencia GPL nie je jedinou, ktorá vychádza z práv 1 – 4. Ostatné však netrvajú na podmienke ich prenosu na odvodený produkt, ktorý sa tak môže stať súčasťou programu dodávaného s komerčnou licenciou. Tieto programy nazývame otvorené (lebo je otvorený ich zdrojový kód). Ich príkladom je jadro Darwin operačného systému MAC OS X, ktoré je odvodené od otvoreného operačného systému FreeBSD. Otvorený a slobodný softvér sa často spájajú do jednej kategórie FOSS – Free and Open Source Software).

Otvorenosť, ktorá je charakteristikou FOSS programov, vedie k vysokokvalitným a bezpečným programom. Dôvodom je práve zverejnenie kódu – každý má možnosť ho študovať, a tak každý môže upozorňovať na chyby a odhaľovať potenciálne problémy. Príkladom je samotný Internet, ktorý je na FOSS vybudovaný. Bez 100% spoľahlivosti by tak zložitý systém nemohol fungovať.

3. FOSS A ŠKOLY

Otvorený a slobodný softvér má vďaka svojim vlastnostiam všetky podmienky na úspešné využitie v školstve. Je zadarmo – ktorá škola má zvyšné prostriedky? Učitelia ho môžu voľne rozdávať svojim žiakom a sami ho môžu slobodne používať doma. Netreba nič objednávať, nie sú potrebné žiadne zmluvy – stačí si vziať a používať. Triviálnym spôsobom zabezpečuje legálnosť. Napriek tomu však komerčný softvér u nás na školách dominuje, čo zrejme má viacero príčin.

Tradícia. Najzávažnejším dôvodom tohoto stavu je zrejme tradícia. FOSS k nám prišiel ako druhý, v čase, keď sa snáď na všetkých osobných počítačoch používala niektorá verzia operačného systému Microsoft. Ten je dnes častokrát braný ako povinná súčasť počítača – mnoho používateľov zrejme ani nevie, že by to mohlo byť inak.

Návyk. Používanie softvéru nie je vždy celkom jednoduché. Mnohí používatelia preto majú obavy z akýchkoľvek zmien, a to aj v rámci nimi používaného prostredia. Na tak závažný krok, akým je zmena operačného systému a všetkých programov, preto treba aj trochu odvahy a dostatočne silnú motiváciu.

Neexistencia inštitucionalizácie. Skutočnosť, že otvorený softvér je dostupný zdarma, je jeho výhodou, ale súčasne aj nevýhodou. O šírenie a propagáciu komerčného produktu sa stará jeho výrobca alebo dodávateľ, FOSS softvér sa ponúka len sám. Na Slovensku existujú viaceré dobrovoľnícke združe-

nia, ktoré si dávajú za cieľ propagáciu otvoreného softvéru (SKLUG, SKOSI), avšak ich možnosti sú obmedzené. Navyše, inštitúcie sú zvyknuté rokovať s inštitúciami, a tak vo výberových konaniach FOSS softvér často nemá kto zastupovať. Na túto skutočnosť upozornili aj riešitelia už spomínanej úlohy OSIN [4], ktorí v svojej správe odporúčali zriadenie organizácie Open Source Academy, ktorej úlohou by bola práve podpora šírenia FOSS. Bohužiaľ, so zriadením takejto organizácie sa neuvažuje [3].

Obavy z neznámeho. FOSS prináša do našej spoločnosti nový prvok – produkt, ktorý je dostupný zdarma. Zdarma obvykle nebýva nič, a tak prvotná nedôvera je prirodzenou reakciou. Autor tohto príspevku sa sám priznáva k svojmu prekvapeniu pri prvom kontakte so slobodným softvérom.

Mýty o Linuxe. Mnohí sú presvedčení, že Linux sa ovláda len pomocou príkazového riadka, a je teda len pre špecialistov. Iným je mýtus o nedostatku softvéru a najmä ovládačov. Pritom Linux podporuje najširšie spektrum zariadení zo všetkých operačných systémov. Nechceme, samozrejme, tvrdiť, že FOSS dokáže nahradiť úplne všetko. Existujú programy, primárne určené najmä na profesionálne aplikácie, ktoré sú dostupné len komerčne – mnohé sú však ako také dostupné ja pre Linux.

Neopodstatnené očakávania. Mnohí používatelia považujú Linux a ďalšie FOSS prostriedky za bezplatnú verziu komerčných produktov. Očakávajú, že poskytujú rovnakú funkcionalitu a identické ovládanie. Nie sú ochotní akceptovať odlišnosti a obetovať potrebný čas na zoznámenie sa s nimi.

Nelegálne šírenie komerčného softvéru. Je známe, a vždy tomu tak bolo, že komerčný softvér sa v rozsiahlej miere šíri nelegálnymi metódami. Takto vlastne zaniká jeden z najpodstatnejších rozdielov medzi FOSS a komerčným softvérom, cena. Je tiež známe, že komerčné firmy takéto správanie tolerujú, pretože zabezpečuje rozširovanie používateľskej základne [6].

4. STRATÉGIE NASADENIA FOSS NA ŠKOLE

Sme presvedčení, že otvorený softvér v mnohých prípadoch môže plnohodnotne nahradiť komerčné aplikácie, a to rovnako z pohľadu funkcionality, ako aj používateľskej prístupnosti. Prechod na nové programové vybavenie je však celkom iný problém, najmä ak ide o zmenu v rámci tak zložitého systému, akým je škola. Tu bezpochyby môže rozhodujúcu úlohu zohrať osobný záujem a nadšenie pre vec konkrétneho riaditeľa, učiteľa či školského administrátora. V prípade, ak sa pre zmenu rozhodne, má v podstate tri možnosti, ako otvorený softvér využívať.

FOSS v prostredí Windows. FOSS aplikácie sú obvykle platformovo nezávislé, čo znamená, že sú rovnako dobre použiteľné vo všetkých hlavných operačných systémoch. Vďaka tomu sa z pohľadu používateľa stierajú rozdiely medzi jednotlivými platformami, pretože v každej z nich sú prístupné rovnaké nástroje. Výhodou tohto prístupu je jednoduchá dostupnosť – stačí sa len rozhodnúť pre nový program a nainštalovať si ho. Medzi aplikácie, ktorými by sa používanie FOSS na škole takto dalo začať by sme mohli zaradiť prehliadač Firefox, kancelársky balík OpenOffice.org, publikačný program Scribus (na tvorbu školského časopisu), grafické aplikácie Gimp a Inkscape, prehrávač médií VLC alebo editor zvukových súborov Audacity. V kategórii aplikácií špeciálne určených na výučbu by sme mohli vymenovať systém na interaktívnu geometriu Kružidlo a pravítko [7] alebo systém na výučbu fyziky Phet[4].

V poslednej dobe získava na popularite nový spôsob používania FOSS aplikácií v prostredí Windows – vo forme tzv. prenositeľných aplikácií[5]. Takýto program možno nainštalovať na výmenné zariadenie (USB kľúč alebo disk, mp3 prehrávač, mobilný telefón s USB pripojením), vďaka čomu si používateľ môže preniesť svoje prostredie so všetkými nastaveniami (záložky, adresáre, pošta, dokumenty) na ľubovoľný počítač, s tým, že po odpojení zariadenia na hostiteľskom počítači nezostane žiaden záznam o jeho činnosti. Táto možnosť je priam ideálna pre školy, pretože rovnaké prostredie môžu mať žiaci v škole aj doma.

Dual boot Windows + Linux. Dual boot umožnuje koexistenciu dvoch alebo viacerých operačných systémov na disku jedného počítača a je zrejme najpopulárnejším spôsobom, ktorým sa zoznamovanie s Linuxom obvykle začína. Je vhodný pre školy a univerzity, pretože učiteľovi alebo študentovi umožní voľbu podľa svojich preferencií. Prináša však zvýšenie nárokov na administráciu a pre individuálne použitie nemusí byť ideálnym riešením, pretože v jednom časovom okamihu sprístupňuje len jeden systém. Ako hovorí Murphyho zákon, vždy máte spustený ten nesprávny systém.

Pre používateľa, ktorý potrebuje pracovať s viacerými systémami alebo chce len vyskúšať nový systém, je vhodnejším riešením spúšťanie druhého operačného systému v špeciálnej aplikácii, tzv. virtuálnom počítači (VMWare, VirtualBox[6]). V tomto prípade sa zachováva prístup k obom (viacerým) systémom súčasne a netreba reinštalovať hostiteľský systém.

Linux. Linux si možno buď len jednoducho vyskúšať, alebo sa možno rozhodnúť pre jeho plné nasadenie vo vyučovaní.

V prípade, ak učiteľ chce jednorazovo žiakov zoznámiť s Linuxom ako alternatívou Windows, môže

4 http://phet.colorado.edu

5 http://portableapps.com/

6 http://www.virtualbox.org/

použiť tzv. live CD (DVD, USB kľúč). Ide o kompletný operačný systém uložený na príslušnom médiu, ktorý sa z neho aj zavádza (či už naozaj, alebo vo virtuálnom stroji). Výhodou je jednoduchosť použitia (CD vložiť do mechaniky, reštartovať počítač a chvíľu počkať). Žiak tu nič nemôže pokaziť, pretože po novom reštarte sa systém vráti do pôvodného stavu. Live CD nepoužíva pevný disk počítača, a tak sa na ňom nainštalovaný systém neporuší. V live verzii sú dnes k dispozícii mnohé distribúcie Linuxu, či už len na vyskúšanie pred inštaláciou (Ubuntu, Fedora, Mandriva), alebo aj na špecializované použitie (Knoppix, Slax).

Plná inštalácia Linuxu v učebni je podľa nás najvhodnejším riešením v prípade rozhodnutia opustiť operačný systém Windows. Len takto sa totiž plne prejavia výhody otvoreného softvéru na škole, spočívajúce v nezávislosti od dodávateľa, nepotrebnosti podpisovania zmlúv a platenia licenčných poplatkov. Nejde však o jednoduché rozhodnutie, pretože okrem toho *či* Linux používať sa treba tiež rozhodnúť *ako* ho používať. Linux totiž neposkytuje len prostriedky pre inštaláciu na jednom žiackom počítači, ale umožňuje zabezpečiť kompletnú počítačovú infraštruktúru školy (web, mail, súborový server, centrálna správa používateľov a pod.). Táto téma je však mimo možností tohto príspevku[7].

Jednou z predností Linuxu je flexibilita, ktorú možno používateľ Windows ani neočakáva. Napríklad, pre rôzne ročníky žiakov možno používať rôzne usporiadanie pracovnej plochy – pre prvákov to môže byť zjednodušené prostredie s niekoľkými detskými aplikáciami, pre maturantov zasa bežný desktop s „profesionálnym“ vzhľadom. Jednoducho možno meniť správcov pracovnej plochy, čím sa zmení vzhľad, však nezmenia sa princípy používania. Vďaka tomu sa žiakovi ľahšie priblíži rozdiel medzi všeobecnými a špeciálnymi vlastnosťami, a tak sa lepšie pripraví na neustály vývoj, s ktorým sa bude stretávať neskôr.

Iným prejavom flexibility Linuxu je možnosť používania terminálových sietí. Každý administrátor vie, koľko práce predstavuje administrácia učebne s 20 počítačmi. Terminálová sieť je také riešenie, kde sa aplikácie spúšťajú na serveri a klientske stanice slúžia len na zobrazovanie [2]. Výhod je viacero. V prvom rade, spravovať treba len jeden počítač namiesto dvadsiatich. Tie ostatné – tenkí klienti – žiaden softvér nemajú, všetko potrebné získajú cez sieť zo serveru. Vďaka tomu nepotrebujú disk a stačí im len málo pamäte. V ich úlohe teda možno ľahko použiť staré počítače, ktoré by inak šli do šrotu. Poruchy takéhoto klienta sa netreba báť – ak sa jeden pokazí, pripojí sa iný. Žiadna nová inštalácia systému nie je potrebná. Dnes sú dokonca dostupné aj špecializované počítače na použitie v úlohe tenkých klientov. Sú nehlučné a tak malé, že sa dajú pripevniť na zadnú stenu monitora. A navyše, ich energetická spotreba je rádovo nižšia ako v prípade klasického počítača.

7 Viac o inštalácii školského servera: http://sk.openacademy.eu/kniha/ako-nainstalovat-skolsky-server

5. ZÁVER

V príspevku sme poukázali možné na problémy súvisiace s implementáciou Stratégie informatizácie regionálneho školstva v SR a ako možné riešenie, ktoré by mohlo ušetriť značné finančné prostriedky, sme navrhli použitie bezplatného slobodného a otvoreného softvéru a operačného systému Linux. Je na samotných školách a na ich zriaďovateľoch, či sa týmto smerom vydajú a na Ministerstve školstva a jeho organizáciách, či takéto riešenie bude v nejakej forme podporované. Prvé lastovičky sa už objavili, o ich prístupe a o používaní otvoreného softvéru na školách sa možno dozvedieť viac na stránke http://sk.openacademy.eu, ktorú budujú učitelia sami pre seba.

LITERATÚRA

[1] SÝKORA. P. *Filozofia projektu INFOVEK* [online]. [cit: 22. 5. 2007]. Dostupné na Internete: http://www.infovek.sk/konferencia/2000/prispevky/filozofia.html.

[2] PAUCHLY, K. *HW a SW riešenie učebne na ZŠ*. In: Otvorený softvér vo vede a vzdelávaní. Bratislava, 2007, ISBN 978-80-969562-7-2, s. 97 – 108.

[3] MICHÁLEK, J. 2 s. Listová odpoveď, 2007.

[4] Riešiteľský kolektív. *Návrh koncepcie nasadenia a podpory OS softvéru* [online]. [cit: 22. 5. 2007]. Dostupné na Internete: http://www.eea.sk/osin/doc/KoncepciaNasadeniaAPodporyOSS.pdf .

[5] RAYMOND, E. *The Cathedral and the Bazaar*. O'Reilly, 2000. ISBN 1-56592-724-9.

[6] LEYDEN, J. *If you must pirate, use counterfeit Windows.* In The Register [online] [cit: 3. 1. 2009]. Dostupné na Internete http://www.theregister.co.uk/2007/03/13/ms_piracy_benefits/.

[7] TULEJA, S. *Dynamická geometria s programom Compass and Ruler (Kružidlo a pravítko)*. In: Otvorený softvér vo vede a vzdelávaní. Bratislava, 2008, ISBN 978-80-89313-04-4, s. 169 – 181.

NÁROČNOSŤ ÚLOH V SÚŤAŽI INFORMATICKÝ BOBOR

RNDr. Peter Tomcsányi

Katedra výučby a základov informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského,
Mlynská Dolina, 842 48 Bratislava,
e-mail: tomcsanyi@fmph.uniba.sk

V našom príspevku uvedieme najprv niektoré základné štatistické údaje o počtoch a úspešnosti súťažiacich v súťaži Informatický bobor, potom sa zameriame na analýzu náročnosti jednotlivých úloh a ich skupín. Uvedieme aj príklady najmenej a najviac náročných úloh.

Kľúčové slová: *Informatická súťaž, iBobor, náročnosť úloh*

1. ÚVOD

Súťaž Informatický bobor, skrátene iBobor [6] si kladie za cieľ osloviť čo najviac učiteľov a žiakov na celom Slovensku bez rozdielu ich informatických vedomostí. Ide jej o šírenie počítačovej gramotnosti, o motivovanie žiakov, študentov aj učiteľov zdokonaľovať sa vo všetkých oblastiach informatiky ako aj používať svoje informatické vedomosti a zručnosti v praxi. Viac informácií nájde čitateľ v [2].

