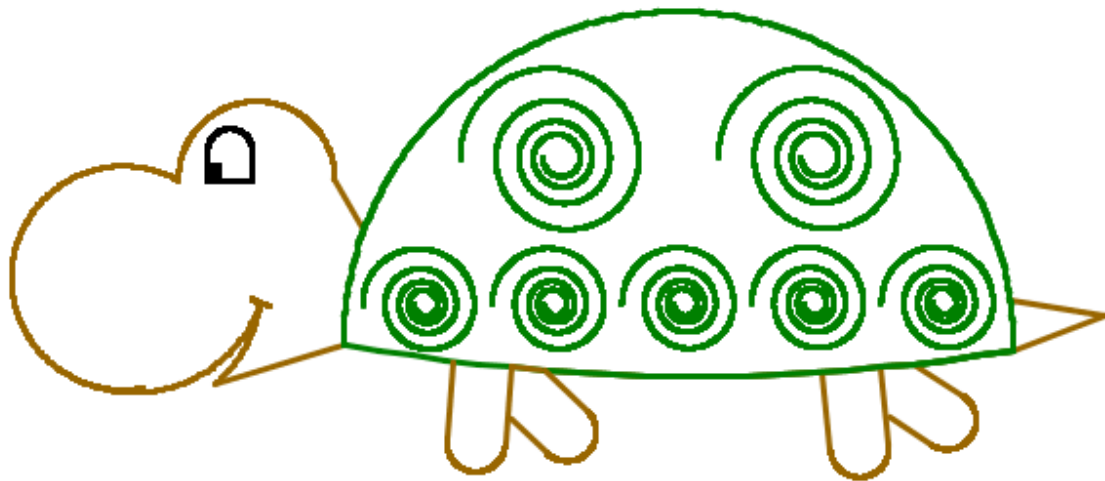


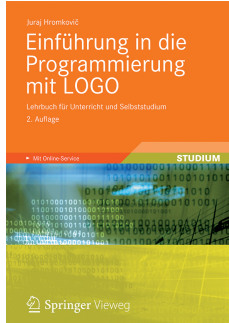
Heidi Gebauer Juraj Hromkovič Lucia Keller
Ivana Kosírová Giovanni Serafini Björn Steffen

Programovanie s LOGO-m



Programovanie s LOGO-m

Obsah tohto skriptu vychádza z učebnice *Einführung in die Programmierung mit LOGO*, kapitoly 1–7. Celá učebnica pozostáva z 15 kapitol a obsahuje veľa dodatočných príkladov, úloh a vysvetlení, ako aj rady pre učiteľov.



Juraj Hromkovič. *Einführung in die Programmierung mit LOGO: Lehrbuch für Unterricht und Selbststudium*, 2. Aufl., Springer Vieweg 2012. ISBN: 978-3-8348-1852-2.

Verzia 3.0, 10. mája 2013, SVN-Rev: 12181

Preklad do slovenčiny: Ivana Kosírová

Programovacie prostredie

Príklady a cvičenia v tomto skripte sú určené pre programovacie prostredie XLogo, ktoré sa da zadarmo získať z webstránky xlogo.tuxfamily.org.

Aby príklady správne fungovali, musí byť jazyk v XLogu nastavený na angličtinu.

Práva na užívanie

ABZ poskytuje predložený materiál zadarmo pre vnútorné použitie a vzdelávacie účely učiteľom a vzdelávacím inštitúciám.

ABZ

Centrum informatiky a vzdelávania (ABZ) ETH Zürich podporuje školy a učiteľov, ktorí by chceli zriadiť alebo rozšíriť vzdelávacie aktivity v oblasti informatiky. Podpora sa pohybuje od individuálnych konzultácií a výuky na mieste ETH profesormi a tímom ABZ po školenia pre učiteľov a údržbu štúdijských materiálov.

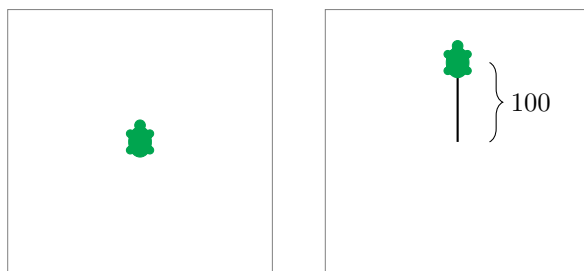
www.abz.inf.ethz.ch

1 Základné inštrukcie

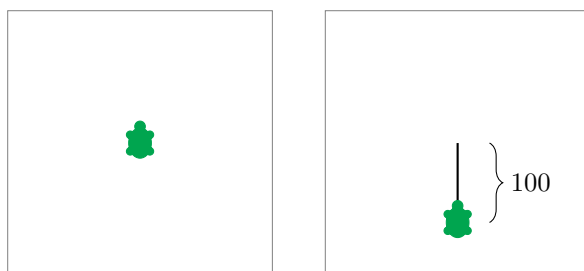
Počítačový príkaz je inštrukcia, ktorej počítač rozumie a vie ju splniť. Vo všeobecnosti rozumie počítač iba veľmi základné inštrukcie, ktoré sa dajú spojiť a vytvoriť tak komplikovanejšie inštrukcie. Takáto postupnosť príkazov sa nazýva **počítačový program**. Napísať program nie je ľahké. Existujú programy, ktoré sú zložené z miliónov príkazov. Aby človek pochopil tak komplikovaný program, je veľmi dôležité pristupovať k písaniu programu premyslene. To sa teraz naučíme v tomto programovacom kurze.

Kreslenie rovných čiar

Príkaz **forward 100** alebo **fd 100** pohne korytnačku 100 krokov vpred:



S príkazom **back 100** alebo **bk 100** môžeš poslať korytnačku 100 krokov vzad:



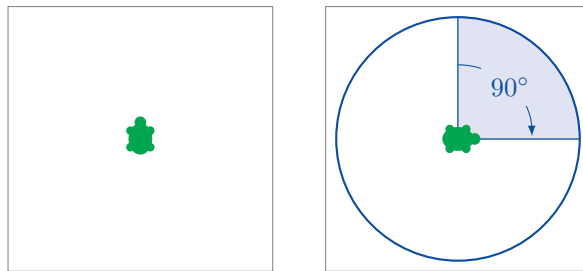
Ako zmazať obrazovku a znova začať

Príkaz **cs** vymaže všetko, čo je na obrazovke a umiestni korytnačku do jej štartovacej pozície.

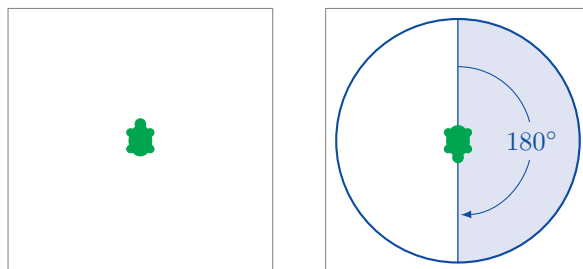
Otáčanie

Korytnačka sa pohybuje vždy len v smere, ktorým sa pozerá.

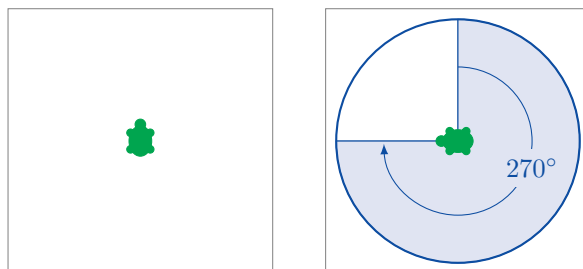
Pomocou príkazu **right 90** alebo **rt 90** otočíš korytnačku o 90° doprava. To predstavuje štvrtinu kruhu:



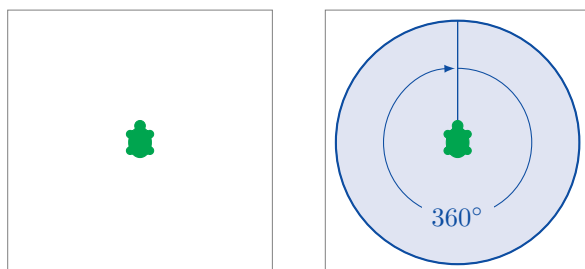
Príkaz **right 180** či **rt 180** otočí korytnačku o 180° smerom doprava. To predstavuje polovicu otočky:



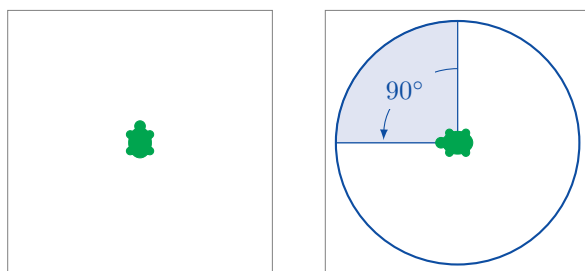
right 270 alebo aj **rt 270** otočí korytnačku o 270° doprava:



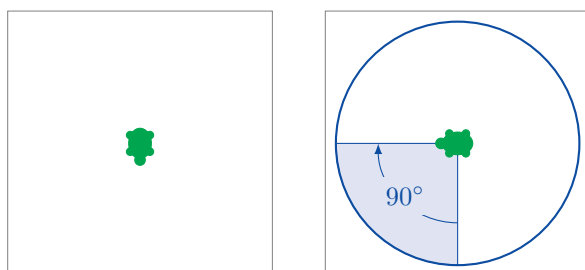
Príkaz **right 360** a **rt 360** otočí korytnačku o 360° doprava. To predstavuje jednu celú otočku:



Použitím príkazu **left 90** alebo **lt 90** otočíš korytnačku o 90° doľava:



Je dôležité si uvedomiť, že spomínané otočenia vníma korytnačka so svojho uhla pohľadu. To sa da ľahko všimnúť na nasledujúcom obrázku pri použití príkazu **rt 90**:



Programovanie

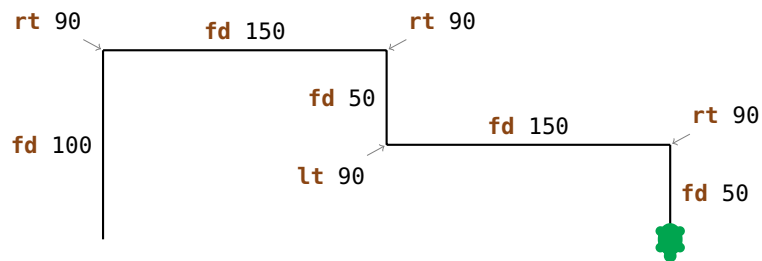
Programovať znamená napísať mnoho príkazov za sebou.

Cvičenie 1

Odpíš a spusti nasledujúci program:

```
fd 100  
rt 90  
fd 150  
rt 90  
fd 50  
lt 90  
fd 150  
rt 90  
fd 50
```

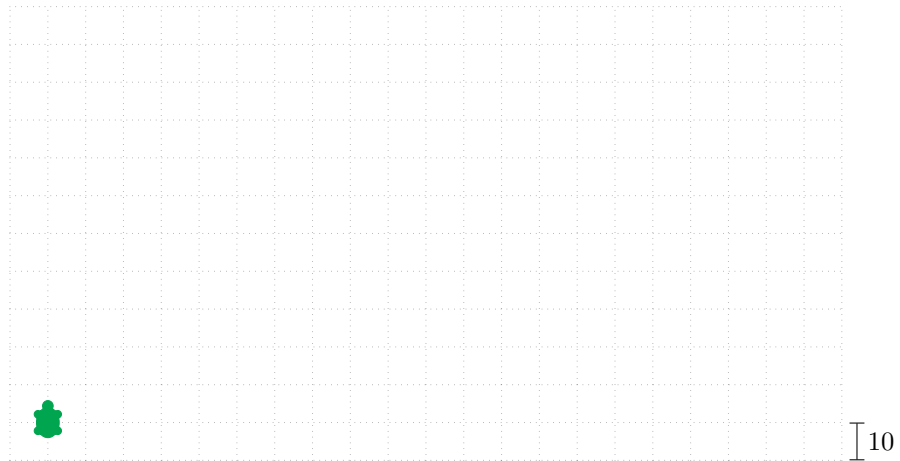
Získal/a si nasledujúci obrázok?



Cvičenie 2

Odpíš a spusti nasledujúci program:

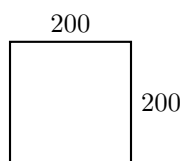
```
fd 100  
rt 90  
fd 200  
rt 90  
fd 80  
rt 90  
fd 100  
rt 90  
fd 50
```



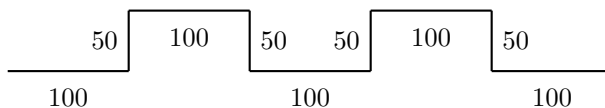
Zakresli do tejto mriežky výsledný obrázok a popíš, ktorý príkaz čo vykonal (pozri Cvičenie 1).

Cvičenie 3

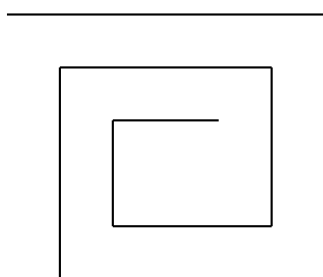
Napíš program pre každý z nasledujúcich obrázkov. Pre všetky obrázky si môžeš zvoliť štartovaciu pozíciu korytnačky.



(a)

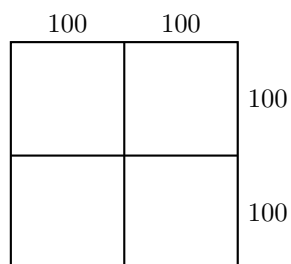


(b)



Veľkosť si smieš zvoliť.

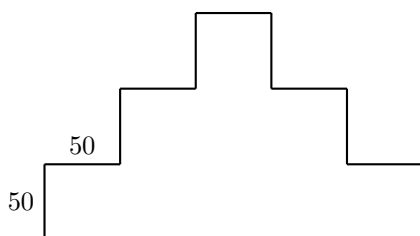
(c)



(d)

Cvičenie 4

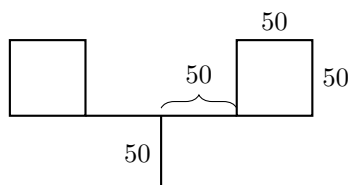
Napíš program, ktorý nakreslí nasledujúci obrázok:



Dokážeš naprogramovať tento obrázok len pomocou príkazov **fd 50** a **rt 90**?

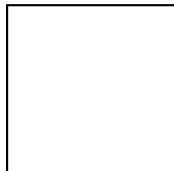
Cvičenie 5

Anička by chcela nakresliť nasledujúci obrázok. Pomôžeš jej?



2 Príkaz **repeat**

Chceme nakresliť štvorec so stranou dĺžky 100,



toto vieme docieľiť s nasledujúcim programom:

```
fd 100  
rt 90  
fd 100  
rt 90  
fd 100  
rt 90  
fd 100  
rt 90
```

Všimni si, že rovnaké dva príkazy

```
fd 100  
rt 90
```

sa štyrikrát opakujú. Nebolo by ľahšie povedať počítaču, aby ich jednoducho štyrikrát zopakoval namiesto aby sme ich vypísali štyrikrát za sebou?

To vieme spraviť nasledovne:

repeat	4	[fd 100 rt 90]
Príkaz na opakovanie programu	Počet opakovaní	Postupnosť príkazov, ktoré sa majú opakovať

Cvičenie 6

Odpíš a spusti nasledujúci program:

```
fd 75 lt 90  
fd 75 lt 90  
fd 75 lt 90  
fd 75 lt 90
```

Aký obrázok nakreslí tento program? Vieš celý program skrátiť pomocou príkazu **repeat**?

Cvičenie 7

Odpíš nasledujúci program a zisti, čo nakreslí:

```
fd 50 rt 60  
fd 50 rt 60  
fd 50 rt 60  
fd 50 rt 60  
fd 50 rt 60  
fd 50 rt 60
```

Skráť program pomocou príkazu **repeat**.

Cvičenie 8

Použi príkaz **repeat** na naprogramovanie štvorca so stranou dĺžky 200.

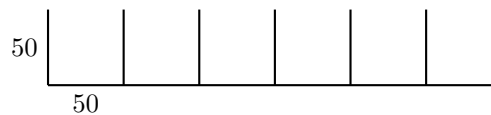
Cvičenie 9

Odpíš nasledujúci program:

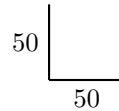
```
fd 100 rt 120  
fd 100 rt 120  
fd 100 rt 120
```

Čo sa stane, keď ho spustíš? Použi príkaz **repeat** na jeho skrátenie.

Chceli by sme nakresliť nasledujúci obrázok pomocou príkazu **repeat**:



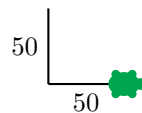
Predtým, než začneme kresliť, si musíme premyslieť, čo za vzor chceme opakovať. Napríklad môžeme použiť ako opakujúci sa vzor nasledujúci obrazec:



Tento obrazec sa dá nakresliť s nasledujúcim programom. Predpokladajme, že začíname v dolnom ľavom rohu:

```
fd 50 bk 50 rt 90 fd 50
```

Po spustení programu je korytnačka v pravom dolnom rohu a pozerá sa doprava:



Všetko, čo musíme teraz urobiť, je otočiť korytnačku tak, aby pozerala dohora. Potom už môžeme znova nakresliť obrazec. Toto vieme docieľiť príkazom **lt 90**.

