

Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika

Programovanie 1

Predmet: Programovanie

Línia: Vlastný odborový kontext informatiky a informatickej výchovy



EURÓPSKA ÚNIA



Európska únia
Európsky sociálny fond

Programovanie 1

Identifikácia modulu

Aktivita projektu: 1.3 Ďalšie vzdelávanie kvalifikovaných učiteľov informatiky na 2. stupni ZŠ a na SŠ

Línia aktivity: Vlastný odborový kontext informatiky a informatickej výchovy

Predmet: Programovanie 1

Zaradenie modulu



Líniu Informatika v tomto vzdelávaní tvorí 9 modulov. 5 z nich sa venuje programovaniu. V rámci týchto piatich modulov sa účastníci stretnú s rôznymi vývojovými prostrediami a programovacími jazykmi. Modul Programovanie 1 je prvý „programátorský“ modul v rámci vzdelávania. Je zameraný na programovanie v prostredí Imagine. Nasledujúce dva moduly Programovanie 2 a Programovanie 3 sa venujú programovaniu v prostredí Turbo Delphi. Modul Programovanie 4 umožňuje pokračovať v programovaní v prostredí Turbo Delphi alebo Imagine (podľa výberu), modul Programovanie 5 ponúka jednu z alternatív Java, Python alebo PHP.

Abstrakt modulu

Modul Programovanie 1 sa venuje programovaniu v programovacom jazyku Logo. Na vzdelávanie budeme využívať prostredie Imagine, verziu so slovenskými príkazmi. Modul bude pozostávať zo štyroch tematických jednotiek. V prvej časti sa účastník modulu naučí používať základné príkazy korytnačej grafiky, cyklus s pevným počtom opakovaní pri kreslení obrázkov s opakujúcimi sa časťami, vytvorí jednoduchý projekt na ovládanie korytnačky tlačidlami. V druhej časti bude účastník vytvárať vlastné príkazy (bez aj s parametrami) na kreslenie mnohoúhelníkov, kružníc, oblúkov a iných obrázkov, skladat príkazy do zložitejších „stavebníc“, čítať príkazy, analyzovať ich a rozširovať, hľadať chyby a opravovať ich. Naučí sa využívať a zakomponovať do svojho projektu posúvače. V tretej časti sa účastník oboznámi s rôznymi udalosťami korytnačky, s nastaveniami korytnačky v rodnom liste. Bude používať podmienky, vkladať nové korytnačky a definovať im udalosti. Bude pokračovať vo vývoji a vylepšovaní projektov z predchádzajúcich častí. V poslednej časti bude účastník pracovať s viacerými korytnačkami, naučí sa ich osloviť, meniť im tvary, spúšťať pre ne jednoduché procesy. Vyvinie niekoľko projektov s využitím korytnačiek, rôznych komponentov (tlačidlá, posúvače), udalostí a procesov. Počas vzdelávania v rámci modulu Programovanie 1 účastník spozná a naučí sa efektívne používať základné nástroje prostredia Imagine, vyvinie niekoľko jednoduchých projektov a získa námety na ďalšie projekty.

Garant predmetu:

PaedDr. Daniela Bezáková
KZVI FMFI UK, Bratislava
bezakova@fmph.uniba.sk

Autori:

PaedDr. Daniela Bezáková
KZVI FMFI UK Bratislava
RNDr. Gabriela Lovászová ,
PhD.
KI FPV UKF, Nitra
Mgr. Peter Kučera
1. súkromné gymnázium,
Bajkalská 20, Bratislava

Rukopis odovzdaný:

27. august 2009



Obsah

Programovanie 1.....	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
Programovanie v prostredí Imagine	4
Kapitola 1: Základy korytnačej grafiky	4
Kapitola 2: Vlastné príkazy	10
Kapitola 3: Udalosti korytnačky. Viac korytnačiek.....	15
Kapitola 4: Tvar korytnačky. Procesy.....	16
Čo sme sa naučili v tomto module.....	18
Preverenie výstupných vedomostí	18
Literatúra a použité zdroje	19

Úvod

Logo vzniklo ako nástroj na učenie a učenie sa. Patrí medzi najsilnejšie nástroje konštruktivismu a konštrukcionizmu. Jeho základnou myšlienkou je: *Učme deti pristupovať k neznámemu objavovateľsky.*

Logo je programovací jazyk a filozofia vzdelávania zároveň.

Seymour Papert [6]

Prvú verziu jazyka Logo vyvinuli Papert a Feurzeig v spoločnosti Bolt, Beranek and Newman v USA v roku 1967. Koncom 70-tych a začiatkom 80-tych rokov začalo Logo prenikať do škôl. Vznikali nové verzie Loga v rôznych jazykových interpretáciách. Začiatkom 90-tych rokov vznikla na Matematicko-fyzikálnej fakulte UK v Bratislave slovenská implementácia Loga - Comenius Logo pre Windows [7]. O necelých desať rokov neskôr vytvorili autori Comenius Loga novšiu verziu - Imagine Logo - prostredie založené na jazyku Logo, ktoré umožňuje pracovať s objektmi, spúšťať paralelné procesy, definovať udalosti a pracovať s multimédiami [8]. Dnes sa táto verzia Loga používa v rôznych jazykových lokalizáciách na vyučovaní v mnohých krajinách sveta: Veľkej Británii, Portugalsku, Maďarsku, Česku, Poľsku, Litve a Brazílii.

Základom Loga je svet korytnačky (či už ide o robotický model pohybujúci sa po zemi, tzv. Roamer, alebo o simuláciu na obrazovke), ktorú dieťa riadi jednoduchými príkazmi na posúvanie dopredu alebo vzad o daný počet krokov a otáčanie vpravo či vľavo o daný uhol. Korytnačke na obrazovke môžeme tiež nastavovať vlastnosti pera: farbu, hrúbku, vzor. Toto riadenie sa zväčša robí za účelom vytvorenia kresby.

Množina základných príkazov Loga bola vyvinutá najmä s ohľadom na detského používateľa. Poskytuje mu silný nástroj na experimentovanie, tvorivú činnosť, na učenie sa objavovaním a skúmaním. Umožňuje už aj s veľmi malými podmnožinami jazyka stavať rôzne "stavbičky", ktoré sú deťom (motivačne) blízke a už od prvých stretnutí s počítačom im ukazujú, že sú schopné aj samostatne tvoriť [9].

Pre realizáciu modulu predpokladáme nainštalované: akademická verzia prostredia Imagine (je zhodná s verziou, ktorú majú základné a stredné školy z projektu Infovek), LogoMotion, webový prehliadač (Firefox, Internet Explorer), Imagine plugin, Acrobat Reader. Realizácia modulu si nevyžaduje špeciálne hardvérové požiadavky.



Cieľ modulu

Cieľom modulu je oboznámiť účastníkov so základnými príkazmi programovacieho jazyka Logo, predstaviť prostredie Imagine, ukázať jeho možnosti a nástroje, ktoré poskytuje pre vývoj jednoduchých projektov (komponenty, udalosti, paralelné procesy...), rozvíjať tvorivosť, algoritmické myslenie a schopnosť riešiť problémy. Účastníci by mali spoznať a vyskúšať si viaceré projekty prostredia Imagine a získať rôzne námety, ktoré môžu využívať pre výučbu programovania so svojimi žiakmi.

Vstupné vedomosti

Požadované prerekvizity

Nepožadujeme absolvovanie žiadnych predchádzajúcich modulov.

