

Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika

Úvod do programovania

Predmet: Úvod do programovania

Línia: Vlastný odborový kontext informatiky a informatickej výchovy



Úvod do programovania

Identifikácia modulu

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

Línia aktivity: Vlastný odborový kontext informatiky a informatickej výchovy

Predmet: Úvod do programovania

Zaradenie modulu

Modul Úvod do programovania je jediným modulom predmetu Úvod do programovania v línii Informatika. Účastníci vzdelávania ho absolvujú v prvom semestri svojho vzdelávania.



Abstrakt modulu

Cieľom modulu je pomocou programovacieho jazyka Logo pripraviť účastníkov vzdelávania na programovanie, oboznámiť ich s niektorými základnými algoritmickými pojmami (príkaz, procedúra, parameter, cyklus, udalosť), ukázať im, že pri programovaní používame obmedzenú množinu príkazov, ktorú ale vieme rozširovať o vlastné príkazy, rozvíjať tvorivosť a kognitívne schopnosti.

Účastník modulu sa naučí používať základné príkazy korytnačej grafiky, používať cyklus s pevným počtom opakovaní na kreslenie obrázkov s opakujúcimi sa časťami, vytvoriť vlastné príkazy bez parametrov aj s parametrami, skladať príkazy do zložitejších „stavebníc“, rozložiť si problém na menšie časti. Oboznámi sa tiež s niektorými modernými technikami pri programovaní (využívanie udalostí, procesov) a vytvorí niekoľko jednoduchých vlastných projektov.

Modul bude pozostávať z tematických jednotiek:

- základy korytnačej grafiky,
- vlastné príkazy,
- udalosti korytnačky, viac korytnačiek,
- živé obrázky.

Na vzdelávanie budeme využívať programovací jazyk Logo, prostredie Imagine, verziu so slovenskými príkazmi.

Garant predmetu:

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

Autori:

PaedDr. Daniela Bezáková,
KZVI FMFI UK
PaedDr. Viera Palmárová,
PhD., KI FPV UKF
Mgr. Katarína Pilátová, KIS
FRI ŽU

Rukopis odovzdaný:

31. máj 2009

Obsah

Úvod do programovania	1
Identifikácia modulu	1
Zaradenie modulu.....	1
Abstrakt modulu.....	1
Obsah.....	2
Úvod	3
Vstupné vedomosti.....	3
Požadované prerekvizity	3
Predpokladané vstupné vedomosti, skúsenosti a zručnosti.....	3
O študijnom materiáli.....	4
Študijný materiál	5
Základy korytnačej grafiky	5
Vlastné príkazy	14
Udalosti korytnačky, viac korytnačiek.....	21
Živé obrázky	25
Zoznam príkazov	28
Čo sme sa naučili v tomto module	30
Preverenie výstupných vedomostí.....	30
Literatúra a použité zdroje	31

Úvod

Programovanie tvorí integrálnu časť súčasných osnov informatiky pre základné a stredné školy. Kvalifikovaný učiteľ informatiky pracuje s počítačom nielen ako používateľ pripravených aplikácií - v ideálnom prípade sám dokáže vytvárať aplikácie na podporu učenia sa u svojich žiakov.

Programovací jazyk Logo vznikol pre deti ako nástroj pre hlbšie porozumenie matematiky. Jeho tvorca, Seymour Papert, ho nazýva objektom, **s ktorým sa rozmýšľa**. Pri tvorbe prvých programov v jazyku Logo si osvojíte základné algoritmické pojmy - (príkaz, procedúra, parameter, cyklus, udalosť, proces). Hoci programovací jazyk pozostáva z obmedzenej množiny príkazov, naučíte sa tieto príkazy kombinovať do vlastných príkazov s novou funkcionalitou.

Jazyk Logo využijete na rozvoj vlastných **metakognitívnych schopností**. Budete **vytvárať** nové príkazy, **čítať** príkazy, **analyzovať** ich a rozširovať, **hľadať** chyby a opravovať ich.

Spoznáte viaceré projekty z učebnice *Tvorivá informatika 1. zošit z programovania* [1]. Vyskúšate si viaceré projekty prostredia Imagine umiestnené na internete, ktoré môžete využívať so svojimi žiakmi.

Programovanie v ľubovoľnom programovacom jazyku vám poskytuje jedinečný priestor pre rozvoj **schopnosti riešiť problémy**. Nezabúdajte, že problémy nie sú vždy jednoduché a najviac si obvykle ceníme tie, ktoré nás stáli najviac úsilia. Schopnosť vytvárať počítačové programy vám dáva do rúk silný nástroj pre podporu vlastného vyučovania. Hlboká porozumenie programátorským konceptom vám pomôže využiť programovací jazyk ako objekt pre rozvoj myslenia detí.

Vstupné vedomosti

Požadované prerekvizity

Pre účasť na tomto module sa vyžaduje úspešné absolvovanie modulu Základná digitálna gramotnosť.

Predpokladané vstupné vedomosti, skúsenosti a zručnosti

Predpokladáme, že ako účastník vzdelávania modulu Úvod do programovania

- ovládajte základy práce so súbormi a priečkami, viete dekomprimovať archívny súbor, spustiť súbor či aplikáciu umiestnenú v danom priečinku,
- viete používať prehliadač, ukladať súbory z internetu do počítača.

O študijnom materiáli

Študijný materiál modulu Úvod do programovania tvorí päť kapitol:

- Základy korytnačej grafiky
- Vlastné príkazy
- Udalosti korytnačky, viac korytnáčiek
- Živé obrázky
- Zoznam príkazov

V prvých štyroch kapitolách študijný materiál kombinuje výklad, riešené ukážkové príklady, neriešené úlohy a úlohy na sebatestovanie. Úlohy označené ako *Úloha* by ste mali riešiť v takom poradí, v akom sú uvedené v materiáli. Úlohy označené ako *Vyskúšajte sa* môžete buď riešiť postupne tak, ako sú zaradené v materiáli, alebo vyriešiť naraz všetky tie, ktoré sa nachádzajú v danej kapitole. Úspešné vyriešenie úloh *Vyskúšajte sa* v rámci jednej kapitoly by malo byť pre vás spätnou väzbou, že ste predchádzajúcu časť materiálu zvládli a môžete prejsť na ďalšiu. Na konci každej kapitoly tiež nájdete množinu tzv. rozširujúcich úloh. Tieto úlohy sú určené len ako doplnujúci materiál na ďalšiu inšpiráciu či precvičovanie. Ich vyriešenie nie je podmienkou na prechod do ďalšej kapitoly.

V piatej kapitole nájdete zhrnuté všetky príkazy, ktoré si v predchádzajúcich kapitolách osvojíte, spolu so stručným popisom.

Pri práci s týmto študijným materiálom budeme využívať prostredie Imagine. Inštalčné CD je súčasťou učebnice *Tvorivá informatika 1. zošit z programovania* [1]. Budeme tiež využívať pripravené projekty z tohto CD a projekty z internetu, najmä z [4]. Pri práci s týmito projektmi je nutné mať v prehliadači nainštalovaný Imagine plugin - jeho inštalácia je súčasťou inštalácie prostredia Imagine na spomínanom CD.

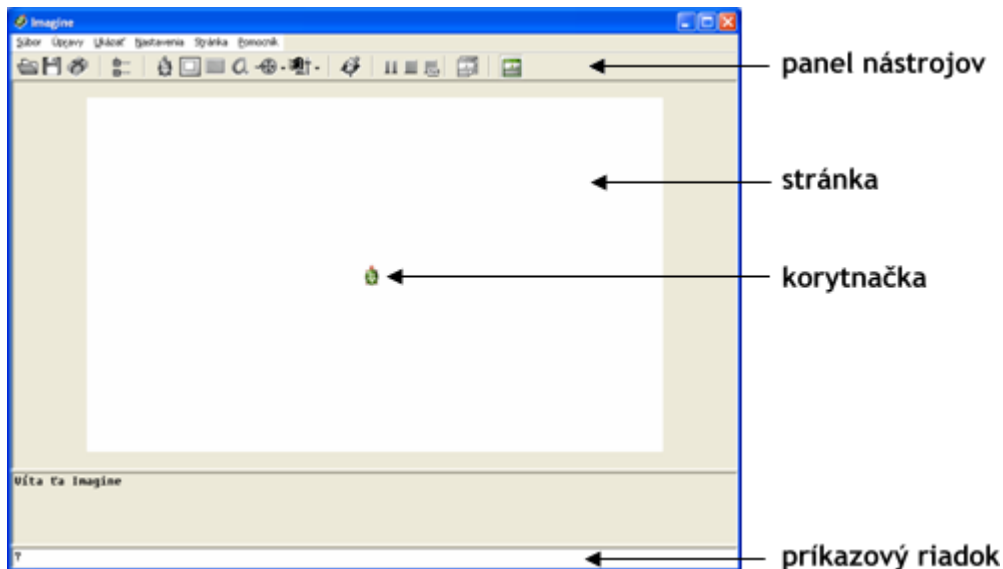


Studijný materiál

Základy korytnačej grafiky

Imagine a korytnačka. Základné príkazy

Hlavnou postavou prostredia Imagine je **korytnačka**. Je to malý kresliaci robot žijúci na **stránke**, ktorý robí to, čo mu povieme. S korytnačkou sa rozprávame pomocou **príkazov** programovacieho jazyka Logo, ktoré píšeme do **príkazového riadka**. Korytnačka rozumie len istej množine príkazov.



Obrázok 1: Prostredie Imagine

Na začiatok sa naučíme niekoľko grafických príkazov, pomocou ktorých dokážeme s korytnačkou nakresliť rôzne obrázky. Do príkazového riadka napíšeme príkaz

? dopredu 100

a stlačíme **Enter**. Korytnačka sa posunula o 100 krokov dopredu a nakreslila pritom čiaru dlhú 100 obrazových bodov.

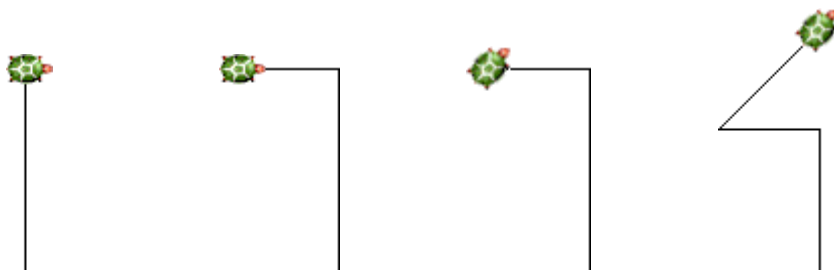
Pokračujme príkazmi:

? vpravo 90

? vzad 50

? vľavo 45

? dopredu 40



Máme za sebou prvý obrázok. Skôr ako nakreslíme ďalší, zmažeme stránku príkazom ? znova. Po vykonaní tohto príkazu bude korytnačka v strede stránky otočená hore.

Niektoré korytnačie príkazy môžeme zadávať aj v skrátenej tvare, napr. namiesto príkazu dopredu 100 môžeme písať len do 100. V tabuľke nájdete skratky pre všetky príkazy, ktoré sme sa doteraz naučili.

