**Pracovné listy: Programovanie grafických objektov v jazyku Python**

Pracovný list 1 Grafika v jazyku Python

Pracovný list 2 Kreslenie čiar

Pracovný list 3 Kreslenie obdĺžnika

Pracovný list 4 Kreslenie elipsy

**Pracovný list 1 Ggrafika v jazyku Python**

|  |
| --- |
| **Obsahový štandard** |
| * vysvetliť pojmy príkaz, parameter príkazu, postupnosť príkazov * formulovať programovací jazyk, program * určiť ako súvisia príkazy a výsledok realizácie programu * zápis algoritmu a vykonanie programov |
| **Výkonový štandard** |
| * riešiť problém skladaním príkazov do súvislostí * aplikovať pravidlá na zostavenie postupnosti príkazov |

Na stránke www.python.org si stiahneme verziu Python vhodný pre náš počítač. Potom si ho nainštalujeme podľa inštrukcií.

Po spustení programu sa otvorí okno s názvom Python Shell. V tomto okne môžeme písať príkazy, ktoré sa po potvrdení klávesom Enter automaticky vykonajú.

Na to, aby sme mohli programovať grafické úlohy v jazyku Python, potrebujeme vytvoriť grafické okno, kde sa budú príkazy vykresľovať. Toto okno si vytvoríme nasledovnými príkazmi:

**import tkinter**

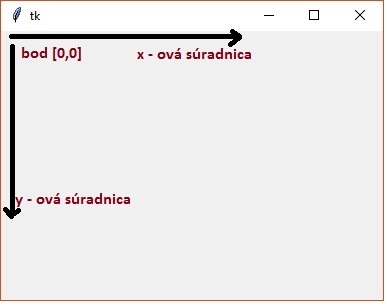
**canvas = tkinter.Canvas()**

**canvas.pack()**

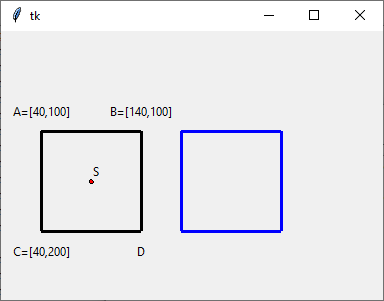
Príkaz **import tkinter** oznámi, že budeme pracovať s modulom tkinter, ktorý obsahuje všetky grafické príkazy. Týmto príkazom vznikla nová premenná tkinter, ktorá obsahuje referenciu na tento modul. Zápis **tkinter.Canvas()** vytvorí grafickú plochu a aby sme s touto plochou mohli ďalej pracovať, uložíme si jej referenciu do premennej canvas. Volanie **canvas.pack()** zabezpečí zobrazenie nového okna aj s novovytvorenou grafickou plochou.

V textovom okne (shell) zadávame grafické príkazy, ktoré sa budú priamo vykresľovať v grafickom okne. Program spustíme klávesom F5.

Súradnicová sústava v prostredí Python má bod [0,0] vľavo hore, x-ové súradnica rastie smerom doprava, y-ová smerom nadol.



Vrcholy A, B, C čierneho štvorca vykresleného v grafickej ploche na obrázku majú nasledovné súradnice:



**Úlohy na precvičenie:**

1. Otvorte stránku [www.python.org](http://www.python.org) a stiahnite Python. (verzia 3.5.2)
2. Na svojom počítači nájdite na ploche alebo v Prieskumníku aplikáciu na spustenie Pythonu.
3. Vyskúšajte napísať program na vytvorenie grafickej plochy canvas. Uložte ho a spustite. Čo vidíme na grafickej ploche?

import tkinter

canvas = tkinter.Canvas()

canvas.pack()

