



PLÁNY HODÍN



PROGRAMOVANIE A KÓDOVANIE V ODBORNOM VZDELÁVANÍ

2023-2-SK01-KA210-VET-000184496 ! 1. 2024 - 31. 12. 2025

CNC

Financované Európskou úniou. Vyjadrené názory a postoje sú názormi a vyhláseniami autora(-ov) a nemusia nevyhnutne odrážať názory a stanoviská Európskej únie alebo SAAIC. Európska únia ani organizácia udeľujúca grant za ne nepreberajú žiadnu zodpovednosť



Funded by
the European Union



PLÁN HODINY



PREDMET: PROGRAMOVANIE

TÉMA: PROGRAMOVANIE V PYTHONĚ – KRESLENIE GEOMETRICKÝCH TVAROV.

- CIELE: 1. ZÍSKAŤ ZÁKLADNÉ ZNALOSTI PROGRAMOVANIA V PYTHONĚ POMOCOU GRAFICKEJ KNIŽNICE TKINTER.**
- 2. NAUČIŤ SA PRACOVAŤ S GRAFICKÝM PLÁTNOM. NAKRESLITE NAŇ JEDNODUCHÉ TVARY.**
- 3. VYTVORIŤ OBRÁZKY NA NÁHODNOM MIESTE POMOCOU KNIŽNICE TKINTER.**

POMÔCKY:

POČÍTAČ S NAINŠTALOVANÝM PYTHONOM A KNIŽNICOU TKINTER.

VZOROVÁ UČEBNICA PYTHONU

PRACOVNÝ LIST S CVIČENIAMÍ

INTRO: UKÁŽEME SI AKO VYTVORÍME A SPUSTÍME PRVÝ PROGRAM, VYTVORÍME GRAFICKÉ PLÁTNO, NA KTORÉ BUDEME KRESLIŤ GRAFICKÉ TVARY. POUŽIJEME PREMENNÚ, VYGENERUJEME NÁHODNÉ HODNOTY A VYKRESLÍME OBRÁZKY NA NÁHODNOM MIESTE POMOCOU KNIŽNICE TKINTER.

ŠTRUKTÚRA HODINY: ZAPOJENIE (4 MINÚTY) – VYKRESLENIE ELEMENTÁRNEHO OBRÁZKU V KARTEZIÁNSKYCH SÚRADNICIACH

ROZHOVOR SO ŽIAKMI O SÚRADNICOVOM SYSTÉME V MODULE TKINTER (ÚLOHA 1)

- **VÝSKUM (12 MINÚT) – SKÚMANIE NOVÉHO PRÍKAZU (ÚLOHA 2 AŽ 4)**
- **VÝKLAD (15 MINÚT) – ZHRNUTIE A VYSVETLENIE POZNATKOV Z ČASTI SKÚMANIE PROBLEMATIKY TVORBY PROGRAMY S GRAFICKÝM UŽÍVATEĽSKÝM ROZHRANÍM**
- **VYPRACOVANIE (9 MINÚT) – SAMOSTATNÉ PROGRAMOVANIE NÁROČNEJŠÍCH ÚLOH (ÚLOHY 5 AŽ 6)**
- **HODNOTENIE (5 MINÚT) – RIEŠENIE SEBAHODNOTIACEHO TESTU, DISKUSIA O ODPOVEDIACH.**

STRUČNÝ PREHL'AD PROGRAMOVANIA V PYTHONĚ (PRE ZAČIATOČNÍKOV).

INŠTALÁCIA KNIŽNICE TKINTER (AK NIE JE NAINŠTALOVANÁ).

PREZENTÁCIA KNIŽNICE TKINTER.

VYSVETLENIE PRINCÍPU KRESLENIA NA PLÁTNO.

1. PRACOVNÝ LIST PRE ŽIAKA

POSTUPOVAŤ BUDEME PODĽA NAŠEJ ELEKTRONICKEJ UČEBNICE PROGRAMOVANIE V PYTHONĚ, KTORÁ JE URČENÁ PRE STREDNÉ ŠKOLY. VZOR UČEBNICE NÁJDETE NA:

[HTTP://WWW.PROGRAMUJEMEVPYTHONE.SK/UKAZKA.PDF](http://www.programujemevpythone.sk/ukazka.pdf)

UKÁŽKA OBSAHUJE JEDNODUCHÉ PRÍKLADY KRESLENIA:

1. KRESLENIE ZÁKLADNÝCH TVAROV:

KRESLENIE KRUHU, OBDĽŽNIKA, ŠTVORCA A TROJUHOĽNÍKA

POMOCOU RÔZNYCH FARIEB A HRÚBOK ČIAR

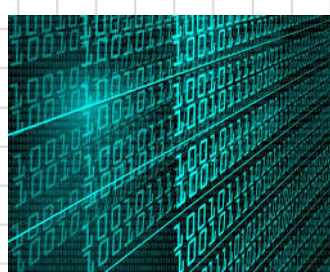
UKÁŽKA KÓDU PRE KAŽDÝ TVAR

FÁZA EXPOZÍCIE – EXPOZÍCIA NOVÉHO PREDMETU

– NA TABUĽU SI STRUČNE POZNAČÍM ZÁKLADNÉ POJMY



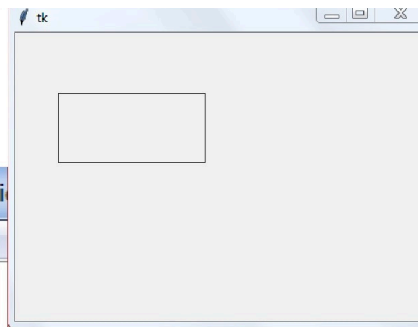
PLÁN HODINY



PRACOVNÝ LIST PRE ŽIAKA

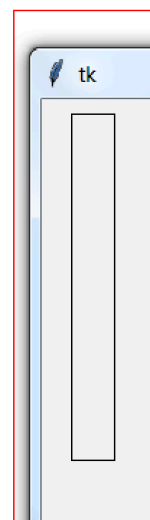
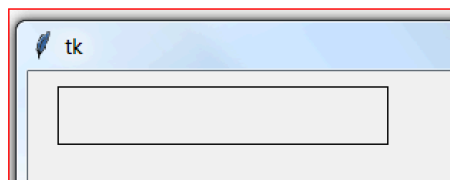
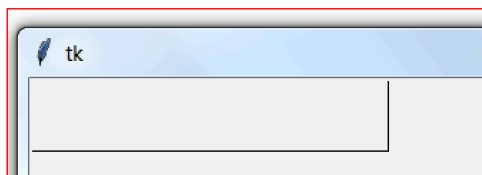
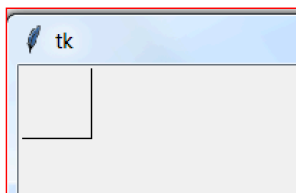
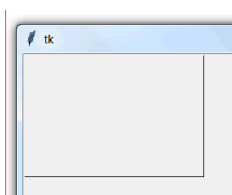
1. Úloha: Vytvor súbor 05lek_p04.py, v ňom uvedené príkazy a program xxxxxspušť:

```
05lek_p04.py - F:/14_Python(SŠ)/PythonSŠ_PracovnéListy/05_PL (ri  
File Edit Format Run Options Window Help  
import tkinter  
canvas = tkinter.Canvas()  
canvas.pack()  
  
canvas.create_rectangle(50, 70, 220, 150)
```



2. Úloha: V zátvorkách príkazu `canvas.create_rectangle(,,,)` sú 4 čísla. Skús ich v programe 05lek_p05.py postupne meniť (zmeny oproti predchádzajúcemu zápisu sú zvýraznené žltou). Program zakaždým spušť, aby si videl, čo nakreslí:

- a) `canvas.create_rectangle(0, 0, 220, 150)`
- b) `canvas.create_rectangle(0, 0, 50, 50)`
- c) `canvas.create_rectangle(0, 0, 250, 50)`
- d) `canvas.create_rectangle(20, 10, 250, 50)`
- e) `canvas.create_rectangle(20, 10, 50, 250)`





PLÁN HODINY

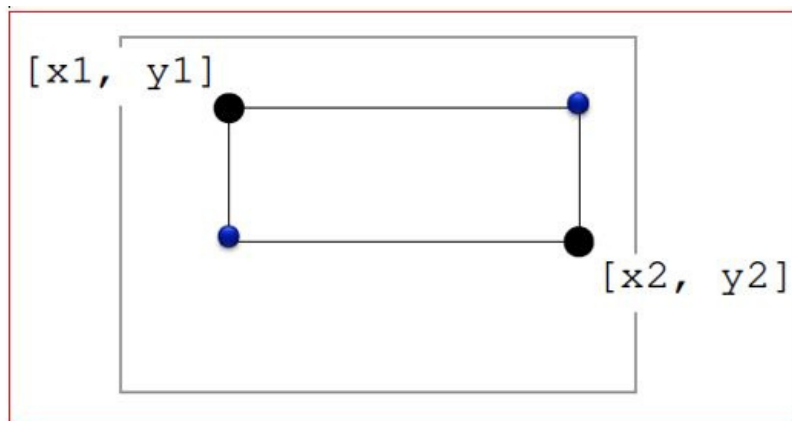


PRACOVNÝ LIST PRE ŽIAKA

3. Úloha: Na počítači leží bod so súradnicami $[0, 0]$ v ľavom hornom rohu. Os x ide zľava doprava. Os y je oproti matematike prevrátená - ide zhora nadol (čím väčšie číslo, tým nižšie).



4. Úloha: V príkaze `canvas.create_rectangle(x1, y1, x2, y2)` píšeme do zátvoriek súradnice protíahlých vrcholov obdĺžnika:



5. Úloha: Zmeň svoj program 05lek_p06.py tak, aby nakreslil obdĺžnik, ktorý má súradnice protíahlých vrcholov $[50, 30]$ a $[300, 200]$.

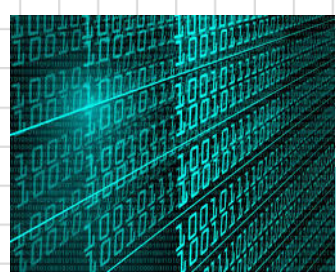
```
*05lek_p05.py - F:/14_Python(SŠ)/PythonSŠ_PracovnéListy/05_PL (riešenia)/05lek_p05.py
File Edit Format Run Options Window Help
import tkinter
canvas = tkinter.Canvas()
canvas.pack()

canvas.create_rectangle(50, 30, 300, 200)
```

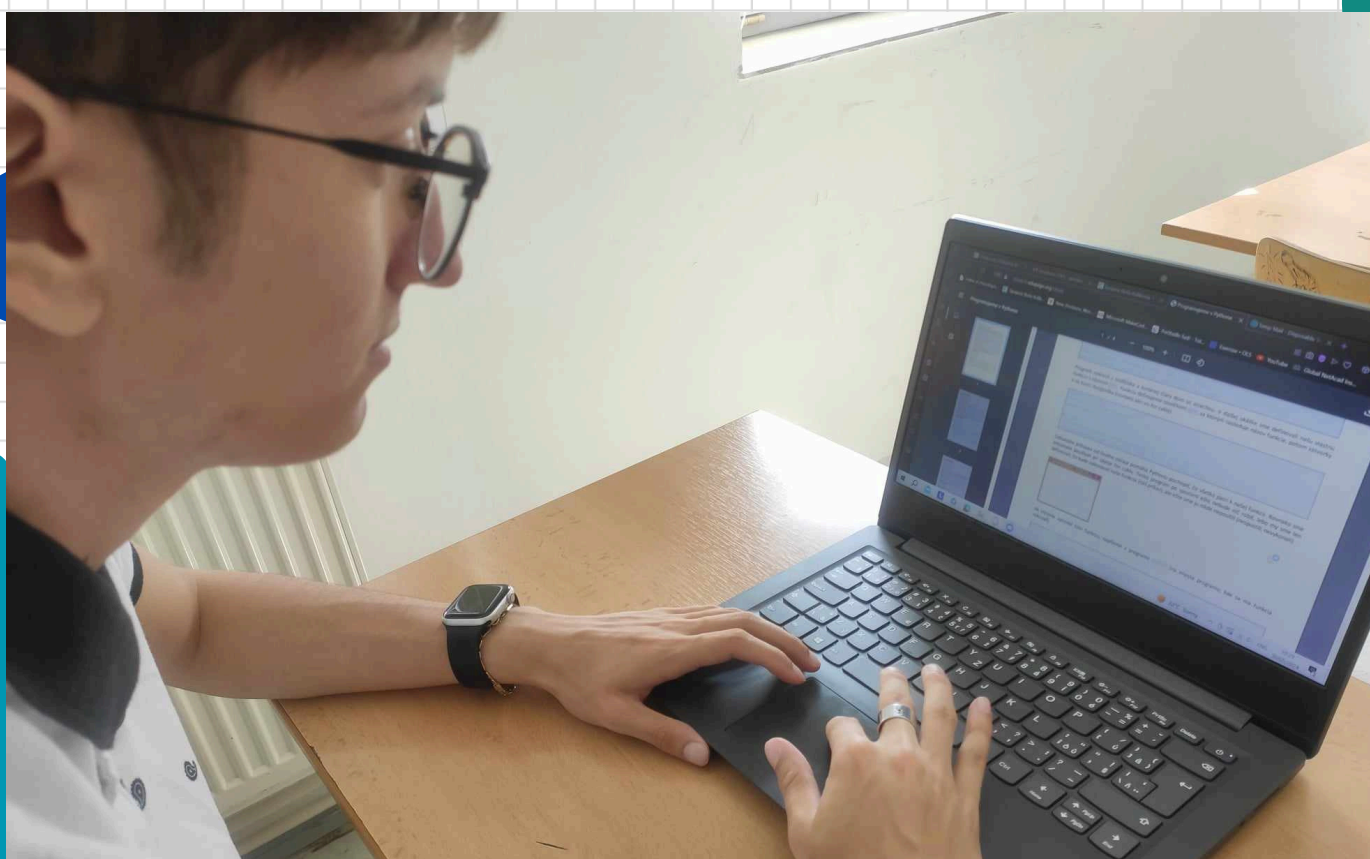
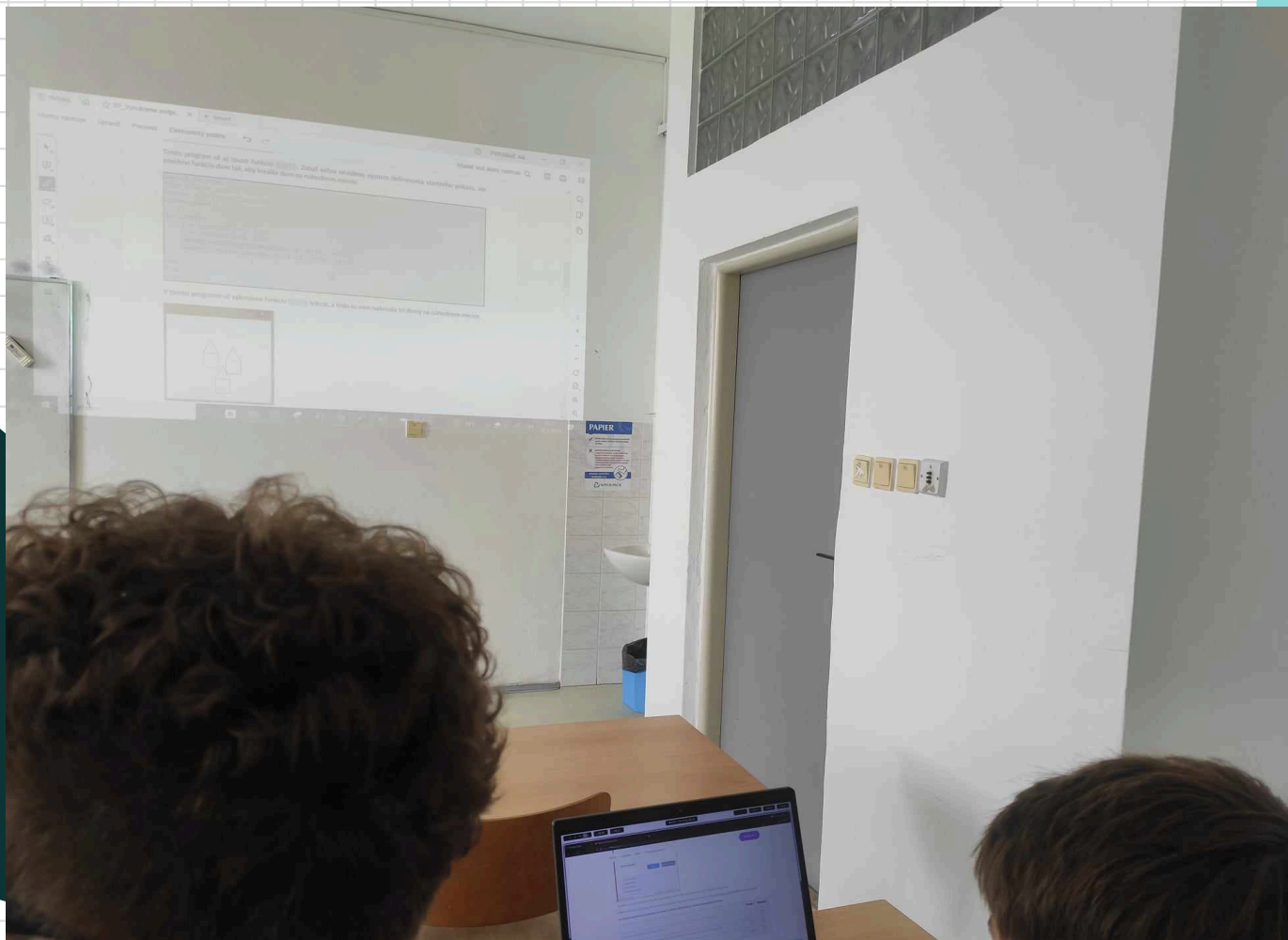
6. Úloha: Spočítaj bez použitia počítača, akú šírku a výšku má obdĺžnik z predchádzajúceho príkladu. Šírka: $300 - 50 = 250$
Výška: $200 - 30 = 170$
7. Úloha: Vytvor nový program 05lek_p08.py a nakresli obdĺžnik, ktorý má jeden vrchol na súradniciach $[200, 100]$, jeho šírka je 60 a výška 140.



PRÍLOHA



FOTODOKUMENTÁCIA VYUČOVACEJ HODINY

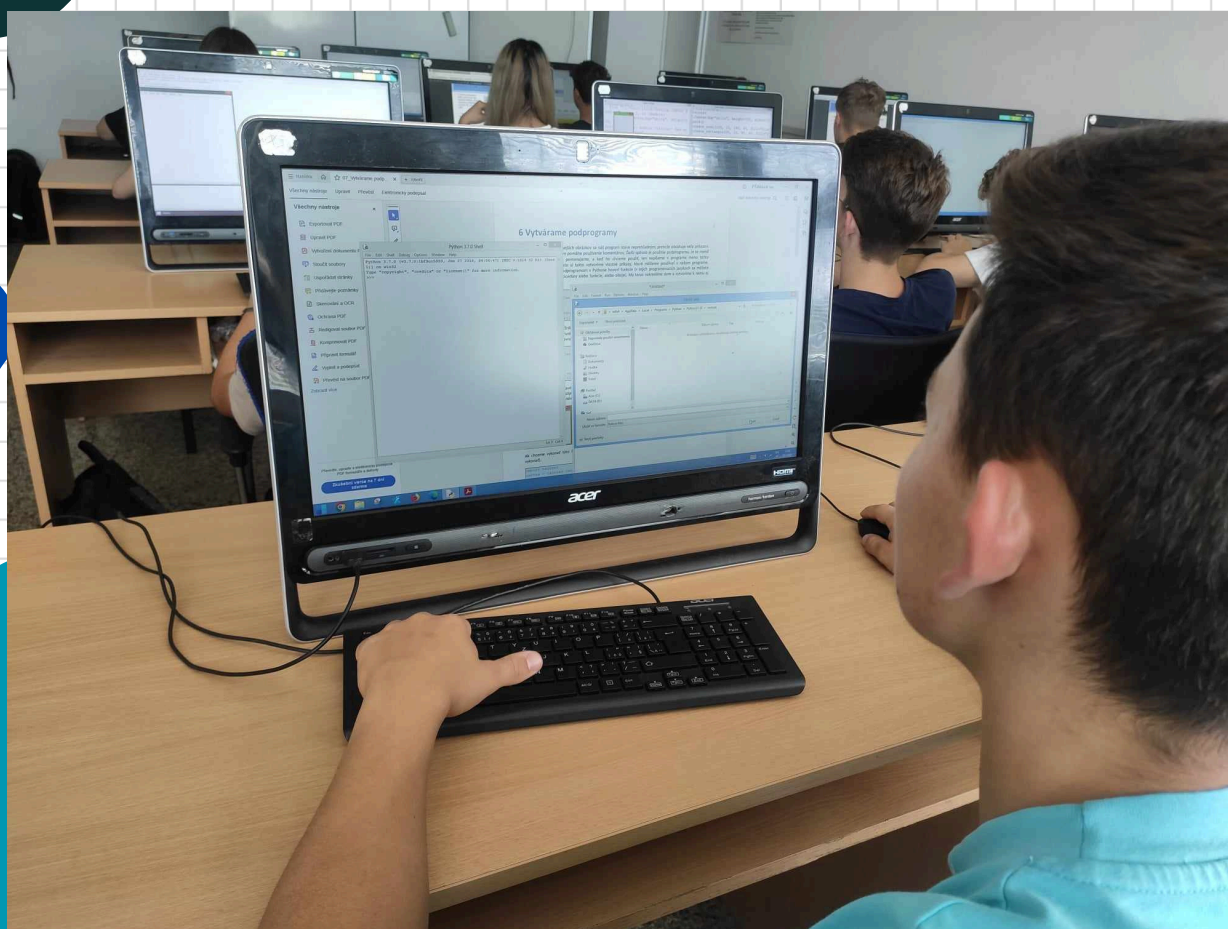
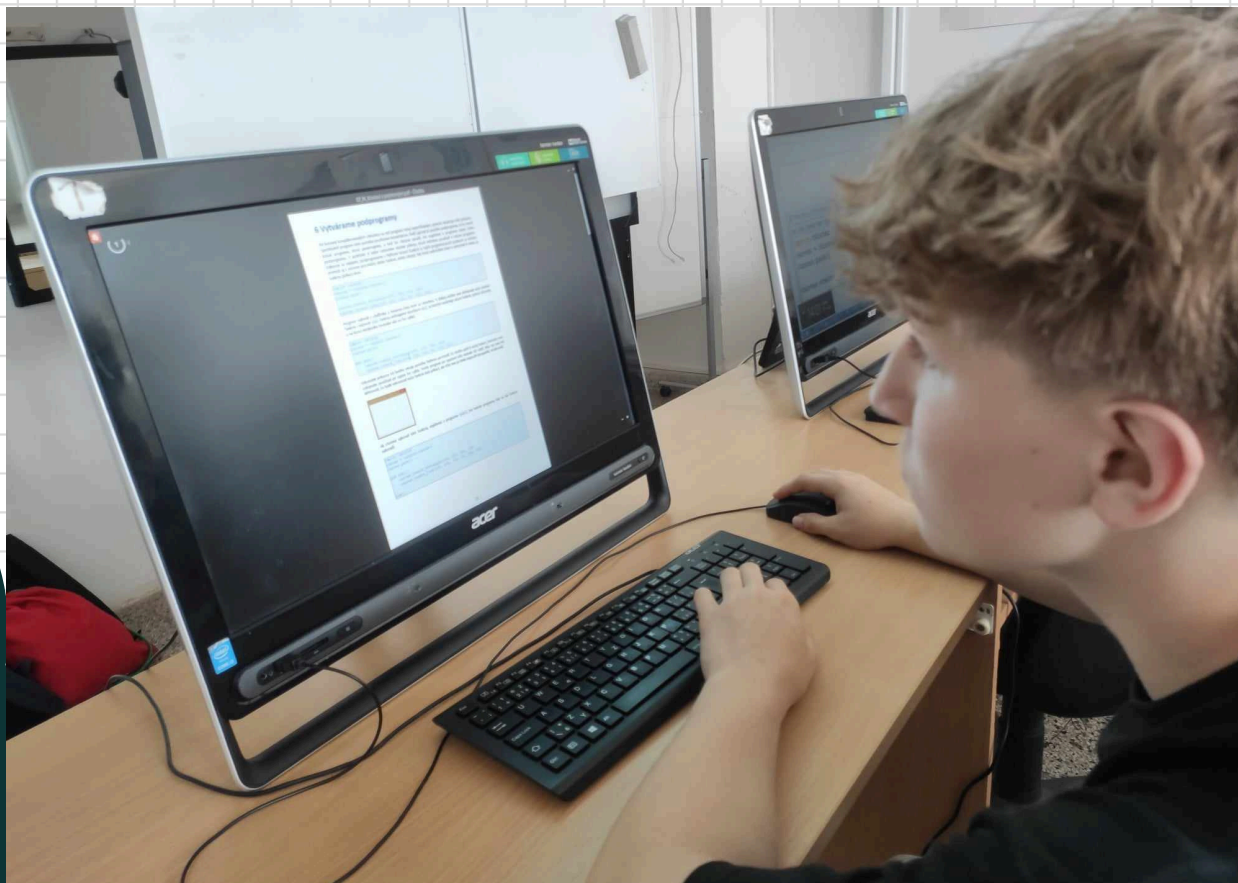




