



UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA ZÁKLADOV A VYUČOVANIA INFORMATIKY
Evidenčné číslo: 64d0458c-d402-4dfb-a863-5b1f1fb95eeb



Metodika prípravy na maturitu z informatiky

Dizertačná práca

Mgr. Juliana Šišková

9.2.3 Teória vyučovania informatiky

školiteľ: prof. RNDr. Ivan Kalaš, PhD.

BRATISLAVA 2012



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Mgr. Juliana Šišková
Študijný program: teória vyučovania informatiky (Jednoodborové štúdium,
doktorandské III. st., denná forma)
Študijný odbor: 9.2.3. teória vyučovania informatiky
Typ záverečnej práce: dizertačná
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Metodika prípravy na maturitu z informatiky

Školiteľ: prof. RNDr. Ivan Kalaš, CSc.
Katedra: FMFI.KZVI - Katedra základov a vyučovania informatiky

Spôsob sprístupnenia elektronickej verzie práce:
bez obmedzenia

Dátum zadania: 27.10.2010

Dátum schválenia: 27.10.2010

prof. RNDr. Ivan Kalaš, CSc.
garant študijného programu

.....
študent

.....
školiteľ

Čestne prehlasujem, že túto prácu som vypracovala samostatne, len s použitím uvedenej literatúry, elektronických zdrojov a s odbornou pomocou školiteľa.

Bratislava, jún 2012

Juliana Šišková

Ďakujem svojmu školiteľovi Ivanovi Kalašovi a kolegom za veľa poznámok pri písaní práce, pomoc s výskumom a motivačným prostredím. Ďakujem nemenovanej učiteľke, vďaka ktorej som mohla výskum realizovať a od ktorej som sa veľa naučila. Ďakujem svojmu manželovi Jozefovi Šiškovi, rodine, Monike Steinovej, Michalovi Forišekovi a ďalším priateľom za pomoc nielen v stresových situáciách. Ďakujem univerzite za udelenie grantu mladých, z ktorého bolo možné zaplatiť potrebné pomôcky k pedagogickému výskumu.

Obsah

Abstrakt	3
Abstract	4
1 Úvod	5
2 Výskumný projekt dizertačnej práce	7
2.1 Výskumný problém	7
2.1.1 Ciele projektu	8
2.1.2 Výskumné otázky	8
2.2 Návrh a metódy riešenia projektu	9
2.2.1 Výber výskumnej stratégie a metodologickej preferencie	9
2.2.2 Výskumné metódy a práca s výstupmi	11
2.2.3 Zabezpečenie kvality výskumu	11
2.2.4 Etika a výskum	12
3 História vyučovania informatiky na Slovensku	13
3.1 Algoritmizácia a programovanie	13
3.2 Používateľská informatika	15
3.3 Modernizovaná informatika	16
4 Vyučovanie informatiky v iných krajinách	18
4.1 Definícia pojmov	18
4.2 ACM model	19
4.3 Rôzne pohľady na informatiku	22

<i>OBSAH</i>	2
4.4 Výučba informatiky v jednotlivých krajinách	23
5 Aktuálny stav vyučovania informatiky na Slovensku	34
5.1 Časová dotácia informatiky	34
5.2 Obsahová náplň informatiky	35
5.2.1 Učebnice	40
5.2.2 Tematický výchovno-vzdelávací plán	40
6 Vzdelávacie ciele a poznávací proces	43
6.1 Taxonómie vzdelávacích cieľov	43
6.2 Etapy poznávacieho procesu	45
7 Priebeh výskumu	46
7.1 Pozorovanie a výučba v rámci mimoškolského prípravného kurzu na maturitu z informatiky	48
7.2 Interview s učiteľmi informatiky	49
7.3 Pozorovanie na seminári z informatiky	51
7.4 Pozorovanie a výučba seminára z informatiky	52
7.4.1 Návrh, implementácia a analýza časovo-tematického plánu	53
7.4.2 Návrh, implementácia a analýza vybraných tém	71
7.4.3 Metodické postrehy a skúsenosti s ich implementáciou	87
8 Výsledky výskumu	95
9 Záver	97
Vlastné publikácie	98
Literatúra	100
A Prepis rozhovoru č. 1	108
B Vedomostno-postojový dotazník	119
C Médium s kópiou webového portálu	120

Abstrakt

Autor:	Mgr. Juliana Šišková
Názov práce:	Metodika prípravy na maturitu z informatiky
Názov školy, fakulty a katedry:	Univerzita Komenského Fakulta matematiky, fyziky a informatiky Katedra základov a vyučovania informatiky
Meno školiteľa:	prof. RNDr. Ivan Kalaš, PhD.
Komisia pre obhajoby:	9.2.3 Teória vyučovania informatiky
Miesto a rok:	Bratislava, 2012
Rozsah práce:	120 s.
Stupeň odbornej kvalifikácie:	PhD.

Cieľom nášho výskumu bolo vyvinúť metodiku pre prvú časť prípravy na maturitu z informatiky, ktorá by vychádzala z reálnych skúseností. Pre dosiahnutie cieľa sme použili metódy kvalitatívneho výskumu a preferovali sme pravidlá výskumu vývojom (angl. design-based research). Vykonali sme niekoľko iterácií výskumu, zbierali sme poznatky pomocou výučby a pozorovania, rozhovorov s učiteľmi a analýzy produktov. Tie sme zakomponovali do metodiky, ktorej výsledkom je časovo-tematický plán, metodika vybraných tém, metodické postrehy z hodín a súbor aktivít pre náš profil študentov. Veľa vzniknutých materiálov je zovšeobeciteľných, čo môže významne pomôcť učiteľom pri príprave takéhoto kurzu.

Kľúčové slová: didaktika informatiky, didaktika programovania, maturita z informatiky, vyššie sekundárne vzdelávanie, ISCED 3

Abstract

Author:	Mgr. Juliana Šišková
Title of thesis:	Methodology of preparation for maturity exam in informatics
University:	Comenius University Faculty of mathematics, physics and informatics Department of informatics education
Supervisor:	prof. RNDr. Ivan Kalaš, PhD.
Committee:	9.2.3 Theory of informatics education
Place and year:	Bratislava, 2012
Volume of thesis:	120 p.
Degree of qualification:	PhD.

The aim of our research was to develop a methodology of the first part of preparation for the maturity exam in informatics, which is based on our teaching experience. To reach our goals, we used methods of qualitative research with a strong preference for design-based research. We executed several iterations of our research and collected results through teaching, observation, discussions with teachers and analysis of products. Afterwards we incorporated them into a methodology, which resulted in a topic schedule, methodology of selected topics, remarks on actual lessons and a collection of activities for students. A large part of the created materials can be easily generalized, which can help teachers create courses for students on other knowledge levels.

Klíčové slová: informatics education, programming education, maturity exams in informatics, upper secondary education, ISCED 3

Kapitola 1

Úvod

Vyučovanie informatiky ako všeobecno vzdelávacieho predmetu na Slovensku je ešte stále vo fáze vývoja. Chýbajú metodické pomôcky, obsahová náplň sa neustále mení. Tento problém neregistrujeme iba na Slovensku. Registrujú ho odborníci vo veľa krajinách, o čom svedčí množstvo publikácií a rôznych koncepcií.

V našej práci sa zameriame na prvú časť prípravy na maturitu. Pre voliteľnú informatiku nie sú známe platné osnovy, pre učiteľov existuje iba veľmi málo metodických pomôcok. My sa budeme snažiť dokumenty vytvoriť kvalifikovane využitím pravidiel pedagogického výskumu. Tie síce budú určené špeciálne pre študentov spĺňajúcich istý profil, ale učitelia ich budú môcť prispôbiť aj pre svojich študentov, alebo sa nimi budú môcť aspoň inšpirovať. Dôležité je tiež poznamenať, že sa zameriame na štvorročné alebo osemročné gymnáziá. Ak bude niekto tvoriť metodiku pre stredné odborné školy, bude môcť vychádzať aj z našej práce.

Súčasťou práce je aj súbor aktivít, ktoré môže učiteľ využiť na svojich hodinách. Metodika je reakciou aj na meniace sa cieľové požiadavky na vedomosti a zručnosti maturantov z informatiky, pozri [Špú08], [Špú10]. Vychádzať ale budeme hlavne z cieľových požiadaviek [Špú08], pretože nové cieľové požiadavky boli platné až ku koncu výskumu.

Do metodiky sme zapracovali skúsenosti učiteľov, odborníkov a aj svetový trend vo vyučovaní informatiky. Ďalším zdrojom poznatkov boli autorkine skúsenosti z práce s na-

danými žiakmi v oblasti informatiky, napr. [Lip09, SŠ10] a predchádzajúceho vyučovania semináru z informatiky na gymnáziu, projektu prípravy študentov na maturitu z informatiky a prijímacie pohovory na vysoké školy a Ďalšieho vzdelávania učiteľov v predmete informatika a hlavne z aplikácie jednotlivých iterácií metodík na strednej škole.

V kapitole 2 prezentujeme výskumný projekt, jeho tému, ciele a otázky, zvolíme výskumnú metodológiu, stratégiu a metódy, ktoré použijeme.

V kapitole 3 prezentujeme vývoj informatiky z historického hľadiska, čo nám pomôže inšpirovať sa pozitívami a vyvarovať sa chybám z minulosti. V kapitole 4 sa budeme venovať vyučovaniu informatiky vo svete, zameriame sa na štúdium, ktoré je prípravou na záverečné skúšky a štúdium na vysokej škole. V kapitole 5 predložíme oficiálne pedagogické dokumenty viažúce sa k téme nášho výskumu a ich dôsledky. V kapitole 6 predstavíme taxonómie vzdelávacích cieľov a etapy poznávacieho procesu, ktoré budeme potrebovať pri tvorbe vzdelávacích cieľov a súboru aktivít v metodike.

Kapitola 7 je kľúčová. Popisujeme celý priebeh výskumu, použité metódy a ich výsledky. Budeme analyzovať, aký obsah by mal pokrývať predmet, ktorý je predmetom výskumu, zistíme reálne výkony študentov, vytvoríme časovo-tematický plán, metodiku vybraných tém a ďalšie postrehy, ktoré môžu pomôcť učiteľovi pri vyučovaní informatiky.

V kapitole 8 prezentujeme výsledky projektu dizertačnej práce.

Kapitola 2

Výskumný projekt dizertačnej práce

V nasledujúcej kapitole prezentujeme náš výskumný projekt dizertačnej práce.

V prvej časti špecifikujeme výskumný problém, definujeme ciele projektu a výskumné otázky.

V druhej časti popíšeme riešenie projektu. Ten bude zahŕňať výber metodológie, výskumnej stratégie a metód, ktoré použijeme. Uvedieme tiež spôsoby na zabezpečenie kvality tohoto výskumu a dodržiavanie etiky.

2.1 Výskumný problém

Informatika na školách je stále vo fáze vývoja, v súčasnosti sa zostavujú maturitné štandardy a neexistujú osnovy informatiky ako všeobecno-vzdelávacieho predmetu v posledných ročníkoch strednej školy. Týmto projektom sme sa snažili vytvoriť metodiku prvého ročníka prípravy na maturitu z informatiky, aby spĺňala požiadavky, ktoré kvalifikovane nastavíme.

Výskumný problém: Tvorba metodiky prípravy na maturitu z informatiky

Výskumná oblasť: Vyučovanie informatiky v príprave na maturitu

2.1.1 Ciele projektu

Cieľom dizertačného výskumu bolo vytvoriť metodiku pre vyučovanie informatiky ako všeobecno-odborného predmetu v príprave na maturitnú skúšku. Z dôvodu rozsahu práce sme výskum smerovali iba pre prvý ročník prípravy na maturitnú skúšku

Metodika, ktorú sme vytvorili, zahŕňa:

- časovo-tematický plán,
- vstupné požiadavky a výstupné štandardy,
- cieľové požiadavky vybraných tém,
- metodické pomôcky, postrehy, ukážky hodín s vybranými cieľmi a ďalšie didaktické aspekty.

Výslednou metodikou sme sa pokúsili zohľadniť všetky didaktické zásady modernej pedagogiky, náročnosť tém, požiadavky vysokých škôl a cieľové požiadavky na maturitu z informatiky.

2.1.2 Výskumné otázky

Vo výskumnom projekte sme stanovili nasledujúce otázky:

- Ako špecifikovať vstupné požiadavky a výstupné štandardy tak, aby študenti spĺňajúci tieto požiadavky dokázali úspešne absolvovať maturitnú skúšku a mali informatické kompetencie potrebné pre štúdium na vysokej škole?
- Aký časovo-tematický plán zvolíť, aby študenti spĺňali výstupné štandardy?
- Aké metodické postupy a prostriedky použiť, aby sme dosiahli špecifikované ciele a čiastkové ciele?

V našom výskume sme sa primárne zamerali na prvú časť prípravy na maturitu z informatiky. Riešenia pre druhú časť iba načrtujeme, pretože tento projekt je široký a riešiť ho budeme detailne.

2.2 Návrh a metódy riešenia projektu

V časti návrh a metódy riešenia projektu prezentujeme, ako sme projekt dizertačnej práce riešili, prečo sme vybrali nižšie spomínanú metodológiu a stratégiu. Tiež uvedieme, ako sme zabezpečili kvalitu výskumu a ako sme dbali na etiku vo výskume.

2.2.1 Výber výskumnej stratégie a metodologickej preferencie

V pedagogickom výskume si výskumník väčšinou vyberá jednu z dvoch základných metodológií výskumu: kvalitatívnu [Gav07] alebo kvantitatívnu [Chr07]. Pre náš projekt som sa rozhodla použiť kvalitatívnu metodológiu.

Metodológia: kvalitatívna

Keďže naším cieľom bolo koncepciu *vytvoriť* a kvalifikovane odôvodniť, potrebovali sme sa dostať do hĺbky problematiky, čo je jedna z charakteristík kvalitatívneho výskumu. V kvantitatívnom výskume by bolo treba narábať s už existujúcou metodikou, čo sa nedá. Do hĺbky sme potrebovali ísť preto, aby sme so žiakmi interagovali, spoznávali chyby, ktoré robia, výnimočnosti, ktorými disponujú, rýchlo na ne reagovali, pretože ich čaká dôležitá skúška v živote – maturita.

Ďalším dôvodom pre výber kvalitatívnej metodológie boli existujúce možnosti. Mali sme príležitosť každým rokom učiť jednu alebo dve triedy, čo je spolu približne 20 žiakov. Toto číslo je vhodné pre kvalitatívny výskum. V kvantitatívnom výskume sa pracuje s väčšou vzorkou žiakov, čo by bolo časovo neúnosné.

Metodologická preferencia: výskum vývojom

Bežne známe výskumné stratégie (napríklad analytická indukcia [Gav07]) pracujú s už existujúcou realitou [Ber09], málokedy je vedec v pozícii nielen výskumníka, ale aj vývojára. Museli sme preto nájsť stratégiu vhodnú špeciálne pre našu potrebu, keďže sme chceli vyvíjať nielen teóriu, ale aj produkt, na ktorý sa teória viaže. Takouto stratégiou je napríklad postup špecifický pre výskum vývojom (angl. design-based research) [Kal09, Lab]. Práve výskum vývojom sme použili pre náš projekt.

Výskumná stratégia: postup špecifický pre výskum vývojom

Výskumná stratégia pre výskum vývojom sa skladá z niekoľkých etáp, pozri [Kal09]¹:

Etapa 0 – orientačná

Táto etapa zahŕňa štúdium literatúry, myšlienkové experimenty s typickými aktérmi, prezentovanie prototypov kolegom a diskusie o nich, pilotné pokusy s úvodnými vyvíjajúcimi sa predstavami o vyvíjanej intervencii s malou skupinkou študentov.

Etapa 1 – prieskumná

Prieskumná etapa predstavuje iniciálny vývoj. Výskumník kladie dôraz na základné používanie vyvíjanej intervencie a sleduje intuitívne indikátory jej potenciálu na podporu poznávacieho procesu.

Etapa 2 – vývojová

Vyvíjaná intervencia dosiahla stav, ktorý už výskumníkovi dovoľuje zamerať svoju pozornosť na otázky učenia sa a konkrétne ciele a výskumné otázky v problémovej doméne.

Etapa 3 – analytická

Vo vývoji intervencie už výskumník robí iba minimálne úpravy, sústreďuje sa na otázky zberu údajov, ich analýzy a dotvárania teórie, ktorú chce na záver svojho výskumu formulovať.

Pri stratégii výskumu vývojom prebehne niekoľko iterácií, ktoré sa skladajú z dvoch fáz:

1. **návrh, vývoj, nasadenie**
2. **pozorovanie, analýza**

V našom výskume sme začali orientačnou etapou, po jej skončení sme iterovali vyššie spomínané dve fázy, až sme postupne prešli prieskumnou, vývojovou a analytickou etapou. Po poslednej etape sme získali výsledok vo forme produktu a teórie viazanej k produktu.

¹popis etáp je upravený potrebu nášho projektu

2.2.2 Výskumné metódy a práca s výstupmi

Na uskutočnenie projektu sme použili niekoľko metód viazaných na kvalitatívny výskum: participačné pozorovanie, interview a obsahovú analýzu produktov.

Participačné pozorovanie

Participačné pozorovanie sme použili v orientačnej (nulte) etape na spoznanie prostredia a reakcií žiakov na rôzne typy a obsah výučby. Tiež sme ho použili v neskorších etapách, keď sme metodiku nasadili. Vtedy bola naša rola nielen výskumník, ale aj učiteľ.

Na zber údajov sme použili metódu terénnych zápiskov, uchovali sme ich na papieri, prípadne v počítači.

Interview

V orientačnej etape sme použili *pološtruktúrované interview* na spoznanie aktuálneho stavu informatiky na Slovensku, účastníkmi boli učitelia z rôznych krajov Slovenska. Druhým cieľom rozhovoru bolo získať informácie o skúsenostiach učiteľov s jednotlivými témami.

Na zber údajov sme použili diktafón. Rozhovory sme uchovali v počítači v zvukovej podobe, po prepise do počítača sme zvukové záznamy zničili.

Obsahová analýza produktov

V projekte sme analyzovali testy (nami zadané, alebo monitory) a odovzdané programy. To nám pomohlo vo fáze analýzy.

2.2.3 Zabezpečenie kvality výskumu

Kvalitu výskumu sme zabezpečili trianguláciou a auditom výskumu.

trianguláciou

Použili sme viaceré metódy zberu údajov: pozorovanie, interview a obsahovú analýzu produktov.

auditom výskumu

Správu o výskume publikujeme v dizertačnej práci, výsledky výskumu sme priebežne publikovali na národných a medzinárodných konferenciách. V dizertačnej práci sme dodržali pravidlá pre dôveryhodnosť výskumu [Gav07].

2.2.4 Etika a výskum

Pri realizácii výskumu sme dodržali nasledujúce etické zásady [Gav07].

Zachovanie dôvernosti informácií

Skúmané osoby majú utajenú identitu. Žiaci mali vlastný kód, ktorý vedia iba oni, výskumník a učiteľ. V rozhovore s učiteľmi sprístupňujeme iba typ školy a kraj, čo sú údaje potrebné pre výskum.

Ochrana súkromia

Účastníci výskumu (napr. učitelia v interview) si vybrali miesto, kde sa výskum odohrával. Toto nebolo možné spraviť pri zúčastnenom pozorovaní, tam sme museli skúmať participantov v domácom prostredí, t.j. v triede.

Súhlas participanta

Interview, ktoré sme uskutočnili, začína formulkou prevzatou z [ŠŠ⁺07] : "Súhlasíš s tým, že toto interview bude nahrávané?". Pred rozhovorom sme učiteľovi vysvetlili, ako bude interview použité, súčasťou je aj poučenie o autorizovaní rozhovoru.

V pozorovaniach sme poučili študentov o postupe, jeho právach na vyňatie vyjadrení zo záverečnej správy a o naložení so získanými údajmi.

Kapitola 3

História vyučovania informatiky na Slovensku

Vyučovanie informatiky na Slovensku prešlo niekoľkými etapami a naďalej si hľadá svoju podobu. Podľa [Kal01] môžeme v minulosti identifikovať tri koncepcie.

V prvej etape mohli študovať informatiku iba vybraní študenti na vybraných gymnáziách a stredných školách s elektrotechnickým zameraním. Nosnou témou informatiky bolo programovanie a počítačové systémy.

Druhá éra bola iniciovaná príchodom počítačov do domácností. Ľudia získali potrebu naučiť sa pracovať s počítačmi. Na stredných školách sa študenti začali učiť pracovať s počítačom a vybranými softvérovými nástrojmi. V tomto čase sa informatika učila na stredných školách, na základných školách iba výnimočne.

V ostatných rokoch sme svedkami toho, že informatika sa dostala ako povinný predmet aj na druhý stupeň základných škôl. Na prvom stupni základných škôl sa v školskom roku 2008/2009 začal učiť nový povinný predmet informatická výchova.

3.1 Algoritmizácia a programovanie

„Programovanie, druhá gramotnosť“

Prvá koncepcia informatiky na školách bola postavená na algoritmizácii a programovaní. Jej heslom bolo „Programovanie, druhá gramotnosť“ [Jer82]. Bola špecifická pre 70., 80. roky a začiatok 90. rokov dvadsiateho storočia. V skorších rokoch sa vyučovala iba na niekoľkých gymnáziách na Slovensku a to ako nepovinný predmet. Neskôr začal počet týchto gymnázií narastať.

V tom čase boli počítače iba na význačných miestach, väčšinou sa študent dostal k počítaču iba v niektorých školských výpočtových laboratóriách, výskumných ústavoch a iných špecializovaných inštitúciách. Keďže študenti väčšinu času nemali priamy prístup k počítačom, programovali na papieri. Ako programovací jazyk používali Fortran, Cobol, vývojové diagramy, neskôr Pascal a keď prišli 8-bitové počítače typu Sinclair, ATARI a podobne, študenti mohli programovať aj v programovacom jazyku BASIC.

Učebnice

K tomuto obdobiu sa viažu štyri učebnice informatiky, prvé dve boli experimentálne a používali sa iba na niektorých gymnáziách. Druhé dve boli už štandardné učebnice, určené pre prvý ročník gymnázií.

Učebnica *Algoritmy pre 3. ročník gymnázia* [GF79] sa skladá nielen z učebného textu, ale aj z množstva príkladov na precvičenie učiva. V prvej časti sa vysvetľuje pojem algoritmus, jeho potreba a príklady. Niekoľko strán autori venujú správnosti a efektívnosti algoritmov. Druhá časť učebnice je zameraná na programovanie. Zaujímavosťou je, že v úvode autori zaviedli pojem FIP = fiktívny počítač. Následne všetky programy boli realizované na tomto fiktívnom počítači s komentárom, že na ľubovoľnom inom počítači sa dajú realizovať. Na záver učebnice sa študenti dozvedeli (a tiež si vyskúšali) rôzne algoritmy triedenia. V tom čase bolo triedenie základná úloha, na ktorej sa vysvetľovali rôzne algoritmy a hlavne rôzne algoritmické metódy.

Rovnako ako predchádzajúca učebnica, aj *Počítačové systémy pre 4. ročník gymnázia* [FŠ82] má podnadpis Experimentálny učebný text pre voliteľnú skupinu odborných predmetov zameranú na programovanie a je tvorená z učebného textu a úloh. Kým pri al-

goritmoch si študenti mohli za pomoci učebnice precvičovať priamo tvorbu algoritmov, tu od nich autori nemohli žiadať zostavovať počítače. Preto úlohy v tejto učebnici sú buď kontrolné otázky alebo, ak to téma dovoľovala, úlohy na simuláciu (operácií, prácu v dvojkovej sústave a pod.). Učebnica prechádza najprv základnými pojmami a organizáciou počítača a následne autori počítač predstavujú v rôznych úrovniach: štruktúry a inštrukcie počítača, mikropočítač a počítač s operačným systémom. Pri takomto podrobnom diele nechýba ani v tom čase samozrejماً vec – zoznam inštrukcií, ak by mal študent príležitosť niekedy programovať na reálnom počítači.

Na prelome osemdesiatych a deväťdesiatych rokov vyšla dvojica učebníc určených pre 1. ročník SŠ: *Informatika a výpočtová technika – Algoritmy* [HG88] a *Informatika a výpočtová technika – Programovanie v jazyku Pascal* [KMH91]. V učebnici o algoritmoch nájdeme vysvetlený pojem algoritmus, dozvieme sa o jeho vlastnostiach, pochopíme, že my sme tí, ktorí musia počítaču povedať, čo má robiť, spoznáme praktické využitie programovania. V učebnici sú detailne vysvetlené pojmy procedurálneho jazyka. Algoritmy sa precvičujú v pseudojazyku, aby žiaci neboli zaťažovaní programovacím jazykom a mohli písať príkazy v slovenskom jazyku. V učebnici Pascalu sa potom riešia veľmi podobné úlohy, avšak zapisujú sa v jazyku Pascal.

Pri preštudovaní všetkých týchto učebníc človek pochopí, že slogan „programovanie ako druhá gramotnosť“ túto éru vystihuje. Do učebníc bola zapracovaná algoritmizácia, programovanie a počítačové systémy. Experimentálne učebné texty boli dokonca tak zložité, že študent, ktorý im rozumie, by mohol pokojne vynechať niektoré vysokoškolské prednášky.

3.2 Používateľská informatika

Začiatkom deväťdesiatych rokov sa začali rozširovať počítače do bežných domácností. Ľudia ich začali využívať na prácu (a samozrejme aj hranie). S počítačom sa však potrebovali naučiť pracovať. Vznikla požiadavka na školy, aby naučili študentov pracovať s počítačom. Tým pádom sa učili ovládať DOS, Windows, rôzne aplikácie ako textové editory (T602, Microsoft Word), Norton Commander alebo AutoCAD. Programovať sa učilo iba na nie-

ktorých gymnáziách. Taktiež sa učili základy fungovania počítača – binárna sústava, logické operácie, logické obvody či architektúra počítača.

Učebnice

V týchto rokoch neboli dostupné učebnice, ak sa na školách nejaké knihy používali, boli to manuály programov určené nie pre študentov, ale skôr pre profesionálov alebo samoukov. Tým pádom tieto materiály vôbec nevyužívali spojenie učiteľ-študent.

3.3 Modernizovaná informatika

Ďalšia zmena koncepcie nastala koncom 90. rokov. Začali vznikať osnovy a hlavne učebnice a pracovné listy. Náplňou sa stalo päť základných tém – informácie okolo nás, počítačové systémy, algoritmy a algoritmizácia, oblasti využitia informatiky a informačná spoločnosť.

Na stredných školách sa na zápis algoritmov používa v tomto období najčastejšie programovací jazyk Pascal/Delphi. Študenti sa snažia porozumieť princípom fungovania počítača, reprezentácií informácie v súvislosti s počítačom, učia sa dodržiavať netiketu.

Informatika sa zo strednej školy postupne rozšírila aj na základné školy. Učia sa rovnaké témy avšak v inom rozsahu a inými prostriedkami. Na zápis algoritmov sa používa napríklad jazyk Imagine Logo.

Učebnice

Postupne vznikajú učebnice a pracovné listy k jednotlivým témam, ktoré sa vzdialili od manuálovej formy a snažia sa vysvetliť princípy.

Stredné školy používajú väčšinou základnú učebnicu informatiky z roku 2001 od I. Kalaša a kolektívu: Informatika pre stredné školy [K⁺05]. K tejto učebnici sa používajú tematické zošity: Práca s grafikou [Sal00], Práca s multimédiami [ŠK05], Algoritmy s Pascalom, Programovanie v Delphi [Bla06], Práca s textom [Mac02], Práca s tabuľkami [LŠ01] a Práca s internetom [JŠB00].

Pre základné školy sa postupne vytvára kolekcia učebníc pod názvom Tvorivá informatika. Učebnice, ktoré sa už v školách naozaj používajú, sú: 1. zošit o obrázkoch [Sal05], 1. zošit z programovania [BK05], 1. zošit s Internetom [VH06] a 1. zošit o práci s textom [BS07].

Ako vidíme, učebnice pokrývajú rovnaké témy, avšak v inom rozsahu a hlavne v inej forme – prevažne vo forme pracovných listov. Podrobnejšie sa tejto téme budeme venovať v kapitole 5.

Kapitola 4

Vyučovanie informatiky v iných krajinách

Vyučovanie informatiky sa líši od krajiny ku krajine. Vo väčšine krajín študenti na konci strednej školy¹ síce absolvujú záverečnú skúšku, ktorá má národný charakter, avšak pripravujú sa na ňu na rôznych školách rozdielne. Keď budeme hovoriť o vyučovaní informatiky v nejakej krajine, bude to vyučovanie vyplývajúce z oficiálnych pedagogických dokumentov.

