



Univerzita Komenského v Bratislave

Fakulta matematiky, fyziky a informatiky

PaedDr. Ján Guniš

Autoreferát dizertačnej práce

Tvorba algoritmov a rozvoj algoritmického myslenia žiakov v stredoškolskej informatike

na získanie akademického titulu philosophiae doctor
v odbore doktorandského štúdia:
9.2.3. teória vyučovania informatiky

Bratislava, 2013

Dizertačná práca bola vypracovaná v externej forme doktorandského štúdia na Katedre základov a vyučovania informatiky, Fakulty matematiky, fyziky a informatiky, Univerzity Komenského v Bratislave.

Predkladateľ: PaedDr. Ján Guniš
Ústav informatiky
Prírodovedecká fakulta
Univerzita P. J. Šafárika v Košiciach
Jesenná 5
041 54 Košice

Školiteľ: doc. RNDr. Gabriel Semanišin, Phd.
Ústav informatiky
Prírodovedecká fakulta
Univerzita P. J. Šafárika v Košiciach
Jesenná 5
040 01 Košice

Oponenti: doc. RNDr. Jana Kapounová, CSc.
Katedra informačných a komunikačných technológií
Pedagogická fakulta
Ostravská univerzita v Ostravě
Fráni Šrámka 3
709 00 Ostrava - Mariánské Hory

doc. RNDr. Gabriela Andrejková, CSc.
Ústav informatiky
Prírodovedecká fakulta
Univerzita P. J. Šafárika v Košiciach
Jesenná 5
040 01 Košice

RNDr. Andrej Blaho, PhD.
Katedra aplikovanej informatiky
FMFI UK
Mlynská dolina
824 48 Bratislava

**Obhajoba dizertačnej práce sa koná o h
pred komisiou pre obhajobu dizertačnej práce v odbore doktorandského štúdia vymenovanou
predsedom odborovej komisie**

9.2.3. teória vyučovania informatiky
(študijný odbor)

Teória vyučovania informatiky
(názov študijného programu doktorandského štúdia)

**na Fakulte matematiky, fyziky a informatiky Univerzity Komenského, Mlynská dolina, 842 48
Bratislava.**

Predseda odborovej komisie:

.....
(meno a priezvisko s uvedením titulov a hodností
a presná adresa jeho zamestnávateľa)

Úvod

Obsah školskej informatiky, ako relatívne mladého všeobecno-vzdelávacieho vyučovacieho predmetu, sa postupne stabilizuje. Vzhľadom na rýchly vývoj v tejto oblasti, resp. v oblastiach ovplyvňujúcich obsah školskej informatiky je prakticky nemožné, aby sa obsah školskej informatiky ustálil do konečnej a nemennej podoby. Aj napriek práve prebiehajúcej školskej reforme sa však domnievame, že „búrlivé“ obdobie školskej informatiky je už za nami. Pre našu prácu je podstatné najmä to, akú pozíciu majú a budú mať v školskej informatike tematické okruhy ako riešenie problémov, tvorba algoritmov a, algoritmizácia a pod. Z pohľadu vzdelávania a vzdelania žiaka nás zaujíma aj pozícia a vzťah týchto tém k ostatným predmetom a témam týchto predmetov.

Prvé snahy o vyvážený obsah školskej informatiky sa objavujú v 90-tych rokoch. V roku 1997 vznikajú učebné osnovy pre gymnáziá. V osnovách sa nachádza tematická okruh „Algoritmy a algoritmizácia“ so zatiaľ stručne uvedeným cieľom: „zvládnuť algoritmické riešenie úloh pomocou počítača“.

V súčasnej verzii učebných osnov informatiky (ŠPÚ, 2008b) nájdeme tento tematický okruh uvedený pod názvom „Postupy, riešenie problémov, algoritmické myslenie“. Stredobodom záujmu je problém, hľadanie riešenia problému a rozvoj algoritmického myslenia. Téma algoritmizácie a programovania tak prestáva byť samoúčelnou, predstavuje sa ako silný „nástroj“ užitočný pri riešení problémov.

Pri bližšom pohľade na vyučovanie však zisťujeme, že výučba tejto témy sa niekedy uberá nie správnym smerom alebo dôraz sa nekladie na podstatné veci. Žiak často vystupuje v úlohe pasívneho prijímateľa informácií a jeho konečná predstava o význame algoritmizácie a programovania je skreslená a nekorešponduje s realitou. Žiaci majú problémy zvládnuť túto tému a neraz pociťujú frustráciu z neúspechu. Rozhodli sme sa preto túto problematiku bližšie preskúmať a navrhnúť metodiku vyučovania, ktorá by čo najviac eliminovala nami identifikované negatívne javy vyučovania tohto tematického okruhu.

Výskumný projekt dizertačnej práce

V tejto kapitole prezentujeme výskumný projekt našej dizertačnej práce. V prvej podkapitole uvádzame výskumný problém, jeho ciele a otázky. V druhej podkapitole uvádzame stratégie a metódy jeho riešenia.

Výskumný problém

Obsah a ciele školskej informatiky sú v novej koncepcii školstva definované v Štátnom vzdelávacom programe (ŠPÚ, 2008b). Overenie funkčnosti a efektívnosti tejto koncepcie je dlhodobý, niekoľkoročný proces. V našom projekte sme sa snažili vytvoriť a overiť metodiku vyučovania tematického okruhu „Postupy, riešenie problémov, algoritmické myslenie“ tak, aby spĺňala požiadavky, ktoré sme vopred kvalifikovane nastavili.

Výskumný problém: Tvorba metodiky vyučovania tematického okruhu „Postupy, riešenie problémov, algoritmické myslenie“ v základnom kurze stredoškolskej informatiky

Výskumná oblasť: Vyučovanie tematického okruhu „Postupy, riešenie problémov, algoritmické myslenie“ v základnom kurze stredoškolskej informatiky.

Výskumný projekt

Hlavným cieľom našej dizertačnej práce bolo navrhnúť a overiť metodiku vyučovania tematického okruhu „Postupy, riešenie problémov, algoritmické myslenie“ na gymnáziách s dôrazom na proces aktívneho učenia sa žiaka a na prepojenie s reálnou praxou prostredníctvom riešenia problémov.

Metodika, ktorú sme vytvorili zahŕňa:

- podrobnejšiu špecifikáciu obsahového a výkonového štandardu definovaného v Štátnom vzdelávacom programe (ŠPÚ, 2008c),
- exemplifikačné úlohy pre diagnostikovanie plnenia požiadaviek vzdelávacieho štandardu,
- pracovné listy pre žiakov,
- metodické usmernenia a komentáre k výučbe tohto tematického okruhu,
- naše postrehy a skúsenosti z vyučovania tohto tematického okruhu.

Pre splnenie hlavného cieľa bolo potrebné splniť aj nasledujúce čiastkové ciele:

- Analyzovať súčasný stav vyučovania tematického okruhu „Postupy, riešenie problémov, algoritmické myslenie“,
- Analyzovať obsah vyučovania a metodické materiály z pohľadu vyučovania algoritmizácie,
- Vymedziť východiská pre tvorbu metodiky,
- Navrhnuť obsah vzdelávania,
- Navrhnuť a overiť metodiku vyučovania tematického okruhu „Postupy, riešenie problémov, algoritmické myslenie“.

Vo vytvorenej metodike sme zohľadnili didaktické zásady modernej pedagogiky, náročnosť a poradie tém, vstupné predispozície žiakov a prepojenie s inými oblasťami ľudskej činnosti.

Výskumné otázky

Vo výskumnom projekte sme stanovili nasledujúce otázky:

- Ako špecifikovať predispozície žiakov, programátorov začiatocníkov, s ktorými prichádzajú na strednú školu?
- Aký časovo-tematický plán navrhnúť, aby žiaci boli schopní dosiahnuť ciele definované v ŠVP?
- Aké metodické postupy a prostriedky zvoliť, aby žiaci dosiahli definované vzdelávacie ciele?
- Ako diagnostikovať úroveň algoritmického myslenia žiakov?

Návrh a metódy riešenia projektu

V nasledujúcej časti uvádzame stratégie a metódy riešenia nášho projektu a zdôvodnenie ich výberu. Uvádzame tiež, ako sme zabezpečili kvalitu a etickosť výskumu.