1. Aké súradnice má vrchol D a bod S na obrázku?
2. Doplňte súradnice pre vrcholy modrého štvorca na obrázku.

**Pracovný list 2 Kreslenie čiar**

|  |
| --- |
| **Obsahový štandard** |
| * orientácia v súradnicovej sústave * vzťahy medzi bodmi ležiacimi na x-ovej a y-ovej súradnici * kreslenie čiar * zmena farby čiary * zmena hrúbky čiary |
| **Výkonový štandard** |
| * vedieť sa orientovať v súradnicovej sústave * vysvetliť vzťahy medzi bodmi na x-ovej a y-ovej súradnici * vedieť napísať príkaz na nakreslenie čiary * aplikovať príkazy na zmenu farby a hrúbky čiary |

Čiary v grafickom režime kreslíme pomocou funkcie **canvas.create\_line()**. Táto funkcia dokáže kresliť lomenú čiaru, ktorá sa skladá z jednej alebo viacerých úsečiek. Parametrom funkcie je postupnosť súradníc v tvare **canvas.create\_line(x1, y1, x2, y2, ...)**. Prvý program môže vyzerať takto:

**Príklad 2.1**

import tkinter

canvas = tkinter.Canvas()

canvas.pack()

canvas.create\_line(20, 120, 130, 50)

Najlepšou ukážkou orientácie na plátne a vykresľovanie lomených čiar je zostrojenie trojuholníka. Zadáme si ľubovoľné 3 body, ktoré spojíme. Úsečky môžeme nakresliť každú zvlášť v príkazovom riadku, alebo jedným príkazom.

**Príklad 2.2**

**import** **tkinter**

canvas = tkinter.Canvas()

canvas.pack()

canvas.create\_line(120, 10, 70, 90)

canvas.create\_line(70, 90, 170, 90)

canvas.create\_line(170, 90, 120, 10)

alebo

import tkinter

canvas = tkinter.Canvas()

canvas.pack()

canvas.create\_line(120, 10, 70, 90, 170, 90, 120, 10)

Pri kreslení čiar môžeme ešte špecifikovať aj ďalšie doplnkové parametre tejto kresby: za postupnosť bodov môžeme uviesť hrúbku nakreslenej čiary (parameter **width=**) a jej farbu (parameter **fill=**). Keď tieto dva parametre nešpecifikujeme, kreslí sa čierna čiara hrúbky 1.

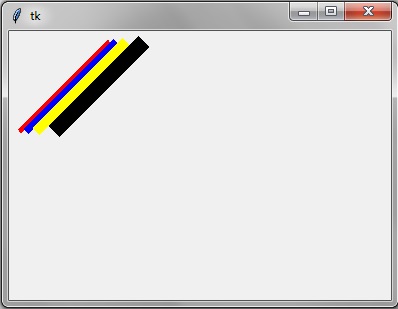
**canvas.create\_line(20, 110, 130, 100, width=10, fill='blue')**

nakreslí úsečku s hrúbkou čiary 10 a modrou farbou

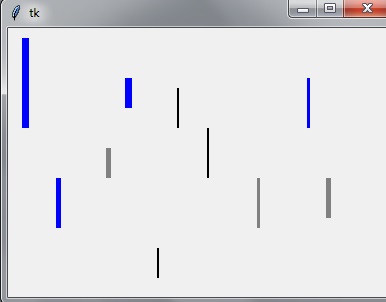
Farby zadávame ako znakové reťazce názvov farieb v angličtine a musia byť v apostrofoch. Dajú sa použiť rôzne akceptované mená farieb, no najčastejšie sa používajú základné farby: 'white', 'black', 'red', 'blue', 'green', 'yellow'.

**Úlohy na precvičenie:**

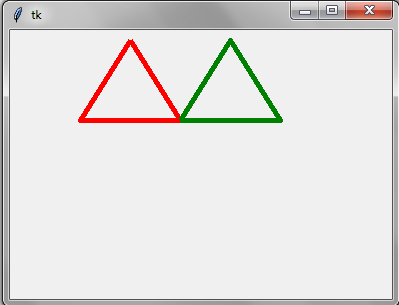
1. Nakresli obrázok podľa obrázku použitím rôznych farieb a hrúbok úsečiek. Začni červenou farbou, pokračuj modrou, žltou a nakoniec čiernou. Hrúbky čiar meň vzostupne.