V tejto kapitole prezentujeme implementácie výučby informatiky vo vybraných krajinách. Pozornosť upriamime na posledné ročníky strednej školy a obsah záverečnej skúšky. Keďže pojem informatika sa v zahraničí používa v rôznych významoch, upresníme jeho význam. Popíšeme tiež model vyučovania, ktorý vyvinula spoločnosť ACM.

4.1 Definícia pojmov

Podľa [vWA02] spoločnosť UNESCO/IFIP definovala pojmy informatika, informačné technológie a informačné a komunikačné technológie nasledovne:

Informatika

Veda, ktorá sa zaoberá tvorbou, realizáciou, hodnotením, použitím a údržbou systé-

¹alebo ekvivalentne

mov na spracovanie informácií; vrátane hardvéru, softvéru, organizačným a ľudským aspektom a industriálnym, komerčným, vládny a politickým dôsledkom.

Informačné technológie (IT)

Technologické aplikácie informatiky v spoločnosti.

Informačné a komunikačné technológie (ICT)

Kombinácia informačných technológií s ostatnými príbuznými technológiami, špeciálne s komunikačnými technológiami.

Organizácia UNESCO varuje, aby sme si dávali pozor na pojem ICT v kontexte vzdelávania, pretože niekde sa tento pojem používa ako vyučovanie základných konceptov a niekde ako čistá aplikácia systémov.

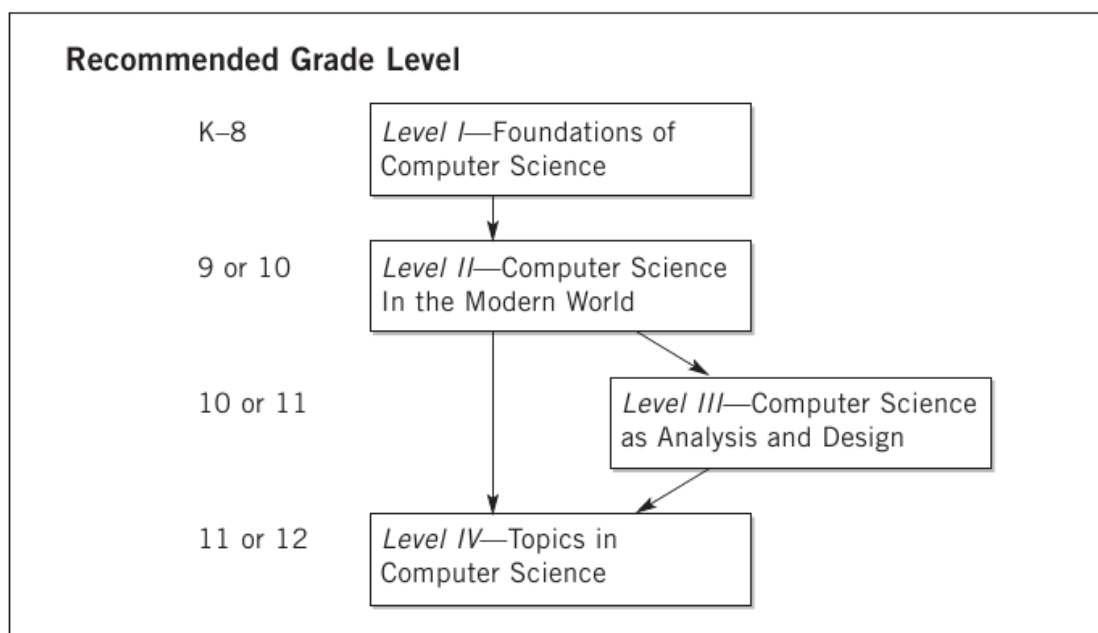
4.2 ACM model

Organizácia Association for Computing Machinery (ACM) sponzorovala vývoj modelu na vyučovanie informatiky v USA pod názvom *Model Curriculum for K-12 Computer Science* [Com05, Tuc10].

Model uvádzame na obrázku 4.1. Pozostáva zo štyroch úrovní, študent by mal v každej porozumieť niektorým častiam informatiky. Týmito časťami podľa [HAB⁺11] sú:

- algoritmicke myslenie,
- spolupráca,
- praktizovanie algoritmov,
- počítače a komunikačné zariadenia,
- spoločnosť, globálne a etické aspekty.

V ôsmom ročníku vzdelávania by študenti mali objavovať koncepty informatiky. To znamená efektívne využívať počítače a pomocou materinského jazyka objavovať základy algoritmickeho myslenia.



Obr. 4.1: ACM K-12 model pre vyučovanie informatiky [Com05]

V *deviatom* alebo *desiatom* ročníku vzdelávania by študenti mali získavať znalosti o hardvéri, softvéri, jazykoch, sieťach a ich vplyve na spoločnosť.

V *desiatom* alebo *jedenástom* ročníku vzdelávania by študenti mali získavať kompetencie v oblasti tvorby algoritmov, riešenia problémov a programovať používajúc princípy softvérového inžinierstva.

V *jedenástom* alebo *dvanástom* ročníku vzdelávania by sa mali študenti hlbšie venovať niektorej z oblastí počítačovej vedy.

Keďže v našej práci sa venujeme vyšším ročníkom strednej školy, bližšie preskúmame obsahovú náplň tretej a štvrtej úrovne v ACM K-12 modeli. V tabuľke 4.1 uvádzame obsah, ktorý tento model definuje pre tretiu úroveň.

Informatika ako analýza a dizajn

- Základné myšlienky o procese tvorby programu a riešenia problémov, súčasťou procesu tvorby softvérových aplikácií je aj štýl, abstrakcia a úvodné diskusie o správnosti a efektívnosti
- Jednoduché údajové štruktúry a ich použitie
- Témy diskkrétnej matematiky: logika, funkcie, množiny a ich vzťahy k informatike
- Tvorba spĺňajúca pravidlá použiteľnosti: tvorba webstránok, interaktívnych hier, dokumentácie
- Princípy počítačov
- Úrovně jazykov, aplikácií a prekladania: charakteristika kompilátorov, operačných systémov a sietí
- Hranice vypočítateľnosti: čo sú výpočtovo ťažké problémy (napr. modelovanie pohyblivej masy v oceáne, riadenie letovej prevádzky, mapovanie génov a pod.) a aké druhy problémov su nevypočítateľné (napr. problém zastavenia)
- Princípy softvérového inžinierstva: softvérové projekty, tímy, životný cyklus v tvorbe softvéru
- Sociálne aspekty: softvér ako duševné vlastníctvo, odborná prax
- Uplatnenie v informatike: vedec, počítačový technik, softvérový inžinier, IT špecialista

Tabuľka 4.1: Náplň informatiky v tretej úrovni v ACM K-12 modeli, podľa [Com05]

4.3 Rôzne pohľady na informatiku

V zahraničí nachádzame rôzne koncepcie vyučovania informatiky. Môžeme ich rozdeliť do troch základných vetiev:

- používanie informačných a komunikačných technológií (ďalej len IKT)
- vedecká informatika (angl. Computer Science – teoretická a aplikovaná informatika)
- vedecká informatika a princípy a používanie IKT

Každá z týchto vetiev sa dá nájsť v nejakej krajine v určitej podobe. Prvá koncepcia je používaná hlavne v menej rozvinutých krajinách, napríklad v Mongolsku [Uya06], v Indonézii [KM10], ale aj v Južnej Kórei [YYK⁺06]. Vedecká informatika sa vyučuje hlavne v Izraeli [ABGZ10]. Môžeme však nájsť aj ďalšie krajiny, v ktorých vedecká informatika silne prevažuje. Posledná koncepcia je v nejakých formách používaná vo väčšine krajín vrátane Slovenska. Rôzne podoby nachádzame vo Veľkej Británii [AQA09a], [AQA09b], [AQA09c], v Litve [BD06], v USA [Tuc10], v Poľsku [GHOK⁺05], v Českej republike [pzvv10b], [pzvv10a], či v Rusku [Kir10].

Používanie IKT

V krajinách, kde je vyučovanie informatiky sústredené na používanie IKT, sa na stredných školách učí hlavne používanie niektorých aplikačných softvérov. Väčšinou je to práca so základnými programami z balíka Microsoft Office, práca s počítačom, sociálne aspekty informatiky.

V Južnej Kórei sa informatický predmet Počítač vyučuje v 7. – 9. ročníku vzdelávania a predmet Informačná spoločnosť a počítač je nepovinný v 10. – 11. ročníku vzdelávania.

V Indonézii je informatika iba ako extrakurikulový predmet, na väčšine škôl je obsahovou náplňou iba používanie aplikačného softvéru. Niektoré školy ponúkajú v posledných ročníkoch strednej školy možnosť v rámci informatiky aj programovať.

Vedecká informatika

Najznámejšia krajina, v ktorej sa vyučuje iba vedecká informatika a nie IKT, je Izrael. Dôraz sa kladie na programovanie a tvorbu algoritmov. Študenti sa učia aj rôzne programovacie paradigmy, princípy počítačových systémov, vypočítateľnosť a softvérové inžinierstvo.

Vedecká informatika a princípy a používanie IKT

Vo väčšine krajín tvorcovia a obhajovatelia kurikúl zistili, že študent si nepotrebuje osvojiť iba základy teoretickej, či aplikovanej informatiky. Počas štúdia potrebuje tiež nadobudnúť zručnosti s prácou na počítači a pochopiť princípy fungovania IKT.

Keďže práca s IKT je pre veľa ľudí potrebná v neskoršej praxi, ale nemala by byť hlavnou náplňou skúšky potrebnej k vysokoškolskému štúdiu informatiky, vo Veľkej Británii a Litve si študenti majú možnosť vybrať z viacerých informatických záverečných skúšok. Niektoré sa venujú vedeckej informatike a niektoré používaniu a princípom IKT.

Ďalšie krajiny majú síce iba jednu záverečnú skúšku, ale počas štúdia študent absolvuje informatiku v rôznych podobách. Napríklad v Poľsku sa do 10. ročníka vzdelávania učí predmet IKT a v posledných ročníkoch sa študent pripravuje na maturitu v rámci predmetu Informatika, ktorý je obsahovo podobný záverečnej skúške z vedeckej informatiky vo Veľkej Británii.

4.4 Výučba informatiky v jednotlivých krajinách

Popíšeme obsahovú náplň informatiky a záverečnej skúšky z informatiky v krajinách, ktoré sme vybrali na základe prepracovanosti kurikula (Veľká Británia, Izrael, USA, Litva) alebo na základe historickej príbuznosti (Česko, Rakúsko, Poľsko). Okrem obsahu kurzov uvedieme aj časovú dotáciu predmetu a spôsob výučby. Opíšeme tiež skúsenosti získané dotyčnou implementáciou.

Veľká Británia

Vyučovanie informatiky vo Veľkej Británii v ročníkoch, ktoré zodpovedajú našim posledným dvom ročníkom strednej školy, je už iba prípravou na záverečnú skúšku A-Level². Študenti si vyberajú vyššiu strednú školu (angl. college) podľa toho, aké záverečné skúšky chcú robiť. Výber skúšok závisí od požiadaviek vysokej školy, na ktorú sa študent chce prihlásiť.

Z informatiky si anglický „maturant“ môže vybrať niekoľko alternatívnych skúšok: *Computing*, *Informačné a Komunikačné Technológie*³ a *Aplikované Informačné a Komunikačné Technológie*⁴. Väčšina informatických odborov na vysokej škole od študentov požaduje, aby mali spravenú skúšku *Computing*. Skúšky *IKT* a *Aplikované IKT* si vyberajú hlavne študenti, ktorí idú pracovať do firiem⁵. V tabuľke 4.2 uvádzame, pre ktorých študentov sú určené jednotlivé informatické skúšky z A-Level.

Tieto tri skúšky sa líšia, pochopiteľne, svojim obsahom. Vhodné je poznamenať, že každá z týchto skúšok a materiálov na ich prípravu (napr. [BL08], [GL09]) obsahuje časť o ergonómii pracoviska a zdravotného hľadiska pri práci s počítačom. Skúška *Computing* sa zameriava na riešenie problémov (vrátane procesu), formálne jazyky a výpočtovú zložitosť, počítačové systémy, fungovanie protokolov a praktické programovanie. Skúška *ICT* obsahuje praktické riešenie problémov, princípy fungovania digitálnych technológií, sociálne a bezpečnostné aspekty informatiky a praktický projekt z riešenia problémov. Skúška *Applied ICT* sa skladá zo sociálnych aspektov informatiky, komunikovania a prezentovania prostredníctvom IKT, práce s údajmi, administrátorskej práce, procesu programovania, používania IKT a princípov technickej podpory užívateľov. Podrobnejší obsah prezentujeme v tabuľke 4.3.

²vyššia forma našej maturity

³IKT

⁴Aplikované IKT

⁵alebo študenti, ktorí sa predbiehajú v počte A-Level predmetov

Computing [AQA09a]

Kurz nepozostáva z učenia sa používania nástrojov, alebo iba zo vzdelávania v jednom programovacom jazyku. Namiesto toho dáva dôraz na algoritmické myslenie. Algoritmické myslenie je druh rozhodovania sa, ktoré používajú ľudia aj stroje. Algoritmické myslenie je dôležitá zručnosť v živote. Algoritmicky myslieť znamená používať abstrakciu a dekompozíciu. Štúdium algoritmiky znamená, štúdium toho, čo sa dá vypočítať a ako. Informatika zahŕňa otázky, ktoré majú potenciál zmeniť pohľad na svet. Môže sa napríklad stať, že niekedy v budúcnosti budeme počítať pomocou DNA, v ktorej jednotlivé gény budeme používať ako akýsi programovací jazyk. Toto nás môže priviesť k otázke: odohráva sa takéto niečo aj v prírode? Môžeme povedať, že v prírode prebiehajú výpočty?

Informačné a Komunikačné Technológie [AQA09b]

Túto špecifikáciu sme vytvorili pre študentov, ktorí sa chcú ďalej vzdelávať alebo pracovať tam, kde porozumenie použitia IKT v spoločnosti a dôsledky ich používania budú cenným aktívom.

Aplikované Informačné a Komunikačné Technológie [AQA09c]

Špecifikácia skúšky Aplikované IKT obsahuje časť pre používateľa IKT, a tiež časť pre IT špecialistu. Používateľa IKT definujeme ako niekoho, kto používa IKT, zväčša to znamená, že používa v práci aplikácie pre osobný počítač. IT špecialista je niekto, kto sa uplatňuje v spoločnosti pomocou IKT. AS level obsahuje iba časť pre používateľa IKT. Kandidáti na A2 level musia úspešne absolovať obe časti. Predpokladáme, že kandidáti na úroveň A2 budú študenti, ktorí túžia pracovať v oblasti IKT.

Tabuľka 4.2: Zameranie záverečných informatických skúšok vo Veľkej Británii

ICT	Applied ICT
<p><i>Praktické riešenie problémov v digitálnom svete</i></p> <p>Zdravie a bezpečnosť vo vzťahu k IKT systémom, analýza, návrh riešení, výber a použitie vstupných zariadení, výber a použitie úložísk prihliadajúc na nároky, média a prístroje, výber a použitie výstupných metód, výber a použitie vhodných softvérových aplikácií, implementácia IKT riešení, testovanie IKT riešení, zhodnotenie IKT riešenia</p> <p><i>Život v digitálnom svete</i></p> <p>IKT systémy a ich súčasti, údaje a informácie, ľudia a IKT systémy, prenos údajov v IKT systémoch, bezpečnosť údajov v IKT systémoch, záloha a obnova, čo nám IKT dáva, faktory ovplyvňujúce použitie IKT, dôsledky používania IKT</p> <p><i>Použitie IKT v digitálnom svete</i></p> <p>Vývoj v budúcnosti, informácie a systémy, riadenie IKT, IKT stratégie, IKT politika, legislatíva, tvorba IKT riešení, metódy vývoja, techniky a nástroje na vývoj systémov, predstavenie veľkých IKT systémov firme, vyškolenie a podpora používateľov, vonkajšie a vnútorné zdroje</p> <p><i>Praktické otázky súvisiace s využívaním IKT v digitálnom svete</i></p> <p>Prieskum problematiky, analýza a výsledky, návrh a plánovanie implementácie, testovanie a dokumentácia, vyhodnotenie implementovaného riešenia, správa o projekte</p>	<p>IKT a spoločnosť</p> <p>IKT a organizácie</p> <p>Manipulácia s údajmi</p> <p>IKT riešenia</p> <p>Základy programovania</p> <p>Počítačové umenie</p> <p>Tvorba webstránok</p> <p>Manažovanie projektu</p> <p>Tvorba softvérových aplikácií</p> <p>Pokročilá práca s tabuľkami</p> <p>Siete a komunikácia</p> <p>Publikovanie</p> <p>Analýza systémov</p> <p>Interaktívne multimédia</p> <p>Podpora IKT používateľov</p>
Computing	
<p><i>Riešenie problémov, Programovanie, Reprezentácia údajov a Praktická skúška</i></p> <ul style="list-style-type: none"> - etapy riešenia problému, rozhodovacie tabuľky, tvorba algoritmov - údajové typy, úloha premenných, príkazy v programovaní, základy štruktúrovaného programovania, údajové štruktúry, validácia - binárna reprezentácia, kódovanie informácie, reprezentácia obrazu, zvuku a iných údajov - životný cyklus tvorby systémov <p><i>Časti počítača, programové vybavenie počítača a internet</i></p> <ul style="list-style-type: none"> - klasifikácia softvérových aplikácií, systémové aplikácie, aplikačný softvér, generácie programovacích jazykov, typy prekladačov - logické obvody, pravdivostné hodnoty a narábanie s nimi, architektúra počítača, počítačové zariadenia - internet a jeho použitie, URL, URI, doménové mená, klient-server model, štandardné protokoly, tvorba webstránok - následky vychádzajúce z používania počítača <p><i>Riešenie problémov, programovanie, operačné systémy, databázy a siete</i></p> <ul style="list-style-type: none"> - porovnávanie algoritmov, O notácia, riešiteľnosť a neriešiteľnosť, problém zastavenia, Turingove stroje, regulárne výrazy - programovacie paradigmy, štandardné algoritmy - reálne čísla - operačné systémy - databázy, model údajov, Tretia normálna forma, relačné databázy, SQL, DDL - metódy v komunikácii, topológia sietí, bezdrôtové siete, skriptovanie na strane servera, bezpečnosť na Internete <p><i>Praktický projekt z Computing</i></p> <ul style="list-style-type: none"> - analýza, návrh, testovanie, prevádzka, údržba, vyhodnotenie 	

Tabuľka 4.3: Obsah skúšok ICT, Applied ICT a Computing v rámci skúšky A-level

Izrael

Izrael je krajina, ktorá je známa tým, že ich vyučovanie informatiky dáva veľký dôraz na programovanie a tvorbu algoritmov. V nasledujúcom texte prezentujeme kurikulum pre vyučovanie informatiky v Izraeli podľa [GEH99] so zmenami, ktoré priniesla niekoľko-ročná implementácia [ABGZ10].

Počas strednej školy študent povinne absolvuje tri moduly informatiky, každý z nich sa vyučuje 90 hodín:

- *Základy vedeckej informatiky 1*
- *Základy vedeckej informatiky 2*
- cvičenia z ponuky: *Informačné systémy, Logické programovanie, Funkcionálne programovanie, Počítačová grafika, Počítačové systémy a Assembler, Úvod do programovania na webe*

Ak sa chce študent ďalej venovať informatike, absolvuje ešte ďalšie dva moduly, každý po 90 hodín:

- *Tvorba softvérových aplikácií*
- pokročilý modul z ponuky: *Vypočítateľnosť, Operačná analýza, Počítačové systémy a Assembler, Objektovo-orientované programovanie*

Základy vedeckej informatiky 1
<ul style="list-style-type: none"> • Úvod • Algoritmické riešenie problémov • Základný výpočtový model • Etapy tvorby algoritmov • Podmienené vykonávanie (if) • Správnosť algoritmu • Opakované vykonávanie (cykly) • Efektívnosť algoritmu

Tabuľka 4.4: Obsahová náplň modulu Základy vedeckej informatiky 1 v Izraeli

Základy vedeckej informatiky 2
<ul style="list-style-type: none">• Trieda string• Polia• Triedy a objekty• Schémy algoritmov (základné algoritmické schémy pre počítavanie, vyhľadávanie, triedenie)• Riešenie problémov

Tabuľka 4.5: Obsahová náplň modul Základy vedeckej informatiky 2 v Izraeli

V trojmodulovom programe študent okrem základov vedeckej informatiky absolvuje buď aplikačný modul alebo niektorú z iných paradigiem programovania. V päťmodulovom programe študent povinne absolvuje *Pokročilé programovanie* a môže si vybrať dva z troch typov modulov: iné paradigmy programovania, teoretický modul alebo aplikačný modul.

USA

V USA sa líši vyučovanie informatiky nielen v závislosti od štátu, ale aj od školy. Množstvo škôl v USA však ponúka kurzy podľa modelu, ktorý vyvinula spoločnosť ACM, pozri podkapitolu 4.2.

Prípravou na záverečnú skúšku je kurz štvrtej úrovne. Podľa modelu by mal byť projektovo orientovaný a mať rovnakú náplň, ako má skúška Advanced Placement Computer Science, ktorú uvádzame v tabuľke 4.6.

Litva

Vyučovanie informatiky v Litve vo vyšších ročníkoch strednej školy je rozdelené na dva predmety, pozri [BD06, DDS06]: *informačné technológie a pokročilý kurz programovania*. Kurzy sú prípravou na záverečnú skúšku, ktorá je obdobou našej maturity. K oboch predmetom existuje samostatná záverečná skúška, takže študent si môže vybrať aj skúšku z Informačných technológií aj z Programovania.

Advanced Placement Computer Science Course
<ul style="list-style-type: none">• Návrh v objektovo-orientovanom programovaní<ul style="list-style-type: none">– Návrh programu– Návrh tried• Implementácia programu<ul style="list-style-type: none">– Techniky implementácie– Konštrukty v programovaní– Knižničné triedy v Java• Analýza programu<ul style="list-style-type: none">– Testovanie– Ladenie– Porozumenie a zmena existujúceho kódu– Rozšírenie existujúceho kódu použitím dedenia– Porozumenie zachytávaniu výnimiek– Porozumenie a vysvetlenie existujúceho kódu– Analýza algoritmov– Reprezentácia čísel a z toho vyplývajúce obmedzenia• Štandardné údajové štruktúry<ul style="list-style-type: none">– Základné údajové typy (int, boolean, double)– Triedy– Zoznamy– Polia• Štandardné algoritmy<ul style="list-style-type: none">– Operácie na vyššie spomenutých údajových štruktúrach– Vyhľadávanie– Triedenie• Kontext informatiky<ul style="list-style-type: none">– Spoľahlivosť systémov– Súkromie– Právne otázky a súkromné vlastníctvo– Sociálne a etické následky používania počítačov

Tabuľka 4.6: Obsahová náplň kurzu Advanced Placement Computer Science v USA [tCB09]

Kurz informačných technológií študent absolvuje v 11. a 12. roku vzdelávania⁶ a je povinný. Predmet sa vyučuje počas 70 hodín, teda dve hodiny týždenne. V tomto kurze študenti pracujú so zložitejšími textami a oboznamujú sa so základnými princípmi sadzby textu, surfujú po webe a komunikujú, tvoria prezentácie, pracujú s tabuľkami a diskutujú o kognitívnych, sociálnych a etických aspektoch informatiky [DDS06]. Obsah záverečnej skúšky je podobný obsahu kurzu. Pozostáva z piatich tém: formátovanie textu, použitie tabuľkového kalkulátora, web, tvorba prezentácií a sociálne a etické aspekty.

Pokročilý kurz programovania si študent môže vybrať v 11. roku vzdelávania, ktorý sa vyučuje taktiež počas 70 hodín. Kurz pokrýva štyri oblasti: základné konštrukcie Pascalu, údajové štruktúry, algoritmy a verzia Pascalovského jazyka vo Free Pascal prostredí.

Záverečná skúška z programovania pozostávala v školskom roku 2004/05 z testu, v ktorom približne polovica otázok bola z informačných technológií a zvyšná časť z programovania a dvoch programátorských úloh. Obsah skúšky z programovania prezentujeme v tabuľke 4.7.

Algoritmy	Údajové štruktúry	Programovací jazyk (Pascal)
Rátanie súm (násobenie, počet, aritmetický priemer). Vyhľadávanie maximálnej (minimálnej) hodnoty. Vstupy, výstupy. Triedenie. Schopnosť zmeniť algoritmy pri zmene údajovej štruktúry.	Integer a real, char, boolean a string. Textový súbor. Jednorozmerné pole. Record. Schopnosť vytvárať jednoduchšie údajové štruktúry.	Štruktúra programu, komentáre, premenné, priradenia a príkazy, matematické a logické operácie, príkaz if. Cykly. Zložené príkazy. Procedúry a funkcie. Zoznam parametrov. Štandardné matematické procedúry a funkcie. Procedúry a funkcie súvisiace so súbormi.
Rozhranie programovacieho jazyka. Štruktúrované programovanie. Testovanie. Dokumentácia k programu. Rozmiestnenie okien. Štýl písania programov.		

Tabuľka 4.7: Skúška z programovania v Litve, podľa [BD06]

⁶ekvivalent k 3. a 4. ročníku SŠ

Česko

V Česku je informatika smerovaná viac k informačným technológiám ako k informatike a aj sa vyučuje pod názvom *Informatika a výpočetní technika*. Obsahom maturitnej skúšky pre základnú [pzvv10b] aj vyššiu úroveň obtiažnosti [pzvv10a] sú nasledujúce tematické celky:

- základy informatiky a teórie informácie,
- technické vybavenie počítačov a počítačových sietí,
- programové vybavenie počítačov,
- človek, spoločnosť a počítačové technológie,
- využívanie služieb internetu,
- počítačové spracovanie textov a tvorba zdieľaného obsahu,
- počítačová grafika, prezentácia informácií a multimédiá,
- hromadné spracovanie dát a číselných údajov,
- algoritmizácia a základy programovania.

K príprave na maturitu sa využívajú učebnice *Informatika a výpočetní technika pro střední školy: teoretická učebnice* [Rou10b] a *Informatika a výpočetní technika pro střední školy: praktická učebnice* [Rou10a]. Spoločne tieto učebnice pokrývajú uvedený obsah požiadaviek na maturitnú skúšku z informatiky.

Poľsko

V Poľsku podľa [GHOK⁺05] a [Kol08] žiaci prvého stupňa základnej školy (6 rokov) absolvujú povinný predmet *informatika*, ktorého náplňou je naučiť sa používať počítač a jeho softvér, aby to mohli využiť na iných predmetoch. Na druhom stupni (3 roky) sa v rámci tohto predmetu používať počítače, počítačové siete a multimédia na vyššej úrovni. Tak tiež začínajú tvoriť algoritmy v programovacom prostredí Logo. Na strednej škole (3 roky) majú povinný predmet *informačné technológie*, ktorý pokračuje v predchádzajúcej výučbe, pričom programovanie je vynechané. Ako voliteľný predmet si študenti môžu zvoliť predmet *informatika*, ktorý sa venuje riešeniu problémov v týchto krokoch: analýza problému,

vytvorenie špecifikácie, navrhnutie riešenia, realizácia a testovanie riešenia. Študentom sú predstavené klasické algoritmy, programovacie jazyky, teória databáz a programovanie interaktívnych webových stránok.

Maturitná skúška pokrýva tieto oblasti:

- porozumenie základným konceptom, metódam a procesom v informatike,
- použitie vedomostí a informácií k riešeniu teoretických a praktických úloh,
- použitie metód informatiky pri vytváraní nových informácií a riešení problémov.

Zaujímavosťou je, že učiteľ informatiky na strednej škole musí podľa štandardov absolvovať štúdium informatiky (nie učiteľstvo).

Rakúsko

Podľa [HAB⁺11] a rozhovorov s Petrom Micheuzom je v Rakúsku vyučovanie informatiky veľmi podobné s vyučovaním v Bavorsku. Informatika sa skladá z týchto oblastí:

- reprezentácia informácií,
- spracovanie a prenos reprezentácie,
- interpretácia reprezentácie.

Hubwieser [Hub00] zaradil vedomosti každej z týchto oblastí do jednej zo štyroch kategórií:

- relevantné aj mimo limitov automatického spracovania informácií, napríklad modelovacie techniky, ktoré môžeme aplikovať na reálne systémy,
- relevantné pre všetky systémy IKT, napríklad algoritmy, hlavné limity vypočítateľnosti,
- relevantné pre určité triedy systémov IKT, napríklad koncept registrového stroja, údajové štruktúry v textových procesoroch alebo tabuľkových kalkulátoroch, princípy objektovo orientovaného programovania,
- relevantné pre konkrétne inštancie systémov IKT, napríklad štruktúra menu v MS Word 2010, ako pridať obľúbené URL do Firefoxu 3.0 a prvky syntaxe Java 2.0.