PRÍLOHA

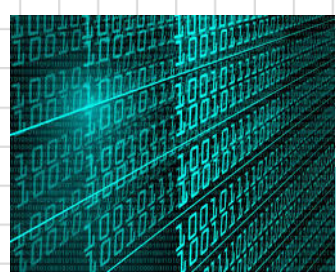


FOTODOKUMENTÁCIA VYUČOVACEJ HODINY

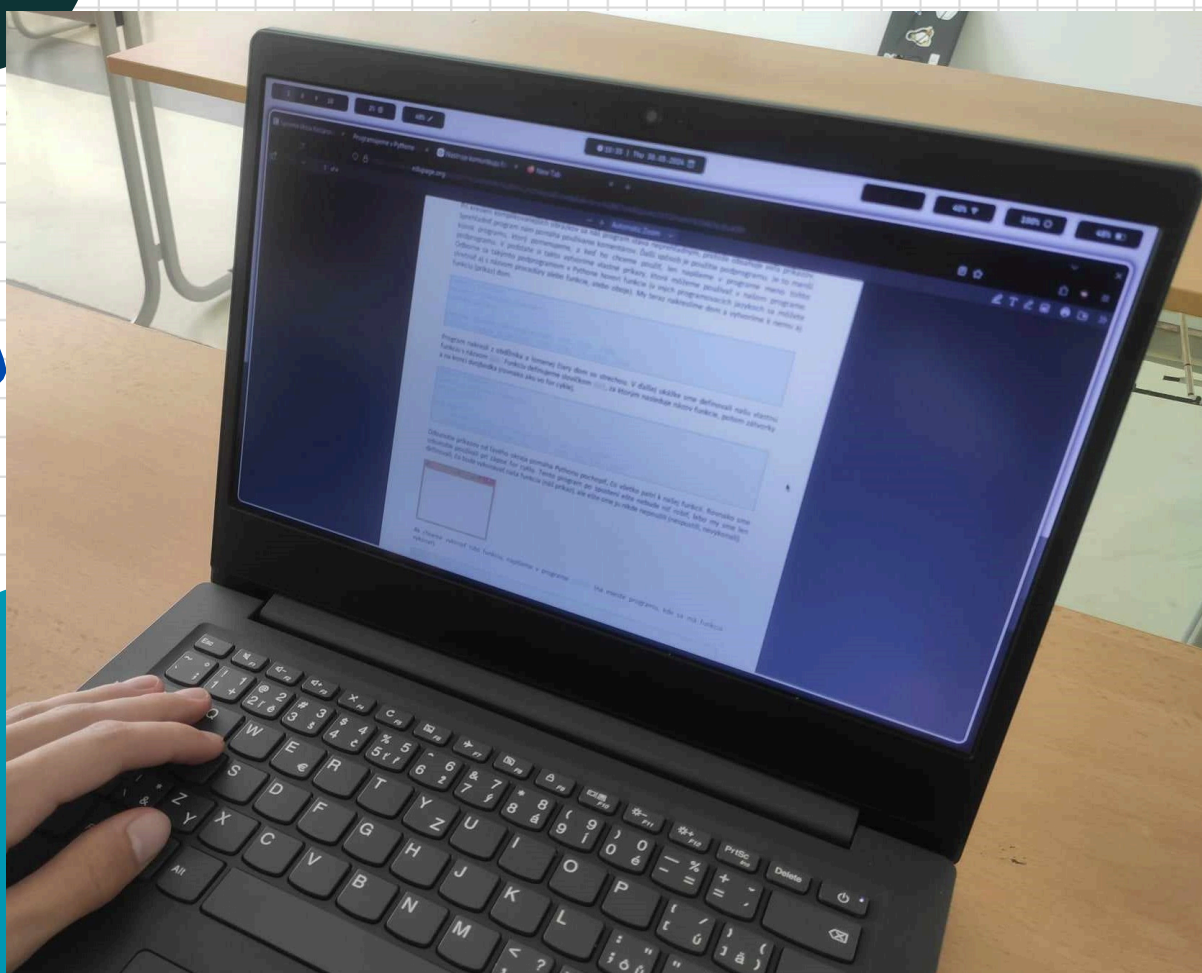
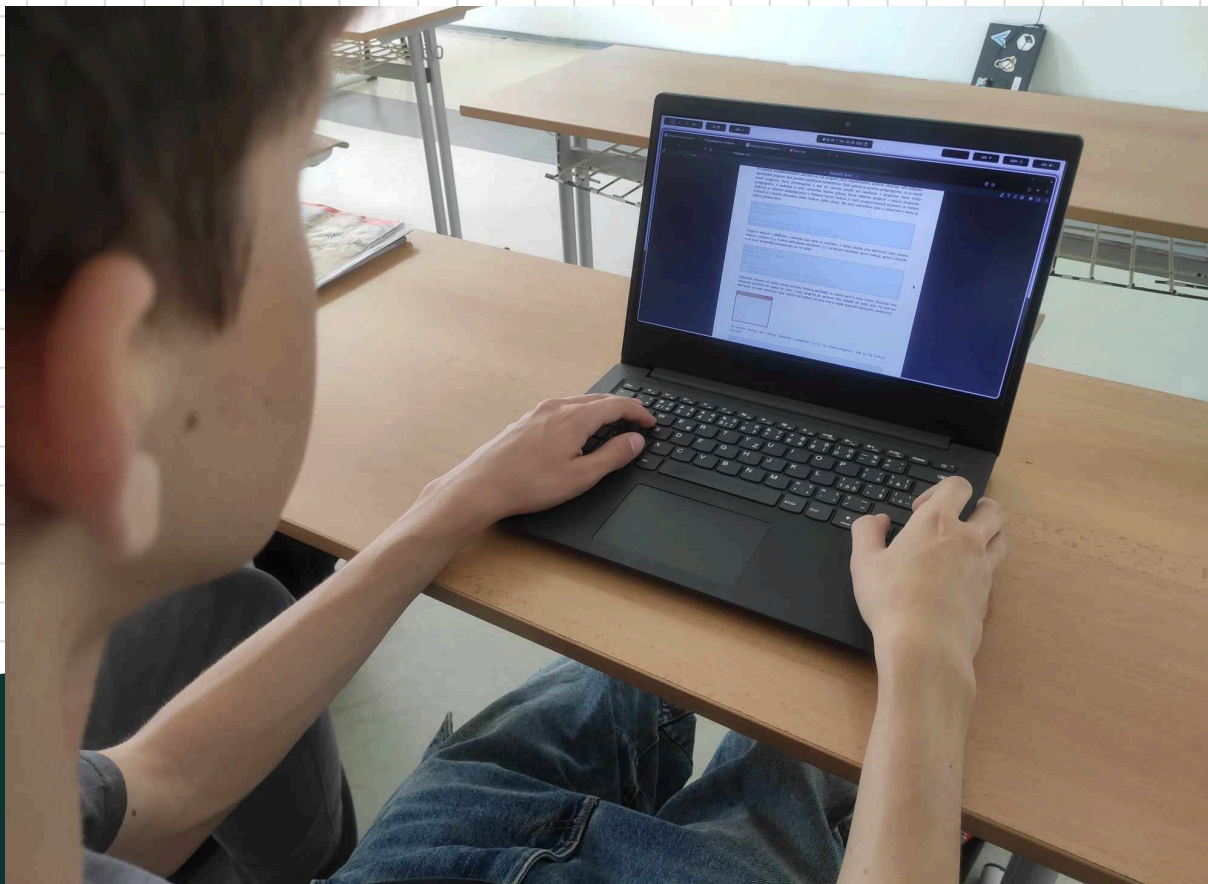




PRÍLOHA

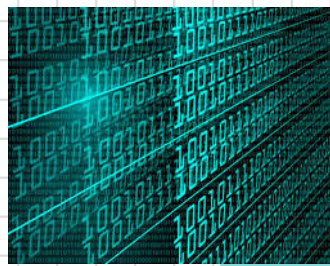


PFOTODOKUMENTÁCIA VYUČOVACEJ HODINY





PRÍLOHA



PRÁCA ŽIAKOV:

```
FROM TKINTER IMPORT *  
CREATE A TKINTER WINDOW  
ROOT = TK()  
CREATE A CANVAS TO DRAW ON  
CANVAS = CANVAS(ROOT, WIDTH=300, HEIGHT=300) CANVAS.PACK()  
CREATE A SQUARE  
CANVAS.CREATE_RECTANGLE(50, 50, 80, 80, FILL="BLUE")  
CREATE A RECTANGLE  
CANVAS.CREATE_RECTANGLE(100, 100, 160, 140, FILL="RED")  
RUN THE TKINTER MAIN LOOP  
ROOT.MAINLOOP()
```




PLÁN HODINY



PREDMET: PROGRAMOVANIE

TÉMA: PROGRAMOVANIE V PYTHONÉ – KRESLENIE A FARBY.

• UČEBNÉ CIELE:.

- 1. POROZUMIEŤ ZÁKLADNÝM PRINCÍPOM KRESLENIA V PYTHONÉ POMOCOU KNIŽNICE TKINTER**
- 2. NAUČIŤ SA POUŽÍVAŤ RÔZNE FARBY A EFEKTY PRI KRESLENÍ V PYTHONÉ.**

POMÔCKY:

**POČÍTAČ S NAINŠTALOVANÝM
PYTHONOM A KNIŽNICOU TKINTER.
VZOROVÁ UČEBNICA PYTHONU
PRACOVNÝ LIST S CVIČENIAMÍ**

**INTRO: POUŽÍVANIE RÔZNYCH FARIEB PRI
KRESLENÍ:**

- DEFINOVANIE FARBY POMOCOU RGB HODNÔT**
 - POUŽITIE RÔZNYCH FARIEB PRE JEDNOTLIVÉ OBJEKTY KRESLENIA**
- VYTVORENIE EFEKTOV POMOCOU
KOMBINÁCIÍ FARIEB**

ŠTRUKTÚRA HODINY: ZAPOJENIE (4 MINÚTY) – VYTVORENIE JEDNODUCHÉHO OBRÁZKA POMOCOU KRESLENIA V PYTHONÉ S VYUŽITÍM FARIEB A EFEKTOV

ROZHOVOR SO ŽIAKMI O ZÁKLADNÝCH PRINCÍPOV KRESLENIA A PRÁCE S FARBAMI V PYTHONÉ POMOCOU KNIŽNICE TKINTER

- VÝSKUM (12 MINÚT) – SKÚMANIE NOVÝCH PRÍKAZOV PRE VÝPLŇ A OBRYS OBRÁZKA**
- VÝKLAD (15 MINÚT) – ZHRNUTIE A VYSVETLENIE POZNATKOV Z ČASTI SKÚMANIE PROBLEMATIKY TVORBY PROGRAMOV SO ZÁKLADNÝCH PRINCÍPOV KRESLENIA A PRÁCE S FARBAMI V PYTHONÉ**
- VYPRACOVANIE (9 MINÚT) – PODPORA ŽIAKOV PRI ĎALŠOM SAMOSTATNOM KRESLENÍ A OBJAVOVANÍ MOŽNOSTÍ PROGRAMOVANIA V PYTHONÉ S VYUŽITÍM FARIEB**
- HODNOTENIE (5 MINÚT) – RIEŠENIE SEBAHODNOTIACEHO TESTU, DISKUSIA O ODPOVEDIACH.**

1. PRACOVNÝ LIST PRE ŽIAKA

**POSTUPOVAŤ BUDEME PODĽA NAŠEJ ELEKTRONICKEJ UČEBNICE PROGRAMOVANIE V PYTHONÉ, KTORÁ JE URČENÁ PRE STREDNÉ ŠKOLY. VZOR UČEBNICE NÁJDETE NA:
[HTTP://WWW.PROGRAMUJEMEVPYTHONE.SK/UKAZKA.PDF](http://www.programujemevpythone.sk/ukazka.pdf)**

UKÁŽKA OBSAHUJE JEDNODUCHÉ PRÍKLADY KRESLENIA VO FARBE:

1. VÝPLŇ TVAROV:

KRUHU, OBDĽŽNIKA, ŠTVORCA A TROJUHLNÍKA

OBRYS RÔZNYCH FARIEB A HRÚBOK ČIAR

UKÁŽKA KÓDU PRE VÝPLŇ A OBRYS

FÁZA EXPOZÍCIE – EXPOZÍCIA NOVÝCH PRÍKAZOV

– NA TABUĽU SI STRUČNE POZNAČÍM ZÁKLADNÉ POJMY



PLÁN HODINY

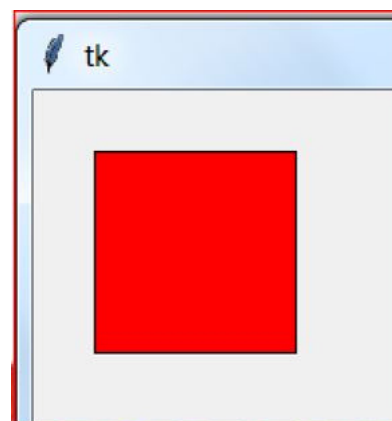


2. PRACOVNÝ LIST PRE ŽIAKA

1. Úloha: Vytvor nový program 06lek_p02.py, pomocou nasledujúceho kódu nakresli vyfarbený obdĺžnik:

```
*06lek_p02.py - F:/14_Python(SŠ)/PythonSŠ_PracovnéListy/06_PL (riešenia)/06lek_p02.py (3.6.2rc1)*  
File Edit Format Run Options Window Help  
import tkinter  
canvas = tkinter.Canvas()  
canvas.pack()  
  
canvas.create_rectangle(30, 30, 130, 130, fill='red')
```

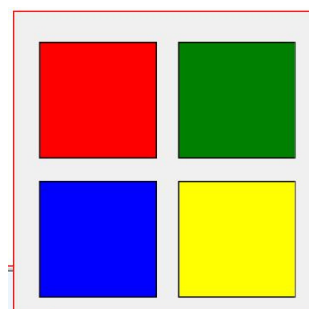
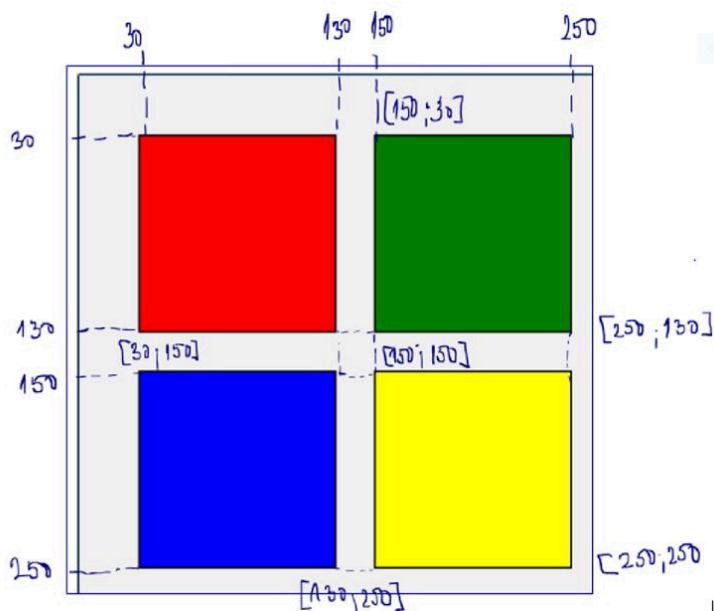
Šírka (aj výška) štvorca je 100



2. Úloha: Pridaj do programu 06lek_p03.py ďalšie 3 príkazy na kreslenie obdĺžnikov, aby si dostal nasledujúci obrázok:

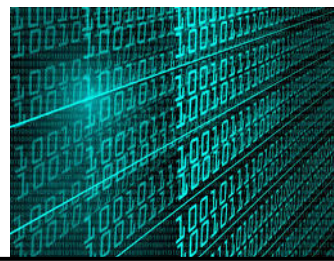
Šírka (aj výška) štvorca je 100

Vzdialenosť medzi štvorcami je 20





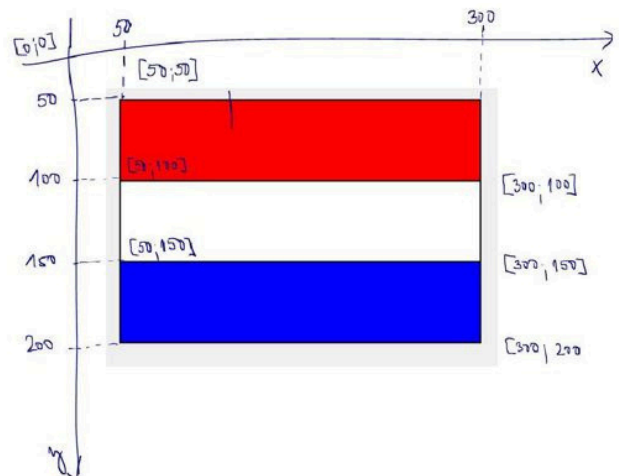
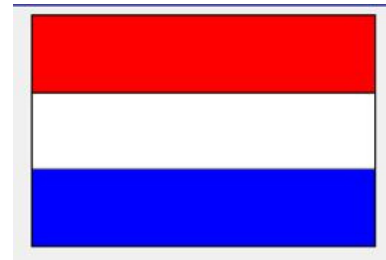
PLÁN HODINY



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

canvas.create_rectangle(30, 30, 130, 130, fill='red')
canvas.create_rectangle(150, 30, 250, 130, fill='green')
canvas.create_rectangle(30, 150, 130, 250, fill='blue')
canvas.create_rectangle(150, 150, 250, 250, fill='yellow')
```

3. Úloha: Vytvor nový program 06lek_p04.py, ktorý nakreslí holandskú vlajku:



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

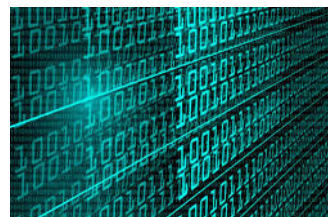
```
canvas.create_rectangle(50, 50, 300, 100, fill='red')
canvas.create_rectangle(50, 100, 300, 150, fill='white')
canvas.create_rectangle(50, 150, 300, 200, fill='blue')
```

4. Úloha: Vytvor program 06lek_p05.py, ktorý nakreslí írsku vlajku s farbou 'orange':

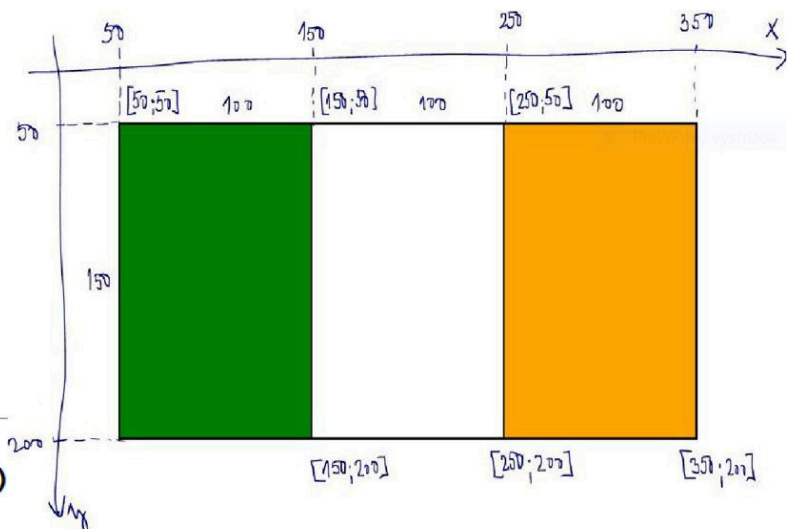




PLÁN HODINY



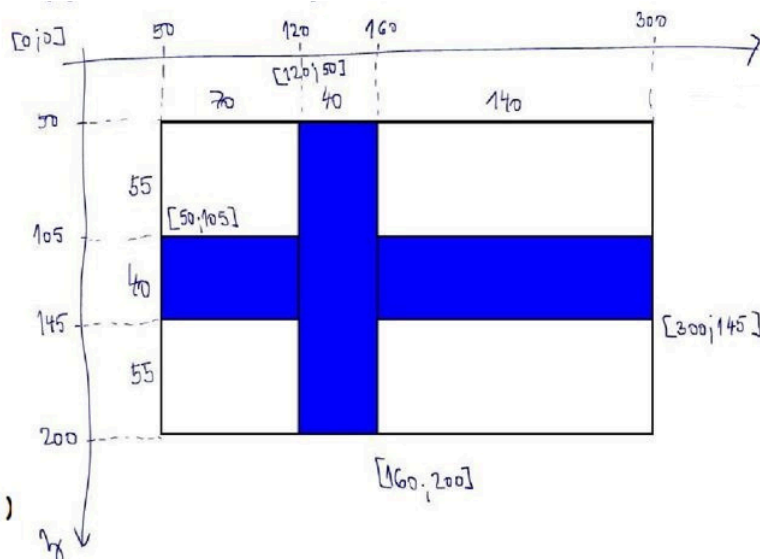
PRACOVNÝ LIST PRE ŽIAKA



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

canvas.create_rectangle(50, 50, 150, 200, fill='green')
canvas.create_rectangle(150, 50, 250, 200, fill='white')
canvas.create_rectangle(250, 50, 350, 200, fill='orange')
```

5. Úloha: Vytvor nový program 06lek_p06.py a nakresli takýto obrázok: Vlajke ktorého štátu sa obrázok podobá?

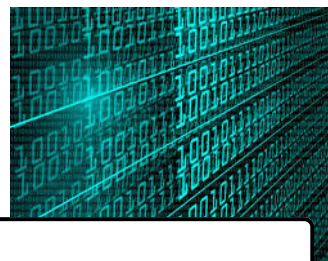


```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()

canvas.create_rectangle(50, 50, 300, 200, fill='white')
canvas.create_rectangle(50, 105, 300, 145, fill='blue')
canvas.create_rectangle(120, 50, 160, 200, fill='blue')
```

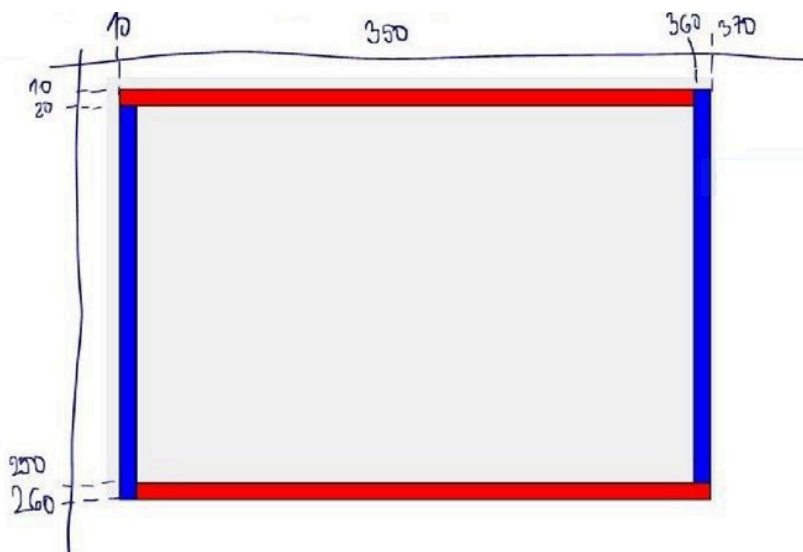


PLÁN HODINY



PRACOVNÝ LIST PRE ŽIAKA

6. Úloha: Vytvor nový program 06lek_p07.py, v ktorom zo štyroch úzkych obdĺžnikov nakresli takýto rámček:



