

Písomný výstup pedagogického klubu

1. Prioritná os	Vzdelávanie
2. Špecifický cieľ	1.1.1 Zvýšiť inkluzívnosť a rovnaký prístup ku kvalitnému vzdelávaniu a zlepšiť výsledky a kompetencie detí a žiakov
3. Prijímateľ	Gymnázium Pavla Horova, Masarykova 1, Michalovce 07179
4. Názov projektu	GPH - Reserata pro Futuro
5. Kód projektu ITMS2014+	312011U411
6. Názov pedagogického klubu	Informatika- matematická gramotnosť
7. Meno koordinátora pedagogického klubu	RNDr. Mária Spišáková
8. Školský polrok	september 2021 – január 2022
9. Odkaz na webové sídlo zverejnenia písomného výstupu	https://www.gphmi.sk/gph-reserata-pro-futuro/

Úvod: Metodika pre vyučovanie programovania Pythonu – práca so súbormi, tlačidlá, vstup Entry a posuvníky na grafickej ploche

Stručná anotácia: Metodika je rozdelená na 3 časti – práca so súbormi, na prácu s používaním tlačidiel a posuvníkov a pre používanie vstupov v projektoch s grafickou plochou. Metodika spracúva programovanie grafického editora v Pythone, umiestňovanie posuvníkov, labelov a tlačidiel a pod.

Kľúčové slová

Python, Button, Label, Entry, file, subor atď.

Zámer a priblíženie témy písomného výstupu

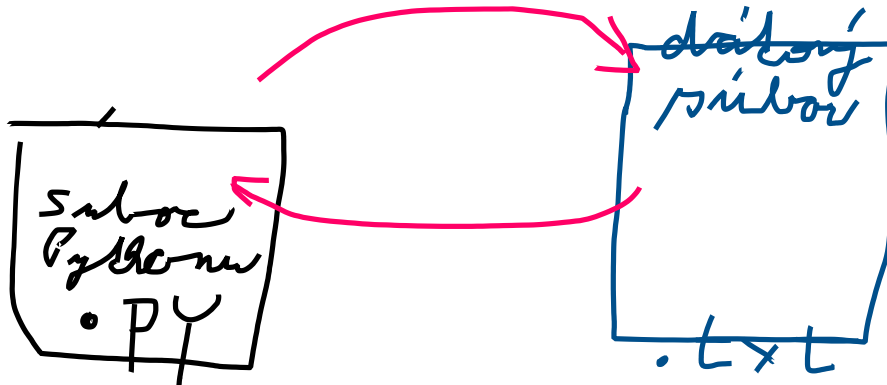
Téma prináša do vyučovania jazyk aPython náročnejšie problémy. Žiaci sa naučia pracovať príkazy, ktorými sa obsluhujú tlačidlá, klikanie myšou po grafickej ploche, ťahanie myšou, mazanie grafickej plochy a pod. Žiaci objavia krásu používania pythonu a jeho využitie v praktických projektoch.

Jadro – popis témy – Metodika vyučovania

PRÁCA SO SÚBORMI

Súbory sú externé „položky“ na disku. Python vie čítať a zapisovať dáta do súborov.

Obrázok nižšie zobrazuje štruktúru súborov.



Ak chceme s externým súborom pracovať, najprv musíme vytvoriť premennú typu súbor a otvoriť ju buď na zápis (write) alebo čítanie (read).

Otvorenie súboru na zápis:

```
subor = open(meno_suboru, 'w')
```

Otvorenie súboru na čítanie:

```
subor = open(meno_suboru, 'r')
```

Súbor, ktorý pbsahuje dáta s diakritikou otvárame takto:

```
subor = open(meno_suboru, 'r', encoding='utf-8')
```

Zápis do súboru

Zápis nejakého reťazca do súboru urobíme pomocou volania:

```
subor.write(reťazec)
```

Zápis do súboru pomocou print()

```
print( zoznam cisel alebo premennych, file=subor)
```

Uzavretie súboru

Je veľmi dôležité po ukončení práce so súborom ho uzavrieť. Znamená to, ako keby ste dátový súbor uložili – (file - save). Tento príkaz je väčšinou na konci programu.

```
subor.close()
```

Príklad:

Napište program, ktorý do externého súboru „slovo.txt“ zapíše slovo AHOJ.

