



Ďalšie vzdelávanie učiteľov
základných škôl a stredných škôl
v predmete *informatika*



ŠTÁTNY PEDAGOGICKÝ ÚSTAV
NATIONAL INSTITUTE FOR EDUCATION

Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika

Tvorba úloh a hodnotenie žiakov v predmete informatika

Predmet: Didaktika všeobecnovzdelávacieho predmetu Informatika

Línia: Didaktika informatiky a informatickej výchovy



EURÓPSKA ÚNIA



Európsky sociálny fond



Európska únia
Európsky sociálny fond

Moderné vzdelávanie pre vedomostnú spoločnosť/Projekt je spolufinancovaný zo zdrojov ES



Dalšie vzdelávanie učiteľov
základných škôl a stredných škôl
v predmete *informatika*

Tvorba úloh a hodnotenie žiakov v predmete informatika

Identifikácia modulu

Aktivita projektu: 1.2 Vzdelávanie nekvalifikovaných učiteľov informatiky na 2. stupni ZŠ a na SŠ

Línia aktivity: Didaktika informatiky a informatickej výchovy

Predmet: Didaktika všeobecnovzdelávacieho predmetu Informatika

Zaradenie modulu



Tento modul je druhým modulom predmetu Didaktika všeobecnovzdelávacieho predmetu informatika.

Didaktika všeobecnovzdelávacieho predmetu informatika				
Aktivizujúce metódy vo výučbe školskej informatiky (2DidInf1)	Tvorba úloh, hodnotenie žiakov v predmete informatika (2DidInf2)	Metodika výučby tematickej oblasti Informácie okolo nás (2DidInf3)	Metodika výučby tematickej oblasti Komunikácia prostredníctvom IKT (2DidInf4)	Metodika výučby tematickej oblasti Princípy fungovania IKT a oblasti Informačná spoločnosť (2DidInf5)

Abstrakt modulu

Účastníci vzdelávania sa v prvej časti modulu oboznámia s pojmom učivo a jeho prvkami, učebná úloha, typológiou úloh, didaktickými funkciami úloh (motivačná, expozičná, fixačná, systemizačná, diagnostická, aplikovaná), požiadavkami na systém úloh.

V druhej časti modulu sa oboznámia s problematikou hodnotenia žiakov v predmete informatika - funkcie a typy hodnotenia, špecifikácia cieľov vyučovania, ukážky vybraných typov hodnotení žiakov (didaktické testy, projekty, slovné hodnotenie, samohodnotenie atď).

V závere modulu sa účastníci vzdelávania oboznámia s požiadavkami na didaktický projekt a naučia sa navrhovať a vytvárať didaktický projekt pre vybranú tému školskej informatiky v súlade s požiadavkami ŠVP.

Garant predmetu:

RNDr. Ľubomír Šnajder, PhD.
UPJŠ v Košiciach
lubomir.snajder@upjs.sk

Autori:

Mgr. Ján Guniš
UPJŠ v Košiciach
RNDr. Ľubomír Šnajder, PhD.
UPJŠ v Košiciach



<http://dvui.ccv.upjs.sk/kurzy/>

Tvorba úloh a hodnotenie žiakov v predmete informatika	1
Identifikácia modulu	1
Zaradenie modulu	1
Abstrakt modulu	1
Obsah	2
Úvod	3
Cieľ modulu	3
Vstupné vedomosti	3
Požadované prerekvizity	3
Predpokladané vstupné vedomosti, skúsenosti a zručnosti	3
Učebné úlohy a hodnotenie žiakov	4
Kapitola 1: Úlohy v školskej informatike	4
Kapitola 2: Hodnotenie žiakov v predmete informatika	12
Kapitola 3: Tvorba didaktického projektu	19
Čo sme sa naučili v tomto module	22
Preverenie výstupných vedomostí	23
Literatúra a použité zdroje	23
Príloha 1: Identifikačná časť didaktického projektu	24
Príloha 2: Logická štruktúra učiva	25
Príloha 3: Očakávané výsledky	26
Príloha 4: Systém úloh	27
Príloha 5: Didaktický test	36
Príloha 6: Návod na riešenia aktivít	38

Úvod

V predchádzajúcom module sme sa zaoberali problematikou aktivizujúcich metód - ich prehľad a spôsobom ich zaradenia do výučby školskej informatiky. V tomto module sa sústredíme na učivo, jeho štrukturálne prvky a tiež na učebné úlohy, ich zostavovanie a spôsoby ich používania vo výučbe informatiky. Ďalej sa budeme zaoberať problematikou hodnotenia žiakov v predmete informatika.

Pri práci s týmto študijným materiálom odporúčame, aby mal účastník vzdelávania k dispozícii počítač s webovým prehliadačom a s pripojením na internet, aplikácie kancelárskeho balíka, softvér pre tvorbu pojmových a myšlienkových máp (napr. XMind, FreeMind), farebnú tlačiareň.

FreeMind si môžete stiahnuť z webovej stránky: <http://freemind.sourceforge.net/>.

XMind si môžete stiahnuť z webovej stránky: <http://www.xmind.net/>.

Cieľ modulu

V module 2DidInf2 účastníci vzdelávania získajú základné poznatky o učive a jeho prvkoch, učebných úlohách, typológii úloh, didaktických funkciách úloh, o požiadavkách na systém úloh. Oboznámia sa s problematikou hodnotenia žiakov v predmete informatika, funkciami a typmi hodnotenia, špecifikáciou cieľov vyučovania, ukážkami vybraných typov hodnotení žiakov. Oboznámia sa s požiadavkami na didaktický projekt a naučia sa navrhovať a vytvárať didaktický projekt pre vybranú tému školskej informatiky v súlade s požiadavkami ŠVP.

Tento modul spolu s prvým modulom sa venujú všeobecnými aspektmi vyučovania informatiky, ktoré sú východiskom pri tvorbe metodík vyučovania tematických oblastí opísaných v ďalších troch moduloch predmetu Didaktika informatiky (2DidInf3, 2DidInf4, 2DidInf5).

Vstupné vedomosti

Požadované prerekvizity

Účastník vzdelávania úspešne absolvoval vzdelávanie v moduloch:

- 2DG1 Základná digitálna gramotnosť
- 2DG2 Digitálna gramotnosť učiteľa
- 2DidInf1 Aktivizujúce metódy vo výučbe školskej informatiky

Predpokladané vstupné vedomosti, skúsenosti a zručnosti

Účastník vzdelávania:

- pozná a vie efektívne využívať nástroje pre spracovanie textov na počítači,
- pozná a dodržiava základné pravidlá počítačovej typografie,
- vie vytvárať počítačové prezentácie, pozná základné pravidlá prezentovania,
- rozumie základným princípom spracovania grafiky, zvuku a videa na počítači,
- pozná a vie efektívne používať vhodné nástroje pre spracovanie grafiky, zvuku a videa, vie ich vhodne kombinovať,
- má základné poznatky o systéme didaktiky a jeho činiteľoch,
- má prehľad o aktivizujúcich metódach a ich využití vo výučbe informatiky,
- vie navrhnuť metodický list pre vybranú tému školskej informatiky vyučovanú s využitím vhodných aktivizujúcich metód.

Učebné úlohy a hodnotenie žiakov

Študijný materiál pozostáva z troch kapitol:

1. Úlohy v školskej informatike
2. Hodnotenie žiakov v predmete informatika
3. Tvorba didaktického projektu

Kapitola 1: Úlohy v školskej informatike

Učivo a jeho prvky

Pri plánovaní výučby informatiky musí učiteľ premyslieť ciele a obsah vzdelávania (učivo) t.j. čo a na akej úrovni sa má žiak naučiť. Samotné učivo pozostáva z rôznych typov prvkov. Podľa (Bajtoš, 2003) **učivo** tvorí systém vedomostí, zručností a návykov, osvojenie ktorých zabezpečuje rozvoj duševných a fyzických schopností žiakov potrebných pre ich ďalší život.

Podľa (Lerner, 1986) rozlišujeme štyri **prvky učiva**:

- **Poznatky** - fakty, pojmy, princípy, vzťahy, zákony, teórie, ... (žiakom osvojené poznatky nazývame **vedomosťami**).
- **Zručnosti a návyky** - skúsenosti z realizácie už známych spôsobov činnosti, zahrňujú v sebe vedomosti a operácie s nimi (rozlišujeme viaceré typy zručností - pracovné, sociálnej komunikácie a správania, pohybové a zdravotné, poznávacie).
- **Skúsenosti z tvorivej činnosti** - skúsenosti z riešenia nových problémov.
- **Emocionálne-citové skúsenosti** - podmieňovanie mravných, estetických a emocionálnych stimulov, potrieb a ideálov, hodnotového systému.

Vo vyučovaní informatiky nesmie chýbať žiaden z uvedených prvkov učiva. Vedomosti, zručnosti a návyky plnia **vzdelávaciu funkciu** vzdelávacieho procesu, skúsenosti z tvorivej činnosti plnia **rozvíjajúcu funkciu** a emocionálno-citové skúsenosti plnia **výchovnú funkciu** vzdelávacieho procesu.

Pre lepšie pochopenie rozličných prvkov učiva v predmete informatika uvádzame príklady prvkov učiva informatiky:



Koncept štruktúry vedomostí podľa Jerome Brunera, podľa ktorého by sa výučba mala sústrediť na generalizácie a pojmy predmetu, nie na jednotlivé fakty. (Pasch, 1998)

- **Fakty:**
 - „V roku 1945 John Presper Eckert, John William Mauchly, Herman Goldstine a Alan M. Turing skonštruovali prvý univerzálny plne elektronický počítač ENIAC“,
 - „Príkladom textových procesorov sú napr. MS Word 2007, Open Office Writer, Google Docs Writer“,
 - „Príkladmi operačných systémov sú Windows, Linux, Unix, Mac OS, BSD“,
 - „Správcu úloh v operačnom systéme MS Windows spustíme stlačením klávesov Ctrl+Alt+Del“.
- **Pojmy (koncepty):**
 - operačný systém,
 - sekcia (oddiel) v textovom procesore,
 - príkaz opakovania,
 - počítačový vírus,
 - digitálny podpis.
- **Princípy, vzťahy medzi pojmi a javmi, generalizácie:**
 - von Neumannova koncepcia počítača,
 - princíp výpočtového modelu klient-server,
 - princíp hromadnej korešpondencie,
 - princíp vyhladzovania hrán pri počítačovej grafike.

- **Poznatky o metóde, postupe, algoritme:**
 - metóda „rozdeľ a panuj“ (lat. divide et impera),
 - metóda návratu (angl. backtracking),
 - postup pri odvírusovaní počítača,
 - pažravý (greedy) algoritmus,
 - algoritmus bublinkového usporadúvania.
- **Hypotézy, zákony, teórie, pravidlá:**
 - Moorov zákon,
 - pravidlá počítačovej typografie,
 - pravidlá sieťovej etikety.
- **Spôsobilosti a zručnosti:**
 - vytvorenie prezentácie obsahujúcej snímky s textami, obrázkami, zvukmi, vlastnými animáciami, hypertextovými prepojeniami, ktorá využíva predlohu snímok,
 - vytvorenie obsahu a registra v textovom procesore,
 - zlúčenie hlavného dokumentu hromadnej korešpondencie so zdrojom dát, použitie filtrovania záznamov a odoslanie dokumentov prostredníctvom elektronickej pošty.
- **Návyky:**
 - pri prijímaní e-mailovej správy otestovať antivírusovým programom prílohu správy, čistenie a triedenie správ,
 - ukladanie súborov so zmysluplným názvom do premyslenej štruktúry priečinkov,
 - pravidelné zmeny hesiel pre prístup na servery a aplikácie,
 - pravidelné zálohovanie dát,
 - pravidelná údržba počítača (aktualizácia OS, antivírusového programu, defragmentácia diskov, čistenie registra),
 - pravidelné hygienické prestávky pri práci s počítačom s cvičením.
- **Skúsenosti z tvorivej činnosti:**
 - navrhnutie a naprogramovanie vlastnej počítačovej hry,
 - tvorba rodinného rozpočtu a monitorovanie jeho plnenia,
 - monitorovanie prípustnej hladiny hluku, osvetlenia v priestoroch školy, vytvorenie odbornej správy, prednesenie výsledkov merania vedeniu školy a prediskutovanie možných riešení.
- **Emocionálno-citové skúsenosti:**
 - kolaboratívna tvorba myšlienkovkej mapy (resp. nejakej spoločnej kresby) k problematike počítačová kriminalita,
 - diskusia k prevencii, príznakom počítačovej (internetovej) závislosti.

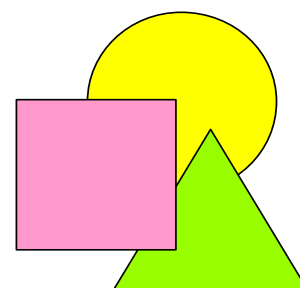
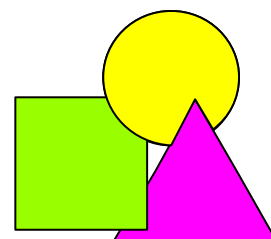
- Porozmýšľajte o doplnení prvkov učiva informatiky o ďalšie príklady. Na ktoré prvkov učiva ste sa doposiaľ najviac sústreďovali pri vyučovaní informatiky?
- Pre vybranú tému školskej informatiky prediskutujte konkrétne prvkov učiva (pojmy, princípy, postupy, zručnosti, spôsobilosti, návyky, ...)

Jednotlivé prvkov učiva prispievajú k rozvoju žiaka vo všetkých troch **doménach**: **kognitívnej** (poznatky, zručnosti, návyky, skúsenosti z tvorivej činnosti), **afektívnej** (emocionálno-citové skúsenosti) aj **psychomotorickej** (zručnosti, návyky). Vzhľadom na charakter predmetu informatika sa budeme zaoberať viac kognitívnymi a afektívnymi cieľmi, menej psychomotorickými cieľmi.

Pri plánovaní učiva okrem výberu uvedených prvkov učiva rozvíjajúcich kognitívnu doménu žiaka je veľmi dôležité uvedomiť si, že existujú rôzne úrovne „ponorenia sa“ do danej problematiky a rôzne úrovne myslenia žiaka použité pri jej štúdiu.

Porovnajzte nasledovné situácie z pohľadu použitia myšlienkových operácií.

1. Žiak reprodukuje naučený algoritmus binárneho vyhľadávania, pričom uvedie program zapísaný vo vybranom programovacom jazyku.



Tri domény vzdelávacích cieľov:

- **kognitívna** - vyžaduje od žiaka, aby si zapamätal a dokázal vybaviť určité informácie, aby zistil, určil, odvodil význam informácie a zaradil ju ako novú informáciu do vzťahu a súvislosti s predošlými vedomosťami,
- **afektívna** - týka sa prejavov emocionálneho chovania - citov, postojov, preferencií, hodnôt,
- **psychomotorická** - zahŕňa zmyslové učenie - pozorovanie, počúvanie, rozprávanie, koordinácia svalov a svalových skupín, (Pasch, 1998)

2. Žiak použije pri hre Háďaj číslo stratégiu delenia intervalu na polovicu, pričom svoj postup popíše len slovne.

Na príklade výučby témy Hromadná korešpondencia ukážeme, aké rôzne vzdelávacie ciele môžeme definovať. Napr. žiak vie:

- vymenovať aplikácie, pomocou ktorých vieme urobiť hromadnú korešpondenciu,
- uviesť príklady využitia hromadnej korešpondencie v praxi,
- vysvetliť vlastnými slovami pojmy:
 - hlavný dokument,
 - zdroj údajov,
 - zlučovacie polia,
- vysvetliť princíp hromadnej korešpondencie,
- zlúčiť hotové pripravené dokumenty (hlavný dokument so zdrojom údajov) do:
 - súboru (listy, obálky, vizitky),
 - na tlačiareň,
 - do e-mailu,
- filtrovať záznamy hromadnej korešpondencie podľa rôznych kritérií,
- pre zadaný problém (typ korešpondencie) vybrať zlučovacie polia vhodné pre použitie v hlavnom dokumente hromadnej korešpondencie,
- zdôvodniť význam použitia hromadnej korešpondencie v praxi a jej dopad na efektívnosť práce,
- rozpoznať situáciu použitia hromadnej korešpondencie, navrhnúť a vytvoriť hlavný dokument so zlučovacími poľami, štruktúru zdroja údajov a spôsob jeho uloženia (textový procesor, tabuľkový kalkulačor, systém riadenia databázy), výber spôsobu a realizácia zlúčenia vytvoreného hlavného dokumentu so zdrojom údajov.