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

canvas.create_rectangle(10, 10, 360, 20, fill='red')
canvas.create_rectangle(10, 20, 20, 260, fill='blue')
canvas.create_rectangle(20, 250, 370, 260, fill='red')
canvas.create_rectangle(360, 10, 370, 250, fill='blue')
```

PRI KRESLENÍ V PYTHONĚ MÁME MOŽNOSŤ VYUŽIŤ VEĽA FARIEB. TU JE VÝBER NIEKTORÝCH Z NICH:

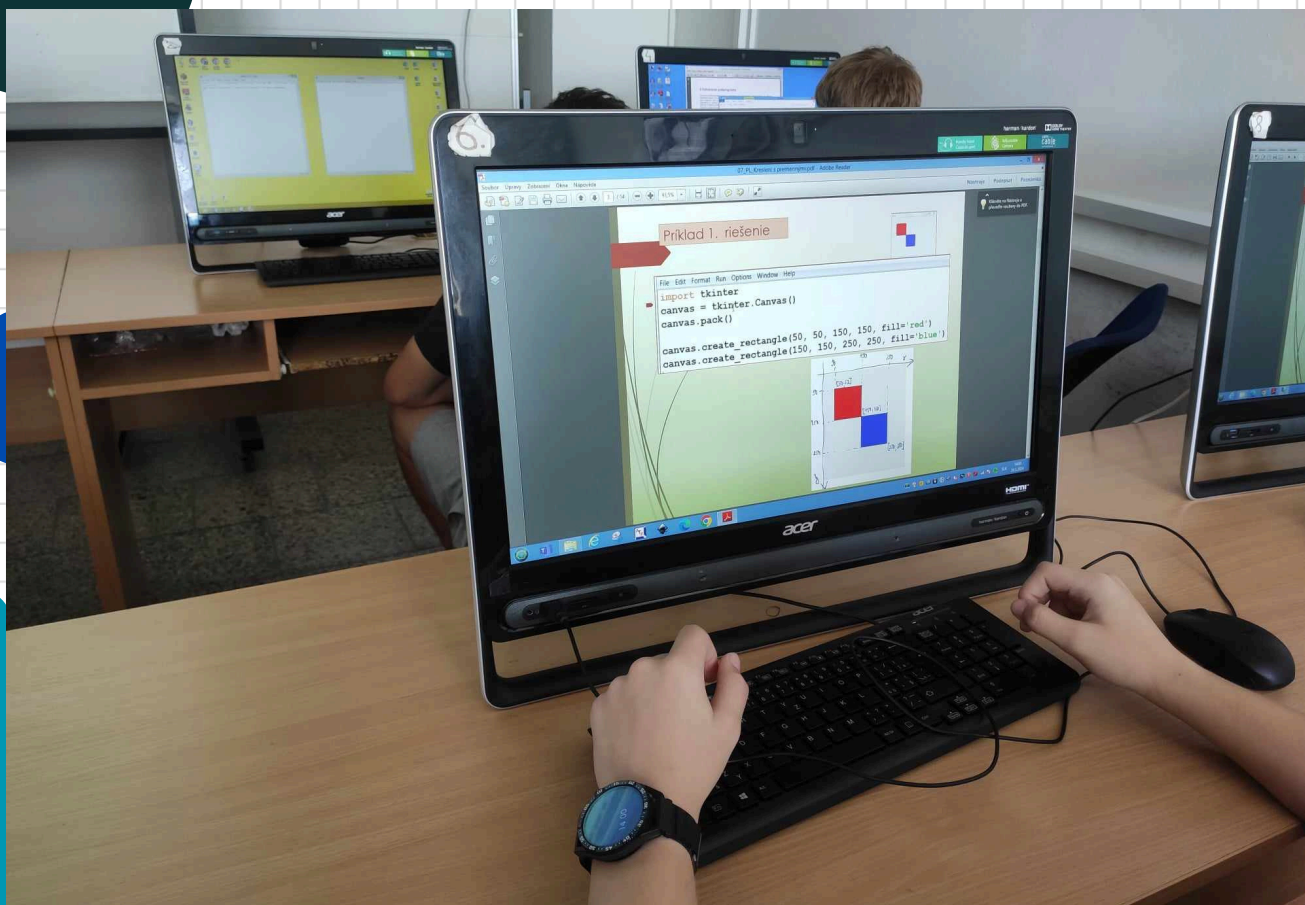
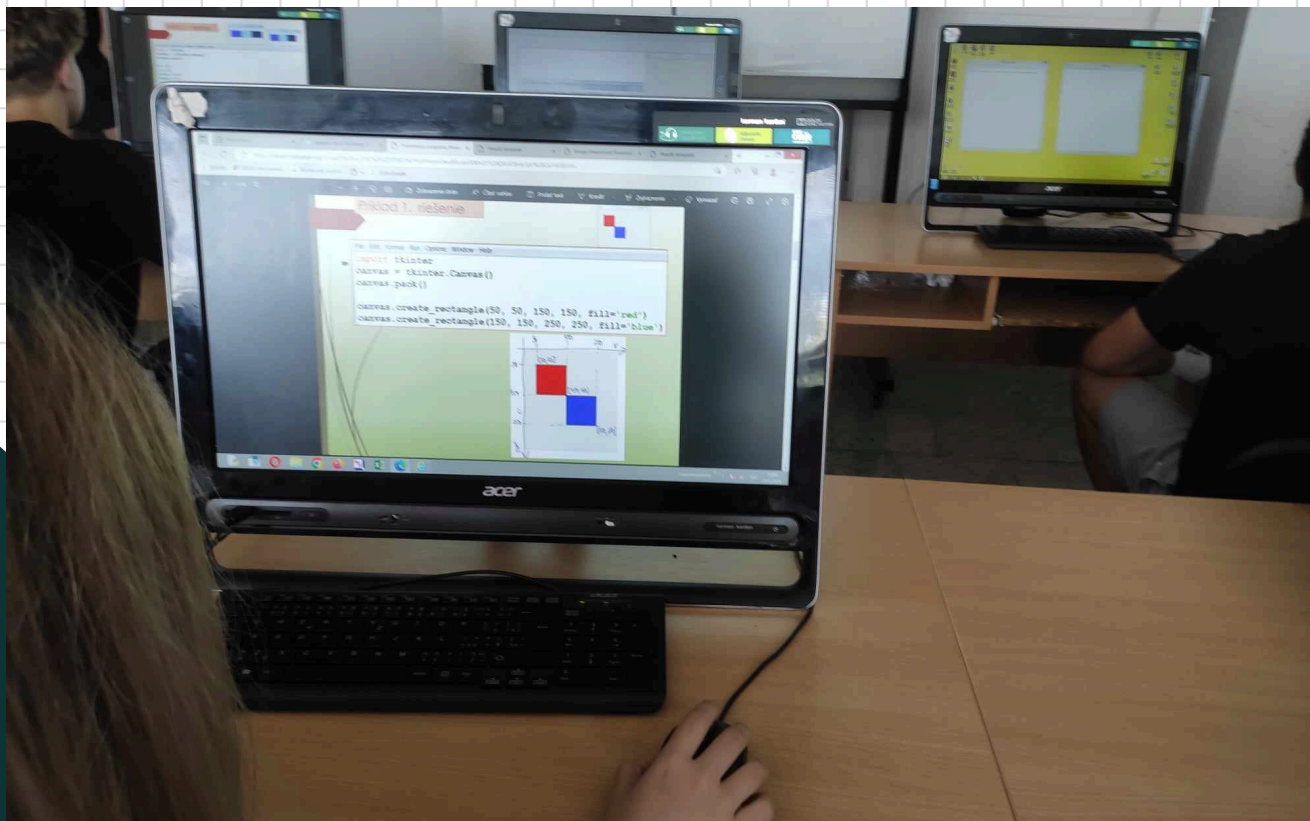
Blue	LightBlue	Cyan	SkyBlue	CornFlowerBlue	DeepSkyBlue	DodgerBlue
RoyalBlue	SlateBlue	SteelBlue	MediumBlue	Navy	Red	SandyBrown
Salmon	Coral	Tomato	Orange	DarkOrange	OrangeRed	IndianRed
Chocolate	Tan	Maroon	Sienna	Brown	SaddleBrown	Pink
Plum	Violet	Orchid	Magenta	Purple	DarkMagenta	Green
PaleGreen	YellowGreen	MediumSeaGreen	LawnGreen	LimeGreen	ForestGreen	DarkGreen
Yellow	Khaki	Gold	Gray	LightGray	Black	White



PRÍLOHA

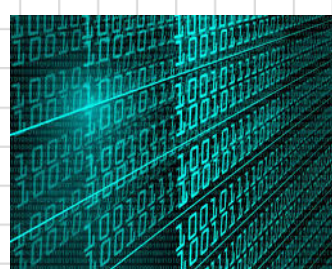


FOTODOKUMENTÁCIA VYUČOVACEJ HODINY

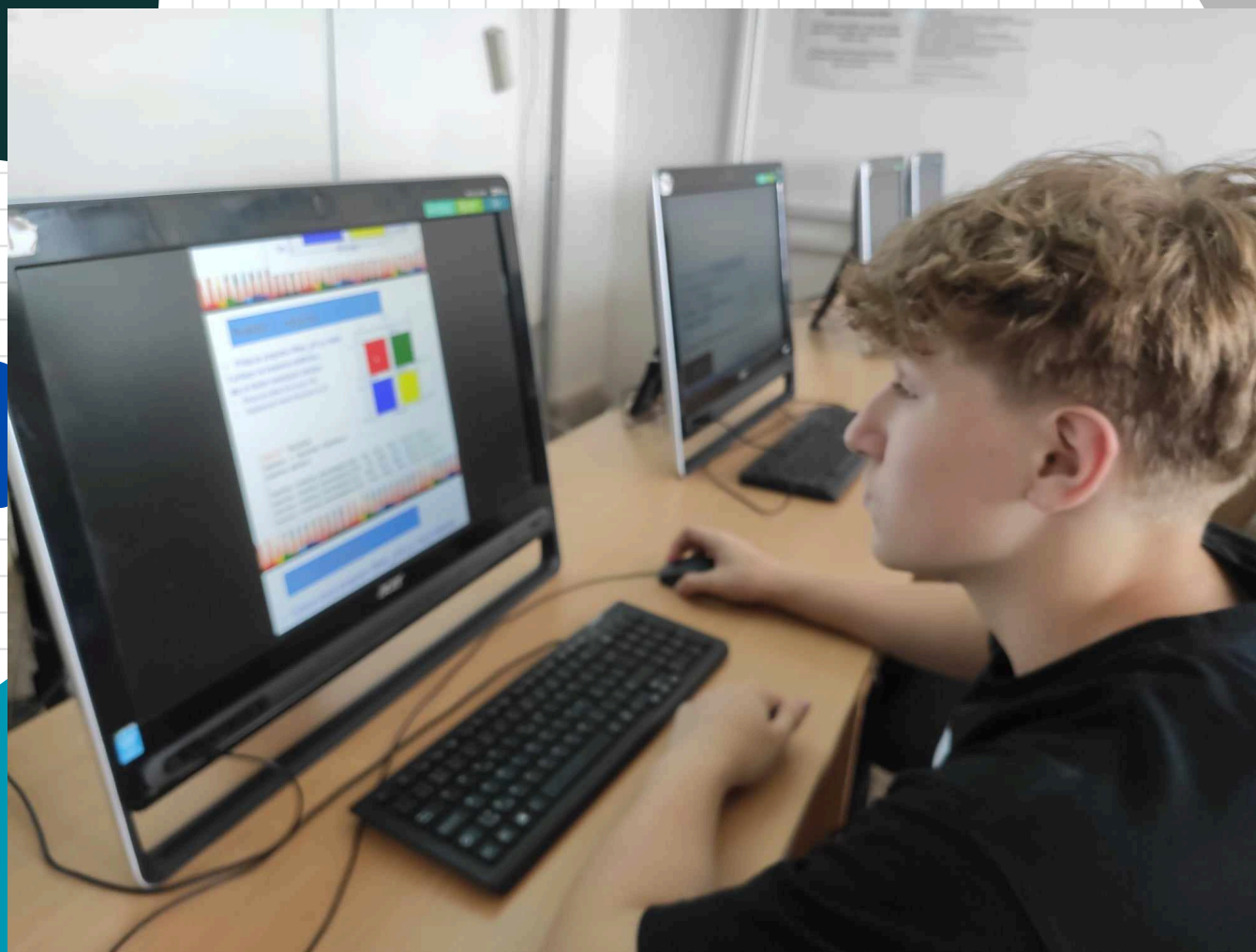
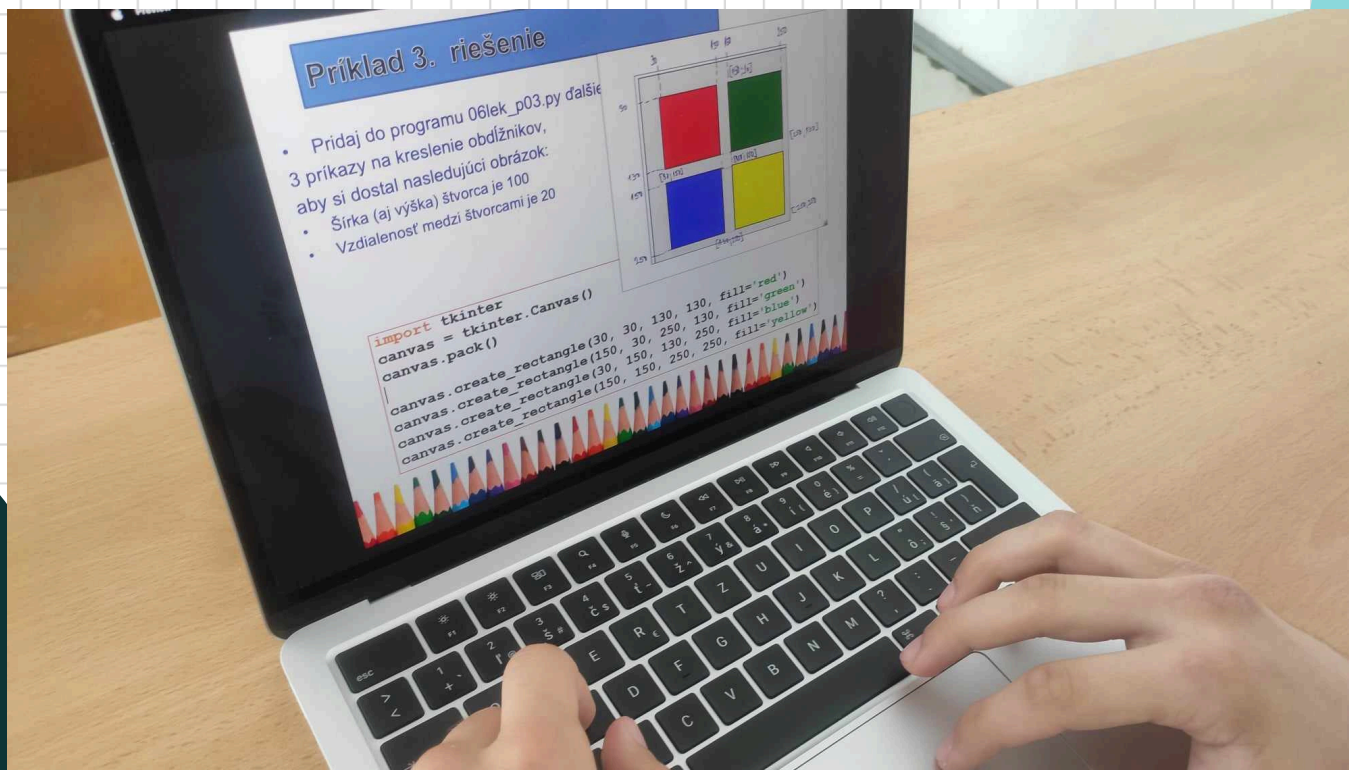




PRÍLOHA

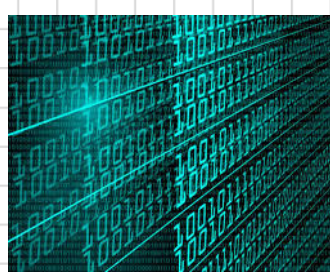


FOTODOKUMENTÁCIA VYUČOVACEJ HODINY

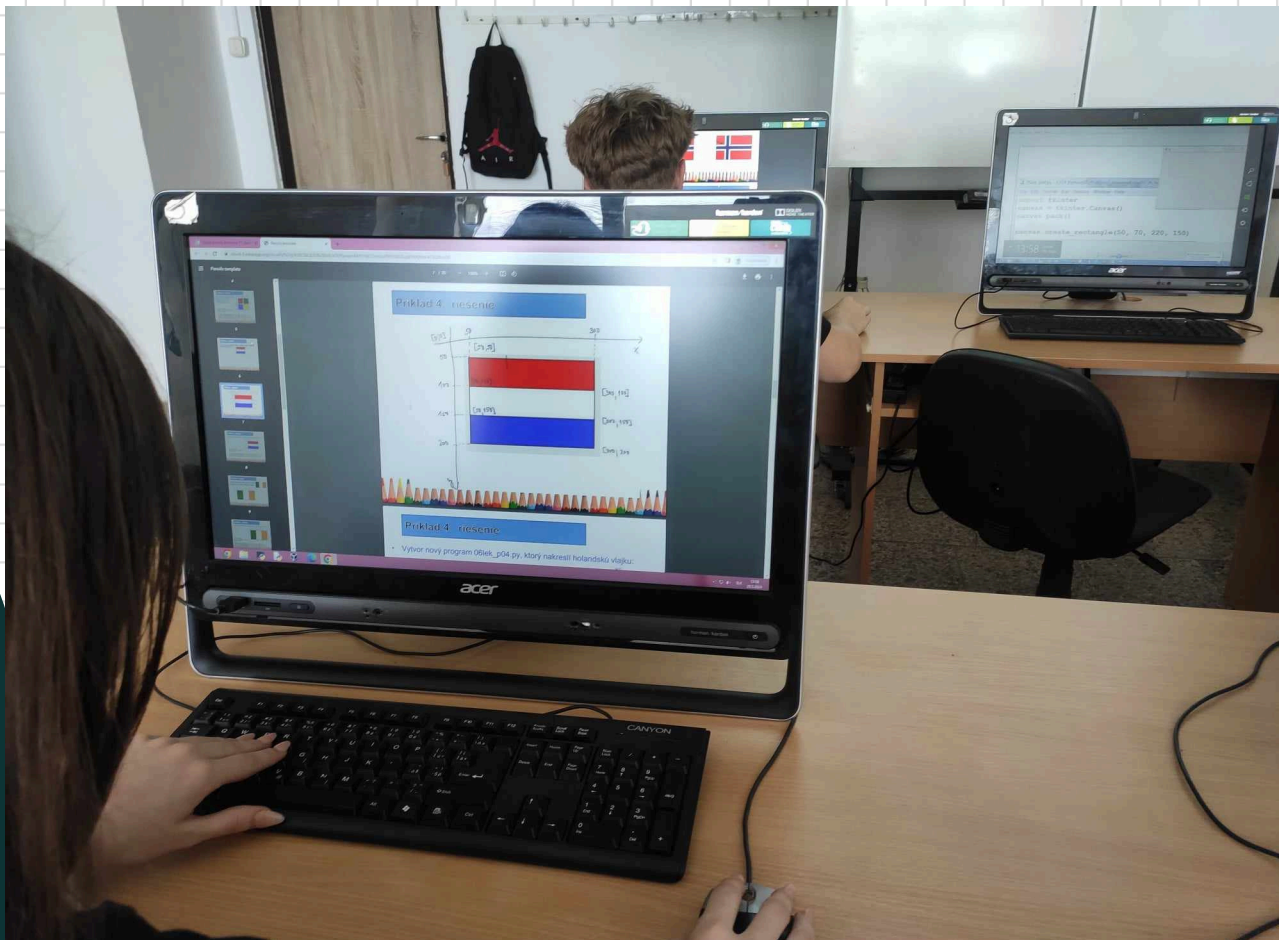




PRÍLOHA

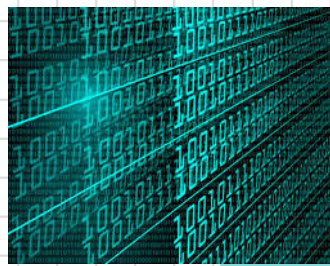


FOTODOKUMENTÁCIA VYUČOVACEJ HODINY





PRÍLOHA



PRÁCA ŽIAKOV

```
import tkinter

# VYTVORÍME OKNO
OKNO = tkinter.Tk()
OKNO.title("TRI OBDĽŽNIKY")

# NASTAVÍME VEĽKOSŤ OKNA
SIRKA = 400
VYSKA = 200
OKNO.geometry(f"{SIRKA}x{VYSKA}")

# VYTVORÍME PLÁTNO
CANVAS = tkinter.Canvas(OKNO, width=SIRKA, height=VYSKA)
CANVAS.pack()

# PRVÝ OBDĽŽNIK
X1 = 50
Y1 = 20
X2 = 150
Y2 = 80
FARBA1 = "red"
CANVAS.create_rectangle(X1, Y1, X2, Y2, fill=FARBA1)

# DRUHÝ OBDĽŽNIK
X2 = 200
Y2 = 50
FARBA2 = "blue"
CANVAS.create_rectangle(X1, Y1, X2, Y2, fill=FARBA2)

# TRETÍ OBDĽŽNIK
X1 = 100
Y1 = 100
X2 = 250
Y2 = 150
FARBA3 = "green"
CANVAS.create_rectangle(X1, Y1, X2, Y2, fill=FARBA3)
```




PLÁN HODINY



PREDMET: PROGRAMOVANIE

TÉMA: PROGRAMOVANIE V PYTHONÉ – KRESLENIE S PREMENNÝMI.

• UČEBNÉ CIELE:.

1. POROZUMIEŤ ZÁKLADNÝM PRINCÍPOM KRESLENIA V PYTHONÉ POMOCOU PREMENNÝCH.

2. NAUČIŤ SA POUŽÍVAŤ PREMENNÉ PRI KRESLENÍ V PYTHONÉ.

POMÔCKY:

**POČÍTAČ S NAINŠTALOVANÝM
PYTHONOM A KNIŽNICOU TKINTER.
VZOROVÁ UČEBNICA PYTHONU
PRACOVNÝ LIST S CVIČENIAM**

**INTRO: POUŽÍVANIE PREMENNÝCH PRI
KRESLENÍ:**

- **DEFINOVANIE A POUŽÍVANIE
PREMENNÝCH V PYTHONÉ**
- **POUŽIŤ PREMENNÉ NA KRESLENIE
TVAROV.**
- **NASTAVIŤ FARBU A VEĽKOSŤ TVAROV.**

- **ŠTRUKTÚRA HODINY: ZAPOJENIE (4 MINÚTY) – VYTVORENIE OBRÁZKA POMOCOU KRESLENIA
V PYTHONÉ POMOCOU PREMENNEJ**

**ROZHOVOR SO ŽIAKMI O ZÁKLADNÝCH PRINCÍPOV KRESLENIA A PRÁCE S FARBAMI V PYTHONÉ
POMOCOU KNIŽNICE TKINTER**

- **VÝSKUM (12 MINÚT) – SKÚMANIE NOVÝCH PRÍKAZOV S PREMENNOU PRE VÝPLŇ A OBRYŠ OBRÁZKA**
- **VÝKLAD (15 MINÚT) – ZHRNUTIE A NAPÍSANIE JEDNODUCHÉHO PROGRAMU V PYTHONÉ, KTORÝ KRESLÍ
RÔZNE TVARY S RÔZNYMI FARBAMI A VEĽKOSŤAMI**
- VYPRACOVANIE (9 MINÚT) – PODPORA ŽIAKOV PRI ĎALŠOM SAMOSTATNOM KRESLENÍ MOŽNOSTÍ
PROGRAMOVANIA V PYTHONÉ S VYUŽITÍM PREMENNÝCH**
- **HODNOTENIE (5 MINÚT) – RIEŠENIE SEBAHODNOTIACEHO TESTU, DISKUSIA O ODPOVEDIACH.**

1. PRACOVNÝ LIST PRE ŽIAKA

**POSTUPOVAŤ BUDEME PODĽA NAŠEJ ELEKTRONICKEJ UČEBNICE PROGRAMOVANIE V PYTHONÉ,
KTORÁ JE URČENÁ PRE STREDNÉ ŠKOLY. VZOR UČEBNICE NÁJDETE NA:**

[HTTP://WWW.PROGRAMUJEMEVPYTHONE.SK/UKAZKA.PDF](http://www.programujemevpythone.sk/ukazka.pdf)

UKÁŽKA OBSAHUJE JEDNODUCHÉ PRÍKLADY KRESLENIA S PREMENNOU:

**1. PROGRAM, KTORÝ NAKRESLÍ ŠTVOREC, OBDĽŽNIKA, TROJUHOĽNÍK A KRUH. V PREMENNÝCH X, Y SÚ
ULOŽENÉ SÚRADNICE L'AVÉHO HORNÉHO ROHU OBJEKTU.**

**2. UKÁŽKA KÓDU S PREMENNÝMI PRE VÝPLŇ A OBRYŠ RÔZNYCH FARIEB A HRÚBOK ČIAR
FÁZA EXPOZÍCIE – EXPOZÍCIA NOVÝCH PRÍKAZOV S PREMENNÝMI**

– NA TABUĽU SI STRUČNE POZNAČÍM POJMOVÚ MAPU ZÁKLADNÝCH POJMOV

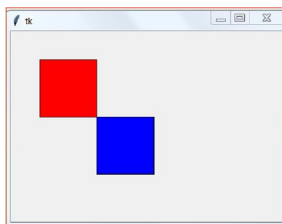


PLÁN HODINY



3. PRACOVNÝ LIST PRE ŽIAKA

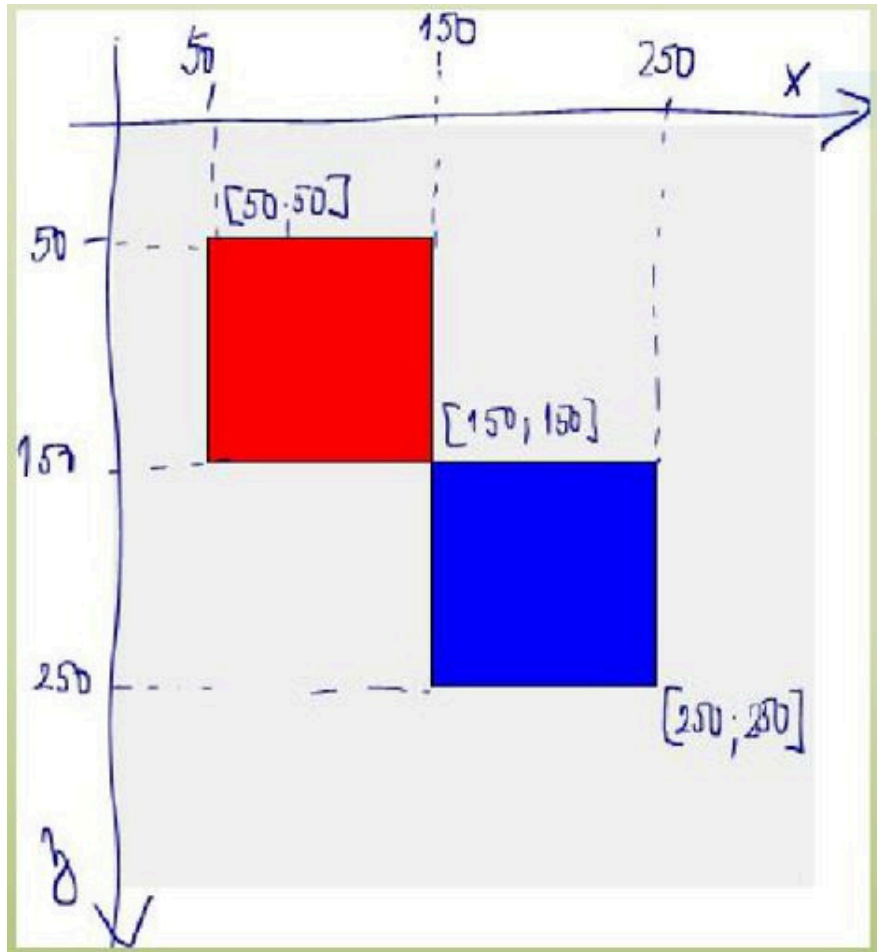
1.Úloha: Vytvor nový program 07lek_p01.py, ktorý nakreslí dva dotýkajúce sa štvorce ako na obrázku. Obidva majú dĺžku strany 100, pritom červený má ľavý horný roh v bode [50, 50].



File Edit Format Run Options Window Help

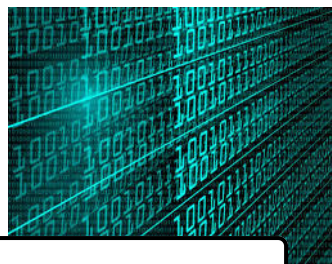
```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()

canvas.create_rectangle(50, 50, 150, 150, fill='red')
canvas.create_rectangle(150, 150, 250, 250, fill='blue')
```





PLÁN HODINY

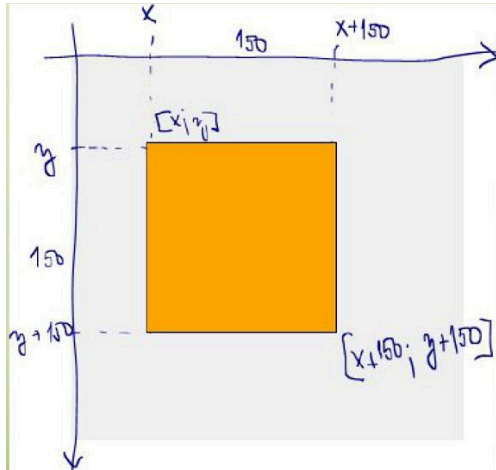
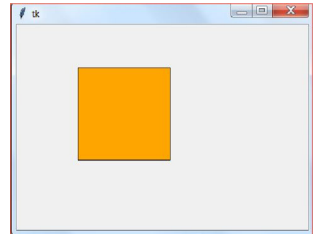


3. PRACOVNÝ LIST PRE ŽIAKA

2. Úloha: Vytvor nový program 07lek_p02.py, ktorý nakreslí štvorec so stranou dĺžky 150.

V premenných x, y sú uložené súradnice ľavého horného rohu štvorca.

Súradnice ľavého horného rohu $[x;y] = [100;70]$ sú zadane ako premenné.

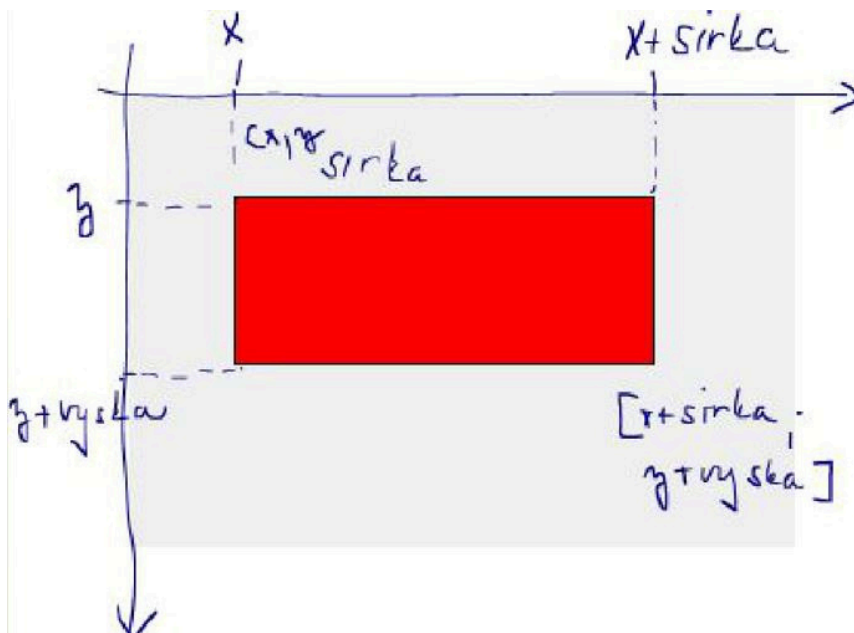