Súťaž vznikla pôvodne v Litve [1], kde bol v tomto školskom roku organizovaný jej 5. ročník, a v súčasnosti sa organizuje v 9 európskych krajinách. Na Slovensku v tomto školskom roku prebehol jej druhý ročník.

2. ZÁKLADNÉ INFORMÁCIE

V školskom roku 2007/08 sa súťaž konala v dňoch 10. – 12. júna 2008 v troch kategóriách. V školskom roku 2008/09 prebehla v dňoch 6. - 11. novembra 2008 v štyroch kategóriách - najmladšia kategória Benjamíni, ktorá mala pôvodne vekové rozpätie 5 rokov, bola rozdelená na dve menšie.

Prehľad súťažných kategórií a počtov riešiteľov:

Benjamíni	5.-7. trieda ZŠ, príma a sekunda osemročného gymnázia
Kadeti	8. - 9. trieda ZŠ, tercia a kvarta osemročného gymnázia
Juniori	1. - 2. ročník gymnázií a stredných škôl, kvinta a sexta osemročného gymnázia
Seniori	ostatní študenti stredných škôl a osemročných gymnázií

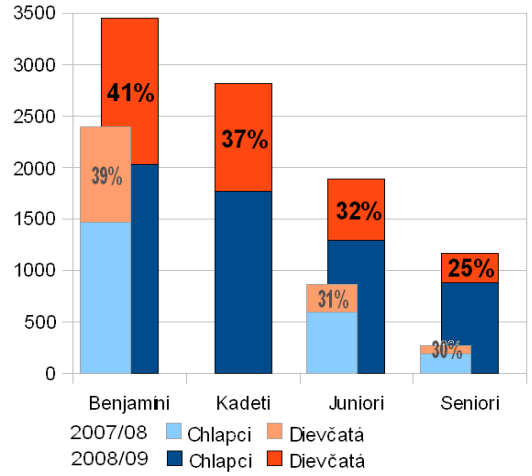
Školský rok 2007/08

	Chlapci	Dievčatá	Spolu	Školy
Benjamíni	1483	938	2421	94
Juniori	604	270	874	45
Seniori	192	81	273	34
Spolu	2279	1289	**3568**	**121**

Školský rok 2008/09

			Spolu		Školy	
	Chlapci	Dievčatá	Počet	Zmena	Počet	Zmena
Benjamíni	2031	1416	3447	259%	181	223%
Kadeti	1768	1045	2813		193	
Juniori	1288	602	1890	216%	95	211%
Seniori	878	289	1167	427%	86	253%
Spolu	5965	3352	**9317**	**261%**	**276**	**228%**

V školskom roku 2008/09 sa iBobra zúčastnilo spolu 9317 žiakov z 276 škôl. Počet žiakov sa zvýšil na 261% minuloročného počtu. So zvyšujúcim sa vekom klesá ako počet súťažiacich tak aj zastúpenie dievčat.



Percentá vyjadrujú počet dievčat z celkového počtu v danej kategórii a roku

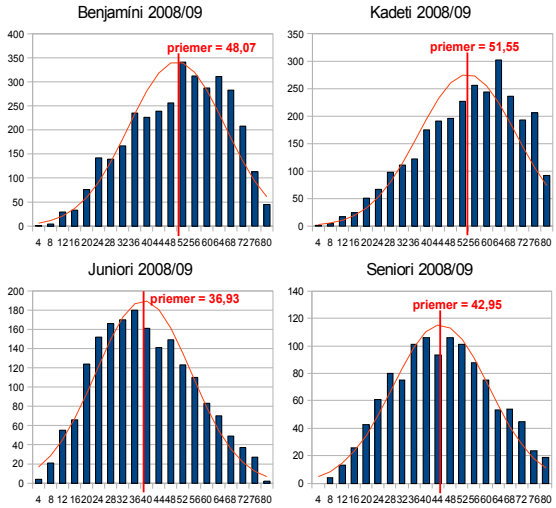
Nárast počtu seniorov až na 427% minuloročného počtu (takmer dvojnásobný rast oproti iným kategóriám) sa dá vysvetliť tým, že minulý rok sa súťaž konala v čase, keď už maturanti odišli zo škôl a teda bola k dispozícii len polovičná základňa potenciálnych súťažiacich.

Podľa údajov o počte škôl a žiakov z webu UIPŠ [7] to znamená, že sa zúčastnilo asi 1,6% žiakov príslušných ročníkov a asi 11,7% všetkých škôl, ktoré majú príslušné ročníky. Je to dobrý výsledok, lebo v mnohých väčších krajinách než je Slovensko (napríklad v Poľsku, Rakúsku či Holandsku), súťažilo v absolútnom počte menej účastníkov, teda mali podstatne menší podiel účastníkov k počtu všetkých žiakov v danej krajine. Slovenský počet účastníkov bol v absolútnom počte druhý najvyšší - viac súťažiacich bolo len v Nemecku. [5].

3. ROZDELENIE BODOV

Nasledujúce histogramy dokumentujú rozdelenie bodov v jednotlivých kategóriách od minima 0,05 bodu po maximum 80 bodov. Modrú farbu má

histogram zostavený z bodov v intervale po 4 body, červená krivka je teoretické normálne rozdelenie vypočítané z priemeru a smerodajnej odchýlky.



Z histogramov vidno, že súťaž bola najľahšia pre kadetov, kde je rozdelenie bodov zreteľne vychýlené smerom do vyšších hodnôt, len o niečo ťažšia bola súťaž pre benjamínov.

Keďže dôležitým cieľom súťaže je motivovať účastníkov, lepšie výsledky v nižších kategóriách môžeme hodnotiť kladne, teda mladší súťažiaci mohli mať viac radosti z dobrých výsledkov v súťaži. Je však možné, že sme po rozdelení minuloročnej kategórie Benjamín na dve kategórie v tomto roku nedostatočne uvážili, o koľko náročnejšie majú byť príklady pre kadetov. V budúcom ročníku budeme preto venovať zvýšenú pozornosť výberu príkladov práve pre kadetov.

Istý vplyv na rozdiel medzi mladšími a staršími mohlo mať aj to, že v prvý súťažný deň (keď mali súťažiť benjamíni a kadeti) došlo k preťaženiu servera a k odloženiu súťaže o 6 dní, pričom nezanedbateľný počet súťažiacich už niektoré otázky videl.

Najťažšia bola súťaž pre juniorov, kde je rozdelenie bodov a aj priemer viditeľne vychýlený do nižších hodnôt. Minulý rok sme takýto posun nespozorovali, preto musíme podrobnejšie preskúmať náročnosť úloh pre juniorov. Negatívnu úlohu pri výbere súťažiacich mohol zohrať aj fakt, že súťažný deň kolidoval so súťažou Zenit a je možné, že najlepší súťažiaci sa radšej zúčastnili Zenitu než iBobra.

Najsymetrickejší graf dostávame pre seniorov. Priemerný počet bodov sa v tejto kategórii najviac približuje hodnote 40 bodov a rozdelenie bodov je najbližšie teoretickému normálnemu rozdeleniu. Teda súťažné úlohy v kategórii senior najlepšie spĺňajú v týchto parametroch kritériá kladené na didaktické testy. Rovnako to bolo aj minulý rok. Je možné, že tvorcovia úloh, väčšinou z akademického prostredia, vedia zrejme najlepšie odhadnúť náročnosť pre túto vekovú kategóriu.

4. ĽAHKÉ A ŤAŽKÉ PRÍKLADY, CITLIVOSŤ

Nasledujúca tabuľka sumarizuje dva dôležité parametre všetkých príkladov [3]:

Hodnota náročnosti Q je percentuálny podiel iných než správnych odpovedí a všetkých odpovedí. Podľa literatúry [3] sa za bežné považujú úlohy s hodnotou Q od 30 do 70, úlohy dosahujúce 20 až 30 sú jednoduché a pod 20 sú veľmi jednoduché. Podobne úlohy s náročnosťou 70 až 80 sú náročné a nad 80 extrémne náročné.

Index citlivosti ULI (upper-lower-index) udáva mieru, do akej príklad rozlíšil lepších a horších súťažiacich. Môže mať hodnoty od -1 do 1. Záporné hodnoty indikujú, že príklad bol mätúci - lepšie ho riešili tí súťažiaci, ktorí skončili v druhej polovici výsledkovej listiny. Podľa [3] sa pre bežné úlohy za dobré považujú hodnoty nad 0,25. Pre jednoduché úlohy (Q je 20 až 30) a pre náročné úlohy (Q je 70 až 80) postačuje hodnota 0,15.

	Benjamíni		Kadeti		Juniori		Seniori		
	Q	ULI	Q	ULI	Q	ULI	Q	ULI	
1	64	0,25	40	0,41	35	0,38	11	0,20	ľahké (3b)
2	71	0,33	32	0,37	42	0,27	51	0,42	
3	71	0,18	21	0,33	44	0,23	59	0,29	
4	35	0,43	10	0,17	48	0,37	69	0,38	
5	3	0,10	38	0,38	78	0,27	53	0,48	
6	28	0,18	39	0,37	64	0,43	47	0,29	stredné (4b)
7	40	0,51	11	0,20	50	0,31	39	0,43	
8	24	0,35	35	0,42	42	0,44	43	0,35	
9	43	0,37	26	0,33	56	0,25	17	0,27	
10	28	0,37	13	0,21	63	0,31	81	0,16	
11	42	0,39	53	0,43	52	0,48	63	0,34	ťažké (5b)
12	35	0,49	78	0,10	70	0,21	41	0,46	
13	32	0,43	56	0,51	41	0,38	32	0,40	
14	76	0,34	24	0,36	81	0,10	46	0,33	
15	21	0,25	34	0,48	44	0,49	60	0,31	

Legenda:

- veľmi jednoduché
- jednoduché
- náročné
- veľmi náročné
- malá hodnota citlivosti

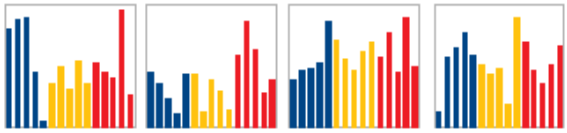
Vidíme, že v súťaži bolo 6 veľmi jednoduchých úloh a 2 veľmi náročné. Malá hodnota citlivosti sa vyskytuje len u dvoch takých úloh, ktoré nepatria do extrémov náročnosti. Žiadna úloha nebola mätúca.

V súlade s odporúčaniami pre zostavovanie testov je dobré mať v teste jeden veľmi jednoduchý príklad. To presne spĺňa kategória benjamíni a ostatné kategórie sa od toho príliš neodchyľujú.

5. OČAKÁVANÁ A SKUTOČNÁ NÁROČNOSŤ

V súťaži sú úlohy v každej kategórii vopred rozdelené na tri úrovne - ľahké, stredné a ťažké (viď posledný stĺpec tabuľky) a podľa toho sa líši aj bodový zisk za vyriešenie tej ktorej úlohy.

Zaujímalo nás, ako toto rozdelenie korešponduje so skutočnou náročnosťou úloh. V nasledujúcom grafe vidíme náročnosť úloh (Q) pre všetky kategórie, pričom sme farebne rozlíšili tri úrovne.



Graf ukazuje, že skutočná náročnosť úloh zistená analýzou výsledkov a zaradenie príkladu do skupiny podľa náročnosti navzájom vôbec nekorešpondujú.

Formálne sa tento fakt dá dokázať štatistickým U-testom Manna a Whitneyho, ktorým sme testovali parameter Q v každej dvojici skupín príkladov v rámci každej kategórie. Ani v jednom prípade nebola na hladine významnosti 5% zamietnutá nulová hypotéza o zhodnosti rozdelení hodnoty Q v dvoch skupinách príkladov.

Úroveň náročnosti navrhuje zadávateľ úlohy. Zadávatelia sú pritom z rôznych krajín, teda predpokladajú odlišné znalosti študentov istého veku. Pri výbere úloh do národnej súťaže sa náročnosti prehodnocujú, ale organizátori sú, celkom prirodzene, ovplyvnení zadávateľom.

Navyše nejde o test zo školských vedomostí, ale o súťaž, v ktorej sa majú ukázať vedomosti a zručnosti súťažiacich bez ohľadu na to, kde ich získali. Je preto ťažké, ak nie nemožné, definovať dopredu predpokladané vedomosti. Tie práve zisťujeme analýzou odpovedí po súťaži.

Preto uvedený nesúlad nemôžeme považovať za zásadný problém, ale za jav, ktorý treba v ďalších ročníkoch sledovať a snažiť sa ho minimalizovať.

Treba vykonať aj analýzu údajov z iných krajín, či je tento jav medzinárodný. Ak sa to potvrdí a dlhodobejšie sa nebude dariť zosúladiť očakávanú náročnosť so skutočnosťou, tak bude možno rozumné zmeniť spôsob bodovania.

6. NÁROČNOSŤ PODĽA TYPOV ÚLOH

V práci [4] sa úlohy iBobra zatrieďujú podľa ich typu do štyroch skupín, pričom niektoré úlohy patria do dvoch skupín. Skupiny sú definované takto:

Digitálna gramotnosť

- základné znalosti o počítačoch a informatike
- počítačová gramotnosť, práca so základnými typmi softvérových aplikácií
- etické a právne aspekty, bezpečnosť, dejiny počítačov a informatiky

Programovanie

- formálny popis riešenia, postupu, chovania
- pochopenie, analyzovanie, interpretovanie a vytváranie takých popisov
- algoritmy, algoritmické myslenie

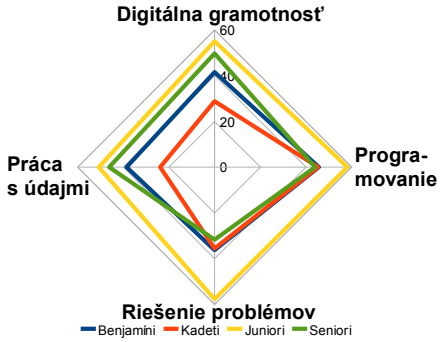
Riešenie problémov

- logické zdôvodňovanie, argumentácia
- hlavolamy, hádanky, problémy
- stratégie riešenia problémov

Práca s údajmi

- reprezentácie údajov, kódovanie, modely
- matematické základy informatiky, kombinatorika
- údaje a štruktúry údajov, informácia a spracovanie údajov

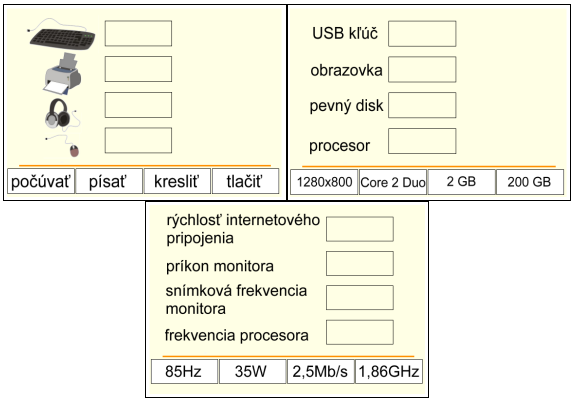
Po zatriedení všetkých úloh do týchto skupín môžeme vypočítať priemernú náročnosť príkladov v rámci jednotlivých skupín. Výsledky tohto výpočtu sme znázornili v nasledujúcom grafe:



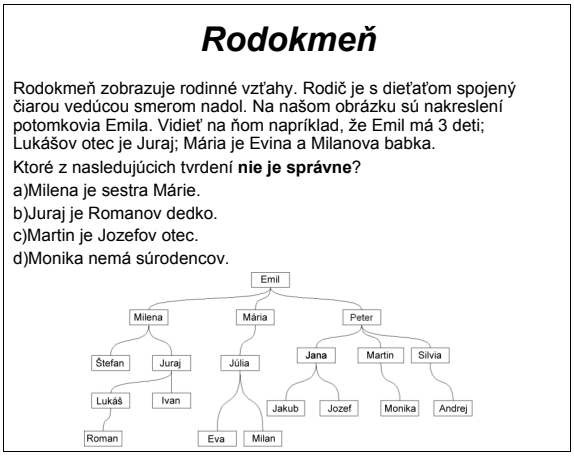
7. NAJJEDNODUCHŠIE ÚLOHY

Najjednoduchšie sa ukázali byť úlohy, v ktorých mali súťažiaci priradiť zodpovedajúce dvojice (úlohy B5, K4 a S1, juniori takúto úlohu nemali).

