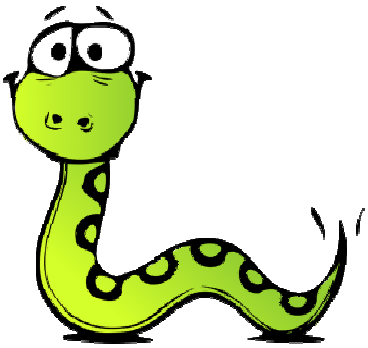
Základy programování v jazyce Python pro střední školy

Metodický list pro učitele

Lekce 14 – Elipsy a kruhy



Andrej Blaho

Ľubomír Salanci

Václav Šimandl

beta verze - 2019

Cíle lekce

- Naučit se kreslit elipsy a kruhy (jako speciální případ elips)
- Seznámit se s pojmem **parametry příkazů**
- Pochopit, jak souvisí kreslení elips s kreslením obdélníků
- Trénovat určování a odvozování vzájemných poloh kruhů, obdélníků a textů

Dovednosti

- Kreslení náčrtů na papír při uvažování a odvozování vzorců pro vzájemnou polohu

Osvojená syntaktická pravidla

- Zápis příkazu `create_oval` a jeho parametrů

Poznámky

- Téma kreslení kruhů je zařazené po náročné trojici témat zaměřených na cykly, aby žáci měli dostatečný prostor na procvičení si cyklů a ujasnění získaných poznatků.

Průběh výuky

Začínáme úlohou na opakování:

1. Napiš program `soucet_99.py`, který pomocí cyklu zjistí, jaký je součet čísel `0 + 1 + 2 + ... + 99`. Výsledek program vypíše pomocí příkazu `print`.

Řešení:

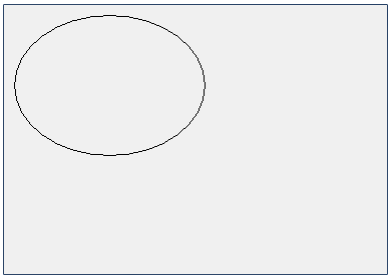
```
soucet = 0
for i in range(100):
    soucet = soucet + i
print('Součet je', soucet)
```

2. V jazyce Python kreslíme elipsy a kruhy příkazem `create_oval`. Vytvoř nový program `elipsa.py` a zapiš do něj následující kód:

```
import tkinter

canvas = tkinter.Canvas()
canvas.pack()

canvas.create_oval(10, 10, 200, 150)
```

Vyzkoušej, co program nakreslí.

Program nakreslí:



3. Přidej na konec programu příkaz pro kreslení obdélníku se stejnými čísly, jako jsou v příkazu `create_oval`. Jaká bude vzájemná pozice elipsy a obdélníku?

   Čísla, která píšeme do závorek v příkazech `canvas.create_oval` a `canvas.create_rectangle`, nazýváme parametry:

   ```
   canvas.create_rectangle(x1, y1, x2, y2)
   canvas.create_oval(x1, y1, x2, y2)
   ```

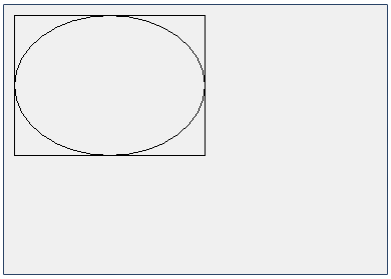
   V příkazu `create_rectangle` určovaly dvojice $[x_1, y_1]$, $[x_2, y_2]$ souřadnice protilehlých vrcholů kresleného obdélníku. V příkazu `create_oval` určují dvojice $[x_1, y_1]$, $[x_2, y_2]$ souřadnice protilehlých vrcholů obdélníku, do kterého se vepíše elipsa. Obdélník se však nenakreslí.

Řešení:

```
import tkinter

canvas = tkinter.Canvas()
canvas.pack()
canvas.create_oval(10, 10, 200, 150)
canvas.create_rectangle(10, 10, 200, 150)
```

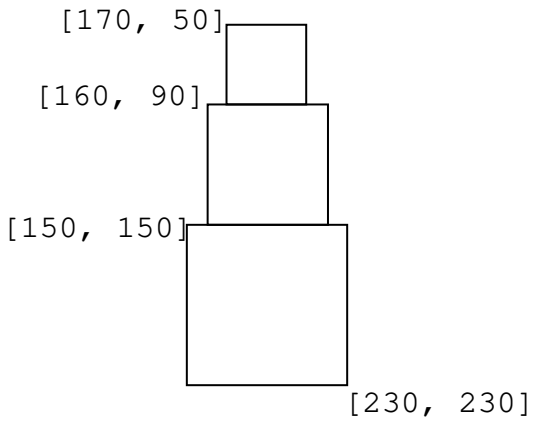
Očekávaná odpověď na otázku „Jaká bude vzájemná pozice elipsy a obdélníku?“ je „Elipsa se dotýká všech stran obdélníku.“

Program nakreslí:



V následující úloze je potřeba dopočítat souřadnice vrcholů čtverců.