```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()

x = 100
y = 70
canvas.create_rectangle(x,y,x + 150,y + 150, fill='orange')
```

3. Úloha: Vytvor program 07lek_p03.py ktorý nakreslí obdĺžnik s ľavým horným rohom na súradniciach x, y , danú šírkou a výškou. Farbu si zvol podľa svojho.

Napríklad keď bude v programe: $x = 100, y = 70$, šírka = 200, výška = 80 nakreslí sa obdĺžnik ako na obrázku:





PLÁN HODINY



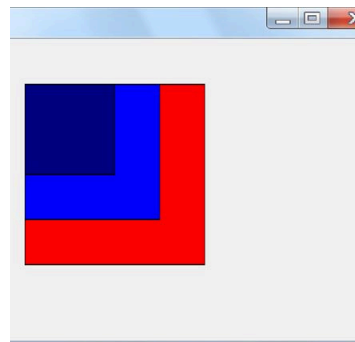
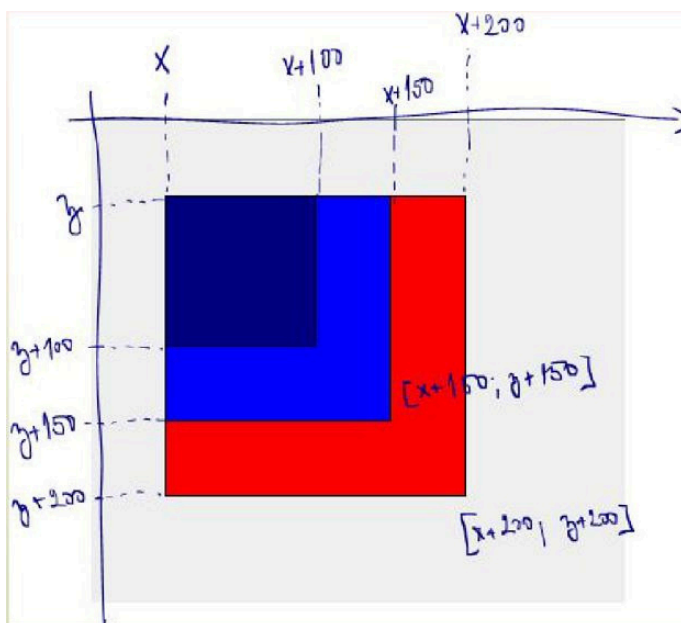
3. PRACOVNÝ LIST PRE ŽIAKA

File Edit Format Run Options Window Help

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

x = 100
y = 70
sirka = 200
vyska = 80
canvas.create_rectangle(x, y, x + sirka, y + vyska, fill='red')
```

4. Úloha: Vytvor program 07lek_p04.py, ktorý nakreslí štvorce ako na obrázku. Tieto štvorce majú spoločný ľavý horný roh, ktorého súradnice sú v premenných x, y.
Štvorce sa postupne zmenšujú tak, že červený má dĺžku strany 200, modrý 150 a tmavomodrý 100.
Súradnice stredu môžu byť napr.: [x;y] = [80;50]



File Edit Format Run Options Window Help

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

x = 80
y = 50
canvas.create_rectangle(x, y, x+200, y+200, fill='red')
canvas.create_rectangle(x, y, x+150, y+150, fill='blue')
canvas.create_rectangle(x, y, x+100, y+100, fill='navy')
```



PLÁN HODINY



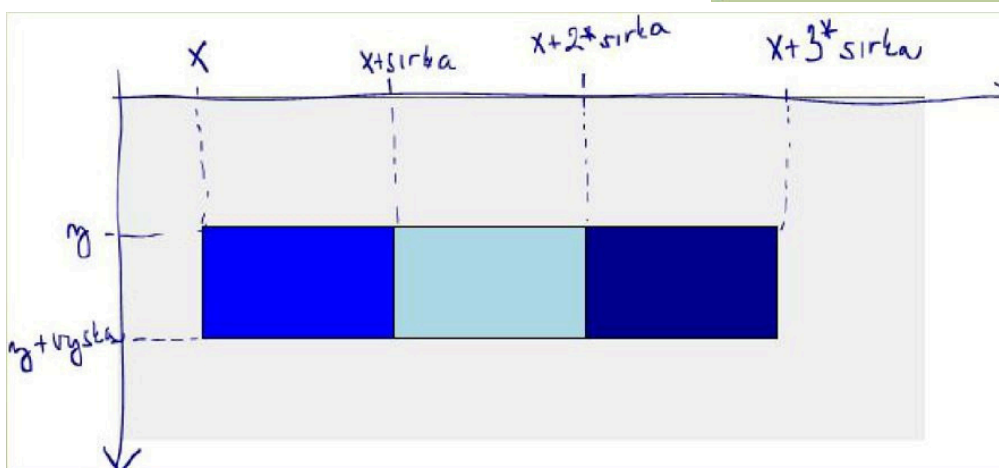
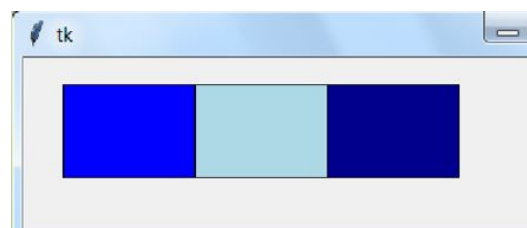
3. PRACOVNÝ LIST PRE ŽIAKA

5. Úloha: Vytvor program 07lek_p05.py, ktorý nakreslí tri vzájomne sa dotýkajúci obdĺžniky:

Súradnice ľavého horného rohu prvého obdĺžnika sú uložené v premenných x , y .

Všetky tri obdĺžniky majú rovnakú šírku a výšku - tieto rozmery sú uložené v premenných $sirka$, $vyska$.

Bude program fungovať správne aj v prípade, že hodnotu premennej $sirka$ zväčšíš o 20 a hodnotu premennej y zväčšíš o 10? Ak nie, program oprav.



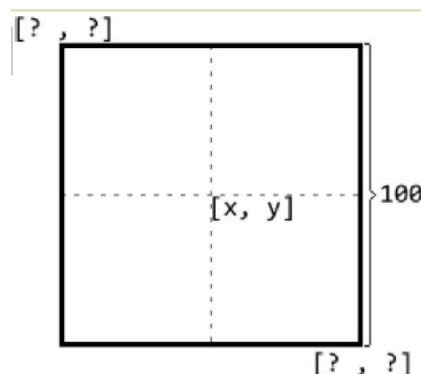
```
File Edit Format Run Options Window Help
import tkinter
canvas = tkinter.Canvas()
canvas.pack()

x = 30
y = 20
sirka = 100
vyska = 70
canvas.create_rectangle(x, y, x+sirka, y+vyska, fill='blue')
canvas.create_rectangle(x+sirka, y, x+2*sirka, y+vyska, fill='light blue')
canvas.create_rectangle(x+2*sirka, y, x+3*sirka, y+vyska, fill='dark blue')
```

6. Úloha: Vytvor nový program 07-lek_p06.py, ktorý nakreslí štvorec, ktorého stred má súradnice x , y a strany majú dĺžku 100.

Súradnice x , y sú uložené v rovnomenných premenných. Aby si mohol tento štvorec nakresliť, musíš vypočítať súradnice jeho ľavého horného i pravého dolného rohu.

Do nového programu napíš kód, ktorý nakreslí zelený štvorec so stredom $[x, y]$ a stranou s dĺžkou 100.



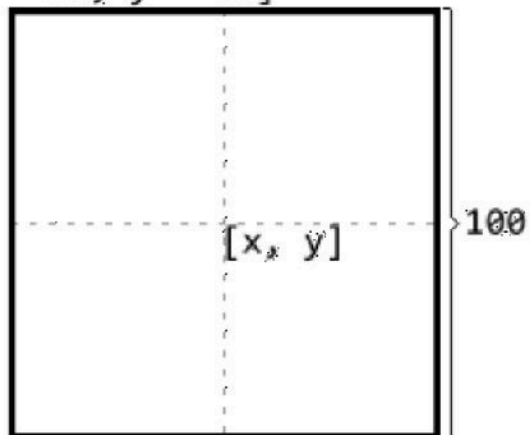


PLÁN HODINY

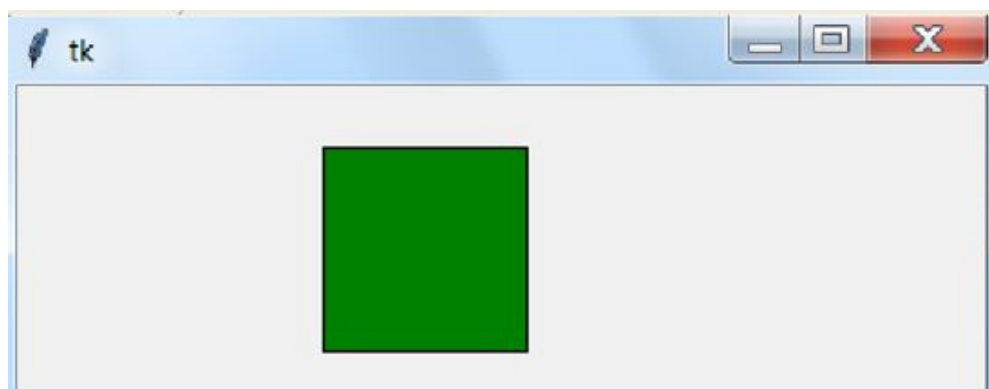


3. PRACOVNÝ LIST PRE ŽIAKA

$[x - 50, y - 50]$



$[x + 50, y + 50]$



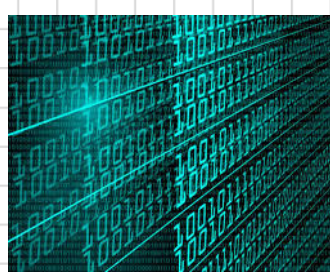
File Edit Format Run Options Window Help

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

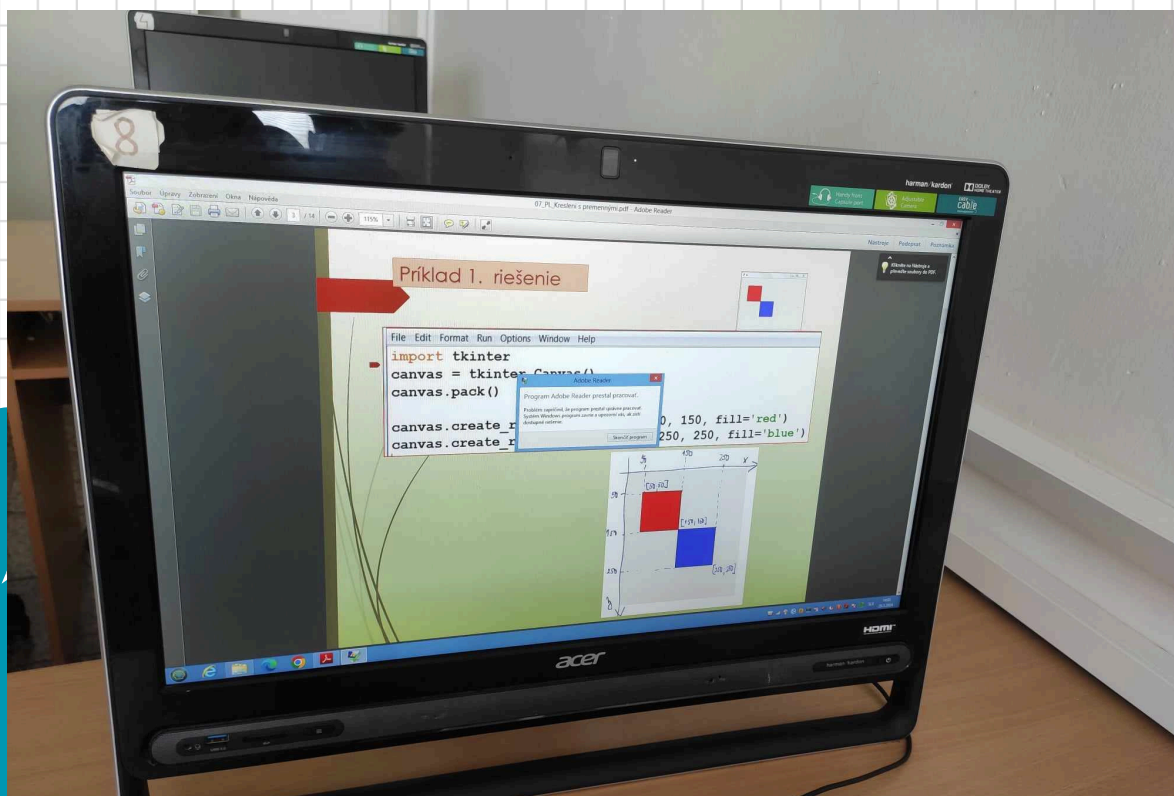
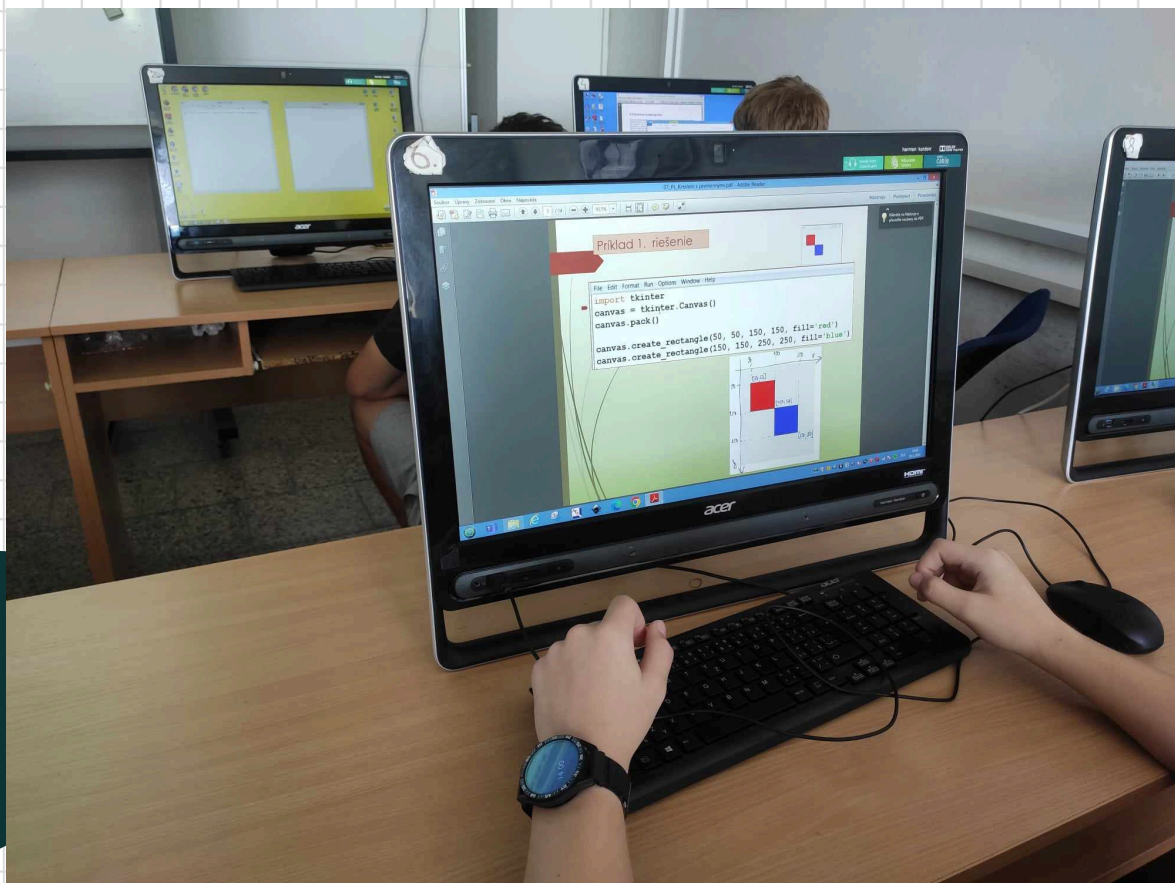
x = 200
y = 80
canvas.create_rectangle(x-50,y-50,x+50,y+50,fill='green')
```




PRÍLOHA



FOTODOKUMENTÁCIA VYUČOVACEJ HODINY

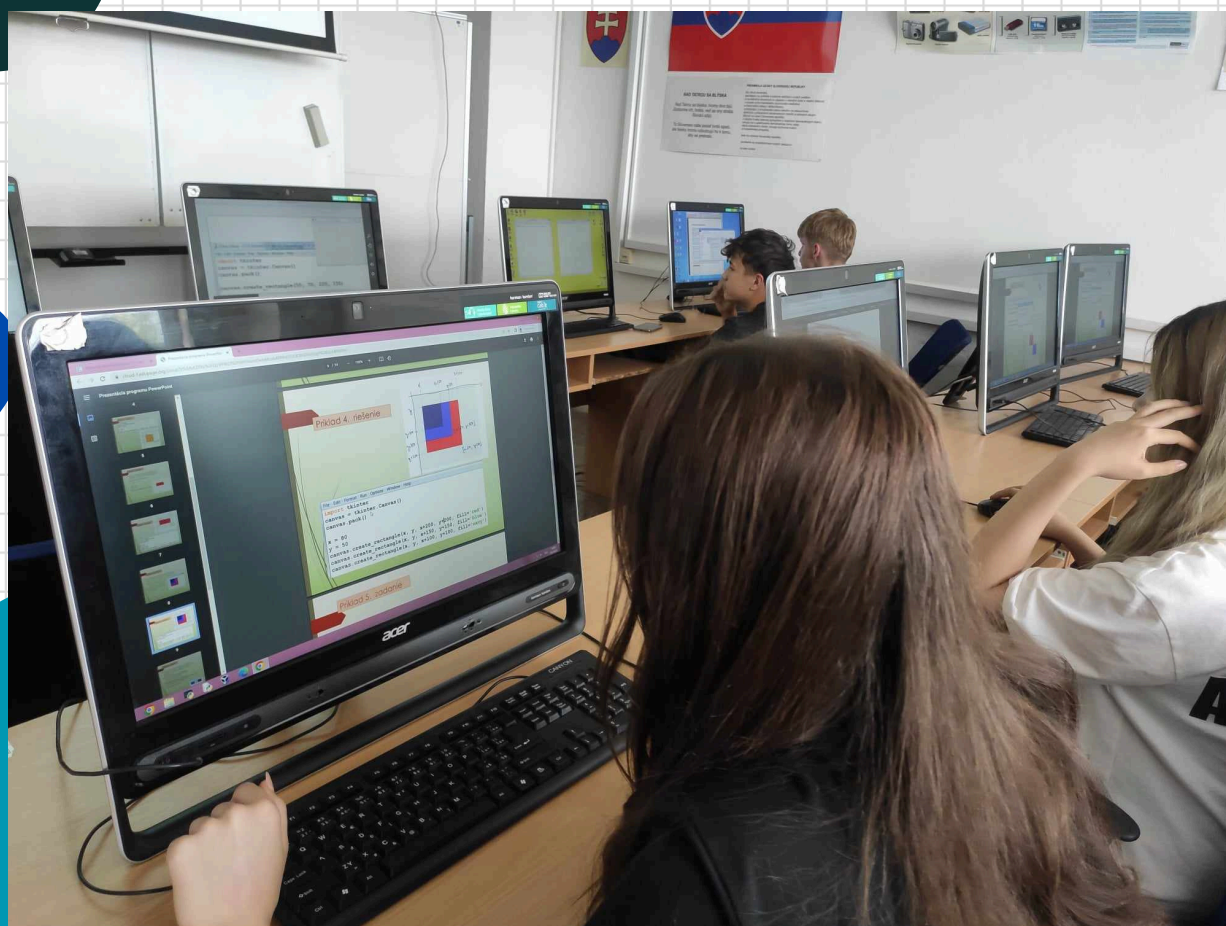
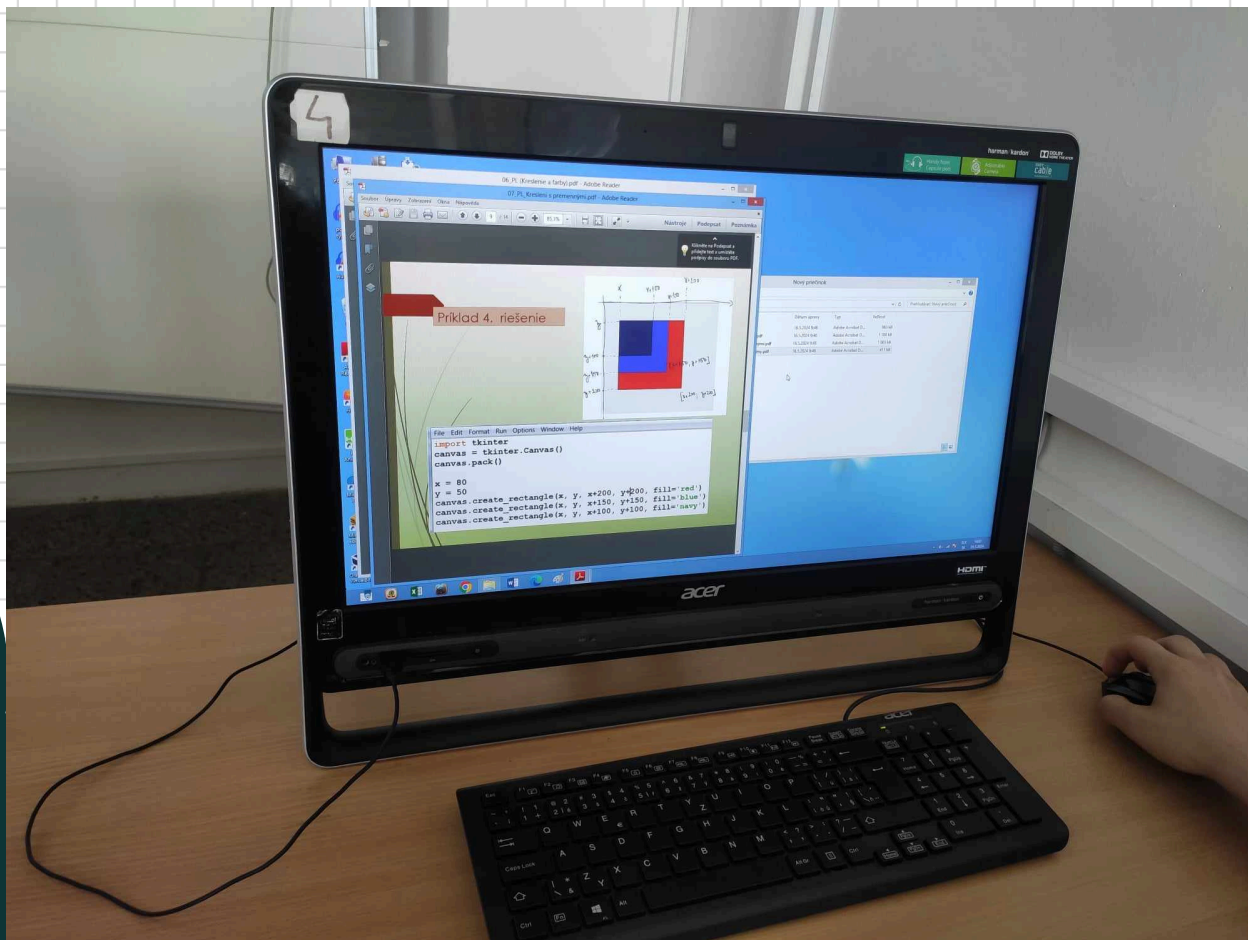




PRÍLOHA

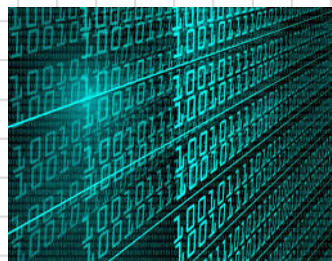


FOTODOKUMENTÁCIA VYUČOVACEJ HODINY





PRÍLOHA



PRÁCA ŽIAKOV:

NAPÍŠ PROGRAM, KTORÝ PREČÍTA TRI SLOVÁ A VYPÍŠE VŠETKÝCH 6 RÔZNYCH PERMUTÁCIÍ. PO SPUSTENÍ MÔŽEŠ DOSTAŤ:

ZADAJ 1. SLOVO: BIELA

ZADAJ 2. SLOVO: MODRÁ

ZADAJ 3. SLOVO: ČERVENÁ

BIELA MODRÁ ČERVENÁ

BIELA ČERVENÁ MODRÁ

MODRÁ BIELA ČERVENÁ

MODRÁ ČERVENÁ BIELA

ČERVENÁ BIELA MODRÁ

ČERVENÁ MODRÁ BIELA

ALEBO POUŽI PREMENNÉ:

ZADAJ 1. SLOVO: X

ZADAJ 2. SLOVO: ***

ZADAJ 3. SLOVO: :-)

X * :-)**

X :-) ***

***** X :-)**

***** :-) X**

:-) X ***

:-) * X**



PLÁN HODINY



PREDMET: PROGRAMOVANIE

TÉMA: PROGRAMOVANIE V PYTHONÉ – PODPROGRAMY.

• UČEBNÉ CIELE:.