Teraz spustíme celý program, aby sme si overili ako funguje:

```
fd 50 bk 50 rt 90 fd 50  
lt 90
```

Získali sme požadovaný obrázok:



Ak spustíme program opäť, získame nasledovné:



Takže vidíme, že naša idea funguje a môžeme opakovať program 6-krát:

```
repeat 6 [ fd 50 bk 50 rt 90 fd 50 lt 90 ]
```

Opakovaný vzor Nastavenie korytnačky

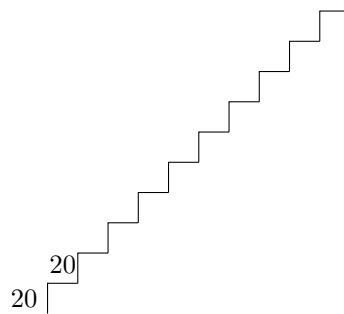
Vela cvičení sa da vyriešiť týmto spôsobom. Vždy si pamätaj, že najprv je treba nájsť opakujúci sa vzor. Potom treba napísať jeden program, ktorý nakreslí *Opakovaný vzor*, a iný program, ktorý *nastaví korytnačku* tak, aby smerovala správnym smerom pre ďalšie opakovanie vzoru. Výsledný program by mal vyzeráť takto:

repeat *Počet opakovaní* [*Opakovaný vzor* *Nastavenie korytnačky*]

Cvičenie 10

Kreslenie schodov.

(a) Nakresli schodisko zložené z 10 schodov veľkosti 20.



- Najprv nájsť opakujúci sa vzor a napísať preň program.
- Premysli si ako napísať program, ktorý otočí korytnačku správnym smerom, aby sa mohol znova nakresliť ďalší vzor.
- Spoj tieto dva programy dohromady a vyrieš úlohu.

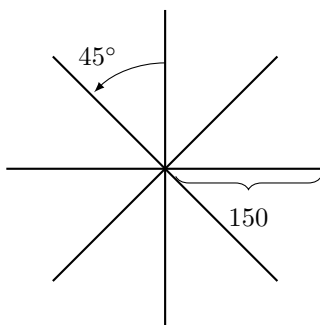
(b) Nakresli schodisko zložené z 5 schodov veľkosti 50.

(c) Nakresli schodisko zložené z 20 schodov veľkosti 10.

Cvičenie 11

Teraz budeme kresliť hviezdy.

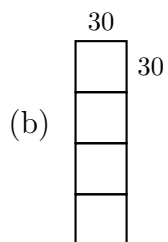
(a) Nakresli túto hviezdu.



(b) Táto hviezda má osem lúčov, každý dĺžky 150 krokov. Dokážeš nakresliť aj hviezdu s 16 lúčmi dĺžky 100?

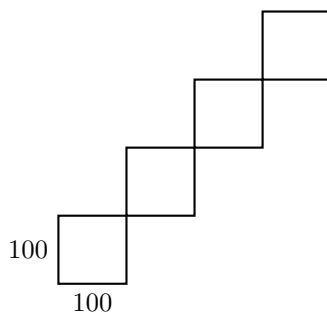
Cvičenie 12

Naprogramuj tieto obrázky:



Cvičenie 13

Naprogramuj tento obrázok:



Cvičenie 14

Odpíš a spusti nasledujúci program:

```
repeat 4 [fd 100 rt 90]
rt 90
repeat 4 [fd 100 rt 90]
rt 90
repeat 4 [fd 100 rt 90]
rt 90
repeat 4 [fd 100 rt 90]
rt 90
```

Čo sa nakreslilo? Dokážeš tento program ešte viac skrátiť?

Prechádzkový mód

Naša korytnačka je zvyčajne v **kresliacom móde**. To znamená, že drží ceruzku a kedykoľvek sa pohne, nakreslí sa čiara.

Naproti tomu sa korytnačka v **prechádzkovom móde** pohybuje bez kreslenia. Prepnúť korytnačku do prechádzkového módu môžeš pomocou príkazu:

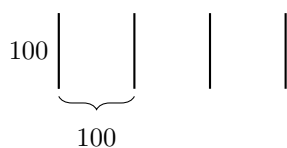
penup alebo skrátené **pu**.

Prepnúť korytnačku späť do kresliaceho módu môžeš pomocou nasledujúceho príkazu:

pendown alebo skrátené **pd**.

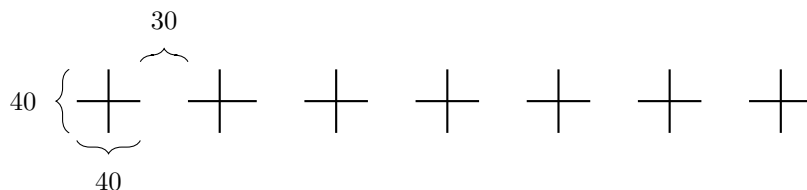
Cvičenie 15

Nakresli tento obrázok pomocou programu:



Cvičenie 16

Naprogramuj nasledujúci obrázok:



3 Ako pomenovať a spustiť program

Každý program, ktorý sme doteraz napísali, môžeme aj pomenovať. Daný program sa následne spustí, ak napíšeme jeho meno do príkazového riadku.

Takto vyzerá program na nakreslenie štvorca s dĺžkou strany 100:

```
repeat 4 [fd 100 rt 90]
```

Tento program môžeme pomenovať **STVOREC100** nasledovne:

```
to STVOREC100
repeat 4 [fd 100 rt 90]
end
```

Rovnaký program sme napísali dvakrát. Raz s a raz bez pomenovania.

Aby sme napísali program s menom, musíme otvoriť **editor**. Takéto programy budú odteraz označené sivou tabuľkou. Akonáhle program v editore dokončíme, musíme kliknúť na tlačítko s korytnačkou, aby sme editor zavreli.

Každý si smie vybrať vlastné meno pre svoj program. My sme sa rozhodli pre **STVOREC100**, lebo sme chceli naznačiť, že daný program nakreslí štvorec s dĺžkou 100. Jediné obmedzenia pre názov programu sú, že sa meno musí skladať z písmen a čísel a musí byť napísané v celku (bez medzier).

Po napísaní programu v editore sa ešte nič nenakreslí. Program sme zatiaľ len pomenovali, ale ešte sme ho nespustili. Ak teraz napíšeme

STVOREC100

do príkazového riadku, spustí sa program **repeat 4 [fd 100 rt 90]**. Na obrazovku sa nakreslí štvorec:



Pozrime sa opäť na Cvičenie 12(a). Jeho riešenie by sme mohli zjednodušiť tak, že najprv napíšeme program pre opakujúci sa motív - štvorec s dĺžkou strany 20 a vhodne ho pomenujeme:

```
to STVOREC20
repeat 4 [fd 20 rt 90]
end
```

Po spustení **STVOREC20** sa korytnačka nachádza v ľavom dolnom rohu štvorca:



Aby sme nakreslili ďalší štvorec, musíme najprv korytnačku premiestniť do pravého dolného rohu. To môžeme urobiť nasledovne:

```
rt 90 fd 20 lt 90
```

Aj tento program pomenujeme:

```
to POZICIA20
rt 90 fd 20 lt 90
end
```

Použitím týchto dvoch programov môžeme napísať program pre Cvičenie 12(a) nasledovne:

```
repeat 10 [STVOREC20 POZICIA20]
```

Výsledný program môžeme taktiež pomenovať. Napríklad:

```
to RAD10
repeat 10 [STVOREC20 POZICIA20]
end
```

Programy **STVOREC20** a **POZICIA20** sa nazývajú **podprogramy** programu **RAD10**.

Cvičenie 17

Vyrieš Cvičenie 12(b) znovu pomocou podprogramu, ktorý kreslí štvorec dĺžky 30. Tvoj finálny program by mal vyzeráť nejako takto:

```
repeat 4 [STVOREC30 POZICIA30]
```

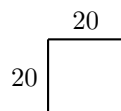
Takže najprv musíš vymyslieť a napísať podprogramy **STVOREC30** a **POZICIA30**.

Cvičenie 18

Vyrieš Cvičenie 13 znova použijúc program **STVOREC100** ako podprogram.

Cvičenie 19

Napíš program, ktorý kreslí schod:



Použi ho ako podprogram a vyrieš s ním Cvičenie 10(a).