Príkaz znova nemá skrátenej tvar.

plný tvar	skrátenej tvar
dopredu	do
vzad	vz
vpravo	vp
vľavo	vl

Otáznik na začiatku postupnosti príkazov znamená, že ich treba napísať do príkazového riadku.

Po každom príkaze nezabudneme stlačiť **Enter**, inak korytnačka príkaz nevykoná.

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

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

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.

Namiesto **parameter** hovoríme niekedy aj **vstup** alebo **premenná**.

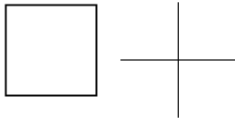
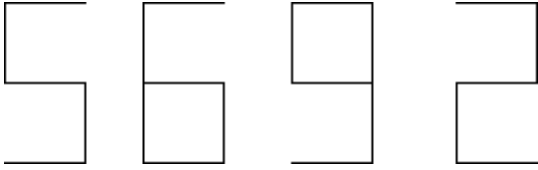
Poznáte nejaký príkaz, ktorý nemá parameter?

Zmažeme stránku príkazom `? znova`.

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

`? do 120 v1 30 do 40 vz 40 vp 60 do 40 vz 40`

Po stlačení klávesu **Enter** ich korytnačka jeden po druhom vykoná.

Úloha 1	V systéme Moodle spustíte projekt <i>Doved' korytnačku do domčeka</i> . Riad'te korytnačku príkazmi <code>do</code> , <code>vz</code> , <code>vp</code> a <code>v1</code> a pomôžte jej dostať sa do domčeka.
Úloha 2	Pomocou príkazov <code>do</code> , <code>vz</code> , <code>vp</code> a <code>v1</code> nakreslite obrázky: 
Vyskúšajte sa 1	Prirad'te číslce k postupnostiam príkazov, ktoré ich nakreslia.  <ol style="list-style-type: none"> <code>v1 90 do 40 vp 90 do 40 vp 90 do 40 v1 90 do 40 v1 90 do 40</code> <code>vp 90 do 40 v1 90 do 40 v1 90 do 40 vp 90 do 40 vp 90 do 40</code> <code>vp 90 do 40 v1 90 do 40 v1 90 do 40 vp 90 vz 40 do 80 vp 90 do 40</code> <code>vp 90 do 40 v1 90 do 80 v1 90 do 40 v1 90 do 40 v1 90 do 40 v1 90 do 40</code>

Naučili sme sa príkazy `dopredu`, `vzad`, `vpravo`, `vľavo`. Všimnime si, že za každým z týchto príkazov sme museli uviesť číslo. Pri príkazoch `dopredu` a `vzad` toto číslo znamenalo počet krokov, o ktoré sa má korytnačka posunúť dopredu či vzad, pri príkazoch `vľavo` a `vpravo` to bola veľkosť uhla, o ktorý sa má otočiť vľavo či vpravo. Vo všeobecnosti toto číslo nazývame **parameter**. Hovoríme, že parameterom príkazu `dopredu` a `vzad` je počet krokov, parameterom príkazu `vpravo` a `vľavo` je uhol. Neskôr uvidíme, že parameterom príkazu môže byť aj iný ako číselný údaj.

Hrúbka pera a farba pera, pomôcky

Korytnačím perom môžeme kresliť čiary rôznej hrúbky. Zatiaľ sme kreslili čiary hrúbky jeden obrazový bod. Hrúbku pera korytnačky zmeníme príkazom `nechHrúbkaPera`. Parameterom príkazu `nechHrúbkaPera` je číslo, ktoré vyjadruje hrúbku kreslenej čiary počtom obrazových bodov. Napríklad hrúbku 5 nastavíme príkazom `nechHrúbkaPera 5`, alebo skrátene `nechHp 5`. Odteraz budú všetky čiary, ktoré korytnačka nakreslí, hrúbky 5 - až dotedy, kým jej príkazom `nechHp` nezmeníme hrúbku na inú.

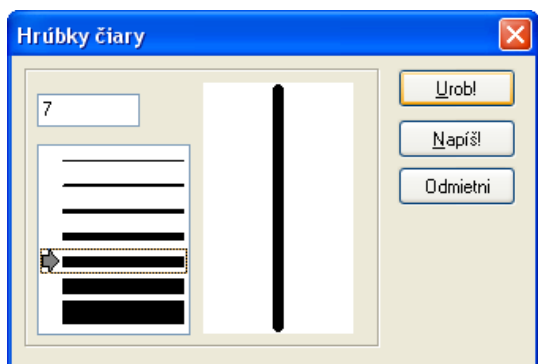


Obrázok 2: Rôzne hrúbky pera

Vyskúšajte nasledujúcu postupnosť príkazov:

`? nechHp 5 do 50`
`? nechHp 10 do 50`
`? nechHp 15 do 50`

Ak nevieme, akú hrúbku zadať, môžeme využiť pomôcku na výber hrúbky pera - napíšeme len `nechHr` a stlačíme kláves `F9`. V otvorenom okne zvolíme vhodnú hrúbku a stlačíme tlačidlo `Urob`.



Obrázok 3: Pomôcka na výber hrúbky pera korytnačky

Tip: Zvoľte veľmi veľkú hrúbku pera (napr. 1000) a posuňte korytnačku o 1 krok dopredu.

Úloha 3

Využitím rôznych hrúbok pera nakreslite nasledujúci obrázok:



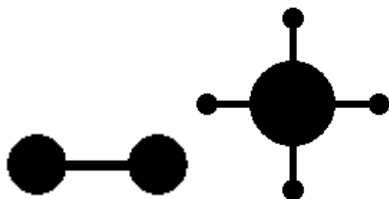
Korytnačka vie okrem čiar kresliť aj bodky. Po zadaní príkazu `bod 30` nakreslí korytnačka bod (kruh) s priemerom 30 obrazových bodov. Parametrom príkazu `bod` je teda priemer bodu. Pomocou príkazu `bod` nakreslíme lízanku:

? `nechHr 3 do 50 bod 30`



Úloha 4

Využitím čiar rôznych hrúbok a bodov rôznej veľkosti nakreslite nasledujúce obrázky:



Vyskúšajte sa 2

Vykonaním postupnosti príkazov
 ? `do 40 vp 120 do 80 vl 120 do 40 vp 60 vz 40 bod 30`
`vz 40`
 korytnačka nakreslí obrázok. Zistite ktorý.



Vrátíme sa k obrázku lízanka. Naša lízanka je nepekná, čierna. Okrem hrúbky môžeme peru korytnačky nastaviť aj farbu. Príkazom `nechFarbaPera "zelená"`, alebo skrátene `nechFp "zelená"` zmeníme farbu pera korytnačky na zelenú. Všetky čiar a bodky bude odteraz korytnačka kresliť zelenou farbou, až kým jej znovu nezmeníme farbu pera. Vyskúšajte napr.

? `bod 30 do 100.`

Vymyslite a nakreslite vlastný obrázok, ktorý využíva aspoň 3 rôzne hrúbky pera.

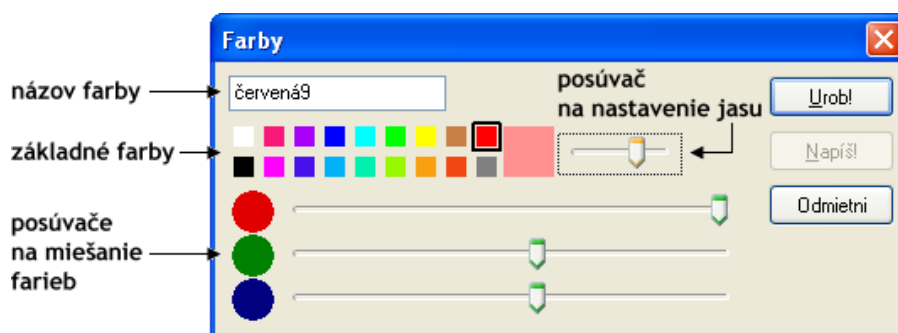
Základné farby, ktoré môžeme používať ako parameter príkazu `nechFp`, sú:

čierna		biela	
ružová		purpurová	
tmavomodrá		fialová	
belasá		modrá	
tmavozelená		azúrová	
olivová		zelená	
oranžová		žltá	
tehlová		hnedá	
sivá		červená	

Obrázok 4: Základné farby

Ak si želáme inú farbu, môžeme využiť pomôcku **Farby**. Napíšeme `nechFp` a stlačíme kláves **F9**. V pomôcke **Farby** si môžeme zvoliť niektorú zo základných farieb a zmeniť posúvačom jej jas (odtieň) alebo si „namiešať“ vlastnú farbu pomocou posúvačov pre červenú, modrú a zelenú zložku. Po nastavení farby nezapudnime stlačiť tlačidlo **Urob**. V príkazovom riadku si všimnite, ako sa po použití pomôcky doplnil príkaz `nechFp`.

Nezabúdajme pred názov farby písať úvodzovku. Farbu pera môžeme zadávať nielen názvom, ale aj ako trojicu celých čísel [červená zelená modrá], tzv. RGB kódom, napríklad `nechFp [0 70 200]`. Všetky tri čísla musia byť celé medzi 0 a 255. Pred kód farby úvodzovku nedávame.

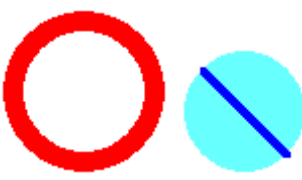


Obrázok 5: Pomôcka na zvolenie farby pera korytnačky

Teraz už vieme nakresliť farebnú lízanku:

```
? nechFp "cervena
? nechHp 3
? do 50
? nechFp "ruzova5
? bod 30
```



Úloha 5	Pomocou doteraz známych príkazov nakreslite nasledujúce obrázky:
	
Úloha 6	Otvorte projekt 1pero (Súbor, Otvoriť projekt) a skúste pomocou poskytnutých tlačidiel vytvoriť zaujímavé obrázky.

Vyskúšajte sa 3

Ktorá z uvedených postupností príkazov nakreslí nasledujúci obrázok?



1. `nechFp "oranžová bod 15 vp 45 do 60 bod 15 vz 30`
`nechFp "modrá3 bod 30`
2. `nechFp "oranžová bod 30 vp 45 do 60 bod 30 vz 30`
`nechFp "modrá3 bod 15`
3. `nechFp "oranžová bod 15 vp 45 do 30 vp 90 do 30`
`bod 15 vz 30 nechFp "modrá3 bod 30`
4. `nechFp "oranžová bod 30 vp 45 do 30 vp 90 do 30`
`bod 30 vz 30 nechFp "modrá3 bod 15`

K príkazom `nechFp` a `nechHp` sme mohli využiť pomôcku - stačilo zadať príkaz a stlačiť kláves **F9**. Zistíte, či podobné pomôcky poskytuje Imagine aj pre príkazy `do`, `vz`, `v1`, `vp`.