- 1. POUŽÍVAŤ PODPROGRAMY NA SPREHL'ADNENIE PRÍKAZOV KRESLENIA V PYTHONÉ**
- 2. NAUČIŤ SA DEFINOVAŤ FUNKCIE PRI KRESLENÍ V PYTHONÉ.**

POMÔCKY:

**POČÍTAČ S NAINŠTALOVANÝM
PYTHONOM A KNIŽNICOU TKINTER.
VZOROVÁ UČEBNICA PYTHONU
PRACOVNÝ LIST S CVIČENIAMÍ**

**INTRO: POUŽÍVANIE RÔZNYCH FARIEB
PRI KRESLENÍ:**

- DEFINOVANIE FUNKCIÍ**
- POUŽITIE PODPROGRAMU PRE JEDNOTLIVÉ
OBJEKTY KRESLENIA**
- FUNKCIU UPRAVIŤ TAK, ABY SME JEJ MOHLI
PRI POUŽITÍ ZADAŤ MIESTO
OPAKOVAŤ PRÍKAZY V CYKLE**

- ŠTRUKTÚRA HODINY: ZAPOJENIE (4 MINÚTY) – PODPROGRAM JE MENŠÍ ÚSEK PROGRAMU, KTORÝ
POMENUJEME, A KEĎ HO CHCEME POUŽIŤ, LEN NAPÍŠEME V PROGRAME MENO TOHTO PODPROGRAMU.
ROZHOVOR SO ŽIAKMI O VÝHODÁCH POUŽÍVANIA PODPPROGRAMOV V PYTHONÉ: – VYUŽITIE
PODPROGRAMOV PRI KRESLENÍ KOMPLIKOVANEJŠÍCH OBRÁZKOV JE VHODNÉ PRETO, ABY SA PROGRAM
NESTAL NEPREHL'ADNÝM, PRETOŽE OBSAHUJE VEĽA PRÍKAZOV. SPREHL'ADNIŤ PROGRAM NÁM POMÁHA
POUŽÍVANIE KOMENTÁROV. LEPŠÍ SPÔSOB JE POUŽITIE PODPROGRAMU.**
- VÝSKUM (12 MINÚT) – SKÚMANIE NOVÝCH PRÍKAZOV – FUNKCIA JE POMENOVANÝ BLOK PRÍKAZOV
(NIEKEDY SA TOMU HOVORÍ AJ PODPROGRAM). POPISUJEME (DEFINUJEME) JU ŠPECIÁLNOU KONŠTRUKCIOU**
- VÝKLAD (15 MINÚT) – ZHRNUTIE A VYSVETLENIE POZNATKOV Z ČASTI SKÚMANIE PROBLEMATIKY
TVORBY PODPROGRAMOV V PYTHONÉ**
- VYPRACOVANIE (9 MINÚT) – PODPORA ŽIAKOV PRI ĎALŠOM SAMOSTATNOM VYTŮVÁRANÍ PODPROGRAMOV
PYTHONÉ**
- HODNOTENIE (5 MINÚT) – RIEŠENIE SEBAHODNOTIACEHO TESTU, DISKUSIA O ODPOVEDIACH.**

1.PRACOVNÝ LIST PRE ŽIAKA

**POSTUPOVAŤ BUDEME PODĽA NAŠEJ ELEKTRONICKEJ UČEBNICE PROGRAMOVANIE V PYTHONÉ,
KTORÁ JE URČENÁ PRE STREDNÉ ŠKOLY. VZOR UČEBNICE NÁJDETE NA:**

[HTTP://WWW.PROGRAMUJEMEVPYTHONE.SK/UKAZKA.PDF](http://www.programujemevpythone.sk/ukazka.pdf)

UKÁŽKA OBSAHUJE JEDNODUCHÉ PRÍKLADY VYTŮVÁRANIA PODPROGRAMOV V PYTHONÉ:

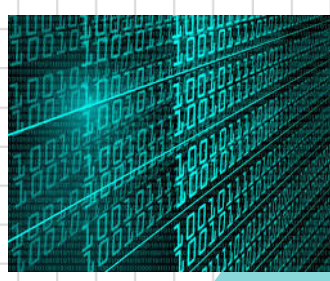
**1. VYTŮVORÍME VLASTNÉ PRÍKAZY, KTORÉ MÔŽEME POUŽÍVAŤ V NAŠOM PROGRAME. ODBORNE SA TAKÝMTO
PODPROGRAMOM V PYTHONÉ HOVORÍ FUNKCIE (V INÝCH PROGRAMOVACÍCH JAZYKOCH SA MÔŽETE
STRETNÚŤ AJ S NÁZVOM PROCEDÚRY ALEBO FUNKCIE, ALEBO OBOJE).**

FÁZA EXPOZÍCIE – ODSUNUTIE PRÍKAZOV

**OD L'AVÉHO OKRAJA POMÁHA PYTHONU POCHOPÍŤ, ČO VŠETKO PATRÍ K NAŠEJ FUNKCII. – NA TABUĽU SI
STRUČNE POZNAČÍM ZÁKLADNÉ POJMY**



PLÁN HODINY



4. PRACOVNÝ LIST PRE ŽIAKA

1.Úloha: Nakreslite dom a vytvorte k nemu aj funkciu (príkaz) dom.

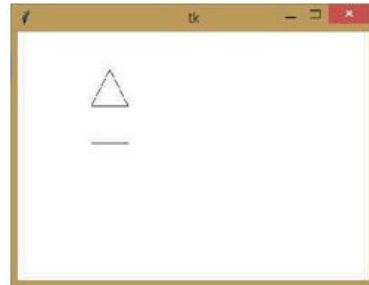
```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
canvas.create_rectangle(100, 100, 150, 150)
canvas.create_line(100, 100, 125, 50, 150, 100)
```

Program nakreslí z obdĺžnika a lomenej čiary dom so strechou. V ďalšej ukážke sme definovali našu vlastnú funkciu s názvom dom. Funkciu definujeme slovíčkom def, za ktorým nasleduje názov funkcie, potom zátvorky a na konci dvojbodka (rovnako ako vo for cykle).

```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
def dom():
    canvas.create_rectangle(100, 100, 150, 150)
    canvas.create_line(100, 100, 125, 50, 150, 100)
```

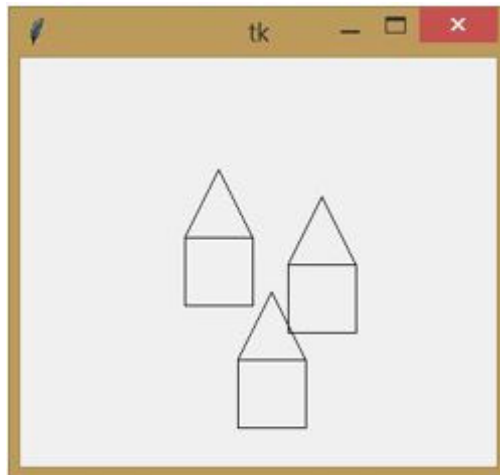
Ak chceme vykonať túto funkciu, napíšeme v programe dom() (na mieste programu, kde sa má funkcia vykonať).

```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
def dom():
    canvas.create_rectangle(100, 100, 150, 150)
    canvas.create_line(100, 100, 125, 50, 150, 100)
dom()
```



Tento program už aj spustí funkciu dom(). Význam definovania vlastného príkazu lepšie uvidíme, ak zmeníme funkciu dom tak, aby kreslila dom na náhodnom mieste.

```
import tkinter
from random import *
canvas = tkinter.Canvas()
canvas.pack()
def dom():
    x = randint(100, 300)
    y = randint(100, 300)
    canvas.create_rectangle(x, y, x+50, y+50)
    canvas.create_line(x, y, x+25, y-50, x+50, y)
dom()
dom()
dom()
```



V tomto programe už vykonáme funkciu dom() trikrát, a teda sa nám nakreslia tri domy na náhodnom mieste.



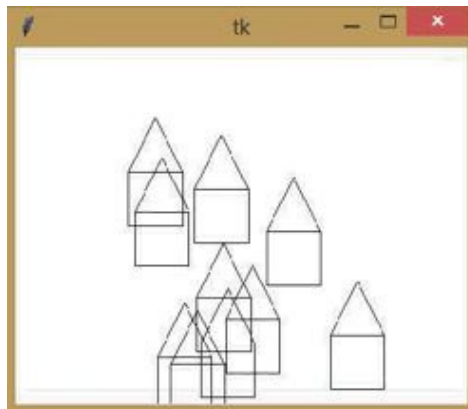
PLÁN HODINY



4. PRACOVNÝ LIST PRE ŽIAKA

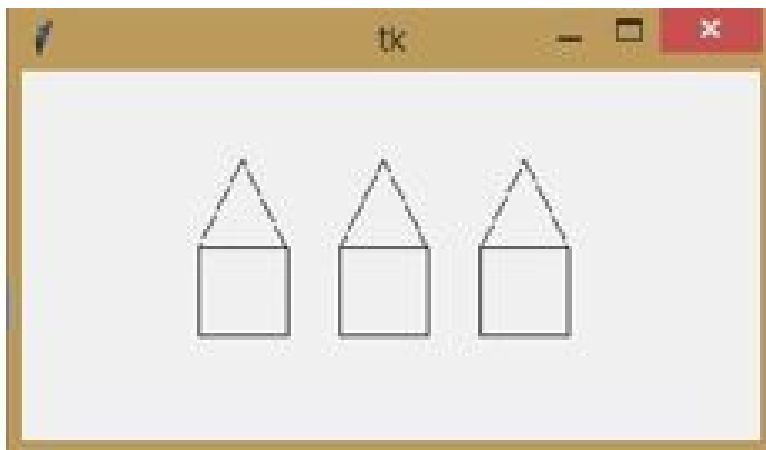
2.Úloha: Použite cyklus **for**, tak môžete ľahko nakresliť aj 10 domov.

```
import tkinter
from random import *
canvas = tkinter.Canvas()
canvas.pack()
def dom():
    x = randint(100, 300)
    y = randint(100, 300)
    canvas.create_rectangle(x, y, x+50, y+50)
    canvas.create_line(x, y, x+25, y-50, x+50, y)
    for i in range(1, 11):
        dom()
```

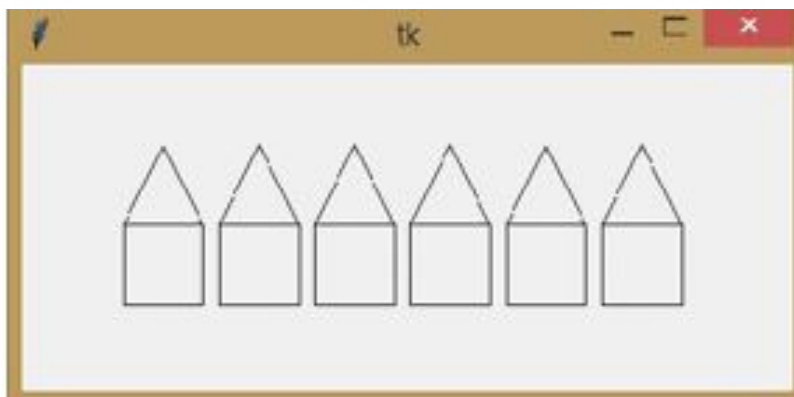


Funkcia sa dá upraviť aj tak, aby sme jej mohli pri použití zadať miesto, kde chceme nakresliť dom. Podobne ako pri používaní funkcie **create_rectangle** číslami v zátvorke ovplyvňujeme miesto nakreslenia. Takýmto číslom (alebo aj iným údajom), ktoré ovplyvňujú funkciu, hovoríme **parametre**. Pri definovaní funkcie s parametrom uvedieme do zátvorky názvy jednotlivých parametrov oddelené čiarkou (ak ich chceme mať viac ako jeden). Tieto názvy môžeme používať v našej funkcii (v podstate sú to akési **premenné**, ktoré sa dajú použiť len v tejto funkcii) a majú tú hodnotu, ktorú sme zadali pri použití funkcie.

```
import tkinter
from random import *
canvas = tkinter.Canvas()
canvas.pack()
def dom(x, y):
    canvas.create_rectangle(x, y, x+50, y+50)
    canvas.create_line(x, y, x+25, y-50, x+50, y)
    dom(100, 100)
    dom(180, 100)
    dom(260, 100)
```

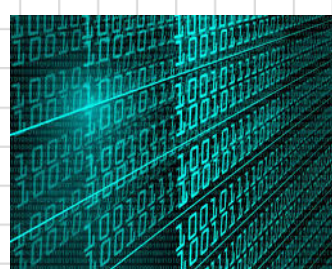


```
import tkinter
from random import *
canvas = tkinter.Canvas()
canvas.pack()
def dom(x, y):
    canvas.create_rectangle(x, y, x+50, y+50)
    canvas.create_line(x, y, x+25, y-50, x+50, y)
    for i in range(1, 7):
        dom(i*60, 100)
```

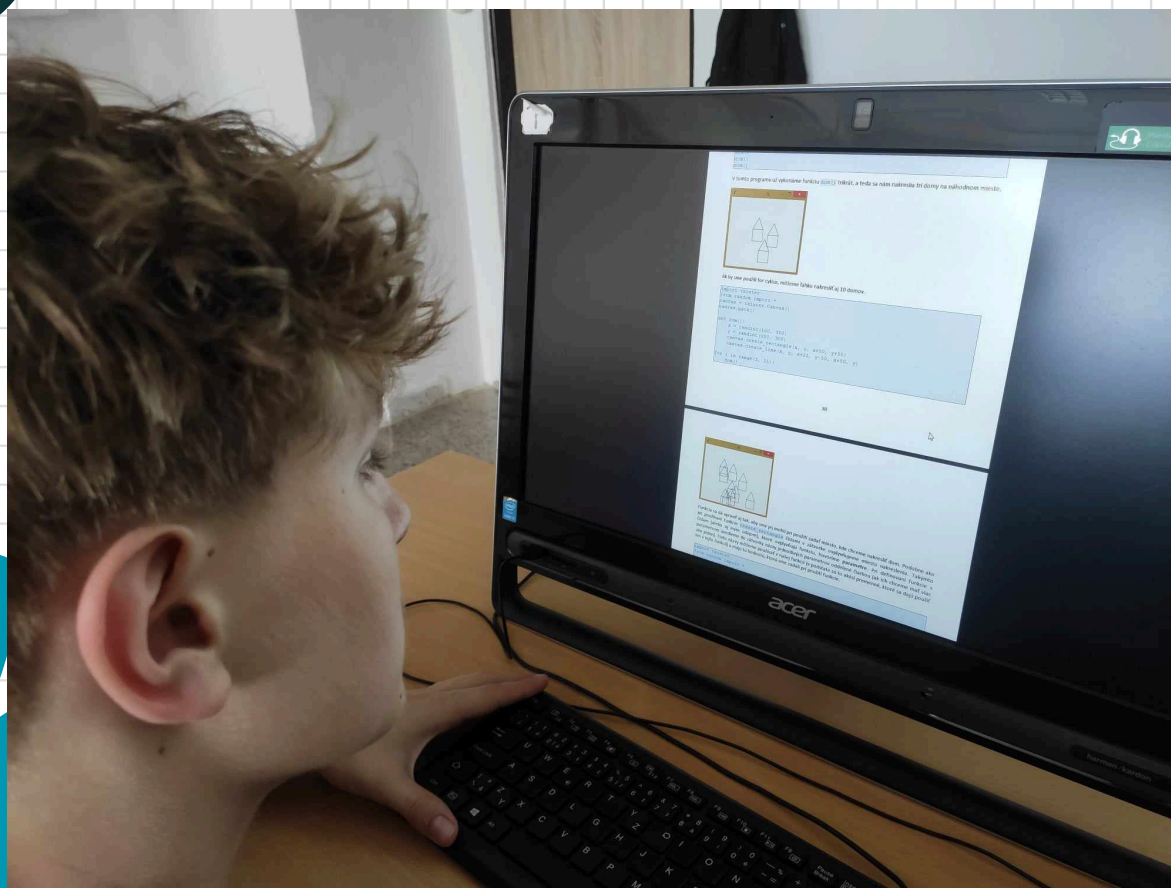
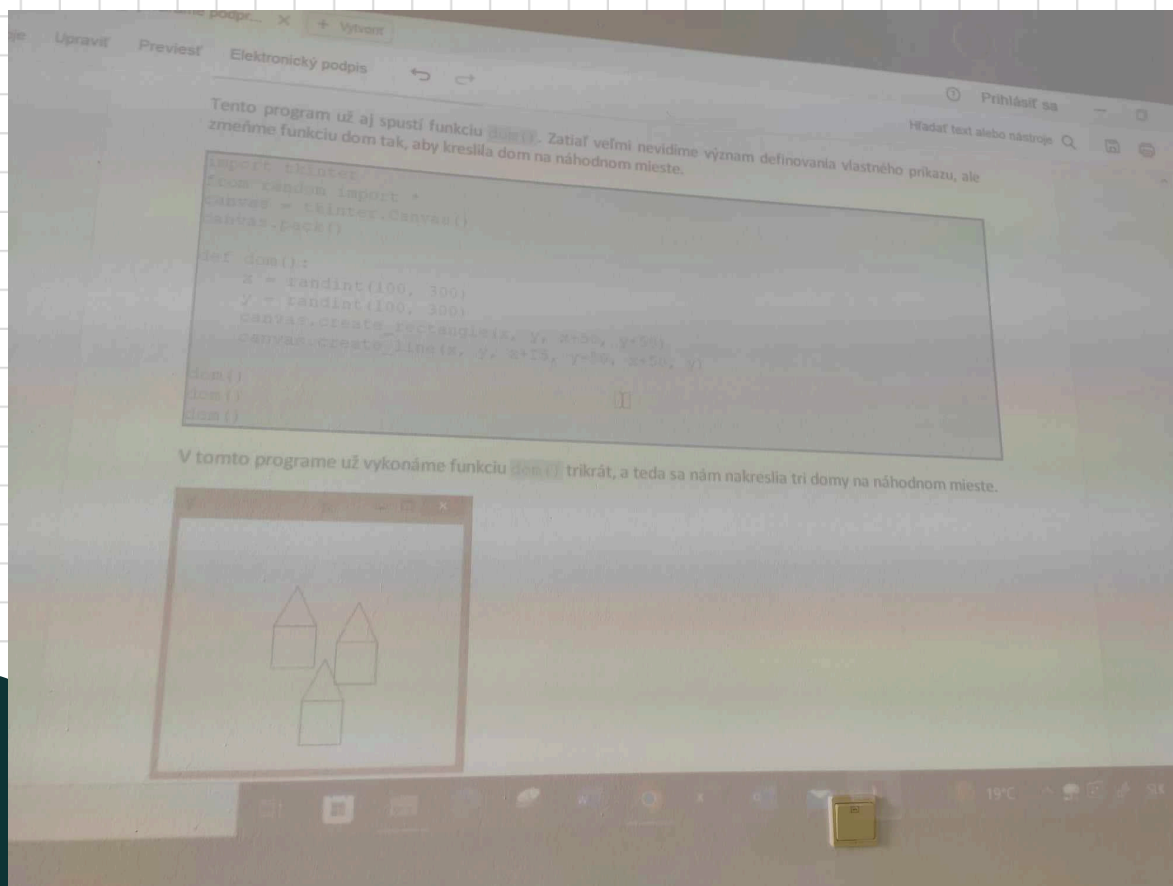




PRÍLOHA

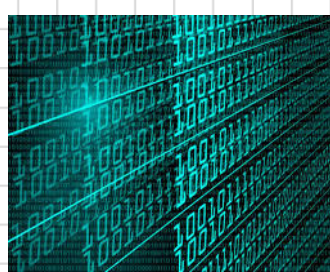


FOTODOKUMENTÁCIA VYUČOVACEJ HODINY

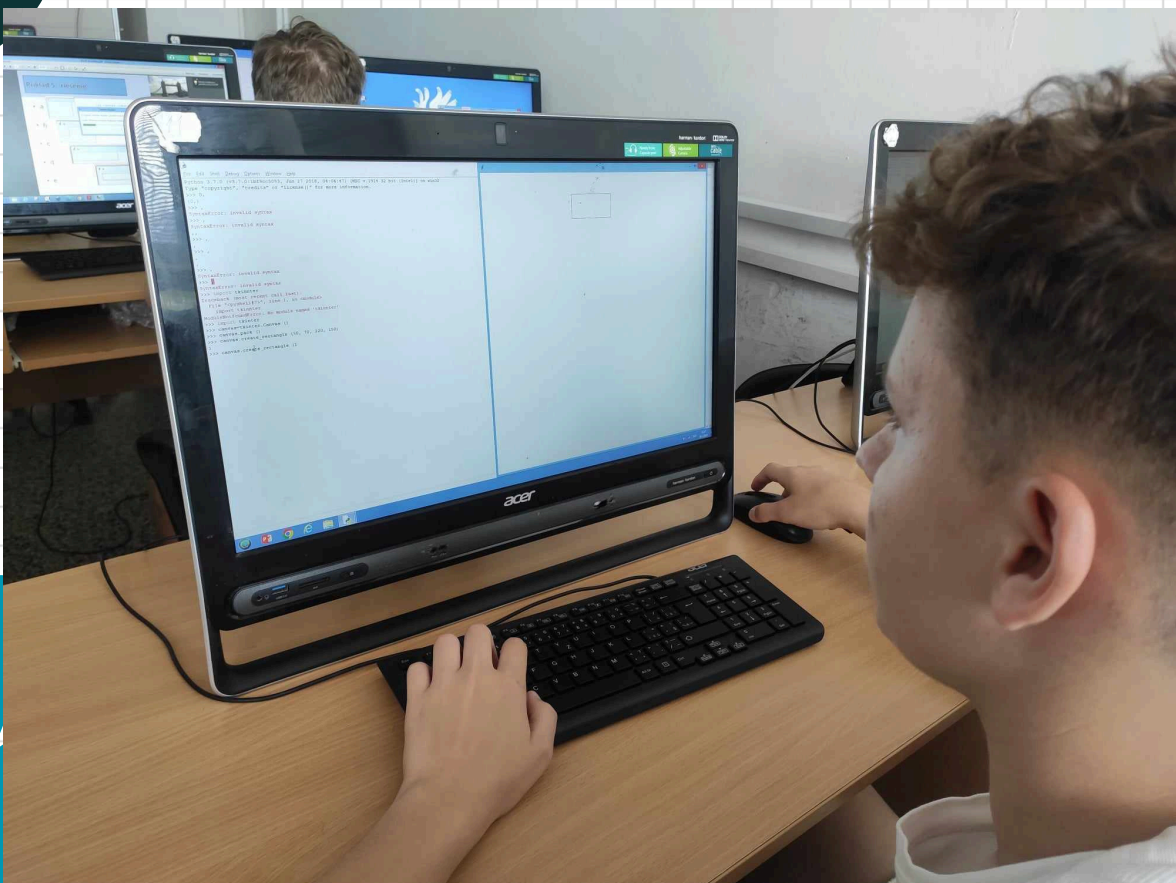
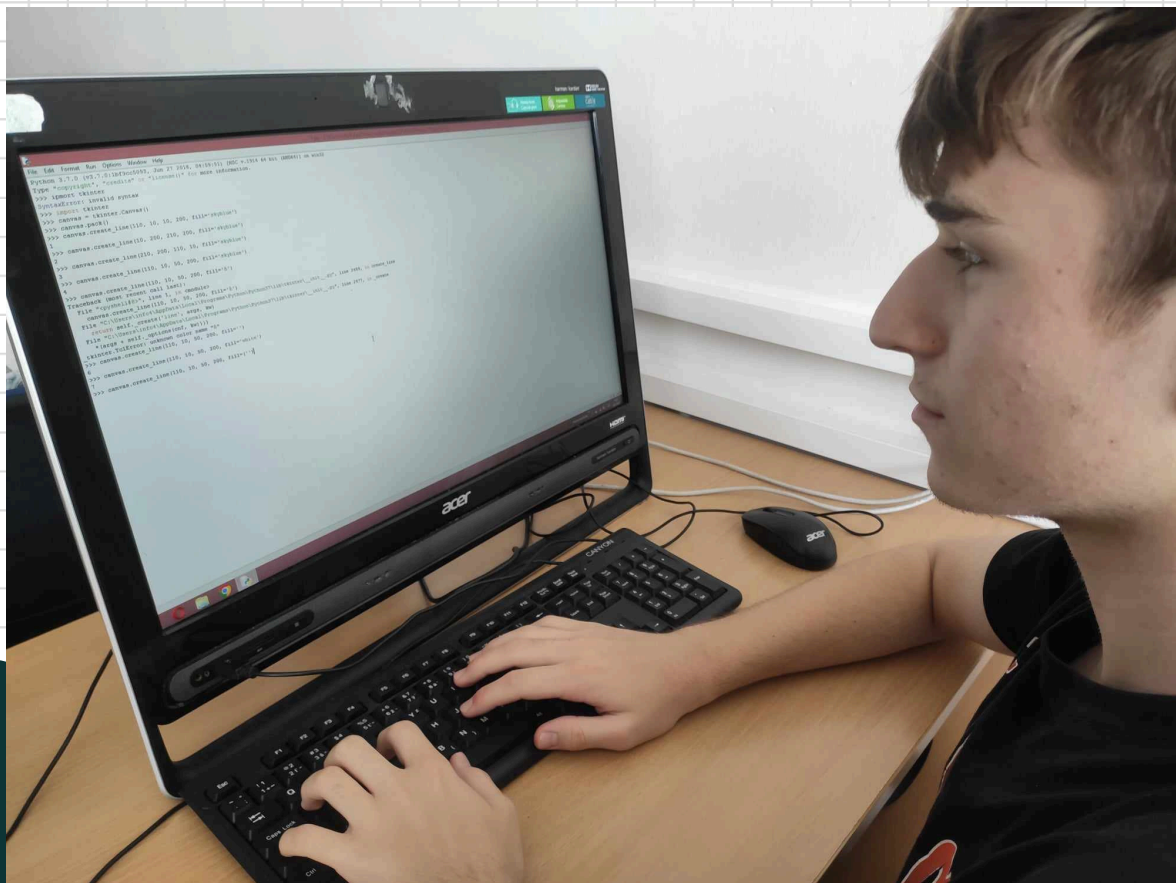




PRÍLOHA



FOTODOKUMENTÁCIA VYUČOVACEJ HODINY





PRÍLOHA



PRÁCA ŽIAKOV:

MECHANIZMUS VOLANIA JE NÁZORNÝ NA VOLANÍ VÝPIS_HVIEZDIČIEK(30):

- 1. ZAPAMÄTÁ SA NÁVRATOVÁ ADRESA VOLANIA**
 - 2. VYTVORÍ SA NOVÁ PREMENNÁ POČET (FORMÁLNY PARAMETER) A PRIRADÍ SA DO NEJ HODNOTA SKUTOČNÉHO PARAMETRA 30**
 - 3. VYKONAJÚ SA VŠETKY PRÍKAZY V DEFINÍCII FUNKCIE (TELO FUNKCIE)**
 - 4. ZRUŠIA SA VŠETKY PREMENNÉ, KTORÉ VZNIKLI POČAS BEHU FUNKCIE**
 - 5. RIADENIE SA VRÁTI NA MIESTO, KDE BOLO VOLANIE FUNKCIE**
- ZAPÍŠME NOVÚ FUNKCIU CERVENY_KRUH(), KTORÁ BUDE MAŤ DVA PARAMETRE: SÚRADNICE STREDU KRUHU. FUNKCIA NAKRESLÍ KRUH S POLOMEROM 10 A S DANÝM STREDOM:**

```
DEF CERVENY_KRUH(X, Y):
```

```
    CANVAS.CREATE_OVAL(X - 10, Y - 10, X + 10, Y + 10, FILL='RED')
```

A MÔŽEME JU ZAVOLAŤ NAPRÍKLAD TAKTO:

```
IMPORT TKINTER
```

```
DEF CERVENY_KRUH(X, Y):
```

```
    CANVAS.CREATE_OVAL(X - 10, Y - 10, X + 10, Y + 10, ILL='RED')
```

```
CANVAS = TKINTER.CANVAS()
```

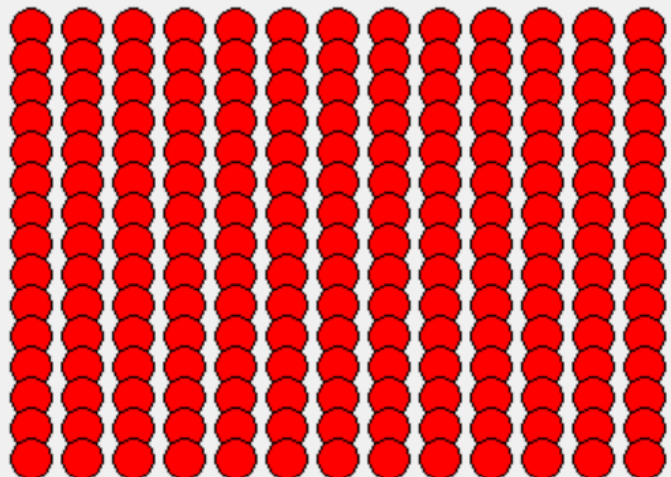
```
CANVAS.PACK()
```

```
FOR X IN RANGE(30, 350, 25):
```

```
    FOR Y IN RANGE(20, 240, 15):
```

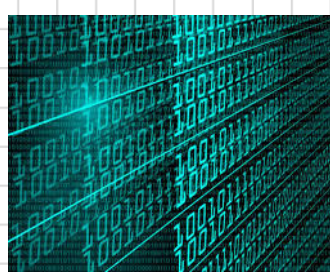
```
        CERVENY_KRUH(X, Y)
```

```
TKINTER.MAINLOOP()
```





PLÁN HODINY



PREDMET: PROGRAMOVANIE

TÉMA: PROGRAMOVANIE V PYTHONÉ – VYTVÁRAME PODPROGRAMY.

