



**Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky**



Ing. Mgr. Roman Horváth

Autoreferát dizertačnej práce

Hľadanie nových postupov vo výučbe programovania na VŠ

na získanie akademického titulu philosophiae doctor

v odbore doktorandského štúdia:

9.2.3. teória vyučovania informatiky

Bratislava 2013

Úvod

V dizertačnej práci sa zaoberáme problematikou vyučovania programovania na vysokej škole, konkrétne úvodom do programovania pre odbor učiteľstvo akademických predmetov študijných programov v kombinácii s informatikou. Ide teda o potenciálnych budúcich učiteľov informatiky na ZŠ alebo SŠ.

Programovanie je podľa ŠVP súčasťou predmetu informatická výchova na prvom stupni ZŠ a informatika na druhom stupni ZŠ a na SŠ. Dôvodom vyučovania programovania na ZŠ/SŠ nie je plošná výchova excelentných programátorov. V primárnom a sekundárnom vzdelávaní ide skôr o riešenie problémov. Programovanie rozvíja schopnosť analyzovať problém, rozložiť ho na čiastkové problémy, získať čiastkové riešenia problému a následne syntetizovať čiastkové výsledky do výsledného riešenia. Umožňuje v rámci riešenia zadaných úloh skúmať možné algoritmické riešenia, premýšľať o spôsobe zápisu algoritmu, porozumieť formálnemu zápisu algoritmu a získať schopnosť interpretácie výsledného riešenia, ktoré spravidla býva sprostredkované formálnym zápisom. Umožňuje konfrontovať formálny zápis s prirodzeným jazykom. Budúci učiteľ informatiky/informatickej výchovy by si mal tieto myšlienky osvojiť.

Pri vyučovaní programovania nás nezaujíma len sám cieľ, teda vyriešenie problému, ale aj postupy hľadania riešenia problému. Dalo by sa povedať, že algoritmizácia rozvíja schopnosť hľadania riešení problémov podobným spôsobom ako tvorivé slovné úlohy v matematike či fyzike. Zároveň ten, kto nadobudne schopnosť tvoriť (hoci aj jednoduché) algoritmy, lepšie porozumie strojovej logike technických prostriedkov, ktoré nás obklopujú, pretože komunikácia so zariadeniami sa od komunikácie s ľuďmi či zvieratami líši a v našej spoločnosti je čoraz rozšírenejšia. Riadi sa istými formálnymi pravidlami. Používame ju pri komunikácii, resp. ovládaní, digitálnych prostriedkov, ako sú mobilné telefóny, bankomaty, počítače, informačné systémy a iné technické zariadenia.

Problematika programovania je pre veľkú časť prijatých študentov úplne nová. Začínajú od začiatku a majú často problém zorientovať sa v nej. Treba podotknúť, že zvládnuť problematiku algoritmizácie, hoci aj na jednoduchej úrovni, nie je triviálne. Nielen naše skúsenosti potvrdzujú, že študenti s tým majú problém a usilujú sa svoju situáciu riešiť rozmanito (napríklad [4], [18], [24], [30], [46], [48] alebo [81] a tak ďalej). Napríklad učením sa hotových programov naspamäť. Toto zistenie bolo pre nás spočiatku šokujúce a mali sme záujem začať plánovať kroky na systematickú elimináciu výskytu tohto javu. Dnes sme názor na tento jav zmenili. Zdá sa, že tento individuálny prístup niektorých študentov nemusí byť slepou cestou k naučeniu sa programovania. Ak je študent schopný a ochotný nezastaviť sa v tejto fáze svojho osobného napredovania, môže to byť pre neho prvý krok. Do veľkej miery to záleží na samotnom študentovi. Učiteľ môže poskytnúť oporu formou prístupu a vhodných softvérových prostriedkov. Ak máme chápať programovanie ako zručnosť, mali by sme pripustiť, že pri získavaní zručnosti môže vzniknúť aj fáza učenia sa naspamäť, fáza mechanického opakovania predvedenej aktivity, ale študent musí byť ochotný ísť za hranice tejto fázy. Cieľom je získanie schopnosti samostatnej tvorivej činnosti. Problém učenia sa programov naspamäť a podobný spôsob (aspoň čiastočného) akceptovania tohto javu spomínajú aj Gaspar a Langevinová v [18]. Odvolávajú sa pritom na Bloomovu taxonómiu vzdelávacích cieľov.

Počítač je technickým prostriedkom umožňujúcim urýchlenie a zjednodušenie niektorých činností, no len pri správnom použití. Rovnako účinne dokáže prácu človeka skomplikovať alebo značne predĺžiť čas potrebný na jej dokončenie. Táto dvojité povaha sa prejavuje všade – to znamená, že aj pri vyučovaní programovania, preto je potrebné vyvíjať dostatočné úsilie na to, aby sa naša práca uberala želaným smerom. V tom spočíva význam didaktiky informatiky, vlastne aj didaktiky ako takej – aplikovať správne didaktické postupy by malo prinášať zvýšenie efektívnosti práce učiteľa.

Projekt dizertačnej práce

V nasledujúcich sekciách opisujeme výskumný problém s oblasťou výskumu, ciele projektu, z ktorých sú odvodené výskumné otázky, navrhovanú metodiku výskumu, s čím súvisia použité metódy a zabezpečenie kvality výskumu. Keďže ide o výskum dotýkajúci sa ďalších osôb, zaoberáme sa aj otázkou etiky výskumu.

Výskumný problém a oblasť výskumu

Študenti vysokých škôl, ktorí sú prijímaní na študijné programy učiteľstva informatiky, väčšinou nemajú predchádzajúce skúsenosti s programovaním. Táto komplexná oblasť je pre nich úplne nová, preto majú problém zorientovať sa v nej. Stará koncepcia vyučovania úvodu do programovania nebola vyhovujúca. Bola zostavená pre prostredie s inou charakteristickou vzorkou študentov. Situácia si vyžadovala lepšie zhodnotiť to, s akou vzorkou študentov pracujeme na našej fakulte, a prispôbiť sa danému stavu. Z toho dôvodu sme sa začali zaoberať návrhom novej koncepcie vyučovania základov programovania a následným overením jej vhodnosti.

Výskumný problém: navrhnuť, implementovať a overiť obsah, sylabus a metodológiu výučby pre úvodný kurz programovania.

Výskumná oblasť: vyučovanie základov programovania v prvom ročníku vysokej školy (učiteľský smer).

Ciele výskumu

Cieľom výskumného projektu bolo navrhnuť, implementovať a overiť koncepciu vyučovania úvodného kurzu programovania, ktorý je vyučovaný v prvom ročníku bakalárskeho štúdia učiteľstva akademických predmetov študijných programov v kombinácii s informatikou.

Cieľom dvoch semestrov úvodného kurzu programovania je uviesť študentov do problematiky programovania, poskytnúť im základné poznatky z oblasti programovania a algoritmizácie a naučiť ich vytvárať základné algoritmy. V našom projekte sa zameriavame na prvý semester úvodného kurzu.

Výskumné otázky

Pri plnení cieľov výskumného projektu budeme sledovať nasledujúce otázky a podotázky:

1. Ako postupovať pri stanovovaní a plnení cieľov výučby úvodného kurzu programovania?
 - Aké sú ciele výučby úvodného kurzu programovania?
 - Ako vytvoriť podrobný sylabus úvodného kurzu programovania, aby boli splnené ciele výučby?
 - Čo má byť obsahom podrobného sylabu na to, aby boli študenti schopní splniť výstupné požiadavky úvodného kurzu programovania?
2. Ako stanoviť výstupné požiadavky na študentov úvodného kurzu programovania?
 - Ako zvoliť podmienky priebežného a záverečného hodnotenia?
3. Ako hodnotiť úspešnosť úvodného kurzu programovania?
 - Akým spôsobom vyhodnotiť znalosti študentov po aplikovaní navrhnutých postupov do vyučovania?

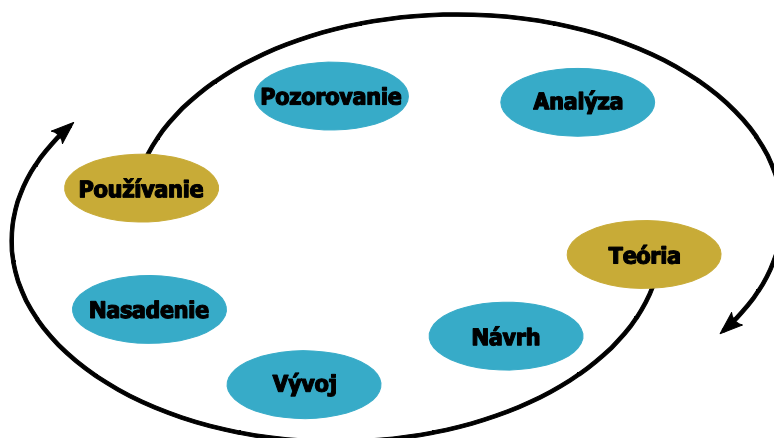
- Ktoré prvky navrhnutej koncepcie boli pre študentov problematické? Je možné nájsť spôsob redukcie týchto problémov? Ktoré prvky koncepcie mali pozitívny dosah a je možné ich posilňovať?

Je možné, že nie na všetky podotázky budeme schopní uspokojivo odpovedať.