Predpokladané vstupné vedomosti, skúsenosti a zručnosti

Účastník modulu by mal:

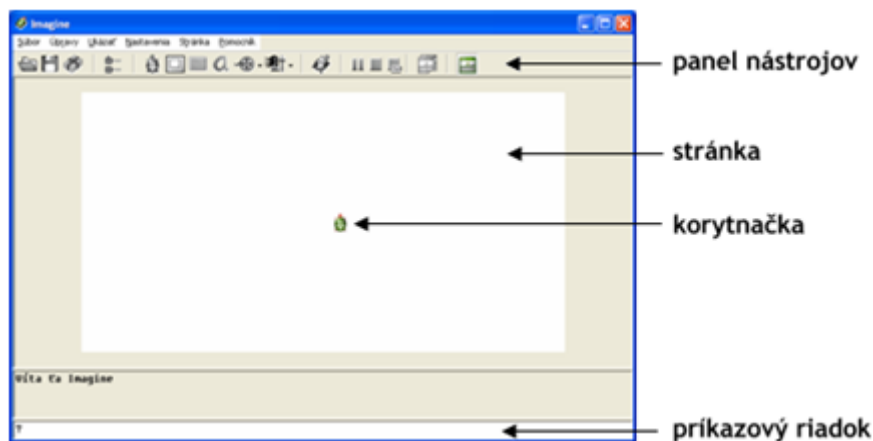
- vedieť pracovať s OS Windows, so súbormi a s prehliadačom,
- rozumieť pojmom cyklus, procedúra, parameter, podmienený príkaz,
- vedieť používať cyklus a vetvenie a definovať procedúry aj s parametrami v nejakom programovacom jazyku,
- vedieť čítať kód v nejakom programovacom jazyku a analyzovať ho,
- vedieť si rozdeliť problém na menšie podproblémy.

Programovanie v prostredí Imagine

Kapitola 1: Základy korytnačej grafiky

Základné príkazy

Hlavnou postavou prostredia Imagine je **korytnačka**. Je to malý kresiaci robot žijúci na **stránke**, ktorý robí to, čo mu povieme. S korytnačkou sa rozprávame pomocou **príkazov** programovacieho jazyka Logo, ktoré píšeme do **príkazového riadku**.



Obrázok 1: Prostredie Imagine

Základnými príkazmi pre korytnačku sú príkazy pre pohyb a otočenie:

dopredu číslo - korytnačka prejde **číslo** krokov (obrazových bodov) v smere, ktorým je natočená
vzad číslo - korytnačka prejde **číslo** krokov v opačnom smere, ako je natočená
vpravo uhol - korytnačka sa otočí o **uhol** (v stupňoch) vpravo
vľavo uhol - korytnačka sa otočí o **uhol** (v stupňoch) vľavo

Otáznik na začiatku postupnosti príkazov znamená, že ich treba napísať do príkazového riadku. Príkaz sa vykoná až po stlačení klávesu Enter.

Príkazy, ktoré obsahujú diakritiku, napr. **vľavo**, môžeme používať aj bez nej, teda **vľavo**.

Tip: Vyskúšajte za príkaz **do** napísať nejaké veľké číslo, napr. **do 1000**. Čo sa stane? Premýšľajte prečo.

So žiakmi môžeme na precvičovanie základných príkazov použiť projekt *Kreslenie jednoduchých obrázkov*. Môžete si ho vyskúšať - nájdete ho v systéme Moodle.

Použitie ukážeme na príklade:

```
? dopredu 40  
? vľavo 45  
? dopredu 30  
? vpravo 90  
? vzad 30
```



Niektoré korytnačie príkazy môžeme zadávať aj v skrátrenom tvare.

plný tvar	skrátенý tvar	plný tvar	skrátенý tvar
dopredu	do	vzad	vz
vpravo	vp	vľavo	vľ

Pred kreslením ďalšieho obrázka zmažeme stránku príkazom **znova**. Po vykonaní tohto príkazu bude korytnačka v strede stránky otočená hore.

Korytnačka pozná aj príkaz na kreslenie bodov:

bod číslo - korytnačka nakreslí kruh s priemerom **číslo** obrazových bodov

Napríklad:

? bod 40

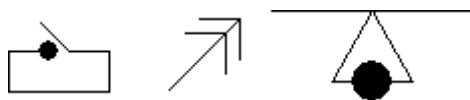


Do príkazového riadku môžeme naraz napísať aj viac príkazov, napr.

? znova vp 90 do 50 bod 20 do 60 bod 50 do 60 bod 20 do 50

Zadanie 1

Pomocou príkazov **do**, **vz**, **vp**, **v1** a **bod** nakreslite niektorý z obrázkov:



Pero, farba a hrúbka. Pomôcky

Korytnačím perom môžeme kresliť čiary rôznej hrúbky a farby. Štandardne je hrúbka pera korytnačky nastavená na 1 obrazový bod a farba pera na čiernu. Tieto nastavenia môžeme zmeniť príkazmi:

`nechHrúbkaPera hrúbka`, resp. `nechHp hrúbka`

`nechFarbaPera farba`, resp. `nechFp farba`

Parametrom príkazu `nechHrúbkaPera` je ľubovoľné prirodzené číslo. Vyskúšajme:

? `nechHp 3` do 40 `nechHp 10` do 40 `nechHp 20` do 40



Obrázok 2: Rôzne hrúbky pera

Parametrom príkazu `nechFarbaPera` je názov farby - reťazec začínajúci úvodzovkou. Vyskúšajme:

? `nechFp "červená do 100`



Obrázok 3: Základné farby

Uvedené mená farieb môžeme používať aj s číselnou príponou od 1 do 12, napríklad `červená7`. Farba s príponou 1 je najtmavšia, s príponou 12 najsvetlejšia.



Obrázok 4: Rôzne odtiene červenej farby

Farbu môžeme zadať aj **RGB kódom**, teda trojicou čísel z intervalu 0..255 v tvare [červená zelená modrá]. Napríklad:

? `nechFp [120 30 80] bod 50`

S akým parametrom musíme použiť príkaz `bod`, aby sme nakreslili kruh, ktorý pokryje celú stránku?

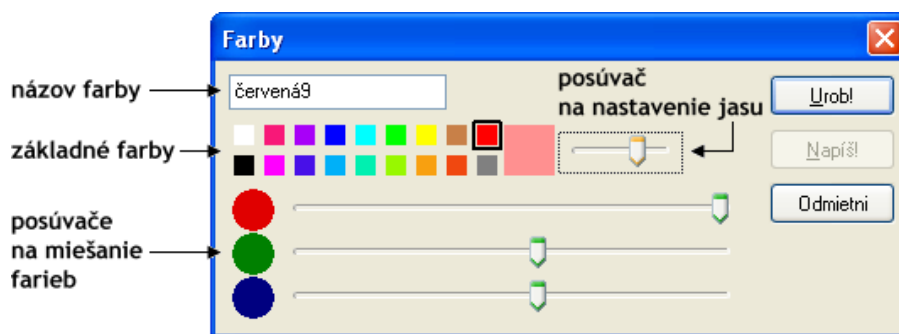
Všetky príkazy, ktoré sme korytnačke zadali, si Imagine pamätá (kým ho nevypneme). Šípkou hore a dolu môžeme listovať v tejto histórii príkazov - jednotlivé už použité príkazy z histórie sa zobrazujú v príkazovom riadku. Po stlačení klávesu **Enter** sa príkaz, ktorý je aktuálne v príkazovom riadku, vykoná. Môžeme si tak uľahčiť „prácu“, ak kreslíme obrázok, v ktorom sa nejaké príkazy alebo postupnosti príkazov opakujú.

Na veľkosti písmen v príkazoch v prostredí Imagine nezáleží. Je jedno, či napíšeme `nechFp`, `nechFp`, či `nechFP`, ide stále o ten istý príkaz. Rôzne veľkosti písmen v príkaze používame na zvýšenie čitateľnosti príkazu.

Názvy farieb môžeme rovnako ako príkazy písať s diakritikou, ale aj bez nej.


Názvy či kódy farieb si nemusíme pamätať. Ak napíšeme príkaz `nechFp` a stlačíme kláves **F9**, Imagine otvorí pomôcku na výber farby. Po nastavení farby stlačíme tlačidlo **Urob** - príkaz v príkazovom riadku sa doplní o farbu a vykoná.

Imagine poskytuje pomôcky aj pre príkazy `nechHp`, `do`, `vz`, `vp` a `v1`. Vyskúšajte ich!