Riešenie projektu sme realizovali v dvoch fázach. Cieľom fázy jeden bolo špecifikovať predispozície žiakov, začínajúcich programátorov. Aj na základe týchto výsledkov sme vo fáze dva vyvinuli metodiku vyučovania tematického okruhu „Postupy, riešenie problémov, algoritmické myslenie“. Aj keď fázy označujeme číslami, neznamená to, že bežali oddelene, nezávisle a fáza dva po skončení fázy jeden.

Výber výskumnej stratégie

Naším cieľom bolo vytvoriť metodiku vyučovania postavenú na chápaní reality žiakmi. Potrebovali sme preskúmať žiacky pohľad a predispozície žiakov. Chceli sme porozumieť tomu, ako žiaci rozmýšľajú a uvažujú. Pracovali sme vždy s menšími skupinkami žiakov v ich autentickom prostredí. V analyzovaní ich výstupov sme sa snažili ísť vždy do hĺbky a na základe toho sme interpretovali ich pochopenie reality. V projekte sme nepracovali s vopred danou teóriou, ale teóriu sme postupne, z analyzovaných javov induktívne vytvárali. Vyššie uvedené atribúty sú charakteristické pre metodológiu kvalitatívneho výskumu, ku ktorej sme sa počas vývoja metodiky priklonili.

Navrhnutú metodiku a jej efektívnosť sme chceli overiť v reálnej výučbe. Chceli sme ukázať, že použitím navrhutej metodiky možno dosiahnuť významný nárast úrovne algoritmického myslenia. V tejto situácii je vhodné použiť nástroje kvantitatívnej metodológie.

Rozhodnutie o metódach

Vo etape vytvárania výskumného projektu (Švaříček, 2007) sme rozpracovali tri kľúčové body: rozhodnutie o vzorke, výber metódy zberu dát a zaistenie vstupu do terénu.

Rozhodnutie o vzorke: Na základe cieľov projektu bola pre nás zaujímavá skupina žiakov, ktorá študovala na strednej škole (resp. na gymnáziu) a doteraz nemala žiadne skúsenosti s programovaním a s cieľovým vyučovaním algoritmizácie a programovania. Celkovo sme skúmali (vo fáze jeden) tri skupiny žiakov v priebehu štyroch rokov.

Vo fáze dva sme boli limitovaní existujúcim Školským vzdelávacím programom školy kde sme výskum realizovali. Jednotlivé iterácie novovznikajúcej metodiky sme vytvárali a testovali počas niekoľkých rokov nášho pôsobenia na tejto škole v tých skupinách žiakov, v ktorých sme realizovali riadnu výučbu.

Metódy zberu dát: Výučba témy „Postupy, riešenie problémov, algoritmické myslenie“ v konečnom dôsledku smeruje k písomnému vyjadreniu algoritmu. Aj z tohto dôvodu sme sa vo fáze jeden rozhodli pre písomnú komunikáciu so žiakmi (pri programovaní platí to, čo je napísané a nie to, čo žiak chcel alebo si myslel). Keďže len z programu (zápisu algoritmu pre robota IPOS) by bolo problematické vyvodit' spôsob uvažovania žiaka, naša komunikácia obsahovala aj doplňujúce otázky výskumníkov k žiackym „programom“ a odpovede žiakov na tieto otázky. Získané dáta sme podrobili obsahovej analýze.

Fázu dva (vývoj metodiky) sme realizovali priamo v triede, počas regulárnej výučby. Okrem písomnej komunikácie so žiakmi (žiacke riešenia úloh a naše komentáre k týmto riešeniam) sme v tejto fáze so žiakmi komunikovali najmä ústne a pozorovali ich reakcie na naše intervencie. Naše pozorovania sme si čiastočne počas

hodiny a po hodine zapisovali (terénne zápisky). Na základe reakcií žiakov sme metodiku priebežne upravovali a doladžovali.

Vstup do terénu: V tomto výskume sme mali výhodu v tom, že výskumník bol zároveň regulárnym učiteľom žiakov. Nepôsobili sme ako cudzinci a žiaci sa správali a reagovali otvorene, bez prehnanej snahy o pozitívnu seba reprezentáciu alebo snahy o únik z interakcie s výskumníkom. Na druhej strane je takáto pozícia výskumníka náročnejšia na udržanie si odstupu a schopnosti vnímať veci objektívne.

Dizajn kvalitatívneho výskumu

Dizajn výskumu (Švaříček, 2007) definuje rámcové usporiadanie alebo plán výskumu. Pri jeho návrhu uvažujeme o podmienkach realizácie výskumu. Definujeme čo a prečo budeme skúmať a ako to budeme skúmať. Pre náš výskum vo fáze jeden sme zvolili dizajn zakotvenej teórie (Švaříček, 2007). Vo fáze dva sme sa inšpirovali dizajnom výskumu vývojom (Kalaš, 2009).

V našom výskume (fáza jeden) sme začali zberom dát, ktoré sme priebežne analyzovali a priradžovali k nim poznámky. Zadaní úloh ktoré žiaci riešili sme postupne vytvárali a modifikovali podľa toho, aké javy sme v dátach nachádzali.

Pre definovanie východísk fázy dva výskumu sme analyzovali problémy vyučovania tematického okruhu „Postupy, riešenie problémov, algoritmické myslenie“, základnú terminológiu skúmanej oblasti, existujúce programovacie prostredia, obsah a ciele vzdelávania v uvedenom tematickom okruhu, učebnice a postupy vyučovania tohto tematického okruhu. Navrhli sme diagnostické nástroje na overenie efektívnosti a validity navrhnutých postupov.

V etape reflexie k produkcii vývojových princípov sa vraciame k pôvodnému problému a zisťujeme, či nami navrhnuté riešenie je skutočne riešením pôvodného problému. Tento postup sme realizovali počas celého vývoja metodiky. Pri každej pedagogickej intervencii sme zisťovali, či táto intervencia napomáha žiakom k dosahovaniu nami vytýčených cieľov. Posledným krokom bolo overenie metodiky ako celku. Tu sme použili metódy kvantitatívneho výskumu

Zabezpečenie kvality výskumu

Okrem samotnej analýzy žiackych produktov vo fáze jeden sme si naše závery potvrdžovali formou otázok žiakom a ich odpoveďami. S každým žiakom sme teda uskutočnili sériu akýchsi neštruktúrovaných „mini“ interview. Vo fáze dva okrem samotného vývoja metodiky sme jej úspešnosť overovali testami a písomkami žiakov, ktoré boli súčasťou riadnej výučby.

Etika a výskum

Súčasťou našej práce bol kvalitatívny výskum na žiakoch strednej školy. Uvedomujúc si „zraniteľnosť“ tínedžerov sme sa snažili (najmä vo fáze jeden), aby sa necítili ako „skúmané pokusné zajace“. Náš výskum sme preto založili na riešení úloh, ktoré žiaci vnímajú ako prirodzenú súčasť ich žiackeho života. Využili sme aj fakt, že sme regulárnymi učiteľmi žiakov, takže naša prítomnosť nebola pre žiakov rušivá a zároveň sme eliminovali vplyv cudzej osoby na prirodzené správanie sa žiakov.

Nechceli sme, aby žiaci zažívali „traumu“ z neúspechu (i keď aj neúspech môže byť hnacou silou k úspechu). Vo fáze jeden projektu žiaci vedeli, že nesprávne

