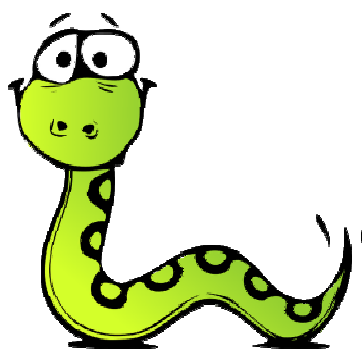


Programování v jazyce Python pro střední školy

Metodický list pro učitele
Sada testových úloh



Andrej Blaho
Lubomír Salanci
Václav Šimandl

V tomto metodickém listu představujeme několik vzorových úloh, které je možné použít při ověřování znalostí a dovedností žáků z programování v Pythonu. Tyto úlohy je možné zařadit buď na závěr celého kurzu, nebo v průběhu kurzu po příslušné lekci. Bylo by vhodné, aby žáci během testu měli k dispozici své poznámky z předchozích lekcí a mohli si tak vyhledat některé detaily zápisu v Pythonu. Je na učiteli, kolik úloh zadá v rámci jednoho testu a jaký čas žákům na řešení úloh poskytne.

Níže uvedené úlohy je potřeba vnímat jako vzorové a věříme, že na jejich základě bude učitel schopen vytvářet své vlastní testové úlohy. Využití úloh v nezměněné podobě s sebou nese mimo jiné riziko, že žáci tento metodický list naleznou na Internetu a jednoduše odevzdají uvedená vzorová řešení.

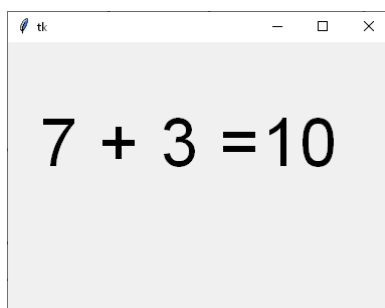
V následující tabulce uvádíme, po jaké lekci je možno danou úlohu **nejdříve** zařadit:

Číslo a název vzorové úlohy	Lekce
1. graficky_soucet.py	11
2. krokovací tabulka	17
3. dve_kostky.py	18
4. ctvrtkruh.py	18
5. rada_ctvercu.py	19
6. tri_kruhy.py	19
7. uhlopricne.py	20

1. Vytvoř nový program `graficky_soucet.py`, který si náhodně zvolí 2 celá čísla, obě z intervalu $\langle 1, 10 \rangle$. Tato čísla vypíše do grafické plochy tak, že jim nastaví parametr `font='arial 50'`. Mezi tato dvě čísla vypíše znak '+', za ně vypíše znak '=' a za tento znak ještě vypíše jejich součet.

To znamená, že použiješ 5 příkazů `create_text`. Všechny vypisované texty budou mít y-ovou souřadnici 100 a jejich x-ová souřadnice se bude zvětšovat o 60.

Výsledek by mohl vypadat například jako na následujícím obrázku:



2. Odkrokuj následující program:

```
soucet = 0
cislo = 5
for i in range(10):
    soucet = soucet + cislo
    cislo = cislo - 1
    if cislo < 0:
        cislo = 5
    # konec těla cyklu
print('součet je', soucet)
```

Krokovací tabulka má tři sloupce. V prvním řádku (pod záhlavím tabulky) jsou hodnoty proměnných před začátkem cyklu. Do dalších řádků zapisuj hodnoty těchto proměnných na konci vykonávání těla cyklu.

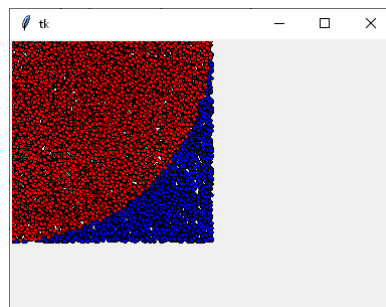
i	cislo	soucet
?	5	0
0		
1		
.		
.		
.		
.		
.		
.		
.		
.		

3. Hrajeme následující hru: házíme najednou dvěma hracími kostkami. Když na obou padne stejné číslo, přičteme si 6 bodů. Když padnou různá čísla, odečteme si 1 bod. Napiš program `dve_kostky.py`, který toto zopakuje 100krát a vypíše celkový počet bodů.

4. Vytvoř program `ctvrtkruh.py`, který bude generovat náhodně rozmístěné tečky v ploše 200×200 (náhodné souřadnice nechť jsou od 1 do 200). Tečky vytvoř jako malé barevné kroužky s poloměrem 2 a vygeneruj jich 10000. Červené tečky budou ty, pro které platí $x^2 + y^2 \leq 200^2$, ostatní budou modré.

Program vypočítá podíl počtu červených teček vzhledem k počtu všech teček. Tento podíl vynásobí 4 a výsledek na závěr vypíše do interaktivního okna. Matematici by uměli dokázat, že tento podíl se blíží číslu π – zkontroluj, zda tomu tak skutečně je.

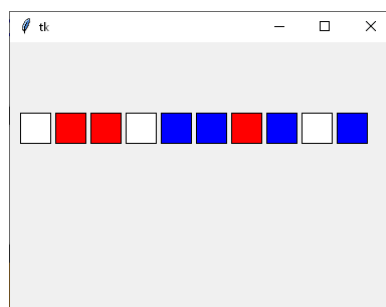
Vygenerovaný obrázek by měl vypadat přibližně takto:



5. Vytvoř nový program `rada_ctvercu.py` a v něm napiš podprogram `ctverec` s parametrem x , který do grafické plochy nakreslí čtverec se stranou délky 30. **Levý dolní** roh čtverce má souřadnice $[x, y]$, kde x je parametrem podprogramu a y je vždy 100. Čtverec bude vyplněný náhodnou barvou z množiny (bílá, modrá, červená).