Cvičenie 20

Vyrieš Cvičenie 11(a) znova použijúc nasledujúci program ako podprogram:

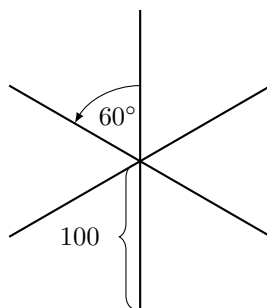
```
to LUC  
fd 150 bk 150  
end
```

Cvičenie 21

Napíš nasledujúci program v editore:

```
to LUCE  
fd 100 bk 200 fd 100  
end
```

Použi program **LUCE** ako podprogram programu **HVIEZDA6**, ktorý nakreslí hviezdu:



Cvičenie 22

Vyrieš Cvičenie 15 a Cvičenie 16 znova pomocou podprogramov.

Cvičenie 23

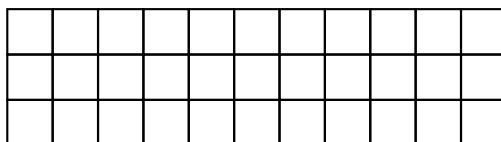
Už sme napísali program **RAD10**. Čo nakreslí nasledujúci program?

```
RAD10 fd 20 lt 90 fd 200 rt 90
```

Over si tvoj návrh pomocou počítača.

Cvičenie 24

Napíš program, ktorý nakreslí nasledujúci obrázok:



Cvičenie 25

Kreslenie štvorcov rôznych veľkostí.

- Napíš program, ktorý nakreslí štvorec dĺžky 50 a nazvi ho **STVOREC50**. Spusti tvoj program, aby si si overil, že správne funguje.
- Napíš program, ktorý nakreslí štvorec dĺžky 75.
- Spusti program

```
STVOREC50  
STVOREC75  
STVOREC100
```

Ako vyzerá výsledný obrázok?

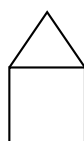
- Ako môžeš zmeniť predchádzajúci program tak, aby sa nakreslili ešte ďalšie tri väčšie štvorce?

Stavanie domov

V nasledujúcom texte chceme pomôcť architektovi pri stavbe sídliska. Aby čo najviac zjednodušil výstavbu, rozhodol sa postaviť všetky domy rovnakým spôsobom. Nato sme mu navrhli nasledujúci rodinný dom:

```
to DOM
rt 90
repeat 4 [fd 50 rt 90]
lt 60 fd 50 rt 120 fd 50 lt 150
end
```

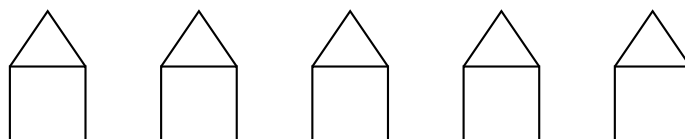
Tento program nakreslí takýto rodinný dom:



Cvičenie 26

Kde sa nachádza korytnačka na začiatku kreslenia domu? Premysli si cestou, ktorou pôjde korytnačka pri kreslení domu používajúc program **DOM**. Kde sa nachádza korytnačka po skončení programu? Zakresli do obrázku vyššie dôsledok každého príkazu (viď Cvičenie 1).

Architekt dal postaviť dom podľa nášho návrhu a je spokojný s výsledkom. Preto chce náš program použiť na postavenie celej ulice. Žiadaná ulica by mala vyzeráť takto:



Keďže všetky domy vyzerajú rovnako, môže použiť program **DOM** päťkrát bez toho, aby musel každý dom znova navrhovať. Nechá korytnačku postaviť prvý dom vľavo a povie jej, aby sa premiestnila do štartovacieho bodu pre výstavbu ďalšieho domu:



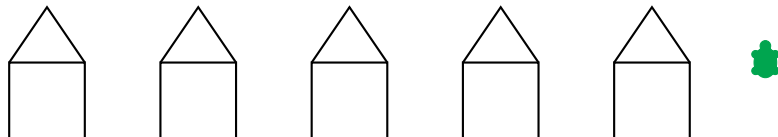
Architekt nato napísal tento program:

```
DOM rt 90 pu fd 50 lt 90 pd
```

Keďže je teraz korytnačka správne umiestnená, môže začať kresliť nasledujúci dom a presunúť sa na pozíciu ďalšieho domu. Tento proces sa opakuje, kým nie je nakreslených všetkých päť domov. Z tohto dôvodu musíme opakovať architektov program päťkrát, aby sme získali ulicu s piatimi domami. Výsledný program nazveme **ULICA**:

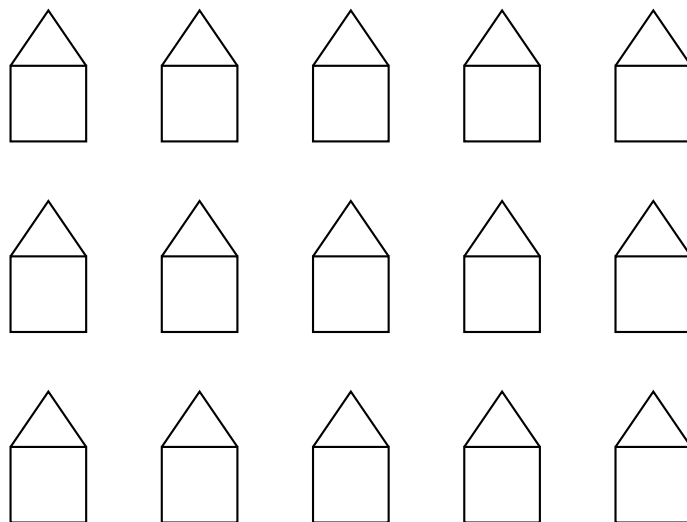
```
to ULICA
repeat 5 [DOM rt 90 pu fd 50 lt 90 pd]
end
```

Po spustení programu sa nachádza korytnačka na mieste, kde môže nakresliť ďalší dom:



Cvičenie 27

Teraz by sme chceli rozšíriť sídlisko o ďalšie ulice. Použi program **ULICA** ako stavebnú jednotku na vystavanie tohto sídliska:



Rada: Po nakreslení ulice sa musí korytnačka premiestniť na správne miesto, aby mohla začať kresliť ďalšiu ulicu.

Ako kresliť hrubé čiary a plné štvorce

Cvičenie 28

Kreslenie hrubých čiar pomocou programu **HRUBA**.
Pomenuj nasledujúci program v editore menom **HRUBA**

```
fd 100  
rt 90  
fd 1  
rt 90  
fd 100  
rt 180
```

a napíš do príkazového riadku

```
HRUBA
```

Čo nakreslí korytnačka? Vezmi si ceruzku a nakresli, ako bol obrázok zhotovený.

Cvičenie 29

Opakuj program **HRUBA** 100-krát pomocou príkazu

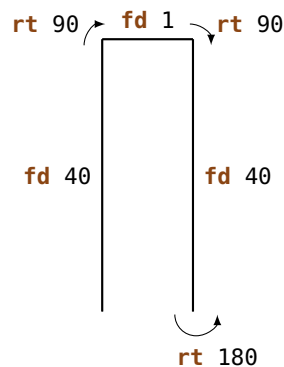
```
repeat 100 [HRUBA]
```

Ako vyzerá výsledný obrázok?

Cvičenie 30

V tomto cvičení budeme kresliť hrubé čiary. Už vieme, že hrubá čiara sa dá nakresliť nasledovne (viď Cvičenie 28):

```
to HRUBA40  
fd 40  
rt 90  
fd 1  
rt 90  
fd 40  
rt 180  
end
```



Dve normálne čiary sú nakreslené tak blízko pri sebe, že to vyzerá ako jedna hrubá čiara. Odpíš program **HRUBA40** a vyskúšaj ho.

Cvičenie 31

Hrubá čiara dĺžky 40 sa dá vnímať aj ako obdĺžnik s hrúbkou 1 a dĺžkou 40. Po spustení **HRUBA40** sa korytnačka pozerá dohora. Ak spustíme program ešte raz, korytnačka prekreslí druhú čiaru. Získame tak obdĺžnik hrúbky 2 a dĺžky 40. Ak zopakujeme **HRUBA40** 40-krát, získame štvorec dĺžky 40. Vyskúšaj to zopakovaním programu **HRUBA40** 40 ráz.

Napíš program s menom **PLNY40**, ktorý nakreslí plný štvorec so stranou dĺžky 40.