Obrázok 5: Pomôcka na nastavenie farby pera korytnačky

Využitím rôznych farieb a hrúbok vieme nakresliť rôzne zaujímavé obrázky.

Zadanie 2	Nakreslime nasledujúci obrázok: 
Riešenie	Najskôr svetlozelenou farbou nakreslíme kruh: ? <code>nechFp "olivová10 bod 60</code> Potom doň bielou farbou nakreslíme hrubšiu čiaru: ? <code>nechFp "biela nechHp 10</code> ? <code>vp 45 do 30 vz 60</code>

Ak chceme kresliť nespojité obrázky alebo nakresliť viac obrázkov na stránke, potrebujeme presunúť korytnačku z jedného miesta na druhé bez kreslenia čiary. Použijeme dvojicu príkazov:

`peroHore`, resp. `ph` - vypne pero korytnačky
`peroDolu`, resp. `pd` - zapne pero korytnačky

Body kreslí korytnačka aj s vypnutým perom. Napríklad obrázok



môžeme nakresliť postupnosťou príkazov
`do 20 ph do 10`
`pd bod 5 ph`
`do 10 pd do 20`,
ale aj
`do 20 ph do 10`
`bod 5`
`do 10 pd do 20`


Vyskúšajme:

? `do 50 ph do 50 pd do 50`

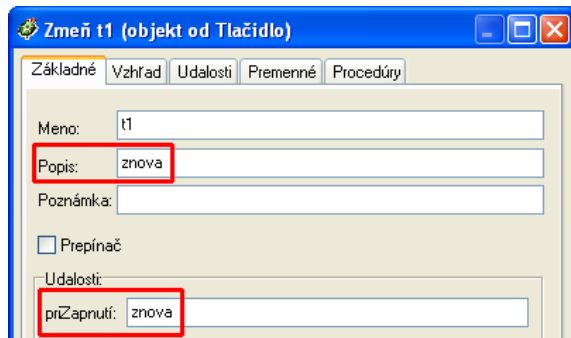
Zadanie 3	Pomocou korytnačky napíšte svoje iniciály.
Zadanie 4	Podľa vlastnej fantázie nakreslite niektorý z obrázkov: semafor, snehuliak, lienka, nejaká dopravná značka.

Tlačidlá. Prvý projekt

Pred kreslením každého obrázka sme zmazali stránku príkazom `znova`. Pohodnejšie by bolo, keby sme vedeli stránku zmazať napr. kliknutím na tlačidlo. Ukážeme si, ako vytvoriť vlastné tlačidlo, ktoré po každom stlačení vykoná príkaz `znova`.

1. V paneli nástrojov klikneme na nástroj **Nové tlačidlo** .
2. Klikneme myšou do stránky tam, kde chceme tlačidlo umiestniť.

- Nové tlačidlo je zatiaľ nefunkčné. Klikneme na nové tlačidlo pravým tlačidlom myši a z ponuky zvolíme **Zmeň t1**.
- V dialógovom okne nastavíme položku **priZapnutí** na **znova** - tým povieme, že vždy, keď sa toto tlačidlo zapne, má sa vykonať príkaz **znova**. Vhodne zmeníme aj položku **Popis**, ktorá definuje text zobrazený na tlačidle.



Obrázok 6: Rodný list tlačidla

- Stlačíme tlačidlo **OK**.

Ak nám nevyhovuje veľkosť tlačidla, klikneme naň pravým tlačidlom myši, zvolíme **Zmeň t1**, prejdeme na záložku **Vzhľad** a zmeníme hodnotu položiek **Šírka** a **Výška**.

Teraz už dokážeme vytvoriť jednoduchý projekt na kreslenie čiar a bodov.

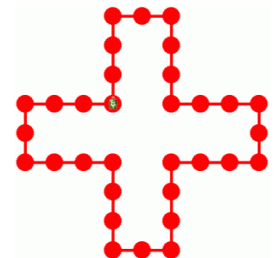
Dialógové okno na nastavenie vlastností tlačidla zvykneme tiež nazývať **rodný list** tlačidla. Podobné rodné listy majú aj iné objekty.


Veľkosť tlačidla môžeme zmeniť aj tak, že stlačíme kláves **Ctrl** a pravým tlačidlom myši ťaháme pravý dolný roh tlačidla.

Tlačidlo môžeme premiestňovať pravým tlačidlom myši so súčasne stlačeným klávesom **Shift**.

Ukladanie v Imagine:
Shift+F2 - Uložiť projekt
F2 - Uložiť projekt ako ...

Pomocou projektu **Kreslenie** nakreslite hraciu plochu, napríklad:



Zadanie 5	<p>Pridajte 4 nové tlačidlá, ktoré budú vykonávať príkazy:</p> <ul style="list-style-type: none"> • bod 50 • do 100 • vp 90 • vl 90 <p>Tlačidlám nastavte vhodné popisy a veľkosti a esteticky ich umiestnite. Projekt uložte pod názvom Kreslenie.</p>
Zadanie 6	<p>Pridajte do svojho projektu niekoľko tlačidiel na zmenu farby pera (zvoľte niekoľko základných farieb), hrúbky pera (niekoľko konkrétnych hrúbok), vypínanie a zapínanie pera, prípadne iné, ktoré považujete za užitočné.</p> <p>Aké tlačidlá treba doplniť do projektu Kreslenie, aby bolo možné vytvoriť aj nasledujúcu hraciu plochu?</p> 

Opakuj

Ako by ste pomocou korytnačky nakreslili nasledujúce obrázky? Čo majú spoločné?




Pravdepodobne sme si všetci uvedomili, že pri kreslení týchto obrázkov opakujeme viackrát tú istú postupnosť príkazov, na čo je vhodné použiť cyklus. Cyklus s pevným počtom opakovaní v jazyku Logo realizujeme príkazom **opakuj**.

`opakuj počet [priказ1 priказ2 ...]` - počet-krát sa vykoná postupnosť príkazov `[priказ1 priказ2 ...]`

Nakreslite na papier nejaký vlastný obrázok s opakujúcimi sa časťami. Dajte papier kolegovi a požiadajte ho, aby obrázok nakreslil v Imagine pomocou príkazu `opakuj`.

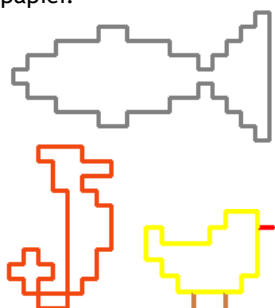
Ako motiváciu k cyklu môžete s deťmi vyskúšať projekt *Moja kamarátka ihlička*
<http://infovekacik.infovek.sk/2007-oktober/robot.php>

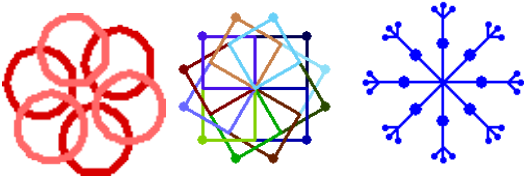

Zadanie 7	Nakreslite obrázky uvedené na začiatku časti Opakuj (štvorec, hviezdička a retiazka).
Riešenie	štvorec: <code>opakuj 4 [do 50 vp 90]</code> hviezdička: <code>opakuj 12 [do 20 vz 20 vp 30]</code> retiazka: <code>opakuj 5 [do 20 bod 10]</code>
Zadanie 8	Podobným spôsobom ako štvorec nakreslíme rovnostranný trojuholník, pravidelný päťuholník, či ďalšie pravidelné mnohouholníky. <ul style="list-style-type: none"> • O aký uhol musíme korytnačku otáčať pri kreslení pravidelného n-uholníka? • Nakreslite pravidelný 12-uholník so stranou dĺžky 10. Potom pravidelný 24-uholník s rovnakou stranou. Na čo sa podobá?
Zadanie 9	Pomocou príkazu <code>opakuj</code> nakreslite nasledujúci obrázok: 

V postupnosti príkazov v hranatých zátvorkách, ktoré chceme opakovať, môžeme znovu použiť príkaz `opakuj`. Ľahko tak nakreslíme napr. obrázky skladajúce sa z niekoľkých trojuholníkov:

? `opakuj 8 [opakuj 3 [do 30 vp 120] vp 45]`

Deti obľubujú kreslenie jednoduchých obrázkov pomocou dlhých „špagetových“ programov s jednoduchými príkazmi. Na skrátenie programu hľadajte v obrázku opakujúce sa časti. Pri programovaní použite ako pomôcku štvorcový papier.