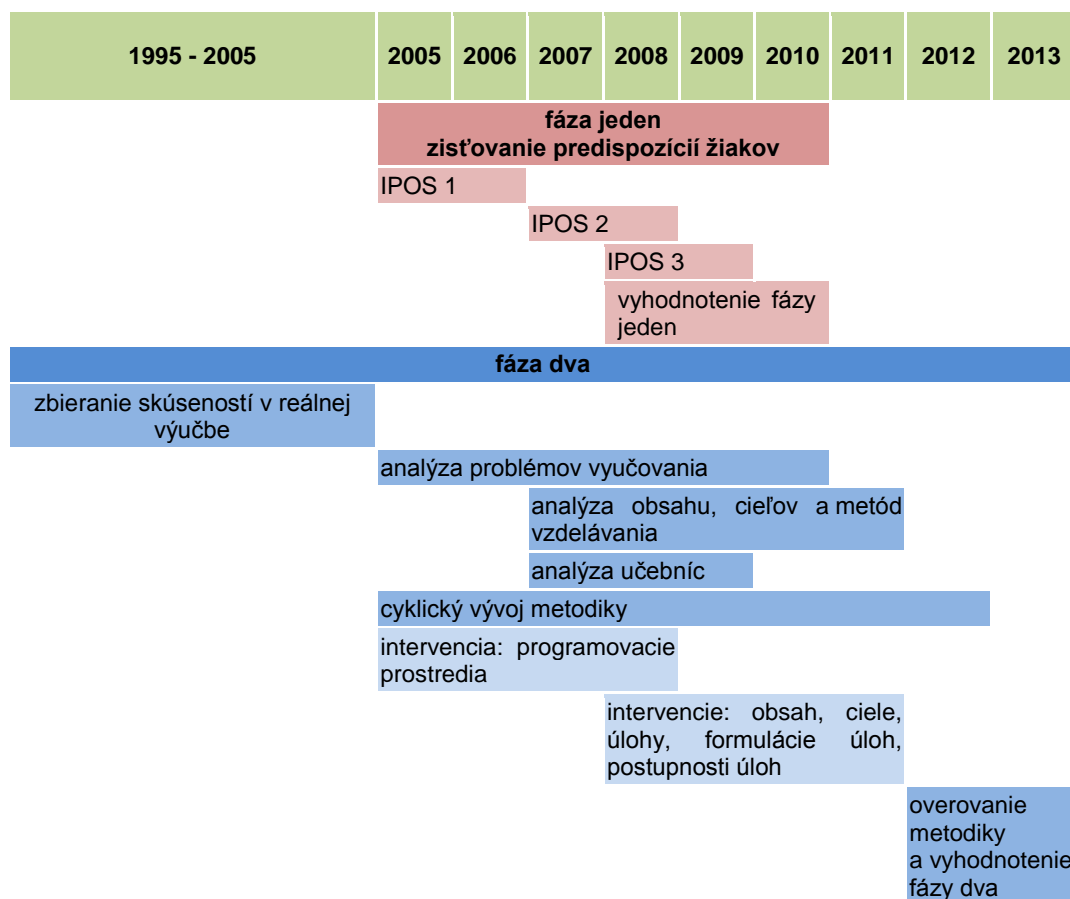
riešenie úloh neexistuje a za komplexné vyriešenie problému môžu byť odmenení dobrou známku. Nezáujem zapojiť sa do výskumu nemal za následok žiadne negatívne dôsledky. Odpovede žiakov neboli prístupné iným žiakom a tak žiaci mohli uvádzať svoje skutočné názory bez obavy zo zosmiešnenia inými spolužiakmi. Žiaci boli oboznámení so skutočnosťou, že sú účastníkmi výskumu a že získané poznatky poslúžia na hlbšie pochopenie ich myslenia a predispozícií s cieľom zefektívniť ich výučbu.

Vo fáze dva bolo riešenie úloh integrálnou súčasťou výučby a žiaci to aj tak vnímali a prijali.

Pri uvádzaní výsledkov výskumu uvádzame kódy žiakov a to bez ohľadu na to, či prezentujeme pozitívne či negatívne príklady (z nášho pohľadu) riešenia žiakov.

Časový harmonogram riešenia projektu

Pre lepšiu orientáciu a nadväznosť uvádzame hrubý časový harmonogram jednotlivých etáp riešenia projektu.



Fáza jeden, vyhodnotenie

Ak si odmyslíme chyby z nepozornosti alebo nesprávne pochopenie problému (napr. riešenie konkrétnej inštancie problému) môžeme konštatovať, že všetky riešenia žiakov boli správne. Význam pojmu správne tu však má trochu iný význam. Riešenia boli správne z pohľadu žiakov, z ich uhla chápania reality, z ich predpokladov. Takmer všetky prípady, keď sme riešenia „označili“ za nesprávne boli

spôsobené nepochopením toho, ako žiak svoje riešenie myslel, resp. chybnou interpretáciou žiakovho riešenia výskumníkom.

Človek, ktorý má skúsenosti s programovaním má istý cit pre definovanie množiny inštrukcií. Intuitívne sa snaží definovať primitívne inštrukcie. Žiaci tento cit ešte nemajú. Často sa zaoberajú problémom ako celkom bez toho, aby riešili jeho detaily. Detaily, na ktorých žiaci pri klasickom programovaní často zlyhávajú sa v riešeníach skúmaných žiakov „nejako“ vyriešia. Nie je potrebné sa nimi zaoberať, z pohľadu žiaka sú nepodstatné.

Žiaci bez problémov používajú základné inštrukcie, ktoré si definovali. Tu treba poznamenať, že základné inštrukcie ale neznamená jednoduché inštrukcie. Zložitú činnosť, ktorá v sebe skrýva netriviálny postup (napr. nájdenie východu z miestnosti) žiaci riešili definovaním základnej inštrukcie.

Dekompozícia problému na menšie podproblémy a následná syntéza riešení jednotlivých podproblémov je jedna zo základných metód riešenia problémov. Počet úrovní do koľkých rozkladáme prvotný problém závisí od zložitosti problému ktorý riešime a nástrojov, ktoré máme k dispozícii. Žiaci si často vystačili len s prvou úrovňou, pretože už na nej mali definované základné inštrukcie. Schému, ktorá sa pri riešení problému používa, žiaci značne zjednodušili.

Žiacke riešenia boli často rozvláčne a nepresné (veď robot to pochopí tak, ako ja chcem). Ich množina inštrukcií nebola optimalizovaná (nepoužité parametre, dlhé popisné názvy a pod.) Takéto programy kontrastujú s programami v existujúcich programovacích jazykoch.

Na jednej strane nás môže tešiť, že žiaci vedia riešiť problémy. Na druhej strane si uvedomujeme, aké závažné problémy pri programovaní v budúcnosti spôsobia ich naivné očakávania ovplyvnené prežívanou realitou.

V tomto prípade žiaci používali vlastný, „flexibilný“ formalizmus, ktorý sa prispôboval ich ponímaniu skutočnosti. Túto skutočnosť žiaci aj využívali. Ak im však poskytneme nejaký vopred daný a nemenný formalizmus, táto flexibilita sa vytratí. Z tohto pohľadu bol tento postup dobrým diagnostickým nástrojom na zistenie predispozícií žiakov.

Aj na základe týchto zistení sme navrhli metodiku vyučovania tematického okruhu „Postupy, riešenie problémov, algoritmické myslenie.“

Fáza dva, návrh metodiky

Zaradenie jednotlivých problémov, ktoré žiaci postupne riešia a ich poradie je závislé od množiny nástrojov a riadiacich štruktúr, ktoré budú použité v riešení problému, od náročnosti riešenia a požiadavke na vyjadrenie riešenia. Pri rozhodovaní o zaradení a poradí jednotlivých problémov sme využili výsledky fázy jeden.

V metodike staviame na tom, čo už žiaci vedia, resp. čo im je prirodzené. Prvky učiva, ktoré by mohli byť problematické (na základe výsledkov fázy 1) zavádzame postupne a cielene ich precvičujeme pri hľadaní riešení uvedených problémov. Riešenia problémov odporúčame so žiakmi realizovať skupinovo. Žiaci sa tak učia argumentovať pre a proti nejakému návrhu, kriticky prijímať iné návrhy a pod. Okrem iného tým podporujeme prácu v tíme a rozvoj sociálnych a komunikačných zručností.

Zadania problémov sme formulovali ako reálne problémy. Zároveň sme chceli dosiahnuť skutočnosť, aby výsledok žiackeho snaženia bol pre žiakov zaujímavý. Ku každému problému existuje krátka zbierka úloh. V zbierke sú zaradené jednoduchšie problémy alebo problémy nadväzujúce na problém z pracovného listu. Tieto úlohy môže učiteľ využiť na precvičovanie (napr. v domácej príprave). Pri riešení týchto úloh si môže žiak (ale aj učiteľ) overiť, ako danú problematiku zvládol.

Pre každý problém je pre žiaka pripravený pracovný list (celkom 9). Cieľom pracovného listu je naučiť žiakov pracovať systematicky a vopred premyslene. Cieľom je, aby sa takýto systematický prístup stal pre žiaka návykom, osvojeným spôsobom správania sa pri hľadaní riešenia ľubovoľných problémov. Pracovný list teda neslúži na to, aby žiak našiel riešenie zadaného problému, ale na to, aby sa naučil riešiť problémy vo všeobecnosti.