• UČEBNÉ CIELE:

- 1. POROZUMIEŤ ZÁKLADNÝM PRINCÍPOM VYTVÁRANIA PODPROGRAMOV V PYTHONÉ POMOCOU KNIŽNICE TKINTER**
- 2. NAUČIŤ SA POUŽÍVAŤ FUNKCIE S PARAMETRAMI V PYTHONÉ.**

POMÔCKY:

POČÍTAČ S NAINŠTALOVANÝM PYTHONOM A KNIŽNICOU TKINTER. VZOROVÁ UČEBNICA PYTHONU PRACOVNÝ LIST S CVIČENIAM

INTRO: POUŽÍVANIE RÔZNYCH FARIEB PRI KRESLENÍ:

- FUNKCIA JE POMENOVANÝ BLOK PRÍKAZOV (PODPROGRAM). POPISUJEME (DEFINUJEME) JU ŠPECIÁLNOU KONŠTRUKCIOU:**
- FUNKCIU MÔŽEME ZAVOLAŤ AŽ VTEDY, KEĎ UŽ PYTHON POZNÁ JEJ DEFINÍCIU**

- ŠTRUKTÚRA HODINY: ZAPOJENIE (4 MINÚTY)**
 - OSVOJENÉ SYNTAKTICKÉ PRAVIDLÁ**
SPÔSOB ZÁPISU DEFINÍCIE PODPROGRAMU: HLAVIČKA A TELO
ODSADENIE VŠETKÝCH PRÍKAZOV V TELE PODPROGRAMU OD ĽAVÉHO OKRAJA
SPÔSOB ZÁPISU VOLANIA PODPROGRAMU (PRÁZDNE ZÁTVORKY ZA NÁZVOM PODPROGRAMU)
 - ROZHOVOR SO ŽIAKMI O VÝHODÁCH POUŽÍVANIA PODPROGRAMOV V PYTHONÉ: – PODPROGRAMY MÔŽU POUŽÍVAŤ GLOBÁLNU PREMENNÚ CANVAS**
PRI FORMÁTOVANÍ ZÁPISU PODPROGRAMU POUŽÍVAME KLÁVES <TAB> A <ENTER>
 - VÝSKUM (12 MINÚT) – SKÚMANIE NOVÝCH PRÍKAZOV – POMOCOU PODPROGRAMOV SA KRESLÍ DO GRAFICKEJ PLOCHY. TERAZ BUDEŠ VYTVÁRAŤ PODPROGRAMY, KTORÉ KRESLIA DO GRAFICKÉHO OKNA.**
 - VÝKLAD (15 MINÚT) – ZHRNUTIE A VYSVETLENIE POZNATKOV Z ČASTI SKÚMANIE PROBLEMATIKY**
 - VYTVOR NOVÝ PROGRAM KRESBA_PODPROGRAM.PY A VYSKÚŠAJ:**
• IMPORT TKINTER
• CANVAS = TKINTER.CANVAS()
• CANVAS.PACK()
• DEF KRESLI():
• CANVAS.CREATE_RECTANGLE(10, 20, 30, 40, FILL='RED')
• KRESLI()
• V TEJTO ÚLOHE TELO PODPROGRAMU OBSAHUJE JEDINÝ PRÍKAZ – NAKRESLENIE OBDĹŽNIKA.
 - VYPRACOVANIE (9 MINÚT) – PODPORA ŽIAKOV PRI ĎALŠOM SAMOSTATNOM VYTVÁRANÍ PODPROGRAMOV PYTHONÉ**
 - HODNOTENIE (5 MINÚT) – RIEŠENIE SEBAHODNOTIACEHO TESTU, DISKUSIA O ODPOVEDIACH.**
- FÁZA EXPOZÍCIE – EXPOZÍCIA NOVÝCH PRÍKAZOV**
– NA TABUĽU SI STRUČNE POZNAČÍM ZÁKLADNÉ POJMY



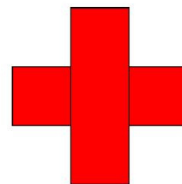
PLÁN HODINY



5. PRACOVNÝ LIST PRE ŽIAKA

1. Úloha: Vytvor nový program kriz.py av ňom definuj podprogram kriz, ktorý po zavolaní nakreslí červený kríž:

```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
def kriz():
    canvas.create_rectangle(150 - 90, 100 - 30, 150 + 90, 100 + 30, fill='red')
    canvas.create_rectangle(150 - 30, 100 - 90, 150 + 30, 100 + 90, fill='red')
    kriz()
```

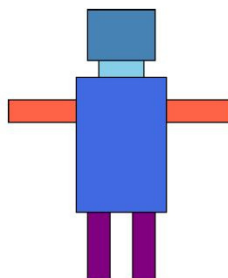


V tejto úlohe telo podprogramu obsahuje postupnosť dvoch príkazov. Pri každom volaní tohto podprogramu sa nakreslí najprv prvý obdĺžnik a cez neho druhý.

2. Úloha: Vytvor nový program robot.py, ktorý bude schopný nakresliť robota.

V programe budú štyri podprogramy – hlava, ruky, nohy, telo – a každý z nich bude schopný nakresliť časť robota. Keď ich zavolaš v nasledujúcom poradí:

```
hlava()
ruky()
nohy()
telo()
nakreslí sa celý robot.
```



```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
def hlava():
    canvas.create_rectangle(160, 40, 200, 100, fill='sky blue')
    canvas.create_rectangle(150, 10, 210, 55, fill='steel blue')
def telo():
    canvas.create_rectangle(140, 70, 220, 190, fill='royal blue')
def ruce():
    canvas.create_rectangle(80, 90, 280, 110, fill='tomato')
def nohy():
    canvas.create_rectangle(150, 160, 170, 250, fill='purple')
    canvas.create_rectangle(190, 160, 210, 250, fill='purple')
hlava()
ruce()
nohy()
telo()
```



PLÁN HODINY



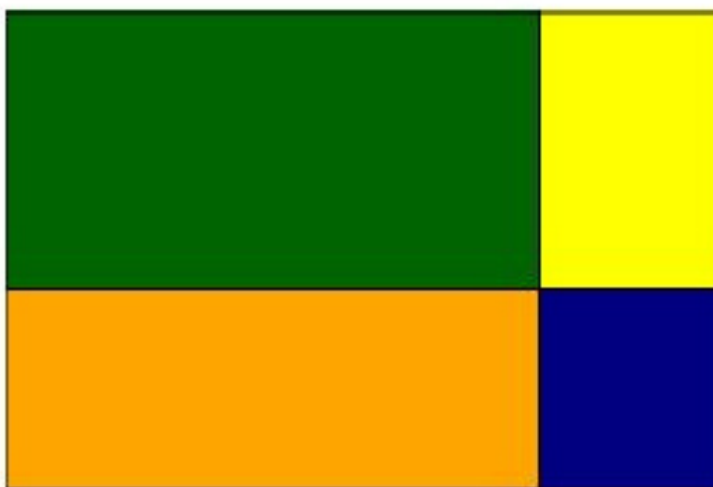
5. PRACOVNÝ LIST PRE ŽIAKA

3. Úloha: Vytvor nový program `sportová_vlajka.py`, ktorý bude kresliť športovú vlajku vašej triedy.

Vlajka bude tvorená štyrmi farebnými obdĺžnikmi, ktoré sa vzájomne dotýkajú v jednom bode ako na obrázku nižšie:

Program si náhodne zvolí súradnice x, y , ktoré predstavujú miesto dotyku všetkých štyroch obdĺžnikov.

Vonkajšie rozmery vlajky nech sú 300×200 ; farby na vlajke si určí podľa svojho uváženia.



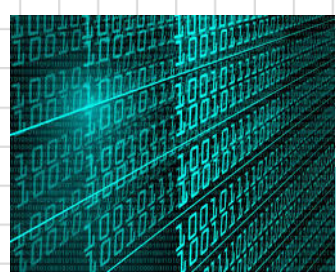
```
import tkinter
import random
canvas = tkinter.Canvas()
canvas.pack()
x = random.randint(50, 250)
y = random.randint(50, 150)
canvas.create_rectangle(10, 10, x, y, fill='darkgreen')
canvas.create_rectangle(x, 10, 310, y, fill='yellow')
canvas.create_rectangle(10, y, x, 210, fill='orange')
canvas.create_rectangle(x, y, 310, 210, fill='navy')
```

Vonkajšie vrcholy obdĺžnika je možné zvoliť ľubovoľne. V našom riešení má ľavý horný vrchol súradnice $[10, 10]$ a pretože rozmery obdĺžnika majú byť 300×200 , protiľahlý vrchol (pravý dolný) má súradnice $[310, 210]$.

Vnútroštný náhodne zvolený bod sme vygenerovali tak, aby bol od okrajov obdĺžnika vzdialený aspoň 50, teda každá zo štyroch oblastí bude mať rozmery minimálne 50×50 .



PRÍLOHA



FOTODOKUMENTÁCIA VYUČOVACEJ HODINY

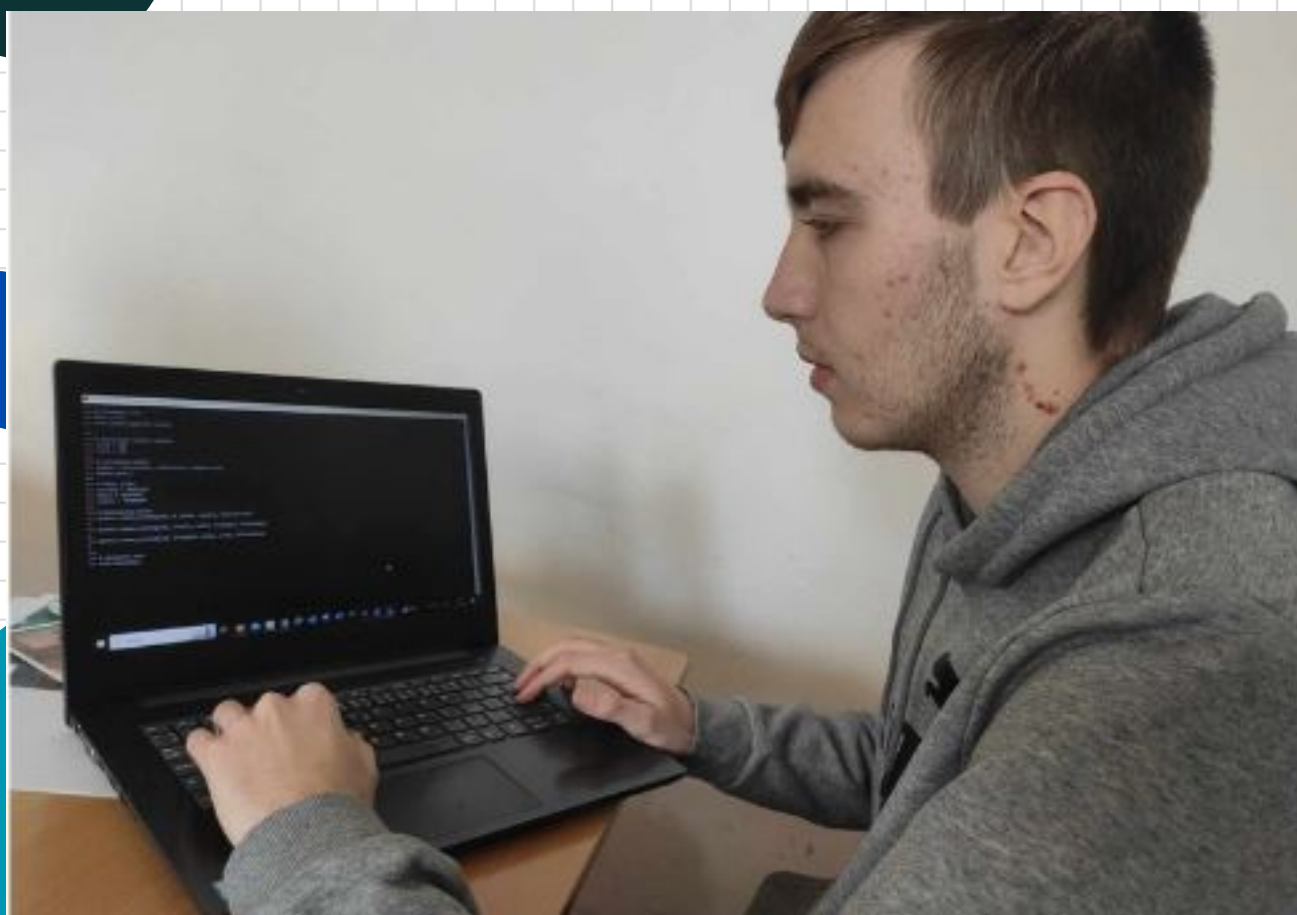




PRÍLOHA

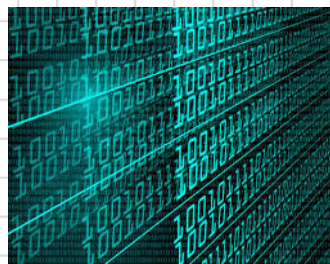


FOTODOKUMENTÁCIA VYUČOVACEJ HODINY





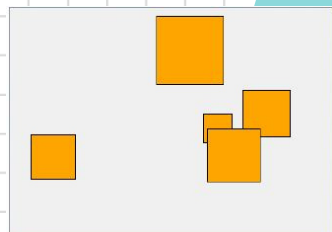
PRÍLOHA



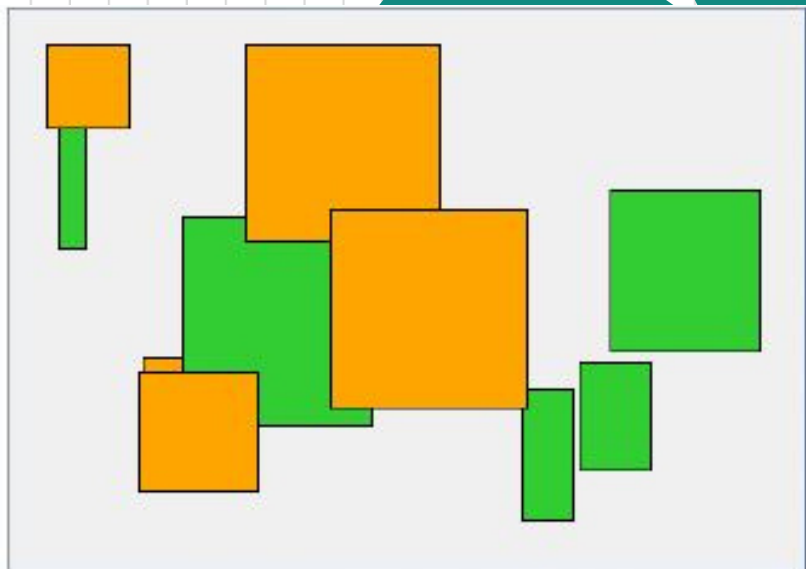
PRÁCA ŽIAKOV:

UPRAV PROGRAM NAHODNY_ŠTVOREC.PY TAK, ABY SA ŠTVORCE KRESLILI NIELEN NA NÁHODNÝCH POZÍCIÁCH, ALE TIEŽ ABY MAL KAŽDÝ ŠTVOREC NÁHODNÚ VEĽKOSŤ Z INTERVALU OD 10 DO 100:

```
DEF NAHODNY_ŠTVOREC():  
X = RANDOM.RANDINT(10, 300)  
Y = RANDOM.RANDINT(10, 200)  
A = RANDOM.RANDINT(10, 100)  
CANVAS.CREATE_RECTANGLE(X, Y, X + A, Y + A, FILL='ORANGE')
```



```
IMPORT TKINTER  
IMPORT RANDOM  
CANVAS = TKINTER.CANVAS()  
CANVAS.PACK()  
DEF NAHODNY_ŠTVOREC():  
X = RANDOM.RANDINT(10, 300)  
Y = RANDOM.RANDINT(10, 200)  
A = RANDOM.RANDINT(10, 100)  
CANVAS.CREATE_RECTANGLE(X, Y, X + A, Y + A, FILL='ORANGE')  
DEF NAHODNY_OBDĽŽNIK():  
X = RANDOM.RANDINT(10, 300)  
Y = RANDOM.RANDINT(10, 200)  
A = RANDOM.RANDINT(10, 100)  
B = RANDOM.RANDINT(10, 100)  
CANVAS.CREATE_RECTANGLE(X, Y, X+A, Y+B, FILL='LIME GREEN')  
NAHODNY_ŠTVOREC()  
NAHODNY_OBDĽŽNIK()  
NAHODNY_ŠTVOREC()  
NAHODNY_OBDĽŽNIK()  
NAHODNY_ŠTVOREC()  
NAHODNY_OBDĽŽNIK()  
NAHODNY_ŠTVOREC()  
NAHODNY_OBDĽŽNIK()  
NAHODNY_ŠTVOREC()  
NAHODNY_OBDĽŽNIK()
```



Plán hodiny

Séria úvodných lekcí o programovaní v jazyku C

Téma: Začínáme s C

Lekcia č. 1

Zameranie a ciele lekcie:

Táto lekcia predstaví študentom programovací jazyk C, jeho históriu a základnú syntax. Na konci tejto lekcie by študenti mali byť schopní napísať a spustiť jednoduchý "Ahoj, svet!" program

Potrebné materiály:

Počítače s nainštalovaným kompilátorom C (napr. GCC)
Projektor a plátno
Tabuľa a fixky
Príručky základnej syntaxe a štruktúry jazyka C

Ciele učenia:

Predstaviť C
Predstavte Poznámkový blok a GCC Napíšte
„Ahoj, svet!“ program.

Štruktúra/činnosť:

Úvod do C (15 minút)

Diskusia (5 minút): Začnite stručnou históriou programovacieho jazyka C. Diskutujte o jeho význame a o tom, ako položil základy mnohých moderných programovacích jazykov. Zvýraznite kľúčové aplikácie, ako sú operačné systémy (napr. UNIX, Linux) a vstavané systémy.

Prezentácia (10 minút): Pomocou projektora zobrazte, kde sa C dnes používa, a diskutujte o tom, prečo je učenie C nevyhnutné. Uveďte príklady populárneho softvéru a systémov napísaných v jazyku C.

Základná syntax a štruktúra (20 minút)

Vysvetlenie (10 minút): Predstavte štruktúru základného programu v jazyku C. Vysvetlite úlohu #include, funkciu main() a základné pravidlá syntaxe. Uveďte príklady syntaktických chýb a ako sa im vyhnúť.

Príklad (10 minút): Napíšte jednoduché "Ahoj, svet!" program na tabuľu alebo plátno projektora. Rozdeľte každý riadok kódu a vysvetlite jeho účel. Povzbudzujte študentov, aby kládli otázky o každej časti kódu.

Praktická aktivita (20 minút)

Aktivita (15 minút): Nechajte študentov, aby si sami napísali "Ahoj, svet!" program na svojich počítačoch. Preved'te ich nastavením ich vývojového prostredia (ak je to potrebné, nainštalujte kompilátor C, ako je GCC). Uistite sa, že každý študent môže úspešne zostaviť a spustiť svoj program.

Peer Review (5 minút): Spárujte študentov, aby si navzájom prezreli svoj kód. Povzbud'te ich, aby diskutovali o akýchkoľvek problémoch alebo otázkach, s ktorými sa stretli.

Otázky a odpovede a rekapitulácia (5 minút)

Otázky a odpovede (3 minúty): Otvorte priestor pre akékoľvek otázky týkajúce sa lekcie. Riešte bežné problémy alebo nejasnosti, ktoré môžu študenti mať.

Rekapitulácia a domáca úloha (2 minúty): Zhrňte kľúčové body lekcie. Pre domácu úlohu požiadajte študentov, aby online preskúmali jednoduchý program v jazyku C, mierne ho upravili (napr. zmenili výstupnú správu) a spustili ho.

Plán hodiny

Séria úvodných lekcí o programovaní v jazyku C

Téma: Začíname s C

Pracovný list č. 1

1. Úvod do C

Kľúčové body:

C je základný programovací jazyk používaný v mnohých aplikáciách vrátane operačných systémov a vstavaných systémov.

Pochopenie C pomáha pri učení sa iných programovacích jazykov, ako sú C++, Java a Python.

Príklady programov:

„Ahoj, svet!": Najjednoduchší program v jazyku C.

```
#include <stdio.h>

int main() { printf("Ahoj svet!\n");
            návrat 0;
}
```

2. Základná syntax a štruktúra

Kľúčové body:

Každý program v jazyku C musí obsahovať `#include <stdio.h>` a funkciu `main()`.

Funkcia `main()` je východiskovým bodom každého programu v jazyku C.

Aktivita:

Napíšte a spustite „Ahoj, svet!“ program.

Identifikujte účel každého riadku v programe.

3. Domáca úloha:

Preskúmajte jednoduchý program C online, mierne ho upravte (napr. zmeňte výstupnú správu) a spustite ho.

Plán hodiny

Séria úvodných lekcí o programovaní v jazyku C

Téma: Premenné a dátové typy

Lekcia č. 2

Zameranie a ciele lekcie:

Naučte študentov o premenných, dátových typoch a základných vstupných/výstupných operáciách v C.

Potrebné materiály:

Počítače s nainštalovaným kompilátorom C (napr. GCC)
Projektor a plátno
Tabuľa a fixky
Príručky s príkladmi deklarácií premenných a bežných vstupných/výstupných operácií

Ciele učenia:

Získajte informácie o premenných/typoch údajov
Napíšte program s printf

Štruktúra/činnosť:

Recenzia predchádzajúcej lekcie (5 minút)

Diskusia (5 minút): Zopakujte si základnú štruktúru programu v jazyku C a "Ahoj, svet!" program z predchádzajúcej hodiny. Zdôraznite dôležitosť pochopenia syntaxe pred prechodom na zložitejšie koncepty.

Úvod do premenných a dátových typov (15 minút)

Vysvetlenie (10 minút): Predstavte koncept premenných ako zástupných symbolov pre údaje. Vysvetlite rozdiel medzi rôznymi typmi údajov (napr. int, float, char, double) a ich typické použitie.

Príklad (5 minút): Napíšte jednoduchý program, ktorý deklaruje rôzne typy premenných a použije printf na výstup ich hodnôt. Ukážte, ako priradovať hodnoty premenným a vykonávať základné aritmetické operácie.

Praktická aktivita č. 1: Variabilné vyhlásenie (15 minút)

Aktivita (10 minút): Nechajte študentov deklarovať premenné rôznych typov údajov v novom programe.

Požiadajte ich, aby vykonali základné operácie, ako je sčítanie a násobenie, a vytlačili výsledky.

Peer Review (5 minút): Študenti si vymenia svoje programy a skontrolujú, či sú deklarácie a operácie premenných správne. Povzbudzujte ich, aby diskutovali o akýchkoľvek rozdieloch alebo chybách.

Úvod do základného vstupu/výstupu (10 minút)

Vysvetlenie (5 minút): Zavedenie funkcie scanf na preberanie užívateľského vstupu. Vysvetlite, ako funguje v spojení s printf vytváranie interaktívnych programov.

Príklad (5 minút): Napíšte program, ktorý sa opýta používateľa na jeho vek a potom mu ho pomocou scanf a printf vytlačí späť.

Praktická aktivita č. 2: Interaktívny program (15 minút)

Aktivita (10 minút): Študenti upravia svoj predchádzajúci program tak, aby obsahoval vstup používateľa.

Napríklad požiadajte používateľa o dve čísla, vykonajte s nimi operáciu (napríklad sčítanie) a vytlačte výsledok.

Podpora (5 minút): Rozobehnite miestnosť, aby ste pomohli študentom s akýmkoľvek syntaktickými alebo logickými chybami, s ktorými sa stretnú.

Otázky a odpovede a rekapitulácia (5 minút)

Otázky a odpovede (3 minúty): Umožnite študentom klásť otázky o premenných, typoch údajov alebo vstupných/výstupných operáciách.