Tieto úlohy boli zaraďované ako ľahké, preto ich malá náročnosť nie je žiadnym prekvapením.



V úlohe K7 mali súťažiaci pochopiť informáciu v grafe. Medzi najjednoduchšími sa ocitla asi preto, lebo správna odpoveď sa týkala vzťahu otec-syn.



Úloha K10 nebola zložitá, bola však dosť technická. Preto sme neočakávali až takú úspešnosť, preto nás potešilo, že sa kadeti tejto úlohy nezľakli.

Zakódovaný obrázok

Bobry majú špeciálny spôsob kódovania obrázkov podľa nasledujúcej tabuľky:

A□, B■, C■, D■, E■, F■.

Napríklad obrázok bude mať kód:
EEEEAA;EAAAEA; EAAAEA; EEEEAA; EAAAEA;EAAAAE; EAAAAE;EEEEEA

Aký kód bude mať tento obrázok?

a)AABAAA;ACCAAA;ACCFAA;CCCFFA;EEEEEE;AEEEEA;
b)AABAAA;ACCAAA;ACCEAA;CCCEEA;FFFFFF;AFFFFA;
c)BBCBBB;BAABBB;BAAEBB;AAAEEB;FFFFFF;BFFFFB;
d)AAABAA;AAACCA;AAECCA;AEECCC;FFFFFF;AFFFFA;

Šesticu najjednoduchších úloh uzatvára úloha S9. Prispel k tomu zrejme fakt, že pred miestom, v ktorom sa odpovede rozlišujú sa nenachádzajú písmená A až D (ktoré sa zakódujú do jednociferného čísla).

Dešifrovanie

Bobor Braňo a bobrica Barbora si často píšu listy. Nechcú však, aby ich vedeli čítať ostatné bobry. Vymysleli si takúto šifru: dané písmeno nájdu v tabuľke a číslo pri ňom vynásobia dvoma. Napríklad slovo BOBOR zašifrujú ako 45045056.

Aké slovo zašifrujú ako 18381250581056741858?

A) INFOSTANICA
B) INFORMATIKA
C) INFOSERVIS
D) ani jedna z možností

A 1	H 8	O 25	V 37
B 2	I 9	P 26	W38
C 3	J 15	Q 27	X 39
D 4	K 16	R 28	Y 45
E 5	L 17	S 29	Z 46
F 6	M 18	T 35	! 47
G 7	N 19	U 36	? 48

8. NAJNÁROČNEJŠIE ÚLOHY

Úloha J14 bola zaradená ako ťažká a jej skutočný výsledok to potvrdil. Úloha vyžadovala veľmi dobré pochopenie zápisu v netradičnom „programovacom jazyku“.

Smajlíky

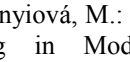
Bobri zistili, že smajlíky sa dajú nakresliť pomocou troch operácií Kruh, Rotuj a Zrkadlo a štyroch symbolov :, ; , -,)

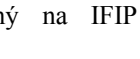
Kruh [] – nakreslí modrý kruh, v ktorého vnútri sú symboly uzatvorené v zátvorkách,

Rotuj [] – otočí symboly v zátvorkách o 90 stupňov v smere hodinových ručičiek,

Zrkadli [] – vytvorí zrkadlový odraz symbolov uzatvorených v zátvorkách, umiestni ho vpravo od pôvodných symbolov

Rotuj [Kruh [: -)]]

Rotuj [Kruh [: - Rotuj [Rotuj [ )]]]] 

Rotuj [Zrkadli [Kruh[: - Rotuj [-]]]] 

Ktorá z nasledujúcich postupností nakreslí takéto žmurkajúce dvojičky smajlíkov?

a) Zrkadli [Rotuj [Kruh [Rotuj [Rotuj [;]] -)]]]
b) Zrkadli [Rotuj [Kruh [; -)]]]
c) Rotuj [Zrkadli [Kruh [Rotuj [Rotuj [;]] -)]]]
d) Kruh [Zrkadli [Rotuj [; -)]]]

Úloha S10 ukazuje typickú chybu pri kopírovaní vzorcov v tabuľkovom procesore. Preto nás prekvapilo, že ju vyriešilo len 19% súťažiacich. Zrejme sa tabuľkovým procesorom nevenuje v školách taká pozornosť, aby boli žiaci schopní riešiť netriviálne úlohy.

Tabuľka s DPH

V tabuľke máme informácie o cenách tovarov bez DPH.

	A	B	C	D	E	F
1	**tovar**	**cena bez DPH**	**cena s DPH**		daň	19%
2	basketbalová lopta	829,41 Sk	=B2+B2*F1			
3	lyže	5 176,47 Sk				
4	tenisová raketa EVO 3	2 532,77 Sk				
5	tenisová loptička TITANUM 4BT	252,10 Sk				
6	kolieskové korčule Moto 90	6 841,18 Sk				

Do bunky C2 sme napísali nasledujúci vzorec: =B2+B2*F1. Tento vzorec sme skopírovali do buniek C3 až C6. Aká hodnota bude v bunke C6?

a) 8 141,00 Sk
b) 6 841,18 Sk
c) 136 823,60 Sk
d) 987,00 Sk

9. ZÁVER

V našej práci sme analyzovali riešenia v súťaži iBobor. Mnohé naše zistenia potvrdzujú, že náročnosť úloh bola primeraná (skôr nižšia v mladších kategóriách) a že extrémne ľahké či ťažké úlohy sa nevyskytli vo veľkom množstve.

Analýza však priniesla aj niekoľko námetov zamyslenie, resp. zlepšenie v ďalších ročníkoch iBobra.

Predovšetkým je to také zlepšenie zatriedenia úloh do skupín podľa náročnosti, ktorým sa nezmení charakter iBobra zo súťaže na testovanie školských vedomostí. Preto niektoré metódy, bežne používané pri testovaní (napríklad adaptívne testovanie alebo kalibrácia príkladov na vybranej skupine žiakov), sa nám nejavia ako vhodné.

LITERATÚRA

[1] Dagiene, V., and Futschek, G.: Bebras International Contest on Informatics and Computer Literacy: Criteria for Good Tasks. In R.T. Mittermeir and M.M. Syslo (Eds.): ISSEP 2008, LNCS 5090, pp. 19–30, Springer, Berlin Heidelberg, 2008.

[2] Hrušecká, A., Pekárová, J., Tomcsányi, P., Tomcsányiová, M.: Informatický bobor – nová súťaž v informačných technológiách pre žiakov základných a stredných škôl. Banská Bystrica : Didinfo, 2008.

[3] Chráska, M.: Metody pedagogického výzkumu, Grada, Praha 2008, ISBN 978-80-247-1369-4

[4] Kalaš, I., Tomcsányiová, M.: Students' Attitude to Programming in Modern Informatics. príspevok zaslaný na IFIP WCCE Bento Gonçalves, 2009.

[5] http://www.bebras.org

[6] http://www.iBobor.sk

[7] http://www.uips.sk/prehlady-skol/prehlad-strednych-skol

INTERAKTÍVNE ÚLOHY V SÚŤAŽI INFORMATICKÝ BOBOR

PAEDDR. MONIKA TOMCSÁNYIOVÁ, PHD.

Katedra základov a vyučovania informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina, Bratislava,
tel. 02/602 95 284, e-mail: tomcsanyiova@fmph.uniba.sk

Po prvých dvoch ročníkoch súťaže Informatický bobor, ktoré sme realizovali a v ktorých spolu súťažilo 13384 žiakov, v článku bližšie skúmame interaktívne úlohy, ktoré sú neoddeliteľnou súčasťou súťaže. Pomocou nich sa snažíme súťaž oživiť, vniesť do nej hravosť, atraktívnosť a dosiahnuť, aby zaujala žiakov základných a stredných škôl. Ďalej sa zamýšľame nad rôznymi druhmi interakcie v úlohách, a uvádzame, aké typy úloh sú vhodné na interakciu. Bližšie tiež analyzujeme jednu interaktívnu úlohu z 2. ročníka, v ktorej sledujeme prístup žiakov k jej riešeniu.

Kľúčové slová: *on-line súťaž Informatický bobor, interaktívne úlohy*

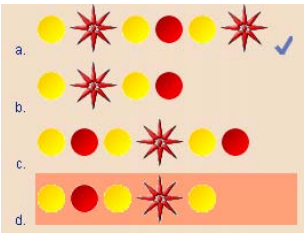
1. ÚVOD

V školskom roku 2008/09 prebehol na Slovensku 2. ročník súťaže Informatický bobor. Zúčastnilo sa ho 9317 žiakov v štyroch súťažných kategóriách (iné krajiny majú tri súťažné kategórie). Reakcie učiteľov aj žiakov na súťaž boli prevažne pozitívne. Podstatnou zmenou v tomto ročníku bol práve vznik novej kategórie s názvom Kadeti. Predovšetkým na podnet učiteľov sme rozdelili skupinu súťažiacich, ktorú tvorili žiaci druhého stupňa ZŠ a nižších ročníkov osemročných gymnázií, na dve kategórie – Benjamíni (5., 6., 7. ročník, príma, sekunda) a Kadeti (8., 9. ročník, tercia, kvarta). Podľa počtu účastníkov majú títo žiaci o súťaž najväčší záujem (Benjamínov súťažilo 3447 a Kadetov 2813) a ich rozdelením na dve kategórie sa pre školy rozšírila možnosť zapojiť do súťaže viac žiakov pri maximálnom využití svojich počítačových učební. Po ukončení súťaže sme pripravili anketu pre žiakov aj učiteľov. Z ankety žiakov okrem iného vyplynulo, že spôsob bodovania, ktorý zahŕňa nielen kladné body pri správnej odpovedi, ale aj záporné pri nesprávnej odpovedi, je pre nich dosť neobvyklý. Ankete učiteľov, ktorú vyplnilo 50 učiteľov, sa budeme podrobnejšie venovať v časti 6 tohto príspevku.

2. SÚŤAŽ JE INTERAKTÍVNA

Súťaž Informatický bobor sme na Slovensku od začiatku navrhli ako interaktívnu, t.j. takú, že ju žiaci riešia on-line vo vyhradenom čase s použitím webového prehliadača, pozri tiež [2]. Interakcia žiak-počítač má v rámci súťaže rôzny charakter:

- interaktívne **vyznačovanie odpovedí žiaka** – takýto typ interakcie je charakteristický pre samotný priebeh súťaže, pričom v každej otázke si žiak vyberá zo štyroch odpovedí,
- žiaci sa ihneď po jej ukončení **dozvedia svoj bodový zisk**. Kvôli možnosti prezradenia odpovedí sa ešte nedozvedia, na ktoré úlohy odpovedali správne a nesprávne, ale systém im prezradí, koľko bodov v súťaži získali,
- relatívne rýchlo (v priebehu týždňa) môžu žiaci **vidieť svoje umiestnenie** v celkovom rebríčku súťaže a môžu sa tak porovnať s ostatnými súťažiacimi,
- uchovávanie údajov v databáze umožňuje, aby si **neskôr každý žiak pozrel riešenia svojich úloh** a zároveň ich porovnal so správnymi odpoveďami. K úlohám sa takto môžu vracať v diskusiách so spolužiakmi, prípadne na hodinách informatiky. Súťažiaci pri prezeraní riešení vidí zadanie otázky, zvýraznenú svoju odpoveď a tiež správnu odpoveď na každú otázku súťaže.



Obr. 1 Označenie odpovede žiaka v rámčeku a správnej odpovede „fajkou“

- niekoľko týždňov po ukončení súťaže je k dispozícii **archív súťaže**, v ňom môžu riešiť úlohy všetci, ktorí majú o súťaž záujem – až 38% učiteľov používa archív súťaže v rámci vyučovania.

Interaktívnosť súťaže je umocnená aj tým, že v každej kategórii je niekoľko **interaktívnych úloh**, t.j. žiak rieši úlohu pomocou klikania do aplikácie, ktorá je jej súčasťou. Žiaci sú zvyknutí z rôznych internetových stránok na interakcie (predovšetkým z hier vytvorených v Javascripte, Jave alebo vo Flashi), mnohí z nich už absolvovali takýto spôsob preverovania vedomostí aj na iných predmetoch – v ankete 65% učiteľov potvrdilo, že ich žiaci sa zúčastňujú rôznych súťaží, najčastejšie uvádzali logickú súťaž Genius Logicus a stránku www.skolahrou.sk.

3. INTERAKTÍVNE ÚLOHY

Úlohy s interakciou žiaka majú v súťaži Informatický bobor významné miesto, vnášajú do nej dynamiku a tiež grafické prvky, čím sa súťaž stáva ešte atraktívnejšou a pútavejšou.

V súťaži sme doteraz navrhli interaktívne úlohy niekoľkých typov:

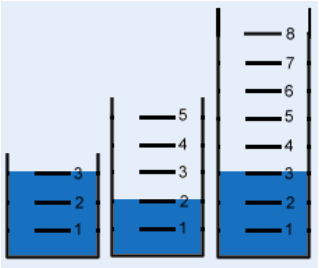
- **Písanie textu do vstupného riadku.** V takto koncipovaných úlohách je najdôležitejšie dobre navrhnúť a sformulovať samotný text zadania tak, aby odpoveď naň bola jednoznačná a skontrolovateľná počítačom (v rámci takejto softvérovej kontroly môže byť kontrolovaných aj niekoľko správnych odpovedí, ktoré sa líšia napr. veľkými a malými písmenami, či diakritikou). V rámci zadávania žiackych odpovedí je tiež dôležité, aby pripravený vstupný riadok neumožnil žiakom vpisovať znaky, ktoré nie sú správneho typu, teda ak napr. odpoveď vyžaduje číselný vstup, do vstupného riadku sa vôbec nedajú zapisovať iné znaky ako číslice. Takéto úlohy boli doteraz v súťaži tri. V jednej z nich bolo potrebné do riadku vpísať krstné meno a v dvoch číslo.
- **Interakcia pri riešení úlohy**, t.j. po prečítaní zadania žiak vo flashovej aplikácii rieši úlohu interakciou pomocou myši (klikaním, ťahaním). Riešenie následne odošle na server, kde sa zapamätá ako ju ten-ktorý žiak vyriešil. Ak sa žiak ešte počas súťaže rozhodne k tejto úlohe vrátiť, zobrazí sa mu jeho doterajšie riešenie, ktoré môže v rámci časového limitu upravovať a znovu odoslať. Takýto spôsob ukladania žiackych riešení je významný aj preto, že po ukončení súťaže vidíme nielen to, koľko žiakov vyriešilo úlohu správne alebo nesprávne, ale aj to, ako vyriešili úlohu tí žiaci, ktorí ju vyriešili nesprávne. Takýto typ úlohy budeme bližšie analyzovať v časti 5.
- **Zadanie úlohy je interaktívne**, t.j. v rámci úlohy je zobrazená aplikácia, s ktorou žiak pracuje a sleduje zákonitosti, ktoré sú popísané v zadaní úlohy. Snaží sa im porozumieť a následne ich aplikovať pri svojom uvažovaní a riešení. V rámci 2. ročníka bola takáto úloha pre Juniorov, v ktorej mali žiaci po kliknutí na bobra sledovať animáciu, svoje pozorovania skonfrontovať s navrhovanými „tanečnými krokmi“ a vyznačiť správne riešenie úlohy.
- Interakcia, ktorá pomáha žiakovi správne alebo **lepšie pochopiť zadanie** úlohy. V takto pripravených úlohách je zadanie najprv popísané slovami, a popri tom si žiak môže interaktívne overiť, či toho zadanie pochopil správne. Takto koncipované zadanie je oveľa názornejšie aj pre tých, ktorí majú problémy s pozorným čítaním textu. Okrem pochopenia zadania úlohy môže interakcia viesť žiaka aj k hľadaniu riešenia. Tak to bolo v úlohe v 2. ročníku pre Juniorov – žiaci si mohli vyskúšať, akým spôsobom sa bobor „hrá“ so slovami – mohli klikať na písmená v slove JAZERO, pričom videli, akým spôsobom sa v tomto slove vymieňajú písmená podľa pravidiel, ktoré boli popísané v zadaní. Ak si žiaci v zadaní nevšimli, že sa vymieňa vždy iba prvé písmeno, pri práci s aplikáciou to pochopili a mohli tak úlohu riešiť s lepším porozumením jej zadania. Taktiež mohli experimentovať dovtedy, kým výmenami nedostali slovo, ktoré sa požadovalo v zadaní. A aj týmto spôsobom mohli určiť, aký najmenší počet výmen je potrebný na zmenu pôvodného slova na hľadané.