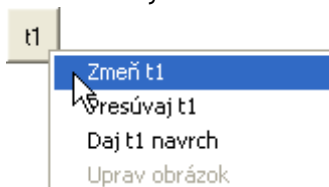
Vlastné tlačidlo, vytvárame prvý projekt

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

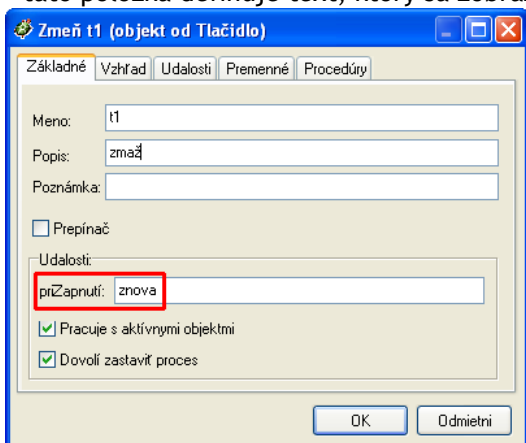
1. V paneli nástrojov klikneme na nástroj **Nové tlačidlo**.



2. Klikneme myšou do stránky tam, kde chceme naše tlačidlo umiestniť.
3. Nové tlačidlo je zatiaľ nefunkčné. Klikneme na nové tlačidlo pravým tlačidlom myši a zvolíme **Zmeň t1**.



4. V dialógovom okne nastavíme položku **priZapnutí** na `znova` - tým povieme, že vždy, keď sa toto tlačidlo zapne (t.j. keď naň klikneme myšou), má sa vykonať príkaz `znova`. Môžeme zmeniť aj položku **Popis** napr. na text `zmaž` - táto položka definuje text, ktorý sa zobrazí na tlačidle.



5. Stlačíme tlačidlo **OK**.
6. Ak nám nevyhovuje veľkosť tlačidla, klikneme naň pravým tlačidlom myši, zvolíme **Zmeň t1**, prejdeme na záložku **Vzhľad** a zmeníme hodnotu

položiek **Šírka** a **Výška** **Veľkosť:** Šírka Výška

Keď už vieme vytvárať tlačidlá, môžeme si vytvoriť svoj prvý projekt - projekt na korytnačie kreslenie (pozri nasledujúcu Úlohu 1).

Úloha 7

Vytvorte 5 tlačidiel, ktoré budú vykonávať tieto príkazy:

Námet: Pridajte do svojho projektu na korytnačie kreslenie tlačidlá na zmenu farby a hrúbky pera.

- bod 50,
- do 100,
- vp 45,
- v1 45,
- znova.

Po vytvorení tlačidiel projekt uložte pod názvom **Kreslenie (Súbor, Uložiť projekt)**.

Náhodné vstupy

Za príkazom `nechHp` sme uvádzali konkrétne číslo, ktoré vyjadruje hrúbku pera korytnačky, napr. `nechHp 4`. Za príkazom `nechFp` sme zase uvádzali konkrétnu farbu, napr. `nechFp "modrá"`.

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

Otvoríme nový projekt. Nakreslíme terč z kruhov náhodných farieb:

- ? `nechFp ?` bod 70
- ? `nechFp ?` bod 60
- ? `nechFp ?` bod 50
- ? `nechFp ?` bod 40
- ? `nechFp ?` bod 30



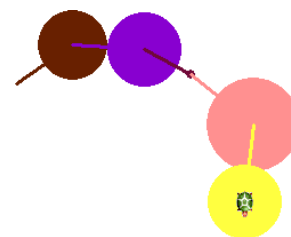
Zistite, či aj pre príkazy `do`, `vz`, `vp`, `v1` a `bod` môžeme použiť náhodné vstupy.

Môžeme farebný terč nakresliť aj takouto postupnosťou príkazov?

- ? `nechFp ?` bod 30
- ? `nechFp ?` bod 40
- ? `nechFp ?` bod 50
- ? `nechFp ?` bod 60
- ? `nechFp ?` bod 70

Náhodné vstupy môžeme použiť napríklad na nakreslenie rôzne veľkých rôznofarebných bodov „rozhádzaných“ na stránke.

- Nastavíme ľubovoľnú farbu: ? `nechFp ?`
- Otočíme korytnačku o ľubovoľný uhol: ? `vp ?`
- Posunieme korytnačku dopredu o ľubovoľnú dĺžku: ? `do ?`
- Nakreslíme ľubovoľne veľký bod: ? `bod ?`



Celú postupnosť zapíšeme do jedného riadku:

- ? `nechFp ? vp ? do ? bod ?`

Využitím histórie príkazov zopakujeme túto postupnosť niekoľkokrát.

Úloha 8

S využitím príkazov s náhodnými vstupnými hodnotami nakreslite obrázok:



Pero korytnačky (ph, pd)

Nakreslime obrázok:





Nakresliť krátke I a dĺžeň by sme už nepochybne vedeli. Ako však dosiahnuť, aby sme medzi dĺžňom a písmenom nemali čiaru? Ako presunúť korytnačku z jedného miesta na druhé bez kreslenia čiar? Použijeme dvojicu príkazov `peroHore` a `peroDolu`.

Príkazom `peroHore`, resp. `ph` vypneme pero korytnačky, príkazom `peroDolu`, resp. `pd` zapneme pero korytnačky.

Naše písmeno Í teda nakreslíme postupnosťou príkazov:

- ? `do 50`
- ? `ph do 10 pd`
- ? `vp 30 do 10`

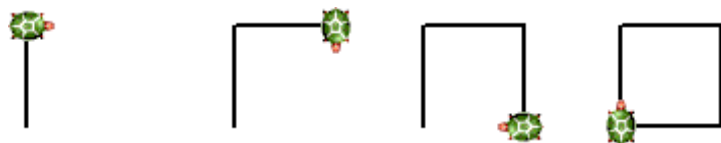
Niekedy sa nám môže zdať, že nás korytnačka „neposlúcha“ - nekreslí čiary, ktoré jej kážeme. Často to však býva len tým, že sme jej vypli pero a zabudli ho späť zapnúť, alebo sme nastavili farbu pera na rovnakú ako je farba pozadia (najčastejšie biela).

Úloha 9	Nakreslite nasledujúci obrázok: 
Vyskúšajte sa 4	<p>Čo treba opraviť alebo doplniť v nasledujúcej skupine príkazov, aby nakreslila nasledujúci obrázok?</p> <pre>nechHp 3 nechFp "belasa do 50 vp 90 do 50 vp 90 do 50 vp 90 do 50 vp 90 do 50 vp 30 nechFp "cervena5 do 50 vp 120 do 50 vp 120 do 50 vp 120</pre> <div style="display: flex; align-items: center; justify-content: flex-end;">  </div>

Poznámka: Korytnačka kreslí body aj s vypnutým perom. Vyskúšajte: ? ph bod 50

Príkaz opakuj

Pozrime sa, ako sa kreslí štvorec.



do 50 vp 90 do 50 vp 90 do 50 vp 90 do 50 vp 90

Pri kreslení štvorca vlastne štyrikrát opakujeme tie isté príkazy: do 50 vp 90. Môžeme to zapísať aj jednoduchšie, pomocou príkazu opakuj 4 [do 50 vp 90].

Pomocou príkazu opakuj vieme zopakovať ľubovoľnú postupnosť príkazov toľkokrát, koľko chceme. Príkaz opakuj má dva parametre: prvým je počet opakovaní, druhým je príkaz alebo postupnosť príkazov, ktoré chceme opakovať - musí byť uzavretá v hranatých zátvorkách!




Opakovaniu postupnosti príkazov hovoríme aj **cyklus**. Cyklus je programová konštrukcia, do ktorej vložíme nejaké príkazy a oznámime, koľkokrát sa budú opakovať.[5]

Vyskúšajme tieto príkazy:

? opakuj 8 [do 50 vz 50 vp 45]

? vp 30 opakuj 10 [do 50 vp 120 do 50 v1 120]

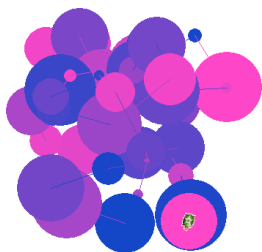
Úloha 10	Pomocou príkazu opakuj nakreslite nasledujúce obrázky: 
Úloha 11	Otvorte projekt 2Opakuj. Kartyčky s príkazmi presúvajte medzi hranaté zátvorky. Klikaním na šípku v kartyčke môžete zmeniť hodnotu parametra príslušného príkazu. Nakreslite nejaké pekné obrázky.

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

Spomeňme si na príklad s mnohými rôzne veľkými rôznofarebnými bodmi rozhádzanými na stránke. Jednu bodku ľubovoľnej veľkosti na ľubovoľnom mieste sme nakreslili príkazmi: nechFp ? vp ? do ? bod ?. Päť takýchto bodiek sme nakreslili tak, že sme uvedenú postupnosť zopakovali päťkrát.

- ? nechFp ? vp ? do ? bod ?
- ? nechFp ? vp ? do ? bod ?
- ? nechFp ? vp ? do ? bod ?
- ? nechFp ? vp ? do ? bod ?
- ? nechFp ? vp ? do ? bod ?

Tip: Ak farbu pera zadávame trojicou čísel, napr. [0 70 200], môžeme namiesto hociktorého z týchto troch čísel použiť otáznik. V tomto prípade otáznik bude znamenať ľubovoľné celé číslo medzi 0 a 255. Vyskúšajte v príklade s bodkami použiť na nastavenie farby namiesto príkazu `nechFp ?` príkaz `nechFp [? 70 200]`.

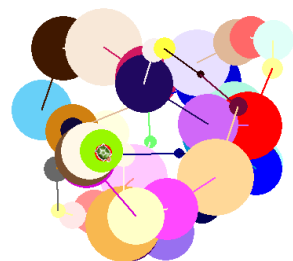


To isté dosiahneme príkazom

? opakuj 5 [nechFp ? vp ? do ? bod ?].

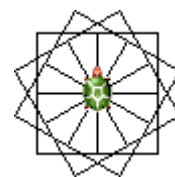
Ak chceme nakresliť 50 bodiek, použijeme

? opakuj 50 [nechFp ? vp ? do ? bod ?].



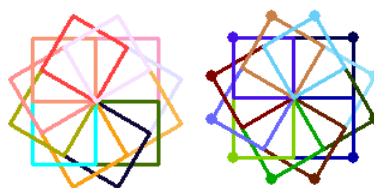
V postupnosti príkazov v hranatých zátvorkách, ktoré chceme opakovať, môžeme znovu použiť príkaz `opakuj`. Ľahko tak nakreslíme napr. obrázky skladajúce sa z niekoľkých štvorcov:

? opakuj 12 [opakuj 4 [do 30 vp 90] vp 30]



Úloha 12

S využitím príkazov s náhodnými vstupnými hodnotami a príkazu `opakuj` nakreslite nasledujúce obrázky:



Vyskúšajte sa 5

Priradte každý obrázok k postupnosti príkazov, ktorá ho nakreslí.