Zadanie 10	Nakreslite niektorý z obrázkov. 
Zadanie 11	Dokončíte srdce tak, aby bolo súmerné podľa zvislej osi.  ? <code>v1 90 do 10 vp 90 do 10 v1 90 do 20 v1 90 do 10 vp 90 do 10 v1 90 do 20 v1 90 opakuj 3 [do 10 vp 90 do 10 v1 90] do 10</code>

Zistíte, ktoré iné príkazy môžeme použiť s parametrom ?.

V tomto prípade žrebuje ? hodnoty od 0 do 255.

Náhodné vstupy

Imagine umožňuje zadávať pre niektoré príkazy náhodné vstupy, napr. náhodnú hrúbku alebo náhodnú farbu. Príkazom `nechFp ?` nastavíme ľubovoľnú farbu pera, príkazom `nechHp ?` nastavíme ľubovoľnú hrúbku pera.

? `nechFp ? nechHp ? opakuj 5 [do 60 vp 144]`

Pri zadávaní farby RGB kódom môžeme dokonca povedať, ktorá zložka má byť náhodná, napr. `[0 0 ?]`, `[150 ? 30]`, `[? 50 ?]`. Vyskúšajme:

? `nechHp 20 opakuj 500 [nechFp [? 50 ?] do 20 vp ?]`

Ak potrebujeme vyberať iba z niekoľkých konkrétnych hodnôt, môžeme využiť operáciu `?prvok`. Operácia má jeden parameter - zoznam hodnôt (v hranatých zátvorkách). Výsledkom operácie je niektorá z týchto hodnôt. Vyskúšajme:

```
? do ?prvok [25 50 75 100] bod ?prvok [10 20 30 40]
? opakuj 500 [nechFp ?prvok [zelená3 zelená4 zelená5 zelená6]
do 20 vp ?prvok [0 90 180 270]]
```

Na generovanie náhodných čísel môžeme použiť aj operáciu `náhodne cislo`. Výsledkom operácie je celé číslo z intervalu $0..cislo-1$. Napr. `náhodne 10` vráti náhodné celé číslo z intervalu $0..9$. Pripočítaním či odpočítaním celého čísla tak dokážeme vygenerovať celé číslo z ľubovoľného intervalu.

V Logu rozlišujeme *príkazy* a *operácie*. Operácie na rozdiel od príkazov vracajú nejakú hodnotu.

Zadanie 12	Nastavme hrúbku pera na celé číslo z intervalu $\langle 5,15 \rangle$.
Riešenie	Vyberáme z jedenástich hodnôt $\{5, 6, \dots, 15\}$, preto použijeme operáciu <code>náhodne</code> so vstupom <code>11</code> , ktorá nám vráti náhodné celé číslo z intervalu $\langle 0, 10 \rangle$. Pripočítaním čísla 5 k tejto hodnote získame celé číslo z intervalu $\langle 5,15 \rangle$. Riešením teda je: <code>? nechHp 5 + nahodne 11</code>


Vypĺňanie obrázkov

S korytnačkou môžeme kresliť aj obrázky s výplňou, a to pomocou dvojice príkazov:

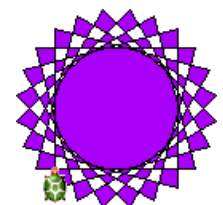
`nechFarbaVýplne farba`, resp. `nechFv farba` - nastaví farbu výplne, farba je zadaná názvom alebo RGB kódom
`vyplň` - korytnačka vyplní farbou výplne oblasť, v ktorej sa práve nachádza


Obrázky s výplňou kreslíme tak, že najskôr nakreslíme obrys, s **vypnutým perom** presunieme korytnačku niekam „dovnútra“ obrázka, nastavíme farbu výplne a použijeme príkaz `vyplň`.

Útvary s obrysom a výplňou môžeme kresliť aj pomocou príkazu `polygon`. Jeho vstupom je návod na kreslenie, ktorým popíšeme obrys polygónu. Obrys sa nakreslí aktuálnou farbou pera a hrúbkou pera korytnačky a je vyplnený aktuálnou farbou výplne. Ak obrys polygónu nie je uzavretý (posledný bod nie je identický s prvým), Imagine ho uzatvorí pridaním jednej koncovej čiary. Vyskúšajte:
`? nechFp "modrá`
`? nechFv "žltá`
`? polygon [opakuj 4`
`[do 50 vp 90]]`
alebo
`? nechfp "čierna`
`? nechFv "fialová`
`? polygon [opakuj 24`
`[do 100 vp 105]]`

Zadanie 13	Nakreslíme štvorec s modrým obrysom a žltou výplňou. 
Riešenie	<code>? nechHp 3 nechFp "modrá</code> <code>? opakuj 4 [do 50 vp 90]</code> <code>? ph vp 45 do 20</code> <code>? nechFv "žltá9 vyplň</code> prípadne ešte <code>? vz 20 vl 45 pd</code>

Pozor! Príkaz `nechFp` mení okrem farby pera aj farbu výplne. Ak by sme v uvedenom riešení zmenili poradie nastavenia farby pera a výplne na `nechFv "žltá nechFp "modrá`, potom nakreslili štvorec a nakoniec ho vyplnili, dostali by sme modrý štvorec s modrou výplňou.



Zadanie 14	Dokončíte program na kreslenie veterníka (použite cyklus): <code>? do 40 vl 135 do 28 vl 45 do 20 vl 90 do 20</code> <code>ph vl 135 do 20 vyplň</code> 
-------------------	--

Prečo je dĺžka jednej strany vrtule veterníka 28? Skúste nakresliť vrtulu aj pomocou príkazu `polygon`. Vyskúšajte rôzne farby a natočenia.

Dobrym námetom na kreslenie obrázkov v Logu sú skladačky origami.



Doplňujúce úlohy a záverečné zadanie k 1. kapitole nájdete v e-learningovom kurze v systéme Moodle.

Čo sme sa naučili

- kresliť pomocou korytnačky jednoduché obrázky z čiar a bodov (pomocou príkazov **do**, **vz**, **vp**, **vl**, **bod**)
- zmazať stránku príkazom **znova**
- meniť farbu a hrúbku pera korytnačky príkazmi **nechFp** a **nechHp**
- využívať pomôcky na zadávanie parametrov pre niektoré príkazy
- vytvárať vlastné tlačidlá a vytvoriť jednoduchý projekt
- kresliť nesúvislé obrázky pomocou príkazov **ph** a **pd**
- kresliť zložitejšie obrázky tak, že opakujeme rovnakú postupnosť príkazov pomocou príkazu **opakuj**
- používať náhodné vstupy, napr. farby a hrúbky pera korytnačky
- kresliť vyplnené obrázky pomocou príkazov **nechFv** a **vyplň**

Kapitola 2: Vlastné príkazy

Vlastné príkazy bez parametra

Vlastné príkazy (procedúry) definujeme v Imagine pomocou príkazu **uprav**:

```
uprav "meno príkazu" - otvorí editovacie okno na definovanie príkazu
meno príkazu
```


Definujeme príkaz **štvorec** na nakreslenie štvorca so stranou dĺžky 50.

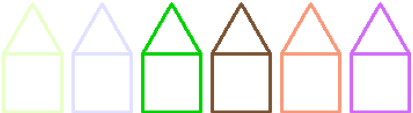
1. Zadáme ? **uprav "štvorec"**
2. V editovacom okne medzi riadky **viem štvorec** a **koniec** napíšeme telo príkazu.