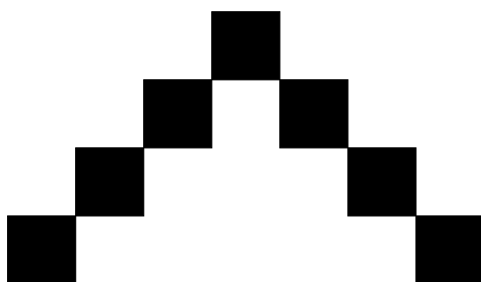
Cvičenie 32

Nakresli tento obrázok pomocou programu **PLNY40**:



Cvičenie 33

Použi program **PLNY40** na nakreslenie tohto obrázka:



Cvičenie 34

Nakresli tento obrázok:



Cvičenie 35

Napíš program, ktorý nakreslí tento obrázok:



Cvičenie 36

Architekt sa rozhodol objednať strechu od iného predajcu. To znamená, že teraz má k dispozícii dve stavebné jednotky: Jednu s menom **STRECHA** a druhú s menom **BAZA**. Napíš dva programy na nakreslenie oboch stavebných blokov. Spoj tieto programy a vytvor nový program **DOM1**, ktorý nakreslí dom.

Cvičenie 37

Rodinné domy v našom sídlisku (pozri Cvičenie 27) sú príliš jednoduché. Skús použiť vlastnú fantáziu a vymysli nový dom. Vystavaj s ním celé nové sídlisko.

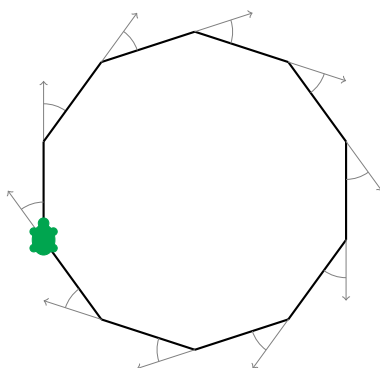
4 Ako nakresliť pravidelné mnohouholníky a kruhy

Pravidelné mnohouholníky

Pravidelný k -uholník je mnohouholník, ktorý má k vrcholov a k strán rovnakej dĺžky. Aby si nakreslil pravidelný 10-uholník ceruzkou, musíš nakresliť 10 strán a po každej strane sa musíš “trochu” potočiť.

O kolko sa musíme potočiť?

Pri kreslení pravidelného mnohouholníka sa korytnačka otáča viackrát, ale nakoniec dosiahne svoju východiskovú pozíciu a pozerá smerom dohora.



To znamená, že počas kreslenia sa otočila o celých 360° . Keď kreslíme pravidelný 10-uholník, musíme ju otočiť presne desaťkrát a to vždy o rovnaký uhol. Preto je uhol, ktorým korytnačku potočíme, rovný:

$$\frac{360^\circ}{10} = 36^\circ$$

A teda musíme každý raz korytnačku potočiť o 36° príkazom: **rt 36**. Podme to vyskúšať pomocou nasledujúceho programu:

```
repeat 10 [ fd 50      rt 36 ]  
           Dĺžka strany  Otočenie o  $36^\circ$ 
```

Cvičenie 38

Nakresli nasledujúce pravidelné mnohouholníky:

- (a) pravidelný 5-uholník so stranou dĺžky 180,
- (b) pravidelný 12-uholník so stranou dĺžky 50,
- (c) pravidelný 4-uholník so stranou dĺžky 200,
- (d) pravidelný 6-uholník so stranou dĺžky 100,
- (e) pravidelný 3-uholník so stranou dĺžky 200,
- (f) pravidelný 18-uholník so stranou dĺžky 20.

Keď chceme nakresliť pravidelný 7-uholník, začneme mať problém, keďže 360 nie je deliteľné siedmimi bezo zvyšku. V takom prípade necháme počítač vyrátať výsledok za nás tým, že napíšeme

```
360/7
```

("/" znamená pre počítač "deliť"). Počítač vyráta výsledok. Preto môžeme nakresliť aj pravidelný 7-uholník so stranou dĺžky 100, a to nasledovne:

```
repeat 7 [fd 100 rt 360/7]
```

Vyskúšaj to.

Kreslenie kruhov

Pomocou príkazov **fd** a **rt** nemôžeme nakresliť naozajstný kruh. Možno si si však všimol/a, že pravidelné mnohouholníky, ktoré majú veľa vrcholov, sa na kruh veľmi podobajú. To znamená, že môžeme kresliť kruhy tak, že nakreslíme mnohouholníky s kratučkými stranami a s veľa vrcholmi.

Cvičenie 39

Vyskúšaj tento program:

```
repeat 360 [fd 1 rt 1]
repeat 180 [fd 3 rt 2]
repeat 360 [fd 2 rt 1]
repeat 360 [fd 3.5 rt 1]
```

3.5 znamená 3 a pol kroku.

Cvičenie 40

- (a) Ako by si nakreslil veľmi malý kruh? Napíš nato program.
- (b) Ako by si nakreslil veľký kruh? Napíš nato program.

Cvičenie 41

Skús nakresliť nasledujúce polkruhy. Veľkosť si smieš zvoliť:



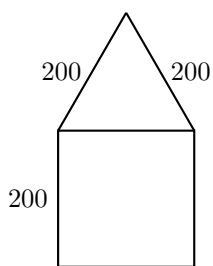
(a)



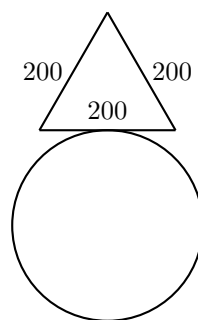
(b)

Cvičenie 42

Použi to, čo si sa naučil/a, na nakreslenie nasledujúcich obrázkov. Veľkosť kruhu si smieš zvoliť:



(a)



(b)

Voľné kreslenie

Nakresli 7-uholník pomocou:

```
repeat 7 [fd 100 rt 360/7]
```

Potom otoč korytnačku o 10° doprava pomocou príkazu

```
rt 10
```

Opakuj oba programy niekoľkokrát a pozri sa na výsledný obrázok. Po každom nakreslení 7-uholníka otočíme korytnačku o 10° pomocou **rt 10**. Ak ju chceme vrátiť do jej vychádzajúcej pozície, musíme celý proces opakovať

$$\frac{360^\circ}{10^\circ} = 36$$

krát. Preto sa pozrime na to, čo vytvorí nasledujúci program:

```
repeat 36 [repeat 7 [fd 100 rt 360/7] rt 10]
```

Cvičenie 43

Nakresli pravidelný mnohoúhelník s 12 vrcholmi a so stranou dĺžky 70. Otoč sa 18-krát a vždy nakresli 12-uholník, kým sa dostaneš do štartovacej pozície.

Poznámka: Najprv môžeš naprogramovať 12-uholník so stranou dĺžky 70 a nazvať program napríklad **UHOLNIK12**. Potom už len musíš dokončiť nasledujúci program:


















```
repeat 18 [UHOLNIK12 rt ... ]
```

Cvičenie 44

Vymysli si podobný obrazec (viď Cvičenie 43). Napíš program, ktorý ho nakreslí.

Farby

Na kreslenie pekných obrázkov potrebujeme aj rôzne farby. Korytnačka dokáže okrem čiernych čiar kresliť čiary aj v mnohých iných farbách. Každé číslo je priradené číslo. Nasledujúca tabuľka obsahuje zoznam farieb na výber:

0		5		9		13	
1		6		10		14	
2		7		11		15	
3		8		12		16	
4							

Pomocou príkazu

setpencolor	X
Príkaz na zmenu farby	Číslo žela- nej farby

zmení korytnačka práve používanú farbu pera na farbu prislúchajúcu k číslu **X**. Rovnako môžeme použiť aj kratšiu verziu príkazu: **setpc**.

Pomocou farieb môžeme kresliť úžasné vzory ako napríklad vzor, ktorý je nakreslený nasledovným programom. Najprv napíšeme dva programy s menom, ktoré kreslia kruhy s rôznou veľkosťou.

```
to KRUH3
repeat 360 [fd 3 rt 1]
end

to KRUH1
repeat 360 [fd 1 rt 1]
end
```

Teraz použijeme tieto programy na vytvorenie vzorov podobných tým, ktoré sme už videli:

```
to VZOR3
repeat 36 [KRUH3 rt 10]
end

to VZOR1
repeat 18 [KRUH1 rt 20]
end
```

A teraz to vyskúšajme aj s farbami:

```
setpc 2
VZOR3 rt 2
setpc 3
VZOR3 rt 2

setpc 4
VZOR3 rt 2
setpc 5
VZOR3 rt 2
```

```
setpc 6
VZOR1 rt 2
setpc 15
VZOR1 rt 2
```

```
setpc 8
VZOR1 rt 2
setpc 9
VZOR1 rt 2
```

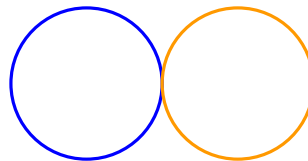
Neváhaj pokračovať a pridať ešte ďalšie vzory. Alebo si vymysli úplne nový vzor.