4. Pomocí čtverců je možné nakreslit věž z kostek. Vytvoř nový program `vez.py` a napiš do něj kód, který ji nakreslí. Při kreslení využij souřadnice z následujícího obrázku:



Na obrázku v zadání si všimněme, že máme k dispozici y-ové souřadnice všech vrcholů čtverců, z čehož umíme dopočítat výšky čtverců:

- První čtverec má souřadnice vrcholů `[170, 50], [?, 90]` ... výška čtverce je `90 - 50 = 40`
- Druhý čtverec má souřadnice vrcholů `[160, 90], [?, 150]` ... výška čtverce je `150 - 90 = 60`
- Třetí čtverec má souřadnice vrcholů `[150, 150], [230, 230]` ... zde není potřeba nic zjišťovat.

Na základě výše uvedeného známe i šířky prvních dvou čtverců a snadno dopočítáme neznámé `x`-ové souřadnice vrcholů těchto čtverců.

Řešení:

```
import tkinter

canvas = tkinter.Canvas()
canvas.pack()

canvas.create_rectangle(170, 50, 210, 90)
canvas.create_rectangle(160, 90, 220, 150)
canvas.create_rectangle(150, 150, 230, 230)
```

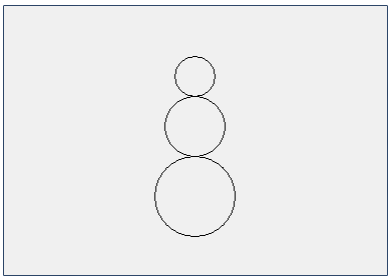
V další úloze očekáváme, že žáci upraví kód předchozího programu, v němž nahradí příkazy `create_rectangle` za `create_oval`.

5. Diskutuj se sousedem, jak nakreslit kruh. Potom změň předchozí program tak, aby se místo věže kreslil sněhulák.

Řešení:

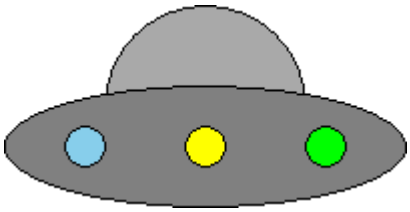
```
import tkinter

canvas = tkinter.Canvas()
canvas.pack()

canvas.create_oval(170, 50, 210, 90)
canvas.create_oval(160, 90, 220, 150)
canvas.create_oval(150, 150, 230, 230)
```

Program nakreslí:



Následují úlohy na trénování:

6. Napiš program `ufo.py`, který pomocí alespoň pěti elips nakreslí UFO. Rozměry i barvy zvol dle svého uvážení. Inspirovat se můžeš (ale nemusíš) na následujícím obrázku:



Barevné elipsy se kreslí podobně jako barevné obdélníky pomocí parametru `fill`:

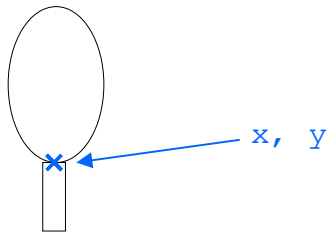
`canvas.create_oval(`x_1`, `y_1`, `x_2`, `y_2`, fill='`*barva*`')`

Řešení:

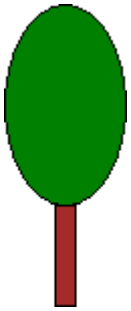
```
import tkinter

canvas = tkinter.Canvas()
canvas.pack()

canvas.create_oval(150, 60, 250, 160, fill='darkgray')
canvas.create_oval(100, 100, 300, 160, fill='gray')
canvas.create_oval(130, 120, 150, 140, fill='skyblue')
canvas.create_oval(190, 120, 210, 140, fill='yellow')
canvas.create_oval(250, 120, 270, 140, fill='lime')
```

Souřadnice, tvary, umístění a barvy mohou být v žákovském řešení jiné než ve vzorovém řešení. Stejně tak žáci nemusí kreslit přesné kruhy. Žáci mohou při řešení postupovat metodou pokusů.

7. Vytvoř nový program `strom.py` a v něm podprogram `strom`, který do proměnných `x`, `y` přiřadí čísla `200`, `150` a pomocí elipsy a obdélníku nakreslí strom. Proměnné `x`, `y` představují souřadnice středu horní strany obdélníku (viz následující obrázek).