Následně napiš for cyklus, pomocí kterého se nakreslí 10 čtverců, jež budou mít mezi sebou rozestupy o velikosti 5.

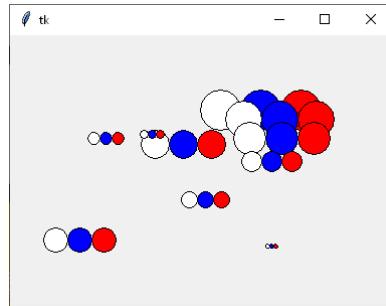
Výsledný obrázek by mohl vypadat například takto:



6. Vytvoř nový program `tri_kruhy.py` a v něm napiš podprogram `tri_kruhy` s parametrem `r`, který nakreslí těsně vedle sebe tři kruhy s poloměrem `r`. Podprogram nejprve náhodně zvolí souřadnice `[x, y]` – to bude střed prostředního kruhu (ten bude modrý), nalevo od něj nakreslí bílý kruh a napravo od něj červený.

Potom napiš for cyklus, v kterém se podprogram `tri_kruhy` zavolá 10krát, pokaždé s jiným poloměrem kruhů: nejprve bude poloměr 20, potom 18, potom 16, potom 14, atd. a poslední tři kruhy budou mít tedy poloměr 2.

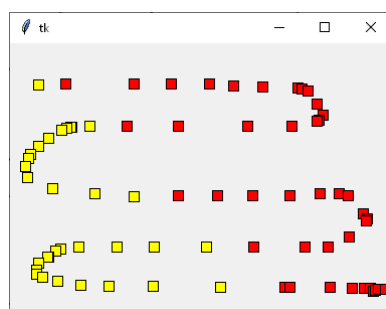
Výsledný obrázek by mohl vypadat například takto:



7. Vytvoř nový program `uhlopricne.py` a v něm napiš podprogram `ctverecek` s parametrem `mys`, který na pozici myši (souřadnice `[mys.x, mys.y]`) nakreslí barevný čtvereček se stranou délky 10. Pokud bude `x`-ová souřadnice větší než `y`-ová, čtvereček bude vybarvený červenou barvou. Jinak bude čtvereček vybarvený žlutou barvou.

Pomocí příkazu `canvas.bind` zajisti volání podprogramu `ctverecek` v případě, že stiskneme levé tlačítko myši a myši potom táhneme (využij '`<B1-Motion>`'). Následně program vyzkoušej.

Pomocí programu bys měl být schopen nakreslit například takovýto obrázek:



Řešení

1. Program graficky_soucet.py:

```
import tkinter
import random

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

cislo1 = random.randint(1, 10)
cislo2 = random.randint(1, 10)
canvas.create_text(50, 100, text=cislo1, font='arial 50')
canvas.create_text(110, 100, text='+', font='arial 50')
canvas.create_text(170, 100, text=cislo2, font='arial 50')
canvas.create_text(230, 100, text='=', font='arial 50')
canvas.create_text(290, 100, text=cislo1 + cislo2,
                    font='arial 50')
```

2. Krokovací tabulka:

i	cislo	soucet
?	5	0
0	4	5
1	3	9
2	2	12
3	1	14
4	0	15
5	5	15
6	4	20
7	3	24
8	2	27
9	1	29

3. Program dve_kostky.py:

```
import random

body = 0
for i in range(100):
    kostka1 = random.randint(1, 6)
    kostka2 = random.randint(1, 6)
    if kostka1 == kostka2:
        body = body + 6
    else:
        body = body - 1
print('Získal jsi', body, 'bodů')
```

4. Program ctvrtekruh.py:

```
import tkinter
import random

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

pocet = 0
for i in range(10000):
    x = random.randint(1, 200)
    y = random.randint(1, 200)
    if x*x + y*y <= 200*200:
        barva = 'red'
        pocet = pocet + 1
    else:
        barva = 'blue'
    canvas.create_oval(x-2, y-2, x+2, y+2, fill=barva)

print(pocet / 10000 * 4)
```

5. Program rada_ctvercu.py:

```
import tkinter
import random

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

def ctverec(x):
    y = 100
    barva = random.choice(['white', 'blue', 'red'])
    canvas.create_rectangle(x, y-30, x+30, y, fill=barva)

for i in range(10):
    ctverec(i*35 + 10)
```

6. Program tri_kruhy.py:

```
import tkinter
import random

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

def tri_kruhy(r):
    x = random.randint(50, 330)
    y = random.randint(50, 210)
    canvas.create_oval(x-3*r, y-r, x-r, y+r, fill='white')
    canvas.create_oval(x-r, y-r, x+r, y+r, fill='blue')
    canvas.create_oval(x+r, y-r, x+3*r, y+r, fill='red')

for i in range(10):
    tri_kruhy(20 - 2 * i)
```

7. program uhlopricke.py:

```
import tkinter

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

def ctverecek(mys):
    x = mys.x
    y = mys.y
    if x > y:
        barva = 'red'
    else:
        barva = 'yellow'
    canvas.create_rectangle(x, y, x+10, y+10, fill=barva)

canvas.bind('<B1-Motion>', ctverecek)
```