

Nový model vyučovania programovania na Pedagogickej fakulte TU

Roman Horváth

Katedra matematiky a informatiky
Pedagogická fakulta, Trnavská univerzita
Priemyselná 4, P. O. Box 9, Trnava 918 43, SR
e-mail: rhovath@truni.sk

Abstract: *The New Model of Teaching Programming at the Faculty of Education TU.* Teaching programming is an area that must constantly seek new solutions. Even at our faculty (PdF TU), we decided to upgrade this process. Over the years, several changes were made and the process of innovation is not over yet. There are not only technical changes, but mainly conceptual. We focus on an introductory programming course – two-term course in the first year of study. Teaching programming is currently supported by Java at our faculty, which means that we do not declare the pure Java teaching. In the classroom we use a set of classes developed (graphic robot [1]) and the tool BlueJ [2] that was designed for the purpose of teaching Java. The set of classes is part of a new model recently implemented at our faculty.

Keywords: problems and solutions in teaching programming, Robot class, Java programming

1 Úvod

Vyučovanie programovania je oblasťou, ktorá musí neustále hľadať nové riešenia. Aj na našej fakulte (PdF TU) sme sa rozhodli inovovať tento proces. Za uplynulé roky bolo vykonaných viacero zmien a proces inovácie sa ešte neskončil. Netýka sa len technických zmien, ale najmä koncepčných. Zameriavame sa na úvodný kurz programovania (dvojsemestrálny predmet v prvom ročníku), na ktorý nadväzujú ďalšie predmety.

Programovanie na našej fakulte v súčasnosti prebieha s využitím jazyka Java, čo znamená, že priamo nedeclarujeme vyučovanie rýdzeho jazyka Java. Vo vyučovaní používame nami vyvinutú sústavu tried grafického robota [1] a nástroj BlueJ [2], ktorý bol navrhnutý pre potreby vyučovania Javy. Sústava tried je súčasťou nového modelu v súčasnosti implementovaného na našej fakulte.

2 Identifikácia problémov a implementácia nového modelu vyučovania

Identifikácia problémov prebiehala formou anonymných esejí a rozhovorov, pri ktorých mali študenti za úlohu opísať svoje postrehy a názory na priebeh, spôsob, nedostatky a výhody vyučovania programovania na Pedagogickej fakulte, ktoré boli schopní postrehnúť. Niekoľko rozhovorov sme vykonali aj s bývalými študentmi našej fakulty.

Najfrekvencovanejšie informácie periodicky sa opakujúce v zozbieraných dátach od roku 2009, môžeme zhrnúť do nasledovných bodov:

- nedostatok času – študenti považovali dva semestre úvodného kurzu programovania za nedostatočný objem času,

- priepasť medzi tým, čo sa učiteľ snaží vysvetliť a tým, čomu študenti skutočne porozumeli – skúsenejší študenti označili spôsob vysvetľovania za dostatočne zrozumiteľný, začiatovníci ho považovali za príliš profesionálny a technický,
- vnútorná motivácia – študenti nastúpili do študijného programu so skreslenými predstavami o predmete informatika a boli demotivovaní faktom, že budú nútení učiť sa programovať,
- disproporcía pracovných skupín – vo vzorke prijímaných študentov sú veľké rozdiely týkajúce sa miery predošlých skúseností s programovaním; v rámci jednej skupiny sa vyskytujú študenti, ktorí sa s programovaním nestretli vôbec a študenti, ktorí programovali a ktorí sa zaoberajú programovaním i vo voľnom čase,
- tímová spolupráca – viacerí študenti ocenili pomoc skúsenejších spolužiakov a my sme sa rozhodli, že budeme podporovať skupinovú spoluprácu pri vyučovaní programovania.

V nasledujúcom texte podrobnejšie opíšeme obsah predošlých bodov, nie však v tom poradí, v akom boli uvedené. Predošlé body sú zoradené viac-menej podľa početnosti výskytu v zozbieraných údajoch. Podrobnosti v nasledujúcom texte sú utriedené podľa vzájomných súvislostí.