Stručný prehľad obsahu pracovných listov

- **Problémy a ich riešenie.** Prehľad učiva: inštrukcia, postupnosť inštrukcií, riešenie problému, algoritmická riešiteľnosť problému.
- **Laboratórne váženie - etapy riešenia problému (1).** Prehľad učiva: vstupy a výstupy algoritmov, vzťah medzi nimi, analýza problému, podstata problému, vykonávateľ algoritmu, algoritmus vyhľadávania, kresliace algoritmy, testovanie správnosti algoritmu.
- **Prievoznický problém - etapy riešenia problému (2).** Prehľad učiva: vstupy a výstupy algoritmov, vzťah medzi nimi, analýza problému, podstata problému, vykonávateľ algoritmu, testovanie správnosti algoritmu, podmienky riešiteľnosti, cyklus s pevným počtom opakovaní, dekompozícia problému na podproblémy, algoritmy usporadúvania, zápis algoritmu, stratégia riešenia problému – vytvor tabuľku, podmienený príkaz.
- **Tanečná choreografia – z jednoduchého vytvorme zložité.** Prehľad učiva: stratégia riešenie problémov – dekompozícia problému, prístup zdola na hor, prístup z hora na dol, jednoznačnosť, elementárnosť a formálnosť vyjadrenia algoritmu, vykonávanie algoritmu, cyklus, vytvorenie modelu problému, analýza algoritmu.
- **Učíme tancovať korytnačku – prvé programovanie.** Prehľad učiva: programovací jazyk, programovacie prostredie, vykonávateľ algoritmu, pomocník programovacieho prostredia, základné príkazy korytnačej grafiky, procedúry, analýza algoritmu, analýza problému, cyklus, časová efektívnosť algoritmu, dekompozícia problému na podproblémy.
- **Impresionistická korytnačka, počítač v rukách umelca.** Prehľad učiva: analýza problému, korytnačia grafika, generovanie náhody, vlastnosti korytnačky, karteziánska súradnicová sústava, pomocník programovacieho prostredia, počítačom generovaná náhoda a tvorivosť počítača, komentár v programe.
- **Maľujeme vlastné obrázky, rozhodnutia a podmienky.** Prehľad učiva: podmienka, zložená podmienka, podmienený príkaz úplný a neúplný, vnorené podmienené príkazy, efektívnosť algoritmu, rôzne spôsoby zápisu algoritmu, pravidlá prehľadného zápisu programu.

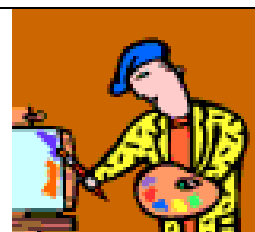
- **Hodiny na webovej stránke alebo „Rozdeľuj a panuj“.** Prehľad učiva: stratégia riešenia problémov – dekompozícia problému, analýza problému, premenná, procedúra s parametrom.
- **Cyklopočítač, povolanie programátora.** Prehľad učiva: analýza problému, špecifikácia zadania problému, analýza vstupných dát, overenie ich platnosti, algoritmy vyhľadávania, štruktúrovaná premenná typu zoznam, počítadlo cyklu, testovanie správnosti algoritmov, vlastný interpreter príkazov, implementácia dátových štruktúr zoznam a rad.

Ukážka pracovného listu s metodickým komentárom

Pre lepšiu názornosť uvádzame ukážku jedného, v poradí siedmeho, pracovného listu. V pracovnom liste využívame a nadväzujeme na žiacke vedomosti a zručnosti, ktoré nadobudli počas riešenie predchádzajúcich problémov.

Maľujeme vlastné obrázky, rozhodnutia a podmienky

V predchádzajúcom pracovnom liste sme naučili korytnačku prekresľovať existujúce obrázky v impresionistickom štýle. Použité vyššie uvedené techniky na maľovanie impresionistických obrazov **bez predlohy**. Naučte korytnačku namaľovať nasledujúce vlajky. Pred samotným programovaním korytnačky odpovedzte na nasledujúce otázky:

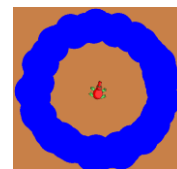
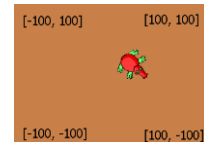
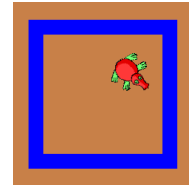


 Ukrajina	Podľa akých pravidiel bude korytnačka meniť farbu kreslenia? Čo nové musí korytnačka vedieť spraviť?
 Nigéria	Podľa akých pravidiel bude korytnačka meniť farbu kreslenia? Čo nové musí korytnačka vedieť spraviť?
 Benin	Podľa akých pravidiel bude korytnačka meniť farbu kreslenia? Čo nové musí korytnačka vedieť spraviť?
 Japonsko	Podľa akých pravidiel bude korytnačka meniť farbu kreslenia? Čo nové musí korytnačka vedieť spraviť?
 Anglicko	Podľa akých pravidiel bude korytnačka meniť farbu kreslenia? Čo nové musí korytnačka vedieť spraviť?
 Švédsko	Podľa akých pravidiel bude korytnačka meniť farbu kreslenia? Čo nové musí korytnačka vedieť spraviť?

Úlohy na precvičenie:

Ak by ste potrebovali vedieť príkaz na kreslenie nejakého útvaru, alebo vykonanie nejakej činnosti pozrite si pomocníka Imagine alebo sa spýtajte svojho vyučujúceho (napr. cez diskusné fórum v LMS Moodle).

- Zvieratá v ZOO sa často bezcieľne pohybujú po kletke. Zviera kráča nejakým smerom až kým nenarazí na stenu kletky. Pri nej sa náhodne otočí a kráča novým smerom až kým opäť nenarazí na stenu. Takto sa to opakuje celý deň. Ako asi vyzerá pohyb takého zvierťa? Stiahnite si projekt **ZOO.imp** a naprogramujte procedúru **pohybVKletke**, ktorá bude pomocou korytnačky simulovať pohyb zvierťa v kletke. Priestor kletky je vyznačený hnedou farbou ("hnedá"), stena kletky má farbu modrú ("modrá").
- Majiteľom ZOO sa kletky veľmi nepáčia. Rozhodli sa preto, že kletky zrušia a zviera sa bude pohybovať „akože“ bez obmedzení. Dostane elektrický obojok a keď prekročí pomyselnú hranicu kletky, dostane slabý elektrický úder. To ho prinúti, aby sa otočilo a pokračovalo iným smerom. Hranice pomyselné kletky určili majitelia súradnicami na ploche, po ktorej sa zviera pohybuje. Naprogramujte procedúru **pohybBezKletky**, ktorá bude pomocou korytnačky simulovať pohyb zvierťa.
- Návštevníci ZOO hromadne protestovali proti elektrickým obojkom zvierat. Majitelia preto nemali na výber a zrušili aj obojky. Problém s tým, aby im zvieratá neušli, nakoniec vyriešili veľmi jednoducho. Vykopali okolo zvierťa priekopu a naplnili ju vodou. Keďže tieto zvieratá nemajú radi vodu, nikdy do nej nevkročia. Naprogramujte procedúru **pohybPoSuchu**, ktorá bude pomocou korytnačky simulovať pohyb zvierťa na ploche ohraničenej vodnou priekopou (farba "modrá").



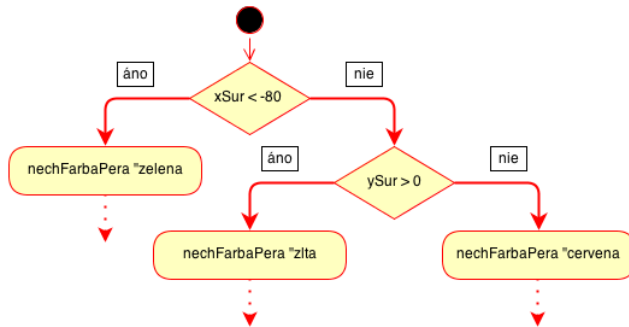
Metodický komentár

Časová náročnosť: 2-3 vyučovacie hodiny

V tomto pracovnom liste sa žiaci oboznámia s ďalšou riadiacou štruktúrou, podmieneným príkazom. Žiaci pokračujú v kontexte predchádzajúceho pracovného listu, ale obrázky nevznikajú prekresľovaním predlohy. Vytvárajú sa na základe vopred definovaných pravidiel (podmienok). Očakávame, že si žiaci uvedomia potrebu rozhodovania sa počas samotného generovania obrázkov. Rozhodovanie sa deje na základe pozície korytnačky v súradnicovom systéme stránky. Zložitosť podmienok sa postupne stupňuje. Zatiaľ čo pre prvé dve zástavy stačí pracovať s jednou súradnicou pozície korytnačky, pri ďalších zástavách je potrebné uvažovať súradnice obidve.