Metodológia výskumu a metodologická preferencia

Pre náš výskum pokladáme za vhodnú kvalitatívnu metodológiu výskumu. Dôvodom výberu tejto stratégie je povaha našej práce, ktorej cieľom je návrh novej koncepcie vyučovania predmetov úvodného kurzu programovania. Optimálnym riešením sa ukazuje byť využitie kvalitatívneho prístupu cez **výskum vývojom** [41] (Design-Based Research).

Pri takto realizovanom výskume má výskumník zároveň úlohu vývojára. Prvej fáze vývoja predchádza prvotná analýza a návrh riešení, ďalej sa výskum realizuje v niekoľkých iteráciách tvorených nasledujúcimi charakteristickými a vzájomne sa prelínajúcimi krokmi: vývoj, nasadenie (počas ktorého produkt používame a realizujeme pozorovanie) a analýza, ktorá cez návrh zmien spätne vedie k vývoju v ďalšej iterácii. Každá ďalšia fáza vývoja sa odvíja od prechádzajúcej. Schéma iterácií je zobrazená na obrázku 1.



Obrázok 1: Schéma iterácií vo výskume vývojom [41]

Výskumné metódy a výskumná vzorka

Počas realizácie projektu sme použili niekoľko **metód** zberu a analýzy údajov [19], [20], [62], [74]:

- *Participačné pozorovanie* – výskumník zároveň zastáva úlohu vyučujúceho na hodinách programovania, ktorých sa dotýka tento projekt. Pozorovanie sme používali vo všetkých fázach projektu, a to v úvodnej fáze výskumu (orientačnej etape) aj v neskorších etapách implementácie. Na zber údajov slúžili terénne zápisky reflektujúce na priebeh a stav na vyučovacích hodinách. Počiatočné zápisky boli zapisované na papier, neskôr boli prepísané do elektronickej podoby.
- *Pološtruktúrované rozhovory* – v úvodnej fáze projektu bola táto metóda použitá na získanie časti informácií o aktuálnej situácii vo vyučovaní programovania na vybraných vysokých školách na Slovensku a hodnotenie vstupnej verzie úvodného kurzu programovania na fakulte, kde je kurz implementovaný. Získavané údaje boli priebežne zaznamenávané na papier, finálna verzia bola prepísaná do elektronickej podoby.
- *Neštruktúrované rozhovory* – v prvých etapách výskumu prebiehali vo vyhradených časoch poloformálne rozhovory so študentmi, počas ktorých bol priebežne zisťovaný aktuálny stav (atmosféra, problémy a pod.) vo výskumnej vzorke študentov. Informácie boli priebežne zaznamenávané na papier

a do poznámok v mobilnom zariadení. Výsledná verzia bola zapracovaná do jestvujúcich údajov uložených v počítači.

- *Obsahová analýza produktov* – počas každého semestra priebežne vznikali produkty študentov odovzdávané v rámci náplne predmetu. Išlo o programy vypracované v rámci domácich úloh, priebežné písomné práce písané na začiatku niektorých cvičení a písomné práce a programy písané v rámci udeľovania záverečného hodnotenia. Každý študentský produkt znamenal bodový zisk a musel byť preto hodnotený. Počas procesu hodnotenia sme vykonávali priebežnú analýzu. Na jej zistenia sme čiastočne reflektovali priebežne – v rámci aktuálneho semestra. Nakoniec sme výsledky analýzy zapracovali do novej verzie koncepcie vyučovania.
- *Anonymné eseje* – v závere každého semestra bola študentom zadaná úloha spísania anonymnej eseje. V nej mali za úlohu sebarefektívne a kriticky zhodnotiť priebeh semestra. Pomôckou na napísanie eseje bola krátka osnova, ktorá mala slúžiť ako návod na jej obsahovú náplň. Odovzdané eseje sme podrobili rekurzívnemu čítaniu a zo zistení sme vyvodili závery, ktoré sme zapracovali do novej koncepcie vyučovania.
- *Štruktúrované úvodné dotazníky a vstupné testy* – táto metóda slúžila na získavanie údajov o nastupujúcich študentoch. Testy a dotazníky boli zadávané študentom v prvom semestri bakalárskeho štúdia priamo na prvých cvičeniach predmetov úvodu do programovania v elektronickej forme (pre opakujúcich študentov bolo ich vyplňanie nepovinné). Ich spracovanie sa uskutočňovalo takisto elektronicke – analýzou, triedením a kvantifikáciou relevantných údajov v dotazníku a hodnotením miery správnosti riešení v teste.

Výskumnú vzorku tvorili všetci študenti prvého ročníka bakalárskeho štúdia, ktorí navštevovali vyučovanie predmetov úvodu do programovania na PdF TU (s prípadným malým percentom opakujúcich študentov).

Zabezpečenie kvality výskumu

Kvalitu výskumu sme zabezpečovali *trianguláciou* a *auditom výskumu*. [62], [74]

Triangulácia spočívala v rôznych metódach zberu údajov (participačné pozorovanie – dianie na hodinách, reakcie študentov, ich osobný rast a podobne; rozhovory so študentmi o konkrétnych problémoch, o ich návrhoch a pod.; obsahová analýza produktov – odovzdaných domácich úloh a písomných prác; anonymné eseje, štruktúrované úvodné dotazníky a vstupné testy).

Audit výskumu vyplýva z povahy práce – správa o výskume je súčasťou tejto dizertačnej práce a čiastočné výsledky boli publikované v príspevkoch na konferenciách a v zborníkoch. Súčasťou auditu boli aj diskusie o priebežnom postupe s kolegami na pracovisku alebo na KZVI FMFI UK.

Etické otázky výskumu

Povinnosťou každého výskumníka je správať sa počas výskumu eticky. To sa dotýka najmä ochrany súkromia skúmaných osôb. V našom výskume bola anonymita osôb zabezpečená buď priamo použitou metódou (anonymná eseje), alebo prijatím konkrétnych opatrení aplikovaných v metóde (priradením kódu každému účastníkovi, ktorého význam bol známy iba výskumníkovi). V prípade rozhovorov bol od účastníkov žiadaný ústny súhlas s rozhovorom. Zaviazali sme sa, že nikde v rámci práce ani iných výstupov nebude uvedené meno žiadneho účastníka (s výnimkou, keď by o to účastník sám požiadal). [19], [20]

Výsledky výskumu

V rámci realizácie výskumného projektu boli vykonané štyri iterácie. Vo štvrtej iterácii (2012) sme sa rozhodli radikálnejšie obmedziť rozsah tém v kurze programovania počas úplne prvého semestra, teda na predmete programovanie a algoritmy 1. K tomuto rozhodnutiu nás nasmerovala jednak spätná väzba od študentov, získavaná počas prechádzajúcich iterácií vrátane ich výsledkov, a jednak vlastné pozorovania, terénne poznámky a ďalšie údaje zbierané počas týchto iterácií. Vo vyjadreniach študentov sa opakovane vyskytovali konkrétne formulácie, ktorými sa usilovali vysvetliť, z akých dôvodov pokladajú úvodný kurz za náročný, alebo čo podľa nich viedlo k tomu, že prestali zvládať priebežné kognitívne spracúvanie obsahu semestra. V prvých iteráciách sa často opakovali vyjadrenia typu „*nechytli sme začiatok*“ a „*máme málo času*“. Vyjadrenia tohto charakteru sme počas týchto iterácií považovali za kľúčové v súvislosti s vysvetlením príčin neúspechu študentov.

Najnovšia verzia kurzu bola nasmerovaná tak, aby prvý semester príliš nepresahoval hranice nadobúdania základných zručností programovania. Zmena nebola vykonaná skokovo. Pred začatím každej iterácie projektu sme zvažovali vhodnosť a rozsah zmien zameraných týmto smerom, pretože z našich skúseností a analýz údajov opakovane vyplývalo, že pre našu vzorku študentov bude potrebné vykonať ďalšie zmeny. Výsledkom všetkých zmien je výraznejšie zameranie sa prvej časti kurzu na opakovanie a upevňovanie základov.

Údaje sme počas výskumu zbierali z viacerých zdrojov. Analyzovali sme priebežné písomné práce, domáce úlohy, viedli sme osobné formálne aj neformálne rozhovory so študentmi, zadávali sme anonymné eseje a podobne. V poslednej iterácii sme po skončení semestra študentom predložili doplnkový dotazník, ktorého cieľom bolo spresnenie niektorých informácií získaných počas semestra.

Zadania záverečných anonymných esejí mali pološtruktúrovaný charakter. Študenti boli vyzvaní, aby napísali čokoľvek, čo súvisí s predmetmi programovania, resp. vo vzťahu k nim. Mali k dispozícii množinu otázok, ku ktorým im bolo vysvetlené, že im majú iba pomôcť pri kompozícii obsahu eseje, v žiadnom prípade nie sú záväzné. Odovzdané eseje boli tematicky zamerané podobne ako počas predchádzajúcich rokov. Informácie, ktoré sme získali, nasvedčujú zlepšeniu viacerých skúmaných faktorov.