Rekapitulácia a domáca úloha (2 minúty): Zhrňte lekciu. Zadať domácu úlohu, v ktorej študenti vytvoria program, ktorý prevezme vstup od používateľa pre tri rôzne typy údajov a vygeneruje ich vo formátovanom reťazci.

Plán hodiny

Séria úvodných lekcí o programovaní v jazyku C

Téma: Premenné a dátové typy

Pracovný list č. 2

1. Úvod do premenných

Kľúčové body:

Premenné sú zástupné symboly pre údaje. V C má každá premenná špecifický dátový typ.

Bežné dátové typy: int (celé číslo), float (číslo s pohyblivou rádovou čiarkou), char (znak), double (s pohyblivou rádovou čiarkou s dvojitou presnosťou).

Príklad kódu:

```
int vek = 25;  
výška plaváka = 5,9;  
počiatočný znak = 'A';  
dvojnásobná vzdialenosť =  
12345,67;
```

2. Vstup a výstup

Kľúčové body:

Použite printf na výstup dát a scanf na prijatie vstupu.

Príklad kódu:

```
int vek;  
printf("Zadajte svoj vek: ");  
scanf("%d", &vek);  
printf("Zadali ste: %d\n", vek);
```

3. Domáca úloha:

Vytvorte program, ktorý prevezme vstup od používateľa pre tri rôzne typy údajov a odošle ich vo formátovanom reťazci.

Plán hodiny

Séria úvodných lekcí o programovaní v jazyku C

Téma: Riadiace štruktúry

Lekcia č. 3

Zameranie a ciele lekcie:

Táto lekcia predstaví študentom riadiace štruktúry v C, vrátane príkazov if-else a slučiek. Na konci hodiny by študenti mali byť schopní používať tieto štruktúry na riadenie toku svojich programov.

Potrebné materiály:

Počítače s nainštalovaným kompilátorom C (napr. GCC)
Projektor a plátno
Tabuľa a fixky
Materiály s príkladmi príkazov a slučiek if-else

Ciele učenia:

Získajte informácie o základnom vstupe/výstupe
Získajte informácie o príkazoch if/if-else

Štruktúra/činnosť:

Recenzia predchádzajúcej lekcie (5 minút)

Diskusia (5 minút): Zopakujte si pojmy premenných, dátových typov a základného vstupu/výstupu. Zdôraznite, ako budú tieto prvky použité v spojení s riadiacimi štruktúrami.

Úvod do vyhlásení If-Else (15 minút)

Vysvetlenie (10 minút): Vysvetlite syntax a použitie príkazov if-else. Diskutujte o koncepte podmienenej logiky a o tom, ako umožňuje programom robiť rozhodnutia.

Príklad (5 minút): Napíšte program, ktorý sa opýta používateľa na číslo a vypíše, či je párne alebo nepárne.

Vysvetlite, ako sa používa príkaz if-else na kontrolu podmienok.

Praktická aktivita č. 1: Podmienená logika (15 minút)

Aktivita (10 minút): Študenti napíšu program, ktorý berie vstup od používateľa a používa príkazy if-else na rozhodovanie (napr. kontroluje, či je číslo kladné, záporné alebo nulové).

Peer Review (5 minút): Nechajte študentov vymeniť si svoje programy a zopakovať si použitú logiku. Povzbudzujte ich, aby si navzájom otestovali svoje programy s rôznymi vstupmi.

Úvod do slučiek (15 minút)

Vysvetlenie (10 minút): Zavedenie cyklov for a while. Vysvetlite, ako sa slučky používajú na opakovanie akcií v programe a diskutujte o scenároch, kde je každý typ slučky vhodný.

Príklad (5 minút): Napíšte program, ktorý vypíše čísla od 1 do 10 pomocou cyklu for. Potom ho prepíšte pomocou slučky while.

Praktická aktivita č. 2: Looping Structures (15 minút)

Aktivita (10 minút): Žiaci napíšu program, ktorý pomocou cyklu vypočíta faktoriál čísla alebo vygeneruje tabuľku násobenia.

Podpora (5 minút): Pomôžte študentom s ladením ich slučiek a zabezpečením správnosti logiky.

Otázky a odpovede a rekapitulácia (5 minút)

Otázky a odpovede (3 minúty): Odpovedzte na akékoľvek otázky týkajúce sa riadiacich štruktúr. Diskutujte o bežných chybách, ako sú nekonečné slučky alebo nesprávne použitie if-else.

Rekapitulácia a domáca úloha (2 minúty): Zhrňte lekcii. Zadaajte domácu úlohu, v ktorej študenti napíšu program pomocou príkazov if-else a slučiek na vytvorenie jednoduchej kalkulačky.

Plán hodiny

Séria úvodných lekcí o programovaní v jazyku C

Téma: Riadiace štruktúry

Pracovný list č. 3

1. Vyhlásenia If-Else

Kľúčové body:

príkazy if-else umožňujú programu rozhodovať sa na základe podmienok.

Príklad kódu:

```
int číslo;  
printf("Zadajte číslo: ");  
scanf("%d", &číslo);  
  
if (číslo % 2 == 0) { printf("Číslo je párne.\n");  
} else { printf("Číslo je nepárne.\n");  
}
```

2. Slučky

Kľúčové body:

Slučky umožňujú programu opakovať akcie. Bežné slučky sú for a while.

Príklad kódu:

```
for (int i = 1; i <= 10; i++) {  
    printf("%d\n", i);  
}
```

3. Domáca úloha:

Napíšte program pomocou príkazov if-else a cyklov na vytvorenie jednoduchkej kalkulačky.

Text vášho odseku

Plán hodiny

Séria úvodných lekcí o programovaní v jazyku C

Téma: Funkcie

Lekcia č. 4

Zameranie a ciele lekcie:

Táto lekcia učí študentov o funkciách v C, ich význame a ako ich vytvárať a používať. Na konci hodiny by študenti mali byť schopní definovať a volať základné funkcie vo svojich programoch.

Potrebné materiály:

Počítače s nainštalovaným kompilátorom C (napr. GCC)
Projektor a plátno
Tabuľa a fixky

- Materiály s príkladmi definícií funkcií a volaní funkcií

Ciele učenia:

Naučte sa definovať základné funkcie v C
Hands-on praxi

Štruktúra/činnosť:

Recenzia predchádzajúcej lekcie (5 minút)

Diskusia (5 minút): Rýchlo si prečítajte riadiace štruktúry a zdôraznite, ako funkcie pomôžu organizovať a opätovne použiť kód v zložitejších programoch.

Úvod do funkcií (15 minút)

Vysvetlenie (10 minút): Definujte funkcie, vysvetlite účel rozdelenia kódu do menších, opakovane použiteľných blokov. Diskutujte o syntaxi funkcie vrátane návratových typov, názvov funkcií a parametrov.

Príklad (5 minút): Napíšte jednoduchú funkciu, ktorá vypočíta druhú mocninu čísla. Vysvetlite, ako je funkcia definovaná, volaná a ako vracia hodnotu.

Praktická aktivita č. 1: Funkcie písania (15 minút)

Aktivita (10 minút): Nechajte študentov napísať svoju vlastnú funkciu, ktorá vykonáva konkrétnu úlohu, ako je napríklad prevod stupňov Celzia na stupne Fahrenheita. Uistite sa, že rozumejú tomu, ako odovzdávať parametre a návratové hodnoty.

Peer Review (5 minút): Študenti si vymieňajú svoje programy a navzájom si testujú svoje funkcie.

Rozšírenie funkcií (15 minút)

Vysvetlenie (10 minút): Diskutujte o tom, ako používať viacero funkcií v jednom programe. Vysvetlite, ako môžu funkcie volať iné funkcie a ako spravovať závislosti funkcií.

Príklad (5 minút): Napíšte program, ktorý obsahuje dve funkcie: jednu na overenie vstupu a druhú na vykonanie výpočtu. Ukážte, ako tieto funkcie interagujú.

Praktická aktivita č. 2: Interakcia funkcií (15 minút)

Aktivita (10 minút): Študenti vytvoria program, ktorý využíva najmenej dve funkcie, ktoré spolupracujú, ako je napríklad jednoduchá kalkulačka, ktorá overuje zadanie pred vykonaním operácií.

Podpora (5 minút): Poskytnite návod na ladenie a pochopenie rozsahu funkcií.

Otázky a odpovede a rekapitulácia (5 minút)

Otázky a odpovede (3 minúty): Odpovedzte na akékoľvek otázky týkajúce sa funkcií. Diskutujte o bežných problémoch, ako je preťaženie funkcií a konflikty názvov.

Rekapitulácia a domáca úloha (2 minúty): Zhrňte lekcii. Pre domácu úlohu požiadajte študentov, aby upravili program tak, aby obsahoval funkcie pre vstup, spracovanie a výstup.

Plán hodiny

Séria úvodných lekcí o programovaní v jazyku C

Téma: Funkcie

Materiál č. 4

1. Úvod do funkcií

Kľúčové body:

Funkcie sú opakovane použiteľné bloky kódu, ktoré vykonávajú špecifické úlohy.

Funkcie majú návratový typ, názov a parametre.

Príklad kódu:

```
int square(int num) { return num * num;
}

int main() { int vysledok = square(5);
    printf("Štvorec 5 je %d\n", vysledok);
    návrat 0;
}
```

2. Funkcie na písanie a volanie

Aktivita:

Napíšte funkciu, ktorá prevedie stupne Celzia na stupne Fahrenheita a zavolá ju v `main()`.

3. Domáca úloha:

Upravte program tak, aby obsahoval funkcie pre vstup, spracovanie a výstup.

Plán hodiny

Séria úvodných lekcí o programovaní v jazyku C

Téma: Polia a reťazce

Lekcia č. 5

Zameranie a ciele lekcie:

Táto lekcia predstaví študentom polia a reťazce v C a vysvetlí im, ako ich deklarovať, inicializovať a používať v programoch. Na konci hodiny by študenti mali byť schopní manipulovať s poľami a reťazcami v základných programoch v jazyku C.

Potrebné materiály:

Počítače s nainštalovaným kompilátorom C (napr. GCC)
Projektor a plátno
Tabuľa a fixky

- Materiály s príkladmi operácií s poľami a reťazcami

Ciele učenia:

Zavedenie polí a reťazcov Praktické precvičenie

Štruktúra/činnosť:

Recenzia predchádzajúcej lekcie (5 minút)

Diskusia (5 minút): Zopakujte si koncept funkcií a ich využitie v C. Vyzdvihnite, ako môžu funkcie zjednodušiť prácu s poľami a reťazcami.

Úvod do polí (15 minút)

Vysvetlenie (10 minút): Vysvetlite, čo sú polia, vrátane ich syntaxe a ako ich deklarovať a inicializovať. Diskutujte o rôznych typoch polí, ako sú jednorozmerné a viacrozmerné polia.

Príklad (5 minút): Napíšte program, ktorý použije pole na uloženie a sčítanie zoznamu čísel. Vysvetlite každý krok programu, od deklarácie až po iteráciu a súčet.

Praktická aktivita č. 1: Manipulácia s poľami (15 minút)

Aktivita (10 minút): Študenti napíšu program, ktorý používa pole na ukladanie používateľských vstupov (napr. zoznam známk) a potom vypočíta priemer.

Podpora (5 minút): Pomôžte študentom ladiť problémy, ako sú chyby v poli mimo hraníc a nesprávne indexovanie.

Úvod do sláčikov (15 minút)

Vysvetlenie (10 minút): Diskutujte o tom, ako sú reťazce reprezentované v C pomocou polí znakov. Vysvetlite, ako deklarovať, inicializovať a manipulovať s reťazcami.

Príklad (5 minút): Napíšte program, ktorý prevezme zadaný reťazec od používateľa a vytlačí ho naopak.

Preveďte študentov logikou manipulácie so strunami.

Praktická aktivita č. 2: Operácie s reťazcom (15 minút)

Aktivita (10 minút): Žiaci napíšu program, ktorý sa pýta na meno používateľa a následne vykonáva rôzne operácie, ako je počítanie počtu znakov alebo nájdenie prvej samohlásky.

Podpora (5 minút): Poskytnite pomoc pri problémoch s ukončením reťazcov a bežných chybách pri manipulácii s reťazcami.

Otázky a odpovede a rekapitulácia (5 minút)

Otázky a odpovede (3 minúty): Umožnite študentom klásť otázky o poliach a reťazcoch. Diskutujte o dôležitosti pochopenia týchto pojmov pre pokročilejšie programovanie.

Rekapitulácia a domáca úloha (2 minúty): Zhrňte lekciu. Zadať domácu úlohu, v ktorej študenti napíšu program, ktorý používa polia aj reťazce na správu jednoduchého zoznamu kontaktov.

Plán hodiny – cvičenia

Séria úvodných lekcí o programovaní v jazyku C

Téma: Polia a reťazce

Pracovný list č. 5

1. Úvod do polí

Kľúčové body:

Polia sú kolekcie premenných rovnakého typu.

Polia majú nulový index; prvý prvok je na indexe 0.

Príklad kódu:

```
int čísla[5] = {1, 2, 3, 4, 5};  
printf("Prve cislo: %d\n", cisla[0]);
```

2. Úvod do sláčikov

Kľúčové body:

Reťazce v C sú polia znakov končiace znakom null `'\0'`.

Príklad kódu:

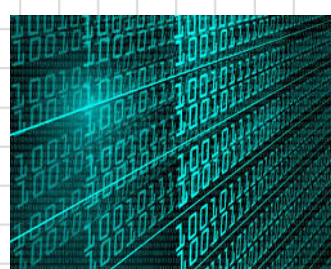
```
názov znaku[20];  
printf("Zadajte svoje meno: ");  
scanf("%s", meno);  
printf("Ahoj, %s!\n", meno);
```

3. Domáca úloha:

Napíšte program, ktorý používa polia aj reťazce na správu jednoduchého zoznamu kontaktov.



PLÁN HODINY



PREDMET: PROGRAMOVANIE A KÓDOVANIE

TÉMA: ÚVODNÉ LEKCIE O PROGRAMOVACOM JAZYKU PYTHON – LEKCIA 1

ZAMERANIE A CIELE LEKCIE:

1. TÁTO LEKCIA OBOZNÁMI ŠTUDENTOV SO ZÁKLADMI PROGRAMOVACIEHO JAZYKA PYTHON VRÁTANE JEHO HISTÓRIE, JAZYKOVÝCH VLASTNOSTÍ A PÍSANIA PRVÉHO PROGRAMU PYTHON.
2. NA KONCI TEJTO LEKCIE BY ŠTUDENTI MALI BYŤ SCHOPNÍ POROZUMIEŤ ZÁKLADNEJ SYNTAXI JAZYKA PYTHON A BYŤ SCHOPNÍ ZOSTAVIŤ A SPUSTIŤ JEDNODUCHÉ "HELLO, WORLD" PROGRAM.

POTREBNÉ MATERIÁLY:

- POČÍTAČE S NAINŠTALOVANÝM PYTHONOM (NAPR. VS CODE, PYCHARM ALEBO INÉ VÝVOJOVÉ PROSTREDIE)
- PROJEKTOR A PLÁTNO
- TABUĽA A FIXKY
- ZÁKLADNÁ SYNTAX A PRIRADENIA ŠTRUKTÚR PYTHONU

1. ÚVOD DO JAZYKA PYTHON (15 MINÚT)

ROZHOVOR (5 MINÚT): ZAČNITE STRUČNOU HISTÓRIOU PROGRAMOVACIEHO JAZYKA PYTHON, KTORÝ VYVINUL GUIDO VAN ROSSUM KONCOM OSEMDESIATYCH ROKOV. DISKUTUJTE O VÝVOJI JAZYKA A O TOM, AKO SA STAL POPULÁRNYM VĎAKA JEHO L'AHKÉMU UČENIU A ŠIROKÉMU POUŽITIU. ZDÔRAZNITE, AKO PYTHON POLOŽIL ZÁKLADY PRE MODERNÉ PROGRAMOVACIE JAZYKY A AKO JE OBZVLÁŠŤ POPULÁRNY VO VEDECKEJ VÝPOČTOVEJ TECHNIKE, VÝVOJI WEBOVÝCH APLIKÁCIÍ A SPRACOVANÍ ÚDAJOV.

PREZENTÁCIA (10 MINÚT): NAČRTNITE SÚČASNÉ VYUŽITIE JAZYKA PYTHON V RÔZNYCH ODVETVIACH. DISKUTUJTE O TOM, PREČO BY STE SA MALI NAUČIŤ PYTHON, A UKÁŽTE PRÍKLADY POPULÁRNYCH APLIKÁCIÍ ZALOŽENÝCH NA PYTHON, AKO JE NAPRÍKLAD ANALÝZA ÚDAJOV PRE YOUTUBE, INSTAGRAM A NASA.

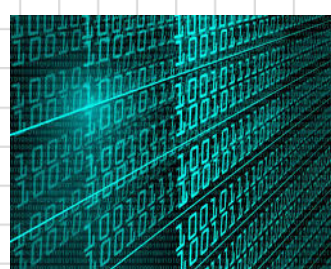
2. ZÁKLADNÁ SYNTAX A ŠTRUKTÚRA (20 MINÚT)

VYSVETLENIE (10 MINÚT): UKÁŽTE ŠTRUKTÚRU JEDNODUCHÉHO PROGRAMU PYTHON. VYSVETLITE POUŽITIE IMPORTU, ÚLOHU FUNKCIE MAIN() A SYNTAKTICKÉ VLASTNOSTI PYTHONU, AKO JE DÔLEŽITOSŤ ODSADZOVANIA. VYSVETLITE, AKO PYTHON NARÁBA S VÝRAZMI A AKÉ CHYBY MOŽNO UROBIŤ. POUKAZUJTE NA JEDNODUCHÉ CHYBY, AKO JE NAPRÍKLAD NEODSADENIE, A VYSVETLITE, AKO SA IM VYHNÚŤ.

PRÍKLAD (10 MINÚT): NAPIŠTE "AHOJ, SVET" PROGRAM NA DOSKE ALEBO PLÁTNE PROJEKTORA. PODROBNE VYSVETLITE ÚLOHU KAŽDÉHO RIADKU KÓDU, VRÁTANE TOHO, AKO FUNGUJE FUNKCIA PRINT(), ZAOBCHÁDZANIE S REŤAZCAMI A POCHOPENIE VÝSLEDKU VÝSTUPU. POVZBUDZUJTE ŠTUDENTOV, ABY KLÁDLI OTÁZKY O KAŽDEJ ČASTI KÓDU.



PLÁN HODINY



3. PRAKTICKÁ AKTIVITA (20 MINÚT)

AKTIVITA (15 MINÚT): POŽIADAJTE ŠTUDENTOV, ABY NAPÍSAI SVOJE VLASTNÉ "AHOJ, SVET" ICH PROGRAM V PYTHONÉ. POMÔŽTE IM NASTAVIŤ ICH VÝVOJOVÉ PROSTREDIE (NAPRÍKLAD POMOCOU IDLE ALEBO PYCHARM) A UISTITE SA, ŽE KAŽDÝ MÔŽE ÚSPEŠNE SPUSTIŤ ICH PROGRAM. UISTITE SA, ŽE VŠETCI ŠTUDENTI CHÁPU, AKO KÓD FUNGUJE A AKO SA POHYBOVAŤ V PROSTREDÍ PYTHONU.

PEER REVIEW (5 MINÚT): SPÁRUJTE ŠTUDENTOV, ABY SI NAVZÁJOM PREZRELI SVOJ KÓD. POVZBUDZUJTE ICH, ABY DISKUTOVALI O AKÝCHKOL'VEK OTÁZKACH ALEBO PROBLÉMOCH, KTORÉ SA VYSKYTNÚ, A POMÁHAJTE SI NAVZÁJOM ZLEPŠOVAŤ KÓD.

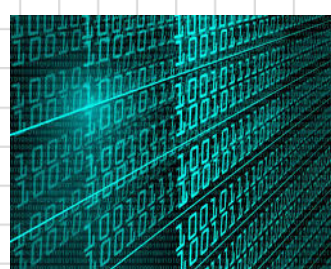
4. OTÁZKY, ODPOVEDE A ZHRNUTIE (5 MINÚT)

OTÁZKY A ODPOVEDE (3 MINÚTY): OTVORTE PRIESTOR PRE ŠTUDENTOV, ABY MOHLI KLÁŠŤ OTÁZKY. DISKUTUJTE O BEŽNÝCH PROBLÉMOCH ALEBO BODOCH ZMÄTKU, KTORÉ SA MOHLI POČAS HODINY VYSKYTNÚŤ. POSKYTNITE ŠTUDENTOM PRÍLEŽITOSŤ POSKYTNÚŤ SPÄTNÚ VÄZBU O TOM, AKO POCHOPILI LEKCIU.

ZHRNUTIE A DOMÁCA ÚLOHA (2 MINÚTY): ZHRŇTE HLAVNÉ BODY LEKCIE, ZVÝRAZNITE ZÁKLADNÚ SYNTAX PYTHONU A "AHOJ, SVET" DÔLEŽITOSŤ PROGRAMU. AKO DOMÁCU ÚLOHU POŽIADAJTE ŠTUDENTOV, ABY NAŠLI JEDNODUCHÝ PROGRAM PYTHON ONLINE, UPRAVILI JEHO VÝSTUPNÚ SPRÁVU A SPUSTILI HO. POŽIADAJTE ICH, ABY PRINIESLI UPRAVENÝ KÓD DO ĎALŠEJ TRIEDY A PRIPRAVILI SA NA JEHO PREZENTÁCIU.



PLÁN HODINY



PREDMET: PROGRAMOVANIE A KÓDOVANIE

TÉMA: ÚVODNÉ LEKCIE O PROGRAMOVACOM JAZYKU PYTHON – LEKCIA 1

1. ÚVOD DO JAZYKA PYTHON

HLAVNÉ BODY:

PYTHON JE L'AHKO NAUČITELNÝ PROGRAMOVACÍ JAZYK NA VYSOKEJ ÚROVNI POUŽÍVANÝ V MNOHÝCH OBLASTIACH, AKO JE VÝVOJ WEBOVÝCH APLIKÁCIÍ, SPRACOVANIE ÚDAJOV, VEDECKÉ VÝPOČTY A UMELÁ INTELIGENCIA.

VÝVOJ JAZYKA PYTHON SA DATUJE DO KONCA OSEMDESIATYCH ROKOV MINULÉHO STOROČIA, KEĎ GUIDO VAN ROSSUM VYTVORIL JAZYK PRE L'AHKÉ UČENIE A ČITATEĽNOSŤ.

BEŽNÉ APLIKÁCIE PYTHONU ZAHŔŇAJÚ POPULÁRNE PLATFORMY, AKO SÚ YOUTUBE, INSTAGRAM A ANALYTIKA ÚDAJOV NASA.

PRÍKLAD PROGRAMU:

"AHOJ SVET" – JEDEN Z NAJJEDNODUCHŠÍCH PROGRAMOV PYTHON:

VYTLAČIŤ ("AHOJ, SVET")

2. ZÁKLADNÁ SYNTAX A ŠTRUKTÚRA

HLAVNÉ BODY:

V PYTHONE JE ŠTRUKTÚRA KÓDU VEĽMI ZÁVISLÁ OD ODSADENIA. ODSADENIE DEFINUJE BLOKY KÓDU, TAKŽE JE NEVYHNUTNÉ HO SPRÁVNE POUŽIŤ.

FUNKCIA PRINT() JE JEDNÝM Z NAJZÁKLADNEJŠÍCH NÁSTROJOV V PYTHONE, KTORÝ NÁM UMOŽŇUJE ZOBRAZIŤ TEXT A ĎALŠIE ÚDAJE NA OBRAZOVKE.

PYTHON NA ROZDIEL OD MNOHÝCH INÝCH PROGRAMOVACÍCH JAZYKOV NEVYŽADUJE NA KONCI RIADKOV BODKOČIARKY.

AKTIVITA:

NAPÍŠTE A SPUSTITE "AHOJ, SVET!" PROGRAM VO VLASTNOM VÝVOJOVOM PROSTREDÍ.

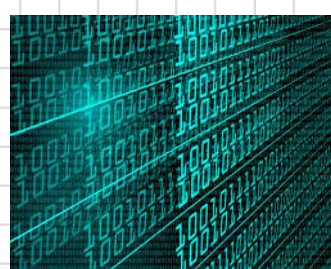
URČITE ÚČEL KAŽDÉHO RIADKU PROGRAMU A UISTITE SA, ŽE ROZUMIETE TOMU, AKO PROGRAM FUNGUJE.

3. DOMÁCE ÚLOHY

VYTVORTE PROGRAM PYTHON, KTORÝ ZNIE "AHOJ, SVET!" ZOBRAZÍ OSOBNÚ SPRÁVU (NAPRÍKLAD VAŠE MENO ALEBO OBL'ÚBENÝ CITÁŤ) NAMIESTO SPRÁVY. SKÚSTE EXPERIMENTOVAŤ S FUNKCIOU PRINT(), AKO JE POUŽITIE RÔZNYCH ÚVODZOVIEK ALEBO PRIDANIE NOVÉHO RIADKU DO PROGRAMU, KTORÝ ZOBRAZUJE ĎALŠÍ TEXT. UPRAVENÝ PROGRAM SI ULOŽTE A PRINESTE HO DO ĎALŠEJ TRIEDY, ABY STE HO MOHLI PREZENTOVAŤ TRIEDE.



PLÁN HODINY



PREDMET: PROGRAMOVANIE A KÓDOVANIE

TÉMA: HODNOTY A TYPY V PROGRAMOVACOM JAZYKU PYTHON - LEKCIA 2

ZAMERANIE A CIELE LEKCIE:

ÚVOD DO ZÁKLADNÝCH DÁTOVÝCH TYPOV POUŽÍVANÝCH V PYTHON, VRÁTANE CELÝCH ČÍSEL, REÁLNYCH ČÍSEL, REŤAZCOV A LOGICKÝCH HODNÔT.

NA KONCI HODINY BY ŠTUDENTI MALI BYŤ SCHOPNÍ IDENTIFIKOVAŤ A SPRÁVNE APLIKOVAŤ ZÁKLADNÉ DÁTOVÉ TYPY A POROZUMIEŤ TOMU, AKO OPERÁCIE OVPLYVŇUJÚ TYP TÝCHTO HODNÔT.