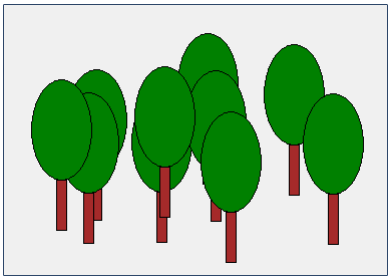
Korunu stromu nakresli jako zelenou elipsu se šířkou 60 a výškou 100. Kmen bude představován hnědým obdélníkem, který bude široký 10 a vysoký 50. Pozor, mezi kmenem a korunou by neměla být mezera (viz obrázek níže).



Řešení:

```
import tkinter

canvas = tkinter.Canvas()
canvas.pack()

def strom():
    x = 200
    y = 150
    canvas.create_rectangle(x - 5, y, x + 5, y + 50,
        fill='brown')
    canvas.create_oval(x - 30, y - 100, x + 30, y,
        fill='green')

strom()
```

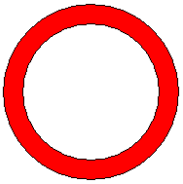
8. Uprav předchozí program tak, aby se kreslil les. V podprogramu `strom` přiřaď do proměnných `x`, `y` náhodné souřadnice a zavolej tento podprogram desetkrát.



Řešení:

```
import tkinter
import random

canvas = tkinter.Canvas()
canvas.pack()

def strom():
    x = random.randint(50, 330)
    y = random.randint(100, 210)
    canvas.create_rectangle(x - 5, y, x + 5, y + 50,
        fill='brown')
    canvas.create_oval(x - 30, y - 100, x + 30, y,
        fill='green')

for i in range(10):
    strom()
```

9. Diskutuj se sousedem, jak nakreslit kruh, jestliže znáš jeho střed a poloměr. Potom vytvoř nový program `znacka.py`, který pomocí příkazu `canvas.create_oval` nakreslí dopravní značku *Zákaz vjezdu* (viz obrázek níže). Značka bude tvořena dvěma soustřednými kruhy, jejichž společný střed bude mít souřadnice `[200, 100]`. Velký červený kruh bude mít poloměr `45` a bílý kruh bude mít poloměr `35`.



Řešení:

```
import tkinter
import random

canvas = tkinter.Canvas()
canvas.pack()

canvas.create_oval(200 - 45, 100 - 45, 200 + 45, 100 + 45,
    fill='red')
canvas.create_oval(200 - 35, 100 - 35, 200 + 35, 100 + 35,
    fill='white')
```

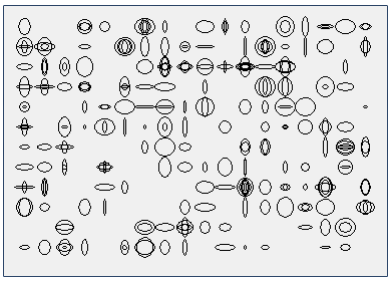
V další úloze očekáváme, že žáci upraví předchozí program, do něhož přidají příkazy pro nakreslení textu.

10. Uprav předchozí program tak, aby nakreslil dopravní značku *Průjezd zakázán* (viz obrázek níže). Tato značka se od značky *Zákaz vjezdu* liší jen nápisem uvnitř bílého kruhu.

Řešení:

```
import tkinter
import random

canvas = tkinter.Canvas()
canvas.pack()

canvas.create_oval(200 - 45, 100 - 45, 200 + 45, 100 + 45,
    fill='red')
canvas.create_oval(200 - 35, 100 - 35, 200 + 35, 100 + 35,
    fill='white')
canvas.create_text(200, 100 - 7, text='PRŮJEZD')
canvas.create_text(200, 100 + 7, text='ZAKÁZÁN')
```

11* Mimozemšťané nám poslali následující zprávu:



Zřejmě očekávají, že jim odpovíme podobně vypadající zprávou. Napiš program `ufo_zprava.py`, který takovou (byť náhodnou) zprávu vygeneruje.

Zjistili jsme, že zpráva se skládá z 256 malých elips. Elipsy jsou kreslené do mřížky, která má 18 sloupců a 12 řádků. Každé políčko mřížky má rozměry `20x20`. Elipsu nakreslíš tak, že:

- vygeneruješ náhodná čísla pro pořadové číslo řádku a pořadové číslo sloupce, vynásobíš je 20 a to budou souřadnice středu elipsy,
- vygeneruješ náhodná čísla od 1 do 20 pro šířku a výšku malé elipsy,
- když znáš střed a velikost elipsy, tak ji nakreslíš.

Toto zopakuješ 256krát.

Řešení:

```
import tkinter
import random

canvas = tkinter.Canvas()
canvas.pack()

for i in range(256):
    x = random.randint(1, 18) * 20
    y = random.randint(1, 12) * 20
    a = random.randint(1, 20)
    b = random.randint(1, 20)
    canvas.create_oval(x - a / 2, y - b / 2, x + a / 2,
        y + b / 2)
```