Podľa týchto pravidiel bolo vytvorené kurikulum informatiky, ktorého obsah uvádzame v tabuľke 4.8.

Téma	Koncept
reprezentácia informácie	reprezentácia, interpretácia
objektovo orientované modelovanie dokumentov	objekt, atribút, trieda, metóda, agregácia
hierarchické štruktúry	strom, koreň, list, vrchol, hrana
sieťové štruktúry	linka, kotva, internet, cyklické štruktúry, referencia, hypertext
výmena informácií	e-mail, príloha, mail server, princípy doručovania e-mailu
základne koncepty algoritmov	reprezentácia algoritmov, kontrolné štruktúry (postupnosť príkazov, podmienky a cykly)
modelovanie funkcionality	data-flow diagramy, funkcie, parametre, návratová hodnota, zreťazenie, jednoduché údajové štruktúry
modelovanie údajov	objekt (záznam), trieda (tabuľka), asociácia (relácia), dotazovací jazyk (SQL, ochrana a bezpečnosť dát)
objektovo orientované modelovanie a programovanie	objektové a triedové diagramy, stavové a postupnostné grafy, premenné, priradovací príkaz, pole, zapúzdrenie údajov, generalizácia, polymorfizmus, špecializácia, stav objektov, stavové automaty
generalizácia a špecializácia	dedenie, polymorfizmus, hierarchie tried
softvérový projekt	kombinácia viacerých modelovacích a implementačných techník (t.j. OOP a databázové systémy)
rekurzívne údajové štruktúry	zoznamy, stromy, grafy, rekurzívne algoritmy
softvérové inžinierstvo	plánovanie projektu, životný cyklus softvéru, procesný model, koordinácia paralelných procesov
formálne jazyky	abeceda, BNF, gramatiky, syntax, sémantika, syntax diagram, konečný automat
synchronizácia paralelných procesov	komunikačné protokoly, vrstvé modely, topológia počítačových sietí, internet, koncepty semaforu a monitoru
základná funkcionality počítača	komponenty (CPU, pamäť, úložné systémy), von Neumannová architektúra, registrové stroje, assembler, počítač ako stavový automat
limity vypočítateľnosti	časová zložitosť, problémy neriešiteľné vôbec, problémy neriešiteľné efektívne, problém zastavenia, šifrovanie údajov

Tabuľka 4.8: Obsahová náplň informatiky v Rakúsku

Kapitola 5

Aktuálny stav vyučovania informatiky na Slovensku

V posledných rokoch sa vyučovanie informatiky zmenilo. Zmenila sa časová dotácia v jednotlivých ročníkoch, na základnej škole pribudol predmet informatická výchova a k maturite z informatiky vznikli nové cieľové požiadavky.

V tejto časti predstavíme časovú dotáciu informatiky vyplývajúcu zo Štátneho vzdelávacieho programu [Hau08a, Hau08b] a vyhlášky o ukončovaní štúdia [Vyh08]. Preskúmame obsahovú náplň informatiky ako všeobecnovzdelávacieho predmetu v rôznych stupňoch štúdia z pohľadu obsahového štandardu. Predstavíme v súčasnosti platné cieľové požiadavky na vedomosti a zručnosti maturantov z informatiky.

5.1 Časová dotácia informatiky

Podľa Štátneho vzdelávacieho programu škola vytvára rámcový učebný plán ako súčasť Školského vzdelávacieho programu, pozri [SH08]. Študent musí povinne absolvovať počet hodín týždenne, ktorý uvádza Štátny vzdelávací program. Škola mu môže ponúknuť predmet vo vyššom rozsahu, ale iba v takom počte hodín, aby sa pre všetky predmety zmestila do obmedzenia pre voliteľné hodiny. V tabuľkách 5.1, 5.2 a 5.3 predkladáme počty ho-

KAPITOLA 5. AKTUÁLNY STAV VYUČOVANIA INFORMATIKY NA SLOVENSKU³⁵

dín informatických predmetov týždenne, ktoré absolvuje študent gymnázia od začiatku základnej školy.

Predmet	1.	2.	3.	4.	1.	2.	3.	4.	5.	6.	7.	8.	Spolu
Informatika	–	–	–	–	0.5	0.5	0.5	0.5	1	1	1	–	5
Informatická výchova	–	1	1	1	–	–	–	–	–	–	–	–	3

Tabuľka 5.1: Počet hodín povinnej informatiky pre jednotlivé ročníky podľa rámcových učebných plánov pre osemročné gymnázia a základné školy podľa [Hau08b, Hau08a]

Predmet	1.	2.	3.	4.	5.	6.	7.	8.	9.	1.	2.	3.	4.	Spolu
Informatika	–	–	–	–	0.5	0.5	0.5	0.5	0.5	1	1	1	–	5.5
Informatická výchova	–	1	1	1	–	–	–	–	–	–	–	–	–	3

Tabuľka 5.2: Počet hodín povinnej informatiky pre jednotlivé ročníky podľa rámcových učebných plánov pre štvorročné gymnázia a základné školy podľa [Hau08b, Hau08a]

Predmet	1.	2.	3.	4.	5.	6.	7.	8.	1.	2.	3.	4.	5.	Spolu
Informatika	–	–	–	–	0.5	0.5	0.5	0.5	1	1	1	–	–	5
Informatická výchova	–	1	1	1	–	–	–	–	–	–	–	–	–	3

Tabuľka 5.3: Počet hodín povinnej informatiky pre jednotlivé ročníky podľa rámcových učebných plánov pre bilingválne gymnázia a základné školy podľa [Hau08b, Hau08a]

Ako môžeme vidieť, študent počas gymnázia povinne absolvuje spolu tri hodiny informatiky týždenne. Študent, ktorý zodpovedá profilu, ktorý skúmame, t.j. maturant z informatiky, musí podľa [Vyh08] na gymnázium absolvovať spolu šesť hodín informatických predmetov týždenne. Zvyšné mu ponúkne škola v rámci voliteľných predmetov. Sumárne teda počas gymnázia musí absolvovať približne 200 vyučovacích hodín.

5.2 Obsahová náplň informatiky

Vzdelávací obsah informatiky pre ISCED 2 a ISCED 3 (pozri [Hau08a] a [Hau08b]) je rozdelený na päť tematických okruhov:

- *Informácie okolo nás,*

KAPITOLA 5. AKTUÁLNY STAV VYUČOVANIA INFORMATIKY NA SLOVENSKU³⁶

- *Komunikácia prostredníctvom IKT,*
- *Postupy, riešenie problémov, algoritmické myslenie,*
- *Princípy fungovania IKT,*
- *Informačná spoločnosť.*

Vyučovanie tém uvedených okruhov sa líši na základnej a strednej škole rozsahom, hĺbkou a formou. Toto načrtujeme na príklade tematického okruhu *Informácie okolo nás* v tabuľke 5.4.

Prezentovaný vzdelávací obsah tematického okruhu *Informácie okolo nás* na strednej škole je platný pre štúdium podľa Štátneho vzdelávacieho programu, t.j. pri dotácii spolu tri hodiny týždenne za celé štúdium na gymnáziu. Maturant musí absolvovať aspoň šesť hodín týždenne. Pre takého študenta je nutné pokryť rozdiely medzi ŠVP a cieľovými požiadavkami na maturitu z informatiky.

Na začiatku výskumného projektu boli platné cieľové požiadavky na maturitu z informatiky platné pre školský rok 2010/2011 [Špú08], preto v tabuľke 5.5 pre porovnanie prezentujeme obsahový štandard týchto cieľových požiadavok.

Základná škola, 2. stupeň	Stredná škola
typy informácií, reprezentácia, bit, bajt	Údaj, informácia, znalosť, jednotky informácie, digitalizácia, kódovanie. Písmo — forma kódovania. Číselné sústavy, prevody. Komprimácia. Šifry. Reprezentácia údajov v počítači, čísla, znaky. Zber, spracovanie, prezentovanie informácie.
formátovanie textu, nadpisy, odrážky, obrázky v texte	Textová informácia — kódovanie, jednoduchý, formátovaný dokument, štýl, aplikácie na spracovanie textov, pokročilé formátovanie — hlavička, päta, štýly, automatický obsah.
grafická informácia, fotografia, animácia	Grafická informácia — rastrová, vektorová grafika; animovaná grafika, video, kódovanie farieb; grafické formáty; aplikácie na spracovanie grafickej informácie.
informácie v tabuľkách, bunka, vzťahy medzi bunkami, grafy	Číselná informácia, spracovanie a vyhodnocovanie, tabuľkový kalkulátor – bunka, hárok, vzorec, funkcia, odkazy, grafy, triedenie, vyhľadávanie, filtrovanie.
úprava zvukov, hudobný formát, prehrávanie a vytváranie videa	Zvuková informácia — formáty, aplikácie na nahrávanie, spracovanie, konverzie, prehrávanie.
prezentácia, snímka, prezentačný program, prezentácia na webe	Prezentácia informácií — aplikácie na tvorbu prezentácií – snímka, stránka, spôsoby tvorby prezentácií. Prezentácia informácií na webovej stránke – Aplikácie na tvorbu webových stránok — hypertext, odkazy. Pravidlá prezentovania, zásady tvorby prezentácie.
encyklopédia, odkazy	Vstup a výstup informácie v závislosti od jej typu. Uchovávanie informácie — typy a limity zariadení.
	Prenos informácií medzi aplikáciami.

Tabuľka 5.4: Rozdiely medzi vyučovaním tematického okruhu *Informácie okolo nás* na základnej a strednej škole, pozri [Hau08a, Hau08b]

Informácie okolo nás

- údaj, informácia, jednotky informácie, digitalizácia, písmo
- číselné sústavy – prevody
- reprezentácia údajov v počítači – čísla, znaky
- textová informácia – aplikácie na tvorbu a spracovávanie rôznych formátov, kódovanie, jednoduchý, formátovaný dokument
- textový editor – formátovanie, štýl
- tabuľkový kalkulátor – bunka, hárok, vzorec, funkcia, odkazy, grafy, triedenie, vyhľadávanie, filtrovanie
- prezentácie – snímka, stránka, spôsoby tvorby prezentácií, prezentácie na webe – hypertext, odkazy
- grafická informácia – rastrová, vektorová grafika; kódovanie farieb; grafické formáty
- grafický editor – typické nástroje na úpravu v rastrových editoroch
- zvuková informácia – formáty, programy na nahrávanie, spracovanie, konverzie, prehrávanie
- uchovávanie informácie – typy a limity zariadení

Počítačové systémy

- základné pojmy – hardvér, softvér
- počítač – princíp práce počítača, časti počítača – mikroprocesor, pamäte (vnútorné, vonkajšie)
- prídavné zariadenia – klávesnica, myš, monitor, tlačiareň, skener, modem, tablet, mikrofón, reproduktor; rozdelenie prídavných zariadení podľa vstupu a výstupu údajov, využitie jednotlivých prídavných zariadení pri zbere, spracúvaní, uchovávaní a prezentácii údajov
- softvér – rozdelenie podľa oblastí použitia
- operačný systém – základné vlastnosti a funkcie (spravovanie zariadení, priečinkov a súborov)
- počítačová sieť – výhody, architektúra, rozdelenie sietí podľa rozľahlosti (spôsoby pripojenia)

(pokračuje na nasledujúcej strane)

(pokračovanie z predchádzajúcej strany)

Internet

- internet – história, základné pojmy (adresa, URL, poskytovateľ služieb), služby, klient/server
- neinteraktívna komunikácia – e-pošta
- interaktívna komunikácia – rozhovor (Talk, IRC, ICQ)
- web – prehliadače, webová stránka, vyhľadávanie informácií
- bezpečnosť
- netiketa

Algoritmy a programovanie

- etapy riešenia problému – rozbor problému, algoritmus, program, ladenie
- programovací jazyk – syntax, spustenie programu, logické chyby, chyby počas behu programu
- pojmy – príkazy (priradenie, vstup, výstup), riadiace štruktúry (podmienené príkazy, cykly), podprogramy, premenné, typy (číselný, logický, znakový) – množina operácií, údajové štruktúry (jednorozmerné pole, reťazec, textový súbor)

Informačná spoločnosť

- informatika (použitie, dôsledky a súvislosti) v rôznych oblastiach – administratíva, elektronická kancelária, vzdelávanie, šport, umenie, zábava, virtuálna realita
- softvérová firma – pojmy upgrade, registrácia softvéru, elektronická dokumentácia
- riziká informačných technológií – vírusy (pojmy, typy vírusov, detekovanie, prevencia), kriminalita, nevyžiadané e-maily (spam), spyware, poplašné správy (hoax)
- etika a právo – autorské práva na softvér (freeware, adware, shareware, demoverzia, multilicencia,...)

Tabuľka 5.5: Obsahový štandard platných cieľových požiadaviek na vedomosti a zručnosti maturantov z informatiky, podľa [Špú08]

Cieľové požiadavky na vedomosti a zručnosti maturantov sa počas behu projektu zmenili, pozri [Špú10]. Tematické okruhy v požiadavkách sa zjednotili s okruhmi definovanými v Štátnom vzdelávacom programe. Pribudlo veľa nových tém (napr. šifrovanie, kompresia údajov, hashovanie), niektoré témy sa rozšírili (napr. webová prezentácia – štandardy W3C, WCAG, programovanie na strane klienta aj servera, komunikácia s databázovým serverom).

5.2.1 Učebnice

Na pomoc pri výučbe informatiky ako všeobecno-vzdelávacieho predmetu sa používa učebnica Informatika pre stredné školy, pozri [K⁺05]. V nej je spracovaných všetkých päť tematických okruhov, a to v rozsahu Štátneho vzdelávacieho programu. Záver učebnice sa navyše venuje téme *Čo je to informatika*, čím pomáha študentom rozhodnúť sa, či budú pokračovať v štúdiu informatiky aj na vysokej škole.

Na prípravu na maturitu z informatiky je dostupná kniha Informatika na maturity a prijímacie skúšky [SKLŠ11], ktorá pokrýva a mierne presahuje obsahové štandardy definované v cieľových požiadavkách na maturitu z informatiky [Špú08].

5.2.2 Tematický výchovno-vzdelávací plán

Dôležitou súčasťou pedagogickej dokumentácie je tematický výchovno-vzdelávací plán, donedávna iba časovo-tematický plán. Na obrázku 5.1 a 5.2 s povolením autora predostrieme príklad časovo-tematického plánu pre seminár z informatiky, ktorý sa používa na nemenovanom osemročnom gymnáziu.

Časovo-tematický plán - **Seminár z informatiky**

Ročník: **Septima**
 Počet h. týždenne: **1**
 Počet h. ročne: **33**

Školský rok: **2009/2010**
 Vyučujúci:

Mes.	Hod.	Téma učiva	
Sept.	1. 2. 3.	Úvod, poučenie o zásadách práce v učebni inf. Oboznámenie sa s obsahom výučby.	
Okt.	4. 5. 6. 7.	História počítačov Prídavné zariadenia. Základné komponenty počítača. Operačný systém Pamäte Textový editor Formátovanie a úprava textu	
Nov.	8. 9. 10. 11.	Informatika, informatická spoločnosť Tabuľkový editor Preverovanie vedomostí Zvuk Jazyk HTML Štruktúra HTML dokumentu Aplikácie na tvorbu www stránok Programovanie www stránky	
Dec.	12. 13. 14.	Programovanie www stránky Preverovanie vedomostí Netiketa, Aplikačný software Prezentačný editor Prezentačný editor Komunikácia Vyhľadávanie informácií Vyhľadávanie informácií	
Jan.	15. 16. 17.	Grafický rastrový editor Grafický vektorový editor Animácie Opakovanie Preverovanie vedomostí Internet, vyhľadávanie informácií Počítačové siete	
Feb.	18. 19. 20.	Komunikácia v sieti Programovací jazyk PASCAL Číselné sústavy Programovanie Údaj, informácia, Grafika Programovanie Preverovanie vedomostí Programovanie	

Obr. 5.1: Ukážka časovo-tematického plánu semináru z informatiky v septime osemročného gymnázia

KAPITOLA 5. AKTUÁLNY STAV VYUČOVANIA INFORMATIKY NA SLOVENSKU42

Mar.	21. 22. 23. 24.	Programovanie Preverovanie vedomostí Programovanie Opakovanie Preverovanie vedomostí Programovanie Autorské práva Programovanie	
Apr.	25. 26. 27. 28.	Opakovanie Opakovanie Preverovanie vedomostí Programovanie Programovanie Programovanie Softwarová forma Programovanie	
Máj	29. 30. 31.	Počítačové vírusy Programovanie Preverovanie vedomostí Programovanie Hodnotenie Hodnotenie	
Jún	32. 33.	Záverečné opakovanie Časová rezerva Časová rezerva Hodnotenie celoročnej činnosti	

Prerokované a schválené na predmetovej komisii dňa

Schválilo Ministerstvo školstva Slovenskej republiky 24.2.1997 pod číslom 1252/96-15 s platnosťou od 1.9.1997

Obr. 5.2: Ukážka časovo-tematického plánu semináru z informatiky v septime osemročného gymnázia

Kapitola 6

Vzdelávacie ciele a poznávací proces

V tejto kapitole predstavíme taxonómie vzdelávacích cieľov špecifické pre informatiku a dobre známe etapy poznávacieho procesu. Tieto pojmy je treba upresniť, lebo ich budeme často využívať pri tvorbe vzdelávacích cieľov, či vytváraní súboru aktivít, ktorý tvoríme v súlade s poznávacím procesom.

6.1 Taxonómie vzdelávacích cieľov

V súčasnosti sa pri tvorbe vzdelávacích cieľov najčastejšie používa revidovaná forma Bloomovej taxonómie. Tá je vhodná, ak chceme špecifikovať výsledné vzdelávacie ciele. Nie vždy nám však stačí, treba si uvedomiť, že každý študent je iný a pre každého je najvhodnejšia iná cesta za dosiahnutím vzdelávacieho cieľa. Z tohto dôvodu predstavíme aj tzv. dvojrozmernú adaptáciu Bloomovej taxonómie. Vo svete sa používa aj mnoho ďalších taxonómií, napríklad SOLO taxonómia a ABC taxonómia, pozri [FJA⁺07], ktorým sa však z dôvodu rozsahu a potreby práce nebudeme viac venovať.

Revidovaná Bloomova taxonómia

Na špecifikovanie vzdelávacích cieľov budeme používať dobre známu revidovanú Bloomovú taxonómiu. Tá vznikla v deväťdesiatych rokoch minulého storočia upravením Bloomovej

taxonómie pre kognitívnu doménu. Namiesto podstatných mien sa používajú slovesá na označenie kategórie sloviac patriacej k jednej úrovni poznania a vymenilo sa poradie dvoch kategórií. Slovesá patriace ku každej kategórii si môže čitateľ pozrieť napríklad v [Cla10].

Podľa [FJA⁺07] je ďalším vylepšením revidovanej Bloomovej taxonómie rozdelenie poznatkov do štyroch dimenzií: faktické poznatky, konceptuálne poznatky, procedurálne poznatky a metakognitívne poznatky. To sa dá zobrazíť do dvojrozmernej tabuľky:

		Dimenzia poznatkov			
		A. Faktické poznatky	B. Konceptuálne poznatky	C. Procedurálne poznatky	D. Metakognitívne poznatky
Dimenzia kognitívnych procesov	1. Zapamätať si				
	2. Porozumieť				
	3. Aplikovať				
	4. Analyzovať				
	5. Hodnotiť				
	6. Tvoriť				

Dvojrozmerná adaptácia Bloomovej taxonómie

V roku 2007 vytvoril Fuller a kolektív dvojrozmernú adaptáciu Bloomovej taxonómie, ktorá je prispôbená učeniu sa programovať. Autori v nej zohľadňujú rozličné potreby študentov, pretože niektorí študenti sa venujú najprv teórii a potom ju prakticky skúšajú a niektorí práve naopak, najprv veľa prakticky programujú, a až potom dosahujú teoretické poznatky. Túto adaptáciu je vhodné použiť, keď učiteľ analyzuje stupeň poznania nejakého konceptu u jednotlivých študentov. Získavanie nových poznatkov sa dá zobrazíť šípkami

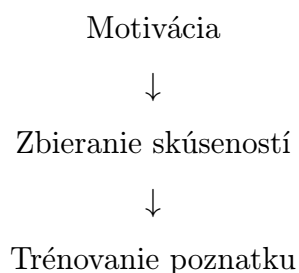
ako cesta, po ktorej sa študent posúva v matici. Najvyšší dosiahnutý cieľ sa nachádza v pravom hornom rohu. Detailnejšie je táto taxonómia popísaná v [FJA⁺07].

Produkcia	Vytvoriť				
	Aplikovať				
	nič				
		Zapamätať si	Porozumieť	Analyzovať	Hodnotiť
		Interpretácia			

6.2 Etapy poznávacieho procesu

Pripomeňme etapy poznávacieho procesu, ktoré budeme využívať pri tvorbe súboru úloh.

Podľa [STB10a] by sme v programovaní mali dodržiavať nasledovnú schému pre vznik poznatku:



Teda súbor úloh je treba voliť tak, aby sme prešli všetkými etapami, a to v uvedenom poradí. Najprv je treba študentov motivovať. Neznamená to len, že majú riešiť zábavné úlohy. Znamená to, že pri riešení úloh by mali cítiť potrebu použiť daný poznatok. V ďalších úlohách by mali získavať skúsenosti s daným poznatkom a nakoniec ho tiež natrénovať.

Kapitola 7

Priebeh výskumu

Cieľom projektu dizertačnej práce bolo vytvoriť metodiku pre vyučovanie informatiky v prvej časti prípravy na maturitu. Použili sme na to iteratívny vývoj podľa pravidiel výskumu vývojom. V tejto kapitole prezentujeme priebeh výskumu ako aj návrh metodiky a časovo-tematického plánu, ktorý je výstupom druhej iterácie výskumu. Z dôvodu rozsahu práce uvedieme podrobnú metodiku iba pre dve vybrané témy. Ďalšie informácie o kurze sú uvedené v prílohe.

Výskum sa skladal z týchto častí:

1. Pozorovanie a výučba v rámci mimoškolského prípravného kurzu na maturitu z informatiky

Do mimoškolského prípravného kurzu sa zapojili študenti z rôznych škôl v bratislavskom kraji, ktorí mali pocit, že ich v škole nepripravujú dôsledne na maturitu z informatiky a štúdium na vysokej škole. To sme využili na získanie informácií o problémových témach v informatike. Pozorovali sme ich pri riešení problémov a analyzovali test z informatiky (monitor), ktorí absolvovali bez domácej prípravy.

2. Interview s učiteľmi informatiky

S učiteľmi informatiky sme spravili pološtruktúrované interview, aby sme spoznali problémy vo vyučovaní informatiky a inšpirovali sa ich cennými skúsenosťami.

3. Nultá iterácia – pozorovanie na seminári z informatiky

V nulte iterácii sme na bratislavskom gymnáziu pozorovali vyučovanie seminára z informatiky, kde sa študenti pripravovali hlavne na maturitnú skúšku z informatiky a štúdium na vysokej škole. Získali sme mnoho informácií a inšpirácií k tvorbe metodiky. Pozorovali sme učiteľa aj žiakov, pýtali sa na postoje študentov, niekoľko vyučovacích hodín programovania odučili a pozorovali výsledky. Taktiež sme mali možnosť uskutočniť pozorovanie na maturitnej skúške a analyzovali sme výsledky testu z informatiky (monitor).

4. Prvá iterácia – pozorovanie a výučba seminára z informatiky

V prvej iterácii sme na inom bratislavskom gymnáziu začali výučbu seminára z informatiky podľa prototypu časovo-tematického plánu, ktorý sme menili priebežne počas školského roka, pretože sa ukázal ako príliš obtiažny. Každú vyučovaciu hodinu sme konzultovali s učiteľkou, ktorá bola prítomná pri výučbe a dala nám mnoho rád a nápadov. Na začiatku školského roka sme zistili informácie o vedomostiach a skúsenostiach študentov rozhovormi s učiteľkou aj s nimi samotnými. Na konci školského roka sme študentom položili vedomostný a postojový dotazník. Týchto študentov sme pozorovali aj o rok neskôr na maturitnej skúške.

5. Druhá iterácia – pozorovanie a výučba seminára z informatiky

Druhú iteráciu sme vykonali na rovnakej škole ako prvú iteráciu a postupovali podobne. Podľa nadobudnutých skúseností a výsledkov sme upravili časovo-tematický plán a súbor aktivít, s ktorými sme vstupovali do druhej iterácie. Vyučovacie hodiny sme konzultovali s učiteľkou, ktorá bola znova prítomná pri výučbe. Študenti absolvovali vedomostno-postojový dotazník na začiatku a na konci školského roka.

Ďalšie nápady a inšpirácie sme získali zo štúdia odbornej literatúry, účasti na vedeckých konferenciách, rozhovormi s kolegami a pedagogickej praxe získanej v minulosti alebo počas vzdelávania DVUI.

7.1 Pozorovanie a výučba v rámci mimoškolského prípravného kurzu na maturitu z informatiky

Do prípravného kurzu na maturitu z informatiky a štúdium informatiky na vysokej škole sa zapojilo 12 študentov zo stredných škôl v Bratislavskom kraji. Boli rozdelení na dve skupiny podľa pokročilosti v programovaní. V pokročilej skupine nebol problém s programovaním na maturitnej úrovni, ale zistili sme nedostatky vo zvyšných tematických okruhoch. V začiatničkej skupine sme sa venovali programovaniu, pretože napriek tomu, že väčšina týchto študentov navštevovala štvrtý ročník strednej školy, s programovaním sa zatiaľ nestretli, alebo začali programovať v škole až v tomto poslednom ročníku. Polovica študentov zo začiatničkej skupiny boli gymnazisti.

Tento kurz sa však od zvyšného výskumu líši hlavne tým, že študenti sa prihlásili dobrovoľne, vo svojom voľnom čase, takže mali veľkú motiváciu snažiť sa nielen na kurze, ale aj doma. Preto nebol problém zvoliť vysoké tempo a príklady na precvičenie nových vedomostí riešili aj doma.

Kvalitatívnou analýzou riešení testu (monitor 2004 [Špú04]) sme zistili, že študenti mali problémy hlavne s:

- autorskými právami tvorcov softvéru,
- počítaním veľkosti súboru,
- princípmi fungovania počítačovej siete,
- logickými podmienkami,
- trasovaním programu.

Keďže na riešenie mali študenti iba polovicu času, programátorskú časť monitoru nestihli vyplniť.

Analyzujme zistené problémy. Niektoré nedostatky mohli vzniknúť kvôli tomu, že kurz sa konal v prvom školskom polroku. Napríklad nedostatok informácií o princípoch fungovania počítačovej siete nie je takým veľkým problémom, pretože ich môžu získať ešte počas druhej časti školského roka. Na druhej strane neznalosť autorských práv tvorcov softvéru je

problémom, pretože je to návyk, ktorý mali získať už v minulosti. Za veľký problém považujeme zlé vytváranie logických podmienok. To je problém informatiky ale aj matematiky. Pre nás to znamená, že práci s podmienkami budeme musieť venovať viac času.

7.2 Interview s učiteľmi informatiky

Aby sme zmapovali stav vyučovania informatiky na gymnáziách na Slovensku, spravili sme s tromi učiteľmi pološtruktúrované interview. Všetci učili na gymnáziu bez informatického alebo matematického zamerania, a to dvaja v Bratislave a jeden v Banskej Bystrici. Prepis interview čitateľ nájde v prílohe.

Témy, ktoré sme pripravili, boli tieto:

- typ školy,
- Čo učíš, na ktoré predmety máš aprobáciu?
- Čo a kde si vyštudoval/a?
- Ako dlho učíš informatiku (na tejto škole, celkovo)?
- na tejto škole – Ako využívate rámcový učebný plán?
- Aké okruhy učíte v povinnej informatike (1./2. ročník) a do akej miery
- Koľko študentov sa približne hlási na seminár z informatiky (samostatne pre 3. a samostatne pre 4. ročník)
- Robievajú študenti na seminári z informatiky nejaké projekty? (medzipredmetové, či informatické)
- 1 počítač na 1 žiaka? Všetky hodiny sú pri počítačoch?
- Ako dlho a kedy sa venujete programovaniu, čo stihnete prebrať?
- Semináristi – v akom jazyku programujú, s čím majú najviac problémov, do akej úrovne programovania sa dostane koľko percent študentov (najlepšie sú hranice 0, 20, 50, 80, 100)?
- Ako sa učia programovať? Skúšajú si svoj algoritmus fyzicky? Ak áno, spravia to raz a zabudnú, alebo si pri nových témach znova pripomenú aktivitu? Ak vysvetľujú algoritmus, vysvetľujú ho ako preklad jazyka alebo povedia význam časti kódu?