Riešenie:

```
subor = open("slovo.txt", "w", encoding= "utf-8")  
  
subor.write("AHOJ")  
  
subor.close()
```

Po uložení a spustení programu v priečinku, kde je uložený tento váš zdrojový súbor .py vznikol nový textový súbor *slovo.txt*.

Príklad:

Napište program pre učiteľa, ktorý do externého súboru „ziaci.txt“ zapíše 5 mien žiakov, ktoré sa načítajú na vstupe. Zápis v textovom súbore nech je v tvare:

1. žiak: meno a priezvisko

2. žiak: meno a priezvisko

3. Žiak: meno a priezvisko

Riešenie:

```
subor = open("ziaci.txt", "w", encoding= "utf-8")

for i in range(5):
    meno = input("Zadaj pacienta: ")
    subor.write(f"{i+1}. pacient = {meno} \n")

subor.close()
```

Úlohy:

1. Načítajte cez vstup číslo n a do súboru zapíšte n riadkov hviezdíčiek:

```
*
**
***
****
*****
*****
*****
```

2. Program prečíta nejaké slovo a do súboru vypíše trojuholník, ktorý sa bude skladať z písmen tohto slova: v 1. riadku je prvé písmeno, v 2. sú prvé dve, v 3. sú prvé tri, ... v poslednom riadku je kompletne slovo. Môžeš dostať takýto výstup:

```
zadaj slovo: Python
P
Py
Pyt
Pyth
Pytho
```

Python

Čítanie zo súboru a vypisovanie obsahu súboru:

```
t = open('subor.txt', 'r')
for riadok in t:
    print(repr(riadok))
t.close()
```

Alebo:

```
t = open('subor.txt', 'r')
riadok = t.readline()
while riadok != '': # da sa to nahradiť: while riadok:
    print(riadok)
    riadok = t.readline()
t.close()
```

Funkcia

funkcia **repr()** - reprezentácia reťazca

```
repr(reťazec)
```

vráti takú reťazcovú reprezentáciu daného parametra, aby sme po jeho vypísaní (napr. funkciou `print()`) dostali presný taký tvar, aký očakáva Python pri zadávaní konštanty, teda aj s apostrofmi, prípadne aj so znakom `'\'` pri špeciálnych znakoch.

Metóda **strip()**:

odrezáva medzery a ENTER na konci reťazcov:

volanie:

```
r = r.strip()
```

Použitie for-cyklu pre čítanie zo súboru

```
t = open('subor.txt', 'r')
for riadok in t:
    print(repr(riadok))
t.close()
```

Metóda read()

`read()`, prečíta naraz obsah súboru:

```
t = open('subor.txt', 'r')
cely_subor = t.read()
t.close()
print(cely_subor, end='')
```

Otvorenie súboru na dopisovanie na koniec

Ak otvárame súbor a použijeme `'a'`, ako atribút, ktoré označuje anglické **append**, tak sa nezmaže obsah sboru, ale sa na oniec súboru dopisujú ďalšie riadky:

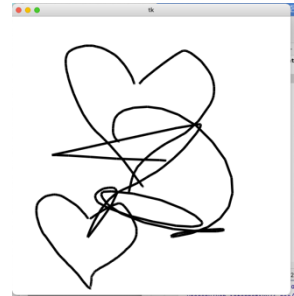
```
t = open('subor3.txt', 'a')
t.write('pridany riadok na koniec\nna este jeden\n')
t.close()
```