Cvičenie 45

Použi program **VZOR3** s oranžovou farbou. Potom zmeň pomocou príkazu **setpc 7** farbu na bielu. Čo sa stane, ak znova spustíš program **VZOR3**?

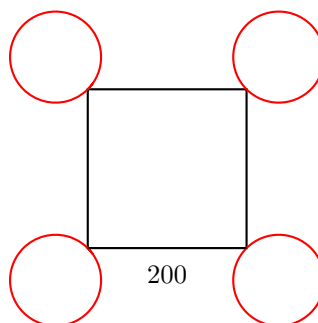
Cvičenie 46

Nakresli nasledujúci obrázok. Korytnačka sa na začiatku nachádza v priesečníku kruhov.



Cvičenie 47

Naprogramuj nasledujúci obrázok. Veľkosť kruhu si smieš zvoliť.



Slovo **:VRCHOLOV** nazývame **parameter**. V horeuvedenom príklade sa hodnoty 3, 4, 5 a 6 nazývajú **hodnoty parametra :VRCHOLOV**. Počítač vie, že je to parameter vďaka **:**. Preto všade, kde chceme použiť parameter, musíme napísať **:** pred jeho meno.

Cvičenie 48

Každý z týchto programov nakreslí štvorec rôznej veľkosti.

```
repeat 4 [fd 100 rt 90]
repeat 4 [fd 50 rt 90]
repeat 4 [fd 200 rt 90]
```

Žlté čísla 100, 50, 200 sa môžu vnímať ako hodnoty parametra, ktorý určuje veľkosť štvorca. Napíš program s parametrom **:VELKOST**, ktorý kreslí štvorce ľubovoľnej veľkosti:

```
to STVOREC :VELKOST
...
end
```

Cvičenie 49

Nasledovné programy kreslia kruhy rôznych veľkostí:

```
repeat 360 [fd 1 rt 1]
repeat 360 [fd 12 rt 1]
repeat 360 [fd 3 rt 1]
```

Napíš program s parametrom, ktorý vie nakresliť kruh ľubovoľnej veľkosti. Vyskúšaj si svoj program použijúc 1, 2, 3, 4 a 5 ako rôzne hodnoty parametra. Meno parametra si môžeš vybrať, ale nezabudni pred meno parametra napísať dvojbodku.

Cvičenie 50

Spomínaš si ešte, ako sa kreslia hrubé čiary (Cvičenie 28)? Napíš program s parametrom, ktorý dokáže nakresliť hrubú čiaru ľubovoľnej dĺžky.

Rada: Môžeš začať tým, že napíšeš programy, ktoré kreslia hrubé čiary dlhé 100 a 50 krokov, aby si zistil/a, čo by mal byť parameter programu.

Cvičenie 51

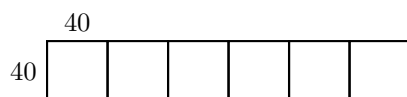
Napíš program s parametrom, ktorý vie nakresliť rovnostranný trojuholník ľubovoľnej veľkosti. Použi svoj program na nakreslenie trojuholníkov veľkosti

20, 40, 60, 80, 100, 120, 140, 160 a 180

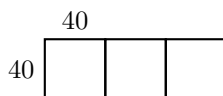
jeden za druhým. Ako vyzerá výsledný obrázok?

Cvičenie 52

A teraz by sme chceli kresliť štvorce veľkosti 40, jeden vedľa druhého. Napíš program **STVORCE** s parametrom **:KOLKO**. Parameter **:KOLKO** určuje, koľko štvorcov bude nakreslených. To znamená, že po spustení **STVORCE 6** nakreslí korytnačka tento obrázok:

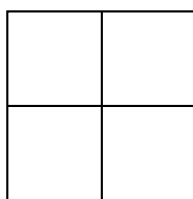


Takýto obrázok sa nakreslí príkazom **STVORCE 3**:



Cvičenie 53

Napíš program, ktorý nakreslí nasledujúci obrázok zložený zo 4 štvorcov. Veľkosť štvorca je určená parametrom.

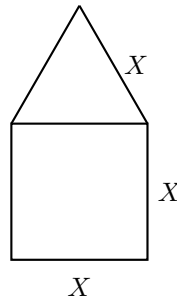


Cvičenie 54

Napíš program s parametrom, ktorý vie nakresliť pravidelný šesťuholník ľubovoľnej veľkosti. Vyskúšaj svoj program nakreslením šesťuholníkov veľkosti 40, 60 a 80.

Cvičenie 55

Napíš program s parametrom **:X**, ktorý vie nakresliť nasledujúci dom:



Programy s viacerými parametrami

Program môže mať viac ako jeden parameter. Keď kreslíme napríklad mnohouholníky, môže mať náš program jeden parameter **:VRCHOLOV**, určujúci množstvo vrcholov, a druhý parameter **:VELKOST**, určujúci veľkosť strany.

V nasledujúcich programoch je parameter **:VRCHOLOV** označený žltou a parameter **:VELKOST** zelenou farbou:

repeat 13 [fd 100 rt 360/13]	nakreslí 13-uholník dĺžky 100
repeat 3 [fd 300 rt 360/3]	nakreslí 3-uholník dĺžky 300
repeat 17 [fd 10 rt 360/17]	nakreslí 17-uholník dĺžky 10
repeat 60 [fd 3 rt 360/60]	nakreslí 60-uholník dĺžky 3

Preto môžeme teraz napísať program s dvoma parametrami, ktorý vie nakresliť ľubovoľný mnohouholník:

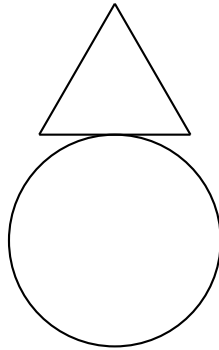
```
to MNOHO :VRCHOLOV :VELKOST
repeat :VRCHOLOV [fd :VELKOST rt 360/:VRCHOLOV]
end
```

Vyskúšaj program **MNOHO** a doplň, čo sa nakreslí:

MNOHO 12 60	nakreslí	dĺžky
MNOHO 12 45	nakreslí	dĺžky
MNOHO 8 30	nakreslí	dĺžky
MNOHO 9 30	nakreslí	dĺžky
MNOHO 7 31	nakreslí	dĺžky
MNOHO 11 50	nakreslí	dĺžky

Cvičenie 56

Napíš program s dvoma parametrami, ktorý vie nakresliť nasledujúci obrázok. Veľkosť kruhu ako aj veľkosť trojuholníka je voliteľná pomocou parametrov.



Cvičenie 57

Program

```
fd 100 rt 90 fd 200 rt 90 fd 100 rt 90 fd 200
```

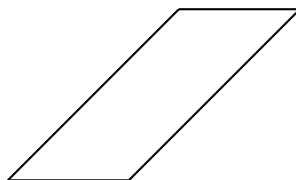
nakreslí obdĺžnik, ktorý je 100 krokov vysoký a 200 krokov široký. Odpíš program a over, či to naozaj funguje. Napíš iný program s dvoma parametrami, ktorý vie nakresliť obdĺžnik ľubovoľnej výšky a šírky.

Cvičenie 58

Nasledujúci program:

```
repeat 2 [rt 45 fd 200 rt 45 fd 100 rt 90]
```

nakreslí rovnobežník:



Napíš program s dvoma parametrami, ktorý vie nakresliť takýto rovnobežník ľubovoľnej veľkosti.

Cvičenie 59

Nakresliť kvetinku môžeš nasledovne: Začni s nakreslením kruhu pomocou

MNOHO 360 2

potom otoč korytnačku trochu doprava

rt 20

a nakresli ďalší kruh

MNOHO 360 2

Opakuj to niekoľkokrát: **rt 20 MNOHO 360 2 rt 20 MNOHO 360 2 ...**

Po dokončení kvetu by mala byť korytnačka vo svojej východiskovej polohe. Korytnačka nakreslila 18 kruhov a pritom sa obrátila o 20° medzi každým z nich. Spolu urobila jednu celú otočku: $18 \cdot 20^\circ = 360^\circ$.

Celý program môžeme napísať takto:

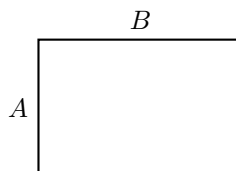
repeat 18 [MNOHO 360 2 rt 20]

Vyskúšaj ho.

- Môžeš tiež nakresliť kvetinku s 10 alebo dokonca 20 okvetnými lístkami (kruhmi). Ako to urobíš? Napíš nato program a vyskúšaj ho.
- Dokážeš napísať program s parametrom, ktorý nakreslí kvetinku s ľubovoľným počtom okvetných lístkov (kruhov)?
- Dokážeš napísať program, ktorý používa nasledujúce hodnoty ako parametre:
 - počet okvetných lístkov (kruhov) a
 - veľkosť kruhov?