Taxonómia cieľov
v **afektívnej oblasti** podľa
D. R. Krathwohla:

1. Prijímanie (vnímavosť)
2. Reagovanie
3. Oceňovanie hodnoty
4. Integrovanie hodnoty
5. Začleňovanie hodnoty do charakterovej štruktúry osobnosti

Taxonómia cieľov
v **psychomotorickej oblasti**
podľa M. Simpsona:

1. Vnímanie činnosti, zmyslová činnosť
2. Pripravenosť na činnosť
3. Napodobňovanie činnosti, riadená činnosť
4. Mechanická činnosť, zručnosť
5. Komplexná automatická činnosť
6. Prispôsobovanie, adaptácia činnosti
7. Tvorivá činnosť

(Bajtoš, 2003)

Niektoré z uvedených cieľov sa týkajú vedomostí, iné zručností, niektoré vyžadujú reprodukčnú činnosť, iné tvorivosť žiaka. Je v rukách učiteľa, ktoré z uvedených cieľov vyberie pre svoju výučbu v súlade s profilom absolventa danej školy a pridelenej časovej dotácie na túto tému.

Pri určovaní hĺbky vedomostí, zručností na rôznych úrovniach myslenia žiaka využívame niektorú zo zaužívaných **taxonómií kognitívnych cieľov**. Najznámejšou z nich je **Revidovaná Bloomova taxonómia (RBT)** pozostávajúcej zo šiestich úrovní s charakteristickými aktívnymi slovesami. (Anderson, 2001)

Vytvoriť - vytvoriť pôvodný a zložitý produkt z jednoduchších komponentov (napr. tvorba unikátnych digitálnych príbehov, blogov, podcastov, wiki, webquestov).

Vyhodnotiť - stanoviť hodnoty produktu na základe skôr naučených noriem a kritérií, posudzovať prácu druhých aj vlastnú prácu.

Analyzovať - rozčleniť zložitú vec na jednoduchšie, vysvetliť spôsob alebo príčiny jej usporiadania.

Aplikovať - použiť skôr naučené informácie pri spracovaní nových informácií.

Porozumieť - vyjadriť vedomosti vlastnými slovami.

Zapamätať si - znovuvybaviť, reprodukovať, rozoznať vedomosti.

Vytvoriť

*vytvoriť, zostaviť, skonštruovať, modifikovať,
kombinovať, organizovať, predpovedať*



Obrázok 1: Revidovaná Bloomova taxonómia kognitívnych cieľov s uvedením aktívnych slovies

Zatriedte spomínané kognitívne ciele k problematike výučby témy Hromadná korešpondencia do jednotlivých úrovní Revidovanej Bloomovej taxonómie.

Pri plánovaní učiva každý učiteľ realizuje rozbor obsahu učiva s cieľom vystihnúť jeho výchovnú a vzdelávaciu hodnotu, čo označujeme pojmom **didaktická analýza učiva**. Pozostáva z troch častí (Bajtoš, 2003):

- **pojmovej analýzy** (vytvorenie logickej štruktúry pojmov a vzťahov medzi nimi s ohľadom na hierarchiu pojmov podľa miery ich všeobecnosti),
- **operačnej analýzy** (analýza činnosti a operácií žiakov k dosiahnutiu cieľa, vyplynie z rozboru učebných úloh, je reprezentovaná výberom učebných úloh),
- **analýzy učiva vzhľadom k medzipredmetovým vzťahom** (rozbor časovej a obsahovej nadväznosti učiva).

Podrobnejšie a názornejšie sa problematike didaktickej analýzy učiva budeme venovať v kapitole Tvorba didaktického projektu.

Učebné úlohy

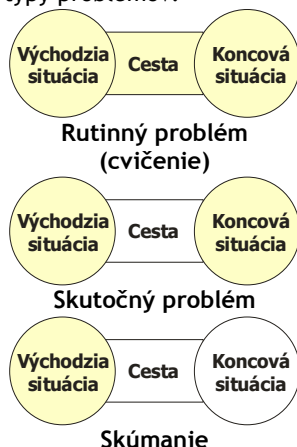
V súlade s (Kalhous, 2002, Šveda, 1992) pod **učebnou úlohou** rozumieme každú požiadavku na merateľnú činnosť žiaka zameranú na dosiahnutie cieľov učenia sa. V tomto širokom ponímaní príkladmi úlohy sú - **otázka, diskusia, príklad, cvičenie, problém, projekt**, atď.

Vo vyučovaní informatiky používame učebné úlohy ako prostriedok aktivizácie žiakov, spätnej väzby a riadenia výučby. (Kalhous, 2002)

Podľa (Kopka, 2004) problém pozostáva z troch zložiek:

1. Východisková situácia
2. Cesta (od východiskovej situácie k cieľu)
3. Cieľ (koncová situácia)

Podľa úrovne presnosti popisu jednotlivých zložiek (otvorenosti/uzavretosti) rozlišujeme nasledovné typy problémov:



Pri rutinných úlohách máme všetky zložky presne určené (uzavreté), pri tvorivých úlohách máme viacero zložiek otvorených.

M. Zelina uvádza štyri stupne tvorivých úloh v závislosti od miery otvorenosti zadania resp. výsledku úlohy a miery konvergentnosti riešenia úlohy. (Zelina, 1990)

Manipuláciou so zadaním úlohy (podmienkami úlohy) vieme zjednodušiť resp. sťažiť riešenie danej úlohy. Napríklad:

Robot Karel stojí:

- v konkrétnej pozícii, v neznámej pozícii, v nejakej výhodnej pozícii,
- v miestnosti s danými rozmermi, s neznámymi rozmermi,
- v prázdnej miestnosti, v čiastočne obsadenej miestnosti.
- po vykonaní úlohy na pôvodnej, ľubovoľnej pozícii.

Prostredníctvom učebných úloh rozvíjame kognitívnu, afektívnu aj psychomotorickú oblasť. Viacerí autori (Holoušová, 1983, Šveda, 1992) odporúčajú projektovať **súbory (systémy) učebných úloh**. Odporúčame, aby zostavené systémy úloh spĺňali nasledovné kritériá:

- pokryť nielen **kognitívne**, ale aj **afektívne** a **psychomotorické** ciele výučby danej témy,
- orientovať sa viac na **vyššie myšlienkové operácie** (nápomocné sú formulácie úloh typu „vytvorte riešenie“, „modifikujte riešenie, resp. upravte, doplňte, zameňte, prehodte, zrušte“, „analyzujte riešenie/čo robí daný program?“, „nájdite chybu v riešení“, „porovnajte uvedené riešenia“ atď),
- zaradiť úlohy s **postupne gradovanou náročnosťou** (s použitím jedného, či viacerých kontextov) s ohľadom na jednotlivé štádiá poznávacieho procesu,
- zohľadniť **etapy vyučovacieho procesu** (zastúpené úlohami so zodpovedajúcimi didaktickými funkciami)
 - **motivačná**,
 - **expozičná** (prvotné osvojenie poznatkov, prípravné, propedeutické),
 - **fixačná** (upevňovacie, proti stereotypu, opakovacie, systemizačné),
 - **diagnostická**,
 - **aplikačná**.

Každá úloha má tri kľúčové parametre (Švec, 2004):

- **Stimulačný (motivačný) parameter** (navodenie záujmu o poznávanie)
- **Operačný parameter** (navodenie a rozvíjanie učebných operácií, odpovedajúce vzdelávacím cieľom)
- **Regulačný parameter** (učebné úlohy regulujú činnosť žiakov pri ich riešení, regulačnú silu úlohy ovplyvňuje jej určenosť, t.j. presnosť určenia jej zložiek, heuristickosť úlohy)

Chyby žiaka pri riešení by sa mali včas detekovať, identifikovať, interpretovať a korigovať. Prostredníctvom diferenciacie učebných úloh, môžeme realizovať individuálny prístup k žiakom, a to zadávaním rôzneho počtu učebných úloh a ich náročnosti na myšlienkové operácie a tiež miery pomoci žiakovi.

Príklady nesprávnych zadaní úloh:

- používanie uzavretých otázok uvedených opytovacími zámenami (čo?, ako?, kedy?), na ne môžu žiaci odpovedať jedným slovom, či holou vetou - nekultivuje jazykový prejav žiaka, pokrývajú len menej náročné myšlienkové operácie žiaka,
- všeobecná formulácia (povedz niečo o ...),
- vyjadrenie nedokončenou oznamovacou vetou,
- kumulovanie viacerých otázok za sebou, bez vyjadrenia požadovaného výkonu.

Na ilustráciu pokrytia vybranej problematiky uvádzame na obrázku 2 spôsob pokrytia kognitívnych cieľov jednotlivými úlohami súťaže PALMA junior. (Guniš, 2007, 2009)

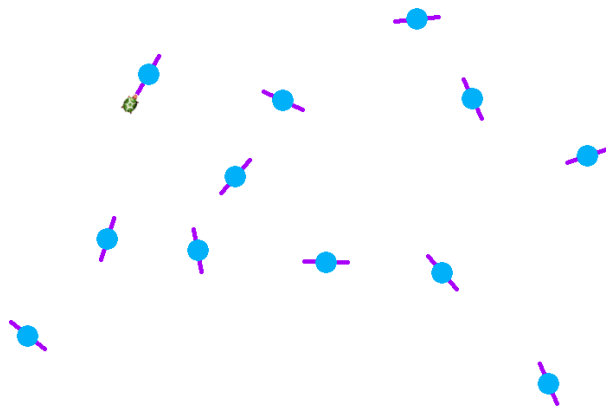
	Programovanie												Algoritmy						Matematika													
	korytnačia grafika	Karteziansky súradnicový systém	procedúry bez parametrov	procedúry s parametrami	príkazy opakovania - zadany počet krát	príkazy opakovania - s podmienkou	vnorené príkazy opakovania	podmienky	premenne - používanie, menenie, operácie	náhodnosť	vizuálne komponenty, udalosti	použitie viacerých korytnačiek	rekúzia	zoznamy	korytnačka ako vykonávateľ algoritmu	prechádzanie cez 2D pole	prechádzanie labyrintom, testovanie podmienky	náhodnosť vo vymedzenom intervale	kreslenie tvarov s určitými vlastnosťami	algoritmy usporadúvania	algoritmy vyhľadávania	hľadanie a vyhodnotenie výrazov	relatívna orientácia v korytnačej geometrii	orientácia v Kartezianskom súradnicovom systéme	vlastnosti rovinných objektov	výpočty s uhlami	postupnosti čísel	aritmetické funkcie	Pythagorova veta	výpočet pozície v 2D poli	faktoriácia čísel	prevody jednotiek
Vlajka EÚ	*		*		*		*		*						*							*	*	*	*	*						*
Hodiny	*			*					*						*							*	*	*	*	*						
Mikuláš	*	*			*										*							*	*	*	*	*						
Pretekárska loptička	*					*			*						*		*					*	*	*	*	*						
Špirála	*				*				*						*							*	*	*	*	*						
IQ test	*	*							*		*				*							*	*	*	*	*	*	*	*	*	*	*
Šetrič obrazovky	*			*	*				*	*	*				*			*				*	*	*	*	*	*	*	*	*	*	*
Tapetovanie	*		*	*	*	*	*	*	*						*	*	*					*	*	*	*	*	*	*	*	*	*	*
Snehuliak	*			*	*	*	*	*	*						*							*	*	*	*	*	*	*	*	*	*	*
Hviezda	*	*	*	*	*	*	*	*	*						*		*		*			*	*	*	*	*	*	*	*	*	*	*
Pyramída	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Hra uhádní číslo					*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Mince	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Vločka	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Vlajky štátov EÚ	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Hra uhádní štát	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Cestovanie po EÚ	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Tekvica	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Stupne víťazov	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Terč	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Klaunove domčeky	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Stupne víťazov 2	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Akvárium	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Postavička z písmen	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Hrad	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Ostrov pokladov	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Rodičia včely	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Lienka	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Mušľa	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

Obrázok 2: Pokrytie kognitívnej zložky cieľov v oblasti programovania, algoritmizácie a matematiky úlohami PALMA junior

Ukázkou pokrytia afektívnych a kognitívnych cieľov je úloha o cukríkoch v súťaži PALMA junior (Guniš, 2009):

Mamka korytnačka kúpila svojim šikovným detičkám cukríky. Keď prídu zo školy, chce ich prekvapiť a pekne im usporiadať cukríky na stolík. Nie tak neporiadne, ako je to na obrázku.

Napiš procedúru usporiadať, ktorá vykreslí cukríky v ľubovoľnom, ale pravidelnom usporiadaní. Nezabudni na to, že mamky korytnačky môžu mať rôzne počty detí. A deti majú rady cukríky aj iných farieb, než aké sú nakreslené na obrázku. Čo si myslíš, dá sa podľa uvedeného obrázka s istotou povedať, koľko mala mamka detí? Napiš nám svoje vysvetlenie. Svoje riešenie ulož ako Imagine projekt pod názvom cukríky.imp.



Ukážky zadání úloh:

Motivačná úloha
(príkazy opakovania, algoritmus binárneho vyhľadávania)

Vytvorte hru, v ktorej počítač uhádne myslené číslo zo zadaného intervalu.

Motiváciu vieme realizovať aj prostredníctvom príbehu, diskusie, demonštrácie.

Úloha na prvotné osvojenie poznatkov
(práca s vektorovou grafikou)

Vytvorte vlastné logo (dopravnú značku, vlajku štátu) pozostávajúce zo základných geometrických útvarov.

Upevňovacia úloha
(práca so zvukom)

Nahrajte najnižší tón, ktorý viete zaspievať a vypočítajte jeho frekvenciu.

Nahrajte výslovnosť anglických slov (napr. zvieratá) a upravte ich (orežte hluché miesta, nastavte rovnakú hlasitosť pri všetkých slovách) a uložte ich do formátu OGG.

Systemizačná úloha

(práca s tabuľkami, rôzne typy dát, funkcie súčet a priemer, podmienené formátovanie, tvorba grafov)

Vytvorte tabuľku s prehľadom nakúpených potravín z viacerých nákupov s uvedením ich ceny, gramáže, kalorickej hodnoty, zaradením potraviny do kategórie zdravá/nezdravá.

Aplikačná úloha

(tvorba prezentácií)

Vytvorte multimediálnu prezentáciu s predstavením vašej virtuálnej firmy (napr. v rámci projektu Junior Achievement), s priebežnými výsledkami jej hospodárenia.

(hromadná korešpondencia)

Vytvorte a pošlite pozvánky pre svojich spolužiakov na konkurz do muzikálu na výročnú školskú akadémiu.

Aplikačné úlohy môžeme zamerať na rôzne vyučovacie predmety, život školy, rodiny, spoločnosti, záujmy žiakov atď.

Propedeutická úloha

(na časovú a priestorovú zložitost')

Robot Karel zabľúdi v lese. Napíš príkaz, pomocou ktorého sa dostane do výšky 10 tehličiek, aby sa rozhlíadal, kde sa nachádza. Koľko času strávil pri tomto šplhaní, keď každý jednoduchý príkaz trvá 1s? Koľko tehličiek pri tom použil, akú plochu pri tom zabral?

V tejto úlohe sme okrem estetiky (vykreslenia pravidelného usporiadania cukríkov) sledovali aj vzájomný pozitívny vzťah matky a detí (matka má rada deti a chce ich odmeniť) a tiež zmysel pre spravodlivosť (spravodlivé rozdelenie odmeny všetkým deťom). Úloha je divergentná, čím prispieva k rozvoju tvorivého myslenia. Ďalej sme úlohou sledovali rozvíjanie matematického myslenia (aplikovanie poznatkov o deliteľnosti čísel), jeho dôslednosť (určenie všetkých deliteľov čísla), rozvíjanie kritického myslenia (zdôvodnenie riešenia), aplikáciu príkazov opakovania, vytvorenie algoritmu na pravidelné vykreslenie cukríkov atď.

Pri riešení náročnejších úloh je dôležité, aby žiak vedel použiť niektorú zo stratégií riešenia problému (napr. rozloženie problému do podproblémov, prevedenie problému na známy problém, riešenie problému od konca atď.). Učiteľ môže žiakovi uľahčiť riešenie náročnejšej úlohy tým, že mu poskytne jednoduchšie úlohy, ktoré ho pripravujú na riešenie danej úlohy. V príprave na matematické a programátorské súťaže sa zvyknú používať sady prípravných úloh.

Na konkrétnej úlohe ukážeme zadania jej prípravných úloh. (Šnajder, 2008)

Nakreslite vlajku s danou šírkou a výškou s tromi farebnými pruhmi s horizontálnou resp. vertikálnou orientáciou.