- Netiketa, kedy a koľko sa jej venujú, čo z nej?
- Textový formát, kedy, koľko, akým štýlom,
- Tabuľky, kedy, koľko, akým štýlom,
- Prezentácie, kedy, koľko, akým štýlom,
- Databázy, ak áno, kedy, koľko, akým štýlom,
- Tvorba webstránok, kedy, koľko, akým štýlom,
- Vyhľadávanie na internete,
- Grafika – vektorová, rastrová,
- Zvukové formáty,
- E-mail – aj princípy?
- Interaktívna komunikácia – Aj princípy? Protokoly + riziká pre rôzne typy protokolov?
- Šifrovanie, digitálny podpis,
- Reprezentácia čísel; Majú vytvorený vzťah medzi reprezentáciou čísel a chybami pri zaokrúhľovaní, pretečení?
- Kódovanie, kompresia,
- Počítačové systémy, siete.

Z rozhovorov sme sa dozvedeli veľa dôležitých informácií. Niektoré črty mali spoločné všetky rozhovory: učitelia delili informatiku na programovanie a teóriu nielen v rozhovore, ale aj na hodinách; študenti absolvovali pred maturitou vyžadovaných 6 hodín informatiky počas strednej školy, ale nie viac; na seminár z informatiky sa hlásia dvaja až piati študenti z celého ročníka; študenti, ktorí sa nezúčastnia seminára, sa programovaniu venujú mesiac až tri mesiace, a to v pascále.

Dozvedeli sme sa, že študenti majú veľké problémy v programovaní s konceptom premennej a s počítaním súradníc. Problém so softvérom nebýva prakticky vykonať úlohu, ale dodržiavať pravidlá a rozumieť princípom. Niektorí učitelia sa venujú skôr získavaniu praktických skúseností a dodržiavaniu pravidiel, niektorí naopak teórii a na praktické skúsenosti nemajú dost času.

Najdôležitejším zistením bolo, že učitelia a aj žiaci neustále delia informatiku na programovanie a teóriu, napriek tomu, že tieto oblasti sú veľmi prepojené. To nás inšpirovalo k tomu, aby sme témy prepájali a volili poradie tém tak, aby na seba nadväzovali.

7.3 Pozorovanie na seminári z informatiky

V nulte iterácii sme pozorovali vyučovanie seminára z informatiky na bratislavskom gymnáziu bez informatického alebo matematického zamerania. Títo študenti, aby mohli maturovať z informatiky, mali v záverečnom štvrtom ročníku šesť hodín informatiky týždenne, prípadne druhá skupina štyri hodiny týždenne. Dovtedy absolvovali iba dve hodiny týždenne v prvom ročníku gymnázia, kde získavali praktické skúsenosti s aplikačným softvérom a mesiac programovali v pascalle.

Každý týždeň venovali dve hodiny „teórii“ a zvyšný čas programovaniu. V teórii učiteľ vysvetľoval rôzne princípy bez praktických ukážok. Poradie tém sa učiteľ snažil voliť tak, aby na seba nadväzovali.

Zvyšné hodiny sa študenti učili programovať v prostredí Delphi. Väčšina úloh bola zameraná na nejaký výpočet, ale niekoľkokrát sa objavili aj úlohy, kde sa študenti hrali s grafikou.

Učiteľ pravidelne vyzýval študentov, aby sa pri programovaní starali hlavne o funkčnosť svojho programu, nie aby sa hrali s dizajnom. Dizajn si potom mohli upraviť doma, čo mnohí aj robili.

Všetky teoretické hodiny boli vedené formou diskusie, kde rozprával hlavne učiteľ a študenti sa zapájali otázkami. Učiteľ sa venoval detailom, aby každý študent porozumel danej téme do hĺbky. Nebol problém s tým, že by študenti nedokázali počas dvoch vyučovacích hodín udržať pozornosť. Bolo to vďaka zaujímavým informáciám.

Počas celého školského roka prepojil učiteľ vysvetľovanie teórie s programovaním iba raz. Študenti mali naprogramovať aplikáciu, ktorá bude prevádzať čísla z jednej číselnej sústavy do druhej.

Niektoré hodiny sme skúsili suplovať. Na suplovaných hodinách sme zistili, že študenti

mali najradšej úlohy, kde mali simulovať nejaký pohyb. Napríklad ich veľmi zaujali úlohy, keď pomocou programovania vytvárali animáciu krajinky, v ktorej padal sneh a usádzal sa na zem. Druhou animáciou, ktorú sme zadali iba jednému študentovi, bol ohňostroj. Radosť z naprogramovania tejto aplikácie presunul aj na spolužiakov a chválil sa, že vďaka programovaniu bude môcť riešiť aj úlohy z fyziky. Toto poznanie sme považovali za úspech, pretože je dôležité, aby študenti vedeli, že počítače majú veľkú silu v tom, že vďaka nim dokážeme pomerne ľahko simulovať problémy z rôznych oblastí života.

Zistili sme tiež, že na suplovaných hodinách už neboli takí nadšení, ak mali riešiť matematické úlohy, ktoré neboli zadané dostatočne atraktívne. Napríklad úlohy na zistenie prvočíselnosti považovali iba za nutnosť, aby sa naučili programovať. Zlepšenie nastalo, keď sme matematické úlohy obalili rozprávkou. Namiesto zistenia súčtu čísel študenti programovali výpis trojuholníkových čísel alebo Fibbonacciho čísla programovali potom, ako si vypočuli rozprávku o množení zajačikov.

Na maturitnej skúške nenastali veľké problémy. Zistili sme, že jeden zo študentov nepreukázal pochopenie funkcií operačného systému, pretože rozprával hlavne o klávesových skratkách. Niekoľko študentov nezvládlo programátorskú časť maturity, až na jeden prípad to bolo hlavne z časového hľadiska. Jeden študent, ktorý mal problémy aj na hodinách, neporozumel koncept cyklu.

7.4 Pozorovanie a výučba seminára z informatiky

V prvej a druhej iterácii výskumu sme vyučovali prvú časť seminára z informatiky na bratislavskom gymnáziu bez informatického alebo matematického zamerania. Postupne sme vytvárali časovo-tematický plán, súbor aktivít ako aj metodiku pre vyučovanie informatiky. Všetky tieto výsledky sú navrhnuté špeciálne pre vybranú skupinu študentov. Veríme však, že učiteľ si môže metodiku upraviť pre potreby svojej skupiny, či sa inšpirovať z našich výsledkov.

Študenti z našej výskumnej vzorky absolvovali doposiaľ dve hodiny týždenne v prvom ročníku strednej školy. Keďže maturovať môžu podľa vyhlášky o ukončovaní štúdia [Vyh08]

až po absolvovaní šiestich vyučovacích hodinách týždenne, znamená to, že musia absolvovať ešte štyri hodiny týždenne, aby boli pripustení k maturitnej skúške. V treťom ročníku majú zapísaný seminár v rozsahu dve hodiny týždenne a vo štvrtom ročníku taktiež dve hodiny týždenne.

Títo študenti majú motiváciu venovať sa informatike a majú dobre vyvinuté analytické myslenie. Vďaka tomu budeme môcť na seminári postupovať rýchlejšie, ako by sme postupovali u študentov, ktorí sa informatike nechcú venovať.

7.4.1 Návrh, implementácia a analýza časovo-tematického plánu

Aby sme mohli vytvárať nejakú metodiku, museli sme zistiť vstupné vedomosti študentov. To sa nám podarilo pomocou rozhovorov s ich učiteľkou a rôznych otázok na začiatku školského roka, ale aj počas neho. V tabuľke 7.1 uvádzame zistené informácie a porovnávame ich s výkonovým štandardom podľa Štátneho vzdelávacieho programu [Hau08b]. Poznamenajme, že tento Štátny vzdelávací program je platný až od roku 2008, takže je platný pre študentov, ktorí absolvovali informatiku v rozsahu troch vyučovacích hodín týždenne, čo neplatí pre našu výskumnú vzorku. Použili sme ho, pretože naň nadväzujú cieľové požiadavky na maturitu z informatiky [Špú08] a [Špú10], podľa ktorých budú maturovať naši študenti.

	ŠVP	komentár
okolo nás	dokáže vysvetliť význam pojmov údaj, informácia, digitalizácia, kódovanie, šifrovanie, komprimácia	nedokáže vysvetliť význam pojmov kódovanie a šifrovanie
	pozná princípy kódovania rôznych typov informácie	spĺňa iba pre grafické informácie, avšak nie detailne
	dokáže vysvetliť princíp digitalizácie v závislosti od typu informácie	nedokáže vysvetliť princípy, dokáže však demonštrovať digitalizáciu zvuku a obrazu
	pozná princíp komprimácie dát, používa komprimačný program	používa komprimačný program
	ovláda prevody medzi dvojkovou a desiatkovou číselnou sústavou	neovláda

(pokračuje na nasledujúcej strane)

(pokračovanie z predchádzajúcej strany)

Informácie	pozná druhy aplikácií na spracovanie informácií (podľa typu informácie) a vie charakterizovať ich typických predstaviteľov	spĺňa
	vie vybrať vhodnú aplikáciu v závislosti od typu informácie, vie zdôvodniť výber	spĺňa
	efektívne používa nástroje aplikácií na spracovanie informácií (podľa typu informácie)	spĺňa
	pozná a dodržiava základné pravidlá (formálne, estetické) a odporúčania spracovania rôznych typov informácií	spĺňa
	vie spracovať informácie tak, aby sa neznížila ich informačná hodnota a informácie boli prístupné, použiteľné a jasné	chýba uvedomenie si týchto aspektov
	pozná vlastnosti (výhody, nevýhody) bežných formátov dokumentov v závislosti od typu informácie	nesplňa
	vie demonštrovať možnosti prenosu častí rôznych typov dokumentov medzi rôznymi aplikáciami	spĺňa
	vie použiť jednoduché šifry	nesplňa
Princípy fungovania IKT	dokáže vymenovať jednotlivé časti počítača von Neumannovského typu, pozná ich využitie, princíp fungovania a význam	dokáže vymenovať niektoré časti počítača a pozná ich využitie
	pozná približné kapacity jednotlivých druhov pamätí a obmedzenia ich použitia.	nepozná obmedzenia
	vie vymenovať a charakterizovať základné prídavné zariadenia	nesplňa
	vie charakterizovať operačný systém a efektívne ho používať	
	vie demonštrovať získavanie informácií o systéme, zariadeniach, priečinkoch a súboroch	spĺňa
	dokáže vysvetliť činnosti operačného systému pri práci so súborami a priečinkami	nesplňa
stredníctvom IKT	pozná princípy fungovania internetu (klient-server) a niektoré jeho služby	nesplňa
	pozná možnosti Web2	nesplňa
	pozná princípy a dokáže demonštrovať použitie e-pošty na konkrétnom klientovi	spĺňa
	pozná základné princípy a dokáže demonštrovať použitie interaktívnej komunikácie	spĺňa

(pokračuje na nasledujúcej strane)

(pokračovanie z predchádzajúcej strany)

Komunikácia pro-	využíva služby webu na získavanie informácií	spĺňa
	pozná rôzne spôsoby vyhľadávania informácie (index, katalóg)	nespĺňa
	dokáže vytvoriť webovú prezentáciu využitím služieb internetu	nespĺňa
	pozná a dodržiava pravidlá Netikety	spĺňa
	pozná spôsoby ochrany počítača zapojeného v sieti a osoby na ňom pracujúcej	spĺňa
	pozná internetové nástroje/služby e-spoločnosti	nespĺňa
Postupy, riešenie problémov, algoritmičné myslenie	dokáže analyzovať problém, navrhnúť algoritmus riešenia problému, zapísať algoritmus v zrozumiteľnej formálnej podobe, overiť správnosť algoritmu	nespĺňa
	dokáže riešiť problémy pomocou algoritmov, vie ich zapísať do programovacieho jazyka, hľadať a opravovať chyby	nespĺňa
	rozumie hotovým programom, dokáže určiť vlastnosti vstupov, výstupov a vzťahy medzi nimi, vie ich testovať a modifikovať	nespĺňa
	dokáže riešiť úlohy pomocou príkazov s rôznymi obmedzeniami použitia príkazov, premenných, typov a operácií	nespĺňa
	používa základné typy používaného programovacieho jazyka	nespĺňa
	vie rozpoznať a odstrániť syntaktické chyby, opraviť chyby vzniknuté počas behu programu, identifikovať miesta programu, na ktorých môže dôjsť k chybám počas behu programu	nespĺňa
spoločnosť	pozná súčasné trendy IKT, ich limity a riziká	nepozná limity a riziká
	pozná výhody a možnosti e-vzdelávania a dištančného vzdelávania, pozná možnosti využitia IKT v iných predmetoch	pozná možnosti využitia IKT v iných predmetoch
	dokáže špecifikovať základné znaky informačnej spoločnosti, vymedziť kladné a záporné stránky informačnej spoločnosti	nespĺňa
	vie charakterizovať jednotlivé typy softvéru z hľadiska právnej ochrany (freeware, shareware, ...) a rozumie, ako sa dajú používať	spĺňa

(pokračuje na nasledujúcej strane)

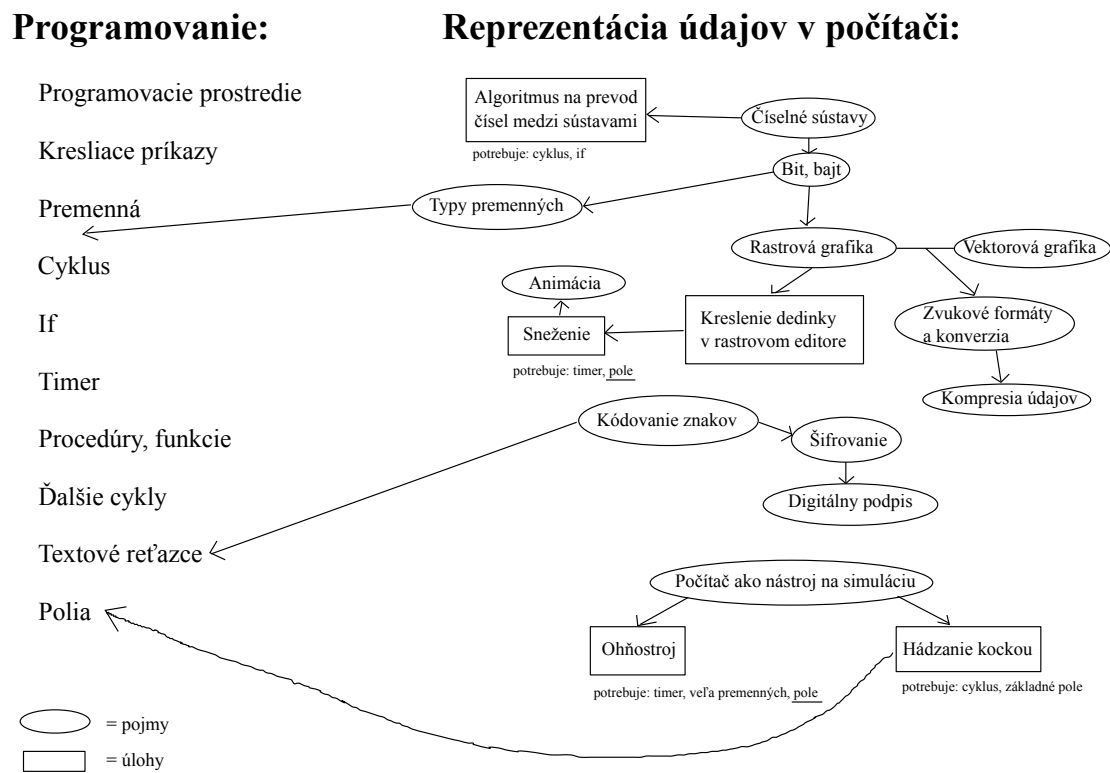
(pokračovanie z predchádzajúcej strany)

Informačná	chápe potrebu právnej ochrany programov	nesplňa
	vie vysvetliť pojmy "licencia na používanie softvéru", "autorské práva tvorcov softvéru", multilicencia	nesplňa
	dokáže vymenovať jednotlivé typy softvérového pirátstva	nesplňa
	dokáže charakterizovať činnosť počítačových vírusov, vysvetliť škody, ktoré môže spôsobiť a princíp práce antivírusových programov, demonštrovať ich použitie	splňa
	pozná kultúrne, sociálne a zdravotné aspekty používania počítačov a služieb internetu	nesplňa

Tabuľka 7.1: Zistené výkony študentov porovnané s výkonovým štandardom ŠVP

Aby sme usporiadali témy do časovo-tematického plánu, v prvom rade bolo treba zistiť, ako jednotlivé témy na seba nadväzujú, prípadne aké aktivity môžeme pri vyučovaní využiť. Vytvorili sme myšlienkovú mapu 7.1, na ktorej sú nadväznosti znázornené šípkami. V prípade aktivít sme zistili ich prerekvizity, aby sme ich mohli zaradiť do vyučovacieho procesu.

Vychádzajúc z uvedenej myšlienkovvej mapy sme vytvorili časovo-tematický plán platný pre prvú iteráciu výskumu 7.2. Programátorské témy sú zoradené podľa didaktiky programovania [STB10a, STB10b]. Pre každý seminár udávame koncept, ktorý sa študenti naučia používať a tiež obsah seminára.



Obr. 7.1: Myšlienková mapa pre tvorbu časovo-tematického plánu

téma	sem.	koncepty	obsah
úvod do programovania	1.	vývojové prostredie vlastnosti objektov	zoznámenie sa s vývojovým prostredím, kreslenie elíps, obdĺžnikov, nastavenie farby
	2.	premenná	práca s premennou, priradenie, deklarácia, inicializácia
	3.	algoritmus trasovanie	zostavovanie algoritmu v ľudskej reči
reprezentácia údajov v počítači	4.	číselná sústava premenná	prevádzanie čísel z desiatkovej do dvojkovej sústavy, práca s premennou, priradenie, deklarácia, inicializácia, operácie <i>mod</i> a <i>div</i>
	5.	jednotky informácie cyklus	prevádzanie čísel medzi sústavami (10, 2, 8, 16), základné a odvodené jednotky informácie, cyklus s pevným počtom opakovaní
cyklus s pevným počtom opakovaní a vetvenie	6.	riadiaca premenná náhodné čísla grafická plocha	jednoduché použitie riadiacej premennej, generovanie čísel v rozsahu veľkosti grafickej plochy, nastavenie farebných zložiek a náhodnej farby
	7.	vetvenie	programovanie úloh na vetvenie v programe
	8.	vnorený cyklus	programovanie úloh na vnorené cykly a sekvenciu cyklov, použitie vhodného nástroja na zistenie farby obrázka
	9.		programovanie úloh na precvičenie cyklov
	10.	trasovanie	programovanie úloh na precvičenie cyklov, používanie programovania ako prostriedku na riešenie matematických úloh
	11.		programovanie úloh na precvičenie cyklov
časovač	12.	časovač	použitie časovača, používanie programovania ako prostriedku na tvorbu hier

(pokračuje na nasledujúcej strane)

(pokračovanie z predchádzajúcej strany)

časovač	13.		práca s obrázkami
	14.		vytváranie hry na tréning práce s časovačom, premennými a obrázkami
funkcia	15.	funkcia reprezentácia reálnych čísel	programovanie pomocou funkcií, práca s reálnymi číslami, získanie skúsenosti s problémami s porovnávaním reálnych čísel vyplývajúce z ich reprezentácie, používanie programovania ako prostriedku na simuláciu problémov
	16.	procedúra	programovanie úloh na opakovanie práce s cyklami a vetvením, použitie procedúr a funkcií, práca s obrázkami a zvukom
reprezentácia obr. údajov	17.	reprezentácia obrazových údajov	programovanie úloh vyžadujúcich vnorené cykly a vetvenie, práca s obrázkom a obrazovými bodmi
podmienený cyklus	18.	podmienený cyklus	programovanie úloh na podmienený cyklus, používanie programovania ako prostriedku k riešeniu matematických úloh
	19.		programovanie úloh na podmienený cyklus
	20.	konzolová aplikácia	práca s konzolovou aplikáciou, tvorba algoritmu na hľadanie minima, získanie skúsenosti s prekročením rozsahu celočíselných premenných
reprezentácia textových údajov	21.	znak kódovanie znakov	tvorba algoritmu na prevod medzi číselnými sústavami, programovanie úloh na prácu so znakmi, práca s textovým editorom a kódovaním znakov (ASCII, UTF, iné)
	22.	textový súbor udalosti klávesnice	programovanie s textovými súbormi, programovanie aplikácie na odchyťovanie stlačených kláves
	23.	znakové reťazce šifrovanie	programovanie úloh na spracovanie reťazcov a jednoduché šifrovanie

(pokračuje na nasledujúcej strane)

(pokračovanie z predchádzajúcej strany)

reprezentácia textových údajov	24.	šifrovanie bezpečnosť	metódy šifrovania a dešifrovania, pochopenie rizík práce s internetom (šifrovanie komunikácie, certifikát, digitálny podpis)
	25.		programovanie úloh na spracovanie reťazcov, šifrovanie, frekvenčnú analýzu
jednorozmerné pole	26.	jednorozmerné pole	programovanie úloh s jednoduchým použitím jednorozmerného poľa
	27.		programovanie úloh na použitie jednorozmerného poľa ako počítadla, používanie programovania ako prostriedku na simuláciu problémov
	28.	udalosti myši	programovanie komplexného programu na kreslenie, využitie jednorozmerného poľa na funkciu <i>undo</i>

Tabuľka 7.2: Časovo tematický plán hodín informatiky v prvej iterácii

Jedným z nedostatkov uvedeného plánu bol aj fakt, že výskum sme začali robiť až na treťom seminári, keď už začala prebiehať výučba.

Napriek tomu, že tento časový plán sme upravovali počas školského roku na základe napredovania študentov, mal mnoho problémov. Najväčší problém bol rýchlosť výučby. Keďže študenti sa rozhodli maturovať z informatiky a mali dobre rozvinuté analytické myslenie, naivne sme predpokladali, že napríklad pri téme cyklus s pevným počtom opakovaní nebude potrebné nielen dôsledne trénovať poznatok, ale tiež že poznávací proces treba aplikovať aj na jednotlivé „podkoncepty“, napríklad pre cyklus s pevným počtom opakovaní je podkonceptom riadiaca premenná, či vnorené cykly.

Ďalším problémom, ktorý nastal bolo, že študenti sa nedostali na dostatočne vysokú úroveň podľa Bloomovej taxonómie pri ladení programu pomocou trasovania, dostali sa iba na úroveň porozumenia, iba v prípade šikovnejších študentov na úroveň aplikácie. Toto bolo treba napraviť v ďalšej iterácii.

Ak sa pozrieme na uvedený časovo-tematický plán, najväčšie nedostatky sa nachádzali v prvej časti plánu. Je to preto, že sme ešte nemali dostatočné skúsenosti s učením takejto skupiny. Neskôr sa nám podarilo lepšie pochopiť ich myslenie. Ďalším dôvodom je aj fakt, že v tomto roku sa na seminár prihlásil neobvyklý počet študentov, ktorí za informatiku považovali hlavne používanie aplikačného softvéru, napríklad úpravy fotografií a videa a maturitu z informatiky brali ako ľahkú cestu, pretože s počítačom vedia narábať. Aj keď sme boli sklamaní, že pár študentov seminár na začiatku vzdalo, zistili sme, že to sa deje každý rok. Toto je jeden, ale nie jediný dôvod, prečo odporúčame vyučovať tematický okruh riešenia problémov aj v nižších ročníkoch. Neznamená to však, že ich v nižších ročníkoch máme odradiť, práve naopak. Len je potrebné, aby už pri špecializovaní sa vedeli, čo je súčasťou informatiky.

Po dôkladnej analýze pozorovaní sme zmenili časovo-tematický plán platný pre ďalšiu iteráciu výskumu. Zmenili sme nielen časový plán pre jednotlivé témy, ale sme upravili aj súbor aktivít podľa zistených skutočností. Nový časovo-tematický plán uvádzame v tabuľke 7.3.

téma	sem.	koncepty	obsah
úvod do programovania	1.	vývojové prostredie vlastnosti objektov	zoznámenie sa s vývojovým prostredím, kreslenie elíps, obdĺžnikov, nastavenie farby
	2.	algoritmus trasovanie	zostavovanie algoritmu v ľudskej reči, kreslenie elíps podľa daných pravidiel
reprezentácia údajov v počítači	3.	číselná sústava premenná	prevádzanie čísel z desiatkovej do dvojkovej sústavy, práca s premennou, priradenie, deklarácia, inicializácia, operácie <i>mod</i> a <i>div</i>
	4.	jednotky informácie	prevádzanie čísel medzi sústavami (10, 2, 8, 16), základné a odvodené jednotky informácie
	5.	náhodné čísla grafická plocha	generovanie čísel v rozsahu veľkosti grafickej plochy, nastavenie farebných zložiek a náhodnej farby, výmena obsahu dvoch premenných
cyklus s pevným počtom opakovaní	6.	cyklus riadiaca premenná	cyklus s pevným počtom opakovaní, jednoduché použitie riadiacej premennej
	7.	vnorený cyklus	programovanie úloh na vnorené cykly a sekvenciu cyklov, ladenie programu pomocou trasovania
	8.		programovanie úloh na vnorené cykly a sekvenciu cyklov, použitie vhodného nástroja na zistenie farby obrázka
	9.	riadiaca premenná cyklu	zložité použitie riadiacej premennej cyklu, hľadanie zákonitostí v obrazcoch
	10.		programovanie úloh na precvičenie cyklov, používanie programovania ako prostriedku na riešenie matematických úloh

(pokračuje na nasledujúcej strane)

(pokračovanie z predchádzajúcej strany)

vetvenie	11.	vetvenie časovač	programovanie úloh na vetvenie s jednoduchými podmienky, použitie časovaču, používanie programovania ako prostriedku na tvorbu hier
	12.	logické spojky	booleovské skladanie podmienok, výpočet skóre, práca s obrázkami
	13.		vytváranie hry na tréning práce s časovačom, premennými a obrázkami
funkcia	14.	funkcia reprezentácia reálnych čísel	programovanie pomocou funkcií, práca s reálnymi číslami, získanie skúsenosti s problémami s porovnávaním reálnych čísel vyplývajúce z ich reprezentácie, používanie programovania ako prostriedku na simuláciu problémov
reprezentácia obrazových údajov	15.	reprezentácia obrazových údajov	programovanie úloh vyžadujúcich vnorené cykly a vetvenie, práca s obrázkom a obrazovými bodmi
	16.	grafické formáty	grafické formáty
podmienený cyklus	17.	podmienený cyklus	programovanie úloh na podmienený cyklus, používanie programovania ako prostriedku k riešeniu matematických úloh, ladenie chýb pomocou trasovania
	18.	konzolová aplikácia	práca s konzolovou aplikáciou, tvorba algoritmu na hľadanie minima, získanie skúsenosti s prekročením rozsahu celočíselných premenných
reprezentácia textových údajov	19.	znak kódovanie znakov	tvorba algoritmu na prevod medzi číselnými sústavami, programovanie úloh na prácu so znakmi, práca s textovým editorom a kódovaním znakov (ASCII, UTF, iné)
	20.	textový súbor udalosti klávesnice	programovanie s textovými súbormi, programovanie aplikácie na odchyťovanie stlačených kláves

(pokračuje na nasledujúcej strane)

(pokračovanie z predchádzajúcej strany)

reprezentácia textových údajov	21.	znakové reťazce šifrovanie	programovanie úloh na spracovanie reťazcov a jednoduché šifrovanie
	22.	šifrovanie bezpečnosť	metódy šifrovania a dešifrovania, pochopenie rizík práce s internetom (šifrovanie komunikácie, certifikát, digitálny podpis)
	23.		programovanie úloh na spracovanie reťazcov, šifrovanie, frekvenčnú analýzu
jednorozmerné pole	24.	jednorozmerné pole	programovanie úloh s jednoduchým použitím jednorozmerného poľa
	25.		programovanie úloh na použitie jednorozmerného poľa ako počítadla, používanie programovania ako prostriedku na simuláciu problémov

Tabuľka 7.3: Časovo tematický plán hodín informatiky v druhej iterácii

Zistili sme, že uvedený plán mal podstatne menej nedostatkov. Z riešení domácich úloh a programov použitých na overovanie vedomostí sme zistili, že problémy, ktoré nastávajú, sú už oveľa menej významné, ako v predchádzajúcom pláne.