Očakávame, že sa žiaci naučia vytvárať jednoduché a zložené podmienky, pracovať s neúplným aj úplným podmieneným príkazom, vytvárať a používať vnorené podmienené príkazy. Odporúčame venovať priestor aj efektívnosti vzájomného zostavenia podmienok. Naše skúsenosti ukazujú, že v otázke množstva testovaných podmienok majú žiaci spočiatku mylnú predstavu o efektívnosti navrhnutého postupu. Efektívnosť vyjadrujú celkovým súčtom podmienok v postupe, teda súčtom vnútorných uzlov rozhodovacieho stromu. Na samotnú zložitosť má však vplyv počet vyhodnotených podmienok, teda hĺbka rozhodovacieho stromu. Kvôli zjednodušeniu predpokladáme, že pravdepodobnosti vyhodnotenia podmienok sú rovnaké. V nadväznosti na toto zdôvodnenie žiaci prichádzajú s ďalším mylným úsudkom. Vo svojich postupoch začnú používať zložené podmienky. Tým sa im môže podariť opticky znížiť hĺbku stromu. Neuvedomia si však dva dôsledky. Pri testovaní napr. jednoduchej alternatívy je nutné v niektorých prípadoch vyhodnotiť až

tri podmienky. Niektoré jednoduché podmienky už sú súčasťou viacerých zložených podmienok, takže je potrebné ich vyhodnocovať viackrát. Pri zápise podmienok odporúčame najskôr proces rozhodovania zapísať prirodzeným spôsobom (v prirodzenom jazyku) a až potom ho prepísať do programovacie jazyka. Pripomíname však, že formalizmus takého zápisu nie je cieľom, ale len nástrojom na dosiahnutie lepšieho pochopenia logiky programu.



```

ak2 xSur < -80 [
  nechFP "zelena"
][
  ak2 ySur > 0 [
    nechFP "zlta"
  ][
    nechFP "cervena"
  ]
]
  
```

Rôzne úrovne formalizmu pri zápise podmienok v algoritme kreslenia vlajky Beninu

Prostredie Imagine Logo nepodporuje farebné zvýraznenie syntaxe a vizuálne párovanie zátvoriek. Odporúčame preto postupne budovať u žiakov nasledovné návyky:

- ak napíšem ľavú zátvorku, napíšem k nej hneď aj zodpovedajúcu pravú zátvorku a kód vpišem medzi ne,
- ak chcem použiť úplný podmienený príkaz, napíšem obidva páry zátvoriek a kód budem vpišovať do párov zátvoriek,
- vnorené príkazy budem odsadzovať oproti nadradeným.

Kreslenie japonskej vlajky je mierne odlišné. Do úvahy berieme vzdialenosť od stredu a nielen samotné porovnanie súradníc pozície korytnačky. Žiaci tak musia vyriešiť menší podproblém, výpočet vzdialenosť korytnačky od stredu.

Pomery strán vlajok a ich jednotlivých častí sú presne dané. V tomto prípade nepovažujeme za potrebné dodržať presne definované rozmery. Jednotlivé pomery častí vlajok môžeme určiť približne, odhadom.

Úlohy na precvičenie

Úlohy sú zamerané na precvičovanie vytvárania podmienok a používania podmieneného príkazu. Tentoraz žiaci nič nekreslia, pohybujú virtuálnym zvieratkom podľa zadaných pravidiel. Formulácia tretej úlohy je zámerná. Jej riešenie je totiž identické s riešením prvej úlohy. Môžeme ju teda chápať aj ako úlohu proti stereotypu. Očakávame, že si to žiaci uvedomia a nebudú opätovne riešiť a zapisovať riešenie už raz vyriešeného problému.

V pripravenom súbore sú tlačidlá pre generovanie prostredia pre každú z troch situácií. Skúsenosť s ich použitím je propedeutikou k práci s grafickými komponentmi.

Overenie metodiky

Overovanie efektívnosti navrhnutej metodiky sme realizovali počas regulárnej výučby. Celkovo sme mali k dispozícii 4 skupiny žiakov z rôznych tried. V dvoch

skupinách sme počas vyučovania použili navrhnutú metodiku, dve skupiny slúžili ako kontrolné a učitelia v nich použili postup, na ktorí sú zvyknutí. Vo všetkých skupinách sme pred začiatkom overovania použili pretest na meranie úrovne algoritmického myslenia. Po ukončení testovania sme pre vyhodnotenie efektívnosti navrhnutej metodiky použili vo všetkých skupinách midtest. V postupe overovania efektívnosti navrhnutej metodiky sme použili kvantitatívne štatistické metódy pre testovanie nulových hypotéz. Ako prostriedok pre overenie hypotéz sme použili testy významnosti (F-test, t-test).

Na základe výsledkov predchádzajúcich testov konštatujeme, že použitím navrhnutej metodiky sme v experimentálnych skupinách štatisticky významne (na hladine významnosti $\alpha=0,01$) zvýšili úroveň algoritmického myslenia žiakov oproti úrovni algoritmického myslenia žiakov v kontrolných skupinách. Potvrdili sme, že navrhnutá metodika je efektívnejšia než spôsob vyučovania ktorý učitelia na uvedenej škole doteraz používali.

Záver

V našej práci sme sa zaoberali rozvojom algoritmického myslenia žiakov pri výučbe tematického okruhu „Postupy, riešenie problémov, algoritmické myslenie“. Analyzovali sme možné príčiny spôsobujúce problémy vyučovania tohto tematického okruhu. Špecifikovali sme predispozície žiakov, s ktorými vstupujú do procesu učenia sa algoritmov a programovať. Na algoritmické myslenie žiakov sme sa pozerali z nadhľadu a ukázali sme, že toto myslenie je dôležité nielen na hodinách informatiky a programovania, ale aj pre úspešné a aktívne zapojenie sa človeka do globálnej spoločnosti. Navrhli a overili sme vhodnú metodiku. V navrhnutej metodike kladieme dôraz na aktívne učenie sa žiakov prostredníctvom nachádzania riešení reálnych problémov.

Podrobne sme analyzovali význam pojmu algoritmické myslenie a navrhli sme model, ktorý nám umožnil s touto komplexnou schopnosťou lepšie manipulovať. Naznačili sme našu snahu o širšie prepojenie žiackych vedomostí a zručností zo školskej informatiky s ostatnými školskými predmetmi. Upozornili sme na vágnosť existujúcich vzdelávacích štandardov a z časti sme navrhli riešenie.

Projekt našej práce sme riešili v dvoch fázach.

Výsledkom fázy jeden nášho výskumu sú zistenia, ktoré približujú, s akými predispozíciami prichádzajú programátori, začiatočníci. Napriek tomu, že výsledky tohto kvalitatívneho výskumu nemožno zovšeobecniť na celú populáciu žiakov, programátorov začiatočníkov, domnievame sa, že pomerne dobre charakterizujú túto skupinu žiakov.

Na základe výsledkov fázy jeden, výsledkov analýz súčasnej situácie v oblasti vyučovania informatiky a našich niekoľkoročných skúsenosti sme definovali východiská pre fázu dva riešenia projektu. Vďaka spolupráci s učiteľmi z praxe sme vyvinuli metodiku vyučovania tematického okruhu „Postupy, riešenie problémov, algoritmické myslenie“. Cieľom vyučovania podľa nami navrhnutej metodiky nie je len samotné písanie programov – kódovanie. Snažili sme sa metodiku navrhnuť tak, aby žiakom programovanie dávalo zmysel a dobrú predstavu o jeho praktickom využití, rozvíjali sme algoritmické myslenie žiakov. Našou snahou bolo priblížiť žiakom programovanie ako účinný nástroj využiteľný pri riešení reálnych problémov. V procese žiakovho učenia sa kladieme dôraz na aktívny prístup žiaka, na analytickú

fázu pri riešení problémov a samotný proces vedúci k nájdeniu riešenia daného problému.

Metodiky sme vyvíjali počas reálnej výučby a pre reálnu výučbu. Tieto skutočnosti nám podstatne zúžili manévrovací priestor. Domnievame sa však, že práve vďaka tomu je možné túto metodicu (príp. po vhodných úpravách) implementovať do vyučovania aj na iných školách podobného zamerania.