Tieto úlohy by mohli predchádzať rôzne prípravné úlohy napr.:

- *Nakreslite štvorec s konkrétnou dĺžkou strany.*
- *Nakreslite obdĺžnik s konkrétnou dĺžkou strán.*
- *Nakreslite obdĺžnik s konkrétnou dĺžkou strán vyfarbený konkrétnou farbou.*
- *Nakreslite obdĺžnik s danou dĺžkou strán vyfarbený danou farbou.*
- *Nakreslite zástavu s danou dĺžkou strán s tromi vertikálnymi pruhmi vyfarbenými tromi danými farbami.*

Na druhej strane vieme pôvodnú úlohu ďalej rozvíjať (viac zovšeobecňovať, parametrizovať) napr. pomocou nasledovných zadaní:

- *Nakreslite zástavu s danou dĺžkou strán s tromi pruhmi rôznej šírky vyfarbenými tromi danými farbami s danou orientáciou.*
- *Nakreslite zástavu s danou dĺžkou strán s daným počtom pruhov rovnakej šírky vyfarbenými danými farbami s danou orientáciou.*
- *Nakreslite zástavu s danou dĺžkou strán s daným počtom pruhov rôznej šírky vyfarbenými danými farbami s danou orientáciou.*

Pri zostavovaní systémov úloh je dobré mať k dispozícii **zbierku úloh**. Ideálne je, ak máme k dispozícii elektronickú zbierku úloh, v ktorej sú jednotlivé úlohy uložené s rôznymi atribútmi, napr.

- **didaktická funkcia** (napr. motivačná, na prvotné osvojovanie, upevňovacia, systemizačná, aplikačná, propedeutická, diagnostická),
- **myšlienkové operácie** (napr. na zapamätanie, porozumenie, aplikovanie, analyzovanie, vyhodnotenie, tvorbu),
- **prvky učiva** (napr. na jednotlivé príkazy, údajové typy, programovacie techniky a algoritmy, aplikačné oblasti),

spolu s možnosťou zobrazenia návodu na riešenie resp. s riešením danej úlohy.

Veľmi dôležitým činiteľom pri riešení učebných úloh je prostredie, v ktorom sa úloha rieši. Pri použití softvérovej aplikácie je možné riešiť interaktívne úlohy, umožňujúce okamžitú spätnú väzbu pre žiaka. Príkladom interaktívnych úloh umožňujúcich bádať a objavovať žiakov sú úlohy na využitie metódy čiernej skrinky (http://di.ics.upjs.sk/vyucba/pomocne_materiary/cierna_skrinka/) a niektoré úlohy v súťaži Informatický bobor (<http://ibobor.sk/>). Viac o typoch úloh vhodných na interakciu nájdeme v publikácii (Tomcsányiová, 2009).

Vzhľadom na osobnosť učiteľa, jeho tvorivý prístup a náročnosť pokrytia viacerých dimenzií a typov úloh by mal byť učiteľ informatiky schopný **preformulovať zadania úloh**, aby pokryli očakávané ciele vyučovania.

Prostredníctvom nasledujúcich aktivít si precvičíme svoje zručnosti v oblasti tvorby úloh rôznych typov (motivačných, prípravných, systemizačných, aplikačných), preformulovania zadání úloh a tvorby systému úloh.

Aktivita 1	Vytvorte zadanie a návrh riešenia motivačnej úlohy (príbeh, diskusia) k téme šifrovanie.
Aktivita 2	Vytvorte zadanie motivačnej úlohy - počítačovej hry k vybranej téme z tematickej oblasti Postupy, riešenie problémov, algoritmické myslenie. K zadaniu úlohy uveďte metodický komentár.
Aktivita 3	Preformulujte zadanie nasledovnej úlohy: „Vypočítanie NSD dvoch čísel.“ - tak, aby bolo zrozumiteľné a jednoznačné, - tak, aby cieľom úlohy bolo nájdenie a opravenie logickej chyby v zadanom programe, - tak, aby cieľom úlohy bolo porovnanie dvoch riešení toho istého problému, - tak, aby sa úloha stala náročnejšou (napr. zmenou vstupných, výstupných podmienok, obmedzením a pod).
Aktivita 4	K uvedenému zadaniu úlohy vytvorte dve pomocné (prípravné, návodné) úlohy. „Vytvorte program, ktorý vykreslí vlajku zadanej krajiny zo zakladajúcich krajín EÚ.“
Aktivita 5	K nasledovným cieľom vytvorte zadanie systemizačnej úlohy: - určiť hodnoty v grafe a ich závislosti, - špecifikovať relevantné vstupné údaje, - špecifikovať relevantné výstupné údaje a vzťahy medzi vstupnými a výstupnými údajmi, - navrhnuť vhodnú reprezentáciu údajov a implementovať ju v tabulkovom kalkulátore (vytvoriť tabuľku, graf), - interpretovať výstupné údaje a porovnať dosiahnuté výsledky, stanoviť trendy vývoja.

Aktivita 6	Preformulujte nasledovné zadanie úlohy tak, aby úloha plnila aplikačnú funkciu. Vytvorte program, ktorý na vstupe dostane dve postupnosti čísel. Výstupom programu je skalárny súčin týchto dvoch postupností. Napríklad: Vstup: 1 2 3 1 0 2 Výstup: 7 ($1 \times 1 + 2 \times 0 + 3 \times 2$)
Aktivita 7	K vybranej náročnejšej programátorskej úlohe (problému, projektu) navrhnete etapy jej riešenia a k nim odpovedajúce zadania úloh.
Aktivita 8	Ktoré myšlienkové operácie (v zmysle niektorej z taxonómií) overuje nasledovná úloha: Na konci brigády pridali študentom k ich platom 10% odmenu. Vytvorte program, ktorému zadáme pôvodné výšky plátov a ktorý vypočíta pre každého študenta upravenú výšku platu a určí počet študentov, ktorých výška upraveného platu presiahla 500 €.

Čo sme sa naučili

vymedziť pojem učivo,

rozlíšiť dimenzie učiva a prvky učiva, uviesť ich konkrétne príklady,

vymedziť pojem úloha a pojem komponenty úlohy,

opísať didaktické funkcie úloh,

preformulovať úlohu tak, aby rozvíjala vyššie myšlienkové operácie,

vytvoriť motivačnú úlohu,

vytvoriť aplikačnú úlohu,

vytvoriť systemizačnú úlohu,

k danej úlohe vytvoriť pomocné úlohy,

vytvoriť postupnosť gradovaných úloh k danej téme školskej informatiky.

Kapitola 2: Hodnotenie žiakov v predmete informatika

Každá cieľavedomá ľudská činnosť, t.j. aj vzdelávací proces pozostáva z troch fáz - plánovanie, realizácia, kontrola. **Kontrola vzdelávacieho procesu** je proces, ktorým sa zisťujú a posudzujú jeho výsledky. Pozostáva z dvoch neoddeliteľných častí - **zisťovanie** a **posúdenie** výsledkov vzdelávacieho procesu. Zisťovanie výsledkov

realizujeme pomocou preverovania, skúšania. Jeho účelom je meranie výkonu žiaka, zistenie toho čo vie, čo nevie, v čom sa zlepšil. Ide o zisťovanie stupňa dosiahnutia cieľov vzdelávacieho procesu. Posúdenie výsledkov vzdelávacieho procesu nazývame **hodnotením**. Jeho podstatou je porovnanie výsledkov žiaka s určenými požiadavkami (vzory, normy, štandardy). Hodnotenie vyjadrené v koncentrovanej podobe (známka, stupeň, číslo) sa nazýva **klasifikácia**. (Turek, 2008)

Hodnotenie má veľký význam nielen pre učiteľa, ale aj pre žiaka. Pre učiteľa je spätnou väzbou o jeho práci, umožňuje mu posúdiť pripravenosť žiaka pre ďalšie učenie sa. Pre žiaka je hodnotenie spätnou väzbou o jeho výkone a prospechu, motivuje ho k lepšej organizácii svojej práce, k dosiahnutiu lepších výkonov. Prehľad funkcií hodnotenia je uvedený na okraji strany.

Existuje viacero typov hodnotení žiakov:

- **podľa spôsobu vyjadrovania (zdroja informácii):**
 - ústne - žiaci sa pri skúšaní vyjadrujú ústne,
 - písomné - žiaci sa pri skúšaní vyjadrujú písomne (napr. didaktický test, referát, správa),
 - praktické - žiaci pri skúšaní vykonávajú praktickú činnosť (napr. aktivita na počítači),
- **podľa počtu žiakov:**
 - individuálne (jeden žiak),
 - skupinové (traja až piati žiaci),
 - frontálne, hromadné (učiteľ skúša naraz všetkých žiakov triedy),
- **podľa časového zaradenia a funkcie:**
 - prijímacie skúšky (napr. pri výberovom konaní na SŠ),
 - priebežné skúšanie (napr. z učiva jednej vyučovacej hodiny),
 - súhrnné skúšanie (napr. z učiva tematického celku),
 - záverečné skúšky (napr. maturitné skúšky),
 - opravné skúšky.

Existuje viacero spôsobov získavania podkladov na hodnotenie a klasifikáciu:

- sústavným diagnostickým pozorovaním žiaka,
- sledovaním výkonu žiaka a jeho pripravenosti na vyučovanie,
- rôznymi druhmi skúšania (písomné, ústne, praktické) a didaktickými testami,
- analýzou výsledkov rôznych činností žiaka,
- konzultáciou s ostatnými učiteľmi,
- rozhovormi so žiakom a zástupcami žiaka.

Hodnotenie v informatike má svoje špecifiká. Z vlastných skúsenosti odporúčame:

- klásť dôraz na pojmy a princípy, nie na fakty a konkrétne postupy, napr. neskúšať žiakov z horúcich klávesov, historických faktov, značiek produktov,
- zamerať sa na princíp fungovania triedy nástrojov s podobným zameraním, nie na ovládanie konkrétneho softvéru, napr. neskúšame žiakov z ovládania MS Wordu ani ovládania textového procesora, ale z problematiky spracovania textov využitím digitálnych technológií,
- vopred oznámiť žiakom kritériá hodnotenia žiackych produktov,
- komplexnejšie žiacke produkty (napr. prezentácie, webové stránky, textové dokumenty, programy) hodnotiť z viacerých aspektov viacerými odborníkmi - napr. prezentáciu žiaka z fyzikálnej problematiky hodnotí slovenčinár (gramatika, štylistika, rétorika), fyzikár (odbornosť a správnosť), výtvarník (farebná a priestorová kompozícia), učiteľ náuky o spoločnosti (autorský zákon, citovanie informačných zdrojov, etiketa), učiteľ informatiky (vhodnosť výberu nástroja a efektívnosť postupu spracovania, mieru prístupnosti a použiteľnosti),
- ak pri hodnotení žiackych projektov nemáme k dispozícii ochotných a kompetentných učiteľov, ktorí by ohodnotili žiacke projekty z iných aspektov, v hodnotení učiteľa informatiky by sa mali vyskytnúť aj tieto

Funkcie hodnotenia:

- **kontrolná** (určenie miery vedomostí, zručností a návykov žiakov, rozpoznanie ich schopností, postojov a hodnotovej orientácie, zistenie predností a nedostatkov),
- **prognostická** (určenie perspektívy žiakov, možnosti ich ďalšieho vývoja, čím sa môžu stať, s akými predpokladmi),
- **motivačná** (udržiavanie a zvyšovanie študijnej aktivity žiakov),
- **spoločenská a profesijná** (pripravuje žiaka na zaradenie sa do spoločnosti),
- **výchovná** (formovanie pozitívnych vlastností a postojov - česťnosť, samostatnosť, pracovitnosť a odstraňovanie negatívnych vlastností),
- **informačná** (dokumentácia výsledkov vzdelávacieho procesu a ich oznámenie rodičom žiaka, budúcim zamestnávateľom),
- **rozvíjajúca** (rozvoj schopnosti sebakontroly a sebahodnotenia žiakov),
- **spätnoväzbová** (získanie informácií o úspešnosti vzdelávacieho procesu žiakmi a učiteľom na reguláciu ďalšieho postupu),
- **certifikačná** (zisťovanie, či si žiak osvojil požadované poznatky a zručnosti, či má požadované schopnosti a postoje, aby mu mohol byť vydaný diplom, certifikát).

(Bajtoš, 2003)

- aspekty s menšou váhou oproti informatickým aspektom,
- súčasťou sumatívneho hodnotenia žiaka je hodnotenie metódou portfólia (súbor prác žiaka za určité obdobie, ktorý poskytuje informácie o skúsenostiach a pracovných výsledkoch žiaka, vrátane komentárov učiteľa),
- pri skúšaní nevyužívať len testy typu ECDL, lebo overujú len dosiahnutie nižších kognívnych cieľov (zapamätanie, porozumenie, aplikáciu),
- nepodceňovať význam slovného hodnotenia a využívať ho na posilnenie spätnej väzby, motivácie, vzťahu k predmetu, škole, učiteľovi, podrobnejší popis chýb (diagnóza) a tiež komentár a odporúčania (terapia) - tento spôsob bežne využívame aj pri hodnotení zadaní účastníkov v projekte ĎVUI v prostredí LMS Moodle.

Hodnotenie portfólia

Hodnotenie žiaka v škole je zamerané predovšetkým na kognitívnu časť jeho osobnosti. Cítíme však, že takýto zúžený pohľad nie je celkom správny. Postojové kvality žiaka a praktické zručnosti sú tiež neoddeliteľnou súčasťou jeho osobnosti. Jedným zo spôsobov, ako brať aj tieto kvality žiaka do úvahy je **autentické hodnotenie**. Pri autentickom hodnotení nie je podkladom pre hodnotenie žiaka „umelá“ školská úloha, ale jeho výkon, ktorý má zmysel aj mimo školy.

Pri autentickom hodnotení nebudeme hodnotiť spracovanie textovej informácie v textovom procesore, ale plagát informujúci o Benefičnom koncerte školy. Nebudeme hodnotiť výpočet v tabuľkovom kalkulátore ale spracovanie rozpočtu školského výletu a pod.

Jedným z prostriedkov autentického hodnotenia je tzv. hodnotenie portfólia. Pod pojmom portfólio rozumieme súhrn žiackych prác a ďalších podkladov zhromaždených v nejakej kompaktnej podobe (napr. v škatuli, šanóne), ktoré nám poskytujú informácie o jeho skúsenostiach a pracovných výsledkoch. Súčasťou portfólia môžu byť písomky, školské práce, referáty, prezentácia, postery, výkresy, grafické návrhy, programy a pod. Okrem produktov žiaka, môžu byť súčasťou portfólia aj hodnotenia týchto produktov učiteľmi, referencie zákazníkov a pod.

Prostredníctvom portfólia sa môže učiteľ pozerat' na žiaka ako na komplexnú osobnosť s rôznymi schopnosťami, záujmami, potrebami, prianiami, úspechmi, ale aj neúspechmi. Portfólio dokumentuje žiakov profesijný rast, jeho pokroky, naznačuje budúce smerovanie. Portfólio môže pomôcť učiteľovi odpovedať na otázky „V čom je tento žiak dobrý?“, „Ako mu pomôcť k lepším výkonom“, „Prečo má tento žiak problémy?“. Portfólio neraz ukáže, že aj zle prosperujúci žiak môže mimo obzor učiteľa dosahovať nezanedbateľné úspechy.

Na webovej stránke http://www.gjar-po.sk/studenti/profesijne_s_tranky.htm nájdete zoznam portfólií, profesijných stránok žiakov Gymnázia J. A. Raymana v Prešove. Žiaci si svoje portfólia začínajú budovať koncom prvého ročníka a pokračujú v jeho udržiavaní počas celého štúdia. Niektorí v tejto činnosti pokračujú aj počas štúdia na vysokej škole.

Portfólio a jeho udržiavanie má význam aj pre žiaka samotného. Učí ho k samostatnosti a zodpovednosti za svoje počínanie.

Zamyslime sa!

Ivan, ktorý reprezentoval školu v olympiáde z matematiky mal rôzne úľavy zo štúdia. Sústredenia, študijné voľná a pod. Jeho prospech v ostatných predmetoch zaostával. Učitelia to však tolerovali a občas prižmúрили oko.

Kamil bol jeho spolužiak. Nepatril medzi najlepších žiakov a učiteľia ho vnímali ako priemerného. Nedostávali sa mu žiadne úľavy ani zvýhodnenia. Vo svojom voľnom čase, mimo školy, sa však vášnivo venoval astronómii. Vedel navrhnúť systém šošoviek a zostrojiť ďalekohľad. Poznal zákonitosti vesmíru a vedel o nich zaujímavo rozprávať. V škole ho však vnímali ako priemerného žiaka s malým záujmom o školu.

V čom sa líšil Kamil od Ivana?

Zaslúžil by si Kamil rovnaké zvýhodnenia ako Ivan? Prečo?

Ak by Kamil mal svoje portfólio, mohli by mu učitelia pomôcť dostať sa z priemernosti? Ako?

Hodnotenie portfólia predpokladá splnenie týchto podmienok (Slavík, 1999):

- Sú vopred vymedzené ciele a kritériá hodnotenia.
- Na základe cieľov a kritérií je určené, čo má byť súčasťou portfólia, kto má portfólio udržiavať a ako často sa má portfólio „aktualizovať“.
- Kritériá hodnotenia sú použité ku komplexnému posúdeniu žiakovho tvorivého chovania.