1. opakuj 6 [do 20 v1 30 opakuj 3 [do 40 vp 120] vp 30 vz 20 vp 60]
2. opakuj 6 [do 20 v1 45 opakuj 4 [do 30 vp 90] vp 45 vz 20 vp 60]
3. opakuj 6 [v1 30 opakuj 3 [do 40 vp 120] vp 30 vp 60]
4. opakuj 12 [v1 30 opakuj 3 [do 40 vp 120] vp 30 vp 30]

Rozširujúce úlohy

Úloha 13

Pomocou príkazov `do`, `vz`, `vp` a `v1` nakreslite nasledujúce obrázky:

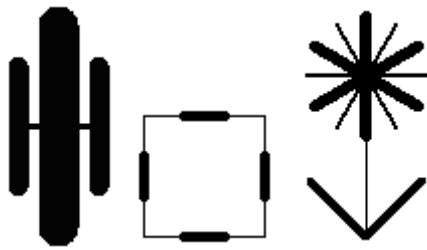


Úloha 14

V systéme Moodle spustíte projekt *Kreslenie jednoduchých obrázkov*. Prečítajte si zadanie a vyriešte niekoľko jednoduchých a niekoľko zložitejších úloh.

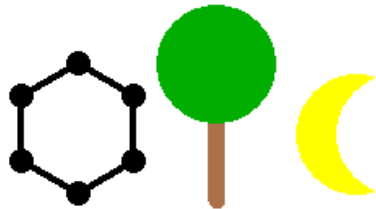
Úloha 15

Využitím rôznych hrúbok pera nakreslite nasledujúce obrázky:



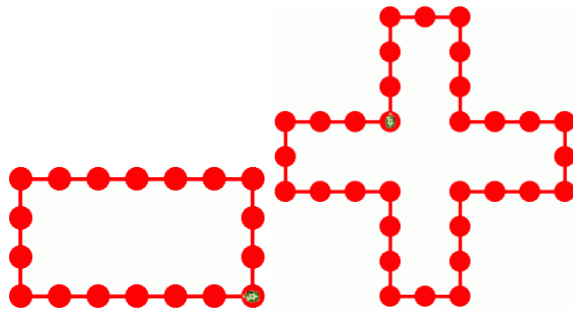
Úloha 16

Využitím čiar rôznych hrúbok a farieb, či bodov rôznej veľkosti nakreslite nasledujúce obrázky:



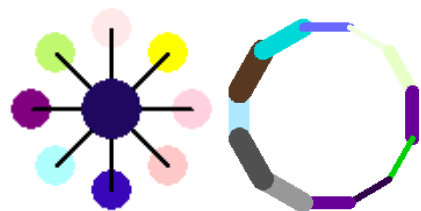
Úloha 17

Otvorte vami vytvorený projekt **Kreslenie** a pomocou tlačidiel nakreslite hracie plochy pre hru Človeče, nehnevaj sa napr.



Úloha 18

S využitím príkazov s náhodnými vstupnými hodnotami nakreslite podobné obrázky:



Úloha 19

Využitie príkazy `pd` a `ph` a nakreslite nasledujúce obrázky:



Úloha 20

Pomocou príkazu `opakuuj` nakreslite nasledujúce obrázky:

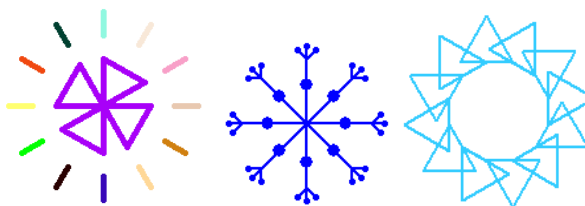


Úloha 21

V systéme Moodle spustite projekt *Moja kamarátka ihlička*. Prečítajte si návod a pokúste sa vyšíť aspoň 3 obrázky rôznej náročnosti.

Úloha 22

Využitím príkazu **opakuje** nakreslite nasledujúce obrázky:



Čo sme sa naučili

- rozprávať sa s korytnačkou pomocou príkazov
- presúvať korytnačku príkazmi **dopredu** a **vzad** o zadaný počet krokov a otáčať ju príkazmi **vľavo** a **vpravo** o zadaný uhol
- zmazať stránku príkazom **znova** a kresliť bodky príkazom **bod**
- 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
- používať náhodné vstupy, napr. farby a hrúbky pera korytnačky
- kresliť nesúvislé obrázky pomocou príkazov **ph** a **pd**
- kresliť zložitejšie obrázky tak, že opakujeme rovnakú postupnosť príkazov pomocou príkazu **opakuje**
- čo je to **parameter** a **cyklus**

Vlastné príkazy

Naučme korytnačku vlastný príkaz

Stromček môžeme nakresliť napr. nasledujúcimi príkazmi:

```
? nechFp "hneda
? nechHp 10
? do 60
? nechFp "zelena4
? bod 50
```



Ak chceme nakresliť vedľa prvého stromčeka ďalší, musíme korytnačku najskôr presunúť a potom zopakovať tých istých 5 príkazov. Keby korytnačka poznala príkaz stromček, bolo by to jednoduchšie.

Korytnačka síce príkaz stromček nepozná, ale môžeme ju ho naučiť. Do príkazového riadka napíšeme

```
? uprav "stromcek.
```

Otvorí sa okno na definovanie príkazov. Medzi zobrazené riadky **viem stromcek** a **koniec** napíšeme postupnosť príkazov, ktorými sa kreslí stromček. Po dopísaní stlačíme tlačidlo **OK**.

stromcek	viem stromcek nechfp "hneda nechhp 10 do 60 nechfp "zelena4 bod 50 koniec
----------	---

Odteraz už korytnačka pozná príkaz **stromcek**, môžeme ho teda používať rovnako, ako ostatné príkazy.

```
? stromcek
```

Nakreslime druhý stromček vedľa prvého:

- Po dokreslení prvého stromčeka zostala korytnačka v korune. Preto ju najskôr so zdvihnutým perom presunieme dolu o dĺžku kmeňa: ? ph vz 60.
- Potom ju posunieme o kúsok doprava: ? vp 90 do 100.
- Nakoniec ju natočíme späť na sever a zapneme jej pero: ? vl 90 pd.
- Potom znovu zadáme príkaz ? stromcek.

Nevýhodou je, že sme sa pri presúvaní museli vracat' o dĺžku kmeňa. Možno sme už medzitým zabudli, aký dlhý kmeň bol. Preto by bolo rozumnejšie, aby súčasťou príkazu `stromcek` bolo aj presunutie korytnačky na miesto, kde stromček začala kresliť. Upravíme teda príkaz `stromcek` - znovu do príkazového riadku napíšeme ? uprav "stromcek a doplníme príkaz `stromcek`.

```
stromcek
viem stromcek
  nechfp "hneda
  nechhp 10
  do 60
  nechfp "zelena4
  bod 50
  ph
  vz 60
  pd
  koniec
```

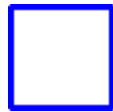
Často je rozumné, aby korytnačka skončila svoju kresbu tam, kde začala.

Teraz už ľahko nakreslíme aj celý sad.

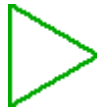
? opakuj 8 [stromcek ph vp 90 do 40 vl 90 pd]

Naučíme korytnačku kresliť štvorec, trojuholník a hviezdičku.

viem stvorec
opakuj 4 [do 50 vp 90]
koniec



viem trojuholnik
opakuj 3 [do 50 vp 120]
koniec



viem hviezda
opakuj 5 [do 50 vp 144]
koniec



Trojuholník kreslíme príkazom opakuj 3 [do 50 vp 120], štvorec kreslíme príkazom opakuj 4 [do 50 vp 90], päťuholník kreslíme príkazom opakuj 5 [do 50 vp 72]. Akým príkazom nakreslíme pravidelný

- 6-uholník?
- 10-uholník?
- n-uholník?

Pomôcka: všimnite si, že súčin uhla otočenia a počtu opakovaní je vždy rovný 360.

Uložíme projekt pod názvom `vlastnePríkazy`.

Úloha 1

Definujte vlastný príkaz na nakreslenie niektorého z obrázkov:



Z vlastných príkazov môžeme budovať ďalšie a ďalšie príkazy.

Naučme korytnačku kresliť domček. Nakreslíme ho tak, aby po skončení kreslenia bola korytnačka na tom mieste, kde začala kresliť.

```
viem domcek
  stvorec
  do 50
  vp 30
  trojuholnik
  vl 30
  vz 50
  koniec
```



Úloha 2

Zo systému Moodle si stiahnite projekt *Domčeky pre našu korytnačku*. Spustite ho a postupujte podľa zadania - definujte príkaz na nakreslenie jedného z domov (okrem domu1).

Úloha 3

Definujte vhodné vlastné príkazy a pomocou nich nakreslite nasledujúci obrázok:



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

Naučíme korytnačku kresliť kružnicu. Nakreslíme ju ako 24-uholník so stranou dĺžky 10.

viem kruznica
opakuj 24 [do 10 vp 15]
koniec



Definujeme tiež príkaz obluk, ktorý nakreslí štvrt' kružnicu.

viem obluk
opakuj 6 [do 10 vp 15]
koniec

Oblúky môžeme využiť na kreslenie mnohých obrázkov.

Napríklad z dvoch oblúkov vytvoríme lúpeň:



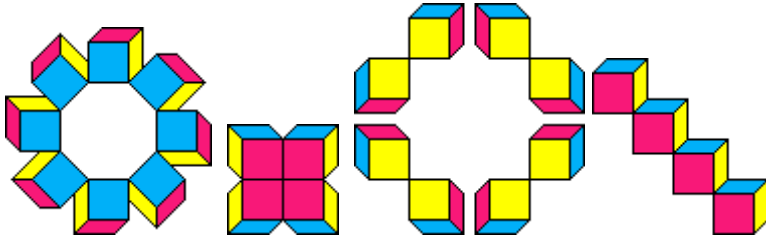
viem lupen
opakuj 2 [obluk vp 90]
koniec



Z viacerých lúpeňov vytvoríme kvet:

viem kvet
nechFp ?
opakuj 6 [lupen vp 60]
koniec



Úloha 4	<p>Definujte vlastné príkazy na nakreslenie nasledujúcich obrázkov. Využite príkaz <code>obluk</code>.</p> 
Vyskúšajte sa 1	<p>Definujte príkaz <code>kocka</code> s hranou dĺžky 20, ktorý nakreslí takýto obrázok (korytnačka by mala skončiť na rovnakom mieste, ako začala, natočená na sever).</p>  <p>Pomocou príkazu <code>kocka</code> sme nakreslili niekoľko obrázkov (vyfarbili sme ich dodatočne pomocou nástroja Výplň v paneli kreslenia). Zistite, ktorý riadok patrí ku ktorému obrázku.</p> <ol style="list-style-type: none"> 1. opakuj 4 [kocka vp 90 do 60 vp 180 kocka vp 180] 2. opakuj 4 [kocka vp 90] 3. opakuj 8 [kocka vp 90 do 20 v1 45] 4. opakuj 4 [kocka vp 90 do 20 v1 90 vz 20] 

Vlastné príkazy s parametrom

Príkaz `dopredu` sme vždy používali spolu s číselným parametrom určujúcim dĺžku o koľko bodov sa korytnačka posunie. Táto dĺžka mohla byť rôzna - korytnačka tak mohla nakresliť rôzne dlhé úsečky. Obrázky nakreslené našimi vlastnými príkazmi (napr. `stromcek`, `stvorec`, `trojuholnik`) boli stále rovnako veľké, pretože sme v nich na pohyb dopredu používali iba konštanty (napr. `do 50`, `do 60` a pod.). Môžeme však definovať aj také príkazy, v ktorých dĺžku nakreslenej úsečky či veľkosť posunutia určíme premennou - parametrom.