Napríklad v esejach z poslednej iterácie sa študenti v menšej miere sťažovali na nízku časovú dotáciu úvodného kurzu programovania – negatívny názor vyjadrilo asi 6 % respondentov. V predchádzajúcich rokoch to bolo približne 12 % respondentov. Podľa nášho názoru k tomu prispelo otvorenie kurzu propedeutiky programovania. Počet kontaktných hodín predmetov súvisiacich s úvodom do programovania je podľa nás v súčasnosti dostatočný – týždenne dve kontaktné hodiny prednášok, dve kontaktné hodiny cvičení z predmetov programovanie a algoritmy 1, 2 a dve kontaktné hodiny cvičení z predmetu propedeutika programovania 1, 2. Stále sa však pokladáme za dôležité, aby študenti venovali dostatok času priebežnej aktivite počas semestra, preto aj v budúcnosti plánujeme udržiavať pri predmetoch programovania dostatočnú úroveň objemu domácej práce študentov.

Značná časť študentov sa s nami v esejach zadaných počas iterácií zhodovala v názoroch na dôležitosť pravidelných príprav a poctivej domácej práce. V poslednej iterácii sa k tejto téme vyjadrilo asi 25 % respondentov, v predchádzajúcich iteráciách sa ich percento pohybovalo v rozmedzí 20 – 45 %. Viacerí sebakriticky zhodnotili, že keby sa doma pravidelne a lepšie pripravovali, úroveň ich vedomostí by bola vyššia. Často sme sa stretávali s opisom priebehu ich domácej práce a v súvislosti s tým s vyjadrovaním frustrácie pri neúspešných pokusoch o riešenie domácich úloh a naopak, s vyjadrením zvýšenia motivácie pri dosiahnutí úspechu. Študenti sa síce vyjadrovali, že by boli radšej, keby nebolo nutné sa v rámci

predmetu toľko venovať domácej práci a písaniu priebežných písomných prác, no napriek tomu chápu ich význam a preto ich prítomnosť v kurze rešpektujú.

V záverečných anonymných esejach študenti uvádzali aj svoje názory na obsah úvodného kurzu. V poslednej iterácii sa niektorí skúsenejší študenti vyjadrili, že im kurz „nič nové nedal“ s pripomienkami typu: „aspoň sme si zopakovali, čo sme vedeli“ alebo „drvivá časť mojich spolužiakov nepatrí medzi skúsenejších, pre nich to bolo dobré“. Vyjadrenia začiatočníkov sa zmenili oproti tým, ktoré sme získavali v predchádzajúcich iteráciách. Väčšina študentov o tomto semestri tvrdila, že bol „primerane náročný a keď/keby vynaložili dostatok úsilia, nemali by zlyhať.“ Vyskytlo sa percento účastníkov (približne 29 %), ktorí opäť tvrdili, že „sa v témach strácali.“ Úplnú elimináciu takýchto jednotlivcov by sme pravdepodobne dosiahli jedine prísnejšími kritériami prijímacieho konania. Takýto krok za súčasných okolností nie je možný.

Okrem prvej iterácie bolo súčasťou úvodných kurzov písanie priebežných písomných prác zadávaných na začiatku niektorých cvičení. V odpovediach študentov dochádzalo relatívne často k výskytu javu súvisiaceho s tým, že študenti majú tendenciu vytvárať si a udržiavať nesprávne chápanie niektorých konceptov. Na minimalizáciu tohto javu je dôležité pravidelné poskytovanie spätnej väzby. Na našich hodinách to riešime neustálym opakovaním problematických pasáží, ktoré sme schopní identifikovať prostredníctvom priebežných písomných prác. Po každej písomnej práci so študentmi diskutujeme o správnych odpovediach a riešení úloh. Napriek neustálemu opakovaniu sa niektoré nesprávne porozumenia často nedarí odstrániť ani po skončení semestra, pretože sa opäť vyskytujú v teoretických častiach písomných prác záverečného hodnotenia.

Tento opakujúci sa jav v súčinnosti s opakujúcou sa informáciou z esejí o „nedostatku študijných materiálov v slovenskom jazyku“ nás motivuje k spracovaniu nového elektronického interaktívneho vzdelávacieho materiálu na podporu vyučovania programovania. Stručná charakteristika je uvedená v závere práce (na strane 12). V iteráciách pred rokom 2012 sme sa opakovane stretli so sťažnosťami študentov súvisiacimi s malým objemom vhodných študijných materiálov k predmetom úvodu do programovania (problém formulovalo v priemere 25 % respondentov). V esejach z roku 2012 sa vyskytlo menej ohlasov s charakterom požiadaviek na poskytnutie väčšieho množstva študijných materiálov, ktoré by boli *dostatočne zrozumiteľné* (to znamená, že nielen v slovenskom jazyku, ale aj napísané primerane zrozumiteľným štýlom) a v doplnkovom dotazníku až 94 % opýtaných študentov vyjadrilo názor, že k predmetu bol poskytnutý dostatok študijných materiálov (objemovo), ale nie všetkým študentom poskytované materiály vyhovovali (kvalitatívne). V esejach a v otázke s voľnou odpoveďou v doplnkovom dotazníku sa viacero študentov vyjadrilo, že by v tomto smere očakávali zlepšenie (v esejach asi 15 % respondentov a v dotazníkoch asi 25 % respondentov). Pre nový materiál je preto naplánované iné kvalitatívne vyhotovenie (okrem iného má byť interaktívny) a je plánovaný pre širšiu cieľovú skupinu študentov, preto by mohol nahradiť alebo doplniť aj študijné materiály pre iné predmety programovania.

O skupine tried grafického robota sa študenti vyjadrovali kladne. Väčšina názorov bola súhlasných a pozitívnych. Keďže nejestvuje žiadna oblasť ľudskej aktivity, v ktorej by bolo možné vyhovieť 100 % účastníkov, aj pri hodnotení grafického robota sa vyskytlo niekoľko menej pozitívnych názorov a pripomienok, ktoré sa dotýkali napríklad inej predstavy o požiadavkách na úvodný kurz programovania – tie mali byť (podľa názoru študentov) vyššie. Autormi takýchto výrokov boli evidentne programátorsky zdatnejší študenti, ktorí od úvodného kurzu programovania (s ohľadom na to, že ide o vysokoškolský kurz) pravdepodobne očakávali výraznejšie zdokonaľovanie sa vo svojich zručnostiach.

Ako postupovať pri stanovovaní a plnení cieľov výučby úvodného kurzu programovania?

Najvyšším cieľom kompletnej série kurzov programovania zahŕňajúcej úvodné kurzy je nadobudnutie úrovne samostatnej tvorivej činnosti študentmi. Tento proces vyžaduje od študentov vynaloženie množstva namáhavej práce pri učení sa a tréningu. Úvodný kurz programovania by mal študentov navigovať na túto cestu. Ciele úvodného kurzu programovania by mali byť stanovené s ohľadom na to, že výsledkom vzdelávania úvodného kurzu nie je produkcia „skvelých programátorov“. Cieľom úvodného kurzu by malo byť vybudovanie pevných základov programovania, nevyhnutných pre ďalšie napredovanie. Jednou z možností, ako dosahovať uvedené ciele, je napríklad popri neustálom opakovaní najdôležitejších konceptov sledovať nesprávne formované poznatky a usilovať sa tieto nedostatky odstraňovať.

S ohľadom na uvedené by stanovenie cieľových požiadaviek na vedomosti študentov po absolvovaní kompletného úvodného kurzu malo zahŕňať: *základné koncepty a základnú terminológiu* z oblasti programovania a algoritmov a znalosť *základných algoritmov*. Náš výskum mal zároveň odpovedať na otázky dotýkajúce sa vyučovania vzhľadom na typickú vzorku študentov nastupujúcich na našu fakultu. Konštatujeme, že v našom prostredí úsilie o vyučovanie konceptov presahujúcich základné oblasti v rámci úvodného kurzu programovania nebolo úspešné. Pre študentov bolo často problematické i zvládnutie základných konceptov. Skúsenosti získané pozorovaním, analýzami priebežných prác spolu s údajmi získanými anonymnou spätnou väzbou od študentov boli príčinou zúženia objemu vyučovaných tém a posilnenie sústredenia prvej časti úvodného kurzu na posilňovanie a upevňovanie základov a získanie základnej programátorskej zručnosti. Ďalšie pozorovania na predmetoch programovania pre pokročilých naznačujú, že tento prístup má pozitívny vplyv na neskoršie zvládanie zložitejších konceptov z oblasti programovania. Pozorovania vykonávané na predmetoch programovania pre pokročilých nie sú súčasťou tejto práce, ale pomôžu odpovedať na otázky súvisiace s hodnotením úspešnosti úvodného kurzu v budúcnosti.

Ako stanoviť výstupné požiadavky na študentov úvodného kurzu programovania?

Východiskami pre stanovovanie výstupných požiadaviek úvodného kurzu programovania sú ciele úvodného kurzu a vyššie spomenuté okolnosti vedúce k ich formulácii.

To znamená, že výstupné požiadavky by mali zahŕňať:

- nasledujúcu *základnú terminológiu*: údajový typ, premenná, konštanta, literál, znak, reťazec atď.;
- nasledujúce *základné koncepty programovania*: sekvencia, podmienené spracovanie, cyklus, vnorené cykly, jednoduché a viacnásobné vetvenie, metódy, volanie metód, rekurzia atď.;
- nasledujúce *základné algoritmy*: lineárne a binárne vyhľadávanie, základné triediace algoritmy, backtracking, prípadne iné rekurzívne algoritmy.