Vzhľadom k tomu, že portfólio uchováva prejavy žiackej tvorivosti je pre učiteľa jeho hodnotenie náročná činnosť. Vyžaduje si dostatok pedagogickej tvorivosti a citu pre dôležité momenty v živote a práci žiaka. Pri jeho hodnotení by sme nemali zabúdať na fakt, že proces tvorby je považovaný za pedagogicky cennejší než samotné výsledky uložené v portfóliu. Pri hodnotení výsledkov berieme do úvahy celú činnosť ktorá sa za nimi skrýva a neobmedzujeme sa len na to, či sa výsledok žiakovi podaril alebo nie.

Pri verejných portfóliách žiakov bud'me opatrní so zverejňovaním hodnotení žiackych produktov. Toto hodnotenie je osobným údajom žiaka.

Školská informatika poskytuje žiakom nástroje a možnosti ako spracovať a udržiavať si svoje portfólio prostriedkami digitálnych technológií v elektronickej podobe. Či už ako systém súborov na disku počítača alebo prostredníctvom svojej profesijnej stránky.

Didaktický test

Pod pojmom **didaktický test** budeme rozumieť nástroj na objektívne meranie úrovne zvládnutia učiva skupinou ľudí, v našom prípade žiakov. Didaktický test je na rozdiel od ostatných typov skúšok (ústna odpoveď, písomná odpoveď na učiteľom čítané otázky, cvičenia atď.) navrhovaný, overovaný, hodnotený a interpretovaný podľa vopred určených pravidiel. (Lavický, 2005)

Didaktický test je teda nástrojom systematického merania výsledkov výučby. Zaujímá nás meranie tých javov, ktoré tvoria oblasť cieľov vzdelávania, t.j. tých, ktoré súvisia s výsledkami vzdelávania. (Rosa, 2007) Vychádzame z toho, že existuje viacero úrovní osvojenia si poznatkov, ktoré je možné usporiadať do systému. Týmto systémom je taxonómia vzdelávacích cieľov.

Konkrétny didaktický test môžeme charakterizovať jeho vlastnosťami. Medzi základné vlastnosti testov patria: validita, reliabilita, citlivosť, objektívnosť a použiteľnosť.

Validita (vysoká, stredná, nízka) je miera vhodnosti testu na konkrétny účel. Vedomostný test je validný, ak zisťuje skutočné vedomosti z oblasti, ktorú preveruje (Rosa, 2007). V praxi sa najčastejšie stretávame a pre nás bude najdôležitejšia obsahová validita. Obsahová validita určuje, do akej miery je obsah testu v zhode s tým, ako sa učivo prezentuje v štandardoch (Štátny vzdelávací program, Školský vzdelávací program, maturitný štandard), v učebniciach, odbornej literatúre, v profile absolventa a pod. Obsah úloh testu by mal predstavovať reprezentatívnu vzorku učiva.

Reliabilita, spoľahlivosť testu udáva, do akej miery sa podarilo potlačiť vplyv náhody pri testovaní. Udáva, na koľko rovnaké výsledky by účastníci dosiahli pri rovnakom alebo veľmi podobnom teste. Čím je reliabilita vyššia, tým menší vplyv má náhoda na výsledok. Koeficient korelácie nad 0,85 je postačujúci na prijatie záverov z jedného testu. Koeficient nad 0,65 umožňuje brať výsledok testu ako jeden z podkladov, na základe ktorých sa budeme rozhodovať. Koeficient pod 0,65 znamená, že test nie je spoľahlivý a jeho výsledky nemožno brať pri rozhodovaní do úvahy.

Citlivosť testu charakterizuje schopnosť testu rozlíšiť žiakov s dobrými vedomosťami od žiakov so slabými vedomosťami. Test, v ktorom všetci žiaci obstoja

veľmi dobre má nízku citlivosť. Ak test vykazuje malú citlivosť, vyradíme úlohy, ktoré vyriešilo správne najviac žiakov a nahradíme ich náročnejšími úlohami.

Objektívnosť testu rozdeľujeme na formálnu a obsahovú. Pri formálnej objektívnosti musíme zabezpečiť, aby výsledok testu neovplyvňovali žiadne vonkajšie podnety (vhodné a rovnaké podmienky pre všetkých: teplo, svetlo, čas, pomôcky...). Obsahovú objektívnosť dosiahneme tým, že vylúčime všetky pochybnosti o správnosti riešenia. Rôzni hodnotitelia tak budú vyhodnocovať rovnaké odpovede rovnako. Test by nemal byť výhodný pre časť testovaných žiakov (napr. ak test postavíme na problémy spracovania výsledkov futbalových zápasov v tabuľkovom kalkulatore, chlapci budú pravdepodobne vo výhode).

Použitelnosť testu má zabezpečiť ľahké opravovanie, vyhodnotenie a vyvodenie záverov z výsledkov testu.

Tvorba didaktického testu

Tvorba didaktického testu pozostáva z niekoľkých krokov. Zároveň by sme mali dodržať niekoľko pravidiel.

Testy podľa účelu môžu byť napr.:

- vstupné (prijímačky),
- výstupné (maturita),
- diferenčné (rozdelenie žiakov do skupín)
- priebežné (spätná väzba)
- overujúci (overenie dosiahnutia vzdelávacích cieľov)

Parametre testu:

- forma (ústna, písomná, elektronická ...)
- počet tém (monotematický, polytematický)
- počet foriem/skupín (jedna, dve, viac)

1. Definujeme účel testu (overujúci, diferenčný), parametre testu (koľko tém, formát testu, počet foriem ...).
2. Zostavíme špecifikačnú tabuľku. S jej pomocou dosiahneme, aby test mal požadovanú obsahovú validitu a vyvážené zastúpenie oblastí učiva a taxonómie úloh, náročnosť myšlienkových operácií.
 - Tému (resp. témy) testu rozdelíme na menšie časti. Každý časti priradíme určitú váhu (napr. podľa počtu hodín, ktoré sme výučbe danej časti venovali, rozsah učiva v učebnici a pod.)
 - Určíme počet úloh testu. Pre dosiahnutie požadovanej úrovne reliability by test mal obsahovať minimálne 10 úloh. Maximálny počet je úloh limitovaný časom na riešenie testu (napr. organizáciou vyučovania). Samotný test by nemal trvať viac ako 40 min čistého času. Počet úloh závisí aj od druhu otázok použitých v teste.
 - Pri každej časti učiva určíme, akú úroveň osvojenia poznatkov by mali úlohy testovať. Z dôvodu kvality vedomostí by sme mali preferovať vyššie úrovne osvojenia.

	čas výučby (hod)	%	úlohy	%	úroveň kognitívnych cieľov podľa RBT					
					1	2	3	4	5	6
typ pole, index, prvok	1	25	123	30		ooo				
načítanie, výpis prvkov, manipulácia s prvkami, zmena			45	20				oo		
nájdenie prvku s danou vlastnosťou	1	25	9					o		
algoritmy na poliach	2	50	6 7 8 9 10	50			o	o o o	o	

Obrázok 3: Špecifikačná tabuľka pre didaktický test na tému pole

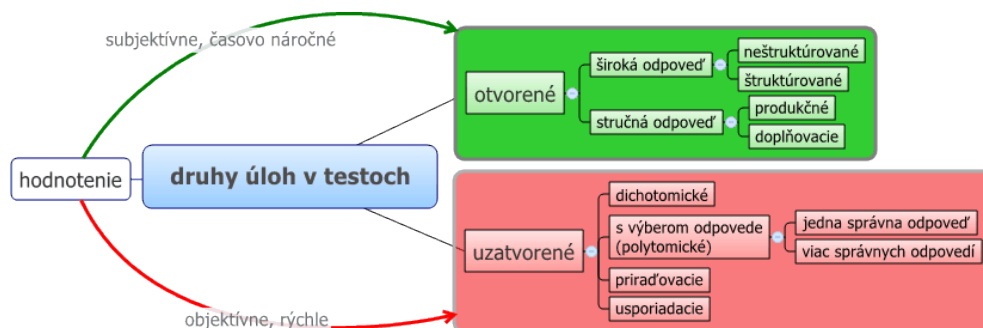
3. Vytvoríme testovacie úlohy podľa predpokladov špecifikačnej tabuľky. Ťažisko konštrukcie testu je v pretransformovaní definovaných cieľov výučby do úloh.
4. Jednotlivým úlohám pridelíme bodové skóre (skórovanie). Úlohy v teste

nemusia byť rovnocenné. Úlohy overujúce vyššie úrovne porozumenia ohodnotíme vyššie. Ak je počet bodov menší ako 20, používa sa zložené skórovanie. Jednotlivým úlohám môžeme prideliť aj viac ako jeden bod.

5. Test formálne upravíme tak, aby bol pre žiaka prehľadný, s dostatočným priestorom na pomocné úkony. Zvolíme vhodný formát tlače.

Typy úloh

Zrejme neexistuje všeobecne uznávaná a zároveň presná klasifikácia typov úloh používaných v testoch. V jednom teste môžeme použiť rôzne typy úloh. Test zostavený len z jedného typu úloh sa považuje za „zlý“. V testoch sa môžeme stretnúť s nasledujúcimi typmi úloh:



Obrázok 4: Typy úloh v didaktickom teste

Nasledujúce aktivity sú zamerané na prediskutovanie spôsobov hodnotenia žiakov, rôznych aspektov hodnotenia a klasifikácie žiakov. Ďalej účastníci vzdelávania budú vytvárať hodnotiace kritériá pre projekty, a aj didaktický test k vybranej téme informatiky. Prediskutujú sa tiež špecifiká hodnotenia programovania, využívanie ďalších spôsobov hodnotenia žiakov - slovné hodnotenie, samohodnotenie, súťaže a hry, spôsoby evalvácie vlastnej výučby učiteľa.

Aktivita 9	Uved'te, aké formy hodnotenia používate vo výučbe informatiky, prípadne iného predmetu. Máte jednotný systém hodnotenia? Ako často hodnotíte žiakov? Koľko známok majú vaši žiaci z informatiky za jeden polrok?
Aktivita 10	Ku ktorej koncepcii vzdelávania máte najbližšie? Čo hodnotíte? Len kvalitu žiackych výkonov, alebo aj ich úsilie, či originalitu, estetickú úpravu, či ďalšie nonkognitívne aspekty? Ktoré z týchto aspektov by sa mali odraziť v klasifikačnej známke?
Aktivita 11	Navrhните hodnotenia žiakov pri téme Hromadná korešpondencia. Popíšte, akým spôsobom ste ho zostavovali (výber spôsobu hodnotenia, výber úloh, formulovanie úloh).
Aktivita 12	Zaznamenávate si chyby, ktorých sa dopustili žiaci počas výkladu, precvičovania učiva, či v previerkach? Pre vybranú tému školskej informatiky uved'te zoznam typických žiackych chýb.

Vždy oznamujete žiakom dopredu termín previerky?

Kedy pripravujete úlohy na previerku, deň vopred, či na začiatku výučby danej témy?

Ako zvyknete opravovať písomky, po úlohách alebo po žiakoch, prečo?

Žiak riešil úlohu inak než autor úlohy, pričom sa dopustil chyby. Ako by ste postupovali pri udeľovaní počtu bodov? Vážite si viac, keď má žiak super - nápad, ale ho nedotiahne do konca, alebo keď vyrieši správne očakávaným postupom?

V prípade, že máte posúdiť prácu skupiny, či tímu, dáte každému rovnakú známku, alebo to zariadite tak, aby ste to vedeli aj individuálne ohodnotiť?

Okrem momentálneho výkonu žiaka v danej previerke sledujete aj jeho výsledok vzhľadom na predchádzajúce výsledky?

Ako pridelujete body za neúplné riešenie úlohy?

Aktivita 13	Podľa akých kritérií hodnotíte žiacke projekty? Aké sú typické nedostatky v žiackych projektoch? Akým spôsobom chválite žiakov za ich výborné výsledky?
Aktivita 14	V supermarkete predávajú ananásy (1 €) a kivi (0,3 €). Za prvé tri kusy tovaru rovnakého druhu zaplatíme plnú cenu, na všetky ďalšie kusy toho istého druhu sa uplatní zľava 20 %. Vytvorte program, ktorý pre zadaný počet ananásov a kivi vypíše celkovú cenu nákupu. Napište riešenie tejto úlohy. Opravte chyby a obodujte (oznámkujte) uvedené žiacke riešenie: <pre>viem cena :a :k urobTu "cenaA 0 ak :a < 3 [urobTu "cenaA :a * 1] ak :a > 4 [urobTu "cenaA 3 * 1 + (:a-3) * 0.2 * 1] urobTu "cenaK 0 ak :k < 3 [urobTu "cenaK :c * 0.3] ak :k > 4 [urobTu "cenaK 3 * 0.3 + (:k-3) * 0.2 * 0.3] pis :cenaA + :cenaK koniec</pre> Uved'te javy, za ktoré budete v tejto úlohe udeľovať body.
Aktivita 15	Uved'te príklad slovného hodnotenia na riešení konkrétnej úlohy z vybranej oblasti školskej informatiky.
Aktivita 16	Aký máte názor na to, aby sa žiaci zapájali do vlastného hodnotenia sa, prípadne hodnotenia svojich spolužiakov? Máte osobné skúsenosti so samohodnotením, resp. vzájomným hodnotením sa žiakov? Ak áno, uved'te ich.
Aktivita 17	Prediskutujte problematiku využívania súťaží a hier, prípadne iných alternatívnych prístupov hodnotenia žiakov vo výučbe informatiky.
Aktivita 18	Akými spôsobmi robíte evaluáciu vlastnej výučby? Dávate žiakom dotazník na konci školského roka o výučbe informatiky? Ak áno, ktoré otázky zvyčajne kladiete?

Čo sme sa naučili

rozlišovať pojmy pedagogická kontrola, klasifikácia, hodnotenie,
vymenovať a okomentovať vlastnými slovami jednotlivé funkcie hodnotenia,
špecifikovať ciele vyučovania vo vzťahu k hodnoteniu,

vymenovať a okomentovať vlastnými slovami typy hodnotení a spôsoby získavania podkladov na hodnotenie a klasifikáciu,

vymedziť pojem didaktický test, jeho základné charakteristiky, klasifikácie didaktických testov, formy testovacích položiek didaktických testov,

popísať životný cyklus didaktického testu a zažiť ho pri riešení konkrétnej aktivity,

popísať a prediskutovať jednotlivé príklady a špecifiká vybraných spôsobov hodnotenia (didaktické testy, projekty, dotazníky, posudzovacie škály, slovné hodnotenie, samohodnotenie).

Kapitola 3: Tvorba didaktického projektu

S pojmom didaktický projekt sa môžeme stretnúť v didaktickej literatúre, na edukačných webových stránkach. Didaktický projekt môže mať viacero významov:

- softvérová edukačná pomôcka,
- projektová aktivita študentov,
- forma prípravy učiteľa.

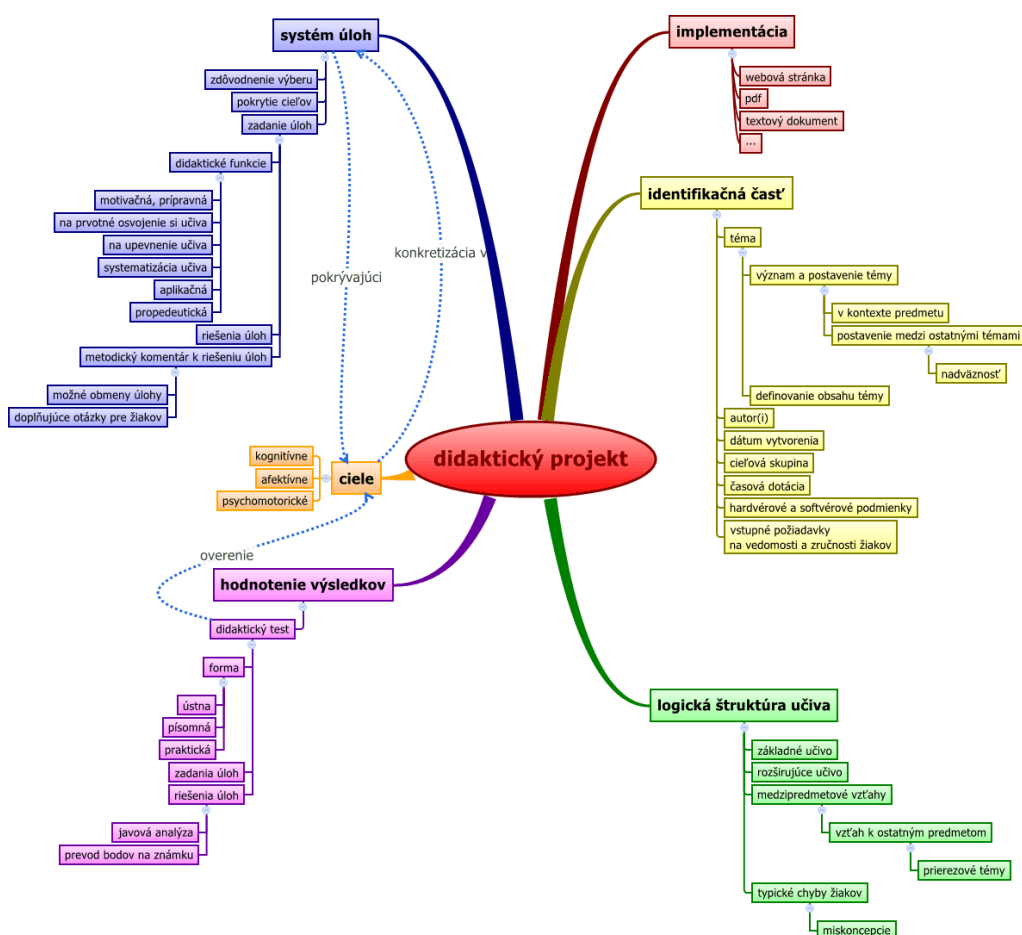
V tomto materiáli pod pojmom didaktický projekt budeme rozumieť výsledok prípravy učiteľa na vyučovanie konkrétnej témy. Chápeme ho ako plánovací prostriedok výučby. Okrem plánovacích kompetencií tvorbou didaktického projektu učiteľ rozvíja aj svoje odborné-predmetové, psychodidaktické, komunikačné, diagnostické, organizačné, poradenské, konzultatívne a sebareflexívne kompetencie.