Otvorte projekt `vlastnePríkazy` a uložte ho pod názvom `príkazySPparametrom`.

Zmeníme príkaz `stromcek` tak, aby dĺžku kmeňa určoval parameter.

```
viem stromcek :kmen
nechfp "hneda
nechhp 10
do :kmen
nechfp "zelena4
bod 50
ph
vz :kmen
pd
koniec
```

? uprav "stromcek

Do 1. riadku za meno príkazu dáme medzeru, dvojbodku a meno parametra, v tomto prípade :kmen.

Tam, kde chceme použiť hodnotu parametra, napr. za príkazom do, tiež uvedieme dvojbodku a meno parametra (do :kmen, vz :kmen).

Príkaz stromcek odteraz môžeme používať len s číslom, napr.

? stromcek 100.

Skutočná hodnota 100 sa cez parameter :kmen prenesie do definície a vykoná.

Vyskúšajme ? stromcek 50.

Definujeme príkaz stvorec s parametrom :dlzka určujúcim dĺžku strany štvorca.

```
viem stvorec :dlzka
opakuj 4 [do :dlzka vp 90]
koniec
```

Vyskúšame:

? stvorec 20



? stvorec 40



Pri vytváraní vlastných príkazov s parametrami dodržiavame niekoľko zásad:

- parametrom dávame mená podľa toho, čo v príkaze vyjadrujú
- pred meno parametra píšeme dvojbodku, teda :strana, :dlzka
- meno parametra nemôže obsahovať medzeru, teda :dlzka kmena nie je správne meno parametra. Namiesto medzier použijete bodku :dlzka.kmena, podčiarkovník :dlzka_kmena alebo rôznu veľkosť písmen :dlzkaKmena.

Úloha 5

Zmeňte príkazy trojuholnik a obluk na príkazy s parametrom :dlzka.



? trojuholnik 30



? trojuhonik 40



? obluk 10



? obluk 15

Úloha 6

Pomocou vhodného vlastného príkazu s parametrom nakreslite obrázok:



V príkaze domcek sme využívali príkazy stvorec a trojuholnik. Teraz, keď už vieme kresliť štvorec a trojuholnik s rôznou dĺžkou, môžeme aj príkaz domcek zmeniť tak, aby dĺžka jeho základu a strechy bola parametrom.

```
viem domcek :dlzka
nechfp ?
stvorec :dlzka
do :dlzka
vp 30
trojuholnik :dlzka
vl 30
vz :dlzka
koniec
```



Nezabudnite ukladať svoj projekt.

Úloha 7

Pomocou príkazu `obluk` s parametrom `:dlzka` definujte príkaz `lupen` s parametrom `:dlzka`.



Vyskúšajte sa 2

Čo treba doplniť v príkaze `kamen`, aby sa dal pomocou neho nakresliť nasledujúci obrázok?

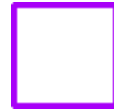
```
? opakuj 4 [kamen 50 vp 90 do 50 vl 90]
```



```
viem kamen :d
pd
nechHp 3
nechFp "zelená
opakuj 4 [ ..... do ... vp 90]
nechFp "tmavozelena5
.....
ph
koniec
```

Zatiaľ sme korytnačku naučili len príkazy, ktorých parametrom bola dĺžka. Parametrom našich vlastných príkazov však môže byť aj iný údaj, napr. farba pera, hrúbka pera, počet. Naučíme korytnačku dva rôzne príkazy na kreslenie štvorca. V jednom bude parametrom hrúbka pera, v druhom farba pera korytnačky. Tiež definujeme príkaz `sad`, ktorého parametrom bude počet stromčekov.

```
viem stvorecH :hrubka
nechHp :hrubka
opakuj 4 [do 50 vp 90]
koniec
```

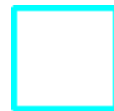


? stvorecH 3



? stvorecH 10

```
viem stvorecF :farba
nechFp :farba
opakuj 4 [do 50 vp 90]
koniec
```



? stvorecF "azurova



? stvorecF "olivova

```
viem sad :pocet
opakuj :pocet [stromcek 40 ph vp 90 do 40 vl 90 pd]
koniec
```



? sad 7



? sad 3

Úloha 8

Napište príkaz `slnko` s parametrom `:pocet`, ktorý bude určovať počet lúčov. Pomôcka: po každom lúči sa korytnačka otočí o uhol $360/:pocet$.



V niektorých príkazoch by sa nám hodilo, aby sme parametrom vedeli určiť nielen jednu hodnotu, ale viac rôznych údajov. Dva parametre potrebujeme napr. pri kreslení obdĺžnika - dĺžku a šírku. Pri kreslení kvetu by sme mohli jedným parametrom určiť počet lupeňov, druhým veľkosť lupeňa a tretím farbu.

```
viem obdlznik :dlzka :sirka
  opakuj 2 [do :dlzka vp 90 do :sirka vp 90]
koniec
```



? obdlznik 20 50



? obdlznik 60 30

```
viem kvet :pocet :dlzka :farba
  nechFp :farba
  opakuj :pocet [lupen :dlzka vp 360/:pocet]
koniec
```



? kvet2 8 5 "ruzova



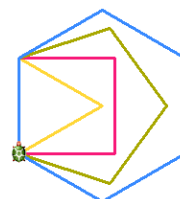
? kvet 5 10 "oranzova

Definujte príkaz `poly` s tromi parametrami:

- počet strán,
- veľkosť strany,
- uhol otočenia.

```
viem poly :n :d :uhol
  opakuj :n
  [do :d vp :uhol]
koniec
```

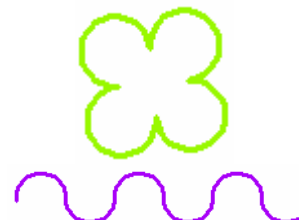
Využitím príkazu `poly` so správnymi parametrami nakreslite obrázok:

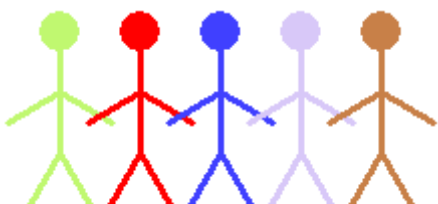


Pomocou príkazu `poly` môžeme okrem pravidelných n -uholníkov kresliť aj hviezdičky či časti kružnice. Vyskúšajte


```
? poly 5 50 144
? poly 9 80 160
? poly 24 10 15
? poly 12 10 15
? poly 6 10 15
```







Nakreslite pomocou príkazu `poly` nasledujúce obrázky:



Úloha 9	Upravte príkaz <code>slnko</code> tak, aby okrem parametra <code>:pocet</code> mal aj parameter <code>:dlzka</code> , ktorý bude určovať dĺžku lúčov.
Vyskúšajte sa 3	<p>Vytvorte príkaz <code>panacik</code> s parametrom <code>:velkost</code> - telo príkazu skopírujte zo systému Moodle (odkaz <i>Príkaz panáčik</i>). Upravte príkaz tak, aby dĺžka rúk, nôh, častí od nôh po ruky a od rúk po hlavu, a tiež veľkosť (priemer) hlavy boli určené parametrom <code>:velkost</code>. Definujte tiež príkaz <code>radPanacikov</code> s dvoma parametrami vyjadrujúcimi počet panáčikov v rade a veľkosť panáčikov. Pomocou tohto príkazu nakreslite rady s rôznym počtom panáčikov, napr.</p> <pre>? radPanacikov 5 30</pre> 

Rozširujúce úlohy

Úloha 10	<p>Definujte vlastné príkazy na nakreslenie nasledujúcich obrázkov:</p> 
Úloha 11	<p>Definujte vlastné príkazy na nakreslenie nasledujúcich obrázkov. Využite v nich vhodné príkazy <code>stvorec</code> a <code>trojuholnik</code>:</p>

	
Úloha 12	<p>Definujte vhodné vlastné príkazy a pomocou nich nakreslite nasledujúce obrázky:</p> 
Úloha 13	<p>Máme zadaný vlastný príkaz <code>kvet</code>, ktorý nakreslí nasledujúci obrázok.</p>  <p>Ktorý z nasledujúcich príkazov vykreslí nasledujúci obrázok?</p> <ol style="list-style-type: none"> 1. opakuj 6 [kvet vp 60] 2. opakuj 9 [kvet vp 40] 3. opakuj ? [kvet vp 90] 4. opakuj 10 [kvet vp ?] 
Úloha 14	<p>Definujte vlastné príkazy na nakreslenie nasledujúcich obrázkov. Využite príkaz <code>obluk</code>:</p> 
Úloha 15	<p>Pomocou príkazov <code>stvorec</code>, <code>trojuholnik</code> a <code>obluk</code> s parametrom <code>:dlzka</code> nakreslite nasledujúce obrázky:</p> 
Úloha 16	<p>Napište príkazy s jedným parametrom <code>:pocet</code> (počet schodov,</p>

počet lát, počet okien) na nakreslenie nasledujúcich obrázkov:



Úloha 17

Príkazy vytvorené v predchádzajúcej úlohe upravte tak, aby sa prostredníctvom ďalších parametrov dala:

- pyramíde určiť výška a šírka schodu (nielen ich počet),
- plotu určiť hrúbka a výška latky (nielen ich počet),
- radu okien určiť farba a hrúbka rámu (nielen ich počet).

Úloha 18

Napište príkaz `panelak`, pomocou ktorého bude možné kresliť paneláky s rôznym počtom poschodí a rôznym počtom okien na poschodí.



Čo sme sa naučili

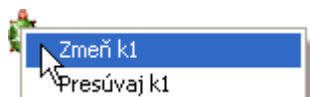
- naučiť korytnačku nový príkaz
- definovať príkaz s jedným alebo viac parametrami
- používať vo vlastných príkazoch rôzne typy parametrov: dĺžku, počet, farbu, hrúbku ...