Interaktívny spôsob zadávania alebo riešenia úloh robí súťaž zaujímavejšou a často aj populárnejšou. Nevýhodou však je to, že pri interaktívne koncipovaných úlohách je pravdepodobnejšie, ako pri klasickej súťaži, že nastanú technické problémy. Podľa štatistiky vidíme, že 3,6% Benjamínov, 2,3% Kadetov a 2,8% Seniorov malo pri riešení problémy so zobrazením flashových aplikácií (Juniori nemali interaktívne riešenú úlohu, ale iba takú, ktorej zadanie bolo interaktívne a pri ktorej ho interakcia pomáhala lepšie pochopiť). Väčšina týchto problémov je spojená s tým, že na počítačoch je nainštalovaný rôzny softvér, školy používajú rôzne internetové prehliadače alebo sú na nich nastavené rôzne stupne zabezpečenia – a to často aj pre počítače v rámci jednej školy.

4. ÚLOHY VHODNÉ NA INTERAKCIU

Pri navrhovaní úloh pre žiakov, pozri aj [1], je potrebné dobre zvážiť, ktoré úlohy sú vhodné na to, aby mali nejaký z vyššie spomínaných druhov interakcie. Pre interaktívne zadanie, riešenie alebo lepšie pochopenie zadania považujeme za vhodné nasledujúce typy úloh:

- Rôzne druhy **logicko-algoritmických hier**, v ktorých je úlohou žiaka nájsť postup pre jej vyriešenie alebo riešenie objaviť a následne ho vyznačiť. Takýchto úloh bolo v súťaži najviac, spomeňme ukážkovú úlohu Sudoku, umiestňovanie čísel podľa rôznych pravidiel, či prelievanie tekutín v nádobách tak, aby v niektorej nádobe dostali príslušný počet litrov vody – 2007/08 Benjamíni, Juniori a tiež Seniori.



Obr. 2 Prelievaj vodu tak, aby boli v niektorej nádobe štyri litre

- Úlohy **na preverovanie znalostí**. Žiak môže napr. ťahať kartičky a vytvárať tak páry, ktoré patria k sebe. Takéto úlohy sa dajú pripraviť z rôznych oblastí informatiky a tiež rozličných stupňov náročnosti. Tri úlohy takéhoto charakteru boli v 2. ročníku súťaže – žiaci ich riešili bez problémov, správnu odpoveď našlo viac ako 93% Benjamínov, 87,5% Kadetov a 86,3% Seniorov.
- Úlohy, v ktorých je potrebné niečo **vypočítať, zistiť logickým úsudkom** a hodnotu zapísať do vstupného riadku. Takýto typ úlohy je vhodný najmä vtedy, keď by výber zo štyroch ponúkaných alternatív danú úlohu veľmi zjednodušil, pretože samotný úsudok alebo výpočet je v rámci jej riešenia dôležitý.
- Úlohy, ktoré umožnia žiakom **interaktívne si vyskúšať** na konkrétnych a malých vzorkách vstupných údajoch rôzne algoritmy, ako napr. rôzne druhy triedenia, kódovania, šifrovania – doteraz sme takúto úlohu v súťaži ešte nemali.
- Také úlohy, ktoré pomáhajú žiakom priamo **aplikovať nejaké postupy riešenia**, pričom toto riešenie graficky znázorňujú. Úloha sa stáva prístupnejšou a zrozumiteľnejšou pre širší okruh žiakov.
- Úlohy, v ktorých je **riešením obrázok** nakreslený nejakou pevne zvolenou skupinou príkazov, ktorú žiak nepozná, pozri tiež [3]. Pri interaktívne zadanej úlohe si súťažiaci môže najprv vyskúšať príkazy, overiť si, či im porozumel a následne tak správne vyriešiť úlohu.

V prvých dvoch ročníkoch súťaže sme z celkového počtu 105 úloh súťaže realizovali 18 interaktívnych úloh, t.j. 17% tak, ako to uvádza nasledujúca tabuľka:

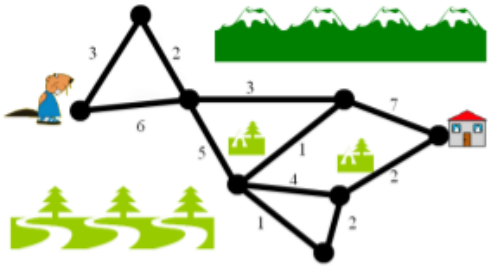
	2007/08	2008/09
Benjamíni	3	3
Kadeti	–	3
Juniori	3	2
Seniori	3	1

Okrem toho sú na stránkach súťaže ešte ďalšie dve interaktívne úlohy v rámci ukážkových úloh.

5. ÚLOHA NAJRÝCHLEJŠIA CESTA

K typu interaktívnych úloh, ktorých riešenie úlohy vyžaduje interakciu, patrí úloha z 2. ročníka súťaže v kategórii Kadeti o hľadaní najkratšej cesty.

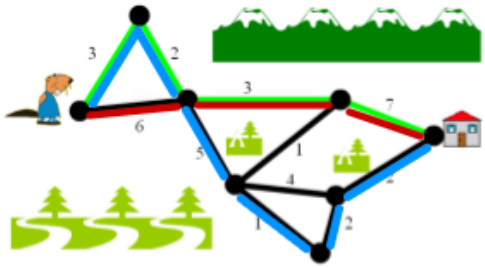
Bobor sa ponáhľa domov. Pomôž mu nájsť najrýchlejšiu cestu. Pri každom úseku cesty vidíš, za koľko minút ho bobor prejde. Klikaním označ najrýchlejšiu cestu od bobra k domčeku.



Obr. 3 Obrázok k súťažnej úlohe

Úlohu v súťaži videlo 2755 žiakov (58 malo technické problémy s jej zobrazením kvôli nefunkčnosti flashových aplikácií na ich počítačoch). Správne riešenie malo 1294, nesprávne ju vyriešilo 1408 žiakov a pre možnosť neriešiť úlohu sa rozhodlo 53 žiakov. Neriešenie úlohy znamená, že žiak za úlohu body nezíska, ale ani nestratí. Pri správnom riešení žiak získava body podľa náročnosti úlohy, a to 3, 4 alebo 5 bodov. Pri označení nesprávneho riešenia úlohy žiak body stratí, a to opäť podľa náročnosti úlohy, a to buď 1 bod pri ľahkej úlohe, 1,33 bodov pri stredne náročnej alebo 1,66 bodov pri ťažkej úlohe.

Ďalej sme sa zamerali iba na tých žiakov, ktorí úlohu **vyriešili nesprávne**. Z databázy riešení sme zistili, že 854 žiakov ako najkratšiu cestu našlo riešenie, ktoré je na našom ilustračnom obrázku vyznačené zelenou farbou.



Obr. 3 Nesprávne riešenia, ktoré označilo najviac žiakov

Z analýzy tohto najčastejšieho nesprávneho riešenia môžeme usúdiť, že žiaci si zadanie úlohy prečítali správne – pri hľadaní cesty z miesta, kde stojí na začiatku bobor, našli také dve cesty, ktorých prejdenie mu trvá iba 5 minút a nie 6 minút, ktoré sú určené pre jednu cestu. Z analýzy ich ďalšieho riešenia však vidíme, že pri poslednom úseku si potom nesprávne vybrali cestu, ktorá sa iba javí ako najkratšia. Môžeme sa iba domnievať, že zrejme preto, že neboli schopní hľadať cesty, ktoré bobrovi trvajú kratšie tým spôsobom, aby v grafickom znázornení ciest videli také, ktoré sú od priamej cesty topologicky vzdialené, pozri tiež [4]. Ďalšie najčastejšie nesprávne odpovede – 183 žiakov – sú na obrázku vyznačené ako modrá cesta od bobra k domčeku. Červenou sme vyznačili v poradí tretie najčastejšie nesprávne riešenie, ktoré zvolilo 178

žiakov. Ďalších súťažiacich, ktorí našli ešte iné nesprávne riešenia môžeme rozdeliť do dvoch kategórí – takých, čo našli riešenie, ktoré prechádza trasu od bobra k domčeku, pričom po každej z ciest ide práve raz (111 žiakov) a tých, ktorí vyznačli neúplnú cestu alebo vyznačujú viac ciest (62 súťažiacich).

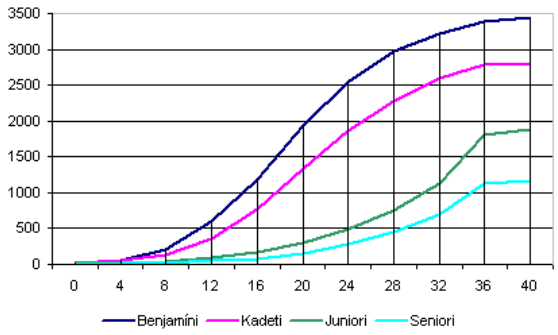
6. RIZIKÁ ON-LINE SÚŤAŽENIA

Okrem toho, že on-line súťaž je v súčasnosti typom súťaže, s ktorým sa žiaci veľmi často nestretávajú, a aj preto je zrejme pre deti atraktívna, môže mať takto koncipovaná súťaž aj niekoľko rizík. Medzi najvýznamnejšie z nich (odhliadnuc od technických problémov) patria:

- **Opisovanie** – žiaci sedia pri počítačoch, ktoré sú často umiestnené veľmi blízko seba, a tak nie je problém nahliadnuť na susedný monitor a všimnúť si, akú odpoveď má vyznačenú spolužiak – tieto obavy uvádza 30% opýtaných učiteľov. Počas súťaže sa tomuto neduhu snažíme zabrániť obmieňaním poradia úloh a odpovedí pre každého súťažiaceho.
- **Prezradenie riešení** žiakom, ktorí budú súťažiť v nasledujúcich zmenách pri počítačoch (podľa štatistík zaťaženosti servera sa vo väčšine škôl v rámci troch hodín trvania súťaže pri počítačoch postupne vystriedali tri zmeny) – obavu o prezradenie však v ankete vyjadrilo iba 16% učiteľov.
- Takúto on-line súťaž môžu žiaci považovať za **hru**, t.j. zadanie čítajú rýchlo, často ho správne nepochopia, o samotnom riešení veľmi neuvažujú a tak ako v hrách, jednotlivé odpovede tipujú – v dotazníku to uviedlo až 66% učiteľov.

Obavy, že súťažiaci, hlavne v kategóriách Benjamíni a Kadeti, berú takto koncipovanú súťaž ako hru, potvrdzuje aj štatistika, ktorú sme vytvorili z údajov o dobe riešenia jednotlivých žiakov. Sledovali sme, ako dlho v rámci súťaže riešili žiaci úlohy, na ktoré mali časový limit 40 minút. Na kumulatívnom grafe vidíme počty žiakov v jednotlivých kategóriách a priebeh času, za ktorý žiaci ukončili súťaž – stlačením príslušného tlačidla na stránke. V kategórii Benjamíni je priemerný čas riešenia úloh 24:19 min., pričom do 25 minút súťaž ukončilo až 2549 benjamínov, čo je takmer 74% všetkých žiakov.

U Kadetov je priemerný čas riešenia 25:57 min. a 66% žiakov v tejto kategórii ukončilo súťaž do 28 minút. Priemerný čas riešenia u starších žiakov v kategórii Juniori a Seniori je vyšší. Z databázy sme odčítali priemerný čas pre obe kategórie 32 minút. Ako vidieť z grafu, viac ako 95% Juniorov a Seniorov súťažilo do 36. minúty.



Obr. 4 Kumulatívny graf časov súťaženia žiakov v jednotlivých kategóriách

7. ZÁVER

Doterajšie dva ročníky súťaže Informatický bobor poukazujú na to, že žiaci majú záujem o takýto druh súťaže. Jej výhodou je, že dokáže osloviť veľkú skupinu žiakov vo veku 10-18 rokov, a tak môže pozitívne ovplyvniť obľúbenosť informatiky a jej postupné priblíženie všetkým žiakom. Takto navrhnutá súťaž tiež umožní vzbudiť záujem žiakov a študentov o hlbšie znalosti z digitálnej gramotnosti, zoznámi ich s algoritmizáciou, umožní im riešiť rôzne typy úloh, uvedie ich do tajov kódovania, spracovávania informácií a oboznámi ich aj s ďalšími informatickými pojmami. Žiaci, aj vďaka súťaži, môžu lepšie pochopiť, čo je informatika a veríme, že sa začnú zaujímať o informatické pozadie aj v témach a oblastiach, za ktorými ho dovtedy nevideli.

LITERATÚRA

[1] Dagiene, V., and Futschek, G.: Bebras International Contest on Informatics and Computer Literacy: Criteria for Good Tasks. In R.T. Mittermeir and M.M. Syslo (Eds.): ISSEP 2008, LNCS 5090, pp. 19–30, Springer, Berlin Heidelberg, 2008.

[2] Hrušecká, A., Pekárová, J., Tomcsányi, P., Tomcsányiová, M.: Informatický bobor – nová súťaž v informačných technológiách pre žiakov základných a stredných škôl. Banská Bystrica : Didinfo, 2008.

[3] Kalaš, I., Tomcsányiová, M.: Students' Attitude to Programming in Modern Informatics. príspevok zaslaný na IFIP WCCE Bento Gonçalves, 2009.

[4] Piaget, J., Inhelder, B.: Psychológia dieťaťa. Bratislava : Sofa, 144 s. ISBN 80-85752-33-6.

DĚTI A LINUX.

RNDR. ŠÁRKA VAVREČKOVÁ, PHD.