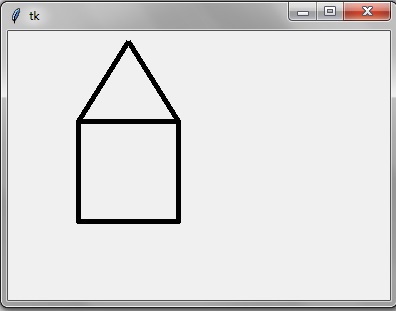
1. Vonku začalo pršať a pri pohľade z okna bolo vidieť ako padá dážď. Na obrázku je ukážka. Zostav program, ktorý bude vykresľovať rôzne dlhé úsečky modrej, čiernej a sivej farby a taktiež rôznej hrúbky na rôznych pozíciách.



1. Nakresli trojuholník červenej farby, ktorý sa bude dotýkať vrcholom trojuholníka zelenej farby, podobne ako na obrázku.



1. Pomocou úsečiek nakresli domček.



**Metodické odporúčania**

Najväčším problémom pre žiakov je zorientovanie sa v súradnicovej sústave. Odporúčame do zošita najskôr obrázok prekresliť, vypočítať súradnice bodov a až potom úlohu naprogramovať. Najčastejšou chybou býva zlé zadanie súradníc či zadanie nepárneho počtu súradníc.

**Riešenie úloh:**

**Úloha 1**

import tkinter

canvas=tkinter.Canvas()

canvas.pack()

canvas.create\_line(10,100,100,10, width=5, fill='red')

canvas.create\_line(17,100,107,10, width=7, fill='blue')

canvas.create\_line(27,100,117,10, width=10, fill='yellow')

canvas.create\_line(45,100,135,10, width=15, fill='black')

**Úloha 2**

import tkinter

canvas=tkinter.Canvas()

canvas.pack()

canvas.create\_line(17,100,17,10, width=7, fill='blue')

canvas.create\_line(100,120,100,150, width=5, fill='grey')

canvas.create\_line(200,100,200,150, width=2, fill='black')

canvas.create\_line(170,60,170,100, width=2, fill='black')

canvas.create\_line(50,150,50,200, width=5, fill='blue')

canvas.create\_line(300,50,300,100, width=3, fill='blue')

canvas.create\_line(250,150,250,200, width=3, fill='grey')

canvas.create\_line(120,50,120,80, width=7, fill='blue')

canvas.create\_line(320,150,320,190, width=5, fill='grey')

canvas.create\_line(150,220,150,250, width=2, fill='black')

**Úloha 3**

import tkinter

canvas=tkinter.Canvas()

canvas.pack()

canvas.create\_line(120, 10, 70, 90, 170, 90, 120, 10, width=5, fill='red')

canvas.create\_line(170, 90, 220, 10, 270, 90, 170, 90, width=5, fill='green')

**Úloha 4**

import tkinter

canvas=tkinter.Canvas()

canvas.pack()

canvas.create\_line(40, 100, 140, 100, width=3)

canvas.create\_line(140, 100, 140, 200, width=3)

canvas.create\_line(40, 200, 140, 200, width=3)

canvas.create\_line(40, 200, 40, 100, width=3)

canvas.create\_line(40, 100, 90, 50, width=3)

canvas.create\_line(90, 50, 140, 100, width=3)

**Pracovný list 3 Kreslenie obdĺžnika**

|  |
| --- |
| **Obsahový štandard** |
| * orientácia v súradnicovej sústave * kreslenie obdĺžnika * zmena farby obvodovej čiary * zmena hrúbky obvodovej čiary * zmena farby výplne obdĺžnika |
| **Výkonový štandard** |
| * vedieť sa orientovať v súradnicovej sústave * vedieť napísať príkaz na nakreslenie obdĺžnika * aplikovať príkazy na zmenu farby a hrúbky čiary obdĺžnika * použiť príkaz outline * rozoznať príkazy fill a outline |