Z uvedených bodov vyplývajú typy úloh, ktoré by študenti mali zvládať po absolvovaní úvodného kurzu programovania. Okrem úloh priamo súvisiacich so základnými zručnosťami (použitie cyklov, vetvení a podobne, čo je v našej koncepcii súčasťou prvej časti úvodného kurzu, ktorú sme sledovali), sú to základné algoritmy (ktoré sme v našej koncepcii presunuli do druhej časti úvodného kurzu).

Ako zvoliť podmienky priebežného a záverečného hodnotenia?

Podmienky priebežného a záverečného hodnotenia by mali byť stanovené tak, aby študenti boli vedení k vyvíjaniu dostatočného objemu priebežnej aktivity počas semestra. Tento predmet nie je zvládnuteľný bez priebežných príprav a dostatočnej miery praktického tréningu. Záverečné hodnotenie by malo byť zostavené tak, aby overilo mieru zvládnutia tých poznatkov, ktoré boli počas úvodného kurzu preberané.

Ponechanie väčšieho priestoru na samostatné štúdium (pokročilých) konceptov odporúčame realizovať až v kurzoch programovania pre pokročilých študentov.

V *priebežnom hodnotení* sú uvedené tri povinné zložky s rôzne vyváženým percentuálnym vplyvom na *celkové hodnotenie*: *účasť na cvičeniach* (15 %), *domáce úlohy* (15 %), *priebežné písomné práce* (30 %).

Spomínali sme, že *domácimi úlohami* chceme dosiahnuť zvýšenie miery domácej práce študentov, ktorú náležite oceníme. Preto sú domáce úlohy v nových verziách kurzu povinné a hodnotené. Otázka kopírovania domácich úloh je riešená zápornými trestnými bodmi.

Priebežné písomné práce majú plniť úlohu priebežnej kontroly, spätnej väzby a kladú si za cieľ aj vyvolanie potreby zvýšenia objemu času zameraného na pravidelné prípravy zo strany študentov.

Účasť na cvičeniach je povinná a bodovo hodnotená, pretože práve na cvičeniach venujeme množstvo času spoločnému riešeniu rôznych problémov a dávame priestor aj na samostatnú prácu študentov, pri ktorej môžu žiadať pomoc spolužiakov alebo vyučujúceho.

Záverečné hodnotenie je realizované formou skúšky pozostávajúcej z teoretických otázok a praktických úloh. Je to relatívne štandardný spôsob hodnotenia. Otázky a úlohy sú zadávané tak, aby čo najlepšie pokryli učivo preberané počas semestra.

Celkové hodnotenie je tvorené súčtom bodov získaných v priebežnom a záverečnom hodnotení. Študent musí dosiahnuť hranicu 50 bodov zo stanoveného maxima sto bodov.

Ako hodnotiť úspešnosť úvodného kurzu programovania?

Spôsob posudzovania úspešnosti úvodného kurzu nespočíva len posúdení počtu úspešných absolventov kurzu, ale aj v tom, ako sa prejaví vedomosti absolventov počas ďalšieho štúdia – na kurzoch, ktoré nadväzujú na úvodný kurz programovania a pri závere štúdia, keď majú študenti možnosť aplikovať získané vedomosti pri vypracovaní aplikačne/prakticky orientovaných záverečných/bakalárskych prác.

Počas minulých rokov preukazovali študenti (vrátane tých študentov prvých iterácií nášho výskumu, ktorí sa v čase ukončenia výskumu dopracovali do záverečnej fázy bakalárskeho štúdia) nízky záujem o aplikačne orientované bakalárske práce. Rovnako vedomosti študentov na pokračujúcich kurzoch programovania boli práve také, aby sme mohli zhodnotiť, že aktívne vzdelávanie, upevňovanie základov a neustále opakovanie vedomostí musí pokračovať aj počas týchto kurzov programovania. Pozvoľna sa však tento stav mení a čoraz väčšie percento študentov je schopné lepšie aplikovať svoje poznatky z programovania vo vyšších ročníkoch.

Takto stanovený spôsob hodnotenia úspešnosti úvodného kurzu programovania vyžaduje sledovanie vývoja vo väčšom časovom rámci. Výsledky posledných iterácií sa ukážu počas najbližších rokov. V dôsledku relatívne aktuálnych zmien študijného plánu bolo doteraz možné sledovať trend iba pre študentov prvých dvoch iterácií, čo je málo, pretože z dôvodu výrazne odlišného počtu študentov spomenutých dvoch iterácií sú porovnania náročné a zmeny ťažko sledovateľné.

Ktoré prvky navrhutej koncepcie boli pre študentov problematické? Je možné nájsť spôsoby ich redukcie? Ktoré prvky koncepcie mali pozitívny vplyv a je možné ich posilňovať?

Problematické prvky úvodného kurzu programovania sa snažíme detegovať priebežne. Následne vyvíjame aktivitu zameranú na ich redukciu, aby mali čo najmenší negatívny vplyv na ďalší postup študentov. V súčinnosti s tým sa zameriavame na posilňovanie väzieb medzi časťami preberaného učiva, používame pri tom aj metódu neustáleho opakovania preberaných prvkov.

Jedným z prvkov pri návrhu koncepcie bolo hľadanie správneho načasovania definície nových termínov počas semestra a s tým súvisiace určenie poradia preberaných tém. Táto činnosť je čiastočne závislá od získavania skúseností. Poradie tém v rámci sledovanej prvej časti úvodného kurzu programovania bolo výraznejšie ovplyvnené tými témami, ktoré boli presunuté zo zimného semestra do letného. Okrem toho sme poradie tém upravovali na základe analýzy úvodných kurzov programovania a ovplyvnili nás aj informácie získané štúdiom článkov s tematikou organizácie úvodných kurzov programovania (napríklad [31]).

Okrem poradia zohralo významnú úlohu zjednodušovanie vysvetľovania a oddiaľovanie definovania nových termínov dokedy sme to nepovažovali za nevyhnutné. V súvislosti s týmto prvkom našej koncepcie sa nám podarilo odhaliť iný jav. Vo väčšine prípadov považujeme zjednodušenie vysvetľovania za pozitívum, no v niektorých prípadoch sme mali možnosť presvedčiť sa o tom, že prílišné zjednodušenie môže mať negatívny vplyv na pochopenie učiva alebo na spôsob používania vyučovaných konceptov. Aj v tomto smere sme vykonali adekvátnu nápravu. V ďalších iteráciách sa preto budeme zaoberať aj sledovaním výskytu podobných situácií pri vyučovaní iných tém.

Vo vyučovacom procese sa systematicky vyskytuje aj iný jav, ktorý môžeme aspoň čiastočne ovplyvniť spôsobom organizácie semestra. Je ním nedostatočná pravidelná a systematická príprava študentov. Študenti sami priznávajú (v anonymných esejach), že to je (pretrvávajúci) problém – sú sebakritickí, čo je chvályhodné, ale problém to nerieši. Okrem organizačných prvkov, ktorými sa usilujeme ovplyvniť tento stav, vnímame ako dôležité aj dostatočné informovanie študentov o tejto problematike. V súvislosti s tým odporúčame dôrazne naliehať na to, aby študenti nepodceňovali svoje pravidelné prípravy a poctivo sa venovali nielen štúdiu teoretických konceptov, ale najmä riešeniu praktických úloh zadávaných buď vo forme *domácich úloh*, alebo úloh publikovaných v on-line *zbierkach úloh*.

Najvýznamnejším pozitívnym vplyvom meniacej sa koncepcie bolo zjemnenie prechodov medzi témami, zvýšenie času venovaného jednotlivým témam počas prvej časti úvodného kurzu, *vhodnejšie usporiadanie tém* a priebežné *posilňovanie vzájomných väzieb* preberaného učiva. Tieto prvky kurzu plánujeme zachovávať a zlepšovať. Z ďalších konkrétnych prvkov koncepcie, ktoré je dôležité zachovávať a prípadne posilňovať, by sme mohli spomenúť podporu *plánovania pred začatím programovania* a podporu *experimentovania* študentov *s jestvujúcimi programami*. Posledný spomenutý prvok súvisí so schopnosťou študentov čítať programy iných programátorov. Aj toto je kompetencia, ktorej získavanie by nemalo byť vytláčané do úzadia. Posledné dva prvky môžu vo vzájomnej súčinnosti viesť k významným pokrokom študujúceho.

Záver

Vykonali sme štyri iterácie vývoja úvodného kurzu programovania vyučovaného s použitím programovacieho jazyka Java. Výskum bol zameraný na prvú časť úvodného kurzu programovania, ktorú sme po zozbieraných skúsenostiach výraznejšie zamerali na upevňovanie základných konceptov programovania. V rámci celého kurzu používame nástroj, ktorý sme vyvinuli – skupinu tried grafického robota. Jej vývoj prebiehal súčasne s výskumom, preto mohol reflektovať na potreby kurzu. Môžeme skonštatovať, že vývoj skupiny tried grafického robota dospel do štádia, keď úplne pokrýva potreby prvej časti úvodného kurzu, avšak v budúcnosti plánujeme nástroj rozširovať o prvky, ktoré budú používané v druhej časti úvodného kurzu a v ďalších kurzoch programovania.