Ústav informatiky, Filozoficko-přírodovědecká fakulta Slezské univerzity v Opavě,
tel. +420 553 684 366, e-mail: sarka.vavreckova@fpf.slu.cz, web: http://fpf.slu.cz/~vav10ui

Na počítačích ve většině škol a domácích počítačích využívaných dětmi najdeme většinou Windows. V tomto článku se zamýšlíme nad tím, zda by v některých případech nebylo lepší místo Windows používat Linux. Je všeobecně známo, že po finanční stránce Linux vítězí, ale pořád se objevují argumenty poukazující na složitost Linuxu, nedostatek výukových programů a her, a také odlišnost od „majoritního" operačního systému. V textu se budeme zabývat vztahem dětí k Linuxu, volbou vhodné distribuce Linuxu a grafického prostředí, problematikou správy zaměřenou na školství a především nabídkou výukových programů a didaktických her pro tento operační systém.

Kľúčové slová: *Linux, distribuce, KDE, GNOME, výukový software, didaktická hra, SMART Board*

1. ÚVOD

V současné době se děti dostávají do kontaktu s počítačem velmi brzy, ať už doma, u kamaráda nebo ve škole. Tento fakt nelze moc ovlivňovat, souvisí s dobou, ve které děti vyrůstají, a s požadavky, které jsou na ně kladeny ve vyšším věku. Rodiče a učitelé však mají možnost ovlivnit alespoň to, k čemu má dítě na počítači přístup.

Účelem je zajistit, aby programy na počítači neměly negativní vliv na vývoj dítěte, ba naopak měly vliv pozitivní. Dítě se potřebuje naučit orientovat v operačním systému, aby zvládalo základní úkony související s používáním počítače (třebaže s těmito kompetencemi není třeba spěchat), a také přivítá vhodné výukové programy a didaktické hry, od věci nejsou běžné pomůcky jako je kalkulačka nebo textový a grafický editor, starší děti vyžadují Internet.

V článku vycházím ze svých zkušeností při výuce informatiky na základní, střední i vysoké škole.

2. LINUX A DĚTI

Děti mají oproti dospělým obrovskou výhodu v přizpůsobivosti. Dospělí v mnoha případech přistupují k počítači nejdřív s jistými obavami, které je poněkud svazují v experimentech („co kdybych něco pokazil"), později s určitou rutinou, jsou raději, když je vše „na svém místě" a případné změny spíše odmítají. V kurzech se seznámí většinou jen s Windows a u toho také zůstávají.

Děti přistupují k počítači bez předsudků a bez předchozích očekávání ohledně toho, „jak se co dělá", mají jen určitou představu o tom, že lze hýbat myší a „někde" najít zajímavé hry nebo kreslení. Nemají problém přizpůsobit se během krátké doby hned několika různým prostředím, rychle si zapamatují, že když obrazovka vypadá takto, potom musí myší zajet do dolního/horního rohu vlevo, klepnout a hledat ve stromové struktuře to, co je zrovna zajímá. V těchto nejjednodušších úkonech si jsou Windows a Linux velmi podobné a není důvod odmítat Linux jen proto, že „vypadá jinak".

Devizou Linuxu vzhledem k dětem je nepřeberné množství programů včetně výukového softwaru a her, které se nainstalují už při instalaci operačního systému. Je pravda, že některé z nich nejsou dosud lokalizované do češtiny nebo slovenštiny, nebo sice ano, ale kromě nápovědy, ale v současné době se děti už na prvním stupni základních škol učí anglicky (s problémy se setkáváme spíše u vyučujících a rodičů) a navíc u některých programů vlastně slova ani nepotřebují. Pokud nainstalované programy nestačí, lze jednoduše (pomocí příslušného programu s grafickým rozhraním) nainstalovat další buď z instalačního média nebo z Internetu.

3. VHODNÉ DISTRIBUCE

Linux je systém Unixového typu, jehož hlavními vlastnostmi jsou důraz na bezpečnost a snadná dostupnost. Je šířen pod licencí GPL, která zaručuje, že uživatel má přístup ke zdrojovému kódu, může si takto získané programy jakkoliv upravovat (včetně změny zdrojového kódu a znovupřeložení) a také dále distribuovat (se zachováním autorských práv). Linux si můžeme koupit, ale také zcela legálně stáhnout na Internetu (většinou ve formě ISO obrazů instalačních CD nebo DVD), vypálit a nainstalovat. Získání formou koupě přichází v úvahu, pokud nemáme k dispozici rychlé připojení na Internet nebo požadujeme něco víc než jen instalační média (příručka v mateřském jazyce, instalační podpora, uživatelská podpora, certifikace apod.). V každém případě můžeme jedno instalační médium použít na jakémkoliv počtu počítačů (narozdíl od limitu u licencí a multilicencí Windows).

Linux není jenom jeden, ale existují stovky (možná tisíce) různých *distribucí*. Některé distribuce jsou obecně použitelné pro jakýkoliv účel (podobně jako Windows), jiné slouží některému konkrétnímu účelu a existují i distribuce určené pro vzdělávání. V současné době lze ve školství používat Linux čtyřmi způsoby:

1) LiveCD nebo LiveDVD – příslušné datové médium vložíme do mechaniky a restartujeme systém. Je nutné mít v BIOSu v posloupnosti boo-

tovatelných médií tuto mechaniku jako první, aby přednostně nenastartoval systém z pevného disku počítače. Právě z důvodu tohoto nastavení v BIOSu není metoda zcela bezpečná, přesto je použitelná v učebnách, které jsou pod dohledem (nebo spustíme ISO obraz ve virtuálním počítači). Další nevýhodou je náročnost rozšíření o další software, a především závislost rychlosti běhu systému na rychlosti CD/DVD mechaniky. Tato forma se doporučuje také pro vyzkoušení, ať už přímo v učebně, nebo zapůjčením studentům (žákům) domů. Ve školství se takto používá například česká distribuce *Danix*, *Ubuntu* nebo *Mandriva One*.

2) Tenký klient – tuto formu lze použít i u velmi starých počítačů nebo bezdiskových stanic (klientů), je třeba mít jeden výkonný počítač jako server. Jednotlivé počítače-klienty fungují jako terminály (samozřejmě s grafickým prostředím), systém běží pouze na serveru. Požadavky klientů jsou zasílány sítí serveru, který posílá zpracovaná grafická i negrafická data zpět konkrétní komponentě (jako je grafická karta). Jde o velmi levné řešení, ovšem je třeba počítat s vyšší zátěží sítě. Tímto způsobem lze připojit dokonce i notebooky. Existují distribuce přímo určené pro tenké klienty, například *PXES* Linux, případně lze použít *LTSP* (The Linux Terminal Server Project) v kombinaci s některou podporovanou distribucí.

3) Tlustý klient – podobně jako u tenkého klienta bootování probíhá po síti (tj. instalace je provedena jen na serveru), ale po nabootování vše běží na klientovi. Oproti předchozí metodě je síť silněji vytížena pouze při spouštění počítačů nebo při „hromadném“ spouštění další aplikace, na klienty jsou kladeny vyšší nároky ohledně vybavenosti operační pamětí a procesorem. Jde vlastně o jakési sdílení disku na serveru. Pro tento účel lze přizpůsobit většinu distribucí.

4) Linux je nainstalován na každém počítači zvlášť – počítače mohou být samozřejmě propojeny, ale každý z nich obsahuje plnohodnotnou instalaci systému. Uživatelé se přihlašují pouze lokálně, běžným způsobem však lze sdílet data, případně aplikace nebo připojení k Internetu. Existuje také možnost sdílení uživatelských účtů a adresářů, pak můžeme využít i přihlašování po síti. Lze použít prakticky jakoukoliv distribuci Linuxu, existují také postupy, jak zautomatizovat instalaci na více počítačů stejného typu.

Na jednom počítači můžeme mít nainstalováno více operačních systémů, není důvod se například zbavovat již dříve nainstalovaných Windows. Většina distribucí se při instalaci zeptá, zda chceme použít oblasti na disku, které původně patřily Windows (čímž Windows zlikvidujeme), a nebo je zmenšit a vytvořit vlastní pro Linux. Pak vždy při startu počítače jsme tázáni, který systém chceme spustit.

Velmi populární distribucí ve školství (a také na domácích počítačích, ke kterým mohou přistupovat děti) je *EdUbuntu*. Ve skutečnosti jde o distribuci *Ubuntu* nebo *KUbuntu* rozšířenou o podporu výuky, především výukové programy a didaktické hry. Další populární distribucí je český *Danix*, jehož varianta Evolution je přímo učena pro školství. Danix je LiveDVD (bez instalace) a taky se dá nainstalovat. Na některých školách najdeme distribuci *Mandriva*, a to opět buď jako LiveDVD (Mandriva One) nebo instalovanou na disku. Z dalších použitelných distribucí jmenujme *OpenSUSE*, které se vyznačuje značnou mírou lokalizace do češtiny/slovenštiny, *Debian* s obrovskou zásobou instalovatelných aplikací, *Fedoru* (instaluje se) nebo *Knoppix* (Live).

4. GRAFICKÉ PROSTŘEDÍ

V Linuxu existuje více grafických prostředí a uživatelé mají možnost vybrat si to, které jim vyhovuje. Každá distribuce má jedno přednastavené a obvykle je možné doinstalovat i jedno nebo více dalších. Tato možnost je poněkud omezena u Live distribucí. Totéž prostředí může u různých distribucí vypadat trochu jinak, programátor distribuce si každé prostředí obvykle trochu přizpůsobí.

Jednotlivé programy jsou většinou psány vždy pro určité prostředí, ale ve skutečnosti fungují i v jiných prostředích. Jedinou podmínkou je instalace potřebných knihoven (něco jako DLL knihovny ve Windows), aby tyto programy měly dostupný zdroj potřebných prvků. Závislosti na knihovnách jsou při instalaci programů řešeny automaticky, uživatel může být dotázán, zda souhlasí s instalací dalších softwarových balíků.

Nejpoužívanějším grafickým prostředím ve školství je zřejmě *KDE* (The K Desktop Environment). Je hodně podobné Windows, ale narozdíl od Windows velmi dobře a rozsáhle konfigurovatelné. V současné době jde o výchozí prostředí ve většině distribucí ve školství. Na výkonnějších počítačích se používá 3D akcelerace (prostředí *Compiz-Fusion*).

Dalším rozšířeným grafickým prostředím je *GNOME*. Oproti KDE je celkově jednodušší, o něco méně konfigurovatelné, méně náročné na zdroje (procesor, paměť) a méně se podobá Windows. Také u GNOME lze použít 3D prostředí.

5. ADMINISTRACE A SPRÁVA SOFTWARU

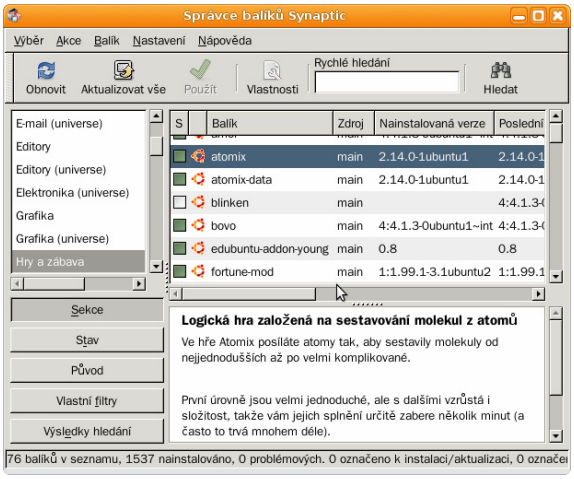
Z důvodu nedostatku místa se zde nebudeme o tomto tématu příliš rozepisovat. V tomto směru doporučujeme postupy publikované školami, které v minulosti řešily přechod na Linux, například podrobný tutoriál zveřejnilo Arcibiskupské gymnázium v Praze ([3]) nebo Jaroslav Krejčí z jedné ze základních škol v Přerově ([2]). Další odkazy jsou v [1]. Linux je už v základní instalaci dobře vybaven nástroji pro správu systému. S běžně dostupným hardwarem obvykle nebývají problémy, někdy je

potřeba doinstalovat proprietální ovladače (ke stáhnutí na Internetu) pro grafickou kartu, pokud chceme využívat 3D efekty. Software pro vedení účetnictví, hromadnou korespondenci a fakturaci také existuje a rok od roku je v této oblasti větší výběr.

Na škole se hodí jednoduchý software pro správu uživatelů, serverů, sledování zatížení a automatizaci správy. Pro tento účel můžeme použít programy (snad spíše webová rozhraní) *Webmin* a *Usermin*. První z nich je určen pro administrátory, druhý pro běžné uživatele, oba existují i v české verzi.

Školní prostředí má na software také své specifické požadavky. Může se hodit aplikace, která umožní vyučujícímu kontrolovat, co se děje na žákovských počítačích. K tomuto účelu lze použít aplikaci *Teacher Control Panel*, která zobrazuje na učitelově počítači náhledy obrazovek ostatních počítačů. V programech pro eLearning a obecně pro zveřejňování kurzů a souvisejících aktivit na webu je pro Linux široký výběr. Kromě dobře známého *Moodle* máme k dispozici *Claroline*, *LMS Olat*, *LON-CAPA*, *eSchool Builder* a další.

Na mnoha školách se používá interaktivní tabule (SMART board) a na připojeném počítači potřebujeme software na její ovládání. V Linuxu můžeme pro tento účel využít software ze stejného zdroje jako v případě Windows, *SMART Notebook*.



Obr. 1 Správa softwaru v prostředí GNOME (program *Synaptic*)

Správa softwaru v Linuxu není složitá. Většina distribucí (resp. grafických prostředí) je dodávána s jednoduchými nástroji určenými k tomuto účelu, například v Ubuntu najdeme program *Synaptic* (viz obr. 1), jinde je *KPackage*, *Adept*, *RPM Drake* v Mandrivě, *Yast2* v SUSE nebo jiný. Vždy hledáme v hlavním menu položku Správa balíčků nebo podobnou, a nebo přímo název programu. Software je poskytován a evidován v tzv. *balíčcích*, tedy když instalujeme některou aplikaci, instalujeme jeden nebo více balíčků. Protože každý program má určité požadavky na systém, na který se instaluje (knihovny grafického prostředí, skriptovací jazyk, podpora zpracování grafiky nebo textu apod.), může se stát, že instalovaný balíček závisí na jiných balíčcích. Na tento fakt jsme upozorněni a dotázáni, zda souhlasíme. Každý program na správu softwaru má určeny zdroje, ze kterých bere informace o zobrazených balíčcích. Přímo v programu obvykle můžeme tyto zdroje ovlivnit (přidat, odebrat, zneaktivnit apod.). Protože tyto programy mají vlastní databázi všech dostupných balíčků, neobcházíme je při instalaci ani odinstalování, pokud to je možné.

Jak bylo výše řečeno, mnoho potřebných programů včetně výukových je nainstalováno zároveň se systémem. Najdeme jak jednoduché nástroje (kalkulačka, jednoduchý textový editor, grafický, apod.), tak i složitější (kancelářské programy, zpracování zvuku a videa, napojení fotoaparátu nebo kamery).

Další doinstalujeme pomocí programu na správu softwaru. Často není problém zprovoznit programy, které (zdánlivě) pro Linux nejsou určeny. Některé jsou programovány v multiplatformním jazyce (Java, Flash, .NET), ty fungují i v Linuxu bez dalších úprav, například novější tituly firmy Terasoft. Další (Windows EXE) se většinou dají zprovoznit pomocí *Wine*, což je jakési prostředí tvořící komunikační mezivrstvu mezi linuxovým jádrem a Windows aplikacemi. Wine je většinou již instalováno, je třeba jen provést konfiguraci pro konkrétní aplikace. DOS aplikace (hry, účetnictví) zprovozníme emulací *FreeDOS*u na *DOSEmu* nebo *DOSBox*u, existují také virtuální počítače (*QEmu*, *VirtualBox* apod.), do kterých můžeme nainstalovat jiný operační systém včetně Windows a v emulovaném systému spouštět aplikace, které se na Linuxu nedaří zprovoznit.