V zozbieraných dátach sa často opakovala téma „veľkých problémoch s porozumením“ učiteľovho vysvetľovania študentmi. Mnohí uviedli, že sami (v rámci svojej skupiny) postrehli, že „iba niekoľko spolužiakov skutočne porozumelo podstate.“ V dátach získaných z rozhovorov a esejí je evidentné, že študenti sa o problémoch pri učení sa programovania vzájomne rozprávajú. Niektorí vyjadrili svoj problém s učením sa expresívne, slovami: „mali sme problém s chytením vlaku.“ Predpokladáme, že tomuto problému sa nedá vyhnúť na sto percent, ale mal by byť záležitosťou malej časti študentov daného ročníka, nie väčšiny. Pokúsili sme sa klásť väčší dôraz na nadväznosť preberaného učiva, pretože problém zachytenia kontinuity preberanej látky sa spravidla u daného jednotlivca zväčšuje. Keď nastane takýto prípad, je takmer nemožné dobehnúť zameškané a pokračovať plynule vo vzdelávaní sa.

Pri analýze vyjadrení o oblastiach, ktoré študenti považovali za problematické, ktorú sme vykonávali na neskôr získaných dátach, sa síce ukázalo, že takmer každá časť osnov bola minimálne jedným študentom označená za problematickú, preto sme selektovali najčastejšie spomínané časti osnov. Iróniou bolo zistenie, že študenti najčastejšie spomínali tie časti osnov, ktorým bolo v rámci vyučovania venované najviac času. Navyše išlo o základné prvky programovacieho jazyka (vetviace výroky, cykly, tvorba podmienok, definícia metód a podobne; jedine jednoduché vetvenie bolo študentmi označované za ľahko pochopiteľné). Zdá sa, že neúnavným opakovaním základov sa nedarí dosiahnuť želaný efekt porozumenia a zvládnutia naučeného učiva. Podozrenie padá na nedostatočnú vnútornú motiváciu študentov.

Bezpochyby, študenti potrebujú čas na pochopenie i tých základných konceptov programovania a programovacieho jazyka. Najmä ak sa predtým s programovaním nikdy nestretli. Skutočne, predmetom esejí číslo dva bol práve čas. Študenti často žiadali zníženie tempa a sťažovali sa na rýchly postup na hodinách. Písali o tom, že potrebujú trochu času na to, aby mohli nad predloženým problémom sami porozmýšľať a venovať sa mu. Rozprávali o tom, že nie je možné naraz odpisovať program z tabule (plátna a projektora), počúvať a rozmýšľať nad daným problémom. Chválili tie momenty, kedy vyučujúci zastavil a nechal ich rozmýšľať nad problémom, pričom v skôr získaných dátach (2009/2010) bolo podotknuté, že takých momentov nebolo veľa. Preto sme sa postarali o nápravu. V neskôr získaných dátach (2011) už nebol dôvod na to, aby sa takáto poznámka v dátach objavila, pretože študenti dostávali priestor na rozmýšľanie pravidelne, v miere únosnej pre časový rámec vyučovacej hodiny.

Už v prvých dátach (2009) sa objavovali požiadavky na zníženie tempa. Aj po ich akceptovaní, študenti (nasledovných ročníkov) stále tvrdili, že tempo je príliš rýchle. Požiadavky sa stále opakovali. Na jednej strane rozumieme, že mnohé zistené problémy sú reálne a súvisia s efektom nazývaným kognitívne preťaženie [3], [4] a mali by sme sa snažiť zabrániť tomu, aby k takémuto preťaženiu u študentov prišlo. Avšak na problém nedostatku času nejestvuje priamočiare riešenie. Kurzy programovania sú vsadené do presného časového rámca. Stretávame sa s problémom, či a ako je možné rozumne znížiť objem preberaného učiva, pretože nemáme možnosť zvýšiť časovú dotáciu predmetu.