Literatúra

- ANDREJKOVÁ, Gabriela. *Programovanie a algoritmy*. Košice: Univerzita Pavla Jozefa Šafárika v Košiciach, 2002. ISBN 80-7097-493-1.
- BELUŠOVÁ, Mária, Mário VARGA a Ružena ZIMANOVÁ. *Informatika pre stredné školy: Algoritmy s Pascalom*. Bratislava: SPN, 2002. ISBN 80-08-03289-8.
- BELL, Tim, Ian H. WITTEN a Mike FELLOWS. *Computer Science Unplugged: off-line activities and games for all ages*. New Zeland: Computer Science Unplugged, 1998.
- BLACKWELL, Alan F. What is Programming?. In: *14th Annual Workshop*. London, UK: Brunel University, 2002, s. 204-218. Dostupné z: <http://www.ppig.org/papers/14th-blackwell.pdf>
- BLAHO, Andrej. *Informatika pre stredné školy: Programovanie v Delphi*. Bratislava: SPN – Mladé letá, 2006. ISBN 80-10-00421-9.
- BLAHO, Andrej a Ivan KALAŠ. *Tvorivá informatika: 1. zošit z programovania*. Bratislava: SPN – Mladé letá, 2006. ISBN 80-10-00019-1.
- Committee on Logic Education. Algorithmic Thinking. *Committee on Logic Education of the Association for Symbolic Logic (ASL)* [online]. 2010 [cit. 2012-11-15]. Dostupné z: <http://www.ucalgary.ca/aslc/nctm/Q2A.html>
- CSTA K-12 Computer Science Standards. *Computer Science Teachers Association* [online]. 2005, 2013 [cit. 2013-03-08]. Dostupné z: <http://csta.acm.org/Curriculum/sub/K12Standards.html>
- CUNY, Jan, Larry SNYDER a Jeannette M. WING. What is computational thinking?. *Center for Computational Thinking, Carnegie Mellon* [online]. [cit. 2010-10-10]. Dostupné z: <http://www.cs.cmu.edu/~CompThink/>
- ĎURIČ, L. a J. ŠTEFANOVIČ J. *Psychológia pre učiteľov*. Bratislava: SPN, 1997. ISBN 67-027-77-I.
- EDMONDS, Jeff. *How to think about algorithms*. New York: Cambridge University Press, 2008, xiii, 448 p. ISBN 05-216-1410-4.
- FOREHAND, Mary. Bloom's Taxonomy. *Emerging Perspectives on Learning, Teaching and Technology* [online]. 2005 [cit. 2009-02-02]. Dostupné z: http://projects.coe.uga.edu/epltt/index.php?title=Bloom%27s_Taxonomy
- FUTSCHEK, Gerald. Algorithmic Thinking: The Key for Understanding Computer Science. In: MITTERMEIR, Roland. *Informatics education: the bridge between using and understanding computers : International Conference in Informatics in Secondary Schools - Evolution and Perspectives, ISSEP 2006, Vilnius, Lithuania, November 7-11, 2006 : proceedings*. New York: Springer, 2006, s. 156-168. ISBN 3-540-48218-0. DOI: 3-540-48218-0.

- FUTSCHEK, Gerald a Julia MOSCHITZ. Developing Algorithmic Thinking by Inventing and Playing Algorithms. In: *Constructionist approaches to creative learning, thinking and education*. Paris, 2010, s. 10. ISBN 978-80-89186-65-5.
- GALČÍK, F., GURSKÝ, P. *Programovanie, algoritmy, zložitosť 1a* [online]. 2013. vyd. 2013 [cit. 2013-02-14]. Dostupné z: <http://web.ics.upjs.sk/paz1a/>
- GARCIA, Byron. Algorithm March with Prisoners. *YouTube* [online]. 1.10.2006 [cit. 2013-03-05]. Dostupné z: <http://www.youtube.com/watch?v=WjMd2Vabcv8>
- GAVORA, Peter. *Spríevodca metodológiou kvalitatívneho výskumu*. Bratislava: Regent, spol. s r. o., 2006. ISBN 80-88904-46-3.
- KOBÁ, Grażyna. *Informatyka: dla liceum ogólnokształcącego*. Wrocław: Migra, 2003. ISBN 83-916-8486-5.
- GUNIŠ, Ján. Programovanie a jeho pozícia v predmete informatika. In: *Zborník DidInfo 2006: Informatika na slovenských školách, vývoj a perspektívy*. Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied, 2006a, s. 53-56. ISBN 80-8083-202-1.
- GUNIŠ, Ján.: Individuálny algoritmický jazyk ako prirodzená forma vyjadrovania sa študentov. In: *Zborník konferencie DidInfo 2007 Banská Bystrica* [CD ROM]. Univerzita Mateja Bela, Fakulta prírodných vied, Banská Bystrica. 2007. ISBN 978-80-8083-367-1.
- GUNIŠ, Ján: Rozvíjanie algoritmického myslenia pomocou tabuľkového kalkulátora. In: *Matematika Informatika Fyzika*. Prešov: MPC Prešov, 2006b, č. 29. s. 150-157. ISSN 1335-7794.
- GUZDIAL, Mark. Programming Environments for Novices. In: FINCHER, Sally a Marian PETRE. *Computer science education research*. New York: Taylor, 2004, s. 127-154. DOI: ISBN 90 265 1969 9.
- HAREL, Idit. Sand Castles Go Digital. *MaMaMedia* [online]. 2003 [cit. 2013-04-23]. Dostupné z: http://www.mamamedia.com/areas/grownups/new/21_learning/sand_castles.html
- HEJNÝ, Milan a František KUŘINA. *Dítě, škola a matematika: konstruktivistické přístupy k vyučování*. 2., aktualiz. vyd. Praha: Portál, 2009, 232 s. Pedagogická praxe. ISBN 978-807-3673-970.
- HROMKOVIČ, Juraj. *Algorithmic adventures: from knowledge to magic*. New York: Springer, 2009, xiii, 363 p. ISBN 978-3-540-85986-4.
- CHRÁSKA, Miroslav. *Metody pedagogického výzkumu: základy kvantitativního výzkumu*. Vydání 1. Praha: Grada Publishing, 2007, 265 s. ISBN 978-80-247-1369-4.
- JERŠOV, Andrej, P. Programovanie - druhá gramotnosť. In: *Pokroky matematiky, fyziky a astronomie*. Praha: Jednota českých matematiků a fyziků, 1982, s. 325-335. ISSN 0032-2423.
- JOHNSON, Colin G. a Ursula FULLER. Is Bloom's Taxonomy Appropriate for Computer Science?. In: *Proceedings of the 6th Baltic Sea conference on Computing education research*. New York, NY, USA: ACM, 2006, s. 120-123.
- KALAŠ, Ivan. Pedagogický výskum v informatike a informatizácii (1. časť). In: *DidInfo 2008*. Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied, 2008, 10 strán. ISBN 978-80-8083-556-9.