Konkrétne príklady didaktických projektov (v rôznych významoch) si môžete prezrieť na webovej adrese:
http://www.npu.cz/pp/arc_hivy/detskazona/didakticke-projekty/.

Didaktické projekty študentov PF UPJŠ:

Vektorová animácia v Macromedia Flash
http://s.ics.upjs.sk/~porembova/system_uloh2.html

Spracovanie dát pomocou tabuľkového kalkúlátora
<http://s.ics.upjs.sk/~szalayova/din/system.html>



Obrázok 5: Štruktúra didaktického projektu

Didaktický projekt má svoju vnútornú štruktúru. Súčasťou didaktického projektu je definovanie podmienok výučby, logická analýza témy učiva, definovanie vzdelávacích cieľov, systém gradovaných úloh s riešeniami a komentármi a didaktický test. Jednotlivé časti didaktického projektu znázorňuje nasledujúci obrázok.

Vytvorenie didaktického projektu je náročná činnosť. Pre lepšie pochopenie demonštrujeme vytvorenie didaktického projektu na konkrétnej téme: *Štruktúrovaný údajový typ pole*.

Proces tvorby didaktického projektu

Proces tvorby didaktického projektu pozostáva z nasledovných činností:

- identifikácia témy (podmienky výučby),
- analýza učiva
 - tvorba logickej štruktúry učiva (prvky učiva a vzťahy medzi nimi),
- stanovenie učebných cieľov,
- tvorba systému gradovaných úloh,
- tvorba didaktického testu.

Pre vytvorenie kvalitného didaktického projektu je potrebné uvedené body opakovane vykonávať. Pri každom prechode sa snažíme o kvalitatívne vylepšenie danej časti. Po vyskúšaní a overení didaktického projektu v praxi môžeme zistené skutočnosti zapracovať do projektu.

Identifikačná časť

Súčasťou identifikačnej časti je určenie cieľovej skupiny, hodinovej dotácie a podmienok (softvér, hardvér, ostatné podmienky) výučby.

Definujeme význam a postavenie vybranej témy k ostatným témam, ich vzájomné nadväznosti. Určíme vstupné požiadavky na vedomosti a zručnosti žiakov (Príloha 1).

Analýza učiva, logická štruktúra učiva

Preštudovaním štátneho a školského vzdelávacieho programu, časovo-tematického plánu predmetu informatika, maturitného štandardu, prístupných učebníc a príkladov vymedzíme základné a rozširujúce prvky učiva vybranej témy. Určíme význam a postavenie zvolenej témy v kontexte k ostatným témam školskej informatiky. V rámci rozvoja medzipredmetových vzťahov je vhodné uvažovať aj o využití danej témy v iných predmetoch, resp. v prierezových témach. Výsledkom tejto činnosti je logická štruktúra učiva v nadväznosti na ostatné témy (Príloha 2).

Súčasťou tejto etapy je aj určenie typických žiackych chýb. Ich identifikáciu môžeme vykonať na základe skúseností s výučbou tejto témy alebo ich môžeme odhadnúť z vlastných skúseností.

Aktivita 19

Analyzujte učivo vybranej témy školskej informatiky. Popíšte význam výučby témy v kontexte celého predmetu a jej postavenie medzi ostatnými témami. Určite jednotlivé prvky učiva a vzťahy medzi nimi. Uvažujte o význame a využití tejto témy v kontexte školského vzdelávacieho programu.

Stanovenie učebných cieľov

Ďalšou etapou je stanovenie učebných cieľov vo vyučovaní vybranej témy. Ciele definujeme hlavne v kognitívnej oblasti. Na zreteli by sme však mali mať aj ciele v afektívnej a psychomotorickej doméne. Pri stanovovaní cieľov odporúčame

vychádzať z niektorej zo známych taxonómií učebných cieľov. Ciele pre tému *Pole* nájdeme v Prílohe 3.

Aktivita 20

Definujte učebné ciele v kognitívnej, afektívnej a psychomotorickej oblasti pre vyučovanie vami vybranej témy. V kognitívnej oblasti sa zamerajte aj na vyššie vzdelávacie ciele.

Tvorba systému úloh

Pod systémom úloh rozumieme skupinu úloh, v ktorej jednotlivé úlohy plnia konkrétne didaktické funkcie v súlade so štruktúrou poznávacieho procesu, cieľmi a podmienkami vzdelávania (Šveda, 1992). Úlohy v systéme úloh by mali byť zaradené podľa narastajúcej náročnosti, mali by pokrývať stanovené vzdelávacie a výchovné ciele, rozvíjať poznávacie procesy na rôznych úrovniach včítane tvorivosti študentov. V systéme úloh rozlišujeme úlohy s týmito didaktickými funkciami:

- úlohy na motiváciu poznávacej činnosti žiakov,
- prípravné úlohy predchádzajúce zavedenie pojmu, princípu, algoritmu atď,
- úlohy na prvotné osvojenie učiva,
- úlohy na upevnenie učiva,
- úlohy na systemizáciu učiva,
- úlohy propedeutického charakteru,
- úlohy aplikačné (vzťah k životu, medzipredmetové vzťahy, záujmy žiakov, ich profesijnú orientáciu).

Na rozvíjanie vyšších úrovní poznávacích procesov (analyzovanie, hodnotenie, tvorenie) zaradíme do systému úloh rôzne typy formulácií úloh:

- vytvorte program, ktorý rieši daný problém,
- modifikujte (upravte, doplňte, vynechajte, premiestnite) jednotlivé časti uvedeného programu, aby riešil daný problém,
- analyzujte, čo robí daný program,
- nájdite a opravte chybu v danom programe,
- porovnajte riešenia uvedených programov, zdôvodnite, ktorý z nich je správny, efektívnejší atď.

Súčasťou systému úloh je komentár k výberu úloh, prehľad pokrytia didaktických funkcií a vzdelávacích cieľov jednotlivými úlohami. Ku každej úlohe uvádzame jej riešenie a metodický komentár (Príloha 4).

Metodický komentár obsahuje zdôvodnenie zaradenia danej úlohy a návod ako s úlohou pracovať, jej možné modifikácie a doplňujúce otázky pre žiakov.

Zadanie 21

Vytvorte systém gradovaných úloh z danej témy. Výber úloh by mal byť v súlade s definovanými cieľmi. Ku každej úlohe vypracujte vzorové riešenie a metodický komentár.

Tvorba didaktického testu

Poslednou etapou tvorby didaktického projektu je tvorba didaktického testu. Úlohy do didaktického testu musia byť vybrané tak, aby reprezentovali prebrané učivo, zastupovali nižšie aj vyššie poznávacie procesy. Na riešenie úloh testu určíme primeraný čas. Dôležitý je aj výber spôsobu testovania. Na základe autorských riešení testovacích úloh vypracujeme detailnú javovú analýzu a jednotlivým pozorovaným javom pridáme bodové skóre. Napokon stanovíme stupnicu prevodu bodového skóre na klasifikačné stupne (Príloha 5).

Didaktický test by mal spĺňať podmienky uvedené v časti *Didaktický test*.

Zadanie 22

Navrhните a vytvorte didaktický test. Navrhните javovú analýzu a bodové ohodnotenie javov.

Zadanie 23

Využitím výsledkov predchádzajúcich aktivít vytvorte didaktický projekt.

Pre lepšiu názornosť a pochopenie didaktického projektu si na webovej stránke http://di.ics.upjs.sk/vyucba/pomocne_materialy/dp/ môžete pozrieť kompletný didaktický projekt na tému *Štruktúrovaný údajový typ pole* v kompaktnej podobe.

Čo sme sa naučili

vymedziť pojem didaktický projekt,
popísať význam a použitie didaktického projektu v príprave učiteľa,
urobiť didaktickú analýzu vybranej témy školskej informatiky,
navrhnuť a vytvoriť didaktický projekt.

Čo sme sa naučili v tomto module

Zhrnutie

vymedziť pojem učivo,
rozlíšiť dimenzie učiva a prvky učiva, uviesť ich konkrétne príklady,
urobiť didaktickú analýzu vybranej témy školskej,
vymedziť pojem úloha a jej komponenty,
opísať didaktické funkcie úloh,
preformulovať úlohu tak, aby rozvíjala vyššie myšlienkové operácie,
vytvoriť motivačnú úlohu,
vytvoriť aplikačnú úlohu,
vytvoriť systemizačnú úlohu,
k danej úlohe vytvoriť pomocné úlohy,
vytvoriť postupnosť gradovaných úloh k danej téme školskej informatiky.
rozlišovať pojmy pedagogická kontrola, klasifikácia, hodnotenie,
vymenovať a okomentovať vlastnými slovami jednotlivé funkcie hodnotenia,
špecifikovať ciele vyučovania vo vzťahu k hodnoteniu,
vymenovať a okomentovať vlastnými slovami typy hodnotení a spôsoby získavania podkladov na hodnotenie a klasifikáciu,
vymedziť pojem didaktický test, jeho základné charakteristiky, klasifikácie

didaktických testov, formy testovacích položiek didaktických testov,

popísať životný cyklus didaktického testu a zažiť ho na riešení konkrétnej aktivity,

popísať a prediskutovať jednotlivé príklady a špecifiká vybraných spôsobov hodnotenia (didaktické testy, projekty, dotazníky, posudzovacie škály, slovné hodnotenie, samohodnotenie),

vymedziť pojem didaktický projekt,

popísať význam a použitie didaktického projektu v príprave učiteľa,

navrhnuť a vytvoriť didaktický projekt.

Preverenie výstupných vedomostí

Účastník vzdelávania preukáže požadované vedomosti a zručnosti prostredníctvom riešenia priebežných aktivít (zameraných na tvorbu úloh s rôznymi didaktickými funkciami, preformulovanie úloh, aby rozvíjali vyššie myšlienkové operácie, tvorbu didaktických testov) a hlavne vypracovaním záverečného zadania - didaktického projektu pre vybranú tému školskej informatiky.

Literatúra a použité zdroje

- [1] ANDERSON, L., A., KRATWOHL, D., R. (2001) *A Taxonomy of Learning, Teaching and Assessing: Revision of Bloom's Taxonomy of Educational Objectives*. New York : Longman, 2001
- [2] BAJTOŠ, J. (2003) *Teória a prax didaktiky*. 1. vyd. Žilina : Žilinská univerzita, 2003. 384 s. ISBN 80-8070-130-X
- [3] GAVORA, P. (1999) *Akí sú moji žiaci? Pedagogická diagnostika žiaka*. Bratislava : Práca, spol. s r. o. 1999. ISBN 80-7094-335-1
- [4] GUNIŠ, J., ŠNAJDER, L. (2009) *Programátorská súťaž PALMA junior*. Dostupné na internete: <<http://di.upjs.sk/palmaj/>>
- [5] GUNIŠ, J., ŠNAJDER, L., GUNIŠOVÁ, V., SZABOÓVÁ, Z., HRICOVÁ, A. (2007) PALMA junior - programming competition in Imagine. In: *Proceeding of the 11th European Logo Conference EuroLogo 2007 Bratislava*. Knižničné a edičné centrum FMFI UK, Bratislava 2007. pp. 54. ISBN 978-80-89186-20-4
- [6] HOLOUŠOVÁ, D. (1983) *Teorie učebních úloh*. In: *Studijní text pro přípravu učitele pedagogiky na nové pojetí výchovně vzdělávací práce na SPgŠ*. Praha : UÚVPP, 983
- [7] KALHOUS, Z., OBST, O. a kol. (2002) *Školní didaktika*. 1. vyd. Praha : Portál, 2002. 448 s. ISBN 80-7178-253-X
- [8] KOPKA, J. (2004) *Výzkumný přístup při výuce matematiky*. Ústí nad Labem : UJEP v Ústí nad Labem. 2004. ISBN 978-80-7044-604-8
- [9] LAVICKÝ, T. (2005) *Tvorba a využívanie školských testov*. Prešov : MPC, 2005
- [10] LERNER, I., J. (1986) *Didaktické základy metod výuky*. Praha : SPN, 1986
- [11] PASCH, M. et al. (1998) *Od vzdělávacího programu k vyučovací hodině*. Praha : Portál, 1998
- [12] ROSA, V. (2007) *Metodika tvorby didaktických testov - študijný text pre učiteľov*. Bratislava : Štátny pedagogický ústav. 2007. ISBN 978-80-89225-32-3
- [13] SLAVÍK, J. (1999) *Hodnocení v současné škole - Východiska a nové metody pro praxi*. Praha : Portál. 1999. ISBN 80-7178-262-9
- [14] ŠNAJDER, L., GUNIŠ, J., GUNIŠOVÁ, V. (2008) Methodology design of algorithm development teaching based on content analysis of pupils' solutions. In: *The 3rd International Conference - ISSEP 2008 Informatics in Secondary Schools - Evolution and Perspective July 1 - 4, 2008*, Torun, Poland. Faculty of Mathematics and Computer Science, Nicolaus Copernicus University, Toruń, Poland 2008. p. 20 - 29. ISBN 978-83-60425-31-2
- [15] ŠNAJDER, L., GUNIŠ, J. (2007) Didaktický projekt ako prostriedok rozvíjania plánovacích kompetencií učiteľa informatiky. In: *Zborník konferencie DidInfo*

- 2007 Banská Bystrica [CD ROM]. Univerzita Mateja Bela, Fakulta prírodných vied, Banská Bystrica. 2007. ISBN 978-80-8083-367-1
- [16]ŠVEC, V., FILOVÁ, H., ŠIMONÍK, O. (2004) *Praktikum didaktických dovedností*. Brno : Masarykova univerzita v Brně, Pedagogická fakulta, 2004. 90 s. ISBN 80-210-2698-7
- [17]ŠVEDA, D. (1992) *Tvorba systémov úloh v matematike*. Prešov : MC v Prešove, 1992. s. 43, ISBN 80-85410-36-2
- [18]TOMCSÁNYIOVÁ, M. (2009) Interaktívne úlohy v súťaži Informatický bobor. In: *Zborník konferencie DidInfo 2009 Banská Bystrica. Univerzita Mateja Bela, Fakulta prírodných vied, Banská Bystrica*. 2009. ISBN 978-80-8083-720-4.
- [19]TUREK, I. (2008) *Didaktika*. Bratislava : Iura Edition, 2008. ISBN 978-80-8078-198-9
- [20]VANÍČEK, J. (2008) O projektové výuce ve školní informatice. In: *MIF 32, XVII. ročník, Prešov 2008*. s. 15. ISSN 1335-7794
- [21]VOMÁČKOVÁ, H. (1991) *Didaktika výpočetní techniky pro střední školy s ekonomickými studijními obory*. Praha : SPN, 1991. ISBN 99-00-00400-X
- [22]ZELINA, M. (1990) *Tvorivost' v matematike*. Bratislava : Krajský pedagogický ústav v Bratislave, 1990. ISBN 80-85185-34-2