6. VÝUKOVÉ PROGRAMY

Asi nejznámějším balíkem výukových programů a her pro menší děti v Linuxu je *GCompris*. Tento program nabízí celou řadu aktivit cílených na děti od 2 do 10 let, všechny jsou dostupné přes jednoduché strukturované grafické rozhraní. Menší děti se mohou učit zacházet s myší a klávesnicí nebo poskládat puzzle, děti školou povinné si procvičují aritmetické operace, znalost hodin, uspořádání položek podle času, orientaci v bludišti, šachy, elektřinu, koloběh vody, ... Aktivit je velmi mnoho a navíc lze balík rozšířit o moduly z Internetu nebo vytvořit vlastní. Oficiální stránky projektu existují také ve slovenštině, samotné prostředí je dostupné i v češtině.

Konkurenčním projektem je *ChildsPlay*, taktéž obsahující sjednocující prostředí pro různé aktivity. Je psán v Pythonu, což opět znamená snadnou rozšiřitelnost – na domovské stránce projektu jsou ke stažení další aktivity (pluginy). V současné době se jedná o dva spolupracující projekty používající totéž rozhraní – ChildsPlay pro (didaktické) hry a *SchoolsPlay* zahrnující výukové moduly (ovšem taktéž mající formu hry).

Pro prostředí KDE (dá se však provozovat i v jiných prostředích) existuje balík výukových progra-

mů *KDE-Edu* (The KDE Education Project) pro děti od 3 do 18 let. Programy, které jsou jeho součástí, nemají přímo společné rozhraní, ale jejich rozhraní a nápověda jsou standardizované. Děti si mohou procvičovat písmenka (KLetters), slova a jejich význam (KAnagram, KHangMan – spíše v angličtině), zlomky (KBruch), vykreslovat funkce v 2D a 3D grafech (KAlgebra, KmPlot), počítat s procenty (KPercentage), učit se zeměpis (KGeography, Marble), procvičovat si chemii (Kalzium) a fyziku (Steps), atd.

Projekt *Tux4kids* zahrnuje programy TuxMath, TuxPaint a TuxType. Především TuxPaint, jednoduchý grafický editor s objevným, velmi dobře vyvedeným prostředím s razítky a grafickými štětci, zaujme i nejmenší děti, TuxMath procvičuje jednoduché matematické výpočty (například pouze zadání správného výsledku zachrání vesnici tučňáků před dopadem komety) a TuxType učí psát na klávesnici.

Velmi mnoho výukových programů existuje například pro chemii. Balík programů *GNOME Chemistery Utils* zahrnuje kromě jiného 2D editor chemických struktur, 3D zobrazení molekul, prohlížeč a editor krystalových struktur, periodickou tabulku s názvy prvků ve více jazycích, několik widgetů a také plugin do produktů Mozilly zobrazující molekulární struktury.

Mnohé výukové programy v Linuxu používají tzv. kartičkovou metodu (flash cards) – například *Multiplication Flash* ve výuce násobení. Existují také programy určené pro vyučující, kteří si chtějí vyrobit vlastní sady „kartiček“ členěné podle lekcí a dalších kritérií, například *Memorize Word Flashcard System* pro výuku angličtiny, ve kterém si můžeme vytvořit multimediální kartičky (včetně obrázků a zvuku, program dokáže automaticky stahovat data ze slovníků na Internetu, zvukové soubory s výslovností, a spolupracovat s Google Images). Princip flash cards ve zjednodušené formě používá také například program *KWordQuiz* (kromě toho i odpovědi s vícenásobným výběrem nebo formu otázka-odpověď). Na domovské stránce KWordQuiz jsou ke stažení již předpřipravené soubory s testy, kromě jiného také test základní slovní zásoby angličtina/čeština, další předpřipravené testy z několika různých předmětů lze upravit pro vlastní potřebu. Dalším jednoduchým flashcard systémem je *Parley*. Z tohoto programu je odvozen KWordQuiz a oba programy používají stejný formát pro vstupní soubory s daty (kartičkami) – KVTML soubory.

Větší děti, které si chtějí hravou formou procvičovat logické myšlení, využijí projekt *GBrainy*. Je to souhrn logických skládaček, trenažérů paměti a dalších podobných her s více úrovněmi obtížnosti.

Squeak je implementace (a také úprava) programovacího jazyka SmallTalk. Je to čistě objektový jazyk s grafickým vývojovým prostředím, obohacený o 2D a 3D grafiku, práci se zvukem a obecně multimédii a síťové funkce. Je upraven tak, aby mohl být používán dokonce i dětmi, které se učí číst a psát. Jeho počátky jsou spojeny s projektem One Laptop Per Child, kde je použit pro naprogramování grafického rozhraní. V souvislosti se Squeakem je třeba zmínit především prostředí *EToys* určené právě pro vzdělávání formou hry. Dalším projektem vyvíjeným ve Squeaku je *Scratch*, inspirovaný mnohem starším projektem Logo, ve kterém děti velmi jednoduše tvoří animované příběhy a hry.

Linux je tradičně vybaven více různými programovacími jazyky, a nebo je lze snadno doinstalovat. Kromě „běžných“ programátorských nástrojů a Squeaku lze dětem doporučit *Alice*, což je vývojové prostředí pro vytváření 3D příběhů, animací a interaktivních her přetahováním objektů myší. Je určeno pro děti, třebaže „za oponou“ vlastně tyto děti používají vyšší programovací jazyk objektového typu jako je Java nebo C#. Pro menší děti se nabízí jiný program, *Little Wizard* (Malý kouzelník) od Marcina Kwadranse, který se svým ovládáním a prostředím podobá populárnímu Baltíkovi od ing. Soukupa.

Při výuce mohou být využívány tzv. myšlenkové mapy (mind maps) – použijeme buď linuxovou variantu známé aplikace *FreeMind* nebo podobnou aplikaci *VYM* (View Your Mind).

7. ZÁVĚR

Pokud máme naprosto bez problémů fungující počítače s Windows, není třeba do nich zasahovat, Linux přichází v úvahu pro dualboot. Jestliže však pořizujeme nové nebo musíme provést větší zásahy do stávajících, pak má smysl o Linuxu uvažovat alespoň v jedné učebně. Podle vlastního průzkumu to děti většinou přivítají a rozhodně jim to neublíží.

Z nedostatku prostoru jsou (průběžně aktualizované) odkazy na všechny projekty zde zmíněné, a také další s tématem související, umístěny na Internet do zdroje [1].

LITERATURA

[1] Stránka se všemi (níže neuvedenými) aktualizovanými odkazy vztahujícími se k tomuto článku: http://fpf.slu.cz/~vav10ui/didact/ [26. 01. 2009]

[2] KREJČÍ, J. *GNU/Linux na základní škole* [online]. 2005, Česká škola.cz. Dostupné na: http://www.ceskaskola.cz/ICTveskole/AR.asp?ARI=102113&CAI=2140 Další díly seriálu jsou na http://www.ceskaskola.cz/Authors/default.asp?AUI=711 [12. 01. 2009]

[3] ŠMILAUER, V. – BERNARD, L. *Využití Linuxových terminálů ve výuce i školní infrastruktuře*. 2005. Dostupné na: http://www.arcig.cz/linux/ [22. 1. 2009]

INFORMATIKA – ČO TO JE?

RNDR. MICHAL WINCZER, PHD.

Katedra základov a vyučovania informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina, 842 48 Bratislava,
tel. 02/6029 5253, e-mail: winczer@fmph.uniba.sk

Uvažujeme nad tým, čo je to informatika ako disciplína a čo z toho vyplýva pre informatiku, ako predmet všeobecného vzdelania.

Kľúčové slová: *informatika, didaktika informatiky*

Computer science is no more about computers than astronomy is about telescopes.

Edsger W. Dijkstra

1. ÚVOD

Žijeme v ére informácií. Slová ako informatika, informatizácia, informačné technológie sa stali súčasťou bežného slovníka. Počítače, mobilné zariadenia, digitálne fotoaparáty, internet, to je len zopár príkladov technológií a zariadení využívajúcich digitálne technológie. Je ich okolo nás stále viac a viac, pričom si to často ani neuvedomujeme. Sú súčasťou automobilov, domácich spotrebičov, pomôcok na voľnočasové aktivity a mnohých ďalších zariadení. Napriek pozornosti, ktorá sa, aspoň navonok, venuje všetkému, čo s informatikou súvisí, sa zdá, že v širokej verejnosti stále nie je ustálený názor, čo informatika znamená. Takáto situácia nie je len u nás, ale aj v iných krajinách, napríklad aj v USA a Izraeli [1,2]. A na prvý pohľad paradoxne v zahraničí (napr. v USA) klesá záujem o jej štúdium, hoci v praxi môžeme pozorovať pravý opak, teda boom informačných technológií, ktoré informatika zastrešuje. Pri druhom pohľade nám príde na um otázka, či neklesá záujem o štúdium informatiky preto, lebo sa vyučuje zle, alebo sa pod predmetmi a štúdiom s týmto označením skrýva, v horšom prípade, niečo celkom iné?

Napriek tomu, že sa táto téma už na DIDINFe objavila [3,5] (hoci po nemecky) budeme sa jej venovať aj v našom príspevku.

Informatikou budeme označovať vednú disciplínu, aj predmet, na ktorom sa žiaci a študenti zoznámia s tým čo a ako skúma, čo tvorí jej náplň a to v širšom kontexte a s pohľadu všeobecného vzdelania. Pričom máme na mysli najmä žiakov a študentov základnej a strednej školy, ale podľa nášho názoru by sa uvedené mohlo týkať aj poslucháčov vysokých škôl, predovšetkým budúcich učiteľov, prípadne študentov iných odborov než informatika.

2. INFORMATIKA NIE JE LEN ...

Informatika nie je len používanie počítača, prípadne softvérových aplikácií, je to oveľa viac. Spomínané zručnosti sú nesporne dôležité, ale je veľkou chybou, ak sa chápanie informatiky redukuje len, alebo vo veľkej miere, na ne. Keď sa za vyučovaním informatiky skrýva najmä, alebo len, kurz používania počítača, operačného systému a základného softvéru (napríklad textového editora, tabuľkového kalkulátora, databázy, softvéru na tvorbu prezentácií, poštového klienta a pod.) mal by sa príslušný predmet radšej nazývať práca s počítačom a je otázne, či si môžeme takýto luxus dovoliť v rámci kurikula, v čase znižovania hodinových dotácií fundamentálnym predmetom.

Informatika nie je technológia. Neslúži výlučne na podporu vyučovania iných predmetov. Podľa [2] sa môžeme stretnúť s názormi, že informatiku charakterizujú špecifické zručnosti, kritické myslenie a schopnosť riešiť problémy, alebo v inom prípade, že je to vetva aplikovanej matematiky, prípadne, že sa zapodieva technológiou a aplikáciami. Tieto atribúty sú dôležité, ale buď nie sú špecifické len pre informatiku, alebo sú priveľmi úzko zamerané, alebo oboje.

Informatika nie je len programovanie, opäť je to viac. Zúženie len na programovanie má zvyčajne za následok odradenie väčšiny študentov. V [2] to pekne vyjadrili: „*programovanie v informatike je ako konštrukčný dôkaz v matematike a ako rozbor literárneho diela v slovenčine*“. Vyplýva z toho, že pred tým, než začneme vyučovať programovanie, je potrebná akási „odborná“ príprava, minimálne v spôsobe uvažovania podobne, ako štúdium matematiky nezačíname tým, že sa učíme formulovať tvrdenia a dokazujeme ich správnosť. Kým v matematike začneme tvoriť dôkazy (ak vôbec začneme) sa osem rokov venujeme zvládnutiu základných operácií, základom logickej argumentácie. „Odbornou“ prípravou pred programovaním, máme na mysli aktivity rozvíjajúce algoritmické myslenie[1], ktoré študentov čo najprirodzenejšie pripravia na potrebu automatického riešenia úloh. Samozrejme to môže byť aj

[1] Pod algoritmickým myslením, tu rozumieme viac než len tvorbu algoritmov. Zahŕňame sem aj schopnosť riešiť problémy, navrhnúť riešenie, schopnosť slovne ho opísať a tiež úsilie o jeho zdôvodnenie. Akýsi mix toho, čo sa v angličtine skrýva za termínmi: „computational thinking“, „problem solving“ alebo „algorithmic thinking“.

formou práce v detských programátorských mikrosvetoch[2]. Myslíme si, že až po takomto úvode je pripravená pôda na zoznamovanie sa s „ozajstným“ programovaním, čím myslíme nielen prepis viacmenej očividného riešenia nejakého problému do programovacieho jazyka, ale aj nájdenie riešenia netriviálneho problému, formulovanie riešenia vo forme algoritmu a jeho naprogramovanie a prípadné zamyslenie sa nad jeho zložitosťou.

Informatika je tiež viac než len teoretická informatika. I keď sa jedná o podstatnú časť informatiky, aj takéto zúženie by bolo ochudobňujúce, mohli by sme to prirovnať k zredukovaniu košatého stromu iba na jeho korene.

3. ČO INFORMATIKA JE A PREČO JE DÔLEŽITÁ?

V [5] sú odpovede na otázky: Patrí informatika medzi metavedy ako sú filozofia alebo matematika, či medzi prírodné vedy, či dokonca medzi spoločenské vedy, alebo je to inžinierska disciplína?

Informatika má črty metavedy, prírodnej vedy aj inžinierskej disciplíny, niekedy sa uvádza, že má tri zdroje: matematiku, inžinierstvo a prírodné vedy. Rovnako ako filozofia a matematika študuje všeobecné kategórie, akými sú náhoda, informácia, zložitosť, determinizmus, nedeterminizmus, dôkaz, komunikácia, algoritmus, aproximácia, simulácia a ďalšie. Rovnako ako prírodné vedy študuje konkrétne objekty a procesy a ich zákonitosti – informácie, programy, spracovanie informácií a iné. A rovnako ako inžinierske disciplíny sa venuje vytváraniu aplikácií a otázkam s tým spojeným (softvérové systémy, testovanie, modelovanie, plánovanie a pod.).

Informatika teda nie je len programovanie, ani len počítačová gramotnosť, ale vedecká disciplína, ktorá študuje výpočtové a algoritmické procesy, vrátane ich princípov, hardvéru, návrhu softvéru, ich aplikácií a dopadov na spoločnosť [9]. Informatika zahŕňa (podľa abecedy):

- aplikácie
- databázy a vyhľadávanie informácií
- logiku
- návrh hardvéru
- návrh softvéru
- počítačovú bezpečnosť
- počítačovú grafiku
- programovacie jazyky a paradigmy
- programovanie
- siete
- spoločenské súvislosti IT
- umelú inteligenciu
- zložitosť

Iný pohľad na informatiku je prezentovaný v [1], informatika sa tu prezentuje ako jazyk technológie. Technológia je aplikácia vedeckých a iných vedomostí, nástrojov a zručností, na riešenie praktických problémov a na rozšírenie schopností ľudí. Najčastejšie sa opisuje ako proces a prejavuje sa produktmi a ich dopadom na spoločnosť. Dnes je zrejmé, že „konceptuálne a technologické nástroje vytvorené v rámci informatiky začínajú mať široké uplatnenie mimo informatiky, najmä vo vedeckých disciplínach skúmajúcich komplexné systémy“ [8]. Informatika neustále prispieva k iným vedeckým oblastiam tým, že ukazuje, ako modelovať procesy v nich ako informačné. Stáva sa vzhľadom na iné vedy, najmä prírodné, rovnako fundamentálna ako matematika a vysvetľuje prírodné a fyzikálne zákonitosti (napríklad čo sa dá a čo sa nedá automatizovať, ako nám pomôže pri riešení úloh náhoda, a ďalšie). Koncepty informatiky poskytujú takú úroveň abstrakcie, že vedci z rôznych oblastí sa dokážu s jej využitím dorozumieť a vzájomne sa poučiť (fyzika, bioinformatika a iné). Podobne, ako sa naučili univerzálne aplikovateľné matematické zručnosti, informatika ponúka možnosť osvojiť si schopnosti riešenia komplexných problémov využitím univerzálnych počítačových prostredí. Na základe uvedeného ozaj môžeme považovať informatiku aj za formu jazyka.