Čiastočným riešením je dávať väčší dôraz na domácu prácu, čo aplikujeme a v priebehu rokov zvyšujeme dôraz domácej prípravy študentov, aby boli študenti donútení pracovať aj doma a pripravovať sa na predmet priebežne počas semestra. Zaviedli sme povinné päťminútovky na začiatkoch cvičení, ktorých výsledky sú súčasťou celkového hodnotenia. Ďalšou časťou riešenia je zmena spôsobu vyučovania. Tú sme realizovali vývojom skupiny tried grafického robota, ktorá je opísaná v [5]. Uvažovali sme (a stále uvažujeme) aj nad aplikáciou väčšej, či menšej časti teórie toku od Mihályho Csíkszentmihályiho [6]. Najst' spôsob, ako dosiahnuť efekt opisovaný touto teóriou (pohltenie aktivitou) je možno záležitosťou dlhšieho skúmania a je otázne, či je tento stav možné navodiť v prostredí, kde je zvýšená pravdepodobnosť výskytu osôb s nedostatočnou vnútornou motiváciou, avšak prinajmenšom aplikácia nasledovných dvoch prvkov teórie: (1) mať stály jasný cieľ, ktorý predložíme študentom a (2) poskytovať im neustálu spätnú väzbu; by sa mala stať neoddeliteľnou súčasťou vyučovania našich predmetov.

Ďalším problémom, ktorý sme postrehli na hodinách a ktorý sa vyskytol i v zozbieraných dátach, je disproporcía pracovných skupín. Postrehli ju i samotní študenti a v dátach sa prejavovala aj diametrálne odlišnými vyjadreniami k rovnakej téme, napríklad väčšina skupiny žiadala spomalenie tempa, ale našli sa v nej i takí, ktorí žiadali zrýchlenie, pričom sa sťažovali na spolužiakov, ktorí systematicky robia tie isté primitívne preklepy s ktorými si (spravidla) nevedia sami poradiť, čím zdržujú celú skupinu... (Niekoľko študentov sebareflektovalo tento problém.) Skúsenejší sa nudia, začiatočníci sa sťažujú na rýchle tempo. V súčasnosti nemáme účinné riešenie tohto problému vhodné na aplikáciu a potrebujeme ho najst'. Samozrejme, ak to naše časové možnosti dovoľia, vytvoríme databázu príkladov, ktorú by za bonifikačnú odmenu mohli na hodinách riešiť skúsenejší študenti v čase, keď sa potrebujeme venovať tým menej skúseným. Potrebovali by sme však instantné riešenie do doby, než takáto databáza príkladov vznikne.

Začiatočníkom pomohlo, keď im skúsenejší spolužiaci boli ochotní pomôcť. Preto schvaľujeme a povzbudzujeme študentov ku skupinovej spolupráci, aby skúsenejší študenti pomáhali tým, ktorí sa s programovaním stretajú prvýkrát. Zdôrazňujeme pritom, že vzájomná pomoc musí mať vhodnú formu, nezahŕňa napríklad odpisovanie domácich úloh. Začiatočníkom taktiež pomohlo, ak mali možnosť učiť sa prostredníctvom upravovania existujúcich programov. Túto časť koncepcie by sme mali posilniť, pretože v súčasnosti jej nevenujeme dostatočnú pozornosť. Nakoniec, viacerí študenti deklarovali, že je výhodou, ak majú vo svojom okolí (rodina, známi, priatelia) niekoho, kto sa s programovaním stretol a vie a je ochotný im pomôcť. Tento faktor však nie sme schopní nijako ovplyvniť.

Problémom úplne iného charakteru, ktorý sme sami postrehli v rámci vyučovania, počas skúškového obdobia a na ktorý upozornili aj niektorí študenti, bolo učenie sa programov naspamäť – bez porozumenia. Tí študenti, ktorí sa anonymne priznali k takémuto počínaniu uviedli, že nevideli inú možnosť. Vznik tohto fenoménu súvisí aj s vnútornou motiváciou a tým, že študenti nastupujúci na štúdium informatiky nemajú jasnú predstavu o tom, čo termín „štúdium informatiky“ znamená.