Príloha 1: Identifikačná časť didaktického projektu

Téma: Štruktúrovaný údajový typ pole

Cieľová skupina: žiaci 2. ročníka gymnázia zameranie 7902 5 05 - informatika

Časová dotácia: 6 hodín

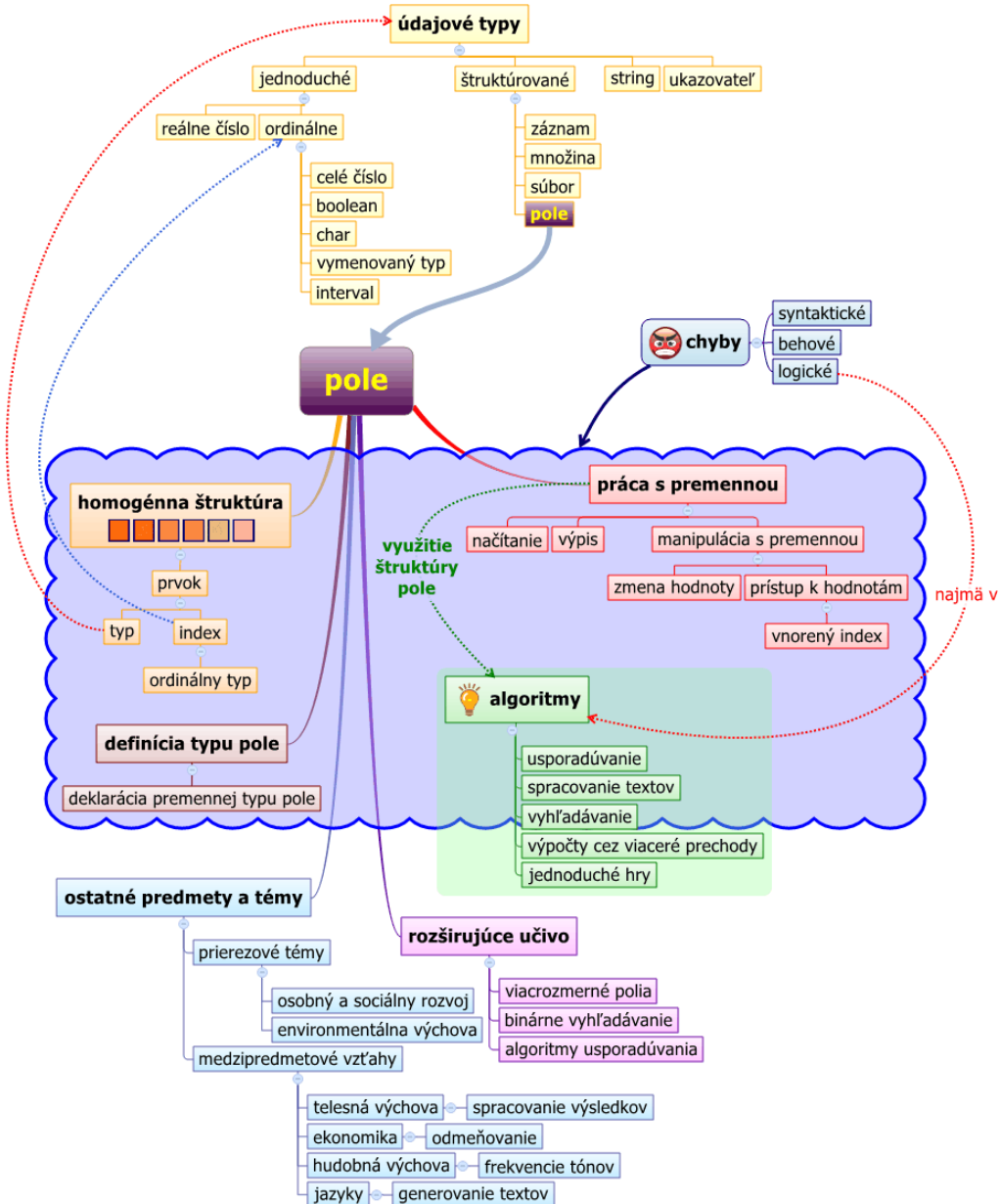
Hardvérové a softvérové podmienky: počítače v LAN, FreePascal (príp. Turbo Pascal, Borland Delphi)

Význam a postavenie témy: Táto téma patrí do základného kurzu programovania. Pole bude prezentované ako prvý štruktúrovaný údajový typ. Vzhľadom na definíciu typu pole a hlavne na používanie a prístup k jednotlivým prvkom poľa by tejto téme mala nevyhnutne predchádzať téma príkazy opakovania. Žiaci by mali vedieť pracovať s jednoduchými údajovými typmi a vedieť používať podmienené príkazy. Po tejto téme by mohla nasledovať téma štruktúrovaný údajový typ záznam (record) alebo témy vyhľadávacie, usporadúvacie či šifrovacie algoritmy.

Vstupné požiadavky: Predpokladané (vstupné) požiadavky na vedomosti, spôsobilosti, návyky, postoje, myšlienkové operácie, vlastnosti:

Poznať jednoduché údajové typy (integer, real, char, boolean) a typ string, vedieť ich používať pri riešení algoritmických problémov využívajúcich príkazy vetvenia (if, case) a príkazov opakovania (for, repeat, while), napr. vyhľadávanie čísel z danej množiny s požadovanou vlastnosťou, tabelácia funkcie, riešenie lineárnej a kvadratickej funkcie, matematické a štatistické výpočty.

Príloha 2: Logická štruktúra učiva



Obrázok 6: Logická štruktúra učiva - údajové typy a štruktúrovaný údajový typ pole

Rozširujúce učivo: Dvojmerné pole (matica s celočíselnými indexmi, hra námorná bitka, šachovnica, plán mesta), binárne vyhľadávanie, algoritmy usporadúvania.

Medzipredmetové vzťahy: Vzťah k prírode, športu (športové výkony), hudbe (frekvencie tónov), škole, financiám (výpočty odmien), využívanie IKT (SMS, GPS).

Typické chyby:

- neordinálny typ indexu,
- hodnota indexu mimo deklarovaného rozsahu,
- zamieňanie prvku poľa a jeho indexu,
- nekompatibilita typov prvkov poľa,
- neinicializovanie poľa.

Príloha 3: Očakávané výsledky

Kognitívne ciele

Po prebraní témy Štruktúrovaný údajový typ pole by mal žiak:

Intuitívne ponímať pole ako homogénnu štruktúru údajov na úrovni príkladov zo života.

Cieľ 1.: vymedziť pole ako štruktúrovaný homogénny údajový typ s prvkami označenými jedným identifikátorom,

Cieľ 2.: vymenovať príklady jednorozmerného poľa zo života (napr. zoznam žiakov v triednej knihe, zoznam pretekárov na štartovacej listine, polička s knihami, regál s policami, skriňa so zásuvkami, izby v hoteli, sedadlá v kine, polička filmu, resp. animácie).

Definovať typ pole a deklarovať premennú typu pole.

Cieľ 3.: vysvetliť čo je index poľa a prvok poľa a aký je medzi nimi rozdiel,

Cieľ 4.: rozlišovať typ indexu poľa a typ prvkov poľa,

Cieľ 5.: vysvetliť, že index poľa musí byť ordinálneho typu (nie nevyhnutne číslo, ale napr. char),

Cieľ 6.: deklarovať premennú typu pole.

Používať premennú typu pole v jednoduchých úlohách.

Cieľ 7.: načítať a vypísať jednotlivé prvky poľa,

Cieľ 8.: manipulovať s prvkami poľa:

- a) priradiť ich hodnoty do iných premenných,
- b) modifikovať ich hodnoty,
- c) vysvetliť vyhodnotenie výrazu s vnoreným indexom (napr. $A[A[3]]$),

Cieľ 9.: vymenovať bežné chyby pri použití poľa (zlý typ indexu, pretečenie indexu, nekompatibilita typov prvkov poľa),

Cieľ 10.: použiť pole pri jednoduchých úlohách (napr. typu: načítaj pole - nájdi maximálnu hodnotu, spočítaj resp. uprav prvky poľa - vypíš pole).

Vytvárať algoritmy využívajúce údajovú štruktúru pole:

Cieľ 11.: použiť pole pri riešení náročnejších problémov niektorých z uvedených typov:

- a) vyhľadávacie algoritmy,
- b) algoritmy usporadúvania,
- c) výpočty hodnôt vyžadujúce viaceré prechody poľom,
- d) problémy zamerané na spracovanie textov,
- e) programovanie jednoduchých hier.

Cieľ 12.: rozpoznať a určiť, v ktorých úlohách je možné efektívne použiť pole a v ktorých nie, vymenovať výhody a nevýhody poľa oproti jednoduchým údajovým typom (jednoduchší prístup k prvkom poľa pri načítaní, spracovaní a vypísaní premenných).

Afektívne ciele

Počas preberania témy Štruktúrovaný údajový typ pole by si mal žiak upevniť:

- zmysel pre poriadok, uvedenie si výhody vhodného označenia údajov pomocou indexov,
- zmysel pre spravodlivosť (rôzne spôsoby odmeňovania ľudí),
- pozitívny vzťah k škole, práci,
- pozitívny vzťah k prírode a športu,
- pozitívny vzťah k technológiám a psychohygienické aspekty ich využívania,
- estetické cítenie (vzťah k hudbe),
- sústredenosť a exaktné vyjadrovanie (v programovacom jazyku),
- aktívny prístup k riešeniu úloh, kritické myslenie.

Príloha 4: Systém úloh

Systém úloh tvorí desať úloh pokrývajúcich uvedené vzdelávacie a výchovné ciele s postupne narastajúcou náročnosťou, ktoré zohľadňujú jednotlivé etapy poznávacieho procesu a rôzne úrovne požadovaných myšlienkových operácií.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11a	C11b	C11c	C11d	C11e	C12
1	✓	✓	✓	✓	✓											
2						✓	✓			✓						
3			✓			✓	✓	✓		✓	✓					
4						✓	✓	✓		✓	✓					✓
5			✓	✓	✓	✓	✓	✓	✓	✓						
6						✓	✓	✓	✓	✓	✓			✓		✓
7					✓	✓		✓	✓	✓					✓	
8						✓	✓	✓	✓	✓		✓	✓			
9					✓	✓	✓		✓					✓	✓	
10						✓	✓	✓	✓	✓	✓		✓			✓

Obrázok 7: Tabuľka pokrytia didaktických funkcií jednotlivými úlohami

	1	2	3	4	5	6	7	8	9	10
motivačná, prípravná	✓									
prvotné osvojenie učiva		✓	✓							
upevnenie učiva				✓	✓	✓	✓	✓	✓	
systemizácia učiva										✓
propedeutický charakter						✓	✓	✓		✓
aplikačná		✓	✓	✓		✓	✓	✓	✓	✓

Obrázok 8: Tabuľka pokrytia vzdelávacích cieľov jednotlivými úlohami

1. úloha

1. Ako by ste označili parkovacie miesta pred hotelom?
2. Ako sú očíslované vozne vlakovkej súpravy?
3. Televíznej vedomostnej súťaže sa zúčastnilo šesť družstiev A, B, C, D, E, F. Ako by ste zaznamenávali priebežné a celkové výsledky?

Riešenie:

1. Parkovacie miesta: 1, 2, 3, ..., alebo H1, H2, H3, ...
2. Číslovanie vozňov: napr.: 361, 362, 363, ...
3. Priebežné výsledky:

kolo: A1, B1, C1, ...

kolo: A2, B2, C2, ...

...

finálové kolo: A, B, C, ...

Komentár k 1. úlohe:

Očakávame, že si žiak uvedomí:

- potrebu označenia (indexovania) prvkov rovnakého typu,
- výhodnosť ordinálneho a súvislého indexovania,
- rozdiel medzi indexom a hodnotou prvku.

Odporúčame doplniť otázky typu:

- Vedeli by ste označiť parkovacie miesta inak ako číslami a písmenkami?
- Ako by ste našli vozeň, keby boli čísla vozňov neusporiadané, resp. nesúvislé usporiadané?
- Vedeli by ste označovať vozne reálnymi číslami? Aké by to malo nevýhody?

Úloha plní motivačnú funkciu a je prípravnou úlohou pre zavedenie štruktúrovaného údajového typu pole. Úlohou chceme rozvíjať zmysel žiakov pre poriadok (označenie usporiadaných údajov).

Komentár k 2. úlohe:

Očakávame, že si žiak uvedomí potrebu zapamätania si všetkých desiatich údajov, aby ich vedel napísať v opačnom poradí a vhodnosť použitia údajovej štruktúry pole. Úlohu riešia žiaci spoločne s učiteľom.

Odporúčame doplniť otázky typu:

- Aké je prevýšenie medzi 2. a 3. bodom trasy, i . a $(i+1)$. bodom? Pre aké i vieme zistiť prevýšenie?
- Aké je celkové prevýšenie trasy medzi i . a j . bodom?

Ide o úlohu na prvotné osvojenie učiva - načítanie a vypísanie prvkov poľa. Úlohou chceme rozvíjať pozitívny vzťah žiakov k prírode a športu.

Komentár k 3. úlohe:

Očakávame, že si žiak uvedomí rozdiel medzi prvkom poľa a jeho indexom, dokáže spracovať údaje a vyhľadať údaje s danou vlastnosťou. Úlohu riešia žiaci spoločne s učiteľom.

Odporúčame doplniť otázky typu:

- Aký výsledok vráti program v časti c), ak sú v niektorých mesiacoch zaslané rovnaké počty správ?
- Ako treba zmeniť program, aby vypísal všetky mesiace, v ktorých bol zaslaný maximálny počet správ, resp. prvý mesiac, v ktorom bolo dosiahnuté maximum?

Ide o úlohu na prvotné osvojenie učiva - načítanie, vypísanie prvkov a indexov poľa, manipuláciu s prvkami. Úlohou chceme u žiakov rozvíjať pozitívny vzťah k využívaniu IKT a jeho psychohygienické aspekty.

2. úloha

Počas turistického výletu na vysokohorskú chatu sme GPS prístrojom zaznamenávali nadmorskú výšku desiatich významných bodov našej trasy. Pri späťsočnej ceste nás čaká presne opačný výškový profil trate. Vytvorte program, ktorému zadáme zaznamenané údaje v tom poradí, ako sme ich zaznamenali. Program vypíše výškový profil trasy na chatu a výškový profil späťsočnej trasy.

Riešenie:

```
program gps_profil trate;
const n = 10;
type pole = array[1..n] of integer;
var vyska : pole;
    i : integer;
begin
  writeln('Zadaj 10 nameranych vysok podla GPS:');
  for i := 1 to 10 do begin
    write('vyska ', i, ': ');
    readln(vyska[i]);
  end;

  writeln('Vyskovy profil trate smerom na chatu:');
  for i := 1 to 10 do writeln('vyska ', i, ': ', vyska[i]);

  writeln('Vyskovy profil trate smerom z chaty:');
  for i := 10 downto 1 do writeln('vyska ', i, ': ', vyska[i]);
end.
```

3. úloha

Mobilný operátor pošle na konci kalendárneho roka výpis, v ktorom je uvedené, koľko SMS sme poslali v každom mesiaci. Vytvorte program, ktorý vypočíta:

- priemerný počet SMS za mesiac,
- koľko najviac a koľko najmenej sme poslali SMS za mesiac,
- v ktorom mesiaci sme poslali najmenej a v ktorom najviac SMS.

Riešenie:

```
Program sms_statistika;
const n = 12;
type pole = array[1..n] of integer;
var sms : pole;
    i, min_mesiac, max_mesiac, sucet : integer;
    priemer: real;
begin
  writeln('Zadavaj postupne pocy SMS v jednotlivych mesiacoch');
  for i := 1 to n do begin
    write(i, '. mesiac:');
    readln(sms[i]);
  end;

  min_mesiac := 1;
  max_mesiac := 1;
  sucet := 0;
  for i := 1 to n do begin
    if sms[i] > sms[max_mesiac] then max_mesiac := i;
    if sms[i] < sms[min_mesiac] then min_mesiac := i;
    sucet := sucet + sms[i];
  end;

  writeln('Priemerny pocet sprav za mesiac: ', sucet/12:5:2);
  writeln('Najviac sms sme poslali v ', max_mesiac, '. mesiaci.');
  writeln('Bolo ich ', sms[max_mesiac], '.');
  writeln('Najmenej sms sme poslali v ', min_mesiac, '. mesiaci.');
  writeln('Bolo ich ', sms[min_mesiac], '.');
end.
```

4. úloha

Na konci žiackej brigády sa zamestnávateľ rozhodol odmeniť žiakov a k ich platu (2.5 € za hodinu) im pridal ešte odmenu. Žiakom, ktorí odpracovali viac ako 30 hodín, dal odmenu 10 % z ich platu. Ostatní dostali odmenu 5 % zo svojho platu. Vytvorte program, ktorému zadáme počty odpracovaných hodín jednotlivých žiakov a ktorý pre každého žiaka vypočíta jemu vyplatenú sumu a určí počet žiakov, ktorým bola vyplatená suma aspoň 100 €.