Skupina tried grafického robota je významnou, no nie jedinou intervenciou zavedenou v procese výučby úvodného kurzu programovania. Priebežne sme upravovali podmienky hodnotenia, presúvali sme témy vyučované počas dvoch semestrov úvodného kurzu programovania, niektoré témy sme z úvodného kurzu vynechali, postupne sme hľadali lepšie spôsoby vysvetľovania učiva s prípadnou podporou technických prostriedkov (výrobou a premietaním animácií a podobne), menili sme charakter úloh riešených počas prednášok a cvičení a zaoberali sme sa tvorbou učebných textov. Viaceré zo spomínaných prvkov budeme priebežne zlepšovať.

Naším plánom do ďalších iterácií je preskúmanie možností spolupráce so študentmi, ktorí už majú skúsenosti s programovaním, napriek tomu sú povinní absolvovať úvodný kurz. Títo študenti boli v doterajšej vzorke prijatých študentov v menšine a predpokladáme, že tento trend bude pokračovať aj naďalej. Chceli by sme nájsť vhodný spôsob ich zaradenia do úvodného kurzu, ktorý by bol prospešný aj pre nich. S tým súvisia okolnosti, a z nich vyplývajúci návrh, opísané v nasledujúcich odsekoch.

V dokumente, ktorý zverejňujeme študentom, deklaruujeme, že: *„vzájomná spolupráca študentov pri riešení programátorských problémov na cvičeniach je vítaná, kooperácia pri príprave na vyučovanie je žiadaná, ale výsledkom takejto kooperácie nesmú byť dve alebo viac identických kópií domácej úlohy odovzdaných rôznymi študentmi!“* Presvedčili sme sa, že záporné trestné body sú dostatočným „strašiakom“ na to, aby účinne zabránili vzájomnej spolupráci študentov pri domácej príprave až do tak miery, ktorá je nežiaduca – zistili sme, že niektorí študenti (údajne zo strachu z postihu) odmietali v akejkoľvek miere pomáhať svojim spolužiakom.

Ďalším predmetom úvah, ktorý považujeme za nevyriešený, sú skúsenejší študenti v oblasti programovania. Uvedomujeme si, že ich rast v rámci úvodného kurzu nemusí byť dostatočný. Preto sme sa začali zaoberať pre nás novou myšlienkou, ktorá by pri dobrej aplikácii mohla uvedené okolnosti riešiť. Ide o zváženie výberu pokročilých študentov úvodnými testami. Vybraní študenti (ak by sa takí našli) by získali privilégium „pomocných asistentov“, v rámci ktorého by mali upravené podmienky získania kreditov. Napríklad odovzdávanie domácich úloh by pre nich bolo nepovinné, body by získavali za „asistenciu“ – pomoc svojim rovesníkom priamo na hodinách. Učením a pomocou iným študentom by sa sami zdokonaľovali. Keďže ide o študentov učiteľstva, je možné, že by túto aktivitu bolo možné zohľadniť na budúcom predmete s náplňou povinnej praxe.

Predbežnými konzultáciami s kolegami sme získali prvé podrobnejšie informácie k tejto téme. Napríklad na Prírodovedeckej fakulte UPJŠ v Košiciach nemali dobré skúsenosti s asistentmi, ktorí boli rovesníkmi študentov. Študenti vyšších ročníkov sú vhodnejší. To by však nespĺňalo cieľ – chceli by sme zvýrazniť a zvýhodniť úlohu tzv. „ťahúňov“ v triede. Zvýhodnení študenti by nevykonávali úlohu plnohodnotných asistentov. Túto otázku bude potrebné podrobnejšie preskúmať...

Medzi ďalšie plánované aktivity patrí spracovanie nového elektronického interaktívneho vzdelávacieho materiálu použiteľného v on-line aj off-line forme. Projekt je v štádiu plánovania a počiatkovej fázy implementácie niektorých jeho budúcich komponentov. Začiatok celkovej realizácie je plánovaný na leto roku 2013. Materiál by mal spájať nasledujúce prvky:

- interaktívna mapa termínov presúvajúca študenta do konkrétnych pasáží vzdelávacieho materiálu,
- záznam osobného postupu študenta (uložený lokálne, s možnosťou on-line registrácie, ktorá by pre našich študentov bola povinnosťou a ktorá by umožnila uloženie záznamov o postupe študenta na serveri),
- postupné „odomykanie“ mapy termínov na základe zaznamenaného osobného postupu študenta,
- interaktívne príklady ukážok kódu,

- fulltextové vyhľadávanie,
- vzdelávací obsah s učebným textom, cieľmi, príkladmi, ukázkami, otázkami, simuláciami...

Interaktívny materiál bude obsahovať aj značnú mieru teoretického pozadia, omnoho väčšiu než poskytujeme v rámci prednášok a cvičení nášho kurzu. Je pravda, že programovanie sa nedá naučiť teoretizovaním, no tak ako pri iných premetoch, aj pri programovaní je potrebné oboznámiť študentov aj s istou mierou teoretického pozadia. Jeho znalosť môže v istých situáciách študentom pomôcť preklenúť prekážky. Nepovažujeme to za všeobecné pravidlo, v skutočnosti si myslíme, že získanie zručnosti programovania je možné len praxou a vytrvalým získavaním skúseností, ktoré sú najcennejšie, no ukazuje sa, že jestvujú typy študentov, ktorí znalosť teórie potrebujú a vyžadujú.

Tiež by sme chceli podrobnejšie preskúmať možnosti zavedenia výučby štýlom OO-first. Tento štýl vyučovania sme zatiaľ v praxi netestovali. Vývoj úvodného kurz bol časovo náročný a naše skúsenosti neboli postačujúce na to, aby sme navrhli novú verziu kurzu priamo s použitím metódy OO-first. Počas návštev konferencií sme sa pri rozhovoroch s kolegami z iných pracovísk stretli i s niekoľkými negatívnymi reakciami v súvislosti s touto metódou. To nás viedlo k opatrnosti v súvislosti s aplikáciou tejto metódy do nášho vyučovania.

Z uvedeného vyplýva, že niektoré otázky, ktoré vznikli priamo počas vývoja, zostali stále otvorené. Okrem spomínaných (napríklad preskúmanie zavedenia odlišného prístupu k programátorsky zdatnejším študentom) by sme mohli uviesť ďalšie: napríklad preskúmanie motivačného významu nepovinných („bonusových“) otázok v priebežných písomných prácach, ďalšie pozorovanie miery vzniku javu učenia sa programov naspamäť, doplnenie príkazového režimu do knižnice grafického robota, na dôležitosť ktorého upozorňuje [4], využitie študentského programovania naživo [18] a podobne.

Veríme, že všetky aplikované a plánované intervencie budú viesť k pokračujúcemu zvyšovaniu kvality vyučovania programovania na našej fakulte. Dúfame, že získané skúsenosti interpretované v tejto práci budú aplikovateľné aj ďalšími kolegami zaoberajúcimi sa vyučovaním programovania vo vysokoškolskom prostredí.

Zoznam bibliografických odkazov

- [1] Antonič, M.: *Delphi namiesto Javy*. Dostupné na: <http://miso.blog.matfyz.sk/p12296-delphi-namiesto-javy>, citované: 29. februára 2010.
- [2] *BlueJ – The interactive Java environment*. Dostupné na: <http://www.bluej.org/>, citované: 24. decembra 2012.
- [3] *BlueJ – Version History*. Dostupné na: <http://bluej.org/help/changes.html>, citované: 1. apríla 2013.
- [4] Brusilovsky, P. – Calabrese, E. – Hvorecky, J. – Kouchnirenko, A. – Miller, P.: Mini-languages: a way to learn programming principles. In *Education and Information Technologies*. Chapman & Hall, 1997. s. 65 – 83. ISSN 1360-2357.
- [5] Coats, M.: *Action Research A Guide for Associate Lecturers*. COBE, 2005. s. 5. Dostupné na: <http://www.open.ac.uk/cobe/docs/AR-Guide-final.pdf>, citované 1. januára 2011.
- [6] Cohen, L. – Manion, L. – Morrison, K.: *Research Methods in Education. 5th Edition*. London and New York : RoutledgeFalmer, 2001. ISBN 0-415-19541-1.
- [7] Cooper, S.: *CS106A Syllabus*. Stanford University. Dostupné na: <http://www.stanford.edu/class/cs106a/handouts/02-syllabus.pdf>, citované: 11. novembra 2010.
- [8] *CS106A Assignments*. Stanford University. Dostupné na: <http://www.stanford.edu/class/cs106a/links.html>, citované: 11. novembra 2010.
- [9] *CS106A Programming Methodology*. Stanford University. Dostupné na: <http://www.stanford.edu/class/cs106a/>, citované: 11. novembra 2010.
- [10] *CS1110. Lecture/lab summary*. Cornell University. Dostupné na: <http://www.cs.cornell.edu/Courses/CS1110/2010fa/handouts/courseOutline.html>, citované: 11. novembra 2010.
- [11] Csikszentmihalyi, M.: *Beyond Boredom and Anxiety: Experiencing Flow in Work and Play*. San Francisco : Jossey-Bass, 1975. ISBN 0-87589-261-2.
- [12] Csikszentmihalyi, M.: *Finding Flow: The Psychology of Engagement with Everyday Life*. New York : Basic Books, 1997.
- [13] Csikszentmihalyi, M.: *Flow, the secret to happiness*. TED, 2004. Dostupné na: http://www.ted.com/talks/mihaly_csikszentmihalyi_on_flow.html, citované: 1. augusta 2012.
- [14] *Download Blue*. Dostupné na: <http://www.cs.kent.ac.uk/people/staff/mik/blue/resources/downloads/load-blue.html>, citované: 16. marca 2010.
- [15] *Eclipse – The Eclipse Foundation open source community website*. Dostupné na: <http://www.eclipse.org/>, citované: 1. apríla 2013.
- [16] Fisher, R.: *Učíme děti myslet a učit se*. Praktický průvodce strategiemi vyučování. Praha : Portál, 2004. Orig. Teaching children to learn. ISBN 80-7178-966-6.
- [17] Fischer, P.: *Teaching Programming to Beginners*. 2004. <http://www2.imm.dtu.dk/~tb/fischer.pdf>, citované: 20. marca 2010.
- [18] Gaspar, A. – Langevin, S.: Restoring “Coding with Intention” in Introductory Programming Courses. In *SIGITE’07*. Florida, USA : ACM, 2007. s. 91 – 98. ISBN 978-1-59593-920-3.
- [19] Gavora, P.: *Spríevodca metodológiou kvalitatívneho výskumu*. Bratislava : Univerzita Komenského v Bratislave, 2007. ISBN 978-80-223-2317-8.
- [20] Gavora, P. a kol.: *Elektronická učebnica pedagogického výskumu*. Bratislava : Univerzita Komenského, 2010. ISBN 978-80-223-2951-4. Dostupné na: <http://www.e-metodologia.fedu.uniba.sk/>, citované: 9. septembra 2011.
- [21] Graham, P.: *Beating the Averages*. 2003. Dostupné na: <http://www.paulgraham.com/avg.html>, citované: 28. februára 2010.