Pozitívom boli viaceré zistenia:

- študenti dokázali v priebehu výskumu odpovedať na otázku, prečo sa v premennej nachádza číslo, ktoré sa tam nachádzať nemalo. Bolo to spôsobené pretečením rozsahu. Rozsah premennej typu integer sme pritom spomenuli niekoľko mesiacov pred touto udalosťou,
- keď sme sa začali venovať téme podmienený cyklus, stačilo študentom vysvetliť syntax a použitie príkazu. Zvolené úlohy narastajúcej obtiažnosti dokázali študenti vyriešiť veľmi rýchlo, nemali ani problém s úlohami vyžadujúcimi zložitejší algoritmus a zložitejšie podmienky. Myslíme si, že to znamená, že zvládli na dostatočne vysokej úrovni koncept cyklu, a aj vytvárania podmienok,
- veľmi rýchlo dokázali riešiť úlohy, kde bolo treba zväčšovať premenné. Musíme však poznamenať, že nech sme sa snažili akokoľvek, problémom bola inicializácia premennej, na ktorú veľakrát zabudli,
- aj keď sme na 23. seminári programovali frekvenčnú analýzu na texte, študent použil rovnaký pojem aj pri riešení problému štatistiky hádzaní kocky. Väčšina žiakov túto úlohu naprogramovala v priebehu 5 minút, napriek tomu, že s poľom pracovali iba jednu vyučovaciu hodinu, a to iba s jednoduchým prístupom k prvku na zadanom indexe.

Plán mal aj niekoľko negatív:

- študenti z tejto výskumnej vzorky ovládali grafické formáty na vysokej úrovni, dokázali si vybrať vhodný formát v závislosti od použitia. Keďže sme sa pomocou programovania venovali reprezentácii grafickej informácie, snažili sme sa, aby nielen rozumeli princípu reprezentácie obrázkov, ale aj dokázali aplikovať nové poznatky. Vďaka praktickým skúsenostiam zo štúdia v nižších ročníkoch toto dokázali aj bez našej pomoci. Jedna vyučovacia hodina venovaná tejto téme bola teda viac-menej zbytočná,

- niektorí študenti si nezvykli písať kód štruktúrovane. Bolo by dobré sa viac venovať funkciám a procedúram, ale nie na samostatnej hodine.

Na overenie vedomostí sme použili niekoľko metód. Analyzovali sme programy, ktoré študenti vytvorili doma alebo na hodinách. Výsledky sme zahrnuli do predchádzajúceho zoznamu pozitív a negatív plánu a aj metodiky. Študenti tiež vyplnili vedomostno-postojový dotazník na začiatku a konci školského roka, ktorý uvádzame v prílohe. Zistili sme, že aj keď študenti na začiatku roku radi programovali s grafikou, neskôr, keď nadobudli dostatočné schopnosti, radšej si vybrali konzolovú aplikáciu, ktorá im umožnila problémy riešiť rýchlejšie. Niektorí študenti tiež nemali dostatočne trvácny poznatok o kódovaní farby. Ostatné otázky študentom nerobili problém.

Porovnajme, aké výkony dosahujú študenti pred kurzom, po kurze a čo majú splňať podľa cieľových požiadaviek na maturitu z informatiky [Špú08]. Síce v platnosti sú už novšie požiadavky na maturitu z informatiky, ale nebolo tak na začiatku nášho výskumu. Taktiež treba pripomenúť, že títo študenti zatiaľ absolvovali iba dve hodiny týždenne z informatiky na strednej škole a jednu hodinu týždenne na základnej škole, čo pri takejto časovej dotácii nestačí na dosiahnutie všetkých požiadaviek novších cieľových požiadaviek na maturitu z informatiky [Špú10].

	pred kurzom	po kurze	cieľové požiadavky
	dokáže vysvetliť princípy, výhody a nevýhody rastrových a vektorových obrázkov	dokáže vysvetliť reprezentáciu rôznych číselných typov v počítači, výhody a nevýhody niektorých najbežnejších grafických formátov (bmp, jpeg, gif, ...), princípy, výhody a nevýhody rastrových a vektorových obrázkov	dokáže vysvetliť reprezentáciu rôznych číselných typov v počítači, výhody a nevýhody niektorých najbežnejších grafických formátov (bmp, jpeg, gif, ...), princípy, výhody a nevýhody rastrových a vektorových obrázkov, princípy kódovania zvuku v súboroch typu (wave, midi, mp3, ...)

(pokračuje na nasledujúcej strane)

(pokračovanie z predchádzajúcej strany)

Informácie okolo nás	vie demonštrovať postup digitalizácie zvuku a jeho ďalších úprav	vie demonštrovať postup digitalizácie zvuku a jeho ďalších úprav	dokáže vysvetliť princíp digitalizácie zvuku a jeho ďalších úprav
		vie demonštrovať použitie rôznych číselných sústav	vie demonštrovať použitie rôznych číselných sústav
	dokáže vysvetliť základné druhy aplikačného softvéru a charakterizovať ich typických predstaviteľov a oblasti využitia	dokáže vysvetliť základné druhy aplikačného softvéru a charakterizovať ich typických predstaviteľov a oblasti využitia	dokáže vysvetliť základné druhy aplikačného softvéru a charakterizovať ich typických predstaviteľov a oblasti využitia
	vie zdôvodniť výber príslušného softvéru pre určitú množinu úloh	vie zdôvodniť výber príslušného softvéru pre určitú množinu úloh	vie zdôvodniť výber príslušného softvéru pre určitú množinu úloh
	efektívne využíva nástroje textového editora na vytvorenie a úpravu dokumentu, efektívne využíva nástroje tabuľkového kalkulátora na vytvorenie a úpravu tabuľky, vie vytvoriť vhodnú grafickú reprezentáciu z údajov v tabuľke, používa tabuľkový kalkulátor pri spracovaní jednoduchých úloh databázového charakteru, triedi a filtruje údaje, efektívne využíva nástroje vhodného aplikačného softvéru na vytvorenie prezentácie a jej predvedenie, efektívne využíva nástroje grafického editora na vytvorenie a úpravu rastrového obrázka,	efektívne využíva nástroje textového editora na vytvorenie a úpravu dokumentu, efektívne využíva nástroje tabuľkového kalkulátora na vytvorenie a úpravu tabuľky, vie vytvoriť vhodnú grafickú reprezentáciu z údajov v tabuľke, používa tabuľkový kalkulátor pri spracovaní jednoduchých úloh databázového charakteru, triedi a filtruje údaje, efektívne využíva nástroje vhodného aplikačného softvéru na vytvorenie prezentácie a jej predvedenie, efektívne využíva nástroje grafického editora na vytvorenie a úpravu rastrového obrázka,	efektívne využíva nástroje textového editora na vytvorenie a úpravu dokumentu, efektívne využíva nástroje tabuľkového kalkulátora na vytvorenie a úpravu tabuľky, vie vytvoriť vhodnú grafickú reprezentáciu z údajov v tabuľke, používa tabuľkový kalkulátor pri spracovaní jednoduchých úloh databázového charakteru, triedi a filtruje údaje, efektívne využíva nástroje vhodného aplikačného softvéru na vytvorenie prezentácie a jej predvedenie, efektívne využíva nástroje grafického editora na vytvorenie a úpravu rastrového obrázka,

(pokračuje na nasledujúcej strane)

(pokračovanie z predchádzajúcej strany)

	vie demonštrovať prenos častí rôznych typov dokumentov medzi rôznymi aplikáciami	vie demonštrovať prenos častí rôznych typov dokumentov medzi rôznymi aplikáciami	vie demonštrovať prenos častí rôznych typov dokumentov medzi rôznymi aplikáciami
		má prvú predstavu o kóde HTML na webovej stránke	dokáže vytvoriť webovú stránku v rôznych aplikáciách (textový editor, tabuľkový kalkulátor, ...)
Princípy fungovania IKT	dokáže vymenovať niektoré časti počítača a pozná ich využitie	dokáže vymenovať niektoré časti počítača a pozná ich využitie	dokáže vymenovať jednotlivé časti počítača von Neumannovského typu
	pozná približne kapacity jednotlivých druhov pamätí	pozná približne kapacity jednotlivých druhov pamätí	dokáže vysvetliť približné kapacity jednotlivých druhov pamätí a obmedzenia ich použitia
			vie vymenovať a charakterizovať základné prídavné zariadenia
			vie charakterizovať operačný systém
	vie demonštrovať získavanie informácií o systéme, zariadeniach, priečinkoch a súboroch	vie demonštrovať získavanie informácií o systéme, zariadeniach, priečinkoch a súboroch	vie demonštrovať získavanie informácií o systéme, zariadeniach, priečinkoch a súboroch
			dokáže vysvetliť činnosti operačného systému pri práci so súbormi a priečinkami
prostredníctvom IKT			dokáže vysvetliť princípy fungovania internetu (klient-server) a niektoré jeho služby
	vie vysvetliť princípy e-pošty a demonštrovať ich na konkrétnom klientovi	vie vysvetliť princípy e-pošty a demonštrovať ich na konkrétnom klientovi	vie vysvetliť princípy e-pošty a demonštrovať ich na konkrétnom klientovi
	dokáže demonštrovať použitie jednej zo služieb rozhovoru (ICQ, jabber)	dokáže demonštrovať použitie jednej zo služieb rozhovoru (ICQ, jabber)	dokáže demonštrovať použitie jednej zo služieb rozhovoru (Talk, IRC, ICQ)

(pokračuje na nasledujúcej strane)

(pokračovanie z predchádzajúcej strany)

Komunikácia	využíva služby webu na získavanie informácií	využíva služby webu na získavanie informácií	využíva služby webu na získavanie informácií
	vie vysvetliť niektoré riziká nedodržania bezpečnosti na internete	vie vysvetliť riziká nedodržania bezpečnosti na internete	vie vysvetliť riziká nedodržania bezpečnosti na internete
Postupy, riešenie problémov, algoritmické myslenie		dokáže riešiť problémy pomocou algoritmov, vie ich zapísať do programovacieho jazyka, hľadať a opravovať chyby	dokáže riešiť problémy pomocou algoritmov, vie ich zapísať do programovacieho jazyka, hľadať a opravovať chyby
		rozumie hotovým programom, dokáže určiť vlastnosti vstupov, výstupov a vzťahy medzi nimi, vie ich testovať a modifikovať	rozumie hotovým programom, dokáže určiť vlastnosti vstupov, výstupov a vzťahy medzi nimi, vie ich testovať a modifikovať
		dokáže riešiť úlohy pomocou príkazov s rôznymi obmedzeniami použitia príkazov, premenných, typov a operácií	dokáže riešiť úlohy pomocou príkazov s rôznymi obmedzeniami použitia príkazov, premenných, typov a operácií
		vie rozpoznať a odstrániť syntaktické chyby, opraviť chyby vzniknuté počas behu programu, identifikovať miesta programu, na ktorých môže dôjsť k chybám počas behu programu	vie rozpoznať a odstrániť syntaktické chyby, opraviť chyby vzniknuté počas behu programu, identifikovať miesta programu, na ktorých môže dôjsť k chybám počas behu programu
		vie pracovať s textovým súborom	vie pracovať s textovým súborom
		používa a definuje procedúry a funkcie bez parametrov aj s parametrami, správne používa lokálne a globálne premenné	používa a definuje procedúry a funkcie bez parametrov aj s parametrami, správne používa lokálne a globálne premenné

(pokračuje na nasledujúcej strane)

(pokračovanie z predchádzajúcej strany)

		dodržiava zásady čitateľného zápisu algoritmu	dodržiava zásady čitateľného zápisu algoritmu
Informačná spoločnosť	vie opísať súčasné trendy IKT	vie vysvetliť súčasné trendy IKT, ich limity a riziká	vie vysvetliť súčasné trendy IKT, ich limity a riziká
		dokáže špecifikovať základné znaky informačnej spoločnosti, vymedziť kladné a záporné stránky informačnej spoločnosti	dokáže špecifikovať základné znaky informačnej spoločnosti, vymedziť kladné a záporné stránky informačnej spoločnosti
	vie charakterizovať jednotlivé typy softvéru z hľadiska právnej ochrany (freeware, shareware, ...) a rozumie, ako sa dajú používať	vie charakterizovať jednotlivé typy softvéru z hľadiska právnej ochrany (freeware, shareware, ...) a rozumie, ako sa dajú používať	vie charakterizovať jednotlivé typy softvéru z hľadiska právnej ochrany (freeware, shareware, ...) a rozumie, ako sa dajú používať
		chápe potrebu právnej ochrany programov a vie vysvetliť dôsledky pri používaní nelegálneho softvéru	chápe potrebu právnej ochrany programov a vie vysvetliť dôsledky pri používaní nelegálneho softvéru
		vie vysvetliť pojmy "licencia na používanie softvéru", "autorské práva tvorcov softvéru", multilicencia	vie vysvetliť pojmy "licencia na používanie softvéru", "autorské práva tvorcov softvéru", multilicencia
	dokáže charakterizovať činnosť počítačových vírusov, vysvetliť škody, ktoré môže spôsobiť a princíp práce antivírusových programov	dokáže charakterizovať činnosť počítačových vírusov, vysvetliť škody, ktoré môže spôsobiť a princíp práce antivírusových programov	dokáže charakterizovať činnosť počítačových vírusov, vysvetliť škody, ktoré môže spôsobiť a princíp práce antivírusových programov
		vie zdôvodniť potrebu ochrany údajov	vie zdôvodniť potrebu ochrany údajov

Tabuľka 7.4: Porovnanie výkonov študentov pred kurzom, po kurze a s cieľovými požiadavkami na maturitu

7.4.2 Návrh, implementácia a analýza vybraných tém

Jedným z cieľov tejto práce bolo vytvoriť metodiku vybraných tém. Z dôvodu rozsahu práce sme vybrali dve témy: dvojková sústava a jednotky informácie a znakové reťazce a bezpečnosť.

Pre každú tému špecifikujeme ciele výučby, navrhujeme možný spôsob realizácie, popíšeme jednotlivé aktivity a ich riziká a nakoniec analyzujeme naplnenie špecifikovaných cieľov a cieľových požiadavok na maturitu z informatiky.

Ciele výučby budeme formulovať, pokiaľ je tak možné, rovnako, ako sú formulované v cieľových požiadavkách na maturitu z informatiky [Špú08], [Špú10], aby sme demonštrovali naplnenie týchto cieľov v príprave na maturitu.

Dvojková sústava a jednotky informácie

Ciele výučby

- zdôvodniť používanie dvojkovej, osmičkovej a šestnástkovej sústavy v kontexte ich používania v počítačovom systéme, ukázať na príklade prevody medzi nimi,
- demonštrovať aritmetické operácie v dvojkovej sústave, zdôvodniť a na príkladoch dokumentovať vlastnosti aritmetiky počítača,
- demonštrovať princíp výpočtu veľkosti pamäte (v bitoch, bajtoch a vyšších jednotkách), času (aj pri prenose údajov) na jednoduchom príklade,
- vysvetliť reprezentáciu rôznych celočíselných typov v počítači, vypočítať veľkosť alokovanej pamäte pri použití dátových štruktúr daných typov a vypočítať interval možných hodnôt pre celočíselné typy premennej.

Realizácia

Samotná téma číselné sústavy môže učiteľov motivovať k tomu, aby použil aktivizujúce metódy, pretože hier k tejto téme existuje veľa. Treba mať však na zreteli, že vyhláška o ukončovaní štúdia [Vyh08] povoľuje študentovi maturovať po absolvovaní aspoň 6 hodín informatiky týždenne (počas strednej školy) a mnohé školy informatike viac času neposkytnú. To znamená, že vo fáze prípravy na maturitu treba vyberať hry, ktoré naplnia

ciele efektívne vzhľadom na čas strávený realizáciou a ich prínos. My sme aktivity zvolili v nasledujúcom poradí:

- (1) Kartičky
- (2) Binárny kamzík
- (3) Rátanie na prstoch

Tieto aktivity aj s príslušnými otázkami sa dajú realizovať v rozsahu jednej vyučovacej hodiny a študenti odchádzajú domov s domácou úlohou precvičiť si prácu s dvojkovou sústavou. Počas nasledujúcich 45 minút majú hodinu programovania, kedy sa učia pracovať s premennými. Keďže už rozumejú pojmu bit a počet bitov, je pre nich jednoduchšie pochopiť rozdiely medzi rôznymi typmi premenných.

Ďalších 90 minút je zvolených nasledovne:

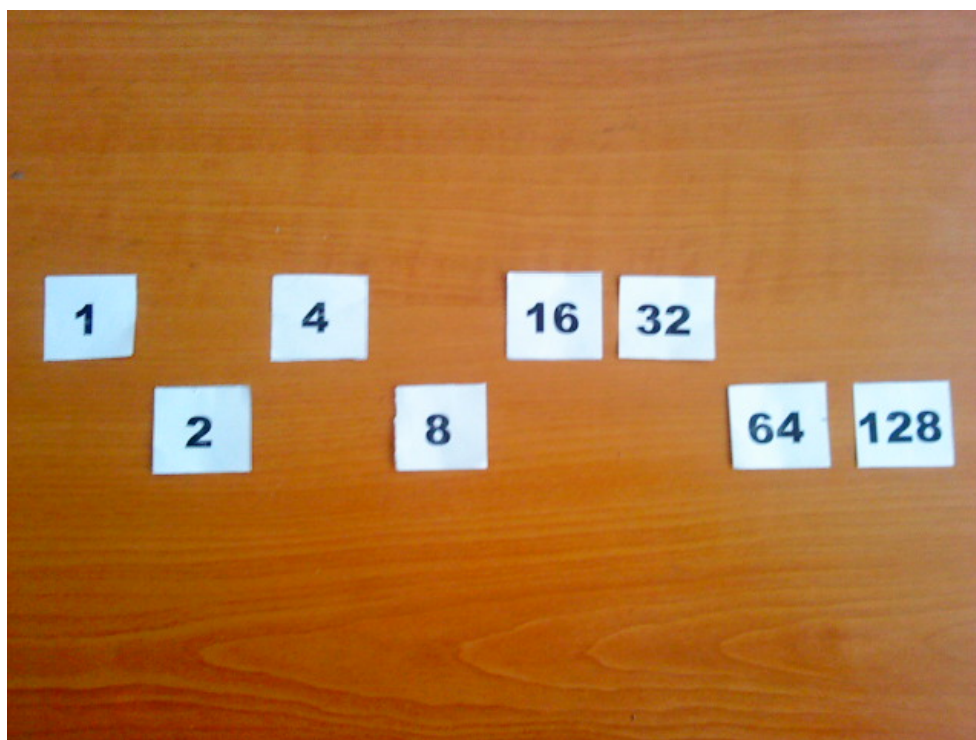
- (4) Opakovanie rátania na prstoch
- (5) Prevod medzi dvojkovou, šestnástkovou a osmičkovou sústavou
- (6) Definovanie jednotky byte, prevody medzi jednotkami bit, byte a vyššími jednotkami

Aj keď by aktivity (5) a (6) mohli vyvolať dojem, že študent je len pasívnym poslucháčom, nemusí to tak byť. Študenta zapájame pri precvičovaní prevodu medzi číselnými sústavami a vhodnými otázkami pri veľkostiach informácie, pretože sa s nimi už v živote pravdepodobne stretol pri parametroch počítača. Uvedený súbor aktivít sme navrhli tak, aby sa ciele naplňali postupne. Aktivity sme vybrali rôznymi metódami pedagogického výskumu: štúdiom odbornej literatúry (v prípade aktivity 1), pozorovaním na škole (aktivity 5, 6), skúsenosťami z mimoškolských aktivít (2, 3), a tiež vlastnými skúsenosťami z pedagogickej praxe (1).

Popis aktivít a naplnenie čiastkových vzdelávacích cieľov

(1) Kartičky

Táto aktivita vychádza z aktivity CS Unplugged [BWF98] a použili ju lektori vzdelávania DVUI v miernej úprave [GSS09]. Každý študent dostane kartičky s číslami 1, 2, 4,



Obr. 7.2: Aktivita kartičky zo vzdelávania DVUI [BWF98]

8, 16, 32, 64 a 128. Učiteľ zadá číslo a študenti musia vybrať kartičky tak, aby ich súčet bol rovný tomuto číslu. Na obrázku 7.2 môžeme vidieť číslo 53.

Počas tejto aktivity študent vytvorí algoritmus pre prevádzanie čísla z desiatkovej do dvojkovej sústavy. Výhodou je tiež to, že nevadí, ak niektorý študent vymyslí algoritmus skôr a niektorý neskôr, pretože každý má vlastné kartičky a súčet vytvárajú samostatne, pričom učiteľ chodí po triede a námatkovo kontroluje. Nestáva sa, že by šikovnejší študent vykrikol skôr svoj výber kartičiek, či nebodaj algoritmus.

Uvedomme si, že študenti takto nevytvoria všeobecný algoritmus na prevod čísel medzi sústavami, ktorý nájdeme v učebnici informatiky [K⁺05]. Algoritmus, ktorý vytvorí, je pre konkrétny prípad prevodu z desiatkovej do dvojkovej sústavy. Tento algoritmus je intuitívnejší a aj rýchlejší, než všeobecný algoritmus. Všeobecný algoritmus sa môžu naučiť neskôr, napríklad pri programovaní. V tejto fáze vzdelávania im môžeme ponechať konkrétny algoritmus, z ktorého budú mať radosť, lebo ho sami vymysleli, čo ich motivuje

k ďalšej práci. Koniec aktivity môžeme doplniť vhodnými otázkami, napríklad „*Aké najväčšie číslo dokážete z takýchto kartičiek spraviť?*“. V tejto aktivite však ešte nezavedieme pojem dvojková sústava, pretože zatiaľ študenti nevedia (ale niektorí si už všimli), že čísla na kartičkách sú schválne špeciálne zadané ako mocniny čísla 2.

(2) *Binárny kamzík*

S aktivitou binárny kamzík sa študenti stretávajú na sústrezeniach Korešpondenčného seminára z programovania [SŠ10]. Taktiež sa s ňou mohli stretnúť učitelia na vzdelávaní DVUI [BBC⁺09b]. Hra je fyzického druhu a študenti počas nej robia drepy. Je vhodné odhadnúť študentov a fyzicky menej zdatných poslať na správnu stranu radu, pretože veľká záťaž by mohla slabších študentov diskvalifikovať. Teraz nastáva chvíľa, kedy učiteľ môže na tabuli ukázať, ako vyzerá číslo v dvojkovej sústave. Študenti zistia, že čísla na kartičkách neboli volené náhodne, ale sú to mocniny dvojky. Učiteľ môže povedať, že v počítači znamená zapnutý bit „*Je tam napätie,*“ a vypnutý bit „*nie je tam napätie.*“

Študenti sa postavia vedľa seba. Každý z nich predstavuje jeden bit. Ak stojí, bit je zapnutý, ak je v drepe, tak je vypnutý. Študenti sú očíslovaní, takže vpravo sa nachádza človek, ktorý predstavuje posledný bit čísla v dvojkovej sústave, vľavo od neho predposledný, vľavo predpredposledný, Učiteľ hovorí náhodné čísla v desiatkovej sústave a študenti sa podľa toho upravia. Časom učiteľ začne čísla voliť nie náhodne, ale tak, aby ukázal, ako ľahko sa v dvojkovej sústave násobí a delí dvojkou, pretože zapnuté bity sa iba posunú. Týmto učiteľ zavedie pojem bitový posun a študentom ukáže efektívnosť tejto operácie. Po niekoľkých číslach študenti skúsia vytvárať postupnosť čísel 0, 1, 2, 3, 4, 5, 6,

V tejto fáze začnú pracovať so sebou ako samostatným bitom a získavajú prvé skúsenosti s počítaním v dvojkovej sústave. Musia zistiť, ako majú robiť drepy, napríklad tretí bit sprava strieda pozície štyrikrát dole, štyrikrát hore. Všimnú si, že po čísle, kde boli za sebou bity zapnuté, nasleduje číslo, kde je jeden bit zapnutý a všetky za ním vypnuté. Po vykonaní drepop sa učiteľ môže pýtať otázky: „*Aký interval čísel dokážete vytvoriť, keby ste boli piati?*“, „*Aký interval dokážete vytvoriť keby vás bolo N ?*“, „*Čo by ste spravili, keby*

ste chceli reprezentovať aj záporné čísla?“

(3) Rátanie na prstoch

Po fyzickej aktivite si študenti zaslúžia odpočinok. Nasledujúca aktivita už patrí iba do fázy automatizácie a vytvára priestor pre domáce cvičenie, pretože potrebné nástroje má človek aj v autobuse cestou zo školy. Aktivita je veľmi podobná binárnemu kamzíkovi, ale na reprezentáciu bitov bude používať prsty. Ak je prst vystretý, reprezentuje zapnutý bit a ak je skrčený, reprezentuje vypnutý bit. Učiteľ volí náhodné čísla a študent ich ukazuje na prstoch.

Všimnime si, že občas študenti ukážu na prstoch nevhodné gesto, čo sa prejaví smiechom v triede. Skúsenejší učiteľ je však obozretný a dokáže vhodne zareagovať. Často práve táto nevhodnosť dokáže študentov motivovať k tréningu mimo školy.

(4) Opakovanie rátania na prstoch

Po týždni je vhodné dvojkovú sústavu pripomenúť, keďže získané kompetencie sú potrebné pre ďalšie aktivity a pochopenie súvislostí. Na to sme vybrali zopakovanie aktivity s prstami, pretože je to časovo nenáročná a každý študent si precvičí prevody bez toho, aby ho predstihol šikovnejší spolužiak. Je vhodné, ak učiteľ nakreslí na tabuľu číslo v dvojkovej sústave aj s príslušnými mocninami, pretože to bude potrebné pre ďalšiu aktivitu.

(5) Prevod medzi dvojkovou, šestnástkovou a osmičkovou sústavou

Prvá fáza poznávacieho procesu je motivácia. Je vhodné ňou začať aj pri vysvetľovaní prevodov. Aká je motivácia pre prevod do šestnástkovej sústavy? Jedna z možností vychádza z histórie. Kedysi programátori nepísali tak ako my do počítača slová v programovacom jazyku, ale zapisovali čísla v šestnástkovej sústave. Síce počítač pracuje s dvojkovou sústavou, ale keby mal programátor prepisovať postupnosť 0 a 1, bolo by to pomalšie a pravdepodobne by sa pomýlil. Oveľa ľahšie mohol prepisovať postupnosť cifier šestnástkovej sústavy. Keď už sú študenti motivovaní na prevod do šestnástkovej sústavy, nič im nebráni sa to naučiť. Je vhodné im ukázať, že štyri bity z dvojkovej sústavy tvoria jednu cifru v šestnástkovej. Rovnaký postup platí aj pre prevod do osmičkovej sústavy. Tam môže

byť motiváciou to, že cifry v šestnástkovej sústave sa nedajú vyjadriť na prstoch.

Ak budú študenti potrebovať prevádzať medzi spomínanými sústavami (so základom 2, 8, 10, 16) najrýchlejší je prechod do dvojkovej sústavy. To potrebujú vedieť na maturitnej skúške, kde hrá úlohu aj časový faktor.

(6) *Definovanie jednotky byte, prevody medzi jednotkami bit, byte a vyššími jednotkami*

Teraz, keď už študenti vedia prevádzať z dvojkovej sústavy do šestnástkovej a osmičkovej, je vhodný čas na zavedenie pojmu *byte*. Ukazuje sa, že študenti nemajú problém ani o niekoľko mesiacov, aké matematické operácie použiť na prevod medzi jednotkami informácie. To je rozdiel oproti výučbe memorovaním, keď po nejakej dobe študent nevie, či pri výpočte mal použiť delenie alebo násobenie a má deliť číslom 3, 8 alebo 256. Rovnako ako prevod do šestnástkovej sústavy znamenal rozdeliť číslo po blokoch veľkosti štyri, teraz bude študent deliť číslo na bloky veľkosti osem. Prospešné je tiež ukázať, ako vyzerá kód v šestnástkovej sústave. Je zapísaný po dvojiciach cifier, kde každá cifra zaberie štyri bity pamäte.

Pre naplnenie špecifikovaných cieľov je potrebné vysvetliť aj prevod medzi vyššími jednotkami. Aby učiteľ študentov motivoval, je lepšie látku nielen vysvetliť, ale aj ju doplniť nejakými výzvami, napríklad „*Aká je ďalšia predpona v postupnosti: kilo, mega, giga? Viete ešte ďalšiu? Dnes sa udávajú veľkosti diskov v terabajtoch, ale napríklad úložisko firmy Google už disponuje pamäťou s väčšou predponou. Aká to je?*“ alebo „*Nie, že mi poviete na maturite jednotku mikrobajt, taká nedáva zmysel, viete prečo?*“ (zdroj anonymný učiteľ), či zaujímavosťami o rozdieloch medzi jednotkou MiB a MB a v čom je problém s fyzikálnymi predponami.

Naplnenie špecifikovaných cieľov a maturitného výkonového štandardu

Podľa cieľových požiadaviek na maturitu z informatiky [Špú08] má študent vedieť z uvedenej témy:

- (a) vysvetliť reprezentáciu rôznych číselných typov v počítači,
- (b) demonštrovať použitie rôznych číselných sústav.