Riešenie:

```
Program brigada_odmenovanie_ziakov;
const max_pocet = 50;
      hodinova_mzda = 2.5;
type pole = array[1..max_pocet] of real;
var hodiny, plat : pole;
    i, pocet_ziakov, pocet_100: integer;

begin
  write('Zadaj pocet ziakov: ');
  readln(pocet_ziakov);
  writeln('Zadavaj pockty odpracovanych hodin ziakov');
  for i := 1 to pocet_ziakov do begin
    write('ziak ', i, ' ');
    readln(hodiny[i]);
  end;

  pocet_100 := 0;

  for i := 1 to pocet_ziakov do
    begin
      plat[i] := hodiny[i] * hodinova_mzda;
      if hodiny[i] > 30 then plat[i] := plat[i] * 1.1
        else plat[i] := plat[i] * 1.05;
      if plat[i] > 100 then inc(pocet_100);
    end;
  writeln('Vyplatene sumy znikom:');
  for i := 1 to pocet_ziakov do writeln('ziak ', i, ': ', plat[i]:5:2);
  writeln('Pocet ziakov s platom vyssim ako 100 EUR je: ', pocet_100);
end.
```

5. úloha

V jednom programe istého roztržitého programátora sme objavili takéto definície a deklarácie:

```
type pole1 = array[1 .. 10] of integer;
pole2 = array['a' .. 'z'] of char;
pole3 = array[1 .. 10] of char;
pole4 = array[1.0 .. 10.0] of string;
var a : pole1;
    b : pole2;
    c : pole3;
    d : pole4;
    i, j : integer;
    z : char;
    r : string;
```

Môže program obsahovať takéto zápisy?

Program navyše obsahoval nasledovné príkazy. Čo bude výsledkom nasledujúcich príkazov?

```
write(a[5]);
for i := 1 to 10 do a[i] := i;
for i := 1 to 5 do begin
  j := a[i];
  a[i] := a[10-i+1];
  a[1-i+1] := j;
end;
```

Komentár k 4. úlohe:

Očakávame, že si žiak uvedomí potrebu vopred vedieť počet prvkov poľa, spôsob, ako manipulovať s prvkami poľa, ako meniť a zisťovať hodnoty prvkov. Úlohu riešia žiaci s čiastočnou pomocou učiteľa.

Odporúčame doplniť otázky typu:

- Vedeli by ste úlohu riešiť použitím len jednej premennej typu pole?
- Vedeli by ste úlohu riešiť aj bez použitia štruktúry pole?
- Vedeli by ste úlohu vyriešiť pre vopred neznámy počet žiakov?

Ide o úlohu na upevnenie si učiva - načítanie, vypísanie prvkov, manipuláciu s prvkami poľa. Úlohou chceme rozvíjať pozitívny vzťah k práci, zmyslu k spravodlivosti pri rozdeľovaní odmien.

Komentár k 5. úlohe:

. Očakávame, že žiak si upevní poznatky o spôsoboch definovania typu pole, deklarácie premennej typu pole, manipulácii s prvkami poľa, jeho indexom. Žiak by si mal uvedomiť typické chyby, ktorých sa môže dopustiť a problémy z toho vyplývajúce. Očakávame, že toto uvedomenie si pomôže žiakovi vyhnúť sa týmto chybám. Úlohu nie je potrebné programovať, žiaci ju riešia spoločne s učiteľom. Je vhodné úlohu zaradiť pred úlohami, ktoré bude žiak riešiť samostatne.

Odporúčame doplniť otázky, v ktorých sa zameriame na najčastejšie chyby, ktoré žiaci doteraz robili. Ide o úlohu na upevnenie si učiva. Úlohou chceme rozvíjať sústredenosť a vytrvalosť žiakov pri práci (pri hľadaní chýb).

```

writeln(a[a[3]]);
for i := 1 to 10 do a[i] := a[i+1];

for i := 'a' to 'z' do b[i] := i;
for r := 'a' to 'z' do b[r] := r;
for z := 'a' to 'z' do b[z] := z;
writeln(b['b']);
writeln(b[b['b']]);
for i := 'f' downto 'b' do write(b[i]);

for i := 1 to 10 do c[i] := chr(ord(a) + i);
writeln(c[b['a']]);
for i := 1 to 10 do c[i] := chr(10 - i + ord(a));

```

Riešenie:

```

program roztrzity_programator;
type pole1 = array[1 .. 10] of integer;
    pole2 = array['a' .. 'z'] of char;
    pole3 = array[1 .. 10] of char;
    {tento zapis je chybný, index pola musí byť ordinalneho typu}
    pole4 = array[1.0 .. 10.0] of string;
var a : pole1;
    b : pole2;
    c : pole3;
    {d : pole4;}
    i, j : integer;
    z : char;
    r : string;

begin
    {nasledujúci príkaz je syntakticky správny, ale premenná 'a' nie je
    ešte inicializovaná}
    write(a[5]);

    {hodnotou prvkov pola 'a' budú hodnoty ich indexov v poli}
    for i := 1 to 10 do a[i] := i;

    {v tretom príkaze cyklu prístupujeme k prvku, ktorý v poli neexistuje,
    programátor chcel pravdepodobne vymeniat prvky v poli, prvý s posledným,
    druhý s predposledným, atď.
    v poslednom príkaze by teda mal byť správny index prvku [10-i+1]}
    for i := 1 to 5 do begin
        j := a[i];
        a[i] := a[10-i+1];
        {toto je chyba}
        a[1-i+1] := j;
        {správne by malo byť}
        a[10-i+1] := j;
    end;

    {predpokladajme, že sme opravili predchádzajúcu chybu,
    vypíšeme prvok pola 'a' s indexom 'a[3]', t.j. s indexom 8, tento prvok
    má hodnotu 3,}
    writeln(a[a[3]]);

    {toto je chyba, ak i=10, tak do premennej a[10] priradujeme hodnotu
    premennej a[11], tento prvok v poli neexistuje}
    for i := 1 to 10 do a[i] := a[i+1];

    {toto je chyba, premenná i (integer) nie je rovnakeho typu ako
    index pola b (char)}
    for i := 'a' to 'z' do b[i] := i;

    {toto je chyba, premenná r (string) nie je rovnakeho typu ako
    index pola b (char), navyše index pola by ani nemohol byť typu string}
    for r := 'a' to 'z' do b[r] := r;

    {hodnotou prvkov pola b'' budú hodnoty ich indexov v poli, aj prvky aj ich
    indexy sú typu char}
    for z := 'a' to 'z' do b[z] := z;

    {vypíšeme prvok pola b s indexom ['b'], tento prvok má hodnotu 'b'}
    writeln(b['b']);

    {vypíšeme prvok pola b s indexom b['b'], t.j. s indexom 'b', tento prvok

```

```

    ma hodnotu 'b'}
writeln(b[b['b']]);

{toto je chyba, premenna i (integer) nie je rovnakeho typu ako
index pola b (char)}
for i := 'f' downto 'b' do write(b[i]);

{toto je chyba, vstupny parameter funkcie ord by mal byt typu chr}
for i := 1 to 10 do c[i] := chr(ord(a) + i);

{toto je chyba, indexy pola c su typu integer, b['a'] je typu char}
writeln(c[b['a']]);

{toto je chyba, vstupny parameter funkcie ord by mal byt typu chr}
for i := 1 to 10 do c[i] := chr(10 - i + ord(a));
end.

```

6. úloha

Janko dostal textovú SMS. Napíšte program, ktorý vypíše počet slov v tejto správe.

Riešenie:

```

program slova_v_sms;
uses crt;
const max_sms = 160;
type pole = array[0 .. max_sms] of char;
var sms : pole;
    i, pocet_znakov, pocet_slov : integer;
    pom : char;

begin
    clrscr;
    sms[0] := ' ';
    writeln('Zadaj text SMS:');
    pocet_znakov := 0;

    repeat
        inc(pocet_znakov);
        pom := readkey;
        write(pom);
        sms[pocet_znakov] := pom;
    until pom = #13;

    dec(pocet_znakov);
    writeln;

    pocet_slov := 0;
    for i := 0 to pocet_znakov-1 do
        if (sms[i]=' ') and (sms[i+1] <> ' ') then inc(pocet_slov);

    writeln('Pocet slov v sms: ', pocet_slov);
end.

```

7. úloha

Počítač môže slúžiť aj ako hudobný nástroj, napríklad klavír. Jednotlivým klávesom vieme priradiť frekvencie tónov, ktoré sa majú zahrat' po ich stlačení. Vytvorte program, ktorý pri stláčaní klávesov bude hrať zodpovedajúce tóny (c=262, cis=277, d=294, dis=311, e=330, f=349, fis=370, g=392, gis=415, a=440, ais=466, h=494).

Riešenie:

```

program klavir;
uses crt;
type pole = array['a' .. 'z'] of integer;
var ton : pole;
    i : char;

begin
    clrscr;
    for i := 'a' to 'z' do ton[i] := 0;
    ton['z'] := 262;
    ton['s'] := 277;
    ton['x'] := 294;

```

Komentár k 6. úlohe:

Očakávame, že si žiak uvedomí možnosť využitia premennej typu pole pri spracovaní textov. Úlohu riešia žiaci samostatne, s prípadnou pomocou učiteľa.

Odporúčame doplniť otázky typu:

- Ako by ste riešili úlohu, ak by medzi slovami mohla byť viac ako jedna medzera, príp. ak by medzery mohli byť na začiatku, resp. na konci správy?
- Vedeli by ste túto úlohu vyriešiť bez použitia premennej typu pole, resp. reťazec?
- Navrhnete algoritmus, ako by ste sa zbavili nadbytočných medzier.

Ide o úlohu na upevnenie si učiva - načítanie prvkov, manipuláciu s prvkami pola, vyhľadávanie prvkov v poli. Táto úloha môže byť propedeutickou úlohou k tvorbe komprimačných algoritmov. Úlohou chceme rozvíjať vzťah k úspornému vyjadrovaniu (šetreniu prostriedkov).

Komentár k 7. úlohe:

Očakávame, že si žiak uvedomí možnosť použitia znakového indexu, rozdiel medzi hodnotou prvku a indexom prvku. Je to otvorená úloha, ktorú je možné riešiť niekoľkými spôsobmi.

Odporúčame doplniť otázky typu:

- Vedeli by ste do programu doplniť funkciu, ktorá zvýši/zníži tóny o jednu oktávu?
- Vedeli by ste úlohu vyriešiť bez pomoci pola?

Komentár k 8. úlohe:

Očakávame, že si žiak uvedomí možnosť prechodu prvkami poľa nie od prvého, resp. nie do posledného prvku, resp. možnosť preusporiadania prvkov poľa. Úlohu je možné riešiť niekoľkými spôsobmi: vypísať prvky v poradí i+1..n, 1..i, rotovať pole i-krát o jedno miesto, rotovať pole o i miest.

Odporúčame doplniť otázku typu:

1. Vedeli by ste úlohu vyriešiť, ak by kolotočiar zastavil kolotoč predčasne?
2. Je výhodnejšie prvky preusporiadať alebo len vypísať v inom poradí?

Ide o úlohu na upevnenie si učiva - manipuláciu s prvkami poľa, preusporiadanie prvkov poľa. Táto úloha môže byť propedeutickou úlohou k tvorbe algoritmov usporadúvania a tiež porozumeniu údajového typu cyklický obojsmerný zoznam. Úlohou chceme rozvíjať sústredenosť a exaktné vyjadrovanie žiakov.

Komentár k 9. úlohe:

Očakávame, že si žiak uvedomí možnosť súčasného použitia viacerých premenných typu pole, spôsob prístupu k hodnotám prvkov jednotlivých polí, mať prvok poľa typu reťazec, ktorého hodnota nie je inicializovaná.

```
ton['d'] := 311;
ton['c'] := 330;
ton['v'] := 349;
ton['g'] := 370;
ton['b'] := 392;
ton['h'] := 415;
ton['n'] := 440;
ton['j'] := 466;
ton['m'] := 494;
writeln('Elektronicky klavir');
writeln('Klavesy z..m, tony');
writeln('Klavesy s, d, g, h, j, poltony');
writeln('Koniec - q');
repeat
  i := readkey;
  if (i>='a') and (i <='z') then begin
    sound(ton[i]);
    delay(100);
    nosound;
  end;
until i = 'q';
end.
```

8. úloha

Deti nastupujú na ruské kolo v takom poradí, ako prišli k pokladni a kúpili si lístky. Nie vždy sa však podarí zastaviť kolotoč tak, aby mohli deti vystúpiť v rovnakom poradí ako nastupovali. Ak sa kolotočiar pri zastavovaní pomýli napr. o dve miesta, deti vystupujú v poradí: tretie, štvrté, ..., posledné, prvé, druhé. Vytvorte program, ktorému zadáme mená detí v tom poradí ako nastupovali na kolotoč a počet miest, o ktoré sa kolotočiar pri zastavaní kolotoča pomýlil. Program vypíše mená detí v tom poradí, ako vystupovali.

Riešenie:

```
program ruske_kolo;
uses crt;
const max = 20;
type pole = array[0 .. max] of string;
var meno : pole;
    i, pocet_deti, minul : integer;

begin
  clrscr;
  writeln('Zadaj pocet deti:');
  readln(pocet_deti);
  writeln('Postupne zadavaj mena deti v poradí, ako nastupovali');
  for i := 0 to pocet_deti-1 do begin
    write(i+1, '. dieta: ');
    readln(meno[i]);
  end;

  write('O kolko miest sa kolotociar minul? ');
  readln(minul);
  if minul < 0
  then minul := minul mod pocet_deti + pocet_deti
  else if (minul >= pocet_deti)
  then minul := minul mod pocet_deti;

  writeln('Deti vystupovali v nasledovnom poradí:');
  for i := minul to pocet_deti-1 do writeln(meno[i]);
  for i := 0 to minul-1 do writeln(meno[i]);
end.
```

9. úloha

Pani učiteľka zistila, že jej žiaci robia príliš veľa chýb v diktátoch. Rozhodla sa preto dávať im viac diktátov. Pomaly jej ale dochádzajú nápady. Bola by rada, keby mala pomocníka, ktorý by jej pomáhal vymýšľať vety. Vety sú pomerne jednoduché, a majú takýto tvar: "[kto] [s kým] [čo robili]". Namiesto každej zátvorky počítač náhodne vyberie jednu z pripravených možností:

[kto] Kamil, ujo Imro, sused Krtko, brat, potápač, ...
[s kým] s medveďom, s kozmonautom, so psom, ...

[čo robili] varili vajíčka, vyprážali rezne, kričali, ...

Vytvorte program, ktorý bude generovať vety tohto typu, pričom použije uvedené slová.

Riešenie:

```
program diktaty;
uses crt;
const max1 = 5;
      max2 = 3;
      max3 = 3;
      kto:array[0 .. max1-1] of string=('Kamil',
                                         'Ujo Imro',
                                         'Sused Krtko',
                                         'Brat',
                                         'Potapac');

      s_kym:array[0 .. max2-1] of string=('s medvedom',
                                           's kozmonautom',
                                           'so psom');

      co_robili:array[0 .. max3-1] of string=('varili vajicka',
                                              'vyprazali rezne',
                                              'kricali');

begin
  clrscr;
  randomize;
  write(kto[random(max1)]);
  write(' ');
  write(s_kym[random(max2)]);
  write(' ');
  write(co_robili[random(max3)]);
  writeln(' ');
end.
```

10. úloha

Počas cyklistického tréningu cyklopočítač cyklistu v pravidelných minútových intervaloch meria nadmorskú výšku (v metroch) a vzdialenosť od posledného merania (v metroch). Po tréningu sa namerané údaje vyhodnotia, aby cyklista vedel akú má kondíciu. Tréner údaje vyhodnocuje ručne, pomocou kalkulačky. Tento postup je však zdĺhavý a určite by sa dal pomocou počítača urýchliť. Vytvorte program, ktorému zadáme namerané údaje, a on vypočíta nasledovné údaje:

1. vzdialenosť v metroch, ktorú cyklista prešiel,
2. čas trvania tréningu,
3. priemernú rýchlosť,
4. dĺžku najdlhšieho úseku, počas ktorého cyklista stúpil a počet metrov, o ktoré vystúpil,
5. dĺžku najdlhšieho úseku, počas ktorého cyklista klesal a počet metrov, o ktoré klesol,
6. prevýšenie, počas ktorého cyklista stúpil,
7. prevýšenie, počas ktorého cyklista klesal,
8. celkové prevýšenie trate,
9. rýchlosť, ktorú cyklista dosiahol v najrýchlejšom úseku,
10. rýchlosť, ktorú cyklista dosiahol v najpomalšom úseku,
11. úsek, v ktorom bolo najprudšie stúpanie,
12. úsek, v ktorom bolo najprudšie klesanie.