Cvičenie 60

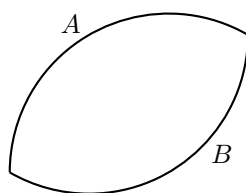
Napíš program, ktorý nakreslí ľubovoľnou farbou ľubovoľný obdĺžnik:



To znamená, že výška A a šírka B ako aj farba pera sú voliteľné pomocou parametrov.

6 Ako kresliť kvety a preniesť parametre do podprogramov

Na lístok



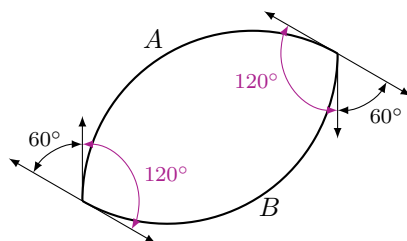
sa dá pozeráť ako na dva oblúky A a B , ktoré sú spolu spojené. Na nakreslenie lístka môžeme použiť nasledujúci program:

```
repeat 120 [fd 2 rt 1]
```

Vyskúšaj ho.

Všimni si, že program je veľmi podobný programu, ktorý kreslí kružnice. Namiesto toho, aby urobil 360 malých krokov s malými otočkami medzi nimi, však opakujeme `[fd 2 rt 1]` len 120-krát, a preto nakreslí korytnačka len tretinu kruhu ($\frac{360^\circ}{3} = 120^\circ$).

Otázkou zostáva, o koľko musíme korytnačku otočiť, než bude môcť začať kresliť druhý oblúk B , ktorý tvorí spodnú časť listu. Poďme sa pozrieť na nasledujúci obrázok:



Ak chceme dosiahnuť korytnačkinu východiskovú pozíciu po vykreslení celého lístka, budeme musieť otočiť korytnačku celkom o 360° . Pri kreslení časti A , otočíme korytnačku o 120° a pri kreslení časti B ju otočíme o ďalších 120° . Z tohto dôvodu je zostávajúci uhol

$$360^\circ - 120^\circ - 120^\circ = 120^\circ.$$

120° rozdelíme rovným dielom medzi dve otočenia na špicoch lístku:

$$\frac{120^\circ}{2} = 60^\circ.$$

Napokon sme získali tento program:

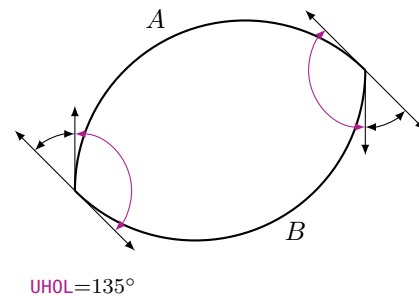
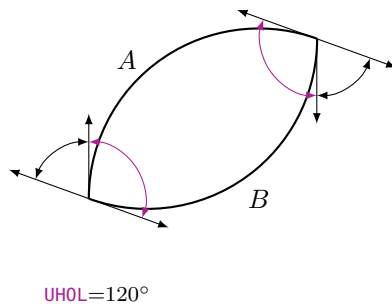
```
repeat 120 [fd 2 rt 1]
rt 60
repeat 120 [fd 2 rt 1]
rt 60
```

alebo ešte jednoduchšie:

```
repeat 2 [repeat 120 [fd 2 rt 1] rt 60]
```

Vyskúšaj to.

teraz by sme radi kreslili aj užšie lístky (kde sú oblúky *A* a *B* kratšie) alebo aj širšie lístky (kde sú oblúky *A* a *B* dlhšie).



Nato opäť použijeme program s parametrom.

Nazveme parameter napríklad **:UHOL**. Uhol otočenia na špici listu možno potom vypočítať takto: Pred začatím kreslenia časti *B* musí korytnačka dokončiť polovicu celkovej otočky, to znamená $\frac{360^\circ}{2} = 180^\circ$. Uhol otočenia na špici listu je teda rovný

$$180^\circ - \text{:UHOL}.$$

Teraz už môžeme napísať náš program v editore:

```
to LISTOK :UHOL
repeat 2 [repeat :UHOL [fd 2 rt 1] rt 180-:UHOL]
end
```

Vyskúšaj program napísaním do príkazového riadku:

```
LISTOK 20  
LISTOK 40  
LISTOK 60  
LISTOK 80  
LISTOK 100
```

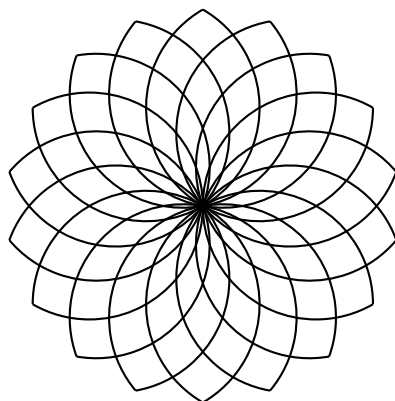
Čo sa stalo?

Cvičenie 61

Kreslenie podľa fantázie.

Začni tým, že nakreslíš kvetinku pomocou nasledujúceho programu:

```
LISTOK 100  
rt 20  
LISTOK 100  
rt 20  
LISTOK 100  
....
```



Kolkokrát musíš zopakovať príkazy **LISTOK 100** a **rt 20**, aby si kvetinku dokončil/a?

Napíš program pre kvetinku v jedinom riadku pomocou vhodného **repeat** príkazu. (Nazabúdaj, že všetky **rt** medzi kreslením jednotlivých lístkov musia dať dohromady 360°).

Cvičenie 62

Príkaz **fd** 2 v programe **LISTOK** určuje veľkosť kruhu, z ktorého vykresľujeme oblúk. Hodnotu 2 môžeme nahradiť parametrom **:VELKOST**. Napíš program

```
LIST :UHOL :VELKOST
```

s dvoma parametrami **:UHOL** a **:VELKOST**, aby sme mohli ľubovoľne určiť šírku ako aj veľkosť lístka. Vyskúšaj svoj program **LIST** nasledovne:

```
LIST 100 1
LIST 100 1.5
rt 100
LIST 80 2
LIST 80 2.5
```

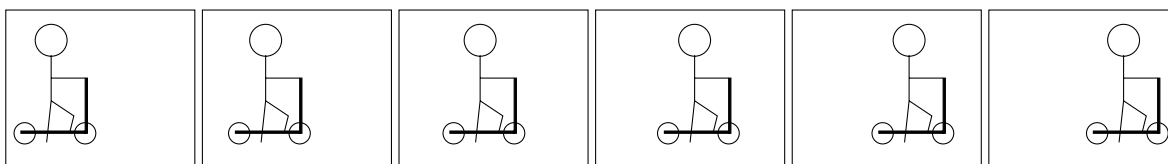
Potom otoč korytnačku o 80° doprava a zopakuj celý program znova.

Cvičenie 63

Vymysli si vlastné vzory, ktoré môžeš nakresliť.

7 Ako programovať animácie

Vieš ako sa vyrábajú kreslené filmy? Najprv nakreslíš niekoľko obrázkov, každý len trochu odlišný od predošlého. Napríklad na nasledujúcich obrázkoch je chlapec na kolobežke posunutý vždy trochu viac doprava:



Keď položíš obrázky jeden na druhý a rýchlo ich prebehneš prstom, bude to vyzerať tak, že sa chlapec naozaj hýbe zľava doprava. Toto nazývame aj **Animácia**.

V tejto Lekcii sa naučíme, ako programovať animácie pomocou korytnačky.

Ako nakresliť štvorec, ktorý zanecháva stopy

Pre našu prvú animáciu sme si zvolili postavičku, ktorá nie je príliš zložitá a ktorú poznáme už dlhšiu dobu: Budeme posúvať štvorec zľava doprava.



Program, ktorý nakreslí štvorec, už dobre poznáme:

```
to STVOREC100
repeat 4 [fd 100 rt 90]
end
```

Akonáhle je štvorec nakreslený, posunieme korytnačku trochu doprava a nakreslíme ďalší štvorec. Opakujeme to niekoľkokrát.

Nasledujúci program nakreslí takýto štvorec 120-krát:

```
to STVORECVP0HYBE
repeat 120 [STVOREC100 rt 90 fd 4 lt 90]
end
```

Cvičenie 64

Napíš programy `STVOREC100` a `STVORECVP0HYBE` v Editore a vyskúšaj `STVORECVP0HYBE`.
Čo sa nakreslilo?

Ako vidíš, nakreslili sa stopy *všetkých* štvorcov. Ale pre našu animáciu chceme vidieť vždy len posledný štvorec a chceme zmazať stopy predchádzajúcich štvorcov.