- [22] Gries, D.: Teaching Programming (not programs). In *International Conference on Informatics in Secondary Schools: Evolution and Perspectives*. 2010. Prezentácia dostupná na: <http://www.issep2010.org/slides/gries.pdf>, citované: 28. februára 2010.
- [23] Gries, D. – Lee, L.: *CS 1110 Introduction to Computing using Java*. Cornell University. Dostupné na: <http://www.cs.cornell.edu/Courses/CS1110/2010fa/index.html>, citované: 11. novembra 2010.
- [24] Guzdial, M.: How we teach Introductory Computer Science is wrong. In *Blog at Communications of the ACM. Trusted insights for Computing's Leading Professionals*. 2009. Dostupné na: <http://cacm.acm.org/blogs/blog-cacm/45725>, citované: 28. februára 2010.
- [25] Harvey, B.: *CS 61A: Structure and Interpretation of Computer Programs – General Course Information*. Berkeley, USA : University of California. Dostupné na: <http://wla.berkeley.edu/~cs61a/fa10/0.pdf>, citované: 11. novembra 2010.
- [26] Hendl, J.: *Kvalitatívny výzkum – Základní teorie, metody a aplikace*. Praha : Portál, 2008. Druhé vydanie. ISBN 978-80-7367-485-4.
- [27] Herrod, J.: *Cognitive Overload*. 2000. Dostupné na: <http://www.ichconsulting.com/fall2000/>, citované: 1. augusta 2011.
- [28] Hoffman D. – Walsh P.: Teaching Programming With Minimal Examples. In *Western Canadian Conference on Computing Education*. Nanaimo, British Columbia : Malaspina University College, 1997. Dostupné na: <http://www.cs.ubc.ca/wccce/program97/walsh/walsh.html>, citované: 20. marca 2010.
- [29] Hong, J.: The Use of Java as an Introductory Programming Language. In *Crossroads. The ACM Student Magazine*. [Atlanta, USA] : Georgia institute of Technology, 1998. Dostupné na: <http://www.acm.org/crossroads/xrds4-4/introjava.html>, citované: 28. februára 2010.
- [30] Ivanović, M. – Komlenov, Ž. – Xinogalos, S.: Technology Enhanced Learning for Programming Courses – Experiences and Comparison. In *14th International Conference on Interactive Collaborative Learning (ICL2011) – 11th International Conference Virtual University (vu'11)*. Piešťany, Slovakia : IEEE, 2011. s. 42 – 45. ISBN 978-1-4577-1746-8.
- [31] Ivanović, M. – Pitner, T.: Technology-enhanced learning for Java programming: Duo cum faciunt idem, non est idem. In *ACM Inroads archive, Volume 2 Issue 1, March 2011*. NY, USA : ACM New York, 2011. s. 55 – 63. ISSN 2153-2184. EISSN 2153-2192.
- [32] *Java naming*. Dostupné na: <http://www.java.com/en/about/javanaming.jsp>, citované 1. marca 2013.
- [33] *Java návody*. Dostupné na: <http://java.skosi.org/javatutorial/>, citované: 1. apríla 2010.
- [34] *Java SE 6 Documentation*. 1993, 2011. Dostupné na: <http://docs.oracle.com/javase/6/docs/index.html>, citované: 1. apríla 2013.
- [35] *jEdit – Programmer's Text Editor*. Dostupné na: <http://jedit.org/>, citované: 1. apríla 2013.
- [36] *JPAZ2 framework*. Košice : Prírodovedecká fakulta UPJŠ v Košiciach. Dostupné na: <http://web.ics.upjs.sk/paz1a/JPAZ/JPAZ2>, citované: 1. marca 2013.
- [37] *JPAZ2 framework – JavaDoc*. Košice : Prírodovedecká fakulta UPJŠ v Košiciach. Dostupné na: <http://web.ics.upjs.sk/paz1a/files/jpaz/doc/index.html>, citované: 1. marca 2013.
- [38] Kalaš, I.: Čo ponúkajú informačné a komunikačné technológie iným predmetom (1. časť). In *Zborník konferencie Infovek 2001*. Bratislava, 2001. s. 71 – 82. ISBN 80-7098-265-5. <http://www.infovek.sk/archivwebu/konferencia/2000/prispevky/ikt.html>.
- [39] Kalaš, I.: Čo ponúkajú informačné a komunikačné technológie iným predmetom (2. časť). In *Zborník seminára Poškole 2001*. Praha : PF UK, 2001. s. 23 – 28. ISBN 80-01-02364-8.
- [40] Kalaš, I.: Čo ponúkajú informačné a komunikačné technológie iným predmetom (3. časť). Informatika a informatizácia. In *Infovek 2001. Zborník príspevkov 2. celoštátnej konferencie*. Stará Turá, 2001. <http://www.infovek.sk/archivwebu/konferencia/2001/zbornik/kalas.html>.
- [41] Kalaš, I.: Pedagogický výskum v informatike a informatizácii (2. časť). In *Zborník príspevkov konferencie DidInfo 2009*. Banská Bystrica : Univerzita Mateja Bela, 2009. ISBN 978-80-8083-720-4.

- [42] Kalhous Z. – Obst, O. a kol.: *Školní didaktika*. Praha : Portál, 2002. ISBN 80-7178-253-X.
- [43] Katz, R. H. – Patterson, D. A.: *CS 61C: Great Ideas in Computer Architecture (Machine Structures)*. Berkeley, USA : University of California. Dostupné na: <http://wla.berkeley.edu/~cs61c/>, citované: 11. novembra 2010.
- [44] Kirschner, P. A. – Sweller, J. – Clark, R. E.: Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching. In *Educational Psychologist*, 41(2), 2006. Lawrence Erlbaum Associates, Inc. p. 75–86. Dostupné na: <http://www.cogtech.usc.edu/publications/kirschner Sweller Clark.pdf>, citované: 18. marca 2010.
- [45] Kotrbčík, M.: *Hocičo namiesto Delhpi a Javy*. Dostupné na: <http://mito.blog.matfyz.sk/p12304-hocico-namiesto-delhpi-a-javy>, citované: 29. februára 2010.
- [46] Kölling, M.: *Quality-oriented teaching of programming*. 2009. Dostupné na: <http://blogs.kent.ac.uk/mik/2009/09/04/quality-oriented-teaching-of-programming/>, citované: 20. marca 2010.
- [47] Kubincová, Z.: *Prvácke programovanie v novom šate?* Dostupné na: <http://zuz.blog.matfyz.sk/p12507-prvacke-programovanie-v-novom-sate>, citované: 29. februára 2010.
- [48] Ladd, B. – Harcourt, E.: Student Competitions and Bots in an Introductory Programming Course. In *Journal of Computing Sciences in Colleges*. USA : Consortium for Computing Sciences in Colleges, 2005. s. 274 – 284. ISSN 1937-4771. EISSN 1937-4763.
- [49] Lewis, D.: *Cognitive Load Theory – a brief introduction to Cognitive load theory*. 7'th revision. 2008. Dostupné na: <http://knol.google.com/k/cognitive-load-theory#>, citované: 28. februára 2010.
- [50] Lister, R.: *After the Gold Rush: Toward Sustainable Scholarship in Computing*. Dostupné na: <http://crpit.com/confpapers/CRPITV78Lister.pdf>, citované: 16. marca 2010.
- [51] Malan, D. J.: *Introduction to Computer Science I / OpenCourseWare*. Cambridge, USA : Harvard College. Dostupné na: <http://cs50.tv/2009/fall/>, citované: 11. novembra 2010.
- [52] Mareš, M.: *Tři otázky k výuce programování*. Bratislava : KAM MFF UK.
- [53] Mazák, J.: *Programovanie 3*. Bratislava : FMFI UK. Dostupné na: <http://www.dcs.fmph.uniba.sk/~mazak/programovanie3/>, citované: 1. apríla 2010.
- [54] Miller, Ch. A.: *Action Research: Making Sense of Data*. Dostupné na: <http://www.coe.fau.edu/sfcel/sensdata.htm>, citované: 11. mája 2009.
- [55] Papert, S.: *Constructionism vs. Instructionism*. Text a videozáznam z prednášky, 1984. Dostupné na: http://www.papert.org/articles/const_inst/, citované 10. apríla 2010.
- [56] Papert, S.: *Mindstorms: children, computers and powerful ideas*. Basic Books, 1980. ISBN 0-465-04627-4.
- [57] Parsons, R. D. – Brown, K. S.: *Teacher as Reflective Practitioner and Action Researcher*. Belmont, California : Wadsworth/Thomson Learning, 2002. ISBN 0534557112.
- [58] Pattis, R. E.: *Karel The Robot: A Gentle Introduction to the Art of Programming*. John Wiley & Sons, 1981. ISBN 0-471-59725-2.
- [59] Petty, G. *Moderní vyučování*. Tretie vydanie. Praha : Portál, 2004. 380 s. ISBN 80-7178-978-X.
- [60] Porubän, J. – Václavík, P.: *Technologie Java*. Košice : TU FEI. Dostupné na: <http://www.kiv.zcu.cz/~ledvina/vyuka/poruban/index.html>, citované: 1. apríla 2010.
- [61] Pratt, D. Ch.: *The Construction of Meanings in and for a Stochastic Domain of Abstraction*. Dizertačná práca. London, UK : University of London, Institute of Education, 1998. Dostupné na: http://people.ioe.ac.uk/dave_pratt/Dave Pratt/Doctoral thesis files/thesis.pdf, citované 15. marca 2011.
- [62] Prokša, M. – Held, Ľ. a kol.: *Metodológia pedagogického výskumu a jeho aplikácia v didaktikách prírodných vied*. Bratislava : Prírodovedecká fakulta Univerzity Komenského v Bratislave, 2008. 229 s. ISBN 978-80-223-2562-2.