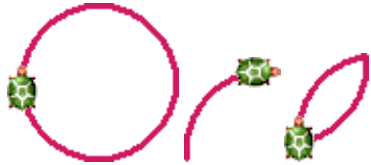

```
štvorec      viem štvorec
              opakuj 4 [do 50 vp 90]
              koniec
```

3. Stlačíme tlačidlo OK.
4. Vyskúšame: ? **štvorec**

Pri úprave existujúceho príkazu postupujeme rovnako.

Všetky definované príkazy môžeme vidieť v okne **Pamät'** pod položkou **Procedúry**. Okno **Pamät'** zobrazíme (aj skryjeme) stlačením tlačidla **Pamät'**  v paneli nástrojov, resp. klávesom F4.

Zadanie 1	Definujte príkaz trojuholník , ktorý nakreslí rovnostranný trojuholník so stranou dĺžky 50. Uložte projekt.
Zadanie 2	Definujte príkaz, ktorý nakreslí domček zo štvorca a trojuholníka. Z niekoľkých domčekov vytvorte ulicu. 
Riešenie	<pre>viem domček štvorec do 50 vp 30 trojuholník vl 30 vz 50 koniec</pre>

	<p>Je výhodné, keď sa korytnačka po nakreslení domčeka vráti tam, kde začala kresliť. Ľahšie sa nám tak nakreslí ulica.</p> <p>? opakuj 10 [nechFp ? domček vp 90 ph do 60 vl 90 pd]</p>
<p>Zadanie 3</p>	<p>Definujte príkaz vzor, ktorý nakreslí vzor aktuálnej farby pera:</p>  <p>S použitím príkazu vzor nakreslite pás vzorov náhodných farieb alebo dvojfarebný:</p> 
<p>Zadanie 4</p>	<p>Naučte korytnačku kresliť kružnicu a oblúk (štvrt'kružnicu). Z dvoch oblúkov vytvorte lupeň.</p> 
<p>Riešenie</p>	<pre>viem kruznica opakuj 24 [do 10 vp 15] koniec viem oblúk opakuj 6 [do 10 vp 15] koniec viem lupeň opakuj 2 [oblúk vp 90] koniec</pre>
<p>Zadanie 5</p>	<p>Definujte príkaz kvet, ktorý nakreslí kvet z 5 lupeňov. Definujte príkaz lúka, ktorý rozsype kvety náhodnej farby po stránke.</p> 

Nezabudnite si priebežne ukladať svoj projekt, aby korytnačka nové príkazy nezabudla.

Obrázky môžeme kombinovať do pásových vzorov a obdĺžnikových rámkov, ktoré sa dajú použiť na rôzne účely (plagáty, pozvánky, vizitky a iné).



Príkaz **lupeň** môžete zmeniť tak, aby bol lupeň vyplnený.

Námet: Na kreslenie obrázkov z čiar a oblúkov môžete použiť projekt *Algor*. Zo systému Moodle si stiahnite súbor *Algor.zip*, rozbaľte ho a spustíte projekt *Algor.imp*. Pozrite si úvodný tutoriál. V priečinku *zadania* otvorte súbor *zadania.htm* a pokúste sa nakresliť niektorý zo zadaných obrázkov.

Vlastné príkazy s parametrom

Príkazy, ktoré sme definovali v predchádzajúcej časti, kreslili obrázky s konštantnou veľkosťou - rovnako veľké štvorce, rovnako veľké trojuholníky, domy, kvety. V tejto časti si ukážeme, ako v jazyku Logo definovať príkaz s parametrom. Budeme pokračovať v ostatnom projekte. Jednotlivým príkazom pridáme vhodné parametre.

Do príkazu **štvorec** doplníme parameter, ktorý bude určovať dĺžku jeho strany.

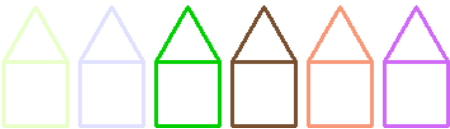
1. Zadáme ? **uprav "štvorec**
2. Do hlavičky príkazu za meno príkazu pridáme parameter **:dĺžka**
3. V tele príkazu použijeme parameter **:dĺžka** ako vstup pre príkaz **do**

Definovanie vlastných príkazov využívajúcich vlastné príkazy je náplňou projektu *Domčeky pre korytnačku*. Môžete si ho stiahnuť zo systému Moodle.

```
viem štvorec :dĺžka
opakuj 4 [do :dĺžka up 90]
koniec
```

4. Použijeme príkaz - tentokrát už s číselným vstupom:
 ? štvorec 30
 ? štvorec 100

- pred meno parametra píšeme dvojbodku, teda `:dĺžka`, `:strana`
- pri viacslavných menách parametrov použijeme na oddelenie slov napr. bodku `:počet.lupeňov`, podčiarkovník `:počet_lupeňov` alebo rôznu veľkosť písmen `:početLupeňov`, nie však medzeru


Zadanie 6	Upravte príkaz <code>trojuholník</code> , aby kreslil trojuholníky s premennou dĺžkou. Podobne príkaz <code>domček</code> , <code>oblúk</code> , <code>lupeň</code> a <code>kvet</code> (pri posledných troch bude parametrom dĺžka strany oblúka).
Zadanie 7	Definujte príkaz <code>ulica</code> s parametrom <code>:počet</code> , ktorý nakreslí <code>:počet</code> domčekov vedľa seba. 

Chceme nakresliť štvorce s postupne zvyšujúcou sa dĺžkou strany 10, 20, ..., 100. Môžeme napísať `? štvorec 10 štvorec 20 ... štvorec 100`, ale to by nebolo pekné riešenie. Využijeme cyklus `opakuj` a operáciu `počítadlo`.

`počítadlo`, resp. `poč` - operácia, ktorá nám povie, ktorá iterácia cyklu práve beží. Má zmysel len v cykloch.

Pozor! `počítadlo` nie je parameter, preto sa pred ním nepíše dvojbodka!

```
? opakuj 10 [štvorec 10*počítadlo]
```

Zadanie 8	Nakreslite obrázky podobné nasledujúcim. Využite cyklus s počítadlom a definované príkazy. 
------------------	--

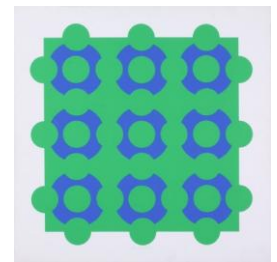
Ak chceme definovať príkaz s viacerými parametrami, uvedieme mená parametrov za názov príkazu oddelené medzerou. Upravíme príkaz `kvet` tak, aby okrem veľkosti lupeňa mal aj premenný počet lupeňov.

```
viem kvet :dĺžka :počet
opakuj :počet [lupeň :dĺžka up 360/:počet]
koniec
```

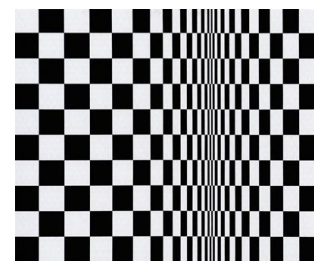
Zadanie 9	Upravte príkaz <code>lúka</code> tak, aby nakreslil daný počet kvetov náhodnej farby s dĺžkou lupeňa náhodne vybranou z hodnôt 3, 5, 7, 9 a s náhodným počtom lupeňov od 5 do 12.
------------------	---

<p>Riešenie</p>	<pre>viem lúka :počet opakuj :počet [nechFp ? kvet ?prvok [3 5 7 9] 5 + náhodne 8 ph vp ? do ? pd] koniec</pre>
<p>Zadanie 10</p>	<p>Upravte príkaz domček tak, aby dostával 2 parametre: dĺžku strany štvorca a dĺžku strany trojuholníka. Trojuholník by mal byť vzhľadom na štvorec vycentrovaný.</p> <p>Upravte príkaz ulica tak, aby dostával parametre :počet a :vzd a nakreslil daný počet domčekov vzdialených od seba o :vzd bodov. Domčeky nech majú základ rozmeru 30, 40 alebo 50 a strechu rozmeru od 30 do 70.</p> 
<p>Zadanie 11</p>	<p>Definujte príkaz(y), ktorý nakreslí nasledujúci motív.</p>  <p>Príklad je prevzatý z [2].</p>
<p>Riešenie</p>	<p>Odmyslíme si zatiaľ farby a pozrime sa, z čoho sa obrázok skladá. V obrázku sa často opakuje vzor:</p>  <p>Vzor je vytvorený zo štvorca a dvoch trojuholníkov. Definujeme príkaz, ktorý ho nakreslí.</p> <pre>viem vzor :d štvorec :d opakuj 2 [trojuholník :d opakuj 2 [do :d vp 90]] koniec</pre> <p>Tento vzor je po štvoricich zoskupený do väčšieho vzoru, v ktorom sú jednotlivé štvorce postupne otočené o 90°:</p>  <pre>viem vzorVäčší :d opakuj 4 [vzor :d do 2*:d vp 90] koniec</pre> <p>A napokon náš motív je zložený zo 4 postupne otočených väčších vzorov.</p>

Porozmýšľajte, ako by sa dali nakresliť obrázky podobné dielam umelcov op-artu Victora Vasarelyho a Bridget Rileyho.