Udalosti korytnačky, viac korytnačiek

Informácie o korytnačke, udalosť priKliknutí

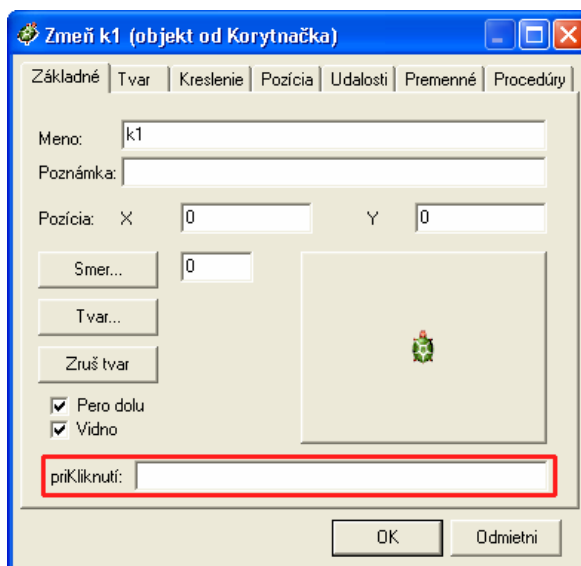
Otvorte projekt `vlastnePríkazy` a uložte ho pod názvom `udalosti`.

Klikneme na korytnačku pravým tlačidlom myši a v zobrazenej ponuke zvolíme `Zmeň k1`.



Zobrazené okno („rodný list“ korytnačky) obsahuje niekoľko záložiek s informáciami o korytnačke. Dozvieme sa napr. meno korytnačky, aký má práve smer, pozíciu, tvar, hrúbku či farbu pera a iné údaje. Tieto údaje môžeme aj meniť.

V záložke `Základné` si všimnime položku - `priKliknutí`. Tu môžeme definovať, ako má korytnačka reagovať, keď na ňu klikneme myšou, t.j. keď nastane `udalosť priKliknutí`.

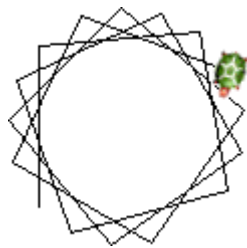


Pozor! Ak neklikneme presne na korytnačku, otvorí sa namiesto ponuky príkazov pre korytnačku ponuka príkazov pre stránku.

`Udalosťou` je napr. aj ťahanie myšou, dvojkliknutie, stlačenie klávesu atď. Na mnohé udalosti môžeme v `Imagine` definovať vlastné reakcie, t.j. definovať, aké príkazy sa majú vykonať, keď príslušná udalosť nastane.

Napišme do riadku **priKliknutí** príказы do 80 vp 85.

priKliknutí:



Zatvoríme okno tlačidlom **OK** a klikneme na korytnačku.
Po niekoľkých kliknutiach nakreslíme takýto obrázok:

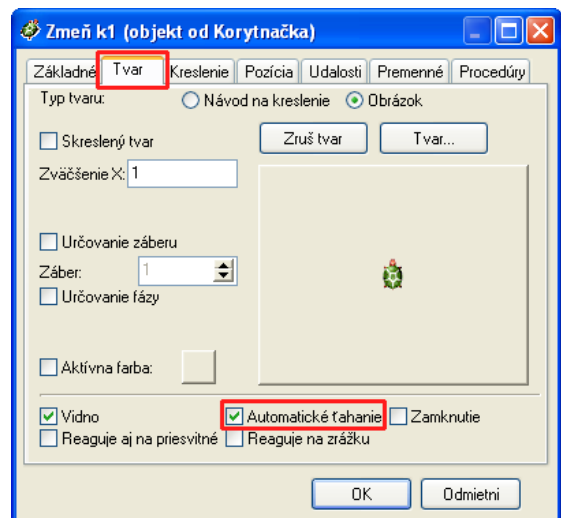
Úloha 1

Pozmeňte predchádzajúce príkazy v udalosti **priKliknutí** tak, aby korytnačka po niekoľkých kliknutiach nakreslila obrázok:

Automatické ťahanie, udalosť priŤahání

Korytnačku sme doteraz presúvali len pomocou príkazov (**do**, **vp**). Ukážeme si iný spôsob. Pokračujeme v projekte **udalosti**.

1. V rodnom liste korytnačky zmažeme všetko, čo je v riadku **priKliknutí** (tým zrušíme udalosť **priKliknutí**).
2. Prejdeme na záložku **Tvar**.
3. Zapneme nastavenie **Automatické ťahanie**.
4. Zatvoríme okno tlačidlom **OK**.



Teraz môžeme korytnačku ťahať myšou. Ak má korytnačka zapnuté pero, tak môžeme takýmto spôsobom kresliť podobne ako nástrojom ceruzka v grafickom editore.



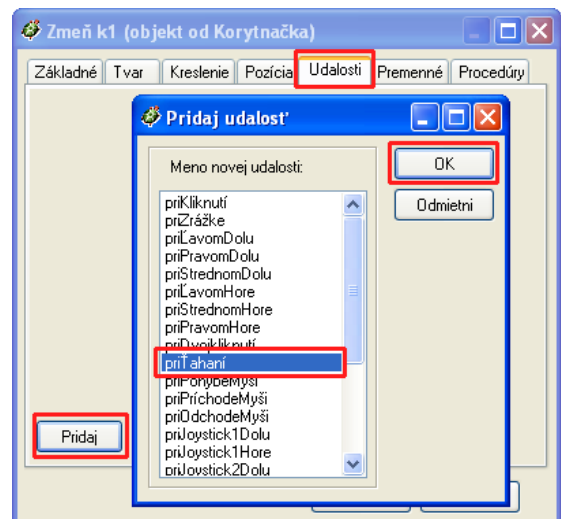
Námet: Vytvorte nový projekt na kreslenie myšou: nastavte korytnačke automatické ťahanie, pridajte tlačidlá na vypnutie a zapnutie pera, tlačidlo na nastavenie náhodnej farby, tlačidlo na nastavenie náhodnej hrúbky a tlačidlo na zmazanie stránky.

S korytnačkou, ktorá má nastavené automatické ťahanie, môžeme kresliť rôzne zaujímavé efekty. Môžeme jej totiž definovať, ako sa má správať, keď ju ťaháme myšou, t.j. nastaviť jej udalosť **priŤahání**.

1. V rodnom liste korytnačky zvolíme záložku **Udalosti**.
2. Klikneme na tlačidlo **Pridaj**.
3. Zo zoznamu udalostí vyberieme udalosť **priŤahání** a stlačíme tlačidlo **OK**.
4. Do riadku **priŤahání** napíšeme príkaz: **nechFp ? bod 50** a klikneme na **OK**.

priŤahání

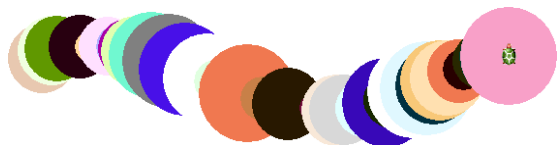
Poznámka: Neodporúčame, aby mala korytnačka definovanú udalosť **priKliknutí** ani **priŤahání**



Teraz môžeme ťahaním nakresliť pekný koráľkový náhrdelník.



Ak potrebujeme definovanú udalosť **priťahani** zmeniť, napríklad chceme kresliť rôzne veľké bodky, otvoríme rodný list korytnačky a zvolíme záložku **Udalosti**. Riadok **priťahani** zmeníme na `nechFp ? bod ?` a stlačíme **OK**.



Zmeníme udalosť **priťahani** takto: `nechFp ? bod ? cakaj 100`. Príkaz `cakaj` spôsobí, že korytnačka medzi vykonávaním jednotlivých príkazov chvíľu počká. Čím je číslo za príkazom `cakaj` väčšie, tým dlhšie bude čakať. Príkaz `cakaj` budeme v udalosti **priťahani** používať na to, aby sa udalosť **priťahani** opakovala menej často (aby jednotlivé bodky či iné obrázky boli ďalej od seba).

Príkazom `cakaj 1000` pozdržíme vykonávanie príkazov na 1 sekundu, príkazom `cakaj 10` na 10 milisekúnd.

Úloha 2

Nastavte udalosť **priťahani** tak, aby ste pri ťahaní korytnačky nakreslil nasledujúci obrázok. Ak sa dá, použite príkaz, ktorý ste už vytvorili, ak nie, definujte nový. Uložte projekt.

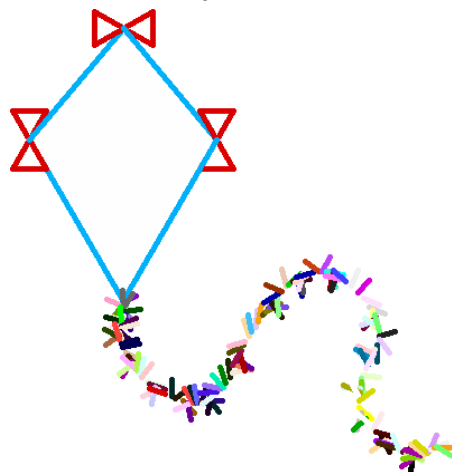


Úloha 3

Otvorte projekt *4tahanie*. Zvoľte si niektorý z príkazov v ľavom stĺpci a ťahajte korytnačku. Pozrite sa, ako sú jednotlivé príkazy definované a niektoré upravte.

Vyskúšajte sa 1

Vymyslite nový chvostík šarkanovi na obrázku a nakreslite ho pomocou nastavenia udalosti **priťahani**.



Tip: Ak chceme vybrať ľubovoľnú farbu z istej množiny farieb, môžeme využiť operáciu `?prvok` na výber náhodného prvku z množiny. Napr. príkazom `? nechFp ?prvok [zelená modrá červená]` zmeníme farbu pera korytnačky na zelenú, modrú alebo červenú. Príkazom `? nechFp ?prvok [modra5 modra6 modra7 modra8 modra9]` zmeníme farbu pera na niektorý odtieň modrej. Takto môžeme kresliť pri ťahaní korytnačky myšou napr. rôzne žlté hviezdičky, rôzne zelené stebľa trávy alebo stromčeky a pod.



Tip: Šarkana si môžete sami nakresliť. V paneli nástrojov zapnite **Ukáž/Skry panel kreslenia**

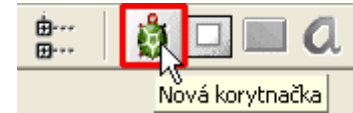


Pomocou nástrojov grafického editora nakreslite šarkana.

Viac korytnáčiek

Doteraz sme všetko robili len s jednou korytnačkou. Korytnáčiek môžeme mať na stránke aj viac a s každou z nich môžeme kresliť niečo iné. Ukážeme si, ako pridáme novú korytnačku.

1. Otvoríme nový projekt. V paneli nástrojov zvolíme nástroj **Nová korytnačka**. Kurzor myši sa zmení na korytnačku.
2. Klikneme do stránky. Na mieste kliknutia sa vytvorí nová korytnačka.



Ak sa pokúsime príkazom zmeniť novej korytnačke farbu či hrúbku pera, zistíme, že to nefunguje. Na príkazy v príkazovom riadku reaguje len prvá korytnačka. Neskôr si ukážeme, ako sa môžeme rozprávať aj s ďalšími korytnačkami. Zatiaľ použijeme na zmenu ich nastavení nástroje v rodnom liste.

Nová korytnačka je rovnaká ako tá prvá. Tiež má svoj rodný list, v ktorom môžeme meniť jej nastavenia a definovať udalosti.