- KALAŠ, Ivan. Pedagogický výskum v informatike a informatizácii (2. časť). In: *DidInfo 2009*. Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied, 2008, s. 15-24. DOI: 978-80-8083-720-4.
- KALAŠ, Ivan. Informatika a informatizácia 2/3. In: *Ivan Kalaš* [online]. 2004 [cit. 2010-11-19]. Dostupné z: http://edi.fmph.uniba.sk/~kalas/Vyucba/Didaktika_informatiky/Informatika_a_informatizacia2.doc
- KALAŠ, Ivan. *Informatika pre stredné školy*. Bratislava: SPN – Mladé letá, s. r. o., 2005. ISBN 80-10-00762-5.
- KALAŠ, Ivan. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika: Základy pedagogického výskumu*. Bratislava: Štátny pedagogický ústav, 2011. ISBN 978-80-8118-082-8.
- KAPLAN, Randy M. Teaching Novice Programmers Programming Wisdom. In: *22nd Annual Workshop*. Madrid: University Carlos III of Madrid, 2010, 9 strán. Dostupné z: <http://www.ppig.org/papers/22nd-Teach-2.pdf>
- KORŠŇÁKOVÁ, Paulína, Jana KOVÁČOVÁ a Daniela HELDOVÁ. *Národná správa OECD PISA Sk 2009*. Bratislava: Národný ústav certifikovaných meraní vzdelávania, 2010. ISBN 978-80-970261-4-1.
- KOUBEK, V. 2.9 Psychológia ako súčasť prípravy učiteľa fyziky. In: *KATEDRA TEORETICKEJ FYZIKY A DIDAKTIKY FYZIKY* [online]. 2005 [cit. 2010-10-05]. Dostupné z: http://www.ddp.fmph.uniba.sk/~koubek/DF_html/2-9.htm
- (KUI) Klub učiteľov informatiky. PRÍRODOVEDECKÁ FAKULTA, Univerzita P. J. Šafárika v Košiciach. *Kluby učiteľov biológie, fyziky, geografie, chémie, informatiky a matematiky* [online]. 2006 [cit. 2011-02-14]. Dostupné z: <http://kluby.science.upjs.sk/course/view.php?id=8>
- LAHTINEN, Essi, Kirsti ALA-MUTKA a Hannu-Matti JÄRVINEN. A study of the difficulties of novice programmers. In: *ITiCSE 2005: proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, June 27-29, 2005, Monte de Caparica, Portugal*. New York, N.Y.: Association for Computing Machinery, c2005, s. 14-18. SIGCSE bulletin, v. 37, no. 3. DOI: 1-59593-024-8.
- MARKECHOVÁ, Dagmar, Anna TIRPÁKOVÁ a Beáta STEHLÍKOVÁ. *Základy štatistiky pre pedagógov*. Nitra: Fakulta prírodných vied UKF v Nitre, 2011. ISBN 978-80-8094-899-3. Dostupné z: http://www.km.fpv.ukf.sk/upload_publikacie/20120130_90405__1.pdf
- MCIVER, Linda, Damian CONWAY. Seven Deadly Sins of Introductory Programming Language Design. In: SPONSORED BY NEW ZEALAND COMPUTER SOCIETY, Australian Computer Society a Edited by Martin PURVIS. *1996 International Conference Software Engineering: Education: proceedings, January 24-27, 1996, Dunedin, New Zealand*. Los Alamitos, Calif: IEEE Computer Society Press, 1996, s. 309-316. ISBN 0-8186-7379-6.
- MIDDLETON, Steve. Salsa Dance Steps: Discover The Basics. SALSA-DANCERS.COM. *The Salsa Dancing Heart Of The Internet* [online]. 2013 [cit. 2013-03-05]. Dostupné z: <http://www.salsa-dancers.com/salsa-dance-steps.html>
- Michelin Prešov Nitra route planner. *ViaMichelin: Michelin route planner and maps*,

restaurants, traffic news and hotel booking [online]. 2010 [cit. 2013-03-05]. Dostupné z: <http://www.viamichelin.com/>

- (MŠ SR) Učebné osnovy gymnázia, štvorročné štúdium: Informatika. In: *Ministerstvo školstva Slovenskej republiky* [online]. 1997 [cit. 2010-10-20]. Dostupné z: http://www.statpedu.sk/documents//16/pedagogicke_dokumeny/gymnazium/ucebne_osno_vy/stvorrocne/UO_inf_4r_gym.pdf
- MOLNÁR, Ľudovít. *Programovanie pre 3. ročník gymnázia*. Bratislava: SPN, 1985. ISBN 9788008009447.
- MOLNÁR, Ľudovít a Miloš FRANEK. *Algoritmy pre 3. ročník gymnázia*. Bratislava: SPN, 1985. ISBN 9788008009430.
- (NR SR) ZÁKON z 22. mája 2008 o výchove a vzdelávaní (školský zákon) a o zmene a doplnení niektorých zákonov. In: <http://www.zbierka.sk/>. 2008. Dostupné z: <http://www.zbierka.sk/sk/predpisy/245-2008-z-z.p-32362.pdf>
- Oreo Commercial -- France. *YouTube* [online]. 27.05.2011 [cit. 2013-03-05]. Dostupné z: http://www.youtube.com/watch?v=rB85iQ0Y_bwMd2Vabcv8
- PAPERT, Seymour a Idit HAREL. Situating Constructionism. *Professor Seymour Papert* [online]. 1991, 2008 [cit. 2013-04-23]. Dostupné z: <http://www.papert.org/articles/SituatingConstructionism.html>
- (PJ) UPJŠ V KOŠICIACH, Prírodovedecká fakulta, Ústav informatiky. *PALMA junior: Programovanie, ALgoritmy, MAtematika, súťaž pre mladých programátorov v Imagine* [online]. 2005 [cit. 2013-03-05]. Dostupné z: <http://di.ics.upjs.sk/palmaj/>
- PECINOVSKÝ, R. Metodika výuky programovania na rozcestí. In: *Zborník konferencie POŠKOLE 2007*. Praha: Vydavateľství ČVUT Praha, 2007, s. 48-57. ISBN 978-80-239-9126-0.
- PETLÁK, Erich. *Kapitoly zo súčasnej didaktiky*. 1. vyd. Bratislava: Iris, 2005, 189 p. ISBN 80-890-1889-0.
- PIAČEK, Jozef a Miloš KRAVČÍK. *FILIT: Otvorená filozofická encyklopédia* [online]. 1999 [cit. 2010-10-12]. Dostupné z: <http://ii.fmph.uniba.sk/~kravcik/filit/>
- PŘÍHONSKÁ, Jana. METODY ŘEŠENÍ ÚLOH – MX2M. In: [online]. 2013 [cit. 2013-02-23]. Dostupné z: http://kmd.fp.tul.cz/lide/prihonska/MX2M/P1_METODY%20RESENI%20ULOH.pdf
- RAMBOUSEK, Vladimír. *Výzkum informační výchovy na základních školách*. Vyd. 1. Plzeň: Koniáš, 2007, 359 s. ISBN 978-80-86948-10-2.
- SALANCI, Ľubomír, Monika TOMCSÁNYIOVÁ a Andrej BLAHO. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika: Didaktika programovania*. Bratislava: Štátny pedagogický ústav, 2010a. ISBN 978-80-8118-065-1.
- SALANCI, Ľubomír, Monika TOMCSÁNYIOVÁ a Andrej BLAHO. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika: Didaktika programovania 2*. Bratislava: Štátny pedagogický ústav, 2010b. ISBN 978-80-8118-053-8.
- SHACKELFORD, Russell. L. a Albert N. BADRE. Why can't smart students solve simple programming problems?. In: *International Journal of Man-Machine StudiesInt*. London, UK: Academic Press Ltd., 1993, s. 985-997. Volume 38: Issue 6. ISSN 0020-7373J.