Skreslené predstavy o zdokonaľovaní svojich zručností v ovládaní textového procesora bývajú rozbité faktom, že budú nútení učiť sa programovať. Študent naplnený takýmito skreslenými predstavami býva po uvedomení si reality demotivovaný a uchýľuje sa i k takémuto neefektívnemu riešeniu – učeníu sa programov naspamäť bez náznaku porozumenia. S takto nezvládnutým učivom študent nedokáže vyriešiť ani dvakrát zadaný rovnaký problém s odlišnými parametrami. Riešenie problému neinformovanosti a zabránenie „šoku z reality“ je vysoko nad rámec našich kompetencií. Vyžaduje informovanosť mladej generácie o význame termínu „štúdium informatiky“. Nezostáva len veriť, že sa tento pretrvávajúci problém časom vyrieši...

Poslednou informáciou získanou od študentov, ktorú tu uvedieme, bola požiadavka na viacej študijných materiálov v slovenskom jazyku. Na splnení tejto požiadavky neustále pracujeme a budeme pracovať. Očakávali sme podobné reakcie. Robíme, čo je v našich silách. Je naplánovaných viacero publikácií, vďaka ktorým časom pokryjeme potreby všetkých predmetov programovania.

Všetky informácie získané z rozhovorov a esejí boli zapracované do nového modelu vyučovania programovania na našej fakulte, ktorý je v súčasnosti (akademický rok 2011/2012) testovaný.

3 Záver

Nový model vyučovania úvodného kurzu programovania bol navrhnutý pred začiatkom akademického roka 2011/2012. V súčasnej dobe prebieha jeho implementácia a overovanie. V rámci návrhu modelu boli nasledovnými formami riešené otázky identifikované v prvej etape:

- zmena spôsobu vyučovania,
- zjednodušenie výkladu učiteľa, lepšia vizualizácia učiva, poskytnutie širšieho priestoru na analýzu predložených problémov študentmi na hodinách,
- prehodnotenie vyučovacích osnov predmetu, revízia plánu, redukcia a transformácia niektorých oblastí,
- vývoj skupiny tried grafického robota (opísaná v [5]),
- integrácia časti teórie toku (podrobnosti v [6]), predovšetkým stanovenie stálych jasných cieľov, ich predloženie študentom a poskytovanie neustálej spätnej väzby, čo by malo pomôcť dosiahnuť stav opisovaný spomínanou teóriou,
- tvorba elektronických vzdelávacích materiálov pre potreby predmetu,
- kladenie väčšieho dôrazu na domácu prácu, poskytovanie programov na samostatnú analýzu a modifikáciu,
- zavedenie povinných päťminútoviek,
- neustále povzbudzovanie študentov ku vzájomnej spolupráci pri učení sa programovania.

Literatúra

- [1] HORVÁTH, R.: *Dokumentácia skupiny tried grafického robota pre Javu*. [online]. <http://cec.truni.sk/horvath/Robot/>. Accessed: 24. Dec 2011.
- [2] *BlueJ – The interactive Java environment*. [online]. <http://www.bluej.org/>. Accessed: 11. Nov 2011.
- [3] 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. [online]. <http://cacm.acm.org/blogs/blog-cacm/45725>. Accessed: 28. Feb 2010.
- [4] HERROD, J.: *Cognitive Overload*. 2000. [online]. <http://www.jchconsulting.com/fall2000/>. Accessed: 1. Aug 2011.

-
- [5] HORVÁTH, R.: Teaching one Language in More Depth is better than Many Languages Superficially. In *ICETA 2011*. Budapest, Hungary : Óbuda University, 2011. s. 71-74. ISBN 978-1-4577-0050-7.
- [6] CSIKSZENTMIHALYI, M.: *Beyond Boredom and Anxiety: Experiencing Flow in Work and Play*. San Francisco : Jossey-Bass, 1975. ISBN 0-87589-261-2.