Informatika je vedecká disciplína. Mala by sa jej venovať rovnaká pozornosť ako tradičným predmetom, pretože patrí medzi základné vedomosti každého vzdelaného občana súčasnosti, bez ktorých nemôže pochopiť svet okolo neho.

4. AKO JU UČIŤ?

Keď sme si informatiku charakterizovali, vyvstáva prirodzená otázka: Ako ju učiť? Pokúsime sa načrtnúť aspoň v hrubých črtách našu predstavu. Je zrejmé, že pri vyučovaní informatiky treba brať do úvahy schopnosti žiakov a prispôsobiť tomu aj spôsob a obsah. Myslíme si že vyučovanie informatiky by sa malo sústreďovať v rovnakej miere na:

- oboznámenie sa s hlavnými informatickými konceptmi v historickej perspektíve a s dôrazom na ich dopad na spoločnosť, prípadne iné vedecké disciplíny (napr. písmo, papier, kníhtlač a pod.),
- skúmanie a pochopenie rôznych foriem spracovávania, uchovávania a prenášania informácií na základe každodenných skúseností
- štúdium teoretických základov

Je zrejmé, že v rôznych etapách vzdelávania by mali mať uvedené súčasti rôznu váhu. Podľa nášho názoru, na základnej škole by mal byť dôraz na prvé dve oblasti. Podrobnejšie sme našu predstavu o spôsobe ako ich vyučovať formulovali v [7]. Na strednej škole, kde sa žiaci zoznamujú aj s abstraktnými pojmami a formálnejšími metódami, by sa mala zvýšiť vá-

[2] Takéto aktivity nechápeme v tomto článku ešte ako programovanie. V každom prípade ich považujeme ako prípravu na programovanie a súčasť získavania zručností vyjadrovať sa „formálnejším“ spôsobom.

ha tretej oblasti. Myslíme si, že tretia oblasť by mohla byť dominantná na strednej škole (v jej druhej polovici) pre tých študentov, ktorí by sa informatike chceli venovať hlbšie, prípadne aj ďalej na vysokej škole.

Ukazuje sa, že dôležitou informatickou kompetenciou je algoritmické myslenie [10]. Patrí do tej istej skupiny kompetencií ako čítanie, písanie a počítanie. Ovládanie jazyka nám umožňuje efektívne komunikovať, počítanie kvantifikovať, algoritmické myslenie nám pomáha systematicky a efektívne vykonávať úlohy a činnosti. Ale pozor, algoritmické myslenie nie je programovanie! Dôraz by mal byť na rozvoj ľudských zručností, ako napríklad v prístupe opísanom v [11]. Jedná sa o aktivity vo forme hry, pri ktorých ľudia vykonávajú jednotlivo alebo spoločne činnosť na základe algoritmov. Dôležité je, že sa pri tom vytvárajú schopnosti abstrakcie, reprezentácie informácií a výpočtového procesu. Podľa schopností žiakov, tieto môžu byť ďalej formalizované, napríklad aj na matematike. Samozrejme sem patria aj aktivity, ktoré sme spomenuli v 2. časti tohto príspevku.

Vynikajúci príklad spracovania informatiky pre stredoškolákov, ktorí majú o ňu hlbší záujem, je podľa nás kniha „Sedem divov informatiky, cesta na hranice toho čo sa dá vypočítať“ [4], ktorá bola na konferencii DIDINFO predstavená [3].

5. ZÁVER

Postavenie informatiky je v dnešnom svete neprehliadnuteľné. Sme presvedčení, že s ňou treba oboznamovať žiakov a študentov primeranou formou počas celého štúdia ak chceme, aby sa v živote dokázali plnohodnotne uplatniť.

Myslíme si, že napriek tomu, že existujú kvalitné osnovy predmetov informatická výchova a informatika na základnej škole a aj predmetu informatika na strednej škole, nie je vždy ich realizáciou zachovaný duch, ktorý mali ich tvorcovia na mysli. Jedna z možných príčin takéhoto stavu je aj nepochopenie úlohy informatiky v rámci všeobecného vzdelania. Veríme, že pokusom o sformulovanie uvedených názorov sa nám podarí rozprúdiť diskusiu, ktorá prispeje k lepšiemu pochopeniu predmetu informatika.

LITERATÚRA

[1] Cohen, A., Haberman, B.: Computer Science“ A Language of technologykej. Inroads – SIGCSE Bulletin, Vol 39, No. 4, December 2007.

[2] Fletcher, H. L., Lu, J. J.: Human Computing Skills: rethinking the K-12 Experience, Communications of the ACM, Vol 52., No. 2, February 2009.

[3] Hromkovič, J.: OPEN CLASS – Sieben Wunder der Informatik Čo všetko z informatiky možno vyučovať na strednej škole, zborník DIDIFO 2006, str.11–15.

[4] Hromkovič, J.: Sieben Wunder der Informatik, Teubner 2006

[5] Hromkovič, J.: Informatik und allgemeine Bildung, zborník DIDIFO 2006, str.7–10.

[6] Kalaš, I.: Digitálne technológie a vízie moderného vzdelávania, DIDINFO 2006, str. 28–33.

[7] Kalaš, I., Winczer, M.: Tvorivá informatika, Informatika okolo nás, SPN, Bratislava. 2007.

[8] Towards Science 2020 (2006) Microsoft Research, http://research.microsoft.com/en-us/um/cambridge/projects/towards2020science/downloads/t2020s_reporta4.pdf

[9] Stephenson, Ch.: The New Educational Imperative: Improving High School Computer Science, http://www.csta.acm.org/Communications/sub/DocsPresentationFiles/TCEAPres07.pdf

[10] Wing, J., M.: Computational thinking, Communications of the ACM, Vol. 49, No. 3, March, 2006.

[11] CS Unpluged programm, http://csunplugged.org

Katedra informatiky

Fakulta prírodných vied
Univerzita Mateja Bela
Tajovského 40
974 01 Banská Bystrica
http://www.fpv.umb.sk/kat/ki

Katedra informatiky FPV UMB v Banskej Bystrici je gestorom štúdia odboru Aplikovaná informatika, ako aj gestorom študijného programu Učiteľstvo informatiky v odbore Učiteľstvo akademických predmetov.

Vedecký študijný program **Aplikovaná informatika** (AI) má už 12-ročnú tradíciu. V rámci štúdia Aplikovanej informatiky 1. stupňa možno získať titul **bakalár (Bc.)**; ukončením 2. stupňa titul **magister (Mgr.).**

Katedra informatiky tiež vychováva **učiteľov informatiky** v kombinácii s ďalšími prírodovednými a humanitnými predmetmi. Toto štúdium je realizované dvojstupňovo, t.j. po úspešnom absolvovaní bakalárskeho štúdia študent pokračuje v magisterskom štúdiu. Učiteľské štúdium ukončené hodnosťou magister otvára možnosť pre najlepších absolventov absolvovať rigoróznu skúšku a získať titul PaedDr.

Katedru informatiky **vedie Doc. Ing. Ľudovít Trajteľ, PhD**. Na katedre pôsobia traja profesori, jeden mimoriadny profesor, dvaja docenti, ďalších sedem učiteľov s hodnosťou PhD. Ostatní učitelia sú zapojení do doktorandského štúdia (na získanie PhD.).

Katedra je zapojená aj do medzinárodnej spolupráce. Na základe bilaterálnych dohôd momentálne **spolupracuje so 7 zahraničnými univerzitami** – Oulu University of Applied Sciences v Oulu a Raahe (Fínsko), Waseda University Tokyo (Japonsko), Baskent University (Turecko), Fatih University Istanbul (Turecko), Univerzita Utrecht (Holandsko), TU Ostrava (Česká republika) a TU Chemnitz (Nemecko). Spolupráca prebieha formou výmenných študijných pobytov, ktoré umožnia našim najlepším študentom študovať v zahraničí. V tomto akademickom roku je opäť realizovaný študijný pobyt pre vybraných študentov našej katedry vo Fínsku v Oulu a Raahe.

SLOVENSKÁ INFORMATICKÁ SPOLOČNOSŤ

Slovenská informatická spoločnosť je mimovládna, nezisková stavovská (profesná) organizácia, pokrývajúca oblasti teoretickej i aplikovanej informatiky, softvérového inžinierstva, informačného inžinierstva, vzdelávania v informatike a príbuzné oblasti. Jej základným poslaním je vytvárať predpoklady pre rozvoj informatiky a informačných technológii na Slovensku. Bola založená v roku 1993 po rozdelení Česko-Slovenskej informatickej spoločnosti. Združuje odborníkov zaoberajúcich sa štúdiom a výskumom teoretických a metodologických problémov, odborníkov z informatickej praxe a aplikácií aj pedagógov v oblasti informatiky, zjednocuje ich úsilie a chráni ich záujmy.

Pri svojej činnosti spolupracuje s inštitúciami zaoberajúcimi sa teóriou, výukou a aplikáciami informatiky. Slovenská informatická spoločnosť reprezentuje slovenskú informatickú komunitu v medzinárodných nevládnych organizáciách ako IFIP (International Federation of Information Processing Societies) a CEPIS (Council of European Professional Informatics Societies) a udržuje kontakty s podobne orientovanými spoločnosťami v zahraničí. Poriada školenia, diskusie, exkurzie, konferencie, sympóziá a semináre. Vyvíja expertnú, publikačnú, vydavateľskú a ďalšiu odbornú činnosť v odbore informatika. Slovenská informatická spoločnosť predkladá iniciatívne príslušným štátnym a verejným orgánom a inštitúciám svoje stanoviská a návrhy. Pomáha im v koordinácii a v riešení zásadných otázok rozvoja informatiky a informačných technológií. Slovenská informatická spoločnosť umožňuje rast mladých informatikov podporou ich účasti na odborných podujatiach formou štipendií resp. organizáciou a podporou rôznych súťaží atď.

Slovenská informatická spoločnosť je odborným garantom Olympiády v informatike. Je tiež držiteľom licencie pre certifikačný program ECDL (European Computer Driving Licence) na Slovensku.

Ďalšie informácie sa dajú nájsť na www.informatika.sk, kde sa nachádza aj formulár prihlášky za člena spoločnosti.

Metodicko-pedagogické centrum je vzdelávacia inštitúcia pre učiteľov na Slovensku. Jej zriaďovateľom je Ministerstvo školstva Slovenskej republiky. Spolupracuje so štátnou školskou správou, Štátnou školskou inšpekciou, ŠPÚ, NÚCEM, vysokými školami a inými vzdelávacími a výskumnými inštitúciami na Slovensku i v zahraničí.

Metodicko-pedagogické centrum ponúka rozvoj kompetencií učiteľov materských, základných, stredných a špeciálnych škôl, školských zariadení:

- *na výkon riadiacich a pedagogických funkcií;*
- *v uplatňovaní inovácií z pedagogiky, psychológie, etiky, odboru, príbuzných vedných disciplín v riadení vyučovania a učenia sa žiakov;*
- *v uplatňovaní komunikačných technológií a didaktických prostriedkov vo vyučovaní;*
- *v tvorbe a riadení školských projektov – vrátane zahraničných projektov.*

Ponúka odborno-metodické materiály.
Vydáva odborno-metodický časopis **Pedagogické rozhľady**.

V súčasnosti zvyšujeme ponuku priamej pedagogicko-poradenskej činnosti školám, školským zariadeniam a neprezenčného vzdelávania učiteľov aj prostredníctvom novovzniknutých regionálnych pracovísk.

Metodicko-pedagogické centrum Ševčenkova 11, 850 05 Bratislava
generálne riaditeľstvo
Pracovisko: Tomášikova 4, 820 09 Bratislava
Tel.: 02 432 93 501

Regionálne pracoviská

- ***BANSKÁ BYSTRICA***
  Horná 97, 975 46 Banská Bystrica
  Tel.: 048 472 2999
- ***BRATISLAVA***
  Ševčenkova 11, 850 05 Bratislava
  Tel.: 02 682 09910
- ***KOŠICE***
  Ul. Zádielska 1, 040 01 Košice
  Tel.: 0917 516 196
- ***NITRA***
  Kozmonautov 5. 949 01 Nitra
  Tel.: 0917 516 191
- ***PREŠOV***
  T. Ševčenku 11, 080 40 Prešov
  Tel.: 051 772 3451
- ***TRENČÍN***
  Pod Sokolicami 14, 911 01 Trenčín
  Tel.: 032 744 3368
- ***ŽILINA***
  Predmestská 1613, 010 01 Žilina
  Tel.: 0911 922 070

Ústav informácií a prognóz školstva
ŠKOLSKÉ VÝPOČTOVÉ STREDISKO
BANSKÁ BYSTRICA

Školské výpočtové stredisko vzniklo 18. júla 1987.
Od 1. 1. 2008 je pracoviskom Ústavu informácií a prognóz školstva Bratislava.
Svoju činnosť realizuje v nasledovných oblastiach:

Informačná oblasť :
Server ÚIPŠ - ŠVS BB prezentuje informácie o stredných školách v rámci prijímacieho pokračovania, informačný systém maturita, MS Select, programy na odvírovanie počítačov, výukové programy, výsledky súťaží.

Spracovateľská oblasť, tvorba portálových riešení:
Spracovanie informácií pedagogického procesu regionálneho školstva: prijímacie pokračovanie na stredné školy, vzdelávacie poukazy, maturity, celoštátne testovanie deviatakov, výkazníctvo. Spracovanie je realizované na úrovni škôl, školských zariadení, Krajského školského úradu, VUC, NUCEM, MŠ SR s podporou moderných otvorených portálových riešení, ktoré si vyžadujú minimálne finančné náklady.

Školiaca oblasť:
Organizovanie školení a vzdelávaní, ktoré sú zamerané na Windows, MS Office, Open Office, databázy, programovacie jazyky, Internet, WWW programovanie, využitie otvoreného softvéru v školstve, Unix, Linux.. Realizujeme e-learningové vzdelávanie pre zamestnancov školstva.

Metodická oblasť:
Zabezpečujeme metodickú pomoc základným a stredným školám pri spracovaní prijímacieho pokračovania na stredné školy, centrám pedagogicko-psychologického poradenstva pri organizovaní výstav a búrz informácií pre žiakov končiacich povinnú školskú dochádzku ("Stredoškolák"). ZŠ a SŠ poskytujeme odborno-metodické poradenstvo pri zavádzaní a využívaní IKT vo vzdelávaní.