Po absolvovaní uvedených hodín študenti dokázali vysvetliť reprezentáciu celočíselných typov v počítači a demonštrovať použitie špeciálnych číselných sústav (so základom 2, 8, 10, 16). Chýbajúce kompetencie nadobudli na seminároch č. 14 a 17.

Analyzujeme teraz špecifikované ciele a overme ich naplnenie:

- zdôvodniť používanie dvojkovej, osmičkovej a šestnástkovej sústavy v kontexte ich používania v počítačovom systéme, ukázať na príklade prevody medzi nimi:
 - študenti dokázali odpovedať na otázky počas aktivity (5),
- demonštrovať aritmetické operácie v dvojkovej sústave, zdôvodniť a na príkladoch dokumentovať vlastnosti aritmetiky počítača:
 - študenti dokázali vykonávať aritmetické operácie počas aktivít (2) a (3),
- demonštrovať princíp výpočtu veľkosti pamäte (v bitoch, bajtoch a vyšších jednotkách), času (aj pri prenose údajov) na jednoduchom príklade:
 - študenti dokázali odpovedať na otázky počas aktivity (2) týkajúce sa rozsahu celočíselných typov,
- vysvetliť reprezentáciu rôznych celočíselných typov v počítači, vypočítať veľkosť alokovanej pamäte pri použití štruktúr daných typov a vypočítať interval možných hodnôt pre celočíselné typy premennej:
 - študenti dokázali odpovedať na otázky počas aktivity (2) týkajúce sa rozsahu celočíselných typov.

Znakové reťazce a bezpečnosť

Ciele výučby

- vedieť pracovať s reťazcami v programovacom jazyku, ktorý sa študenti učia (napr. dĺžka reťazca, hľadanie pozície znaku, zmena hodnoty na zadanej pozícii, vloženie a vymazanie podreťazca, práca s poľom ako s reťazcom znakov),
- používať princíp kódovania znakov na prácu s abecedou,

- používať procedúry a funkcie bez parametrov a s parametrami,
- vedieť zdôvodniť potrebu ochrany údajov a vysvetliť, aké sú jej spôsoby a prostriedky,
- chápať potrebu právnej ochrany programov a vysvetliť dôsledky pri používaní nelegálneho softvéru.

Realizácia

So znakovými reťazcami študenti pracovali takmer od začiatku školského roku. Pracovali s objektom *Editbox*, ktorého vlastnosť *text* je typu znakový reťazec. Často používali funkciu *StrToInt*, ktorá vstupný text konvertuje na číslo, a s týmto číslom už mohli pracovať tak, ako potrebovali. Už mali určitú predstavu, čo znamená typ *string*, a s touto predstavou ďalej pracovali. Iba jeden študent doma sám zisťoval, čo znamená typ *string*.

Až doposiaľ nepotrebovali pracovať s jednotlivými znakmi reťazca. Ak bolo treba, znaky načítavali postupne. Na uvedených seminároch študenti pochopili koncept znakových reťazcov a na konci získali separovaný poznatok pre koncept jednorozmerného poľa.

Aktivity sme zvolili nasledovne:

- (1) Programovanie so znakovými reťazcami
- (2) Programovanie Cézarovej šifry
- (3) Programovanie detskej šifry

Programovanie zvolených úloh sme realizovali počas dvoch za sebou idúcich vyučovacích hodín (90 minút). Poslednú úlohu študenti programovali doma.

Nasledujúci seminár (90 minút) sme venovali pozornosť bezpečnosti komunikácie. Túto tému sme vybrali z viacerých dôvodov: nadväzovala na predchádzajúcu tému kódovanie znakov, čím prehĺbila získané poznatky; nadväzovala na programátorské úlohy zamerané na precvičenie práce s reťazcami; študenti ju potrebujú pre fundovanú prácu s technológiami.

Aktivity sme vybrali v nasledovnom poradí:

- (4) História šifrovania a šifrovanie v súčasnosti
- (5) Elektronický podpis a certifikačné authority

Na poslednom seminári (90 minút), ktorý bol venovaný znakovým reťazcom a bezpečnosti, študenti programovali také úlohy, aby v nich aplikovali teoretické poznatky o šifrovaní. Vďaka kryptoanalýze sme ich motivovali k programovaniu analýzy textu. Samotná úloha frekvenčnej analýzy textu mala viacero rôznych efektívnych riešení. Jedno z intuitívnych riešení je dokonca silnou motiváciou pre používanie údajovej štruktúry pole, čo nám môže pomôcť na nasledujúcich seminároch.

Z uvedených dôvodov sme posledných 90 minút zvolili nasledovne:

- (6) Programovanie šifry ATBAŠ
- (7) Programovanie frekvenčnej analýzy textu
- (8) Programovanie ďalšej analýzy textu

Popis aktivít a naplnenie čiastkových vzdelávacích cieľov

(1) Programovanie so znakovými reťazcami

Aktivít na zbieranie skúseností s programovaním s textovými reťazcami je mnoho. Je vhodné vybrať takú sadu úloh, ktorá poznatok buduje postupne. Syntaktické pravidlá môžeme vysvetliť pri tabuli alebo počítači, ale poznatok vznikne ľahšie, ak si tieto pravidlá študenti vyskúšajú a overia vo vlastnom programe.

Vo výskunnej vzorke už študenti pracovali so znakovými reťazcami, takže priradovanie, konvertovanie a vypisovanie už používajú bez vážnejších problémov. Nepracovali však so samotnými znakmi z reťazca. Takáto práca môže pomôcť pri získavaní skúseností pre prácu s poliami a zároveň vychádza zo skúseností nadobudnutých doposiaľ. Preto sa sústredíme na programátorské úlohy, ktoré pracujú so samotnými znakmi reťazca.

Úlohy sme vybrali nasledovne: Najprv si študenti vyskúšali syntaktické pravidlá – deklarovanie premennej typu *string*, vypísanie dĺžky reťazca, vypísanie znaku so zadaným indexom v reťazci, zreťazovanie reťazcov. V ďalšej úlohe študenti robili program, ktorý načíta a vypíše meno a priezvisko, skúšali ho zreťaziť a zistiť dĺžku celého mena. Aby sme ich motivovali, spýtali sme sa jednoduchú otázku: „*Kto má najdlhšie meno?*“ Aby používali

ako index premennú z cyklu, nasledovala úloha vypísať zadané slovo odzadu. Napísaný program potom využili na poslednú úlohu z tejto série, a to overenie, či je slovo palindróm.

Poslednú úlohu sme zvolili tak, aby mali študenti motiváciu vytvoriť si funkciu. To však spravil iba jeden študent, ktorý programuje aj vo svojom voľnom čase. Z pozorovaní usudzujeme, že by bolo vhodné v budúcnosti do metodiky prepracovať prácu s funkciami, pretože programy, ktoré žiaci zo vzorky píšú, sú slabo štruktúrované. Jedna z foriem, ako motivovať študentov k štruktúrovanosti kódu, je aj pomocou takejto úlohy. Je vhodné, aby učiteľ ukázal zvyšným žiakom program ich spolužiaka, ktorý je vďaka štruktúrovanosti jednoduchší a ľahšie čitateľnejší.

(2) Programovanie Cézarovej šifry

Keďže skoro každý sa v detstve hráva s rôznymi šiframi a študenti potrebujú získať aj kompetencie v oblasti bezpečnosti, vytvorili sme aktivitu, v ktorej študenti naprogramujú nejakú šifru. Cézarovú šifru sme vybrali z dôvodu precvičenia získaných skúseností z práce s kódovaním a abecedou, jednoduchosti a historickej dôležitosti.

Úlohou je napísať program, ktorý vypíše zadaný reťazec v zašifrovanom tvare. Najprv reťazec študenti šifrujú iba posunutím každého písmena v abecede o jedno miesto, neskôr však aj o toľko miest, koľko zadá používateľ. Napríklad:

$$\text{posun} = 3 \Rightarrow \text{ZEBRA} \rightarrow \text{CHEUD}$$

Pre zlepšenie štruktúrovanosti (budúcich) študentských programov sa osvedčilo priamo v zadaní napísať, aby študenti vytvorili funkciu *posun*, ktorá vráti na výstup vstupný znak posunutý v abecede, ako aj motiváciu pre použitie funkcie, ktorú si sami overili.

(3) Programovanie detskej šifry

Keďže programovanie je schopnosť, ktorú je potrebné budovať, rozvíjať a precvičovať, študenti dostali niekoľkokrát povinnú domácu úlohu. Povinné domáce úlohy sme vybrali vždy takej obtiažnosti, aby ich študenti vedeli vyriešiť využitím vedomostí v odučenom rozsahu. Nepovinné domáce úlohy už túto vlastnosť vždy nemali.

Úlohu naprogramovať detskú šifru riešili študenti doma. Program mal pre zadaný reťazec písmen vypísať jeho zašifrovanú podobu, a to tak, že za každú samohlásku pridal písmeno p a samohlásku zopakoval. Napríklad:

Toto je moja šifra → Topotopo jepe mopojava sipifrapa

Očakávali sme, že študenti pri tejto úlohe preukážu schopnosť pracovať s jednotlivými znakmi reťazca a zreťazovať premenné s konštantným reťazcom. Vzhľadom na to, že väčšina riešení využívala funkcie bez toho, aby sme to napísali do zadania a riešenia boli rozdielne, môžeme usúdiť, že ukážka použitia funkcií v minulých aktivitách mala za následok, že študenti aspoň niekoľko týždňov písali štruktúrovaný kód.

Táto úloha bola zároveň aj motivačná vďaka tomu, že vo výskumnej vzorke šifru poznal každý študent a pripomenulo mu to detskú hru. Myslíme si však, že ju nepozná každý študent. Očakávaný efekt motivácie hra splní pravdepodobne iba vtedy, ak ju bude poznať aspoň jeden študent z vyučovanej skupiny a prizná sa zvyšným spolužiakom.

(4) História šifrovania a šifrovanie v súčasnosti

Keďže sme plánovali motivovať študentov k vnímaniu bezpečnosti pri práci s technológiami a správaniu sa v súlade s pravidlami, bolo treba, aby získali informácie o princípoch fungovania moderných šifier. Samotné moderné šifry sú zložité na pochopenie, študenta je ľahšie motivovať príkladmi, ktoré pozná z minulosti alebo ich riešenia sám vymyslí. Preto sme seminár nezačali zložitými šiframi.

Túto aktivitu je vhodné viesť ako debatu, prípadne rozprávanie učiteľa s veľmi častým zapájaním študentov. Jednoduché šifry sa dajú zaradiť v ich historickom usporiadaní, čo pomôže študentom vidieť myslenie objaviteľov týchto šifier a môže im neskôr pomôcť v ich objavovaní. Nie je dôležité podrobne vysvetliť každú šifru, ale je vhodné nevynechať zaujímavosti, ktoré môžu študentov pritiahnúť k ďalšiemu štúdiu.

Pokiaľ sa študenti doteraz venovali kódovaniu informácií, odporúčame začať otázkami: „Čo je to šifrovanie?“, „Aký je rozdiel medzi šifrovaním a kódovaním?“, „Aké šifry po-

znáte?“ Ak na otázky budú vedieť odpovedať niektorí študenti namiesto učiteľa, nielen že ich aktivujeme, ale tiež ostatní študenti budú motivovaní sa naučiť o téme viac, pretože vidia, že kamarátov téma zaujíma a možno sa budú sami seba pýtať „*Keď to vie on, nemal by som to už vedieť aj ja?*“

Pri vysvetľovaní šifrier sme vychádzali z informácií z učebnice vzdelávania DVUI Kapitoly z informatiky 3 [WGF10]. Študenti pochopili princíp fungovania šifrier ATBAŠ, mechanickej šifry využívajúcej drevený kolík, Cézarovej šifry, všeobecnej substitučnej šifry s ukážkou šifry z poviedky o Sherlockovi Holmesovi a Vigenèrovej šifry. Študenti sa naučili nový pojem *klúč* a vymýšľali spôsoby, ako by šifrovaný text dešifrovali, čo je v jednotlivých šifrách kľúčom. Spoznali pojem frekvenčná analýza, ktorý sme využili nielen pri neskorších úlohách z programovania pri práci s textovými reťazcami, ale aj pri práci s jednorozmernými poliami. Spoznali príklady využitia steganografie a jej nelegálnosti v niektorých krajinách a dozvedeli sa informácie o dôvodoch šifrovania a používaných šifrách v druhej svetovej vojne.

Princíp symmetrickej a asymetrickej šifry pochopili študenti na príklade Vernamovej šifry, posielania správy v truhlici s dvoma kľúčmi a šifry RSA. Uvedomme si, že nie každý študent informatiky je vynikajúci v matematike. Podrobné vysvetlenie šifry RSA by mohlo niekoho odradiť, museli sme sa zamerať iba na samotný princíp asymetrickej šifry. Matematicky nadaných študentov je aj napriek tomu vhodné nejak nadchnúť. Preto sme iba stručne opísali šifru RSA, že je založená na matematických operáciách, pokiaľ sa chcú dozvedieť viac, kde informácie nájdu, a že bezpečnosť týchto šifrier závisí na probléme faktorizácie čísel. Ak by sa podarilo nájsť dostatočne rýchly algoritmus na faktorizáciu čísel, celé šifrovanie (napríklad práca s bankou) by bolo porazené a musel by sa nájsť iný spôsob komunikácie. Tento problém je dokonca taký dôležitý, že v minulosti RSA Laboratories poskytovalo odmenu za faktorizáciu jediného čísla. Problém s faktorizovaním sme ihneď využili, lebo študenti mali motiváciu ho riešiť (veď za to bola pred pár rokmi vypísaná veľká finančná odmena). Spoločne popísali niekoľko algoritmov na faktorizáciu čísla a zistili, ktoré algoritmy sú rýchlejšie a ktoré pomalšie, vďaka čomu sa prvýkrát stretli s pojmom *časová zložitosť programu* a s potrebou optimalizácie.

(5) Elektronický podpis a certifikačné authority

S elektronickým podpisom sa stretáva stále viac ľudí. Často aj ľudia, od ktorých sa očakáva, že vedia, aký je princíp elektronického podpisu, tak majú mylnú predstavu, mýlia si ho s obyčajným podpísaním dokumentu elektronickým perom alebo inou technológiou. Stále viac inštitúcii vyžaduje zaručený elektronický podpis, a pritom ľudia nevedia, čo od nich očakávajú. Toto je jeden z dôvodov, prečo sme vybrali ako jednu z tém elektronický podpis a certifikačné authority.

Ďalším dôvodom bolo časté stretávanie sa s certifikátmi pri komunikácii cez internet. Niektorí študenti z výskumnej vzorky sa už viackrát stretli s varovaním internetového prehliadača, že spojenie nie je dôveryhodné a nevedeli, prečo klikajú na tlačidlá, aby sa im napriek tomu načítala požadovaná webová stránka. Tento stav nie je dobrý, pretože to učí ľudí nečítať varovné a chybové správy. Študenti tiež používajú šifrovanú komunikáciu (*HTTPS*) aj pri práci s webovými stránkami vyžadujúcimi prihlasovanie sa a po prihlásení sa pracujú s osobnými údajmi.

Pri tejto aktivite je vhodné pripomenúť všetky tieto situácie a aj to, ako sa v nich správať, aby študenti vedeli, čo znamená správať sa bezpečne.

Elektronický podpis priamo nadväzuje na asymetrické šifrovanie, má veľmi podobný, len opačný princíp. Učiteľ by sa mal sústrediť na to, aby študent neskôr vedel, na čo si dať pozor pri používaní tejto technológie, v ktorej situácii je potrebné vlastniť zaručený elektronický podpis a v ktorej postačí nezaručený. Je vhodné tiež vysvetliť jeho potrebu, napríklad vďaka tomu, aké je ľahké poslať správu z inej e-mailovej adresy. Pri téme certifikátov je znova potrebné, aby učiteľ zdôraznil pravidlá bezpečnosti pre používateľa.

Uvedomme si, že na týchto hodinách sme síce použili debaty na aktivizovanie študentov, avšak najlepšie sa človek niečo naučí, ak si to sám vyskúša. Ak by sme mali viac času, mohli by si študenti vytvoriť elektronický podpis na hodine, to však z časového hľadiska nejde. Z toho dôvodu sme sa rozhodli, že si na hodine aspoň pozrú na počítači učiteľa, ako vyzerá elektronický podpis v praxi pri posielaní e-mailu a z čoho sa potom e-mail skladá. Na stránke predmetu sme potom napísali návod, ako si elektronický podpis nainštalovať na

svojom počítači. Bolo tiež potrebné zdôrazniť, že ide o legálny softvér a dôvody, prečo legálny softvér používame.

(6) Programovanie šifry ATBAŠ

Keďže ďalšiu aktivitu sme využili až po týždennej pauze, bolo treba pripomenúť prácu z minulej hodiny. Na rozcvičku sme vybrali programovanie šifry ATBAŠ, v ktorej bolo treba pracovať s abecedou a jednotlivými znakmi reťazca.

V tejto aktivite si treba dať pozor na to, že nie každý študent vymyslí, ako vypočítať znak z opačnej strany abecedy. Ak sa nájde taký študent, ktorý to sám nevymyslí ani po určitom čase, je vhodné mu dať pomocné otázky, napríklad ak to nevie na počítači, tak si na papieri vyskúšať konkrétne príklady a tie potom zovšeobecniť.

(7) Programovanie frekvenčnej analýzy textu

Keďže programovanie s textovými reťazcami sme využívali na prípravu na prácu s poliami a študentov sme motivovali k dešifrovaniu jednoduchých šifier, ako ďalšiu aktivitu sme vybrali programovanie frekvenčnej analýzy textu.

To znamená, že pre zadaný text mali študenti zistiť, koľkokrát sa v ňom nachádza ktoré písmeno.

Táto úloha mala veľa možných riešení. Problémom tiež bolo, že sme študentom dali možnosť si vybrať, či budú pracovať s grafickým prostredím, s textovými súbormi alebo konzolovou aplikáciou. Vyzvali sme ich k tomu, nech si rozmyslia, ktoré riešenie si vyžiada najmenej práce.

Keďže na tomto seminári riešil úlohy každý študent samostatne, na konci hodiny sme mohli diskutovať o rôznych riešeniach. Vďaka tomu mohli študenti vidieť rôzne riešenia a analyzovať ich výhody a nevýhody, a tým dosiahnuť vyššie kognitívne ciele podľa Bloomovej taxonómie.

Analyzujeme jednotlivé riešenia. V prvom rade je potrebné si uvedomiť, že sú dva základné typy algoritmov riešiacich túto úlohu. Program môže vykonať 26-krát cyklus a v ňom pre dané písmeno vypočítať počet jeho výskytov alebo jeden cyklus, v ktorom bude 26 podmienok. Treba poznamenať, že existuje ešte možnosť použiť len jeden cyklus a potom

použiť pole, ktoré študenti ešte nepoznajú, alebo vymyslieť zložité riešenie, ktoré použije reťazec ako pole.

V prípade, že študent použije jeden cyklus a v ňom 26 podmienok, vstup stačí načítať jedným prechodom, nie je problém s rôznymi technológiami na načítavanie vstupu. Pokiaľ však bude opakovať cyklus 26-krát, vstup bude musieť dokázať načítať do jednej premennej typu reťazec alebo ho opakovane načítavať. Pri práci s grafickým prostredím študenti používali na načítavanie vstupu iba *Editbox*. V tejto úlohe si môžu všimnúť, že do *Editboxu* môže používateľ zadať iba kratší, jednoriadkový text, takže zadanie úlohy by muselo byť obmedzené na kratšie texty. Poznamenajme, že vo výskumnej vzorke sa nikto nepokúsil nájsť vhodnejší objekt, napríklad *Memo*.

Ďalšia možnosť pre načítavanie vstupu boli textové súbory. Túto možnosť si nevybral žiaden študent. Jedno z riešení by mohlo neustále otvárať a zatvárať súbor. Tu je možno vhodné sa študentov spýtať, či považujú takéto riešenie za efektívne. Ďalšia možnosť je načítať do jednej premennej typu reťazec celý súbor. Ak by sa takéto riešenie objavilo, učiteľ by mal zvážiť schopnosti svojich študentov, či im neponúknuť informácie aj o pamäťovej zložitosti.

Načítanie vstupu z konzoly nastoľuje problém nemožnosti opätovného načítavania vstupu. V tomto prípade môže študent použiť vyššie spomenuté riešenie – načítať vstup do jednej premennej typu reťazec.

Vo výskumnej vzorke sme sa stretli s riešeniami, ktoré načítavali vstup 26-krát z *Editboxu*, jedno načítanie z konzolovej aplikácie, kde v cykle bolo 26 podmienok, ale aj neúspešné pokusy o načítanie z konzolovej aplikácie, pretože si študent neuvedomil, že do premennej typu reťazec môže priradiť aj text obsahujúci medzery a iné znaky.

Rôzne riešenia si študenti navzájom ukazovali a diskutovali o ich výhodách a nevýhodách. Dôležité bolo, že medzi riešeniami sa vyskytlo aj také, ktoré prešlo zadaný vstup iba raz a v cykle študent ručne napísal 26 podmienok. Toto riešenie pomohlo motivovať študentov k ďalšej téme, a to jednorozmerné pole.

(8) *Programovanie ďalšej analýzy textu*

Poslednú aktivitu už nestihli spracovať všetci študenti na seminári, niektorí ju dokončili doma. Úlohou bolo pre daný text zistiť, koľko obsahuje znakov a koľko slov. K riešeniu úlohy sme ich motivovali vďaka tomu, že takúto funkciu potrebovali použiť v minulosti, keď písali eseje a ich dĺžka bola ohraničená.

Učiteľ by si mal aj pri tejto úlohe uvedomiť úskalía. Problémom môže byť nesprávny počet medzier, či nevhodné formátovanie.

Naplnenie špecifikovaných cieľov a maturitného výkonového štandardu

Analyzujeme špecifikované ciele a overme ich naplnenie:

- vedieť pracovať s reťazcami v programovacom jazyku, ktorý sa študenti učia (napr. dĺžka reťazca, hľadanie pozície znaku, zmena hodnoty na zadanej pozícii, vloženie a vymazanie podreťazca, práca s poľom ako s reťazcom znakov):
 - študenti naprogramovali program, ktorý vypočítal dĺžku reťazca (meno a priezvisko), našiel pozície znaku (hľadanie medzery v aktivite (8)), zmenil hodnotu na zadanej pozícii (Cézarová šifra), vložil a vymazal podreťazec (detská šifra), pracoval s poľom ako s reťazcom znakov (Cézarová šifra),
- používať princíp kódovania znakov na prácu s abecedou:
 - študenti naprogramovali Cézarovú šifru a šifru ATBAŠ,
- používať procedúry a funkcie bez parametrov aj s parametrami:
 - študenti použili funkciu *posun* pri Cézarovej šifre, používali funkcie pri detskej šifre,
- vedieť zdôvodniť potrebu ochrany údajov a vysvetliť, aké sú jej spôsoby a prostriedky:
 - študenti dokázali odpovedať na otázky počas aktivity (5),
- chápať potrebu právnej ochrany programov a vysvetliť dôsledky pri používaní nelegálneho softvéru:
 - vysvetlili sme pri aktivite (5), kontrola prebehla iba u niektorých študentov pomocou pozorovania ich postojov.

Pomocou dotazníka, ale aj z rozhovorov so študentmi sme zistili, že jednej študentke sa nepáčila téma šifry. Po opýtaní sa na dôvod sme sa dozvedeli, že mala problém s programovaním úlohy, kde bolo treba vypísať reťazec odzadu. To bol vraj jediný dôvod, ale ten ju naladil negatívne na zvyšok témy. Preto je vhodné v budúcnosti túto úlohu zameniť, alebo pridať viac úloh na zbieranie skúseností. Tento jav sa u ostatných študentov nevyskytol.

7.4.3 Metodické postrehy a skúsenosti s ich implementáciou

Počas pedagogickej praxe sme skúsili použiť mnoho metodických prostriedkov na zvýšenie kvality vyučovania. Niektoré sme vybrali z odbornej literatúry, aplikovali a zhodnotili, iné vznikli pozorovaním učiteľov alebo vlastnej pedagogickej praxe.

Organizácia výučby programovania v učebni

Analyzujeme rôzne využívané spôsoby organizácie výučby programovania v triede:

(1) *Samostatné riešenie úloh*

Každý žiak dostane sadu úloh, ktoré má vyriešiť. Pokiaľ je potrebné vysvetliť nové pojmy, buď sú vysvetlené v sade úloh, alebo ich učiteľ vysvetlí na vyučovacej hodine.

Výhody:

- Každý študent môže riešiť úlohy svojim tempom.
- Pokiaľ študent nestihne vyriešiť všetky úlohy, môže pokračovať doma. Ak sa nezúčastnil vyučovania, úlohy môže tiež riešiť doma.
- Študent vyrieši *každú úlohu* sám, prípadne s pomocou učiteľa, takže získava vlastné skúsenosti.
- Študent sa spolieha na seba pri riešení problémov. Učí sa hľadať si chyby, používať pomocníka alebo internet.
- Študent nemá problém s odvádzaním pozornosti od riešenia problémov, pretože ho väčšinou nevyrušujú spolužiaci alebo učiteľ.

Nevýhody:

- Učiteľ musí stíhať pomáhať viacerým študentom. Musí sa vedieť rýchlo orientovať v novom kóde.
- Študent programu iba píše a nepopisuje algoritmus nahlas.
- Študent je konfrontovaný iba s vlastnými chybami, s vlastným uvažovaním, s vlastným kódom.

Ak má učiteľ stíhať pomáhať každému študentovi, takáto metóda je možná iba v malej skupine študentov.

(2) Ukazovanie riešenia pri projektore

Učiteľ ukazuje riešenie pri projektore. Riešenie vzniká v spolupráci so žiakmi.

Výhody:

- Študent vidí program napísaný vždy čitateľne, takže vďaka prirodzenej tendencii napodobňovať bude kód písať tiež čitateľne.
- Programy vytvárajú študenti tímovo, čo ich učí spolupracovať.
- Študent je konfrontovaný aj s myšlienkovým procesom iného človeka, čo ho učí novým myšlienkovým postupom.
- Každú úlohu niekto vyrieši, takže študent nie je demotivovaný tým, že problém nevyrieši.
- Počas vyučovacej hodiny sa stihne preriešiť veľa rôznych úloh.

Nevýhody:

- Každý študent musí riešiť úlohy rovnakým tempom.
- Pokiaľ učiteľ nezadá domácu úlohu alebo cvičné úlohy, študent nevie, aké úlohy má riešiť doma.
- Študent sa učí hlavne program čítať, nie ho tvoriť. To je vyššia úroveň Bloomovej taxonómie.
- Riešenia vytvoria šikovnejší a dominantnejší študenti, takže niektorí študenti nemusia nič robiť.
- Študent väčšinou nie je konfrontovaný s chybným programom, napriek tomu, že chybami sa človek učí. „*Polya pridáva: „Môžeme sa učiť chybami ...“ [Pol45] (str. 213).*

Ukazujú učebnice chybné pokusy? Ťažko, ak vôbec. A potom ľudia znova a znova idú zlou cestou a vytvárajú chybné riešenia.“ [Gin08]

(3) Študenti riešia úlohy postupne pri projektore

Študenti majú dopredu známy zoznam úloh. Postupne ich učiteľ volá k projektoru, kde vždy vyriešia jednu úlohu. Pokiaľ nastane nejaký problém, môže im pomôcť iný študent alebo učiteľ pomocnými otázkami.

Výhody:

- Študent musí písať kód do už existujúceho programu. Musí sa vedieť orientovať v kóde iného človeka, čím získava dôležité pracovné návyky. Experiment ukázal, že študenti neskôr vlastný kód písali čitateľne (formátovanie, komentáre), pretože ich hnevalo, ak tak nepísal niektorý kolega.
- Študenti sa stretávajú s viacerými chybami, nielen s tými, ktoré by sami spravili. „*Postupne nám zapli aj také veci, čo by nám nezapli.*“ (študent 5-4)
- Každý študent vyrieši nejakú úlohu sám. Nie je demotivovaný rýchlymi kolegami.

Nevýhody:

- Niektorý študent sa môže báť riešiť úlohu pred ostatnými.
- Študenti si musia zvyknúť na takýto spôsob výučby, napríklad pochopením jeho výhod.
- Nie každý študent zvláda riešiť vlastné príklady a zároveň sledovať, čo sa deje pri projektore.
- Môže sa stať, že niektorý študent naprogramuje riešenie svojej úlohy, všetky ďalšie opíše.

Pri takejto organizácii vyučovania je dôležité aj priradenie úloh k ľuďom. Slabšieho študenta by mohla demotivovať príliš zložitá úloha. Preto je vhodné, aby si učiteľ pred hodinou zvolil, v akom poradí bude študentov vyvolávať k projektoru.