Riešenie:

```
program trening_cyklistu;
uses crt;
const max_pocet_merani = 100;
type pole = array[0 .. max_pocet_merani-1] of integer;
var vyska, vzdialenost : pole;
    i, pocet_merani, celkova_vzdialenost : integer;
    max_dlzka, stupanie, klesanie : integer;
    min_vyska, max_vyska : integer;
```

Odporúčame doplniť otázku typu:

1. Vedeli by ste doplniť program tak, aby generoval vety typu: "[aký] [kto] [kedy] [s kým] [kde] [ako] [čo robili]?"
2. Ako by ste navrhli program, ktorý by generoval slová typu [spoluhláska] [samohláska] [spoluhláska] [samohláska], napr. mama?
3. Navrhnite algoritmus generovania čísel lotérie (napr. päť prirodzených čísel v rozsahu od 1 do 40) tak, aby sa tieto čísla neopakovali.

Ide o úlohu na upevnenie si učiva - manipuláciu s prvkami pola, prístup k hodnote prvku s náhodným indexom. Táto úloha môže byť propedeutickou úlohou k tvorbe hier a simulácií. Úlohou chceme rozvíjať vzťah k škole a práci.

Komentár k 10. úlohe:

Očakávame, že si žiak uvedomí možnosť využitia premennej typu pole pri spracovaní komplexných úloh. Jednotlivé časti úlohy možno riešiť nezávisle, v rôznom poradí, resp. niektoré podúlohy je možné vynechať. Úlohu riešia žiaci samostatne, s individuálnou konzultáciou s učiteľom. Ide o úlohu systemizačnú, v ktorej žiak pracuje s premennou typu pole na rôznych úrovniach. Táto úloha môže byť propedeutickou úlohou k porozumeniu a využívaniu štruktúrovaného údajového typu zoznam. Úlohou chceme rozvíjať vzťah k prírode a športu.

```

celkove_stupanie, celkove_klesanie : integer;
max_vzdialenost, min_vzdialenost : integer;
max_stupanie, max_klesanie : real;
usek_max_stupania, usek_max_klesania : integer;

begin
  clrscr;
  write('Zadaj pocet merani: ');
  readln(pocet_merani);
  writeln('Postupne zadavaj namerane udaje:');
  for i := 1 to pocet_merani do
    begin
      write('Zadaj prejdenu vzdialenost v bode ', i, '.: ');
      readln(vzdialenost[i]);
      write('Zadaj vysku v bode ', i, '.: ');
      readln(vyska[i]);
    end;
  writeln;

  {vypocet celkovej vzdialenosti}
  celkova_vzdialenost := 0;
  for i := 2 to pocet_merani do
    celkova_vzdialenost := celkova_vzdialenost + vzdialenost[i];
  writeln('Celkova prejdena vzdialenost je: ', celkova_vzdialenost);
  writeln;

  {cas treningu}
  writeln('Trening trval ', pocet_merani-1, ' minut.');
```

```

  writeln;

  {priemerna rychlost}
  writeln('Priemerna rychlost bola: ',
    (celkova_vzdialenost/1000) / ((pocet_merani-1)/60):5:2, ' km/h.');
```

```

  writeln;

  {najdlhsi usek so stupanim s ...}
  max_dlzka := 0; stupanie := 0;
  for i := 2 to pocet_merani do
    if vyska[i] - vyska[i-1] > 0 then
      if vzdialenost[i] > max_dlzka
        then begin
          max_dlzka := vzdialenost[i];
          stupanie := vyska[i] - vyska[i-1];
        end;

  writeln('Dlzka maximalneho useku so stupanim bola: ', max_dlzka, ' m.');
```

```

  writeln('Cyklista na nom vystupil o ', stupanie, ' m.');
```

```

  writeln;

  {najdlhsi usek s klesanim s ...}
  max_dlzka := 0; klesanie := 0;
  for i := 2 to pocet_merani do
    if vyska[i] - vyska[i-1] < 0 then
      if vzdialenost[i] > max_dlzka
        then begin
          max_dlzka := vzdialenost[i];
          klesanie := vyska[i-1] - vyska[i];
        end;

  writeln('Dlzka maximalneho useku s klesanim bola: ', max_dlzka, ' m.');
```

```

  writeln('Cyklista na nom klesol o ', klesanie, ' m.');
```

```

  writeln;

  {celkove stupanie}
  celkove_stupanie := 0;
  for i := 2 to pocet_merani do
    if vyska[i] > vyska[i-1]
      then celkove_stupanie := celkove_stupanie + vyska[i] - vyska[i-1];
  writeln('Celkove stupanie cyklistu bolo: ', celkove_stupanie, ' m.');
```

```

  writeln;

  {celkove klesanie}
  celkove_klesanie := 0;
  for i := 2 to pocet_merani do
    if vyska[i] < vyska[i-1]
      then
        celkove_klesanie := celkove_klesanie + vyska[i-1] - vyska[i];

  writeln('Celkove klesanie cyklistu bolo: ', celkove_klesanie, ' m.');
```

```

  writeln;

```

```

{celkove prevysenie}
min_vyska := maxint;
max_vyska := -maxint;
for i := 1 to pocet_merani do
    if vyska[i] > max_vyska then max_vyska := vyska[i];
    if vyska[i] < min_vyska then min_vyska := vyska[i];
writeln('Celkove prevysenie trate bolo: ', max_vyska - min_vyska, 'm.');
```

```

writeln;

{rychlost v najrychlejšom useku}
max_vzdialenost := -maxint;
for i := 2 to pocet_merani do
    if vzdialenost[i] > max_vzdialenost
        then max_vzdialenost := vzdialenost[i];
writeln('Rychlost v najrychlejšom useku bola: ',
    (max_vzdialenost/1000)/(1/60):5:2, ' km/h.');
```

```

writeln;

{rychlost v najpomalsom useku}
min_vzdialenost := maxint;
for i := 2 to pocet_merani do
    if vzdialenost[i] < min_vzdialenost
        then min_vzdialenost := vzdialenost[i];
writeln('Rychlost v najpomalsom useku bola: ',
    (min_vzdialenost/1000)/(1/60):5:2, ' km/h.');
```

```

writeln;

{najprudšie stupanie}
max_stupanie := -maxint;
usek_max_stupania := 0;
for i := 2 to pocet_merani do
    if (vyska[i]-vyska[i-1]) / vzdialenost[i] > max_stupanie
        then begin
            max_stupanie := (vyska[i]-vyska[i-1]) / vzdialenost[i];
            usek_max_stupania := i;
        end;
writeln('Najprudšie stupanie presiel cyklista v useku: ',
    usek_max_stupania-1, '.');
```

```

writeln;

{najprudšie klesanie}
max_klesanie := maxint;
usek_max_klesania := 0;
for i := 2 to pocet_merani do
    if (vyska[i]-vyska[i-1]) / vzdialenost[i] < max_klesanie
        then begin
            max_klesanie := (vyska[i]-vyska[i-1]) / vzdialenost[i];
            usek_max_klesania := i;
        end;
writeln('Najprudšie klesanie presiel cyklista v useku: ',
    usek_max_klesania-1, '.');
```

```

writeln;
end.

```

Príloha 5: Didaktický test

Úlohy do didaktického testu boli vybrané tak, aby reprezentatívne pokrývali učivo danej témy. Vďaka použitým formuláciám úloh sme mohli týmto testom overiť vyššie poznávacie procesy žiakov (analýzovanie, hodnotenie, tvorenie).

Navrhujeme písomnú podobu didaktického testu s časom riešenia 90 minút.

Zadania úloh, **riešenie** a **bodové skóre** didaktického testu:

1. Ktoré z uvedených definícií typu pole sú nesprávne a prečo?:

```
type pole = array [10..20] of real;  
chyba, pri definícii typu interval sa uvádzajú len dve body - 1 bod  
type pole = array ['0'.. 9] of char;  
chyba, pri definícii typu interval musia byť krajné hodnoty rovnakého typu -  
1 bod  
type pole = array [1.0..2.0] of real;  
chyba, pri definícii typu interval musia byť krajné hodnoty ordinálneho typu -  
1 bod
```

2. Definovali sme typ pole, deklarovali premenné *a*, *i*, *pom* a priradili prvkom poľa hodnoty:

```
type pole = array [1 .. 10] of integer;  
var a : pole;  
    i, pom : integer;  
begin  
    for i := 1 to 10 do a[i] := i + 2;
```

Čo bude výsledkom behu programu, ak program vykoná nasledovné príkazy?

- a. `for i := 1 to 10 do write(a[i], ' ');`
výsledok: 3 4 5 6 7 8 9 10 11 12 - 1 bod
- b. `writeln(a[a[3]]);`
výsledok: 7, resp. podľa výsledku v a) - 1 bod
- c. `for i := 1 to 9 do begin`
 `pom := a[i];`
 `a[i] := a[i+1];`
 `a[i+1] := pom;`
 `end;`
 `write(a[1], ' ', a[10]);`
výsledok: 4 3, resp. podľa výsledku v a) - 2 body
3. Na úseku rieky (od 10. km až po prameň) sme merali úroveň jej znečistenia dusičnanmi. Na každom celom kilometri sme odmerali množstvo dusičnanov vo vode. Vytvorte program, ktorý:
- načíta namerané údaje v poradí, ako sme ich namerali,
 - vypíše vzdialenosť od prameňa (v km) a namerané hodnoty dusičnanov v poradí od prameňa,
 - vypíše tie kilometre rieky, na ktorých hodnota dusičnanov prekročila hodnotu 50 mg/l,
 - vypíše tie kilometre, na ktorých došlo k zvýšeniu hodnoty dusičnanov vzhľadom na hodnotu nameranú na predchádzajúcom kilometri (v smere od prameňa).

Riešenie a javová analýza didaktického testu:

```
Program znečistenie_rieky;  
uses crt;  
const dlzka = 10;  
type pole = array[0 .. dlzka] of real; 1 bod  
var dusicnany : pole; 1 bod  
    i : integer;
```

```

begin
  clrscr;
  writeln('Zadaj namerane hodnoty dusicnanov');
  for i := dlzka downto 0 do begin 1 bod
    write(i, '. kilometer: ');
    readln(dusicnany[i]);
  end;

  writeln;

  writeln('Namerane hodnoty:');
  for i := 0 to dlzka do writeln(i, '. kilometer: ', dusicnany[i]:3:0);
  writeln; 1 bod 1 bod 1 bod

  write('Hodnota dusicnanov prekrocila uroven 50 mg/l na kilometroch: ');
  for i := 0 to dlzka do if dusicnany[i] > 50 then write(i, '.', '); 1 bod
  writeln;

  write('Zvysenie hodnot dusicnanov bolo zaznamenane na kilometroch: ');
  for i := 1 to dlzka do
    if dusicnany[i] > dusicnany[i-1] then write(i, '.', ');
  writeln; 1 bod 1 bod
end.

```

Javová analýza:

1. za každé označenie a správne zdôvodnenie 1 bod - spolu 3 body,
2. a. za správne uvedenie výsledku 1 bod,
b. za správne uvedenie výsledku 1 bod,
c. za správne uvedenie výsledku, t. j. a[1], a[10] 1 bod - spolu 2 body,
3. definícia typu a deklarácia premennej 2 body,
načítanie prvkov 1 bod,
výpis indexov, prvkov v opačnom poradí 3 body,
výpis správnych hodnôt 1 bod,
výpis správnych hodnôt a použitý index nie mimo rozsahu poľa 2 body.

Celkom: 16 bodov

Návrh prevodu bodového skóre na klasifikačnú známku:

16 - 15 bodov - výborný,
 14 - 13 bodov - chválitebný,
 12 - 11 bodov - dobrý,
 10 - 9 bodov - dostatočný,
 8 bodov a menej - nedostatočný.

Príloha 6: Návody na riešenia aktivít

Aktivita 3	Možné zadanie úlohy: a) Vytvorte program, ktorý pre dve dané prirodzené čísla vypočíta ich najväčšieho spoločného deliteľa. b) Nájdite a opravte chyby v uvedenom programe: <pre> program NSD; var x,y:integer; begin writeln('Zadaj 2 prirodzene cisla'); readln(x,y); repeat if x>y then x:=x-y else y:=y-x until x=y; writeln('Najvacsi spolocny delitel cisel ',a,b,'je ',x); end. </pre> c) Porovnajte efektívnosť predložených riešení: • Hľadanie najväčšieho spoločného deliteľa postupným testovaním deliteľnosti od menšieho zo daných čísel až po 1. • Euklidov algoritmus (odpočítavanie). • Vylepšený Euklidov algoritmus (delenie so zvyškom). • Vytvorte program, ktorý pre tri dané prirodzené čísla vypočíta ich najväčšieho spoločného deliteľa. d) Vytvorte program, ktorý pre tri dané prirodzené čísla vypočíta ich najväčšieho spoločného deliteľa.
Aktivita 5	Možné zadanie úlohy: Aké sú trendy žiakov školy v prospechu z maturity? Informačné zdroje: • maturitné výsledky za ostatné roky v podobe grafu v ročenkách školy, • výsledky z ostatného roku maturity.
Aktivita 6	Možné zadanie úlohy: Vytvorte program pre pokladničku v obchode na spočítanie hotovosti v pokladni. (Pokladnička na konci dňa spočíta počty jednotlivých mincí a bankoviek.)

Aktivita 14	Vzorové riešenie v jazyku Imagine Logo by mohlo vyzerat' nasledovne: <pre> viem cenaTovaru :kusov ak2 :kusov <= 3 [vysledok :kusov] [vysledok 3 + (:kusov-3) * 0.8] koniec viem cenaOpravena :t :c pis 1 * cenaTovaru(:t) + 0.3 * cenaTovaru(:c) koniec </pre> Pri hodnotení žiackeho riešenia je vhodné diskutovať: - o „postihoch“ žiaka za opakovanú chybu (chybný výpočet pre 3 a 4 kusy, zľava 20% znamená 0,8 z pôvodnej ceny), - o efektívnosti riešenia (namiesto dvoch AK za sebou je efektívnejší AK2), - o hromadnosti algoritmu (takmer identický postup výpočtu ceny pre tyčinky a cukríky zapíšeme v tvare procedúry s parametrom), - o postupe výpočtu pre nekorektný vstup (záporný alebo nie celý počet kusov).
Aktivita 15	Slovné hodnotenie sa dá uviesť napr. pri hodnotení projektu, kde potrebujeme zahrnúť do hodnotenia aj iné aspekty, ťažko získateľné z udelenej známky (usilovnosť, estetiku, etický prístup, vzťah k rodine, vlasti atď').
Aktivita 16	Samohodnotenie pokladáme za veľmi dôležité, lebo žiak by si mal ním trénovať svoje schopnosti sa sám adekvátne ohodnotiť a uvedomiť si svoju zodpovednosť. Pri vzájomnom žiackom hodnotení si môžu žiaci rozvíjať svoje kritické myslenie, zručnosti argumentovať, komunikačné spôsobilosti a i. Veľmi zaujímavou aktivitou využívajúcou vzájomné hodnotenie je Turnaj mladých fyzikov (http://www.tmfsrc.sk/). Ide o vedeckú diskusiu prebiehajúcu v blokoch - fyzbojoch (fyzikálny súboj), ktoré prebiehajú medzi 3 (4) družstvami v 3 (4) etapách tak, že každé družstvo v jednej etape zaujme jedno z 3 (4) postavení „referent - oponent - recenzent (- pozorovateľ)“.
Aktivita 18	Popri hodnotení žiakov v predmete informatika, je dôležité evalvovať celkovú prácu učiteľa. Učiteľ by mal byť schopný urobiť evalváciu svojej výučby na základe vlastných terénnych zápiskov, známok žiakov, dotazníkov, resp. interview k výučbe atď'.

Tento študijný materiál vznikol ako súčasť národného projektu Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika v rámci Aktivít „Vzdelávanie nekvalifikovaných učiteľov informatiky na 2. stupni ZŠ a na SŠ“.

Autori © Mgr. Ján Guniš
 RNDr. Ľubomír Šnajder, PhD.

Názov Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika

Podnázov Aktivizujúce metódy vo výučbe školskej informatiky

Študijný materiál prešiel recenzným pokračovaním.

Recenzenti Doc. RNDr. Stanislav Krajčí, PhD.
 RNDr. Gabriela Lovaszová, PhD.

Počet strán 40

Náklad 300 ks

Prvé vydanie, Bratislava 2009

Všetky práva vyhradené.

Toto dielo ani žiadnu jeho časť nemožno reprodukovat' bez súhlasu majiteľa práv.

Vydal Štátny pedagogický ústav, Pluhová 8, 830 00 Bratislava, v súčinnosti s Univerzitou Pavla Jozefa Šafárika v Košiciach, Univerzitou Komenského v Bratislave, Univerzitou Konštantína Filozofa v Nitre, Univerzitou Mateja Bela v Banskej Bystrici a Žilinskou univerzitou v Žiline

Vytlačil BRATIA SABOVCI, s r.o., Zvolen

ISBN 978-80-8118-012-5