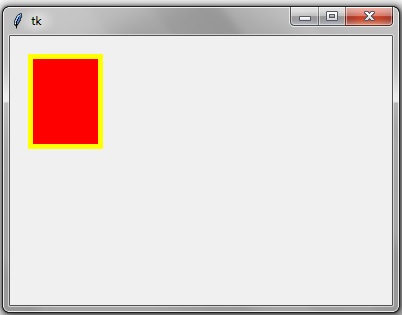
Obdĺžniky kreslíme pomocou funkcie **canvas.create\_rectangle()**. V tejto funkcii sú parametrami súradnice dvoch protiľahlých vrcholov kresleného obdĺžnika. Jej tvar je: **canvas.create\_rectangle(x1, y1, x2, y2),** strany tohto obdĺžnika sú vždy rovnobežné so súradnicovými osami.

Takto kreslené obdĺžniky (resp. štvorce) majú nevyplnené vnútro (vnútro obdĺžnika je priesvitné). Tiež obrys obdĺžnika je tenká čierna čiara hrúbky 1. Pri kreslení obdĺžnikov môžeme ešte špecifikovať aj ďalšie doplnkové parametre ako to bolo pri kreslení čiar a to hrúbku obrysu (parameter **width=**), výplň obdĺžnika (parameter **fill=**) a výplň obrysu (parameter **outline=**). Napr. obdĺžnik s červenou obrysovou čiarou hrúbky 5 a výplňou zelenou bude zapísaný takto **canvas.create\_rectangle(10, 100, 110, 160, outline='red', fill='green', width=5).**

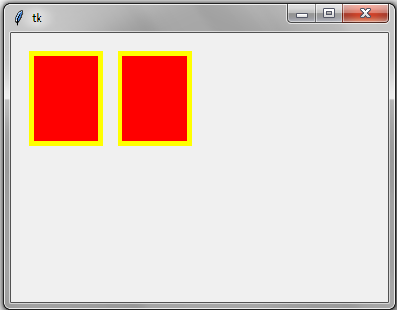
Môže sa stať, že chceme nakresliť niečo bez obrysovej čiary či bez výplne. Vtedy pri zadávaní parametrov použijeme len apostrofy ' ', bez zadania hodnoty.

**Úlohy na precvičenie:**

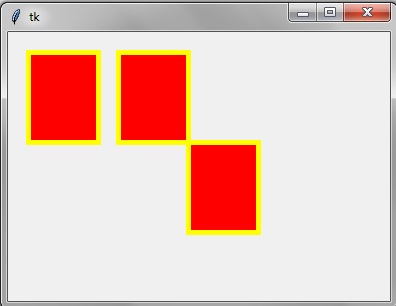
1. Nakresli červený obdĺžnik so žltou obrysovou čiarou a rozmermi 70 x 90.



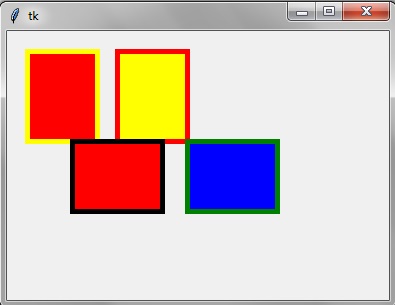
1. Nakresli rovnaký obdĺžnik, posunutý od prvého o 20 bodov.