Victor Vasarely



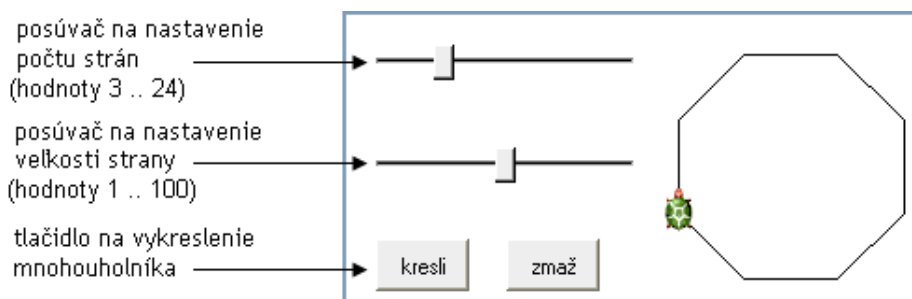
Bridget Riley

```
viem motív :d
  opakuj 4 [vzorVäčší :d do 4*:d vp 90]
koniec
```

Ak do príkazu vzor na začiatok doplníme zmenu farby pomocou RGB kódu s nejakou náhodnou zložkou, dostaneme pekný farebný motív.

Projekt Mnohouholníky

Vyvineme projekt, ktorý umožní kreslenie mnohouholníkov. Načrtneme si ho.




Obrázok 7: Návrh projektu Mnohouholníky

Okrem uvedených prvkov budeme potrebovať príkaz, ktorý nakreslí mnohouholník.

Zadanie 12

Definujte príkaz `poly` s parametrami `:n` a `:strana`, ktorý nakreslí pravidelný `n`-uholník s dĺžkou strany `:strana`. Otestujte príkaz s rôznymi parametrami.

Pridáme na stránku posúvač pre nastavenie počtu strán. V paneli nástrojov zvolíme nástroj **Nová súčiastka** , z ponuky vyberieme **Vodorovný posúvač** a umiestnime ho na stránku.

Nastavíme vlastnosti posúvača. Klikneme naň pravým tlačidlom myši a zvolíme **Zmeň p1**. Nastavíme položku **Minimum** na **3** a **Maximum** na **24**. V záložke **Vzhľad** môžeme prípadne upraviť rozmery posúvača, či mu nastaviť priesvitné pozadie.

Zadanie 13

Podobným spôsobom vytvorte druhý posúvač pre nastavenie veľkosti strany.

Pridajte na stránku tlačidlo, nastavte mu položku **Popis** na **Kresli** a primeranú veľkosť.


Pridajte tlačidlo na zmazanie stránky.

Reakciou na udalosť **priZapnutí** tlačidla na kreslenie bude volanie príkazu `poly`. Parametrami príkazu budú aktuálne hodnoty posúvačov.

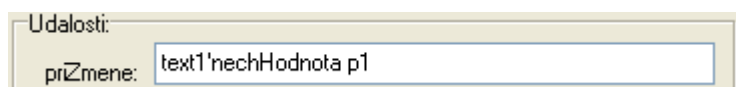
Na hodnotu posúvača sa odvolávame jeho menom. T.j. `p1` je hodnota posúvača `p1`, `p2` je hodnota posúvača `p2`.

Pri zapnutí tlačidla teda vykonáme príkaz `poly p1 p2`.

Projekt už zodpovedá nášmu návrhu, skúsme ho však ešte vylepšiť. Jedným z nedostatkov projektu je, že pri zmene posúvača nevidíme, akú hodnotu sme nastavili. Pridáme preto do projektu textové popisy, ktoré budú zobrazovať aktuálne hodnoty posúvačov.

V paneli nástrojov zvolíme nástroj **Nový text** . Myšou natiahneme obdĺžnik pri posúvači **p1** - tým vytvoríme editovateľné textové pole. Napíšeme doň ľubovoľné číslo a zavrieme okno **Štýl písma**. Objekt, ktorý sme vytvorili, sa volá **text1**.

Teraz potrebujeme „naprogramovať“, že vždy, keď sa zmení hodnota posúvača **p1**, má sa v textovom poličku **text1** zobrazit' hodnota posúvača **p1**. Ide vlastne o nastavenie reakcie na udalosť **prizmene** posúvača **p1**. V rodnom liste posúvača **p1** nastavíme udalosť **prizmene** nasledovne:



Rodný list zatvoríme a vyskúšame ťahať posúvačom **p1**.

V dialógovom okne **Štýl písma** môžeme zmeniť nastavenia písma - oplatí sa aspoň zväčšiť veľkosť.

Meno ľubovoľného objektu v Imagine zistíme tak, že naň klikneme pravým tlačidlom myši.

Zadanie 14

Pridajte druhé textové poličko a vhodne nastavte udalosť **prizmene** pre posúvač **p2**.

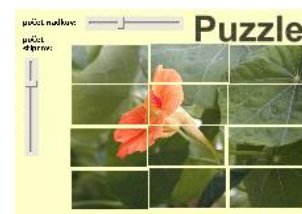
Uloženie projektu pre web

Prostredie Imagine nám umožňuje ukladanie projektov pre web, t.j. vytvorenie webovej stránky s „odkazom“ na špeciálnu verziu projektu, ktorá je spustiteľná vo webovom prehliadači. Po zverejnení tejto dvojice súborov (html súbor a Imagine projekt s príponou iip) na internete môže náš projekt vyskúšať každý návštevník našej stránky. Podmienkou je mať nainštalovaný tzv. „Imagine plugin“. Ak plugin nie je nainštalovaný, stránka projektu ponúkne možnosť jeho stiahnutia.

Projekt uložíme pre web zvolením voľby **Uložiť pre web...** v ponuke **Súbor**. V dialógovom okne **Ulož pre web** zadáme meno výstupného súboru (odporúčame v samostatnom priečinku, keďže sa vytvorí viac súborov). Ďalej môžeme zadať nadpis projektu na výslednej www stránke, prípadne nejaký komentár.

Nezabudnime ešte oba súbory premiestniť do webového miesta a urobiť odkaz na novú stránku.

Takto zverejnené projekty nájdete napr. aj na stránkach <http://imagine.infovek.sk>



Ak vieme tvoriť www stránky, môžeme si časť zdrojového kódu odkazujúcu na projekt z html súboru umiestniť priamo do svojej stránky.

Zadanie 15

Uložte projekt Mnohouholníky pre web.

Čo sme sa naučili

- definovať príkaz bez parametrov i s parametrami
- využívať operáciu **počítadlo** v cykle **opakuje**
- skladať príkazy do zložitejších stavieb
- vytvárať a používať posúvače a textové objekty pri vývoji projektov
- sprístupniť projekty vytvorené v Imagine na webe

Doplňujúce úlohy ku kapitole 2 a záverečné zadanie z 2. kapitoly nájdete v e-learningovom kurze v systéme Moodle.