Zdroj: <http://python.input.sk/07.html>, január 2022

SKICÁR V PYTHONĚ

Jednoduchý skicár v Pythone

Napíšte program, ktorý bude reagovať ako jednoduchý skicár – bude ťahať čiaru pod kurzorom.



```
import tkinter
c = tkinter.Canvas(width = 600, height = 600)
c.pack()
prevX = 300
prevY = 300

def naber(event):
    global prevX, prevY
    prevX= event.x
    prevY= event.y

def kresli(event):
    global prevX, prevY
    x, y = event.x, event.y

    c.create_line(prevX, prevY, x, y, fill="black", width=5)
    prevX, prevY = x, y

def zmaz(event):
    c.delete("all")

c.bind("<B1-Motion>", kresli)
c.bind("<ButtonPress>", naber)

c.mainloop()
```

Zisťovanie atribútov objektov na plátne:

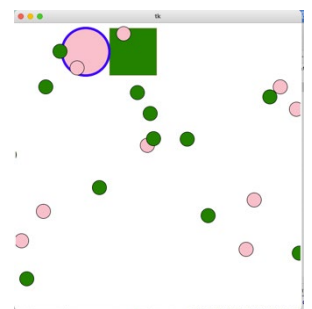
Procedúry na zisťovanie nakreslených objektov v Pythone:

Príkaz	Príklad	vysvetlenie
canvas.delete(cislo objektu)		Vymaže objekt s ID
canvas.itemcget(id_útvary, 'vlastnosť')	Farba = c.itemcget(1, 'fill')	Zistí hodnotu vlastnosti objektu s ID

	canvas.itemcget(1, 'outline')	
canvas.itemconfig(id_útvary, vlastnosť = hodnota)	canvas.itemconfig(1, fill = 'red')	Nastavenie vlastnosti objektu s ID
canvas.coords(id_útvary)	canvas.coords(7)	zistíme súradnice útvaru. Výsledkom je zoznam, jeho prvkami sú všetky súradnice.
canvas.find_all()		vráti n-ticu všetkých identifikátorov existujúcich útvarov v poradí, ako sú na sebe poukladané
canvas.lift(ID)	canvas.lift(1)	Presunie útvar s ID v poradí útvarov na vrch
canvas.lower(ID)	canvas.lower(2)	Presunie útvar s identifikátorom ID na spodok
canvas.find_overlapping(x1, y1, x2, y2).	x1, y1, x2, y2 = event.x-5, event.y-5, event.x+5, event.y+5 podmysou = canvas.find_overlapping(x1, y1, x2, y2)	funkcia vráti n-ticu všetkých identifikátorov útvarov, ktoré sa nachádzajú v zadanom obdĺžniku

Pohyb krúžkov a zisťovanie ich farieb

Program vie zistiť farbu výplne a okraja krúžku s obdĺžnika na grafickej ploche, potom vygeneruje krúžky na náhodných pozíciách farieb, ktoré sú vo veľkých kruhu a obdĺžniku a potom ich bude pohybovať trhane – ako keby to boli molekuly.



```
import tkinter, random
c = tkinter.Canvas(width= 600, height = 600)
c.pack()
obd = c.create_rectangle(200, 10, 300, 110, fill = "green", outline="pink")
kruh = c.create_oval(100, 10, 200, 110, fill = "pink", outline = "blue",
width = 5)
kruh
obd
```



```

farba = c.itemcget(kruh, 'fill')
okraj = c.itemcget(1, 'outline')
hrubka = c.itemcget(1, 'width')
print(f" f = {farba}, okraj = {okraj}, hrubka = {hrubka}")

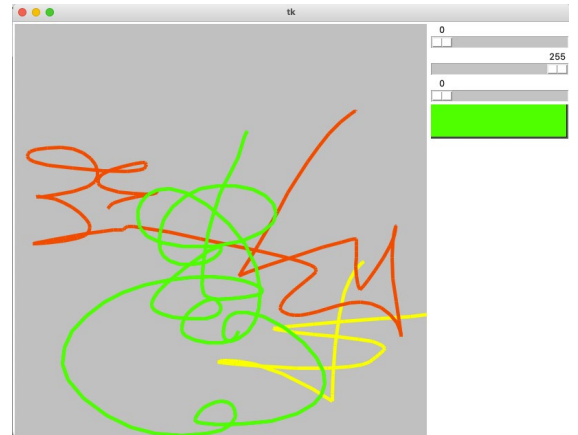
def kruzok(farba):
    x, y = random.randint(0,600), random.randint(0,600)
    c.create_oval(x,y, x+30, y+30, fill = farba)

for i in range(20):
    kruzok(c.itemcget(kruh, 'fill'))
    kruzok(c.itemcget(oba, 'fill'))
while 4:
    for j in range(3,39):
        c.move(j, random.randint(-5, 5), random.randint(-5, 5))
        #c.move(25, 0, random.randint(-5, 5))
    c.after(0)
    c.update()
c.mainloop()