1. Dokresli do obrázku obdĺžnik, ktorý sa bude dotýkať pravého dolného vrcholu, podľa obrázka.



1. Nakresli obdĺžniky podľa obrázka.



**Riešenie úloh:**

**Úloha 1**

import tkinter

canvas = tkinter.Canvas()

canvas.pack()

canvas.create\_rectangle(20,20,90,110, width=5, outline='yellow',fill='red')

**Úloha 2**

import tkinter

canvas=tkinter.Canvas()

canvas.pack()

canvas.create\_rectangle(20,20,90,110, width=5, outline='yellow',fill='red')

canvas.create\_rectangle(110,20,180,110, width=5, outline='yellow',fill='red')

**Úloha 3**

import tkinter

canvas=tkinter.Canvas()

canvas.pack()

canvas.create\_rectangle(20,20,90,110, width=5, outline='yellow',fill='red')

canvas.create\_rectangle(110,20,180,110, width=5, outline='yellow',fill='red')

canvas.create\_rectangle(180,110,250,200, width=5, outline='yellow',fill='red')

**Úloha 4**

import tkinter

canvas=tkinter.Canvas()

canvas.pack()

canvas.create\_rectangle(20,20,90,110, width=5, outline='yellow',fill='red')

canvas.create\_rectangle(110,20,180,110, width=5, outline='red',fill='yellow')

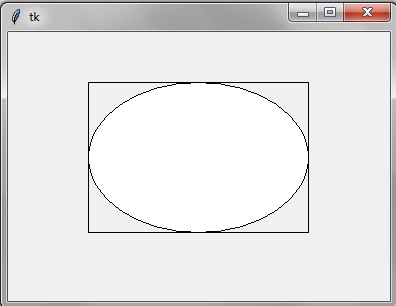
canvas.create\_rectangle(180,110,250,170, width=5, outline='green',fill='blue')

canvas.create\_rectangle(70,110,140,170, width=5, outline='black',fill='red')

**Pracovný list 4 Kreslenie elipsy**

|  |
| --- |
| **Obsahový štandard** |
| * orientácia v súradnicovej sústave * kreslenie elipsy a kruhu * zmena farby obvodovej čiary * zmena hrúbky obvodovej čiary * zmena farby výplne elipsy |
| **Výkonový štandard** |
| * vedieť sa orientovať v súradnicovej sústave * vedieť napísať príkaz na nakreslenie elipsy * vysvetliť rozdiel medzi elipsou a kruhom * aplikovať príkazy na zmenu farby a hrúbky čiary elipsy/kruhu * použiť príkaz outline * rozoznať príkazy fill a outline |

Prechod ku kresleniu elíps, keď vieme nakresliť obdĺžnik, je veľmi jednoduchý. Parametre príkazu vychádzajú z kreslenia obdĺžnika. Pre tkinter sú to presne rovnaké parametre, len elipsa ich trochu inak interpretuje: z dvoch protiľahlých vrcholov nenakreslí obdĺžnik ale **vpísanú** elipsu, t.j. elipsu, ktorá leží vo vnútri tohto „mysleného“ obdĺžnika a dotýka sa jeho strán. Parametrami príkazu sú súradnice dvoch protiľahlých vrcholov „mysleného“ opísaného obdĺžnika v tvare **canvas.create\_oval(x1, y1, x2, y2)**. Strany tohto mysleného obdĺžnika sú rovnobežné so súradnicovými osami.



import tkinter

canvas = tkinter.Canvas()

canvas.pack()

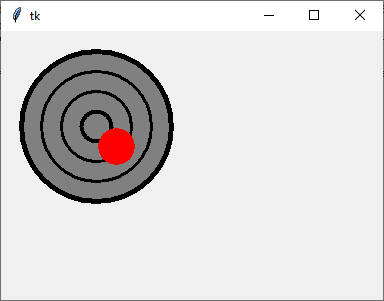
canvas.create\_rectangle(80, 50, 300, 200)

canvas.create\_oval(80, 50, 300, 200, width=3, outline='black', fill='white')

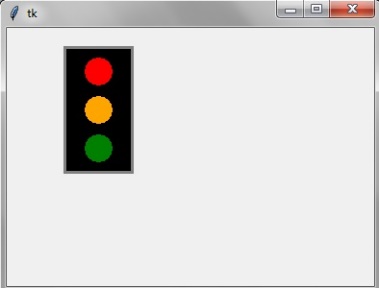
Ďalšie parametre width=, outline= a fill= majú presne rovnaký význam ako boli pre obdĺžnik: hrúbka obrysu elipsy, farba obrysu a farba výplne. Ak nešpecifikujeme žiaden parameter, tak sa kreslí elipsa s čiernym obrysom hrúbky 1, ktorá je bez výplne. Ak zadáme elipsu, ktorá je vpísaná do štvorca, nakreslí sa kružnica.

