

Ukážky vzorových aktivít zameraných na okruh – Algoritmické riešenie problémov

Informatika na základných a stredných školách

doc. RNDr. Ľubomír Salanci, PhD. – člen predmetovej komisie pre informatiku

Algoritmické riešenie problémov – analýza problému

Výkonový štandard	Obsahový štandard	Aktivity
<ul style="list-style-type: none">✓ identifikujú vstupné informácie zo zadania úlohy✓ popíšu očakávané výstupy, výsledky, akcie✓ identifikujú problém, ktorý sa bude riešiť algoritmicky✓ sformulujú a neformálne (prirodzeným jazykom) vyjadria ideu riešenia✓ uvažujú o vlastnostiach vykonávateľa (napr. korytnačka, grafické pero, robot, apod.)✓ naplánujú riešenie úlohy ako postupnosť príkazov vetvenia a opakovania	<p>Vlastnosti a vzťahy: zadaný problém – vstup – výstup</p> <p>Procesy: rozdelenie problému na menšie časti, syntéza riešenia z riešení menších častí, identifikovanie opakujúcich sa vzorov, identifikovanie miest pre rozhodovanie sa (vetvenie a opakovanie), identifikovanie všeobecných vzťahov medzi informáciami</p>	<ul style="list-style-type: none">- uviesť príklady, aké výsledky dostaneme pre zadané vstupy,- mám úlohu, v ktorej sa kreslia vedľa seba krúžky ... čo sa bude opakovať- napr. nesmie byť záporné (uvažovať o obmedzeniach, ktoré súvisia s riešením úlohy)- riešiť úlohu iba s použitím 1 cyklu (uvažovať o obmedzeniach, ktoré súvisia s riešením úlohy)- pre 1 sa program správa inak, ako pre všetky ostatné (uvažovať o hraničných prípadoch)

Algoritmické riešenie problémov – jazyk na zápis riešenia

Výkonový štandard	Obsahový štandard	Aktivity
<ul style="list-style-type: none"> ✓ používajú jazyk na zápis algoritmického riešenia problému (používajú konštrukcie jazyka, aplikujú pravidlá jazyka) ✓ používajú matematické výrazy pri vyjadrovaní vzťahov a podmienok ✓ rozoznajú a odstránia chyby v zápise ✓ vytvárajú zápisy a interpretujú zápisy podľa nových stanovených pravidiel (syntaxe) pre zápis algoritmov 	<p>Pojem: program, programovací jazyk</p> <p>Vlastnosti a vzťahy: zápis algoritmu a vykonanie programu, vstup – vykonanie programu – výstup/akcia</p> <p>Procesy: zostavenie programu, identifikovanie, hľadanie, opravovanie chýb</p>	<p>návod/postup – algoritmus/program</p> <p>vstup – program – výstup</p> <p>programovací jazyk alebo prostredie</p> <p>program (návod): otvorenie, uloženie</p> <p>program (návod): spustenie, vykonanie, zastavenie</p> <p>program (návod): vytvorenie {nový program}</p> <p>program ako návod: dodržiavanie pravidiel zápisu {syntax}</p> <p>návod, postup – algoritmus – program – vykonanie pg.</p>

Algoritmické riešenie problémov – pomocou postupnosti príkazov

Výkonový štandard	Obsahový štandard	Aktivity
<ul style="list-style-type: none"> ✓ riešia problém skladaním príkazov do postupnosti, ✓ aplikujú pravidlá, konštrukcie jazyka pre zostavenie postupnosti príkazov 	<p>Pojmy: príkaz, parameter príkazu, postupnosť príkazov</p> <p>Vlastnosti a vzťahy: ako súvisia príkazy a výsledok realizácie programu</p> <p>Procesy: zostavenie a úprava príkazov,</p>	<p>príkaz a (matematický) výraz</p> <p>postupnosť príkazov {napr. telo programu}</p> <p>parametre príkazu {writeln(1, 2, 3);}</p> <p>príkaz: zostavenie (výberom/zadaním)</p> <p>{Image1.Canvas.TextOut... ťahanie</p>