- [63] Reeves, T. C. – Herrington, J. – Oliver, R.: Design research: A socially responsible approach to instructional technology research in higher education. In *Journal of Computing in Higher Education*, 16(2). Springer US, 2005. s. 97 – 116. ISSN 1042-1726.
- [64] Robertson, J.: Introductory Computer Science Lessons – Take Heart! In *Blog at Communications of the ACM. Trusted insights for Computing's Leading Professionals*. 2009. Dostupné na: <http://cacm.acm.org/blogs/blog-cacm/46781>, citované: 28. februára 2010.
- [65] Sedgewick, B.: *Computer Science 126 – General Computer Science*. Princeton, USA : Princeton University. Dostupné na: <http://www.cs.princeton.edu/courses/archive/fall10/cos126/info.html>, citované: 11. novembra 2010.
- [66] Sedgewick, R. – Wayne, K.: *Computer Science 126 – General Computer Science*. Princeton, USA : Princeton University. Dostupné na: <http://www.cs.princeton.edu/courses/archive/fall10/cos126/lectures.php>, citované: 11. novembra 2010.
- [67] Sedgewick, R. – Wayne, K.: *Introduction to Programming in Java*. Princeton, USA : Princeton University. Dostupné na: <http://www.cs.princeton.edu/introcs/home/>, citované: 11. novembra 2010.
- [68] Sedgewick, R. – Wayne, K.: *Introduction to Programming in Java – Standard Libraries*. Princeton, USA : Princeton University. Dostupné na: <http://www.cs.princeton.edu/introcs/stdlib/>, citované: 1. marca 2013.
- [69] Shewchuk, J.: *CS 61B: Data Structures*. Berkeley, USA : University of California. Dostupné na: <http://www.cs.berkeley.edu/~jrs/61b/>, citované: 11. novembra 2010.
- [70] Spolsky, J.: *The Perils of JavaSchools*. 2005. Dostupné na: <http://joelonsoftware.com/articles/ThePerilsofJavaSchools.html>, citované: 28. februára 2010.
- [71] Spolsky, J. and others: *How best to teach programming to non-programmers?* 2004. Dostupné na: <http://discuss.fogcreek.com/newyork/default.asp?cmd=show&ixPost=1671>, citované: 20. marca 2010.
- [72] Ščerbák, G.: *Java namiesto Delphi*. Dostupné na: <http://gscerbak2.blog.matfyz.sk/p12293-java-namiesto-delphi>, citované: 29. februára 2010.
- [73] *Štátne vzdelávacie programy*. Štátny pedagogický ústav. Dostupné na: <http://www.statpedu.sk/sk/sections/view/statne-vzdelavacie-programy/statny-vzdelavaci-program>, citované: 20. marca 2010.
- [74] Švaříček, R. – Šed'ová, K. a kol.: *Kvalitatívny výzkum v pedagogických vědách*. Praha : Portál, 2007. ISBN 978-80-7367-313-0.
- [75] *The Java Language Environment*. Dostupné na: <http://java.sun.com/docs/white/langenv/Intro.doc2.html>, citované: 6. apríla 2010.
- [76] Turek, I.: *Inovácie v didaktike*. Bratislava : Metodicko-pedagogické centrum v Bratislave, 2004. ISBN 80-8052-188-3.
- [77] *UC Berkeley Webcasts, Video and Podcasts: Computer Science 61A, 001 – The Structure and Interpretation of Computer Programs*. Berkeley, USA : University of California. Dostupné na: http://webcast.berkeley.edu/course_details_new.php?seriesid=2010-D-26275&semesterid=2010-D, citované: 11. novembra 2010.
- [78] *Úplné znenie Zákona č. 131/2002 Z. z. o vysokých školách*. Dostupné na: http://web.tuke.sk/lf/legislativa/131_2002.pdf, citované: 24. augusta 2010.
- [79] Vranič, V.: *Objektovo-orientované programovanie, ak. rok 2009/10*. Bratislava : FIIT STU. <http://www2.fiit.stuba.sk/oop/c/c03.html>, citované: 1. apríla 2010.
- [80] Wang, F. – Hannafin, M. J.: Design-based research and technology-enhanced learning environments. In *Educational Technology Research and Development*, 53(4). 2005. s. 5 – 23. ISSN 1042-1629.
- [81] Yadin, A.: Reducing the Dropout Rate in an Introductory Programming Course. In *ACM Inroads*. New York, USA : 2011. s. 71 – 76. ISSN 2153-2184. EISSN 2153-2192.
- [82] Zakhour, S. – Hammel, S. – Royal, J. – Rabinovitch, I. – Risser, T. – Hoeber, M.: *Java 6. Výukový kurz*. Computer Press, a.s., 2007. Prvé vydanie. ISBN 978-80-251-1575-6.

- [83] *Zákon č. 175/2008 Z. z. úplné znenie zákona č. 131/2002 Z. z. o vysokých školách a o zmene a doplnení niektorých zákonov v znení neskorších predpisov.* Dostupné na: http://www.minedu.sk/data/USERDATA/Legislativa/Zakony/2008_175.pdf, citované: 24. augusta 2010.
- [84] Zelle, J. M.: *Python as a First Language.* Dostupné na: <http://mcsp.wartburg.edu/zelle/python/python-first.html>, citované: 28. februára 2010.