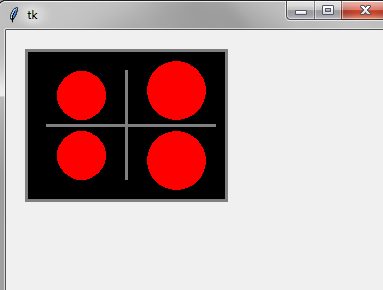
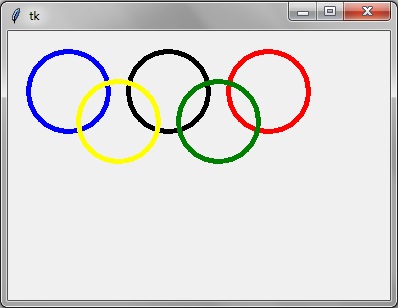
**Úlohy na precvičenie:**

1. Nakresli kruhový terč so 4 čiernymi kruhmi podľa obrázku. Na ľubovoľnom mieste terča bude červený kruh.



1. Nakresli semafor podľa obrázku: čierny obdĺžnik orámovaný sivou čiarou hrúbky 3 a svetelné kruhy s priemerom 30.



1. Nakresli varnú dosku, ktorá bude obsahovať 4 horáky. Najskôr nakresli čierny obdĺžnik orámovaný sivou čiarou hrúbky 3. Menšie horáky majú priemer 30, väčšie majú priemer 50. Nakoniec varnú dosku rozdeľ čiarami.
2. Nakresli olympijské kruhy. Kruhy sú v poradí modrá, čierna a červená v hornom rade, v dolnom žltá a zelená. Veľkosť kruhov môže byť napr. 80.

**Riešenie úloh:**

**Úloha 1**

import tkinter

canvas = tkinter.Canvas()

canvas.pack()

canvas.create\_oval(20,20,170,170, width=5, outline='black',fill='grey')

canvas.create\_oval(40,40,150,150, width=3, outline='black')

canvas.create\_oval(60,60,130,130, width=3, outline='black')

canvas.create\_oval(80,80,110,110, width=4, outline='black')

canvas.create\_oval(110,110,80,80, width=3, outline='black')

canvas.create\_oval(100,100,130,130, width=7, outline='red', fill='red')

**Úloha 2**

import tkinter

canvas=tkinter.Canvas()

canvas.pack()

canvas.create\_rectangle(60, 20, 130, 150, width=3, outline='grey', fill='black')

canvas.create\_oval(80,30,110,60, fill='red')

canvas.create\_oval(80,70,110,100, fill='orange')

canvas.create\_oval(80,110,110,140, fill='green')

**Úloha 3**

import tkinter

canvas=tkinter.Canvas()

canvas.pack()

canvas.create\_rectangle(20, 20, 220, 170, width=3, outline='grey', fill='black')

canvas.create\_oval(50,40,100,90, fill='red')

canvas.create\_oval(50,100,100,150, fill='red')

canvas.create\_oval(140,100,200,160, fill='red')

canvas.create\_oval(140,30,200,90, fill='red')

canvas.create\_line(120,40,120, 150, width=3, fill='grey')

canvas.create\_line(40,95,210, 95, width=3, fill='grey')

**Úloha 4**

import tkinter

canvas = tkinter.Canvas()

canvas.pack()

canvas.create\_oval(20,20,100,100, width=5, outline='blue')

canvas.create\_oval(120,20,200,100, width=5, outline='black')

canvas.create\_oval(220,20,300,100, width=5, outline='red')

canvas.create\_oval(70,50,150,130, width=5, outline='yellow')

canvas.create\_oval(170,50,250,130, width=5, outline='green')