Úloha 4

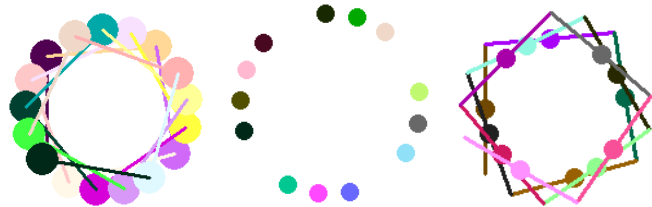
Zapnite novej korytnačke **automatické ťahanie** a definujte jej udalosť **priŤahani** tak, aby kreslila farebné hviezdičky. Hrúbku pera korytnačky zmeňte v jej rodnom liste.



Rozširujúce úlohy

Úloha 5

Pozmeňte predchádzajúce príkazy v udalosti **priKliknutí** tak, aby korytnačka po niekoľkých kliknutiach nakreslila nasledujúce obrázky:



Úloha 6

Nastavte udalosť **priŤahani** tak, aby ste pri ťahaní korytnačky nakreslili nasledujúce obrázky. Ak sa dá, použite príkazy, ktoré ste už vytvorili, ak nie, definujte nové. Uložte projekt.



Úloha 7

Vytvorte novú korytnačku. V rodnom liste jej zmeňte hrúbku pera. Zapnite jej **automatické ťahanie** a definujte udalosť **priŤahani** tak, aby kreslila farebné mašličky.



Námet: Pomocou nástrojov v paneli kreslenia nakreslite troch šarkanov. Každému z nich dokreslite chvostík pomocou inej korytnačky s inou udalosťou **priŤahani**.

Čo sme sa naučili

- pozrieť si informácie o korytnačke v rodnom liste a meniť ich
- definovať udalosť **priKliknutí**, t.j. určiť, ako sa má korytnačka správať, keď na ňu klikneme ľavým tlačidlom myši
- využiť nastavenie **automatické ťahanie** a ťahať korytnačku myšou po stránke

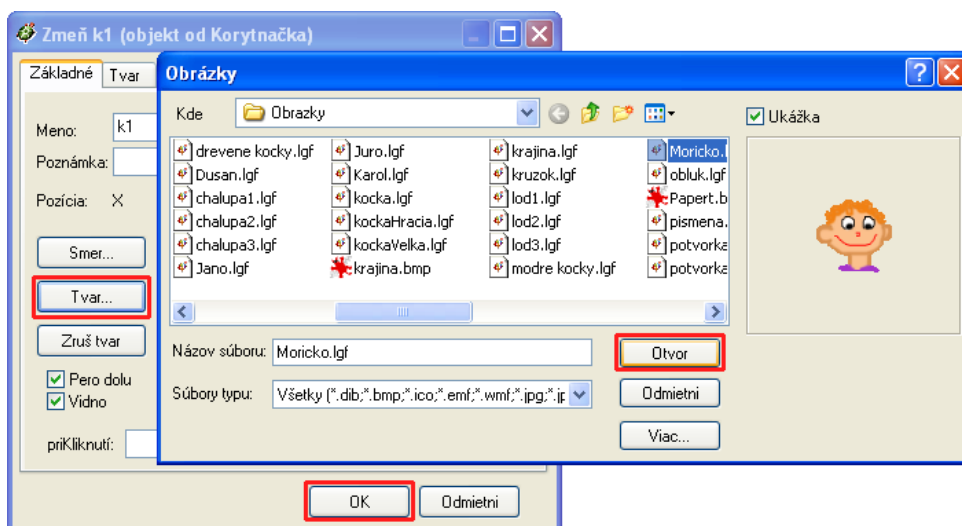
- kresliť pri ťahaní korytnačky rôzne efekty definovaním reakcie na udalosť pri ťahaní
- vytvoriť novú korytnačku
- čo je to *udalosť*

Živé obrázky

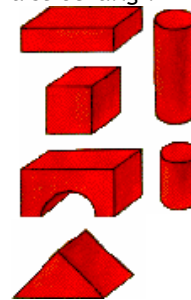
Tvary korytnačiek

Otvorme nový projekt. V rodnom liste korytnačky môžeme okrem iného meniť aj jej tvar. Korytnačka môže vyzeráť napr. ako žabka, autíčko, domček, chlapec či dievča.

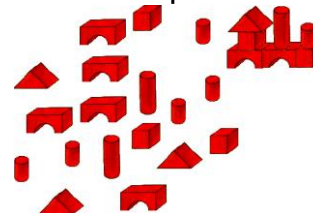
1. Otvoríme rodný list korytnačky a v záložke **Základné** klikneme na tlačidlo **Tvar**.
2. Pozrieme si ponuku obrázkov (stačí klikáť na jednotlivé súbory, vpravo sa zobrazuje ukážka) a vyberieme si nejaký, napr. **Moricko.lgf**.
3. Výber obrázka potvrdíme tlačidlom **Otvor**.
4. V rodnom liste klikneme na tlačidlo **OK**.



Námet: Vytvorte veľa korytnačiek. V rodnom liste každej vypnite pero Pero dolu a nastavte automatické ťahanie. Zmeňte im tvary na obrázky zo súborov: doska.lgf, kocka.lgf, obluk.lgf, stlp.lgf, stlpik.lgf a strecha.lgf.



Projekt uložte ako stavebnica.imp.



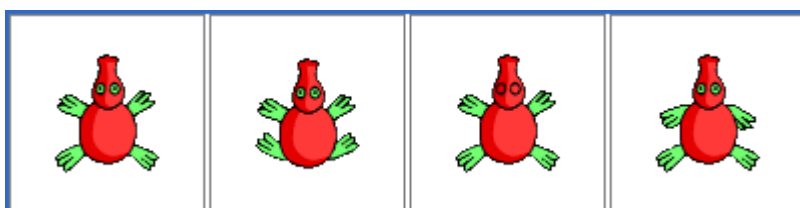
Namiesto korytnačky máme Móricka.

Úloha 1

Pridajte Mórickovi kamarátov - vytvorte nové korytnačky a zmeňte im tvary na iné dievčenské alebo chlapčenské tváričky.



Pridáme ďalšiu korytnačku a zmeníme jej tvar na obrázok **potvorka7.lgf**. Všimnite si, že táto korytnačka je iná ako ostatné, hýbe nohami a žmurká. Jej tvarom je totiž animovaný obrázok, t.j. obrázok, skladajúci sa z niekoľkých fáz, ktoré sa mierne odlišujú.



Obrázok 6: Fázy animovaného obrázka

Uložíme projekt pod názvom **tvaryKorytnaciek**.

Obrázky z korytnáčiek

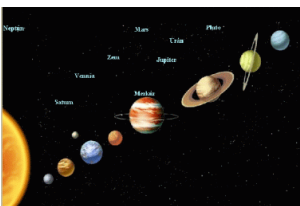
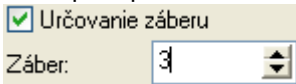
Pomocou korytnáčiek s rôznymi tvarmi vytvoríme pekný obrázok - krajinu s kokosovými palmami, morom, loďkami a domorodcami.

1. Otvoríme nový projekt. Zrušíme korytnačku: klikneme na ňu pravým tlačidlom myši a zvolíme **Zruš k1**.
2. Zmeníme pozadie stránky:
 - a) Klikneme na stránku pravým tlačidlom myši a zvolíme **Pozadie zo súboru ...**
 - b) V priečinku **Pozadia** vyberieme obrázok **Kokosove palmy a rieka.lgf**.
3. V mori vytvoríme novú korytnačku. V rodnom liste korytnačky:
 - a) zmeníme meno korytnačky na **1ođ1**,
 - b) vypneme jej pero **Pero dolu**,
 - c) zmeníme jej tvar na obrázok **lodka.lgf** v priečinku **ine**,
 - d) zapneme jej automatické ťahanie.
4. Vytvoríme ďalšiu loď skopírovaním tej prvej:
 - a) klikneme na loď pravým tlačidlom myši a zvolíme **Skopíruj do schránky**,
 - b) klikneme do stránky pravým tlačidlom myši a zvolíme **Prilep zo schránky**.

Ak treba, presunieme myšou novú loď do mora. Nová loď sa volá **1ođ2**.
5. Pridáme ďalšiu korytnačku - domorodca:
 - a) vytvoríme kópiu jednej z lodí (ako v bode 5)
 - b) v rodnom liste novej korytnačky zmeníme meno na **domorodec1**,
 - c) zmeníme tvar na obrázok **domorodec** z priečinka **ine**.
6. Kopírovaním vytvoríme ďalšieho domorodca.
7. Uložíme projekt pod názvom **palmy!**

Takýto spôsob vytvárania korytnáčiek nazývame **klonovanie**.

Námet: Vytvorte projekt na precvičovanie názvov slnečnej sústavy. Do pozadia stránky vložte obrázok zo súboru **slnečnaSus.jpg**. Vytvorte korytnačku s menom **planeta1**, tvarom zo súboru **planety.lgf**. Zapnite jej automatické ťahanie, vypnite pero a v záložke **Tvar** nastavte položku **Záber** na hodnotu 1. Kopírovaním vytvorte ďalších 8 korytnáčiek. V záložke tvar im nastavte **Zaber** postupne na 2 až 9.



Úloha 2

Kopírovaním vytvorte ďalšie korytnačky, napr.:

- indiána,
- niekoľko včiel,
- niekoľko kokosových orechov,
- opicu,
- prípadne ďalšie korytnačky, ktoré sa hodia k pozadiu.

Všetky potrebné obrázky nájdete v priečinku **ine**. Korytnačky vhodne pomenujte (napr. **indian1**, **vcela1**, **orech1**, **orech2**, ..., **opica1**) a vhodne umiestnite do pozadia. Uložte projekt!



Oživujeme obrázky pomocou procesov

Vrátíme sa k projektu **palmy** a ukážeme si, ako môžeme náš obrázok oživiť.

Ako prvú „oživíme“ jednu **1ođ1**. Loď by sa mala pohybovať smerom vpravo, preto ju najskôr otočíme príkazom `? vpravo 90`. Potom ju rozhýbeme tak, že pre ňu spustíme proces: `? kazdych 50 [do 5]`

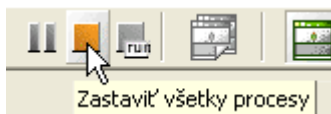
Proces je „malý neviditeľný motorček, ktorý neustále vykonáva nejakú činnosť“ [1].

Príkazom `kazdych 50 [do 5]` sme spustili proces, ktorý každých 50 milisekúnd vykonáva príkaz `do 5`.

ako často postupnosť príkazov, ktorá sa má vykonávať

kazdych 50 [do 5]

Lod' sa bude pohybovať stále, až kým spustený proces nezastavíme, alebo nezatvoríme projekt či nevypneme Imagine. Proces zastavíme nástrojom **Zastaviť všetky procesy** v paneli nástrojov.