- SKORKOVÁ, Anna. *Pedagogické čítanie: Metodika vyučovania telesnej výchovy na stredných odborných školách so zameraním na atletiku*. Prešov: MPC Bratislava Regionálne pracovisko Prešov, 2009.
- SNYDER, L. Interview by F. Olsen "Computer Scientist Says all Students Should Learn to Think 'Algorithmically'". *The Chronicle of Higher Education* [online]. 2000 [cit. 2006-04-23]. Dostupné z: <http://chronicle.com/free/2000/03/2000032201t.htm/>
- ŠNAJDER, Ľubomír. Vykonať (procesory) algoritmov. In: *Matematika Informatika Fyzika*. Prešov: Metodicko-pedagogické centrum Prešov, 2003, s. 35-41. ISSN 1335-7794.
- ŠNAJDER, Ľubomír a Ján GUNIŠ. *Didaktický projekt: informatika - jednorozmerné pole* [online]. 2009a [cit. 2013-03-26]. Dostupné z: http://di.ics.upjs.sk/vyucba/pomocne_materialy/dp/
- ŠNAJDER, Ľubomír a Ján GUNIŠ. Úvaha o dimenziách algoritmického myslenia a jeho rozvíjaní. In: *Zborník konferencie DidInfo 2009 Banská Bystrica*. Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied, 2009b, s. 5. ISBN 978-80-8083-720-4.
- ŠNAJDER, Ľubomír a Ján GUNIŠ. The model of algorithmic thinking: dimensions and levels. In: *Information and Communication Technology in Education: proceedings, Rožnov pod Radhoštěm*. Ostrava: University of Ostrava, 2012, s. 69-78. ISBN 9788074641350.
- ŠNAJDER, Ľubomír, Ján GUNIŠ a Valentína GUNIŠOVÁ. Rozvíjanie algoritmického myslenia prostredníctvom aktívneho a zážitkového učenia sa. In: *Zborník Medzinárodnej vedecko-odbornej konferencie Záujmová činnosť žiakov: stav, problémy, trendy*. Prešov: FHPV PU v Prešove, 2009c, s. 5. ISBN 9788074641350.
- (ŠPÚ) Štátny pedagogický ústav. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika* [online]. 2008a [cit. 2012-04-04]. Dostupné z: <http://dvui.ccv.upjs.sk/>
- (ŠPÚ) Štátny pedagogický ústav: *Štátne vzdelávacie programy* [online]. 2008b [cit. 2012-04-04]. Dostupné z: <http://www.statpedu.sk/sk/Statny-vzdelavaci-program.alej>
- (ŠPÚ) Štátny pedagogický ústav: *Štátny vzdelávací program - príloha 3 - ISCED 3* [online]. 2008c [cit. 2012-04-04]. Dostupné z: <http://www.statpedu.sk/sk/Statny-vzdelavaci-program/Statny-vzdelavaci-program-pre-gymnaziaISCED-3a.alej>
- ŠVARŤÍČEK, Roman a Klára ŠEĎOVÁ. *Kvalitatívny výzkum v pedagogických vedách*. Vyd. 1. Praha: Portál, 2007, 377 s. ISBN 978-80-7367-313-0.
- TRNA, Josef. Konstrukčný výzkum (design-based research) v prírodovedných didaktikách. In: *Scientia in educatione*. Praha: Univerzita Karlova v Praze, Pedagogická fakulta, 2011, s. 3-14. ISSN 1804-7106. Dostupné z: <http://www.scied.cz/FileDownload.aspx?FileID=415>
- TUREK, Ivan. *Didaktika*. 1. vyd. Bratislava: Iura Edition, 2008, 595 s. Ekonomia. ISBN 978-808-0781-989.
- VANÍČEK, Jiří. Programování jako spojovací článek kurikula výuky ICT. In: *Zborník konferencie POŠKOLE 2007*. Praha: ČVUT Praha, 2007, s. 58-63. ISBN 978-80-239-9126-0.
- VARGA, Mário, Andrej BLAHO a Ružena ZIMANOVÁ. *Informatika pre gymnáziá: Algoritmy s Logom*. Bratislava: SPN – Mladé letá, s. r. o., 1999. ISBN 80-08-02965-X.

- WING, Jeannette M. Computational thinking. In: *Magazine Communications of the ACM* [online]. 2006 [cit. 2010-10-01]. Dostupné z: <http://cacm.acm.org/magazines/2006/3/5977-computational-thinking/fulltext>
- WING, Jeannette M. Computational thinking. In: *Journal of Computing Sciences in Colleges*. USA: Consortium for Computing Sciences in Colleges, 2009, s. 2. ISSN 1937-4771. Dostupné z: http://delivery.acm.org/10.1145/1530000/1529997/p6-wing.pdf?ip=158.197.36.243&acc=PUBLIC&key=C2716FEBFA981EF18057A5366E2A8AD988BCC6627890B960&CFID=200903775&CFTOKEN=72523660&__acm__=1365148703_b6225516d5ec2625f9c538bb65626026
- WING, Jeannette. Computational Thinking. In: *OurCS Workshop* [online]. 2011 [cit. 2013-03-08]. Dostupné z: <http://csta.acm.org/Curriculum/sub/CurrFiles/WingCTPrez.pdf>
- WRÓBLEWSKI, Piotr. *Algoritmy: Datové struktury a programovací techniky*. 1. vyd. Překlad Marek Michalek, Bogdan Kiszka. Brno: Computer Press, 2004, 351 s. ISBN 80-251-0343-9.

Vlastná publikačná činnosť súvisiaca s prácou

- GUNIŠ, Ján. Programovanie a jeho pozícia v predmete informatika. In: *Zborník DidInfo 2006: Informatika na slovenských školách, vývoj a perspektívy*. Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied, 2006a, s. 53-56. ISBN 80-8083-202-1.
- GUNIŠ, Ján.: Individuálny algoritmickej jazyk ako prirodzená forma vyjadrovania sa študentov. In: *Zborník konferencie DidInfo 2007 Banská Bystrica* [CD ROM]. Univerzita Mateja Bela, Fakulta prírodných vied, Banská Bystrica. 2007. ISBN 978-80-8083-367-1.
- GUNIŠ, Ján: Rozvíjanie algoritmickeho myslenia pomocou tabuľkového kalkulátora. In: *Matematika Informatika Fyzika*. Prešov: MPC Prešov, 2006b, č. 29. s. 150-157. ISSN 1335-7794.
- ŠNAJDER, Lubomír a Ján GUNIŠ. Úvaha o dimenziách algoritmickeho myslenia a jeho rozvíjaní. In: *Zborník konferencie DidInfo 2009 Banská Bystrica*. Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied, 2009b, s. 5. ISBN 978-80-8083-720-4.
- ŠNAJDER, Lubomír a Ján GUNIŠ. The model of algorithmic thinking: dimensions and levels. In: *Information and Communication Technology in Education: proceedings, Rožnov pod Radhoštěm*. Ostrava: University of Ostrava, 2012, s. 69-78. ISBN 9788074641350.
- ŠNAJDER, Lubomír, Ján GUNIŠ a Valentína GUNIŠOVÁ. Rozvíjanie algoritmickeho myslenia prostredníctvom aktívneho a zážitkového učenia sa. In: *Zborník Medzinárodnej vedecko-odbornej konferencie Záujmová činnosť žiakov: stav, problémy, trendy*. Prešov: FHPV PU v Prešove, 2009c, s. 5. ISBN 9788074641350.
- LUBOMÍR, Šnajder, Ján GUNIŠ a Valentína GUNIŠOVÁ. Methodology design of algorithm development teaching based on content analysis of pupils' solutions. In: *The 3rd International Conference - ISSEP 2008 Informatics in Secondary Schools: Evolution and Perspective*. Torun, Poľsko: Faculty of Mathematics and Computer Science, Nicolaus Copernicus University, 2008, s. 20-29. ISBN 978-83-60425-31-2.
- GUNIŠ, Ján, Miloslava SUDOLSKÁ a Lubomír ŠNAJDER. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika: Aktivizujúce metódy vo výučbe školskej informatiky*. Bratislava: Štrátny pedagogický ústav, 2009. ISBN 978-80-89225-

- GUNIŠ, Ján a Ľubomír ŠNAJDER. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika: Tvorba úloh a hodnotenie žiakov v predmete informatika*. Bratislava: Štrátny pedagogický ústav, 2009. ISBN 978-80-8118-012-5.

Summary

In our work, we discussed the development of algorithmical thinking of pupils in teaching theme "Procedures, problem solving, algorithmical thinking". We analyzed the possible reasons causing the problems of teaching the subject matter. We specified the pupils' predisposition with which they enter into the learning process of algorithmization and programming. We were looking from above on the pupils' algorithmical thinking and we have shown that this thinking is important not only for lessons of informatics and programming, but also for successful and active involvement of human in a global society. We proposed and verified a suitable methodology. We focused on active learning of pupils by finding solutions to real problems in the proposed methodology.

We analyzed in detail the meaning of algorithmical thinking and we proposed a model that allows us to manipulate better with this a comprehensive ability. We indicated our efforts to extend the linking of pupils' knowledge and skills in school informatics with other school subjects. We pointed to the vagueness of the existing educational standards and we propose a solution in part.

We solved the project of our work in two phases.

The result of phase one of our research are the findings that explain predispositions of novice programmers. Although the results of qualitative research cannot be generalized to the entire population of pupils, novice programmers, we believe that they relatively well-characterized this group of pupils.

Based on the results of phase one, the results of analyzes of the current situation of informatics education and our years of experience, we defined starting points for phase two of the project. Thanks to cooperation with teachers from practice, we developed a methodology for teaching of theme "Procedures, problem solving, algorithmical thinking". The aim of teaching by methodology presented by us is not just writing programs - coding. We tried to propose a methodology such a way that programming made sense and a good idea of its practical application for pupils; we developed algorithmical thinking of pupils. Our aim was to bring programming to pupils as an effective tool useful for solving real problems. We place emphasis on proactive learner, to an analytical phase in solving problems and the process leading to the solution of the problem in the process of pupil learning.

We developed a methodology during the real teaching and learning for real. These facts significantly narrow down scope for maneuver. We believe, however, that just because of the fact it is possible to implement this methodology (or after suitable adjustments) to teaching at other schools of similar focus.