	<p>vyhodnotenie postupnosti príkazov, úprava sekvencie príkazov (pridanie, odstránenie príkazu, zmena poradia príkazov)</p>	<p>kartičiek}</p> <p>príkaz: určenie parametra (výberom/zadaním)</p> <pre>{TextOut(0, 0, "Ahoj") writeln("1+2=", 3) }</pre> <p>príkaz: upravenie {writeln(1+2)}</p> <p>príkaz (+ parameter príkazu) – výsledok {vľavo hore sa vypíše Ahoj na obrazovke pribudne 3}</p> <p>výraz: pravidlá zápisu, jednoduché operácie {writeln(1+2*3) ... aj toto môžeme}</p> <p>príkaz - výraz {najskôr sa vyhodnotí výraz, potom sa vypíše 7}</p> <p>príkazy: pravidlá jazyka pre zostavenie sekven. príkazov {begin ... end, príkazy treba oddeliť bodkočiarkou}</p> <p>príkazy: znázornenie činnosti vykonávaného prog.</p> <pre>{Rectangle, Ellipse writeln pohyb</pre>
--	---	--

		<p>robota, ...}</p> <p>príkazy: riešenie úlohy zostavením sekvencie príkazov</p> <p>{zobrazte reklamný nápis "Mám rád ..."} príkazy: úprava sekvencie, pridanie, odstránenie príkazu</p> <p>{farebný reklamný nápis}</p> <p>sekvencia príkazov – výsledok</p> <p>{príkazy sa vykonávajú v poradí}</p> <p>{program nás najskôr pozdraví, potom čaká na vstup, ...}</p> <p>{zlé poradie – zlý výsledok (občas ale nie)}</p> <p>sekvencia príkazov – aké úlohy sa dajú riešiť</p> <p>{na riešenie úlohy stačí postupnosť príkazov}</p> <p>úloha – chyba – príkazy</p> <p>{zlé parametre, poradie príkazov, nesprávna syntax}</p>
--	--	---

Algoritmické riešenie problémov – pomocou nástrojov na interakciu

Výkonový štandard	Obsahový štandard	Aktivity
<ul style="list-style-type: none"> ✓ rozpoznávajú situácie, kedy treba získať vstup, ✓ identifikujú vlastnosti vstupnej informácie (obmedzenia, rozsah, formát), ✓ rozpoznávajú situácie, kedy treba zobrazit' výstup, realizovať akciu, ✓ zapisujú algoritmus, ktorý reaguje na vstup, ✓ vytvárajú hypotézu, ako neznámy algoritmus spracováva zadaný vstup, ak sú dané páry vstup–výstup/akcia 	<p>Vlastnosti a vzťahy: prostriedky jazyka pre: získanie vstupu, spracovanie vstupu a zobrazenie výstupu</p> <p>Procesy: čakanie na neznámy vstup – vykonanie akcie – výstup, následný efekt</p>	<p>vstup, výstup (programu)</p> <p>typy (údajov - intuitívne) {napríklad číslo, text, farba}</p> <p>vstup: prostriedky jazyka pre získanie vstupu {Edit, myš a udalosti <code>readln</code> senzor}</p> <p>vstup: pravidlá používania {syntax, miesto v programe}</p> <p>prostriedok jazyka pre vstup - ako sa prejaví v činnosti {program čaká, kým nezadáme údaje a stlačíme enter}</p> <p>výstup: prostriedky jazyka pre zobrazenie výstupu {kreslenie <code>writeln</code> zvuky pohyb robota}</p> <p>výstup: pravidlá používania {syntax, parametre}</p> <p>prostriedok jazyka pre výstup - ako sa prejaví v činnosti</p>

		<p>{vypíše na obrazovku text, číslo, výsledok výrazu}</p> <p>vstupné informácie - ich reprezentácia - zadanie</p> <p>{dĺžka -> číslo -> použijem príkaz <code>readln</code>}</p> <p>výsledok výpočtu - zobrazenie - ich interpretácia</p> <p>{číslo -> použijem príkaz <code>writeln</code> -> je to obsah štvorca}</p> <p>vstup, výstup - interaktívna komunikácia programu</p> <pre>{write('Zadaj cislo:'); readln(n); writeln(n*n) }</pre> <p>vzťahy: vstupé hodnoty - výpočet - očakávaný vstup</p> <p>{pre 3 sa urobí 3*3 a malo by sa zobrazit' 9}</p> <p>{čo treba zadať, aby program zobrazil 100?}</p> <p>údaje rôznych typov - čo nimi môžeme reprezentovať</p>
--	--	---

		<p>{číslo - dĺžka kroku, čakanie, farba - čiary, výplne, ...}</p> <p>údaje rôznych typov - aké operácie sa dajú aplikovať</p> <p>{čísla sa dajú sčítať, porovnať, ... texty nejde násobiť}</p> <p>ďalšie aktivity:</p> <ul style="list-style-type: none"> - niečo sa udeje pri kliknutí myšou - niečo sa nakreslí pri nastavení posuvnej lišty - šípkami ovládam postavičku v labyrinte - vypočítať súčet čísel, ktoré sú zadávané na vstupe - na bežiacom páse putujú farebné kocky, robot čaká na čiernu kocku, potom ju preloží na iné miesto
--	--	---