```

Skicár – s nastavovaním farieb

V projekte skicár sú 3 posuvníky, ktorými sa menia farby r,g,b. Navolenou farbou sa v skicári kreslí. Navolenú farbu má aj obdĺžnik v druhej grafickej ploche.



```

import tkinter
c = tkinter.Canvas(width = 600, height = 600, bg = "silver")
c.pack(side = "left")
prevX = 300
prevY = 300
r = 100
g = 100
b = 100

def farba(r,g,b):
    return f"#{r:02x}{g:02x}{b:02x}"

def cervena(event):
    global r
    r = posuvac1.get()
    farba(r,g,b)
    c.itemconfig(1, fill=farba(r, g, b))

def zelena(event):
    global g
    g = posuvac2.get()
    farba(r,g,b)
    c.itemconfig(1, fill=farba(r, g, b))

def modra(event):
    global b

```

```

b = posuvac3.get()
farba(r,g,b)
c1.itemconfig(1, fill=farba(r, g, b))

def naber(event):
    global prevX, prevY
    prevX= event.x
    prevY= event.y

def kresli(event):
    global prevX, prevY
    x, y = event.x, event.y

    c.create_line(prevX, prevY, x, y, width=5, fill= farba(r,g,b))
    prevX, prevY = x, y

def zmaz(event):
    c.delete("all")

posuvac1 = tkinter.Scale(from_ = 0, to = 255, orient = 'horizontal',length
= 200, command = cervena)
posuvac1.pack()
posuvac1.set(r)

posuvac2 = tkinter.Scale(from_ = 0, to = 255, orient = 'horizontal',length
= 200, command = zelena)
posuvac2.pack()
posuvac2.set(g)

posuvac3 = tkinter.Scale(from_ = 0, to = 255, orient = 'horizontal',length
= 200, command = modra)
posuvac3.pack()
posuvac3.set(b)

c1 = tkinter.Canvas(width = 200, height = 50, bg = farba(r,g,b))
c1.pack()
c1.create_rectangle(0,0,200,50, fill = farba(r,g,b))
c1.itemconfig(1,fill=farba(r,g,b))

c.bind("<B1-Motion>", kresli)
c.bind("<ButtonPress>", naber)
c.bind("<Button-2>", zmaz)

c.mainloop()

```

Podľa:

K. Kučera: programujeme v Pythone 2. diel

Záver:**Zhrnutia a odporúčania pre činnosť pedagogických zamestnancov**

Metodický materiál je vhodný pre vyučovanie v 2. alebo v 3. ročníku na hodinách PRO. Používajú sa pri ňom pomôcky, ktorými škola disponuje.

11. Vypracoval (meno, priezvisko)	RNDr. Mária Spišáková, PhD.
12. Dátum	28. 1. 2022
13. Podpis	
14. Schválil (meno, priezvisko)	RNDr. Renáta Gaľová
15. Dátum	28. 1. 2022
16. Podpis	