Treba si dať tiež pozor, aby učiteľ nenaprával neustále riešenia študentov. Je vhodné občas napraviť formátovanie. Ale pokiaľ študent vymyslel nejaký algoritmus, ktorý vedie k

riešení, je treba ho považovať za správny. Aj keď učiteľ možno vymyslel efektívnejšie zapísané alebo všeobecnejšie riešenie. Nechceme predsa zničiť u študenta pocit z dobre vykonanej práce. „*Ak majú študenti pocit, že existuje iba jedna správna odpoveď na problém, môžu sa cítiť obmedzení pri vlastnom experimentovaní pred nájdením správneho prístupu.*“ [FP09]

Vo výskumnom projekte sme kombinovali viaceré spôsoby organizácie vyučovania. Pri pozorovaní (nultá iterácia) učiteľ používal hlavne spôsob (2), avšak namiesto projektora používal tabuľu. Pokiaľ boli niektorí študenti rýchli, program si vylepšovali.

V prvej iterácii sme používali prevažne spôsob (1). Najprv sme úlohy zadali na začiatku hodiny, ale študenti, ktorí neboli na seminári, nevedeli, čo sa prebralo na hodine. Preto sme im vytvorili webovú stránku, kde mali napísané všetky zadania úloh. Na webových zadaniach sme občas napísali pomôcku farbou pozadia, vďaka čomu začali študenti pozeráť do zdrojového kódu stránky. To sa ukázalo ako výhoda, pretože sa naučili, ako vyzerá zdrojový kód webovej stránky.

V druhej iterácii sme používali najprv spôsob (1), ktorý mal mnoho výhod, aj s použitím webovej stránky na zadávanie úloh a s písaním pomôcok farbou pozadia (čo malo rovnaký efekt). Neskôr sme skúsili použiť spôsob (3) a spýtali sa ich, kedy mali pocit, že sa naučili viac. Vybrali si spôsob (3). Zistili sme, že študenti začali písať kód čitateľne nielen na hodinách, ale aj pri riešení domácich úloh. Keďže tento spôsob má ale aj spomenuté nevýhody, na zvyšných seminároch sme striedali spôsoby (1) a (3).

Rozdielne potreby študentov

V projekte dizertačnej práce sme sa pokúsili nájsť spôsoby, ako zohľadniť rozdielne potreby a skúsenosti študentov.

V nulte iterácii sme mohli pozorovať učiteľa, ktorý viedol všetkých študentov rovnakým tempom. Požiadal ich, aby nepoužívali veci, ktoré sa ešte neučili. „*Meňte iba to, čo vieme meniť!*“ Tento prístup mal výhody, ale skôr pre učiteľa ako pre žiaka, lebo tým znížil motiváciu hľadať nové informácie a narábať s nimi. Taktiež to odporuje poznatkom modernej pedagogiky, napríklad ak si všimneme dvojrozmernú adaptáciu Bloomovej taxo-

nómie [FJA⁺07], vidíme, že každý študent má svoju vlastnú cestu, ktorou sa mu vytvára poznanie.

Na to, aby sme zohľadnili rozdielne potreby a skúsenosti, sme museli prispôbiť aj vyššie popísanú organizáciu vyučovania v učebni. Organizácie (1) a (3) nám dovolili pracovať so žiakmi na ich úrovni, organizácia (2) to nedovoľovala.

Ďalším spôsobom na zohľadnenie rozličných potrieb študentov bolo zadanie úloh pre pokročilých, úloh na precvičenie a alternatívnych úloh (napr. v úlohe o Garfieldovi uvedenom na priloženom médiu).

Snažili sme sa študentom umožniť výber prostredia. Keď sa naučili pracovať s grafickým prostredím, konzolovou aplikáciou a textovými súborami, pre každú úlohu si potom vyberali, akým spôsobom chcú pracovať s používateľskými vstupmi. Vďaka tomu nielen, že mali študenti pocit seberealizácie, ale aj im to pomohlo analyzovať nové aspekty pri riešení problémov.

V neposlednom rade spomenieme aj prácu s nadanými študentmi. V prvej iterácii sme vyskúšali jednoduchý spôsob. Nadanému študentovi sme dali do rúk zadania korešpondenčného seminára [KSP] a olympiády z informatiky [OI], avšak najdôležitejšie sa javilo dať mu možnosť učiť sa. Ukázali sme mu internetový portál Liaheň [lia]. Taktiež sme s ním komunikovali mimo vyučovania o problémoch, ktoré ho zaujímali (neurónové siete). Výsledkom bolo, že úlohy z portálu Liaheň riešil aj cez letné prázdniny a ďalší rok sa zapojil do Olympiády z informatiky. V druhej iterácii sme skúsili rovnaký postup, avšak neúspešne. Dvomi nadaným študentom sme ukázali portál Liaheň. Avšak do riešenia úloh na portáli sa vo voľnom čase nezapojili. Jeden z možných dôvodov mohol byť aj ten, že sme tak spravili príliš skoro, keď ešte nemali dostatočne silný aparát na vyriešenie najľahšej úlohy. Na toto si treba dať určite pozor aj pri ďalších pokusoch. Nadanie sme však rozvíjali aspoň tak, že sme im dávali podnety v úlohách pre pokročilých, kde občas museli nájsť sami nové informácie.

Výber interakcie s používateľom

Počas pedagogickej praxe sme využili viac prístupov pri výbere interakcie s používateľom. Grafické rozhranie veľmi ľahko motivuje študentov k programovaniu, ale práca s konzolou je pre niektoré úlohy rýchlejšia a efektívnejšia.

V nulte iterácii sme pozorovali používanie grafického rozhrania, ale výber úloh zväčša nezodpovedal výberu interakcie s používateľom. Študenti často používali grafiku iba preto, že museli. Samozrejme, niektoré úlohy učiteľ zvolil graficky atraktívne, ale nebolo ich veľa.

V rámci motivácie študentov sme v prvej iterácii dlho pracovali iba s grafickým prostredím. Nastala však situácia, keď to bolo zbytočné. Skúsili sme ukázať študentom konzolovú aplikáciu. Nielen preto, že aktuálne úlohy práve nepotrebovali prácu s grafikou, ale aj preto, že ďalšiu hodinu sme chceli venovať textovým súborom, v ktorých na načítavanie vstupu a vypisovanie výstupu sa používa rovnaký príkaz ako pri práci s konzolovou aplikáciou. Po týchto seminároch sme vyzvali študentov, nech si vyberajú podľa úlohy, čo chcú použiť. Výsledok bol, že pri vhodných úlohách iba jeden študent používal naďalej grafiku, väčšina si vybrala konzolovú aplikáciu a sami komentovali, že sa už konečne nemusia hrať s úpravou prostredia, ale môžu programovať to, čo potrebujú.

V druhej iterácii sme tiež dlho pracovali s grafickým prostredím v rámci motivácie, neskôr ukázali študentom konzolovú aplikáciu, a potom ich vyzvali k vlastnému výberu interakcie s používateľom. V tejto výskumnej vzorke si tiež radi vybrali konzolovú aplikáciu, pokiaľ to bolo vhodné pre danú úlohu. Úspech potvrdzujú tiež odpovede z dotazníka, napríklad:

Otázka: „Ktorá hodina sa Ti najviac páčila?“

Odpoveď: „Práca s konzolovou aplikáciou, a hlavne tá domáca úloha, čo bola k tomu (dala zabrať, ale bola to sranda).“

Výber úloh

Úlohy by mali zodpovedať hlavne didaktickému hľadisku. Mali by sme rešpektovať poznávací proces a podľa toho úlohy vyberať.

Dôležitá je však aj motivácia študentov riešiť dané úlohy. Málokto bude s radosťou programovať úlohy iba preto, že vďaka tomu dokáže o rok naprogramovať zložitejšiu aplikáciu.

V dnešnej dobe sa používajú technológie na rôznych miestach. Často sa stáva, že ľudia pretláčajú technológie aj tam, kde nie sú potrebné. Hovorí sa, že počítač je dobrý sluha, ale zlý pán. Preto je vhodné študentov naučiť, že počítač sa používa na uľahčenie práce. Podľa Astrachanovho pravidla [Par03] si treba dať pozor, aby sme nezadávali úlohy, ktoré sú vytvorené umelo a dajú sa riešiť ľahšie a rýchlejšie bez počítača (v prípade programátorských úloh aj bez programovania). Nielen, že v tom nebudú vidieť študenti zmysel, ale ich aj naučíme používať počítače tam, kde ich netreba.

Učiteľ by pri výbere úloh nemal preskočiť niekoľko úrovní podľa Bloomovej taxonómie. „*Niektorí učitelia môžu toto interpretovať ako pokyn – vysvetliť syntax cyklu a ukázať pár príkladov na tabuľu. Potom zadajú študentom naprogramovať ich vlastné cykly a sú prekvapení, keď každý semester je mnoho študentov stratených.*“ [SMS08]

V prvej iterácii projektu dizertačnej práce sme spravili vyššie spomenutú chybu. Po dvoch seminároch venovaných cyklu s pevným počtom opakovaní sme zaradili tému podmienky v programovaní. Študenti boli v nových úlohách stratení, nevedeli, čo majú robiť. Nielen, že sme preskočili niekoľko úrovní Bloomovej taxonómie, ale zmenili sme aj spôsob práce. Kým pôvodne programovali pomocou cyklov rôzne obrázky, na tomto seminári programovali hlavne matematické úlohy. V druhej iterácii sme uvedené chyby napravili. Porozumenie konceptu cyklu s pevným počtom opakovaní sme budovali postupne a až po dôkladnom upevnení poznania sme vybrali úlohy, ktoré boli odlišné tým, že už narábali nielen s kreslením útvarov rôznych farieb na vypočítané pozície, ale vykresľovali sme vypočítané čísla – úlohy na súčty v cykle sme zabalili do atraktívnejšej podoby, a to vykresľovanie do trojuholníkov a iných útvarov.

Proces tvorby algoritmu

Podľa Bloomovej taxonómie je syntéza na oveľa vyššej úrovni ako porozumenie. Znamená to, že ak chceme naučiť študentov tvoriť programy, nestačí im veľkú časť algoritmu prezradiť, aj keď oni nami vytvorenému algoritmu porozumejú, prípadne ho aplikujú.

Tvoriť algoritmus treba študentov naučiť. Jeden zo spôsobov popísal Becker na svojej webovej stránke [Bec99]. Učiteľ by mal s pomocou študentov kresliť na tabuľu rozhodovací strom, aby zachytil aj neúspešné cesty pri tvorbe algoritmu.

Je vhodné, aby učiteľ študentom dal nejaké rady, ako tvoriť algoritmus. Podľa vzorov by sa mali študenti rozhodovať, aké koncepty je treba v programe použiť. Dobré je, aby si študenti kládli rôzne otázky: „*Bude sa musieť niečo opakovať?*“, „*Viem dopredu povedať, koľkokrát sa to bude opakovať?*“, „*Bude treba sa niekedy rozhodovať?*“. Ak odpovede učiteľ zaznačí na tabuľu rôzne prípady, ktoré so študentmi našli, študent sa naučí, že keď nevie, čo má spraviť, môže si rovnako jednotlivé prípady a odpovede na otázky zapísať na papier, aby sa k nim mohol ľahko vrátiť pri písaní programu. Je dôležité poznamenať, že tu už hovoríme o zložitejších programoch, kde už nestačí skúšať písať program.

Zložité programy nie je ľahké písať ihneď. Treba si ich rozmeniť na ľahšie. To však študent nedokáže, pokiaľ to niekde neuvidí. Rovnako nestačí, ak bude učiteľ zadávať vždy postupnosť zložitejších úloh, pretože študent nevyskúša postup, ako úlohu zľahčiť.

V prvej iterácii sme úlohy zadávali ako postupnosť, ktorej zložitosť narastala. Dlhو sme neoverili, či študenti dokážu riešiť problémy aj z druhej strany, keď majú zložitú úlohu, rozložiť ju na jednoduchšie. Neskôr sme zistili, že pri príliš zložitých úlohách si už študenti nevedia rady.

V druhej iterácii sme spôsob zadávania úloh zmenili tak, že na prvých pár seminároch sme zadávali úlohy indukčne. Na ďalších hodinách už dostali zložité úlohy a vyzvali sme ich, aby si ich rozdelili na ľahšie. Toto sa javilo ako úspešné. Ak po nejakom čase dostali zložitú úlohu, už vedeli, ako s ňou narábať. Rozdelili si ju na ľahšie úlohy, ktoré vyriešili a zložitejšiu úlohu potom poskladali alebo nabalili z ľahších úloh.

Kapitola 8

Výsledky výskumu

Cieľom nášho výskumu bolo vytvoriť niekoľko dokumentov: časovo-tematický plán, cieľové požiadavky vybraných tém, vstupné požiadavky a výstupné štandardy a metodické pomôcky. Tieto dokumenty sme zahrnuli do predchádzajúcej kapitoly. K cieľom práce sme si položili výskumné otázky, na ktoré v tejto kapitole odpovieme.

Ako špecifikovať vstupné požiadavky a výstupné štandardy tak, aby študenti spĺňajúci tieto požiadavky dokázali úspešne absolvovať maturitnú skúšku a mali informatické kompetencie potrebné pre štúdium na vysokej škole.

Na špecifikáciu vstupných požiadaviek sme použili zistený skutočný stav poznania u študentov. Vstupné požiadavky a výstupné štandardy sme prezentovali v tabuľke 7.4. Pozorovaním sme tiež zistili, že už účastníci výskumu z prvej iterácie dokázali úspešne absolvovať maturitnú skúšku a ich odpovede spĺňali všetky požiadavky na maturitu z informatiky.

Aký časovo-tematický plán zvoliť, aby študenti spĺňali výstupné štandardy?

Na túto otázku odpovedáme v tabuľke 7.3.

Aké metodické postupy a prostriedky použiť, aby sme dosiahli špecifikované ciele a čiastkové ciele?

Z dôvodu rozsahu práce sme vytvorili metodiku vybraných tém: dvojková sústava a jednotky informácie a znakové reťazce a bezpečnosť. Tá je uvedená v časti 7.4.2. K obom metodikám sme špecifikovali vzdelávacie ciele a analyzovali ich naplnenie. Taktiež sme vytvorili metodické potrehy z hodín. Tie uvádzame v časti 7.4.3. Taktiež sme vytvorili portál určený pre našich študentov, kde môže čitateľ uvidieť celý súbor aktivít, ktoré sme použili v druhej iterácii. Ten nájdeme na adrese <http://people.ksp.sk/~julka/tilg/> a jeho kópia sa nachádza na priloženom médiu.

Kapitola 9

Záver

V projekte dizertačnej práce sme sa zaoberali tvorbou metodiky pre prvú časť prípravy na maturitnú skúšku z informatiky. Definovali sme zadanie výskumného projektu dizertačnej práce, navrhli metódy vedúce k výsledku, ktoré sme neskôr aplikovali.

Prezentovali sme prehľad problematiky vyučovania informatiky vo svete a na Slovensku, a to z pohľadu histórie aj súčasnosti.

Popísali sme priebeh výskumu a čiastkové aj celkové výsledky. Na základe analýzy získaných dát sme vytvorili vstupné požiadavky a výstupné štandardy vychádzajúce z reálnych skúseností, časovo-tematický plán, metodiky vybraných tém a metodické pomôcky pre učiteľov informatiky. Výstupom je tiež webový portál určený pre našich študentov, ktorý obsahuje celý súbor aktivít použitých vo vyučovaní.

Dúfame, že naše výsledky budú môcť využiť učitelia pri realizovaní vyučovacích hodín, tvorbe časovo-tematických plánov a inej pedagogickej dokumentácie. Výsledky by mohli byť tiež inšpiráciou pre pedagogické ústavy, metodické centrá a predmetové komisie.

Vlastné publikácie

- [AFŠW10] Gabriela Andrejková, Michal Forišek, Juliana Šišková, a Michal Winczer. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika – Kapitoly z informatiky*. 2010. ISBN 978-80-8118-052-1.
- [BBC⁺09a] Katarína Bachratá, Hynek Bachratý, Peter Czimmermann, Juliana Šišková, a Michal Winczer. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika – Matematika pre učiteľov informatiky 2*. 2009. ISBN 978-80-89225-97-2.
- [BBC⁺09b] Katarína Bachratá, Hynek Bachratý, Oľga Czimmermannová, Peter Czimmermann, Stanislav Krajčí, Peter Novotný, Juliana Šišková, a Michal Winczer. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika – Matematika pre učiteľov informatiky 1*. 2009. ISBN 978-80-89225-50-7.
- [FŠ10] Michal Forišek a Juliana Šišková. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika – Kapitoly z informatiky 1*. 2010. ISBN 978-80-8118-071-2.
- [Lip09] Juliana Lipková. Informatické súťaže na Slovensku. In Branislav Rován, editor, *Zborník konferencie DIDINFO*, str. 129–131. Univerzita Mateja Bela, Banská Bystrica, 2009. ISBN 978-80-8083-720-4.

- [Šiš11] Juliana Šišková. Výučba informatiky vo vyšších ročníkoch stredných škôl v zahraničí. In *Zborník konferencie DIDINFO*, str. 201–205. Univerzita Mateja Bela, Banská Bystrica, 2011. ISBN 978-80-5570-142-4.
- [SŠ10] Monika Steinová a Juliana Šišková. Programming camps: Letting children discover the Computer Science. In Juraj Hromkovič, Richard Kráľovič, a Jan Vahrenhold, editors, *ISSEP*, volume 5941 of *Lecture Notes in Computer Science*, str. 170–181. Springer, 2010. ISBN 978-3-642-11375-8.

Literatúra

- [ABGZ10] Michal Armoni, Tamar Benaya, David Ginat, a Ela Zur. Didactics of Introduction to Computer Science in High School. In Hromkovic et al. [HKV10], str. 36–48. ISBN 978-3-642-11375-8.
- [AQA09a] AQA. GCE, AS and A Level specification, Computing, AS exams 2009 onwards, A2 exams 2010 onwards. <http://web.aqa.org.uk/>, 2009.
- [AQA09b] AQA. GCE, AS and A Level specification, Information and Communication Technology, AS exams 2009 onwards, A2 exams 2010 onwards. <http://web.aqa.org.uk/>, 2009.
- [AQA09c] AQA. General Certificate of Education, Applied Information and Communication Technology 8751/8753/8756/8757/8759 2010, Specification. <http://web.aqa.org.uk/>, 2009.
- [BD06] Jonas Blonskis a Valentina Dagiene. Evolution of Informatics Maturity Exams and Challenge for Learning Programming. In Mittermeir [Mit06], str. 220–229. ISBN 3-540-48218-0.
- [Bec99] B. W. Becker. Pedagogical pattern #56 expose the process. <http://www.cs.uwaterloo.ca/~bwbecker/Patterns/ExposeTheProcess.html>, 1999.
- [Ber09] Jonte Bernhard. Learning through artifacts in engineering education: Some perspectives from the philosophy of technology and engineering science. In *6th*

- European Forum on Continuing Engineering Education, Quality Development in Lifelong Learning - in Theory and Use*, 2009.
- [BK05] Andrej Blaho a Ivan Kalaš. *Tvorivá informatika: 1. zošit z programovania*. Slovenské pedagogické nakladateľstvo – Mladé letá, 2005.
- [BL08] Kevin Bond a Sylvia Langfield. *AQA Computing: AS Exclusively endorsed by AQA*. Nelson Thornes, 2008. ISBN 978-0-7487-8296-3.
- [Bla06] Andrej Blaho. *Informatika pre SŠ – Programovanie v Delphi*. Slovenské pedagogické nakladateľstvo, Bratislava, 2006.
- [BS07] Andrej Blaho a Ľubomír Salanci. *Tvorivá informatika – 1. zošit o práci s textom*. Slovenské pedagogické nakladateľstvo – Mladé letá, Bratislava, 2007.
- [BWF98] Tim Bell, Ian H. Witten, a Mike Fellows. *Computer science unplugged...* <http://csunplugged.org/binary-numbers>, 1998.
- [Chr07] Miroslav Chrásta. *Metódy pedagogického výskumu: Základy kvantitatívneho výskumu*. Grada publishing, Praha, 2007. ISBN 978-80-247-1369-4.
- [Cla10] Donald Clark. Bloom's taxonomy of learning domains. <http://www.nwlink.com/~donclark/hrd/bloom.html>, 2010.
- [Com05] ACM Task Force Curriculum Committee. *A Model Curriculum for K-12 Computer Science*. Association for Computing Machinery, 2. edition, 2005.
- [DDS06] Valentina Dagiene, Gintautas Dzemyda, a Mifodijus Sapagovas. Evolution of the Cultural-Based Paradigm for Informatics Education in Secondary Schools - Two Decades of Lithuanian Experience. In Mittermeir [Mit06], str. 1–12. ISBN 3-540-48218-0.
- [DRFG08] J. D. Dougherty, Susan H. Rodger, Sue Fitzgerald, a Mark Guzdial, editors. *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science*

- Education, SIGCSE 2008, Portland, OR, USA, March 12-15, 2008*. ACM, 2008. ISBN 978-1-59593-799-5.
- [FJA⁺07] Ursula Fuller, Colin G. Johnson, Tuukka Ahoniemi, Diana Cukierman, Isidoro Hernán-Losada, Jana Jacková, Essi Lahtinen, Tracy L. Lewis, Donna McGee Thompson, Charles Riedesel, a Errol Thompson. Developing a computer science-specific learning taxonomy. *SIGCSE Bulletin*, 39(4):152–170, 2007.
- [FP09] Katrina E. Falkner a Edward Palmer. Developing authentic problem solving skills in introductory computing classes. In Sue Fitzgerald, Mark Guzdial, Gary Lewandowski, a Steven A. Wolfman, editors, *SIGCSE*, str. 4–8. ACM, 2009. ISBN 978-1-60558-183-5.
- [FŠ82] René Filustek a Stanislav Šíma. *Počítačové systémy pre 4. ročník gymnázia*. Slovenské pedagogické nakladateľstvo, Bratislava, 1982.
- [Gav07] Peter Gavora. *Spríevodca metodológiou kvalitatívneho výskumu*. Univerzita Komenského, Bratislava, 2007. ISBN 978-80-223-2317-8.
- [GEH99] Judith Gal-Ezer a David Harel. Curriculum and Course Syllabi for a High-School CS Program. *Computer Science Education*, 9(2):114–147, 1999.
- [GF79] Jozef Gruska a Miloš Franek. *Algoritmy pre 3. ročník gymnázia*. Slovenské pedagogické nakladateľstvo, Bratislava, 1979.
- [GHOK⁺05] Ewa Gurbiel, Grazyna Hardt-Olejniczak, Ewa Kolczyk, Helena Krupicka, a Maciej M. Syslo. Informatics and ict in polish education system. In Roland Mittermeir, editor, *ISSEP*, volume 3422 of *Lecture Notes in Computer Science*, str. 46–52. Springer, 2005. ISBN 3-540-25336-X.
- [Gin08] David Ginat. Learning from wrong and creative algorithm design. In Dougherty et al. [DRFG08], str. 26–30. ISBN 978-1-59593-799-5.

- [GL09] Alan Gardner a Carl Lyon. *Oxford Revision Guides: AS & A Level ICT Through Diagrams*. Oxford University Press, 2009. ISBN 978-0-19-918093-6.
- [GSS09] Ján Guniš, Miloslava Sudolská, a Ľubomír Šnajder. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika – Úvod do vzdelávania. Informatika vo všeobecnom vzdelávaní*. 2009. ISBN 978-80-89225-53-8.
- [HAB⁺11] Peter Hubwieser, Michal Armoni, Torsten Brinda, Valentina Dagiene, Ira Diethelm, Michail Giannakos, Maria Knobelsdorf, Johannes Magenheim, Roland Mittermeir, a Sigrid Schubert. Computer Science/Informatics in Secondary Education, working group report. In Guido Röbling, Thomas L. Naps, a Christian Spannagel, editors, *ITiCSE*, str. 19–38. ACM, 2011. ISBN 978-1-4503-0697-3.
- [Hau08a] Július Hauser. Štátny vzdelávací program pre 2. stupeň základnej školy v slovenskej republike, isced 2 – nižšie sekundárne vzdelávanie. <http://www.statpedu.sk/sk/filemanager/download/42>, 2008.
- [Hau08b] Július Hauser. Štátny vzdelávací program pre gymnáziá v slovenskej republike, isced 3a – vyššie sekundárne vzdelávanie. <http://www.statpedu.sk/sk/filemanager/download/43>, 2008.
- [HG88] Jozef Hvorecký a Peter Gabčo. *Informatika a výpočtová technika – Algoritmy*. Slovenské pedagogické nakladateľstvo, Bratislava, 1988.
- [HKV10] Juraj Hromkovic, Richard Královic, a Jan Vahrenhold, editors. *Teaching Fundamentals Concepts of Informatics, 4th International Conference on Informatics in Secondary Schools - Evolution and Perspectives, ISSEP 2010, Zurich, Switzerland, January 13-15, 2010. Proceedings*, volume 5941 of *Lecture Notes in Computer Science*. Springer, 2010. ISBN 978-3-642-11375-8.
- [Hub00] Peter Hubwieser. *Didaktik der Informatik Grundlagen, Konzepte, Beispiele*. Springer, 2000.

- [Jer82] A. P. Jeršov. Programovanie, druhá gramotnosť, 1982.
- [JŠB00] Ľudmila Jašková, Ľubomír Šnajder, a Roman Baranovič. *Práca s Internetom*. Slovenské pedagogické nakladateľstvo, Bratislava, 2000.
- [K⁺05] Ivan Kalaš et al. *Informatika pre stredné školy*. Slovenské pedagogické nakladateľstvo, Bratislava, 2005. ISBN 80-10-00762-5.
- [Kal01] Ivan Kalaš. Čo ponúkajú IKT iným predmetom (3. časť): Informatika a informatizácia. In *Zborník príspevkov 2. celoštátnej konferencie INFOVEK*, str. 52–63. Ústav informácií a prognóz školstva, Bratislava, 2001. ISBN 80-7097-487-7.
- [Kal09] Ivan Kalaš. Pedagogický výskum v informatike a informatizácii (2. časť). In *Zborník konferencie Didinfo*, str. 15–24. Univerzita Mateja Bela, Banská Bystrica, 2009. ISBN 978-80-8083-720-4.
- [Kir10] Vladimir M. Kiryukhin. Mutual influence of the national educational standard and olympiad in informatics contents. *Olympiad in Informatics*, 4:15–29, 2010.
- [KM10] Ilham W. Kurnia a Brian Marshal. Indonesian olympiad in informatics. *Olympiad in Informatics*, 4:67–75, 2010.
- [KMH91] Ľuba Koňuchová, Božena Mannová, a Jozef Hvorecký. *Informatika a výpočtová technika — Programovanie v jazyku Pascal*. Slovenské pedagogické nakladateľstvo, Bratislava, 1991. ISBN 80-08-00753-2.
- [Kol08] Ewa Kolczyk. Algorithm – Fundamental Concept in Preparing Informatics Teachers. In Roland T. Mittermeir a Maciej M. Syslo, editors, *ISSEP*, volume 5090 of *Lecture Notes in Computer Science*, str. 265–271. Springer, 2008. ISBN 978-3-540-69923-1.
- [KSP] Korešpondenčný seminár z programovania. <http://www.ksp.sk/>.

- [Lab] London Knowledge Lab. Kaleidoscope Virtual Doctoral School in Design Based Research. <http://www.lkl.ac.uk/projects/designresearch/>.
- [lia] Programátorská liaheň. <http://liahen.ksp.sk/>.
- [LŠ01] Stanislav Lukáč a Ľubomír Šnajder. *Práca s tabuľkami*. Slovenské pedagogické nakladateľstvo, Bratislava, 2001.
- [Mac02] Jana Machová. *Práca s textom*. Slovenské pedagogické nakladateľstvo, Bratislava, 2002.
- [Mit06] Roland Mittermeir, editor. *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*, volume 4226 of *Lecture Notes in Computer Science*. Springer, 2006. ISBN 3-540-48218-0.
- [OI] Olympiáda z informatiky. <http://www.oi.sk/>.
- [Par03] Nick Parlante. Astrachan's law. *SIGCSE Bulletin*, 35(4):26–27, 2003.
- [Pol45] George Polya. *How to solve it*. Princeton University Press, 1945.
- [pzvv10a] Centrum pro zjišťování výsledků vzdělávání. Katalog požadavku zkoušek společné části maturitní zkoušky, Informatika, vyšší úroveň obtížnosti, 2010.
- [pzvv10b] Centrum pro zjišťování výsledků vzdělávání. Katalog požadavku zkoušek společné části maturitní zkoušky, Informatika, základní úroveň obtížnosti, 2010.
- [Rou10a] Pavel Roubal. *Informatika a výpočetní technika pro střední školy: Praktická učebnice*. Computer Press, 2010. ISBN 978-80-251-3227-2.
- [Rou10b] Pavel Roubal. *Informatika a výpočetní technika pro střední školy: Teoretická učebnice*. Computer Press, 2010. ISBN 978-80-251-3228-9.