Algoritmické riešenie problémov – pomocou premenných

Výkonový štandard	Obsahový štandard	Aktivity
<p>✓ identifikujú zo zadania úlohy, ktoré údaje musia byť zapamätané, resp. sa menia (a teda vyžadujú použitie premenných)</p>	<p>Pojmy: premenná, meno (pomenovanie) premennej, hodnota premennej, operácia (+, -, *, /)</p>	<p>vzťah: pamäť – premenná – na čo slúži {pamätanie údajov, ak treba viackrát rovnaký výpočet}</p>

<ul style="list-style-type: none"> ✓ riešia problémy, v ktorých si treba zapamätať a neskôr použiť zapamätané hodnoty vo výrazoch, ✓ zovšeobecňujú riešenie tak, aby fungovalo nielen s konštantami 	<p>Vlastnosti a vzťahy: pravidlá jazyka pre použitie premennej, meno premennej – hodnota premennej</p> <p>Procesy: nastavenie hodnoty (priradenie), zistenie hodnoty (použitie premennej), zmena hodnoty premennej, vyhodnocovanie výrazu s premennými, číslami a operáciami</p>	<p>{konšt. – jednoúčelový program, prem. – rôzne vstupy}</p> <p>premenná: pravidlá jazyka pre použitie premennej</p> <p>{novú premennú treba deklarovať, pomenovať, ...}</p> <p>{pravidlá pre pomenovanie premenných}</p> <p>premenná: pravidlá jazyka pre nastavenie premennej</p> <p>{príkaz priradenia, syntax, d:=10;}</p> <p>premenná – ako funguje (prvé) nastavenie premennej</p> <p>{vyhodnotí sa pravá strana, výsledok sa uloží do prem.}</p> <p>premenná – ako funguje menenie obsahu premennej</p> <p>{prepíše sa obsah prem. – pôvod. hodnota sa zabudne}</p> <p>premenná: pravidlá jazyka pre zistenie hod. premennej</p> <p>{stačí napísať meno premennej}</p>
---	--	---

		<p>{viditeľosť premennej}</p> <p>premenná – ako funguje zistenie obsahu premennej</p> <p>{procesor sa pozrie do pamäti a vyberie hodnotu}</p> <p>premenná – aké typy údajov si môže pamätať</p> <p>{integer, string, ...}</p> <p>úlohy: keď je premenná súčasťou príkazu (parametra)</p> <p>{parameter príkazu, nastavenia napr. writeln(n)}</p> <p>úlohy: keď je premenná súčasťou výrazu</p> <p>{vyhodnotenie výrazu, napr. writeln(n*n+1)}</p> <p>úlohy: treba zistiť, ako funguje daný program s premen.</p> <p>{ako sa zmení obsah premenných}</p> <p>{trasovanie programu, premenné ako krabičky}</p> <p>problém – chyba – premenná</p>
--	--	---

		<p>{neinicializovaná premenná, nesprávna syntax }</p> <p>ďalšie aktivity:</p> <ul style="list-style-type: none"> - počítanie skóre, zostaviť jednoduché výrazy, riešiť problémy, v ktorých treba použiť premennú ako počítadlo, riešiť problémy, v ktorých treba vyjadriť závislosť pomocou premenných
--	--	---

Algoritmické riešenie problémov – pomocou cyklov

Výkonový štandard	Obsahový štandard	Aktivity
<ul style="list-style-type: none"> ✓ rozpoznajú opakujúce sa vzory ✓ rozpoznajú, aká časť algoritmu sa má vykonať pred, počas aj po skončení cyklu, ✓ riešia problémy, v ktorých treba výsledok získať akumulovaním čiastkových výsledkov v rámci cyklu ✓ riešia problémy, ktoré vyžadujú neznámy počet opakovaní, ✓ riešia problémy, v ktorých sa kombinujú cykly a vetvenia 	<p>Pojmy: opakovanie, počet opakovaní, podmienka vykonávania cyklu, telo cyklu</p> <p>Vlastnosti a vzťahy: ako súvisí počet opakovaní s výsledkom, čo platí po skončení cyklu</p> <p>Procesy: vyhodnotenie hraníc/podmienky cyklu, vykonávanie cyklu</p>	<p>cyklus {for ... to ... do ...}</p> <p>telo cyklu {jeden príkaz, resp. begin...end}</p> <p>počet opakovaní {hranice cyklu}</p> <p>vzťah: program – cyklus – na čo slúži</p> <p>cyklus: pravidlá jazyka pre konštrukciu cyklu</p> <p>{syntax: for prem:=od to po do telo cyklu}</p> <p>konštrukcia cyklu – ako funguje</p> <p>{čo sa opakuje, koľko krát sa opakuje, čo</p>