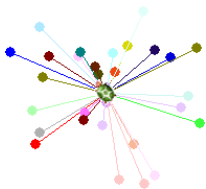
Kapitola 3: Udalosti korytnačky. Viac korytnačiek

Informácie o korytnačke, udalosti korytnačky

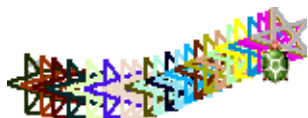
Podobne ako tlačidlo a posúvač, aj korytnačka má svoj rodný list. V jednotlivých záložkách rodného listu môžeme zistiť a meniť rôzne nastavenia korytnačky, či definovať jej reakcie na rôzne udalosti, napr. čo sa má stať, keď na korytnačku klikneme myšou, keď ju pustíme, keď na ňu dvojklikneme, atď.

Definujeme reakciu korytnačky na kliknutie myšou. V záložke **Základné** rodného listu vyplníme položku **príKliknutí** takto:

Plnú verziu 3. kapitoly nájdete v e-learningovom kurze v systéme Moodle. Nájdete v nej ďalšie zaujímavé efekty pri ťahaní korytnačky myšou, vývoj projektu **Labyrint a Človeče**.



`nechPozícia [x y]` - presunie korytnačku na pozíciu [x y].



`čakaj číslo` - pozdrží vykonávanie príkazov na `číslo` milisekúnd

`farbaBodu` - operácia, ktorej výsledkom je farba v tom bode stránky, v ktorom stojí korytnačka

`ak logický_výraz [príkaz1 príkaz2 ...]` - ak je výsledkom logického výrazu hodnota `áno`, vykonajú sa príkazy v zozname

`ySur` - y-ová súradnica aktuálnej pozície korytnačky. Podobne `xSur`.



nástroj *Nová korytnačka*

Plnú verziu 4. kapitoly nájdete v e-learningovom kurze v systéme Moodle.



priKliknutí: `vp ? nechFp ? do ? bod 10 nechPozícia [0 0]`

Zatvorme rodný list a klikneme niekoľkokrát na korytnačku.

Stiahnime a otvorime si projekt `udalosti.imp` zo systému Moodle. Sú v ňom definované príkazy z 2. kapitoly a niektoré ďalšie - pozrite si ich v okne **Pamäť**. V rodnom liste korytnačky prejdime na záložku **Tvar** a zapneme nastavenie **Automatické ťahanie**. Zatvorme rodný list a ťahajme korytnačku ľavým tlačidlom myši. Ak má korytnačka zapnuté pero, tak môžeme takýmto spôsobom kresliť podobne ako nástrojom ceruzka v grafickom editore.

Pri ťahaní môžeme korytnačkou kresliť aj rôzne zaujímavé efekty. V rodnom liste zvolíme záložku **Udalosti** a klikneme na tlačidlo **Pridaj**. Zo zoznamu udalostí vyberieme udalosť **priŤahání** a stlačíme **OK**. Udalosť **priŤahání** zadefinujeme takto:

priŤahání: `nechFp ? hviezda 30`

Po zatvorení rodného listu ťaháme korytnačku. Skúste aj:

? `nechFarbaPozadia "tmavoModrá4`

priŤahání: `nechFp ?prvok [žltá5 žltá6 žltá7 žltá8 žltá9] pd hviezda 30 ph čakaj 100`

Korytnačka sa rozhoduje

Pokračujme v práci so súborom `udalosti.imp`. Nástrojom **Ukáž/Skry panel kreslenia**



zobrazíme panel s nástrojmi na kreslenie. Nakreslíme jednoduché pozadie z dvoch farieb - stačí, ak v hornej polovici stránky nakreslíme obdĺžnik napr. farbou `zelená10`. Na bielej časti stránky chceme pri ťahaní korytnačky kresliť domčeky, na zelenej stromčeky. Pomôže nám operácia `farbaBodu` a podmienený príkaz `ak`.

Udalosť **priŤahání** definujeme takto:

`ak farbaBodu = "biela [dom 30 + náhodne 20]`
`ak farbaBodu = "zelená [strom]`
`čakaj 100`



Rozhodnúť sa môže aj podľa y-ovej súradnice korytnačky.

priŤahání: `ak2 ySur<0 [dom 30 + náhodne 20] [strom] čakaj 200`

Viac korytnačiek

Prostredie *Imagine* nám dovoľuje pracovať s viacerými korytnačkami. Novú korytnačku pridáme tak, že v paneli nástrojov zvolíme nástroj **Nová korytnačka** a klikneme myšou na stránku. Nová korytnačka má tiež svoj rodný list, v ktorom môžeme meniť jej nastavenia a definovať udalosti. Zatiaľ jej však nedokážeme dávať príkazy - naučíme sa to v nasledujúcej kapitole.

Kapitola 4: Tvar korytnačky. Procesy

Tvar korytnačky

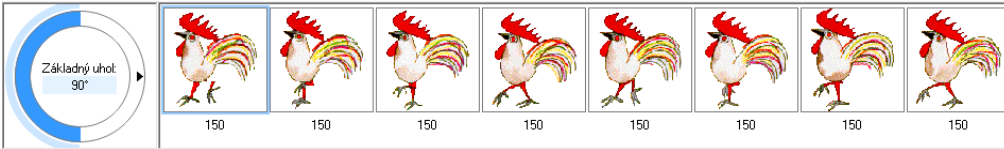
Jedna z vlastností korytnačky, ktorú môžeme meniť, je tvar. Urobíme tak v rodnom liste v záložke **Základné** kliknutím na tlačidlo **Tvar**. V ponuke si vyberieme nejaký obrázok, napr. súbor `kohut.lgf`.

Vytvorme druhú korytnačku a zmeňme jej tvar na obrázok zo súboru `Kohut kraca.lgf`. Rozdiel medzi týmto kohútom a predchádzajúcim je v tom, že kým prvý kohút je statický, druhý kohút sa pohybuje - jeho tvar tvorí animovaný obrázok.



Obrázok 8: Fázy animovaného obrázka

Nastavme druhému kohútovi v rodnom liste smer na 270 (na východ). Všimnime si, že teraz sa kohút zobrazuje inými fázami.



Obrázok 9: Fázy animovaného obrázka pre opačný smer

Pozrime si obrázok kráčajúceho kohúta v grafickom editore Logomotion. Vidíme, že obsahuje dve sady fáz pre dva rôzne smery korytnačky. Skupine obrázkov, ktoré sa zobrazujú pre určitý smer, hovoríme **záber**.

Tvar korytnačky v Imagine si teda môžeme predstaviť ako "dvojrzmerné pole" záberov a fáz. Jednotlivé zábery sa zväčša používajú pre rôzne natočenia korytnačky, fázy v rámci záberov slúžia na animáciu.

Živé obrazy

Otvorme si nový projekt. Nastavme stránke pozadie zo súboru **krajina.bmp** v priečinku **pozadia**. Vytvorme korytnačky s tvarmi zo súborov **auto** a **bicyklista**, v rodnom liste im nastavme automatické ťahanie, smer 90 a vypnime pero.

Obraz oživíme tým, že necháme auto a cyklistu pohybovať sa smerom vpravo. Rozhýbeme ich pomocou príkazu: `? každých 100 [do 10]`. Spustíme tak proces, ktorý každých 100 milisekúnd posunie korytnačku dopredu o 10 krokov.

Pohybuje sa však len jedna korytnačka. Ako rozhýbeme druhú? Ak máme na stránke viac korytnačiek, musíme korytnačku, ktorá má vykonať príkaz, **osloviť**, t.j. uviesť meno korytnačky, apostrof a príkaz, ktorý má vykonať. Napríklad príkazom `? k2' do 100` prikážeme korytnačke **k2** ísť dopredu o 100 krokov.