Zoznam vlastných publikácií

- [85] Andrejková, G. – Horváth, R.: Hodnotenie učebníc programovania. In *DidInfo 2013*. Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, 2013. s. 28 – 39. ISBN 978-80-557-0527-9.
- [86] Horváth, R.: Software testing in introductory programming courses. In *ICETA 2012*. Košice : TU, 2012. s. 133 – 134. ISBN 978-1-4673-5123-2. Dostupné na: <http://ukftp.truni.sk/epc/8319.pdf>.
- [87] Horváth, R.: Porovnanie hodnotenia študentov z predmetu PA1 s bodovaním získaným v prijímacom konaní na PdF TU. In *DidInfo 2012*. Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, Katedra informatiky, 2012. s. 83 – 86. ISBN 978-80-557-0342-8.
- [88] Horváth, R.: *Dokumentácia skupiny tried grafického robota pre Javu.* Trnava : PdF TU, 2012. [20,55 AH]. ISBN 978-80-8082-537-9. Dostupné na: <http://cec.truni.sk/horvath/Robot/>, citované: 4. marca 2013.
- [89] Horváth, R.: Nový model vyučovania programovania na Pedagogickej fakulte TU. In *Acta Facultatis Paedagogicae Universitatis Tyrnaviensis*. Trnava : Trnavská univerzita, Pedagogická fakulta, 2011. s. 98 – 102. ISBN 978-80-8082-514-0. Dostupné na: <http://pdf.truni.sk/actafp/2011/c/#HORVATH>.
- [90] Horváth, R.: Teaching one Language in More Depth is better than Many Languages Superficially. In *ICETA 2011*. Stará Lesná, The High Tatras, Slovakia; Budapest, Hungary : Óbuda University, 2011. s. 71 – 74. ISBN 978-1-4577-0050-7.
- [91] Horváth, R.: *Sieťové nastavenia OS Windows XP.* Trnava : Trnavská univerzita v Trnave, Pedagogická fakulta, 2011. 126s. ISBN 978-80-8082-454-9. Dostupné na: <http://cec.truni.sk/horvath/SietXP/>.
- [92] Horváth, R.: *Multimédiá a Internet.* Trnava : Trnavská univerzita v Trnave, Pedagogická fakulta, 2011. ISBN 978-80-8082-441-9. Dostupné na: <http://cec.truni.sk/horvath/Mal/>.
- [93] Horváth, R.: Paradigmy v programovaní. In *XXII. DidMatTech 2009*. Trnava; Komárno : Trnava University, J. Seyle University, 2010. s. 188 – 194. ISBN 978-80-8122-006-7.
- [94] Horváth, R.: Inovácia vyučovania predmetov programovania na PdF TU. In *Acta Facultatis Paedagogicae Universitatis Tyrnaviensis*. Trnava : PdF TU, 2009. s. 64 – 68. ISBN 978-80-8082-320-7.
- [95] Školková, K. – Horváth, R.: Porovnanie úrovne IKT zručností študentov na Pdf TU. In *XXI. DIDMATTECH 2008*. Eszterházy Károly College, 2009. s. 55 – 62. ISBN 978-963-9894-18-1.
- [96] Horváth, R.: Implementácia programátorského rozhrania pre tvorbu interaktívnych vzdelávacích materiálov. In *DidInfo 2009*. Banská Bystrica : Univerzita Mateja Bela, 2009. s. 94 – 97. ISBN 978-80-8083-720-4.
- [97] Horváth, R. – Školková, K.: Využitie LMS systému na Pedagogickej fakulte Trnavskej univerzity v Trnave. In *eLearning 2008*. Hradec Králové : Gaudeamus, 2008. s. 196 – 201. ISBN 978-80-7041-143-8.
- [98] Školková, K. – Horváth, R. – Mišút, M.: Využitie elektronického vzdelávacieho portálu na Pdf TU. In *Perspective in Education Process at Universities with Technical Orientation in Visegrad Countries*. Nitra : Slovenská poľnohospodárska univerzita v Nitre, 2008. s. 229 – 233. ISBN 978-80-552-0148-1.
- [99] Školková, K. – Horváth, R.: Pripravenosť študentov na Pdf TU z oblasti IKT. In *Nové trendy vo vyučovaní matematiky a informatiky na základných, stredných a vysokých školách*. Žilina : Žilinská univerzita, 2008. 5s. ISBN 978-80-8070-863-4. Dostupné na: <http://ukftp.truni.sk/epc/1832.pdf>.

- [100] Horváth, R. – Školková, K.: Podpora výučby využitím LMS systému na PdF TU. In *Súčasné trendy vo vzdelávaní odbornej a laickej verejnosti v zdravotníckych disciplínach*. Trnava : Trnavská univerzita v Trnave, Fakulta zdravotníctva a sociálnej práce, 2008. s. 138 – 141. ISBN 978-80-8082-214-9.
- [101] Horváth, R.: *Programovanie a algoritmy*. Trnava : Trnavská univerzita, Pedagogická fakulta, 2008. 151s. ISBN 978-80-8082-211-8. Dostupné na: <http://cec.truni.sk/horvath/Pascal/>.
- [102] Horváth, R.: Vyučovanie informatiky na SŠ s využitím elektronického kurzu. In *DidInfo 2007*. Banská Bystrica : Univerzita Mateja Bela, Fakulta prírodných vied, 2007. s. 50. ISBN 978-80-8083-367-1. Dostupné na: <http://ukftp.truni.sk/epc/1827.pdf>.
- [103] Horváth, R. – Školková, K.: Testovanie študentov zo zručností IKT na Pedagogickej fakulte TU. In *DidInfo 2006*. Banská Bystrica : Univerzita Mateja Bela, 2006. s. 109 – 111. ISBN 978-80-8083-202-1.
- [104] Horváth, R.: Elektronický vzdelávací systém na Pedagogickej fakulte Trnavskej univerzity v Trnave. In *Odborný seminár o využití LMS EKP v akademickom prostredí*. Nová Dubnica : 2006. s. 1 – 5. ISBN 80-969555-5-1.
- [105] Horváth, R. – Mišút, M.: The New Improvements of E-learning System at Trnava University. In *ICETA 2005*. Košice : Elfa, 2005. s. 157 – 160. ISBN 80-8086-016-6.
- [106] Horváth, R. – Mišút, M.: Non-traditional in-project communication. In *CO-MAT-TECH 2005*. Bratislava : Vydavateľstvo STU, Materiálovotechnologická fakulta Trnava STU Bratislava, 2005. s. 1 – 5. ISBN 80-227-2286-3.
- [107] Gabaľová, V. – Horváth, R. – Mišút, M.: Skúsenosti s on-line výučbou predmetu IKT. In *Kultúra – priestor interdisciplinárneho myslenia 4*. Nitra : Univerzita Konštantína Filozofa v Nitre, 2005. s. 83 – 86. ISBN 80-8050-837-2.
- [108] Horváth, R. – Školková, K.: Skúsenosti so správou elektronického vzdelávacieho systému. In *Pedagogický software 2004*. České Budějovice : Scientific Pedagogical Publishing, 2004. s. 513 – 515. ISBN 80-85645-49-1.
- [109] Horváth, R. – Mišút, M.: Využitie e-learning štandardov v implementácii elektronického vzdelávania na PdF TU. In *eLearning ve vysokoškolském vzdělávání 2004*. Zlín : Univerzita Tomáše Bati ve Zlíně, 2004. ISBN 80-7318-190-8.
- [110] Gabaľová, V. – Horváth, R. – Mišút, M.: Skúsenosti s on-line vzdelávaním na PdF TU. In *Pedagogický software 2004*. České Budějovice : Scientific Pedagogical Publishing, 2004. s. 509 – 512. ISBN 80-85645-49-1.
- [111] Gabaľová, V. – Horváth, R. – Mišút, M.: On-line system at Faculty of education of Trnava University. In *Information and communication technology in education*. Ostrava : University of Ostrava, Faculty of Science, 2004. ISBN 80-7042-993-3.
- [112] Mišút, M. a kol.: Interaktívna výučba predmetu IKT. In *I&IT '04*. Banská Bystrica : Univerzita Mateja Bela, FPV, Katedra informatiky, 2004. s. 170 – 174. ISBN 80-8083-017-7.
- [113] Horváth, R. – Mišút, M.: Rationalisation of Educational Process by E-Learning. In *ICETA 2004*. Košice : Elfa, 2004. s. 319 – 321. ISBN 80-89066-85-2.
- [114] Horváth, R. – Mišút, M.: Role of standards in development of e-learning system at Trnava's University. In *CO-MAT-TECH 2004*. Bratislava : Slovak University of Technology, 2004. s. 373 – 377. ISBN 80-227-2117-4.
- [115] Horváth, R. – Mišút, M.: Role of standards in development of e-learning system at Trnava's University. In *CO-MAT-TECH 2004*. Bratislava : Vydavateľstvo STU, Materiálovo-technologická fakulta Trnava STU Bratislava, 2004. ISBN 80-227-2121-2.
- [116] Gabaľová, V. – Horváth, R.: Skúsenosti s využívaním LMS systému na Pedagogickej fakulte Trnavskej univerzity. In *DIVAI 2004*. Nitra : KI FPV UKF v Nitre, 2004. s. 46. ISBN 80-8050-691-4.
- [117] Horváth, R. – Mišút, M. – Pokorný, M.: Virtual University in Trnava – Experience. In *ICETA 2003*. Košice : Elfa, 2003. ISBN 80-89066-67-4.

Súhrn

Vyučovanie môže mať vo všeobecnosti mnoho podôb a oblasť programovania je dostatočne rozmanitá aj ako samostatný odbor. V práci je uvedený návrh výučby úvodného kurzu programovania pre cieľovú skupinu budúcich učiteľov informatiky. Predložený návrh je výsledkom viacročnej výučby. Bola použitá kvalitatívna výskumná metodológia „výskum vývojom“ (Design-Based Research), ktorá je založená na spolupôsobení iteratívneho vývoja predmetu skúmania s jeho výskumom. Výskumný projekt začal v roku 2009, predmetom skúmania bola prvá časť úvodného kurzu programovania pre budúcich učiteľov a pred publikovaním výsledkov boli vykonané štyri iterácie. Programovacím jazykom je Java. Počas výskumu opísaného v práci bola pre potreby výučby vyvinutá skupina tried grafického robota, ktorá sa stala neoddeliteľnou súčasťou vyučovacieho procesu. Bola vyvíjaná s ohľadom na výsledky analýzy predchádzajúcej výskumu a spätnej väzby získavanej počas výskumu od študentov. Jedným zo známejších princípov implementovaných v triede je možnosť používania korytnačej grafiky. Okrem nich je v skupine tried implementovaných viacero prvkov zameraných na zjednodušenie vyučovania konceptov, ktoré jazyk Java poskytuje v zbytočne komplikovanej forme.

Summary

The thesis includes a proposal of introductory programming course for the target group which consists of future teachers of informatics. This proposal is the result of a multi-annual teaching. The qualitative research methodology Design-Based Research has been used. The research project started in 2009 and examined the first part of the introductory programming course for future teachers. Four iterations were performed prior to publication of the results. The programming language is Java. The new tool “The Graphical Robot Classes Group” has been developed during the research work. It has become an integral part of the teaching process. The development considered the results of on-going research and analysis of feedback received during the research from students. One of the well-known principles implemented in this tool is the turtle graphics.