<p>✓ stanovujú hranice a podmienky vykonávania cyklov</p>		<p>sa mení}</p> <p>zadanie úlohy – identifikovanie opakujúcich sa vzorov</p> <pre>{vypísať 10 pozdravov -> treba 10 * writeln('Ahoj')}</pre> <p>opakujúce sa vzory – konštrukcia pre cyklus</p> <pre>{10*writeln('Ahoj') -> for i:=1 to 10 do writeln('Ahoj')}</pre> <p>úlohy: s pevným počtom opakovaní</p> <pre>{for i:=1 to 10 do ...}</pre> <p>úlohy: niečo iné sa vykoná pred/po cykle</p> <pre>{príkaz; for ...}</pre> <p>úlohy: telo cyklu obsahuje viac príkrov</p> <pre>{ako sa príkazy vykonajú, begin ... end}</pre> <p>úlohy: treba použiť viac cyklov za sebou</p> <pre>{for i:=1 to 10 do ...; for i:=20 to 30 do ...; }</pre> <p>úlohy: treba zistiť, ako funguje daný program s cyklom</p> <pre>{čo sa nakreslí, čo sa vypočíta} {trasovanie programu, trasovacie tabuľky}</pre>
---	--	--

		<p>{koľko krát treba zopakovať, tak aby sme videli 10 vecí}</p> <p>problém – chyba – cyklus</p> <p>{zlé hranice cyklu, nesprávna syntax}</p> <p>d ďalšie aktivity:</p> <ul style="list-style-type: none"> - použiť obmedzený počte príkazov (tak, že je nutné využiť cyklus) - nakresliť 10 bodiek - slovne povedať, čo vykoná zadaný program - nakresliť mriežku (pomocou úsečiek a dvoch samostatných cyklov) - nakresliť dvojrozmernú mozaiku
--	--	---

Algoritmické riešenie problémov – pomocou vetvenia

Výkonový štandard	Obsahový štandard	Aktivity
<ul style="list-style-type: none"> ✓ rozpoznajú situácie a podmienky, kedy treba použiť vetvenie ✓ rozpoznajú, aká časť algoritmu sa má vykonať pred, v rámci a po skončení vetvenia ✓ riešia problémy, ktoré vyžadujú vetvenie 	<p>Pojmy: vetvenie, podmienka</p> <p>Vlastnosti a vzťahy: pravda/nepravda – splnená/nesplnená podmienka</p> <p>Procesy: zostavovanie a upravovanie vetvenia, vytvorenie podmienky a vyhodnotenie podmienky s negáciami a</p>	<p>vzťah: program – vetvenie – na čo slúži</p> <p>vetvenie: pravidlá jazyka pre konštrukciu vetvenia</p> <p>{syntax: if ... then ... else}</p> <p>{syntax: if ... then ... }</p> <p>podmienka: pravidlá jazyka pre jednoduché</p>

<p>so zloženými podmienkami (s logickými spojkami)</p> <p>✓ riešia problémy, v ktorých sa kombinujú cykly a vetvenia</p>	<p>logickými spojkami (a, alebo)</p>	<p>podmienky</p> <p>{provonania =, <>, <, >}</p> <p>konštrukcia vetvenia – ako funguje</p> <p>{ako sa vyhodnotí podmienka}</p> <p>{čo sa vykoná, kedy sa vykoná/preskočí</p> <p>vetva then...}</p> <p>zadanie úlohy – stanovenie podmienky a alternatív</p> <p>{vypísať "mrzne" alebo "teplo" podľa teploty}</p> <p>{-> keď je <=0 ... "mrzne", inak ... "teplo"}</p> <p>podm. a alternatívay – konštrukcia pre vetvenie</p> <p>{if ... <=0 then ... ('mrzne') else ... ('teplo') }</p> <p>úlohy: niečo sa vykoná iba v určitej situácií</p> <p>{if ... then ...}</p> <p>úlohy: niečo sa vykoná v oboch prípadoch</p> <p>{if ... then ... else...}</p> <p>úlohy: vždy sa niečo ešte vykoná aj pred / po vetvení</p>
--	--------------------------------------	--