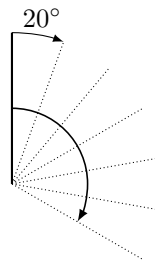


Cvičenie 65

Posuň štvorec zdola nahor namiesto zľava doprava.

Cvičenie 66

Napíš program, ktorý kreslí úsečku dĺžky 20. Použi tento program ako podprogram na otočenie úsečky okolo jej dolného konca v smere hodinových ručičiek:



Ako nakresliť štvorec a ako ho opäť vygumovať

Aby sme zmazali stopy, musíme sa najprv naučiť vygumovať obrázky, ktoré boli práve nakreslené. Nato si musí vziať korytnačka namiesto pera gumu. Korytnačka prechádza z **kresliaceho módu** do **gumovacieho módu** pomocou nového príkazu **penerase** alebo skrátene **pe**.

Cvičenie 67

Premysli si, čo robí nasledujúci program `STVOREC100 pe STVOREC100` bez použitia počítača.

Aby začala korytnačka opäť kresliť, musíme použiť príkaz: **penpaint** alebo skrátene **ppt**. Nový príkaz môžeme rovno použiť (viď Cvičenie 67).

Náš program teraz vyzerá nasledovne:

```
STVOREC100 pe STVOREC100 ppt
```

Cvičenie 68

Vyskúšaj vyššie napísaný program. Čo sa stane? Vieš to vysvetliť?

Ako nechať štvorec chvíľku čakať

Ako nám už (viď Cvičenie 68) ukázalo, štvorec sa hneď po nakreslení vygumuje. Ani si neuvedomíme, že sa nejaký štvorec nakreslil. Preto musíme pred jeho vygumovaním nechať korytnačku chvíľku stáť.

To sa dá spraviť nasledovne:

<code>wait</code>	4
príkaz na čakanie	čas čakania

Cvičenie 69

Vyskúšaj takýto program:

```
STVOREC100 wait 4 pe STVOREC100 ppt
```

Ako pohnúť štvorec zľava doprava

Teraz už vieme použiť príkazy na čakanie korytnačky a vygumovanie štvorca do našej animácie **STVORECVPOHYBE**:

```
to STVORECVPOHYBE
repeat 120 [STVOREC100 wait 4 pe STVOREC100 rt 90 fd 4 lt 90 ppt]
end
```

Vyskúšaj to. Ak ti korytnačka počas animácie prekáža, použi na začiatku programu príkaz **hideturtle** (alebo skráteno: **ht**), ktorý spraví korytnačku neviditeľnou. Taktiež si všimneš, že sa animácia zrýchlila. Ukonči program príkazom **showturtle** (alebo skratkou: **st**) priamo pred príkazom **end**, čo spraví korytnačku opäť viditeľnou.

Cvičenie 70

Posuň štvorec veľkosti 50×50 dohora.

Cvičenie 71

Zmeň program **STVORECVPOHYBE** tak, aby sa štvorec hýbal dvakrát tak rýchlo doprava ako predtým.

Cvičenie 72

Dokážeš zmeniť program **STVORECVPOHYBE** tak, aby sa štvorec hýbal dvakrát pomalšie doprava ako predtým?

Cvičenie 73

Zmeň program **STVORECVPOHYBE** tak, aby sa štvorec hýbal zprava doľava namiesto zľava doprava.

Cvičenie 74

Najprv si premysli, čo urobí nasledujúci program:

```
to STVORECVPOHYBE1
ht
repeat 50 [STVOREC100 wait 5 pe STVOREC100 fd 3 rt 90 fd 3 lt 90 ppt]
STVOREC100
st
end
```

Potom si over svoj predpoklad na počítači.

Cvičenie 75

Najprv si premysli, čo urobí nasledujúci program:

```
to DOKOLA
ht
repeat 360 [STVOREC100 wait 4 pe STVOREC100 fd 5 rt 1 ppt]
STVOREC100
st
end
```

Potom si over, či si mal pravdu pomocou počítača.

Cvičenie 76

Zmeň program **DOKOLA** tak, aby sa štvorec otáčal štyrikrát rýchlejšie ako predtým.

Cvičenie 77

Čo urobí nasledujúci program?

```
repeat 6 [DOKOLA]
```

Cvičenie 78

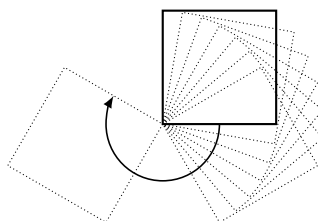
Pozri sa na nasledujúci program

```
to ZEM  
repeat 45 [fd 16 rt 8]  
end
```

Použi ho na vytvorenie animácie, v ktorej obieha Zem okolo Slnka. Slnko môžeš znázorniť podľa vlastnej fantázie.

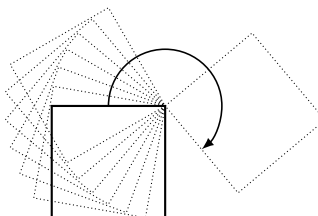
Cvičenie 79

Otáčaj štvorec v smere hodinových ručičiek okolo jeho ľavého dolného rohu. Veľkosť si smieš zvoliť:



Cvičenie 80

A teraz otáčaj štvorec okolo jeho pravého horného rohu:



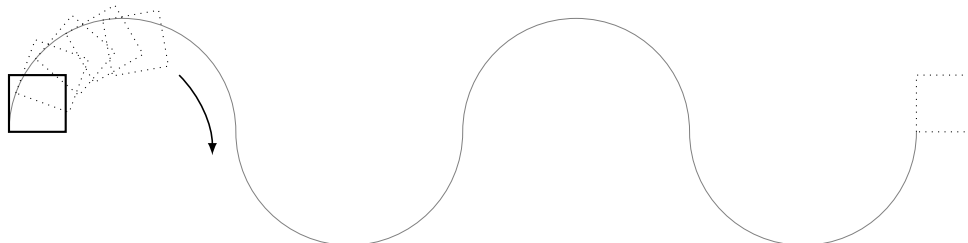
Ak už poznáš parametre (viď Lekcia 5), skús vypracovať nasledujúce cvičenia.

Cvičenie 81

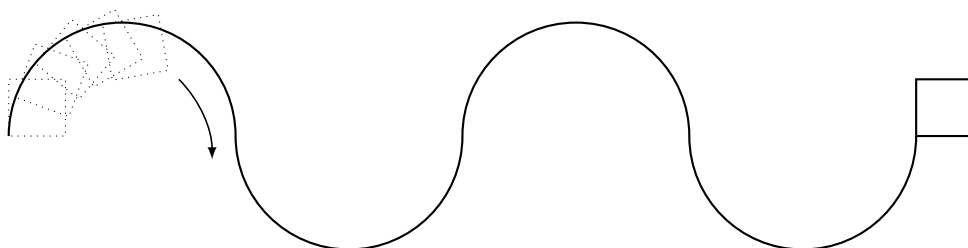
Napiš program s *dvoma parametrami*, ktorý posúva štvorec zľava doprava. Prvý parameter určuje veľkosť štvorca, druhý parameter zas určuje jeho rýchlosť.

Cvičenie 82

- (a) Nechaj štvorec prechádzať po ceste nakreslenej nižšie, ktorá sa skladá zo štyroch polkružníc. Veľkosť štvorca je určená parametrom.



- (b) Teraz by sme chceli nakresliť aj cestu, pozdĺž ktorej sa štvorec hýbe.



- (c) Dokážeš program v (b) vylepšiť tak, že aj počet polkružníc bude zadaný pomocou parametra?

Prehľad príkazov

- fd 100** chod' 100 krokov vpred
- bk 50** chod' 50 krokov vzad
- cs** všetko zmaž a začni od začiatku
- rt 90** otoč sa o 90 stupňov doprava
- lt 180** otoč sa o 180 stupňov doľava
- repeat 4 [...]** program v [...] sa štyrikrát zopakuje
 - pu** korytnačka prejde do chodiaceho módu
 - pd** korytnačka sa vráti do kresliaceho módu
- setpc 3** zmení farbu pera na farbu číslo 3
- to MENO** vytvorí program s menom
- to MENO :PARAMETER** vytvorí program s menom a parametrom
 - end** všetky programy s menom končia týmto príkazom
 - pe** korytnačka prejde do gumovacieho módu
 - ppt** korytnačka prejde do kresliaceho módu z gumovacieho módu
 - wait 5** pozastaví korytnačku na 5 jednotiek času



Programovanie s LOGO-m

Chair of Information Technology and Education
ETH Zurich, CAB F 15.1
Universitätstrasse 6
CH-8092 Zurich
Switzerland

www.ite.ethz.ch
www.abz.inf.ethz.ch