Rozhýbeme lod' ešte raz, ale trochu rýchlejšie, t.j. príkaz `do 5` sa musí vykonávať častejšie, napríklad každých 30 milisekúnd:

? `kazdych 30 [do 5]`

Zastavíme proces. Teraz rozhýbeme korytnačku `domorodec1`. Otvoríme rodný list pre `domorodca1` a zmeníme mu smer na 90, t.j. na východ (môžeme využiť pomôcku stlačením tlačidla **Smer**). Rodný list zatvoríme. Spustíme proces:

? `kazdych 50 [do 2]`

Namiesto `domorodca` sa pohla lod'. Dôvodom je, že na príkazy v príkazovom riadku reaguje len prvá vytvorená korytnačka. Zastavme proces a spustíme ho takýmto spôsobom:

? `domorodec1'kazdych 50 [do 2]`

Pred príkaz `kazdych` sme pridali oslovenie korytnačky, t.j. jej meno a apostrof.

Vždy, keď máme na stránke viac korytnačiek, musíme tú, ktorej chceme dať príkaz, osloviť. Na oslovenie používame meno korytnačky a apostrof. Hneď za apostrofom nasleduje príkaz, ktorý má oslovená korytnačka vykonať. Oslovenie musíme napísať pred každý príkaz, ktorý má korytnačka vykonať.

meno korytnačky príkaz

lod2'do 10

Rozhýbať ďalšieho `domorodca` už pre nás nebude problém. V rodnom liste mu nastavíme smer 270 (na západ). Potom spustíme proces:

? `domorodec2'kazdych 50 [do 2]`

Všimnite si, že teraz sa pohybujú obaja `domorodci` naraz, t.j. dva procesy bežia súčasne. Spustíme ďalší proces - pre včelu:

? `vcela1'kazdych 50 [vp ? do 2]`

Všetky príkazy, ktoré nie sú adresné (teda nepovieme kto ich má vykonať), vykonáva prvá vytvorená korytnačka.

Námet: Vylepšite živý obraz pridaním interakcie - definujte pre niektoré korytnačky reakciu na udalosť **priKliknutí**. Napríklad `domorodci` by sa mohli po kliknutí myšou otočiť opačným smerom (o 180 stupňov), opica by mohla po kliknutí myšou poskočiť (posunúť v smere nahor, chvíľu počkať a vrátiť sa do pôvodnej pozície).

Úloha 3

Spustite vhodné procesy pre ďalšie korytnačky, pre obe lode, pre indiána, pre ostatné včely. Ak je to potrebné, nastavte im pred spustením procesov vhodný smer.

Spustené procesy sa s projektom neukladajú. Definujeme preto príkaz `hybteSa`, ktorým budeme môcť všetky procesy kedykoľvek spustiť.

```
viem hybteSa
domorodec1'kazdych 50 [do 2]
domorodec2'kazdych 50 [do 3]
lod1'kazdych 30 [do 5]
vcela1'kazdych 50 [vp ? do 2]
koniec
```

Úloha 4

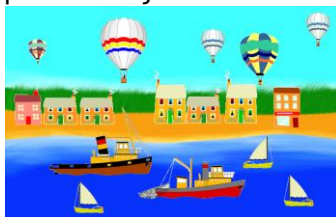
Doplňte do príkazu `hybteSa` spustenie procesov pre ostatné korytnačky. Na stránke vytvorte tlačidlo, ktoré spustí príkaz `hybteSa`. Uložte projekt.

Rozširujúce úlohy

Úloha 5

Vytvorte nejaký zaujímavý obrázok z korytnačiek: zmeňte pozadie stránky, pridajte korytnačky s vhodnými nastaveniami, zmeňte im tvary, prípadne smer. Použite niektorý z nasledujúcich námetov:

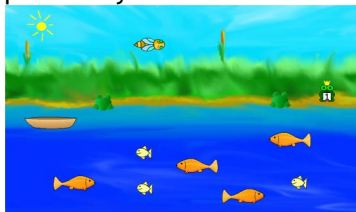
- pozadie krajina



- pozadie cesta



- pozadie rybník



Nezabudnite si svoj projekt uložiť.

Úloha 6

Oživte obraz z predchádzajúcej úlohy. Definujte príkaz, ktorý spustí aspoň 3 procesy pre 3 rôzne korytnačky. Pridajte tlačidlo, ktoré spustí tento príkaz. Aspoň jednej korytnačke definujte nejakú vhodnú udalosť `priKliknutí`.

Čo sme sa naučili

- zmeniť tvar korytnačky na statický alebo animovaný obrázok
- zmeniť pozadie stránky
- dávať príkazy korytnačkám pomocou ich oslovenia
- vytvárať živé obrazy použitím procesov
- čo je to *proces*

Zoznam príkazov

dopredu do	<code>dopredu pocet_krokov</code> , napr. <code>dopredu 10</code> presunie korytnačku o zadaný počet krokov (napr. 10) dopredu, ak má zapnuté pero, nakreslí pritom čiaru danej dĺžky
vzad vz	<code>vzad pocet_krokov</code> , napr. <code>vzad 10</code> presunie korytnačku o zadaný počet krokov (napr. 10) dozadu, ak má zapnuté pero, nakreslí pritom čiaru danej dĺžky

vpravo vp	<code>vpravo uhol</code> , napr. <code>vpravo 90</code> otočí korytnačku o zadaný uhol (napr. 90 stupňov) vpravo
vľavo vl	<code>vľavo uhol</code> , napr. <code>vľavo 90</code> otočí korytnačku o zadaný uhol (napr. 90 stupňov) doľava
nechFarbaPera nechFp	<code>nechFp "farba</code> , napr. <code>nechFp "zelená</code> zmení farbu pera na zadanú farbu (napr. zelenú)
nechHrúbkaPera nechHp	<code>nechHp hrúbka</code> , napr. <code>nechHp 5</code> zmení hrúbku pera na zadanú hrúbku (napr. 5)
bod	<code>bod priemer</code> , napr. <code>bod 50</code> nakreslí bod so zadaným priemerom (napr. 50)
znova	zmaže stránku a korytnačku presunie do stredu stránky
peroHore ph	vypne pero, takže pri pohybe korytnačka nekreslí čiaru
peroDolu pd	zapne pero, takže pri pohybe korytnačka kreslí čiaru
opakuje	<code>opakuje počet [postupnosť príkazov]</code> , napr. <code>opakuje 4 [do 50 vp 90]</code> zopakuje postupnosť príkazov (napr. do 50 vp 90) zadaný počet opakovaní (napr. štyrikrát)
uprav	<code>uprav "názov príkazu</code> , napr. <code>uprav "stvorec</code> otvorí okno na definovanie príkazov, v ktorom sú zobrazené riadky <code>viem názov príkazu</code> <code>koniec</code> (napr. <code>viem stvorec</code> <code>koniec</code>)
cakaj	<code>cakaj počet_milisekúnd</code> , napr. <code>cakaj 100</code> zdrží vykonanie nasledujúceho príkazu o zadaný počet milisekúnd (napr. o 100 milisekúnd)
kazdych	<code>kazdych počet_milisekúnd [príkazy]</code> , napr. <code>kazdych 70 [do 5]</code> spustí proces, ktorý každých niekoľko milisekúnd (napr. 70) vykoná zadaný príkaz alebo postupnosť príkazov (napr. do 5)

Čo sme sa naučili v tomto module

Zhrnutie

V tomto module sme sa naučili:

- používať základné príkazy korytnačej grafiky na kreslenie obrázkov,
 - používať cyklus s pevným počtom opakovaní pri kreslení obrázkov s opakujúcimi sa časťami,
 - vytvárať vlastné príkazy bez parametrov a vlastné príkazy s parametrami rôznych „typov“ (dĺžka, farba, hrúbka),
 - skladat príkazy do zložitejších stavebníc,
 - rozložiť si problém na menšie časti,
 - vytvárať jednoduché projekty s využitím tlačidiel, udalostí a procesov.
- Oboznámili sme sa s pojmami **príkaz**, **parameter**, **cyklus**, **udalosť**, **proces**.

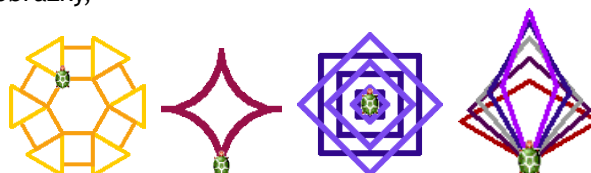
Preverenie výstupných vedomostí

Po absolvovaní modulu by ste mali vedieť:

- pomocou základných príkazov a opakovania nakresliť napr. nasledujúce obrázky,



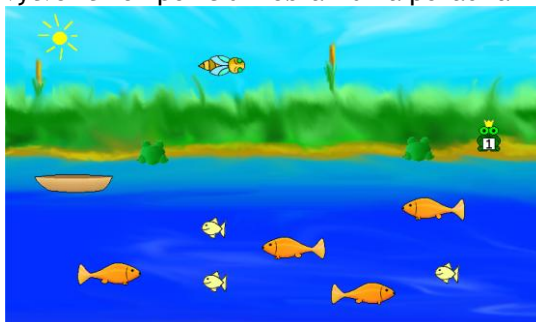
- naučiť korytnačku príkazy, pomocou ktorých nakreslíte napr. takéto obrázky,



- kresliť s korytnačkou pri ťahaní myšou ľubovoľné obrázky, napr.



- vytvoriť kompozíciu z obrázku na pozadí a korytnačiek rôznych tvarov, napr.



- oživiť kompozíciu z korytnačiek rôznych tvarov využitím procesov a udalostí, napr. zabezpečiť, aby v predchádzajúcom projekte ryby plávali, hmyz lietal, žabky skákali a pod.,

vytvárať a používať vo svojich projektoch vlastné tlačidlá,

- vysvetliť (aj na príklade):
 - čo je to parameter,
 - čo je to cyklus a kedy ho používame,
 - čo je to udalosť,
 - čo je to proces.

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.** (2006) *Programovanie v prostredí Imagine*. Bratislava: Metodicko-pedagogické centrum, 2006, 90 strán. ISBN 80-8052-260-X
- [3] **Ondříšková, J.** (2006) *Dištančný kurz detského programovania pre učiteľov v prostredí Imagine (diplomová práca)*. Bratislava, 2006, 63 strán.
- [4] [image.infovek.sk:](http://image.infovek.sk/)
[http://image.infovek.sk/index.php?k=2&v=2,](http://image.infovek.sk/index.php?k=2&v=2)
[http://image.infovek.sk/index.php?k=202,](http://image.infovek.sk/index.php?k=202)
<http://image.infovek.sk/index.php?k=2&v=17>
- [5] **Blaho, A.** (2006) *Programovanie v Delphi*. Bratislava: SPN, 2006, 144 strán. ISBN 80-10-00421-9

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
PaedDr. Viera Palmárová, PhD.
Ing. Katarína Pilátová

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

Podnázov Úvod do programovania

Študijný materiál prešiel recenzným pokračovaním.

Recenzenti PaedDr. Miloslava Sudolská, PhD.
RNDr. František Galčík

Počet strán 32

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 978-80-89225-55-2