Iné:
V spolupráci s rôznymi organizáciami zo školstva a iných rezortov organizujeme a zabezpečujeme súťaže a podujatia miestneho, krajského, celoštátneho a medzinárodného charakteru (školské, krajské a celoštátne kolá Zenit v programovaní, medzinárodná súťaž v stavbe a programovaní modelov z lega LEGODACTA, medzinárodná súťaž v programovaní robotov RoboCup). Víťazné družstvo ŠVS BB sa v roku 2003 zúčastnilo MS vo futbale a tanci robotov v Taliansku v Padove. Z tejto súťaže naši zástupcovia doniesli ocenenie medzinárodnej poroty za najlepší program pre robota - tanečníka.

ADRESA: ***ÚIPŠ - ŠKOLSKÉ VÝPOČTOVÉ STREDISKO***
TAJOVSKÉHO 25
975 73 BANSKÁ BYSTRICA
http://www.svsbb.sk

Slovak Data Systems, s.r.o., **Apple VAD**, Tomášikova 30, 821 01 Bratislava

APPLE V ŠKOLE

Slovak Data Systems je spoločnosť, ktorá zastupuje Apple na Slovensku. Sme presvedčení, že Apple nie je potrebné veľmi predstavovať. Počítače Mac sú počítače s najpokročilejším operačným systémom so sadou aplikácií navrhnutých rovnako brilantne ako samotný počítač. Pri spracovaní médií – zvuku, obrazu už v základnej zostave nemajú konkurenciu. A to nehovoríme o dizajne a ekológii. V súčasnosti sú produkované počítače z recyklovateľného skla a hliníka. Nemožno nespomenúť aj ďaľšie výrobky iPod a iPhone, v súčasnosti najpredávanejšie multiprehávače, veľmi populárne nielen medzi mládežou. Spoločnosť Apple je známa tým, že venuje veľkú pozornosť výučbe s podporou počítačov. Skúsenosti s Macom v škole vo vyspelých krajinách sveta sú obrovské. Stačí prejsť stránky www.apple.com/education a nájdete množstvo príkladov využitia Macu pri výučbe.

Prečo je Mac tak zastúpený práve v škole? Domnievam sa, že je to najmä užívateľské prostredie a zameranie Macu na spracovanie médií. Jednoduché užívateľské prostredie umožňuje žiakom i učiteľom nezameriavať sa na to, ako to urobiť, ale sústrediť sa na to čo chcem urobiť.

Aj naša spoločnosť, ktorá zastupuje Apple na Slovensku, začala začiatkom minulého roku spolupracovať s niekoľkými školami. V rámci spoločnosti sa vytvoril priestor na prácu s EDU.

Po začiatočných skúsenostiach sme prišli k tomu, že žiaci (od prvej triedy základnej školy) veľmi rýchlo, po ukázaní ako pracovať s jednotlivými aplikáciami, sú schopní samostatnej práce s Macom. Ukážky prác žiakov nájdete na stránke http://zskarloveska.blogspot.com/, prípadne fotodokumentáciu z projektu na stránke
http://www.zskarloba.roburnet.sk/index.php?option=com_zoom&Itemid=43&catid=189&PageNo=1

Podobné skúsenosti sme získali aj pri práci s rómskymi deťmi v Kremnici. Ukázalo sa, že rómske deti, tak ako deti z Karloveskej nemajú žiaden problém po niekoľkých minútach zácviku samostatne s Macom pracovať. Ukážky budú v krátkej dobe zverejnené na stránkach Súkromné gymnázium, Dolná 48/19, Kremnica.

Pri práci s rómskymi deťmi (a sme presvedčení, že nielen s rómskymi) je mimoriadne potrebná motivácia, aky ich učenie zaujalo. Prichádzame k tomu, že na to je mimoriadne vhodná metóda zážitkového učenia a kontextuálneho učenia. Túto sme využili pri výuke na Základnej škole Beňadická, v Bratislave. Žiaci 8. ročníka boli motivovaní postavením do role člena filmového štábu, ktorý zachytáva výuku. Úlohou bola zaznamenať pre budúce generácie fyziku dneška. Žiaci si pripravili krátke nahrávky, záznamy experimentov (zohrievanie látky, zmena teploty, zmena skupenstva), záznamy z domácností (para z hrncov pri varení, zarosenie okna, zábery zo sauny) z ktorých si pripravili prezentácie v ktorých mali budúcim generáciám (dosť nechápavým) vysvetliť fyzikálne princípy dejov ktoré zaznamenali. Ďalšou motiváciou k prehĺbeniu poznatku bola výzva na spracovanie prezentácie o výrobe filmu. Ukážky práce nájdete v krátkej dobe na stránkach Cirkevnej základnej školy – Narnia, Beňadická 38, Bratislava www.narnia.sk

Sme presvedčení že metóda zážitkového učenia má svoje opodstatnenie aj pri výučbe informatiky. Vzhľadom na zručnosti žiakov s prácou s počítačom, aj v spojení s multimediálnym prehrávačom vrátane mobilného telefónu, je veľký predpoklad výrazného pozdvihu, najmä, ak sa tieto zručnosti využijú.

V budúcnosti chceme pokračovať v rozširovaní zážitkového vyučovania touto formou za pomoci počítačov. Radi privítame záujemcov zo škôl o spoluprácu pri realizácii takéhoto zážitkového učenia.

Vladimír Repáš, EDU consultant, Slovak Data Systems, s.r.o., Apple VAD, Tomášikova 30, 821 01 Bratislava, tel. 0911 250 628, e-mail: repas@apple.sk, www.apple.sk

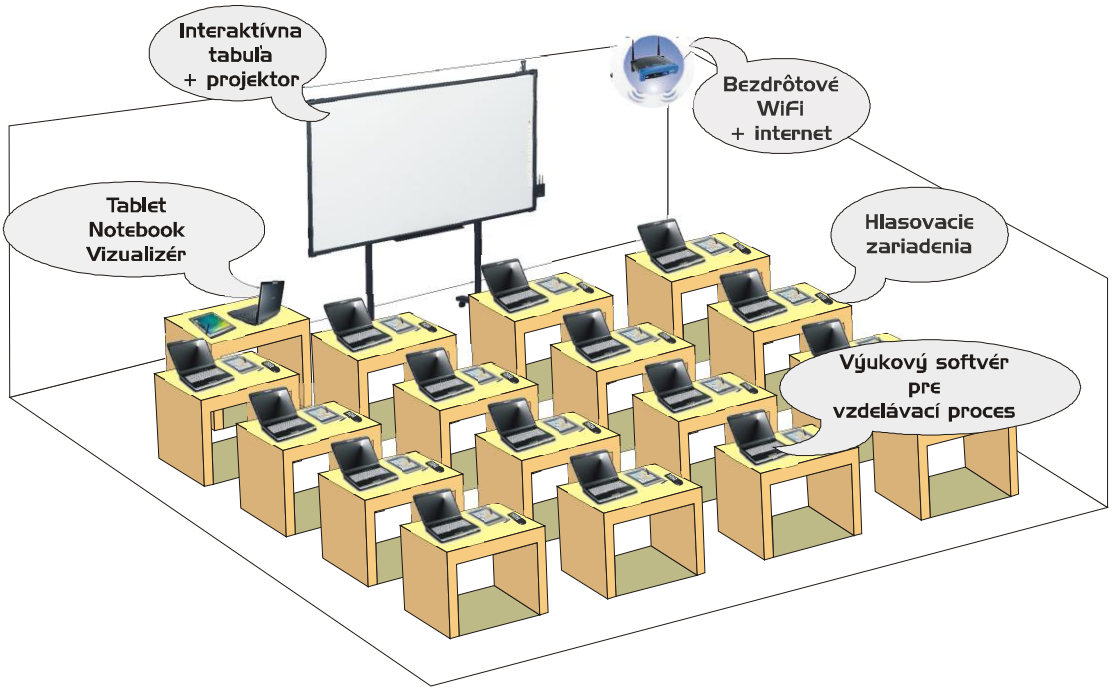
Digitálna škola

Medzi moderné trendy vo vyučovaní nielen informatiky patrí okrem atraktívneho odboru či moderných vyučovacích pomôcok aj zavádzanie inovatívnych modelov vyučovania. Takto sa tradičná škola mení na modernú, digitálnu.

Digitálna škola kladie dôraz na aktivitu žiaka v procese vyučovania. Učitelia používajú také vyučovacie postupy, pri ktorých žiak nie je pasívnym poslucháčom, ale je na vyučovaní aktívny, pracuje, zveruje sa mu samostatná práca a on sám objavuje poznatky.

Moderná škola budúcnosti je založená na využívaní najmodernejšej didaktickej techniky a kvalitných výukových materiálov pre jednotlivé predmety. Tieto si tvoria učitelia sami, prípadne využívajú hotové produkty od profesionálnych tvorcov výukových materiálov. Nevyhnutnou súčasťou vzdelávacieho procesu je získavanie informácií z internetu, či komunikácia na diaľku. Internetové pripojenie predstavuje pre školu v súčasnosti nevyhnutnosť.

Typické vybavenie digitálnej školy je charakterizované nasledovným obrázkom.



Okrem samotného vybavenia učebne didaktickou a prezentačnou technikou je rovnako potrebné zabezpečiť aj nástroje pre plnohodnotné využívanie takejto učebne. Nemal by chýbať širokopásmový prístup do internetu, zabezpečenie proti vírusom a iným ohrozeniam, či centrálny systém, kde budú uložené edukačné produkty využívané počas vyučovania. Tieto by mali byť dostatočne zabezpečené najmä voči neautorizovanému prístupu.

GAMO a.s. je pripravené poskytnúť školám komplexnú podporu pri zriaďovaní digitálnych učební. Okrem dodávky techniky, jej inštalácie, až po vybudovanie komplexného IT riešenia pre plnohodnotné využívanie tejto techniky.

Taktiež sme schopní v spolupráci so školskými zariadeniami pripraviť **kvalitné projekty**, vďaka ktorým budú môcť školy modernizovať svoje materiálno-technické vybavenie s využitím financovania z prostriedkov EU. Výzva operačného programu Vzdelávanie „**Premena tradičnej školy na modernú“** ponúka školám možnosť získať nenávratný finančný príspevok až do výšky 333.000 € (10 miliónov korún), z čoho až 20 % môže ísť na nákup vybavenia a zariadenia. Je to dobrá príležitosť pre školy ako zvýšiť svoju konkurencieschopnosť a atraktívnosť pre žiakov.

GAMO a.s. I Kyjevské námestie 6 I 974 04 Banská Bystrica
Tel: 048/4372 111 I Fax: 048/4132 947
info@gamo.sk I www.gamo.sk I online.gamo.sk
Banská Bystrica I Bratislava I Žilina – Liptovský Mikuláš I Košice

GAMO INFORMAČNÉ TECHNOLÓGIE

SUN MICROSYSTEMS PODÁVA ŠKOLÁM POMOCNÚ RUKU

Desktopové počítačové pracoviská postavené na sieťovej technológii Sun Ray si vďaka svojej spoľahlivosti, jednoduchej údržbe a nízkym prevádzkovým nákladom získavajú čoraz väčšiu popularitu. Akademická obec ich v rámci programu Matching Grant môže mať za polovičnú cenu.

JEDEN PLUS JEDEN

Sun Microsystems celosvetovo organizuje podporný akademický program Matching Grant, ktorý už niekoľko rokov úspešne prebieha i na Slovensku. Jeho princípom je, že ak si škola zakúpi z portfólia firmy určité množstvo výrobkov, rovnaký počet dostane navyše.

TECHNOLÓGIA SUN RAY

Program doteraz oslovil tri slovenské vysoké školy, resp. fakulty. Všetky sa rozhodli využiť sieťové zariadenia Sun Ray. Prostredníctvom týchto zariadení sa operácie na počítači nespracovávajú lokálne, ale centralizovane, na servery. Používatelia tak na to, aby mohli vykonávať bežné úkony, ako je surfovanie po internete, posielanie mailov alebo písanie textov, nepotrebujú plnohodnotné PC s procesorom, pamäťou a pevným diskom, ale vystačia si len s monitorom, klávesnicou a myšou. Vzhľadom na to, že operačný systém, ako aj aplikácie bežia na servery, desktopy si nevyžadujú žiadnu administráciu či upgrady, všetko stačí urobiť raz na servery. V konečnom dôsledku to vedie k výrazným energetickým úsporám, menším nárokom na priestor a zníženiu prevádzkových nákladov. Jeden príklad za všetky – kým bežné PC má priemernú spotrebu od 100 do 150 wattov, Sun Ray si vystačí so štyrmi wattmi.

TRI VYSOKÉ ŠKOLY

Ako prvá na Slovensku program Matching Grant využila Fakulta chemicko-potravinárskych technológií Slovenskej technickej univerzity v Bratislave v roku 2001, ktorá týmto spôsobom zabezpečila pre svoju fakultnú knižnicu 100 zariadení Sun Ray. Slovenská poľnohospodárska univerzita v Nitre si cez Matching Grant zabezpečila 30 zariadení Sun Ray, ktoré jej slúžia ako internetové kiosky. Okrem internetu ich poslucháči využívajú aj na prístup do študentského informačného systému, pomocou ktorého sa môžu napríklad prihlásiť na skúšku. Zatiaľ najväčším tuzemským „klientom“ Matching Grantu je Materiálovo-technologická fakulta Slovenskej technickej univerzity v Trnave, kde v súčasnosti prebieha implementácia 120 zariadení Sun Ray. „Ide skutočne o veľký projekt, pretože fakulta chce do konca tohto roka rozmiestniť po škole dokopy 500 Sun Rayov,“ tvrdí Tomáš Vencelík zo Sun Microsystems.

TRISTOTISÍC DOLÁROV

Sun Microsystems vypisuje Matching Grant celosvetovo dva razy do roka, vždy v druhom a štvrtom kvartály, kedy záujemcovia môžu podávať prihlášky. Okrem vysokých škôl sa do neho môžu zapojiť aj stredné a základné školy a ostatné vzdelávacie inštitúcie. „V prípade záujmu stačí kontaktovať našu spoločnosť,“ podotýka T. Vencelík s tým, že firma na Slovensku za posledné tri roky bezplatne poskytla akademickej obci technológie v hodnote 300 000 dolárov.

INTERNET V KNIŽNICIACH

Technológia Sun Ray nenachádza odozvu len na vysokých školách, ale využívajú ju tiež iné inštitúcie, napríklad Univerzitná knižnica v Bratislave. Okrem toho Sun Microsystems minulý rok ukončil rozsiahly projekt financovaný zo štrukturálnych fondov Európskej únie, v rámci ktorého tieto zariadenia nainštaloval v 110 mestských a obecných knižniciach rozmiestnených po celom Slovensku, kde slúžia ako verejné prístupové body k internetu.

Viac infromácií tu:
http://sk.sun.com/matchinggrant/

Názov:	**DidInfo 2009**
Editor:	prof. RNDr. Branislav Rovan, PhD.
Technický editor:	PaedDr. Ladislav Huraj, PhD.
Vydanie:	1. vydanie
Vydavateľ:	Univerzita Mateja Bela, Fakulta prírodných vied v Banskej Bystrici
Rok:	2009
Rozsah:	192 strán
Formát:	B5
Náklad:	100 ks
Tlač:	Bratia Sabovci, Banská Bystrica
ISBN:	978-80-8083-720-4
EAN:	9788080837204

Title:	**Proceedings of conference DidInfo 2009**
Editor:	prof. RNDr. Branislav Rovan, PhD.
Technical Editor:	Dr. Ladislav Huraj, PhD.
Edition:	First edition
Publisher:	University of Matej Bel, Faculty of Natural Science in Banská Bystrica, Slovakia
Year:	2009
Pages:	192
Print:	Bratia Sabovci, Banská Bystrica, Slovakia
ISBN:	978-80-8083-720-4
EAN:	9788080837204