Zastavme bežiaci proces tlačidlom **Zastaviť všetky procesy**  v paneli nástrojov a spustíme dva nové procesy:

```
? k1'každých 100 [do 10]
```

```
? k2'každých 200 [do 10 v1 30]
```

Príkazy, ktoré napíšeme do príkazového riadka, sa neukladajú. Preto je vhodné definovať vlastný príkaz, napríklad **hýbteSa**, ktorým budeme môcť kedykoľvek rozbehnúť všetky procesy v našom živom obraze.

```
viem hýbteSa
k1'kazdych 100 [do 10]
k2'kazdych 200 [do 10]
k2'kazdych 1000 [v1 ?]
koniec
```

Ak proces pri spúšťaní pomenujeme, môžeme ho potom samostatne zastaviť.

```
? (k2'každých 1000 [v1 ?] "točenie)
```

```
? zastav "točenie
```



Logomotion je grafický editor patriaci k Imagine. Tvar korytnačky zobrazíme v Logomotion tak, že v rodnom liste korytnačky v záložke **Tvar** klikneme na obrázok s jej tvarom.



Zábery tvaru korytnačky nemusia vždy vyjadrovať jej natočenie.

Pozadie stránky zmeníme voľbou **Pozadie zo súboru...** z kontextového menu, ktoré vyvoláme kliknutím pravým tlačidlom myši na stránku.



Proces je „malý neviditeľný motorček, ktorý neustále vykonáva nejakú činnosť.“ [1].

každých číslo [príkaz1 príkaz2 ...] - spustí proces, ktorý každých číslo milisekúnd vykoná inštrukcie príkaz1 príkaz2...

V e-learningovom kurze v systéme Moodle vás ešte v 4. kapitole čaká: klonovanie korytnačiek a vývoj projektov Kocky, Písmenká, Labyrint a Človeče.

Čo sme sa naučili v tomto module

Zhrnutie

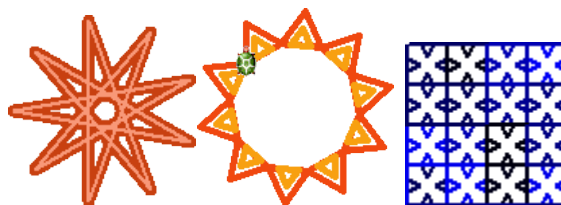
V tomto module sme sa naučili:

- používať základné príkazy korytnačej grafiky a cyklus `opakuj`
- zadávať náhodné vstupy pre rôzne príkazy
- definovať a používať vlastné príkazy
- definovať reakciu korytnačky na rôzne udalosti
- používať podmienené príkazy `ak` a `ak2`
- zistiť hodnoty niektorých nastavení korytnačky
- vytvárať nové korytnačky a meniť im nastavenia v rodnom liste
- zmeniť korytnačke tvar, rôzne využívať zábery obrázka
- spúšťať a zastavovať procesy
- vytvárať jednoduché projekty, využívať v nich korytnačky, tlačidlá, posúvače, textové objekty, udalosti korytnačky, procesy

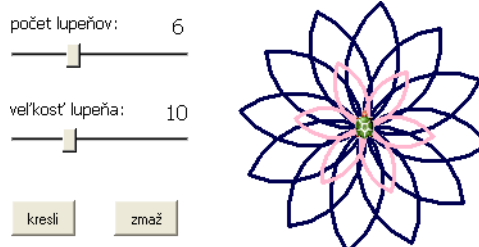
Preverenie výstupných vedomostí

Po absolvovaní modulu by ste mali vedieť:

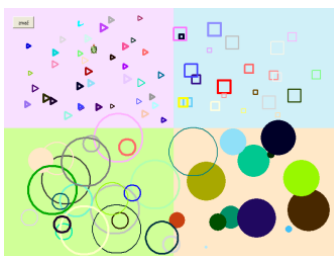
- definovať príkazy, ktoré pomocou základných príkazov pre korytnačku a cyklu nakreslia rôzne obrázky a skladať ich do „stavebníc príkazov“, napr. (obrázky sú prevzaté z [2])



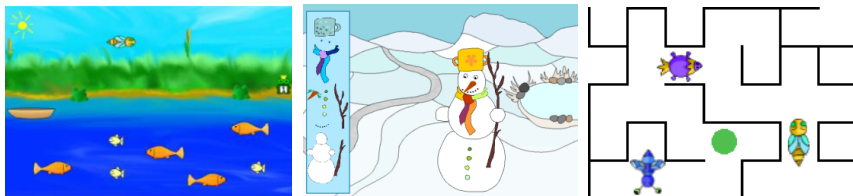
- využívať tlačidlá, textové objekty a posúvače, napr.



- kresliť korytnačkou pri ťahaní myšou rôzne obrázky v závislosti od pozície korytnačky alebo farby pozadia, napr.



- vytvoriť jednoduché projekty, napr.
 - živé obrazy - kompozície z pohybujúcich sa korytnačiek rôznych tvarov umiestnených na vhodnom pozadí, napr. rybník s plávajúcimi rybami, lietajúcimi včelami, žabkami, ktoré pri kliknutí poskočia,
 - rôzne skladačky či stavebnice - projekty, v ktorých z malých dielikov môžeme postaviť väčší útvar, napr. vytvorenie snehuliaka z rôznych typov očí, nosov, hrncov, atď.,
 - zjednodušené verzie niektorých hier s „postavičkami“ pohybujúcimi sa po hracej ploche prípadne reagujúcimi na zrážku s iným objektom, napr. zjednodušené Človeče či Pacman.



Pacman - hra, v ktorej šípkami meníme smer žltého „papkáčika“ pohybujúceho sa v labyrinte tak, aby pojedol všetky bodky a vyhýbal sa duchom. Môžete si ju zahrať na www.freepacman.org.

Literatúra a použité zdroje

- [1] Blaho, A., Kalaš, I. (2005) *Tvorivá informatika 1. zošit z programovania*. Bratislava: SPN, 2005, 48 strán. ISBN 80-10-00019-1
- [2] Hrušecká A., Kalaš I. *Programovanie v prostredí Imagine*, Bratislava: Metodicko-pedagogické centrum
- [3] Ondrišková J. *Dištančný kurz programovanie Imagine - diplomová práca*
- [4] imagine.infovek.sk
- [5] Blaho, A., Kalaš, I. (1998) *Comenius Logo: Tvorivá informatika (1. časť)*, Bratislava: CL Group, 1998, 157 strán. ISBN 80-967999-0-8
- [6] Papert, S. (1999) Introduction: What is Logo? And Who Needs it? In *Logo Philosophy and Implementation* [online]. Logo Computer Systems Inc, 1999 [cit. 16. jún 2009], s. IV-XVI. Dostupné na internete: <http://www.microworlds.com/company/philosophy.pdf>.
- [7] Blaho, A., Kalaš I., Tomcsányi, P. (1993) *Comenius Logo: Environment for Teachers and Environment for Learners*. In *Proceedings of 4th European Logo Conference Supplement: Logo-like Learning Environmnets: Reflection & Prospects*. Athens: Doukas School S.A., 1993. ISBN 960-7203-01-1, s. suppl. 1-11.
- [8] Blaho, A., Kalaš I., Tomcsányi, P. (2000) *Imagine ... Nová generácia tvorivého prostredia pre výučbu*. In *POŠKOLE 2000: Sborník semináře*. Praha: Ediční středisko ČVUT, 2000. ISBN 80-01-02172-6, s. 88 - 93.
- [9] Blaho, A., Kalaš I., Tomcsányi, P. (1995) *Niekoľko slov o Comenius Logu*. In *Informatické listy*, 2005, roč. 1, č. 1. Bratislava: Katedra vyučovania informatiky MFF UK, 1995, s. 7-8.
- [10] infovekacik.infovek.sk

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

Autori © PaedDr. Daniela Bezáková
RNDr. Gabriela Lovászová, PhD.
Mgr. Peter Kučera

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

Podnázov Programovanie 1

Študijný materiál prešiel recenzným pokračovaním.

Recenzenti PaedDr. Ladislav Huraj, PhD.
PaedDr. Viera Palmárová, PhD.

Počet strán 20

Náklad 400 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 978-80-89225-65-1