		<p>{priказ; if ... then ... }</p> <p>úlohy: telo prodm. príkazu obsahuje viac príkazov</p> <p>{if ... then begin ... end}</p> <p>úlohy: treba použiť podm. príkazov za sebou</p> <p>{if ... then ...; if ... then ...;}</p> <p>úlohy: treba podmienený príkaz v cykle</p> <p>{for ... to ... do if ... then ...;}</p> <p>úlohy: treba zistiť, ako funguje daný program s vetvením</p> <p>{čo sa nakreslí, čo sa vypočíta}</p> <p>{trasovanie programu, trasovacie tabuľky}</p> <p>{aký má byť vstup, aby sa vypísalo...}</p> <p>problém – chyba – podmienka</p> <p>{zlá podmienka, zlé telo, nesprávna syntax}</p> <p>ďalšie aktivity:</p> <p>- podľa podmienky vypísať si vysoký, alebo nie</p>
--	--	--

Algoritmické riešenie problémov – interpretáciu zápisu riešenia

Výkonový štandard	Obsahový štandard	Aktivity
<ul style="list-style-type: none"> ✓ krojújú riešenie, simulujú činnosť vykonávateľa s postupnosťou príkazov, s výrazmi a premennými, s vetvením a s cyklami, ✓ vyjadrujú ideu daného návodu (objavujú a vlastnými slovami popíšu ideu zapísaného riešenia – ako program funguje, čo zápis realizuje pre rôzne vstupy) ✓ upravia riešenie úlohy vzhľadom na rôzne dané obmedzenia, ✓ doplňujú, dokončujú, modifikujú rozpracované riešenie, ✓ hľadajú vzťah medzi vstupom, algoritmom a výsledkom, ✓ uvažujú o rôznych riešeniach, navrhujú vylepšenie, 	<p>Vlastnosti a vzťahy: jazyk - vykonanie programu</p> <p>Procesy: krokovanie, čo sa deje v počítači v prípade chyby v programe</p>	<p>d'alšie aktivity:</p> <ul style="list-style-type: none"> - zápis programu a vykonanie programu: {vidím takýto program, program bude toto robiť} - posúdiť, či je spolužiakove riešenie správne - či je výstup programu správny (podľa zadania) - či sa vypočíta to, čo sa má počítať - či je graf správne nakreslený? - dá sa riešenie zapísať aj s menším počtom príkazov?

Algoritmické riešenie problémov – hľadanie a opravovanie chýb

Výkonový štandard	Obsahový štandard	Aktivity
<ul style="list-style-type: none"> ✓ rozpoznajú, že program pracuje nesprávne, ✓ hľadajú chybu vo vlastnom, nesprávne pracujúcom programe a opravujú ju, ✓ zisťujú, pre aké vstupy, v ktorých prípadoch, situáciách program zle pracuje, ✓ uvádzajú kontra príklad, kedy niečo neplatí, nefunguje, ✓ posúdia a overia správnosť riešenia (svojho aj cudzieho), ✓ rozlišujú chybu pri realizácii od chyby v zápise 	<p>Vlastnosti a vzťahy: chyba v postupnosti príkazov (zlý príkaz, chýbajúcu príkaz, vymenený príkaz alebo príkaz navyše), chyba vo výrazoch s premennými, chyba v algoritmoch s cyklami a s vetvením, chyba pri realizácii (logická chyba), chyba v zápise (syntaktická chyba)</p> <p>Procesy: rozpoznanie chyby, hľadanie chyby</p>	<p>chyby: identifikovanie, hľadanie, opravovanie</p> <p>{počíta správne? – tu je chyba – treba ju takto opraviť}</p> <p>chyba - ako zle fungujúci program</p> <p>{nedá sa skompilovať, dáva zlé výsledky, spadne}</p> <p>aktivity:</p> <ul style="list-style-type: none"> - hľadať chyby v programe, kde je zle vymyslený výraz, podmienka, zostavený podmienený príkaz - rozhodnúť, prečo je Jankine riešenie správne a Jožkove nesprávne - hľadať v programe chybu v zle pracujúcom programe, že program krokujeme, kreslíme a zapisujeme si, čo sa deje - opraviť zlý parameter príkazu (má sa kresliť ..., ale kreslí sa iné)

		<p>- bežiaci program sa zle správa, treba zistiť: kedy to zle funguje, v ktorej časti programu (vylúčiť dobré/fungujúce časti)</p>
--	--	--