- [Sal00] Ľubomír Salanci. *Práca s grafikou*. Slovenské pedagogické nakladateľstvo, Bratislava, 2000.
- [Sal05] Ľubomír Salanci. *Tvorivá informatika: 1. zošit o obrázkoch*. Slovenské pedagogické nakladateľstvo – Mladé letá, 2005.
- [SH08] Ivan Stankovský a Július Hauser. Metodika tvorby Školského vzdelávacieho programu. <http://www.statpedu.sk/sk/filemanager/download/65>, 2008.
- [ŠK05] Ľubomír Šnajder a Marián Kireš. *Informatika pre SŠ – Práca s multimédiami*. Slovenské pedagogické nakladateľstvo, Bratislava, 2005.
- [SKLŠ11] Ján Skalka, Cyril Klimeš, Gabriela Lovászová, a Peter Švec. *Informatika na maturity a prijímacie skúšky*. Enigma Publishing s.r.o, Nitra, 2011. ISBN 978-80-89132-49-2.
- [SMS08] Christopher W. Starr, Bill Z. Manaris, a RoxAnn H. Stalvey. Bloom's taxonomy revisited: specifying assessable learning objectives in computer science. In Dougherty et al. [DRFG08], str. 261–265. ISBN 978-1-59593-799-5.
- [Špú04] EXAM Štátny pedagogický ústav. Monitor 2004, pilotné testovanie maturantov, Informatika I-1, 2004.
- [Špú08] Štátny pedagogický ústav. Cieľové požiadavky na vedomosti a zručnosti maturantov z informatiky, 2008.
- [Špú10] Štátny pedagogický ústav. Cieľové požiadavky na vedomosti a zručnosti maturantov z informatiky, 2010.
- [ŠŠ⁺07] Roman Švaríček, Klára Šedřová, et al. *Kvalitatívny výzkum v pedagogických vedách: Pravidla hry*. Portál, Praha, 2007. ISBN 978-80-7367-313-0.
- [STB10a] Ľubomír Salanci, Monika Tomcsányiová, a Andrej Blaho. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika – Didaktika programovania pre SŠ 1*. 2010. ISBN 978-80-8118-079-8.

- [STB10b] Ľubomír Salanci, Monika Tomcsányiová, a Andrej Blaho. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika – Didaktika programovania pre SŠ 2*. 2010. ISBN 978-80-8118-090-3.
- [tCB09] the College Board. Computer Science A, Course Description. <http://www.collegeboard.com/>, 2009.
- [Tuc10] Allen B. Tucker. K-12 Computer Science: Aspirations, realities, and challenges. In Hromkovic et al. [HKV10], str. 22–34. ISBN 978-3-642-11375-8.
- [Uya06] Sambuu Uyanga. The current situation of informatics education in Mongolia. *Informatics in Education*, 5(1):133–146, 2006.
- [VH06] Mário Varga a Andrea Hrušecká. *Tvorivá informatika – 1. zošit s Internetom*. Slovenské pedagogické nakladateľstvo – Mladé letá, Bratislava, 2006.
- [vWA02] Tom van Weert a Jonathan Anderson. *Information and communication technology in education. A curriculum for schools and programme of teacher development*. UNESCO, 2002.
- [Vyh08] 318, Vyhláška Ministerstva školstva Slovenskej republiky z 23. júla 2008 o ukončovaní štúdia na stredných školách, 2008. §6 318/2008.
- [WGF10] Michal Winczer, František Galčík, a Michal Forišek. *Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika – Kapitoly z informatiky 3*. 2010. ISBN 978-80-8118-092-7.
- [YYK⁺06] SeungWook Yoo, YongChul Yeum, Yong Kim, SeungEun Cha, JongHye Kim, HyeSun Jang, SookKyoung Choi, HwanCheol Lee, DaiYoung Kwon, HeeSeop Han, EunMi Shin, JaeShin Song, JongEun Park, a WonGyu Lee. Development of an integrated informatics curriculum for K-12 in Korea. In Mittermeir [Mit06], str. 199–208. ISBN 3-540-48218-0.

Príloha A

Prepis rozhovoru č. 1

Na akej škole učíš, nemusíš to priamo povedať

Osemročné gymnázium v Bratislave

Súhlasíš s tým, že budeme tento rozhovor nahrávať?

Áno

Čo učíš a akú máš aprobáciu?

Informatiku učím a učím vlastne v mojej triede biológiu, čistú biológiu som vyštudovala a potom som si spravila ŠKŠ informatika, dvojročné štúdium, aby som mala špecializáciu informatika. Teraz vlastne hlavne infošku učím na tejto škole a vlastne popri tom moje triedy učím biológiu.

Čiže aj priamo aprobáciu máš biológiu a informatiku?

No, teraz už hej, teraz som si to už dorobila.

Ako dlho učíš informatiku?

Šesť rokov, šesť rokov asi alebo päť.

Na tejto škole?

Na tejto škole druhý rok.

A predtým si kde učila?

To boli základky, to bola Lazaretského a potom na Jesenského.

Ako je to na vašej škole, v ktorom ročníku sa koľko hodín učí informatika?

No, teraz sme už nabehli na ten nový systém, čiže vlastne decká majú normálne od piateho ročníka, ale predtým v osmičke a deviatke sa učila jedna hodina, to bolo namiesto technickej výchovy a na osemročnom gymnáziu sa učila iba jedna hodina do týždňa v kvarte a jedna hodina do týždňa v kvinte. A potom, keď už chceli maturovať, tak bol dvojhodinový voliteľný seminár informatiky v septime a oktáve. V podstate tam je dosť maličko detí, takže mám septimu a oktávu dokopy a majú vlastne dve a dve hodiny dokopy.

Aké okruhy učíte v spoločnej informatike a aké učíte v tých maturitných ročníkoch?

Úplne klasika podľa osnov, vlastne idem aj podľa potreby skupiny. Čiže, keď zistím, že je to šikovná skupinka, tak idem s nimi ďalej, hlbšie, každý má vlastne príležitosť, keď je šikovnejší ako tí ostatní, lebo naozaj sú to veľmi rôznorodé skupiny, tak mám pre nich pripravené alternatívy, že čo môžu robiť a môžu daťme tomu aj programovať, alebo robiť projekty, alebo mi pomáhať, alebo nemusia sa venovať trapnému skicáru, alebo niečomu takému, ak náhodou robíme, no a úplne klasické okruhy, čiže vlastne tie informácie okolo nás, informačná spoločnosť, počítačové systémy, internet, aplikačný softvér. Hlavne to beriem tak prakticky, skôr projektovo, že majú nejakú veľkú tému a vlastne tej sa venujú, na tom si odskúšajú všetko, napríklad si majú spraviť cestovnú kanceláriu, na tom sa naučia word, na tom sa naučia prácu s obrázkami, vyrobiť si stránku k tomu, alebo takéto veci. Pokiaľ prideme k programovaniu, je toho strašne veľa, čo sa tam preberá, dva roky toho je strašne máličko. Takže keď je to šikovná skupina, tak s každým aspoň mesiac programujeme, čo je asi máličko, ale stihnem aspoň také nejaké základy prebrať, záleží od skupiny. Akože s niektorými stihnem viac a s niektorými menej. Teorku si preberajú oni, majú to vo forme takých referátov, že vlastne niečo im poviem ja, také základné pojmy a to si potom rozdelíme ako témy, k tomu si vlastne oni potom rozprávajú, čiže ak je niekto šikovný a má k tomu vzťah, tak potom rozpráva ostatným. No a seminár, to je skôr taká príprava k maturitám, čiže tam hlavne programujeme. A keď sú témy, tak to vlastne hlbším spôsobom a tiež si to robia oni navzájom.

Koľko približne mávate študentov na tom seminári z informatiky?

Toto je škola zameraná na jazyky, to znamená, že aj decká sú takýmto smerom skôr orientované a keď idú maturovať, tak skôr z matiky a takýchto predmetov, takže mne, najviac,

čo mi ľudí maturovalo, tak bolo päť, minulý rok dva, tento rok mi jeden maturuje a zo semináristov, toto už je skupina, ktorú som učila v kvarte a kvinte, takže teraz sa mi tam prihlásilo snáď deväť ľudí naraz, ale oni sa potom vydesia a do toho ďalšieho ročníka potom neprejdú, alebo si povedia, že tá informatika nie je až taká, nie je to iba nejaký aplikačný softvér, takže nie všetci sa rozhodnú, že z toho budú maturovať.

Robievate nejaké projekty?

Jasné, strašne veľa.

Aj medzipredmetové?

Aj medzipredmetové, to asi ani nijak inak nejde, tam si naozaj skúšajú veci od všetkého možného. Mňa jednak to baví, aj to tak vymýšľam cestou v električke. Robili sme hudobnú spoločnosť, detiská si vyrábali videoklipy, robili sme tie cestovné kancelárie a proste kadečo. Tým, že učím aj biológiu, tak aj na biológiu to cpem, takže ... Akože robia dosť.

Takže aj s ostatnými kolegami veľmi dobre spolupracujete.

Na tejto škole hej, tu sa teraz celkom dobre spolupracuje. Aj máme tie interaktívne tabule nakúpené a takéto veci, čiže tuto sa to začína celkom dobre rozbiehať.

Čo sa týka množstva počítačov a žiakov, máte jeden počítač na jedného žiaka?

Jasné, to bola moja podmienka, že takto to chcem. Keď sme mali aj menej počítačov, tak bola trieda rozdelená na tri skupiny, aby naozaj každý ten svoj počítač mal.

Všetky hodiny mávate pri počítačoch?

Hej, aspoň na chvíľočku.

Dobre, tak teraz prejdeme k jednotlivým okruhom, prvý veľký okruh je programovanie. Takže kedy presne sa venujú programovaniu a akým štýlom sa tomu venujú?

No. V kvinte začíname, je to naozaj také to posledné, lebo viem, že algoritmické rozmýšľanie je fajn, ale ja si myslím, že sa ho dá učiť naozaj na všetkom, akože ja ich učím na všetkom fakt, že informatika je taký predmet, aby sa namakali čo najmenej, takže aby si premysleli, čo majú robiť, v akej postupnosti. Čiže naozaj to majú vo všetkom, nemusia len programovať. Príde mi, že pre nich praktické do života je vedieť využívať ten počítač na rôzne spôsoby. Teda aj na programovanie, ale hlavne na tie iné veci. Začína mimo počítača, takou hrou, aby vlastne niečo vymysleli, vyskúšali ako to funguje a tak. Začínam

turbo pascalom, úplne prvá hodina nech si vyskúšajú to čaro dosovského prostredia, že im to zapípa a povedia tam také tie príklady. To je celkom overené, naučia sa tam nejaké premenné a potom prejdeme na delphi. Je to také farebnejšie, krajšie, takže sa im to páči viac. Ale hento je na to, aby si odskúšali taký ten starý spôsob, to mi príde také, že pekné, no a prejdeme ... Nestíham, však to je maximálne tak mesiac, nejaké štyri hodinky, takže ak sa dostanem náhodou k for-cyklom s niektorými skupinami, tak je dobre. Ale tak aspoň si to ochutnajú. Potom majú programovanie hlavne semináristi. A s nimi robím už normálne delphi, vlastne to je taká príprava na maturitu.

Keď si už rovno spomínala tú hru na začiatku, že robíš nejaký algoritmus, tak majú to iba raz, alebo potom znovu, či sa niekedy k tomu vrátia.

Máme na to máličko hodín, takže sa nemám ako k tomu vrátiť, možno s tými semináristami to tak spomeniem vo vyšších ročníkoch, ale tie dva roky infošky, čo som mala k dispozícii je hrozne málo, si myslím.

Na tom seminári sa potom k tej aktivite vraciaš, aj keď vysvetľuješ nejaké procedúry a vrátiš sa k tomu, že toto je vlastne to, čo sme robili?

No, toto som v podstate ešte nevysvetľovala, ale v podstate hej. Keď mi na to príde v podstate slovo, tak áno, určite.

Čo sa týka všeobecnej informatiky, tak vieš nejak povedať percentuálne, že koľko z nich zvláda ako veľmi programovať?

Ako úplne úplne zo všetkých?

Akože, ktorí zvládajú už tak reálne programovať, popísať nejak ten algoritmus v programovacím jazyku? Tak koľko percent zvláda bez toho, aby si im ty radila, keď im dáš úlohu, tak to vedia rovno spraviť. Koľko percent zvláda, keď ich do toho nejak navádzaš, tak potom to zvládnu?

Vieš čo, tie veci, čo robím so všetkými, tak to sú také jednoduchočké veci, takže keď to vysvetlím a poviem, že čo majú robiť, tak tieto veci zvládnu naozaj všetci. Ja sa aj snažím tak, aby to zvládali všetci, tí šikovní, čo prirodzene prídu, že už pomaličky programujú v Cčku a tak, tak o tom sa nemusíme baviť, tí to vedia aj bez toho. No a na seminári, ja sa snažím až tak veľmi neradiť, chcem aby si na to prišli sami, takže tam radšej ich ešte

nechám, nech sa k tomu dokopú a radšej si to ešte pozrú na domácu úlohu, alebo ich tak trošku doľuknem, ale naozaj som si všimla, že keď im ja poradím, tak to spravia zle aj nabudúce. Čiže naozaj nech sa radšej potrápia raz a potom je menšia pravdepodobnosť, že to spravia znova.

A vieš to teda nejak percentuálne odhadnúť, koľkatí to úplne bez problémov zvládnu?

Momentálne v tejto skupine, čo mám, tak tam mám deväť ľudí a tam akurát dvaja majú z toho problémy, ale kvôli tomu, že strašne veľa vymeškávajú, že väčšinou chodia k nejakému doktorovi alebo tak. Ostatných to strašne baví, takže aj chcú, tí by mi ani nedovolili pomôcť. Tí už to berú tak, že radšej si to zoberú domov.

Potom tí ostatní to robia úplne v pohode, že dostanú úlohu a nejakým spôsobom to vyriešia?

Snažia sa, nejak sa k tomu dokopú. Ale zatiaľ berieme také vecičky, že ešte neboli problémy.

A koľko je asi tak tých, ktorí keď to dostanú, tak nemusia ani brať domov, ale rovno to nakódia?

Zhruba polovica. Lebo, keď už chce niekto maturovať, tak to už sú väčšinou takí ľudia, že už programujú, že vedia, že chcú ísť týmto smerom.

Pri tých semináristoch, kde cítiš, že sú nejaké problémy pri programovaní. V ktorej časti toho programovania?

Moji majú najviac problémov so súradnicami a so zobrazovaním, ale to aj kvôli tomu, že ja to až tak neviem vysvetliť, lebo ja mám v tom zmätok. Ja mám hroznú priestorovú predstavivosť, takže ja to potom dlho vysvetľujem, lebo ja to tam nevidím. Pre mňa je to strašné. Tým pádom to asi nemajú dobre vysvetlené a potom to tam aj oni hľadajú a niektorí to v tom aj tak jasne vidia a ja ani nerozumiem, prečo to tam vidia. Akože čo tam vlastne vidia a ide im to. Takže to z tohto dôvodu, to je moja vina. Ostatné veci, tak to čo viem vysvetliť, tak to potom zvládnem.

Ďalšia téma je netiketa, venujú sa jej nejak špeciálne, alebo iba popri ostatných?

Popri ostatných a potom tam mám jednu tému, keď si robia tie záverečné referáty na konci ročníka, tak to je jedna z tém. Takže niekto to spracuje a o tom rozpráva.

Tomu sa venujú v ktorom ročníku?

Toto hneď v kvarte.

Aj sa im nejak pripomína priamo, keď už idú maturovať?

Máme tam v maturitnej otázke takúto tému, takže jeden chalanisko si to aj teraz spracoval a tiež si tieto témy navzájom robia. A tým pádom je to jedna z tém, takže sa to nejak preberie.

Textové formáty. Ako sa tomu venujú, kedy sa tomu venujú?

Máme word, takže v tom sa snažím to nejak zovšeobecňovať, takže toto tam takto nájdete. Takže aby sa to neučili naspamäť, robíme v 2003, ale už tam máme aj nejaké 2007, to je také dobré, lebo tam sa to tým pádom vidí, že to už vyzerá inak. Takže takto a robíme to hlavne v kvarte a prejdeme vlastne všetkým, štýly a podobne. A potom majú ešte projekt, semináristi si vyrábajú vlastné skriptá maturitných otázok kvázi, takže tam takisto to formátovanie nejak

Toto bol čisto textový formát docko, ale nejak vedia rozlíšiť, kedy konvertovať do PDFka, kedy to ponechať v docku, kedy použiť iba obyčajný textový súbor?

Obyčajné textové súbory im ukazujem, keď robíme stránky a tak, takže tým sa nejak nezapodievam, pdfka som im ukazovala, ako vyrobiť a na našej stránke máme taký link priamo na pdf konvertor a v podstate mne odovzdávajú práce, keď robíme kalendár, alebo niečo také, tak lebo doma majú inú verziu, tak mi posielajú pdf. Už mi odovzdali aj nie tak, tak hneď to videli, že tam bol rozdiel.

Toto je teda všetko k textovému formátu, takže tabuľky.

Excel? Robíme, robíme v kvinte, teraz práve. Do takej miery, do akej to potrebujú, aby sme prešli jednak také formátovanie, funkcie, vzorce, nejaké filtrovanie, proste takéto nejaké základne veci. Podmienky, takže sa im to sfarbuje.

Adresáciu, relatívnu, absolútnu?

Áno, jasné. To vo vzorcoch.

Vidno z teba, že to učíš takým štýlom, že ide o to, že nejak už tam nájdeme správnu funkciu.

Áno, v podstate pár základných im predstavím, tie, ktoré viem používať aj ja a tie, ktoré neviem používať, tak bohužiaľ. Oni dostanú papierik s úlohami, ktoré potom vyrábajú.

Nie s tým, že ideme sa naučiť túto funkciu, ale ideš s tým, že ideme vyriešiť tento príklad.

Áno, jasné.

A ako dlho sa tomuto venujú?

2 mesiace

Dost.

No vieš, pre nich je to niekedy problém, keď to nevedia všetci, tak nejdem ďalej. Veľa ľudí to zvláda, ale myslím si, že to je taká praktická vec, takže do toho nerobím len excel, lebo nemám rada, keď robím dlho nejakú tému, takže do toho vkladám nejaké prezentácie, nejaký projekt.

Prezentácie. Ako robia, aj sa nejakým spôsobom učia robiť prezentácie?

Najprv im ukážem nejaké základné triky, princíp vkladania obrázkov. To všetko v kvarte. No, jednak robia mi prezentácie o sebe, takže to len kvázi na hodine a potom si to aj doma sami dokončia a chodia mi to ukazovať. A potom mám, nech sa pochvália, čo vedia o informatike, nech sa pochvália ostatným, takže z toho si robia prezentácie, no a také vyvrcholenie je, že doma robia také, čo máme tie multimediálne príbehy a rozprávky, takže robia v tomto príbeh, dávajú tam hudbu, nahrávajú tam nejaké zvuky.

A nejakým spôsobom sa aj dozvedia, ako nerobiť tie prezentácie?

Jasné.

Robíte niekedy s databázami?

Nie.

Webstránky?

Keď na to príde čas, ale nie je to nejakým zvykom, ukážem im html ako sa robí, aby vedeli aspoň čítať ten kód, nevenujeme sa tomu nejak veľa, potom radšej na seminári, ale moc tomu nevenujem, maximálne 2-3 hodinky.

A potom na seminári sa tomu venujete?

Ak to má niekto vyslovene ako srdcovku, tak on si to vyberie ako tému a je celý rád, že nemusí mať nejakú inú teoretickú tému a proste študenti sú radi, že nepočujú zase iba mňa, takže je to také dobré.

Dozvedia sa potom nejaké tie princípy, aby tie stránky robili napríklad prístupné, aby to nebolo tak, že ja som sa naučil robiť webstránky a teraz si nejak napíšem webstránku, alebo

či sa dozvedia, tak ako ste vy mali teraz na tom školení, že aby to bolo aj pre nevidomých a takéto veci?

Nie, to som ani ja nevedela, tak som to neučila.

Vyhľadávanie na internete, nejak sa tomu venujú?

Áno, máme špeciálnu celú jednu hodinu, jednak, že na internete sa dá nájsť všetko, dám im nájsť nejakú otázku, čo na internete nenájdu, niečo osobné o ňom samotnom. A mám tam päť takých náročných otázok, že je to ťažké vyhľadať. Takže prakticky si vyhľadávajú, povieme si tie princípy, to je dôležitá téma.

Grafika.

To mám rada. To majú v kvarte, aj vektorovú, aj rastrovú. Keďže sú to jedny z prvých hodín, tak si rýchlo overím, kto je na tom ako, to mi aj celkom pomáha zistiť, robia komix, alebo teda stránku z komixu, pavúk, slimák a lienka tam musia v nejakom vzťahu byť a vznikajú nádherné obrázky. Potom robíme v paint.nete, aby sa naučili vo vrstvách robiť. A potom robíme v relevatione animácie. No a vektorovú inkscape.

Na seminári sa k tomu nejak vracia?

Keď je k tomu čas.

Zvukové formáty

robíme hodinu zvuku, potom mám ja prezentáciu, niekedy aj oni, ale som si zvykla ju spraviť ja, lebo k tým príbehom potom aj nahrávame zvuk a robíme to aj prakticky, takže aj to tak preberáme.

Preberá sa to akým štýlom? Aké formáty?

No, klasický taký základný. Čiže všetko o zvuku, vzorkovaciu frekvenciu, celú tu teorku okolo toho, spomeniem im, mp3, midi, oni mi nejaké spomínajú, s čím sa stretávajú, ogg, potom také tie streamovacie, potom hlavne ja som praktik, takže im ukážem nahrávanie vo windowse, také to škaredé, čo sa dá rozťahovať, máme tu normálne sluchátka s mikrofónom, takže oni si nahrávajú.

Ďalšia téma je e-mail

To nerobím, to vedia.

Vedia aj také, že zvládajú kto je priamo príjemca tej správy, komu ide kópia, také princípy.

práveže takéto zvládajú, iba tak medzi rečou, kto nevie, tak mu to dovysvetlím, ale nevenujem tomu hodinu. Mne posielajú maily, takže vidím aj s prílohou.

Takže napríklad všetci dávajú subject - predmet správy?

Áno.

Interaktívna komunikácia?

Áno, to robia.

Prakticky všetci všetko zvládajú? Vedia aj ako vlastne fungujú, že je tam nejaký protokol?

Áno. To neviem ani ja, takže to nevedia. Možno to vedia, ale ja to fakt neviem.

Takže nevedia, že kade prechádza správa?

Nie, možno to vedia, ale ja to neviem.

Venujete sa tomuto nejak špeciálne?

Nie. Len vedia, že je, keď berieme internet a takéto vecičky.

Vedia vôbec, že sa to nazýva interaktívna komunikácia? Aj tí, ktorí majú len všeobecnú informatiku?

Áno, to vedia. Toto vedia, hej, jasne. Keď máme tému internet, tak to preberáme, takže to budú počuť.

Šifrovanie

Mám to ako tému, dosť neradi si to vyberajú ako záverečnú, akože ja ju moc nepreberám, je to skôr taká téma na seminár, takže keď mám veľkú skupinu, tak niekto si to tam vyberie. Ale nevyberú si to všetci a ja z toho nerobím nejakú veľkú vedu, keď sa náhodou nedostane na rad.

Keď potom to majú ako tému, tak čo všetko tam robia?

Ukážem ti. Tam som zahrnula, že môžu rozprávať aj o kodekoch a kompresii, šifrovanie správ, bezpečnosť, také nejaké elektronický podpis.

Napríklad tie typy šifier, idú k tomu, ako vyzerá RSAčko, ako vyzerá nejaká namakaná šifra

Niektorí áno, sú chalaniská, ktoré si toto vedia vybrať a naozaj akože idú po tomto a vyslovene vysvetľujú a potom všetci počúvajú, ale nevyberajú si to veľmi často. Ale ak mám maličkú skupinu a nevyberie si to nikto, tak mi to naozaj nevadí.

Digitálny podpis, ten je zjavne v tom šifrovaní.

Áno, ten som zahrnula do toho.

Aj sú schopní nejak používať digitálny podpis?

Oni to nejak nepotrebujú, mám ľudí, ktorí určite vedia o tom, čo to je, aj prirodzene takéto vecičky okolo toho, mám ľudí, ktorí naozaj dokázali naprogramovať slovníky, hashovanie a takéto veci, že v pohode, ale to je menšina. Ale nevenujem tomu nejakú špeciálnu hodinu, nepovažujem to za také učivo, aby to vedeli všetci.

Potom, keď by sa ich niekto spýtal, čo je to ten digitálny podpis, myslíš, že by vedeli, čo to reálne je, alebo by si mysleli, že to sa na konci mailu pripojí nejaký podpis?

Toto ti neviem povedať. Podľa mňa by väčšina nevedela.

Ďalej je reprezentácia čísel

To robíme. Aj ich učím dvojkovú sústavu a takéto vecičky, to je také celkom zábavné. Ako sa reprezentujú grafická, zvuková reprezentácia. To s nimi preberám.

Kedy a koľko?

V kvarte a mám normálne takú hodinu pripravenú, takže to robíme. A ináč, čo sa týka tých sústav, tak tomu venujem ďalšiu hodinu, čo to robia.

Ďalšiu hodinu myslíš ako?

Ešte okrem tej teorky, asi tomu venujem viacej ako jednu hodinu. To sa vlastne opakuje potom v grafike, aj vo zvuku a jednu mám takú, že to opakujem všetko naraz.

Majú vytvorený nejaký vzťah medzi tou reprezentáciou čísel a potom keď programujú, či si uvedomujú, že tam môžu nastávať nejaké chyby pri zaokrúhľovaní.

Asi nie. Lebo to si ani ja neuvedomujem.

Majú aj nejaký vzťah medzi reprezentáciou čísel a medzi typmi premenných, akože integer.

Vedia, že čo je integer, aké to má byť veľké, to si uvedomujú, ale ani ja si to neviem spojiť s tým.

Kódovanie a kompresia

Počuj, ja z tej teorky, ja som fakt taký praktický typ, že fakt ja tú teorku beriem tak, že vedia to, ale potom hlavne aby to vedeli spraviť.

A zvládajú to potom aj na maturite?

Na maturite to zvládajú, to sú ľudkovia, ktorí si to vybrali, to sú ľudia, ktorí aj keby som to neprebrala, tak to sami vedia.

Potom ešte nejaká ta kompresia

Áno, prakticky, aj čo to je, aj pri zvuku, nevenujem tomu špeciálnu tému. Keď strašne veľa vecí sa dá venovať takto popri tom, príde na to slovo. Príde slovo na všetko, aj je to také spontánnejšie, takže to je lepšie podľa mňa.

Ešte posledné je počítačové systémy a siete, to znamená vstupno-výstupné zariadenia

Hardvér preberáme, máme tu aj také pikošky, takže to si radi chytia do ruky, taký princíp, ako to funguje, tak trochu aspoň také základné veci, aby vedeli, ako to funguje. Veľkí chalani na seminároch s tým žijú, baví ich to viac ako mňa. Takže tam si robia oni prezentácie a potom im žiaria očičká, ako si to odprezentujú a ja tak počúvam a hrozne sa nudím. A siete takisto, ale skôr na tom seminári.

Tak to je odo mňa všetko, ďakujem za rozhovor.

Prosím.

Príloha B

Vedomostno-postojový dotazník

- (1) Ktorá hodina sa Ti najviac páčila?
- (2) Ktorá hodina sa Ti najviac nepáčila?
- (3) Je niektorá téma, ktorú si neočakával na informatike?
- (4) Čo mi ešte chceš povedať?
- (5) Predstav si, že máš farebný obrázok 100×100 bodov. Koľko by mal zaberáť miesta vo formáte .bmp (bitmapa)? (Napíš aj prečo.)
- (6) Do premennej integer sa dajú zapisovať čísla v rozpätí $-2147483648 \dots 2147483647$, čo sú čísla $(-2^{31}) \dots (2^{31} - 1)$. Koľko bitov zaberá integer v pamäti?
- (7) Vchádzaš na webstránku, ktorá by mala byť bezpečná (z hľadiska šifrovania komunikácie). Čo všetko by si mal skontrolovať?
- (8) Čo je to digitálny podpis?
- (9) Aké kódovanie používaš, keď píšeš slovenský text?

Príloha C

Médium s kópiou webového portálu