POTREBNÉ MATERIÁLY:

- POČÍTAČE S NAINŠTALOVANÝM PYTHONOM (NAPRÍKLAD VS CODE, PYCHARM ALEBO INÉ VÝVOJOVÉ PROSTREDIE)
- PROJEKTOR ALEBO PLÁTNO.
- TABUĽA A POPISOVAČE
- PRÍLOHY ILUSTRUJÚCE TYPY ÚDAJOV

ÚVOD DO HODNÔT A TYPOLÓGIE (15 MINÚT)

DISKUSIA (5 MINÚT): ZAČNITE STRUČNOU DISKUSIOU O HODNOTE PROGRAMOVANIA A DÔLEŽITOSTI DÁTOVÝCH TYPOV. PREBERTE, AKO HODNOTY REPREZENTUJÚ ÚDAJE V PROGRAMOCH A AKO TYPY ÚDAJOV URČUJÚ, AKÝM SPÔSOBOM JE MOŽNÉ TIETO ÚDAJE SPRACOVAŤ.

PREZENTÁCIA (10 MINÚT): PREDSTAVTE RÔZNE TYPY ÚDAJOV, AKO SÚ CELÉ ČÍSLA, ČÍSLA S POHYBLIVOU RÁDOVOU ČIARKOU, REŤAZCE A BOOLOVSKÉ HODNOTY. DISKUTUJTE O HLAVNÝCH CHARAKTERISTIKÁCH A APLIKÁCIÁCH KAŽDÉHO TYPU. UVEĎTE PRÍKLADY ICH VYUŽITIA V REÁLNOH ŽIVOTE, AKO SÚ MATEMATICKÉ OPERÁCIE, SPRACOVANIE TEXTU A PODMIENENÁ LOGIKA.

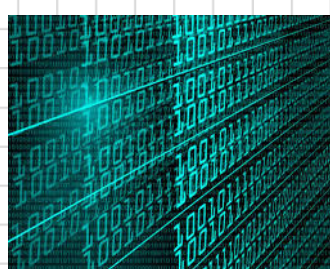
2. ZÁKLADNÁ SYNTAX A ŠTRUKTÚRA (20 MINÚT)

VYSVETLENIE (10 MINÚT): PREDSTAVTE, AKO JE MOŽNÉ DEFINOVAŤ A VYUŽÍVAŤ ZÁKLADNÉ DÁTOVÉ TYPY V PYTHON. OBJAVTE, AKO SA VÝSLEDOK VÝRAZU MENÍ V ZÁVISLOSTI OD ZVOLENÉHO TYPU ÚDAJOV. NAPRÍKLAD, AKO SÚČET CELÉHO ČÍSLA A ČÍSLA S POHYBLIVOU RÁDOVOU ČIARKOU OVPLYVŇUJE VÝSLEDOK A ČO NASTANE, KEĎ SA POKÚSIME SKOMBINOVAŤ CELÉ ČÍSLO A TEXT.

PRÍKLAD (10 MINÚT): PREDLOŽTE JEDNODUCHÉ PRÍKLADY APLIKÁCIE RÔZNYCH TYPOV HODNÔT NA TABUĽU ALEBO PLÁTNO PROJEKTORA. DEMONŠTRUJTE, AKO SČÍTAŤ DVE ČÍSLA, AKO SPOJIŤ DVA TEXTY A AKO VYUŽIŤ LOGICKÉ OPERÁTORY NA FORMULOVANIE PODMIENOK. UVEĎTE, ŽE SPRÁVNE POUŽÍVANIE HODNÔT A TYPOV JE KLÚČOVÉ PRE EFEKTÍVNE FUNGOVANIE PROGRAMOV.



PLÁN HODINY



3. PRAKTICKÁ ČINNOSŤ (20 MINÚT)

AKTIVITA (15 MINÚT): POŽIADATE ŠTUDENTOV, ABY NAPÍSAI JEDNODUCHÝ KÓD V PYTHONĚ, KTORÝ VYUŽÍVA RÔZNE TYPY HODNÔT, AKO SÚ CELÉ ČÍSLA, ČÍSLA S POHYBLIVOU RÁDOVOU ČIARKOU, REŤAZCE A BOOLOVSKÉ HODNOTY. POMÔŽTE IM PRI NASTAVOVANÍ VÝVOJOVÉHO PROSTREDIA A ZABEZPEČTE, ABY KAŽDÝ MOHOL ÚSPEŠNE SPUSTIŤ SVOJ PROGRAM. OVERTE, ČI ŠTUDENTI ROZUMEJÚ VÝSLEDKOM A ROZDIELOM MEDZI JEDNOTLIVÝMI TYPMI.

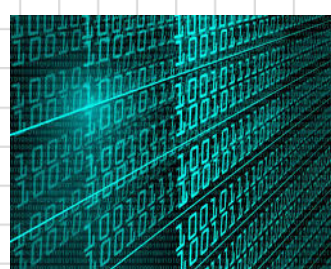
4. OTÁZKY, ODPOVEDE A SÚHRN (5 MINÚT)

OTÁZKY A ODPOVEDE (3 MINÚTY): UMOŽNITE ŠTUDENTOM KLÁŠŤ OTÁZKY. PREBERTE BEŽNÉ PROBLÉMY ALEBO NEJASNOSTI, KTORÉ SA MOHLI POČAS HODINY OBJAVIŤ. POSKYTNITE ŠTUDENTOM PRÍLEŽITOSŤ VYJADRIŤ SPÄTNÚ VÄZBU O TOM, AKO POROZUMELI LEKCII.

ZHRNUTIE A DOMÁCA ÚLOHA (2 MINÚTY): ZHRŇTE KLÚČOVÉ BODY LEKCIE A ZDÔRAZNITE VÝZNAM RÔZNYCH TYPOV ÚDAJOV A ICH SPRÁVNEHO VYUŽITIA. AKO DOMÁCU ÚLOHU POŽIADAJTE ŠTUDENTOV, ABY VYTVORILI JEDNODUCHÝ PROGRAM V PYTHONĚ, KTORÝ VYUŽÍVA RÔZNE TYPY HODNÔT. MÔŽETE ICH POŽIADAŤ, ABY NAPÍSAI PROGRAM, KTORÝ VYPOČÍTA SÚČET DVOCH ČÍSEL A NÁSLEDNE ZOBRAZÍ SPRÁVU S VÝSLEDKOM.



PLÁN HODINY



PREDMET: PROGRAMOVANIE A KÓDOVANIE

TÉMA: HODNOTY A TYPY V PROGRAMOVACOM JAZYKU PYTHON - LEKCIA 2

HODNOTY A KATEGÓRIE

HLAVNÉ BODY:

HODNOTY SÚ ZÁKLADNÝMI KOMPONENTMI PROGRAMOVANIA, KTORÉ REPREZENTUJÚ DÁTA.

PYTHON PODPORUJE RÔZNE DÁTOVÉ TYPY, VRÁTANE CELÝCH ČÍSEL, ČÍSEL S POHYBLIVOU RÁDOVOU ČIARKOU, REŤAZCOV A LOGICKÝCH HODNÔT.

TYPY ÚDAJOV URČUJÚ, AKÝM SPÔSOBOM MÔŽU BYŤ ÚDAJE SPRACOVANÉ A PREZENTOVANÉ V APLIKÁCIÁCH.

PRÍKLAD PROGRAMU:

```
lesson_plan_values_and_types.py > ...  
1  # Examples of different types of values  
2  x = 5                                # integer  
3  y = 3.14                            # floating-point number  
4  name = "Python"                    # string  
5  is_fun = True                      # boolean  
6  
7  print(x + y)                        # Adding two numbers  
8  print(name + " is cool!")          # Concatenating text  
9  print(is_fun and (x > 2))          # Logical expression
```

2. PRAKTICKÁ ÚLOHA

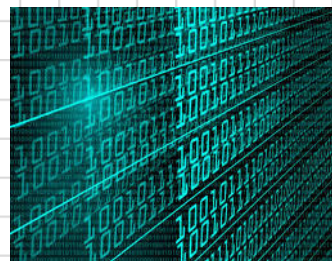
ÚLOHA: VYTVORTE PROGRAM V PYTHONE, KTORÝ VYUŽÍVA HODNOTY RÔZNYCH TYPOV. NAPRÍKLAD SČÍTAJTE DVE ČÍSLA, ZOBRAZTE SPRÁVU NA OBRAZOVKE A POMOCOU LOGICKÉHO OPERÁTORA OVERTE PODMIENKU.

3. DOMÁCE ÚLOHY

NAPÍŠTE PROGRAM, KTORÝ VYPOČÍTA PRIEMER TROCH ČÍSEL A NÁSLEDNE ZOBRAZÍ VÝSLEDOK V SPRÁVE. EXPERIMENTUJTE S RÔZNYMI TYPMI HODNÔT A OPERÁTOROV. ULOŽTE PROGRAM A PRINESTE HO NA ĎALŠIU HODINU, ABY STE HO MOHLI PREZENTOVAŤ TRIEDE.



PLÁN HODINY



PREDMET: PROGRAMOVANIE A KÓDOVANIE

TÉMA: PREMENNÉ V PROGRAMOVACOM JAZYKU PYTHON – LEKCIA 3

ZAMERANIE A CIELE LEKCIE:

ÚVOD DO POJMU PREMENNÉ, ICH VÝZNAM A APLIKÁCIA PRI PROGRAMOVANÍ V PYTHONĚ.

NA KONCI HODINY BY ŠTUDENTI MALI BYŤ SCHOPNÍ VYTVARAŤ PREMENNÉ, PRIRAĐOVAŤ HODNOTY A ROZUMIEŤ, AKO MÔŽU BYŤ PREMENNÉ VYUŽITÉ V RÔZNYCH ČASTIACH PROGRAMU.

POTREBNÉ MATERIÁLY:

- POČÍTAČE S NAINŠTALOVANÝM PYTHONOM (NAPRÍKLAD VS CODE, PYCHARM ALEBO INÉ VÝVOJOVÉ PROSTREDIE)
- PROJEKTOR ALEBO PROJEKČNÉ PLÁTNO
- TABUĽA A POPISOVAČE
- PRÍLOHY ILUSTRUJÚCE TYPY ÚDAJOV

ÚVOD DO CHANGELINGS (15 MINÚT)

DISKUSIA (5 MINÚT): ZAČNITE OBJASNENÍM, ČO PREDSTAVUJE PREMENNÁ A AKÝ MÁ VÝZNAM V PROGRAMOVANÍ. ROZPRÁVAJTE O TOM, AKO JE MOŽNÉ VYUŽIŤ PREMENNÉ NA UCHOVÁVANIE A MANIPULÁCIU S ÚDAJMI V PROGRAMOCH.

PREZENTÁCIA (10 MINÚT): NAČRTNITE RÔZNE TYPY PREMENNÝCH A OBJASNITE ICH FUNGOVANIE. PREDVEĎTE, AKO ULOŽIŤ ČÍSLA, TEXT ALEBO LOGICKÉ HODNOTY DO PREMENNEJ. ZDÔRAZNITE KONVENCIE POMENOVANIA PREMENNÝCH A OSVEDČENÉ POSTUPY.

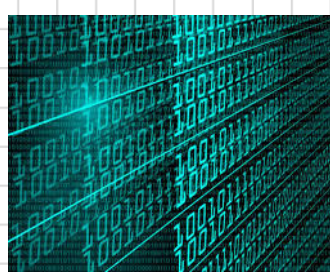
2. VYUŽITIE PREMENNÝCH V PYTHONĚ (20 MINÚT)

VYSVETLENIE (10 MINÚT): PREDVEĎTE, AKO VYTVORIŤ A POUŽÍVAŤ PREMENNÉ V PYTHONĚ. OBJAVTE PRIRADENIE HODNOTY PREMENNEJ, PREHODNOTENIE PREMENNEJ (PRIRADENIE NOVEJ HODNOTY) A KONVERZIU MEDZI RÔZNYMI TYPMI PREMENNÝCH (KONVERZIA TYPU).

PRÍKLAD (10 MINÚT): PREDVEĎTE JEDNODUCHÝ KÓD NA TABUĽU ALEBO PLÁTNŮ PROJEKTORA, KTORÝ ILUSTRUJE VYTVARANIE, PRIRAĐOVANIE A VYUŽÍVANIE PREMENNÝCH V PROGRAME. OBJAVTE, AKO MOŽNO PREMENNÚ APLIKOVAŤ V RÔZNYCH OPERÁCIÁCH A AKÝM SPÔSOBOM PREMENNÉ ZEFEKTÍVŇUJÚ KÓD.



PLÁN HODINY



3. PRAKTICKÁ ČINNOSŤ (20 MINÚT)

AKTIVITA (15 MINÚT): POŽIADAJTE ŠTUDENTOV, ABY VYTVORILI JEDNODUCHÝ PROGRAM V JAZYKU PYTHON, KTORÝ VYUŽÍVA NIEKOĽKO PREMENNÝCH. NAPRÍKLAD, NECH VYTVORIA PREMENNÉ S RÔZNYMI TYPMI HODNÔT A NÁSLEDNE VYKONAJÚ OPERÁCIE S TÝMITO PREMENNÝMI. POMÔŽTE IM PRI NASTAVOVANÍ VÝVOJOVÉHO PROSTREDIA A ZABEZPEČTE, ABY KAŽDÝ MOHOL ÚSPEŠNE SPUSTIŤ SVOJ PROGRAM.

PRÁCA V PÁROCH (5 MINÚT): SPOJTE ŠTUDENTOV, ABY SI NAVZÁJOM SKONTROLOVALI SVOJ KÓD. POVZBUDZUJTE ICH, ABY DISKUTOVALI O AKÝCHKOL'VEK OTÁZKACH ALEBO PROBLÉMOCH, KTORÉ SA OBJAVIA, A VZÁJOMNE SI POMOHLI ZLEPŠIŤ KÓD.

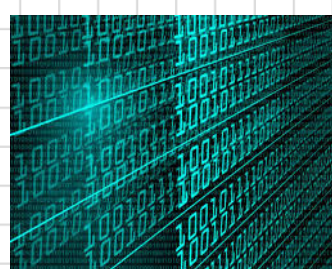
4. OTÁZKY, ODPOVEDE A SÚHRN (5 MINÚT)

OTÁZKY A ODPOVEDE (3 MINÚTY): UMOŽNITE ŠTUDENTOM KLÁŠŤ OTÁZKY. PREBERTE BEŽNÉ PROBLÉMY ALEBO NEJASNOSTI, KTORÉ SA MOHLI POČAS HODINY OBJAVIŤ. POSKYTNITE ŠTUDENTOM PRÍLEŽITOSŤ VYJADRIŤ SPÄTNÚ VÄZBU O TOM, AKO POROZUMELI LEKCII.

ZHRNUTIE A DOMÁCA ÚLOHA (2 MINÚTY): ZHRŇTE KLÚČOVÉ BODY HODINY, ZDÔRAZNITE ZÁKLADNÉ ASPEKTY POUŽÍVANIA PREMENNÝCH A ICH VÝZNAM. AKO DOMÁCU ÚLOHU POŽIADAJTE ŠTUDENTOV, ABY VYTVORILI PROGRAM V PYTHONE, KTORÝ VYUŽÍVA RÔZNE TYPY PREMENNÝCH A VYKONÁVA S NIMI RÔZNE OPERÁCIE.



PLÁN HODINY



PREDMET: PROGRAMOVANIE A KÓDOVANIE

TÉMA: PREMENNÉ V PROGRAMOVACOM JAZYKU PYTHON - LEKCIA 3

PREMENNÉ

HLAVNÉ BODY:

PREMENNÉ PREDSTAVUJÚ KONTAJNERY, KTORÉ UCHOVÁVAJÚ ÚDAJE POČAS VYKONÁVANIA PROGRAMU.

PRI DEFINOVANÍ PREMENNÝCH IM PRIRAĐUJEME NÁZOV A POŽADOVANÚ HODNOTU. NÁZVY PREMENNÝCH SA RIADIA PRAVIDLAMI SYNTAXE PYTHONU, PRIČOM TIETO NÁZVY VÁM UMOŽŇUJÚ IDENTIFIKOVAŤ A MANIPULOVAŤ S ÚDAJMI VO VAŠOM PROGRAME.

PRÍKLAD PROGRAMU:

```
variables_example.py > ...
1  # Example of using variables
2  x = 10                # integer variable
3  y = 5.5               # floating-point variable
4  name = "Alice"        # string variable
5  is_happy = True       # boolean variable
6
7  # Performing operations with variables
8  sum_result = x + y
9  greeting = "Hello, " + name
10 status = is_happy and (x > y)
11
12 print(sum_result)     # Output the result of the addition
13 print(greeting)       # Output the greeting message
14 print(status)         # Output the logical operation result
```

2. PRAKTICKÁ ÚLOHA

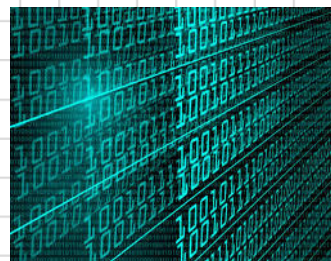
ÚLOHA: VYTVORTE PROGRAM V PYTHONE, KTORÝ VYUŽÍVA HODNOTY RÔZNYCH TYPOV. NAPRÍKLAD SČÍTAJTE DVE ČÍSLA, ZOBRAZTE SPRÁVU NA OBRAZOVKE A POMOCOU LOGICKÉHO OPERÁTORA OVERTE PODMIENKU.

3. DOMÁCA ÚLOHA

NAPÍŠTE PROGRAM NA VÝPOČET PLOCHY A OBVODU KRUHU S DANÝM POLOMEROM A VYUŽITE PREMENNÉ NA UCHOVANIE HODNÔT. ULOŽTE PROGRAM A PRINESTE HO NA ĎALŠIU HODINU, ABY STE HO MOHLI PREZENTOVAŤ TRIEDE.



PLÁN HODINY



PREDMET: PROGRAMOVANIE A KÓDOVANIE

TÉMA: OPERÁTORSKÉ SIGNÁLY A OPERANDY V PROGRAMOVACOM JAZYKU PYTHON - LEKCIA 4

ZAMERANIE A CIELE VYUČOVANIA:

ÚVOD DO RÔZNYCH OPERAČNÝCH SIGNÁLOV (ARITMETICKÉ, POROVNÁVACIE, LOGICKÉ ATĎ.) A ICH APLIKÁCIA V PYTHONĚ.

NA KONCI HODINY BY ŠTUDENTI MALI BYŤ SCHOPNÍ FORMULOVAŤ A HODNOTIŤ JEDNODUCHÉ VÝRAZY A CHÁPAŤ VZŤAHY MEDZI RÔZNYMI OPERÁCIAMI A OPERANDMI.

POTREBNÉ MATERIÁLY:

- POČÍTAČE S NAINŠTALOVANÝM PYTHONOM (NAPRÍKLAD VS CODE, PYCHARM ALEBO INÉ VÝVOJOVÉ PROSTREDIE)
- PROJEKTOR ALEBO PROJEKČNÉ PLÁTNO
- TABUĽA A POPISOVAČE
- DODATOK ZOBRAZUJE PREVÁDZKOVÉ SIGNÁLY A OPERANDY.

ÚVOD DO OPERAČNÝCH SIGNÁLOV A OPERANDOV (15 MINÚT)

DISKUSIA (5 MINÚT): ZAČNITE OBJASNENÍM, ČO PREDSTAVUJE OPERAČNÝ SYMBOL A OPERAND, A AKÝM SPÔSOBOM SA APLIKUJÚ V PROGRAMOVANÍ. MÔŽETE UVIEŠŤ PRÍKLADY Z BEŽNÉHO ŽIVOTA, AKO NAPRÍKLAD VYKONÁVANIE MATEMATICKÝCH OPERÁCIÍ.

PREZENTÁCIA (10 MINÚT): PREDSTAVTE RÔZNE PREVÁDZKOVÉ SIGNÁLY (NAPR. SČÍTANIE, ODČÍTANIE, NÁSOBENIE, DELENIE ATĎ.) A ICH SYNTAX V PYTHONĚ. DISKUTUJTE O TOM, AKO TIETO OPERÁTORY INTERAGUJÚ SO SVOJIMI OPERANDMI, A OBJASNITE ROZDIELY MEDZI ARITMETICKÝMI A LOGICKÝMI OPERÁTORMI.

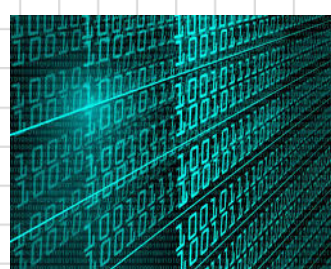
2. POUŽÍVANIE OPERÁTOROV A OPERANDOV V PYTHONĚ (20 MINÚT)

VYSVETLENIE (10 MINÚT): PREDSTAVTE, AKO VYTVÁRAŤ A APLIKOVAŤ ZÁKLADNÉ VÝRAZY V PYTHONĚ PROSTREDNÍCTVOM OPERÁTOROV A OPERANDOV. OBJAVTE PROCES VYHODNOCOVANIA VÝRAZOV A SPÔSOB, AKÝM TIETO OPERÁCIE GENERUJÚ VÝSLEDKY.

PRÍKLAD (10 MINÚT): UVEĎTE JEDNODUCHÝ PRÍKLAD KÓDU NA TABUĽU ALEBO OBRAZOVKU PROJEKTORA, KTORÝ ILUSTRUJE, AKO APLIKOVAŤ ZÁKLADNÉ ARITMETICKÉ A LOGICKÉ OPERÁTORY V PYTHONĚ. OBJAVTE, AKO TIETO OPERÁCIE FUNGUJÚ S RÔZNYMI TYPMI OPERANDOV, AKO SÚ ČÍSLA A BOOLEANY.



PLÁN HODINY



3. PRAKTICKÁ ČINNOSŤ (20 MINÚT)

AKTIVITA (15 MINÚT): POŽIADAJTE ŠTUDENTOV, ABY VYTVOŘILI JEDNODUCHÝ PROGRAM V JAZYKU PYTHON, KTORÝ VYUŽÍVA RÔZNE OPERAČNÉ SYMBOLY A OPERANDY. NAPRÍKLAD, NECH VYPOČÍTAJÚ ZÁKLADNÉ ARITMETICKÉ OPERÁCIE A ZOSTAVIA VÝRAZY, KTORÉ APLIKUJÚ LOGICKÉ OPERÁTORY. POMÔŽTE IM S NASTAVENÍM VÝVOJOVÉHO PROSTREDIA A ZABEZPEČTE, ABY KAŽDÝ MOHOL ÚSPEŠNE SPUSTIŤ SVOJ PROGRAM.

PRÁCA V PÁROCH (5 MINÚT): SPOJTE ŠTUDENTOV, ABY SI NAVZÁJOM SKONTROLOVALI SVOJ KÓD. POVZBUDZUJTE ICH, ABY DISKUTOVALI O AKÝCHKOL'VEK OTÁZKACH ALEBO PROBLÉMOCH, KTORÉ SA OBJAVIA, A VZÁJOMNE SI POMOHLI ZLEPŠIŤ KÓD.

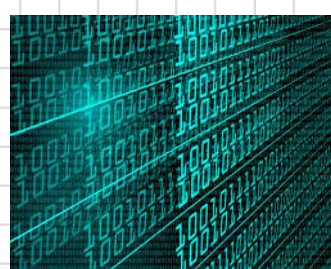
4. OTÁZKY, ODPOVEDE A SÚHRN (5 MINÚT)

OTÁZKY A ODPOVEDE (3 MINÚTY): UMOŽNITE ŠTUDENTOM KLÁSTŤ OTÁZKY. PREBERTE BEŽNÉ PROBLÉMY ALEBO NEJASNOSTI, KTORÉ SA MOHLI POČAS HODINY OBJAVIŤ. POSKYTNITE ŠTUDENTOM PRÍLEŽITOSŤ VYJADRIŤ SPÄTNÚ VÄZBU O TOM, AKO POROZUMELI LEKCII.

ZHRNUTIE A DOMÁCA ÚLOHA (2 MINÚTY): ZHRŇTE KLÚČOVÉ BODY LEKCIE, PRIČOM ZDÔRAZNITE APLIKÁCIU OPERÁTOROV A OPERANDOV V PYTHON. AKO DOMÁCU ÚLOHU POŽIADAJTE ŠTUDENTOV, ABY VYTVOŘILI PROGRAM V PYTHON, KTORÝ ZAHŔŇA RÔZNE ARITMETICKÉ A LOGICKÉ VÝRAZY.



PLÁN HODINY



PREDMET: PROGRAMOVANIE A KÓDOVANIE

TÉMA: OPERÁTORSKÉ SIGNÁLY A OPERANDY V PROGRAMOVACOM JAZYKU PYTHON - LEKCIA 4

OPERAČNÉ SIGNÁLY A OPERANDY

HLAVNÉ BODY:

POMOCOU OPERAČNÝCH SIGNÁLOV MÔŽEME REALIZOVAŤ RÔZNE OPERÁCIE S OPERANDMI, KTORÝMI MÔŽU BYŤ ČÍSLA, TEXTY ALEBO LOGICKÉ HODNOTY.

ARITMETICKÉ OPERÁCIE (NAPRÍKLAD SČÍTANIE, ODČÍTANIE) A LOGICKÉ OPERÁTORY (NAPRÍKLAD A, ALEBO NIE) PREDSTAVUJÚ ZÁKLADNÉ NÁSTROJE PROGRAMOVANIA. JE NEVYHNUTNÉ POROZUMIEŤ, AKO TIETO OPERÁCIE OVPLYVŇUJÚ HODNOTU A TYP OPERANDOV.

PRÍKLAD PROGRAMU:

```
variables_example.py > ...
1  # Example of using variables
2  x = 10                # integer variable
3  y = 5.5               # floating-point variable
4  name = "Alice"        # string variable
5  is_happy = True       # boolean variable
6
7  # Performing operations with variables
8  sum_result = x + y
9  greeting = "Hello, " + name
10 status = is_happy and (x > y)
11
12 print(sum_result)     # Output the result of the addition
13 print(greeting)      # Output the greeting message
14 print(status)         # Output the logical operation result
```

2. PRAKTICKÁ ÚLOHA

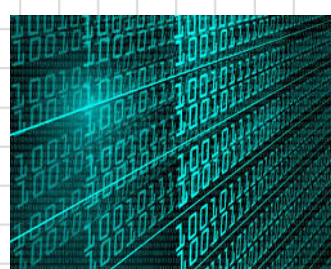
ÚLOHA: VYTVORTE PROGRAM V PYTHONĚ, KTORÝ VYUŽÍVA RÔZNE OPERAČNÉ SIGNÁLY A OPERANDY. NAPRÍKLAD VYKONAJTE SČÍTANIE, DELENIE A LOGICKÉ OPERÁCIE, A NÁSLEDNE VYHODNOTTE VÝRAZY.

3. DOMÁCA ÚLOHA

NAPÍŠTE PROGRAM, KTORÝ VYPOČÍTA PRIEMER DVOCH ČÍSEL A ZOBRAZÍ VÝSLEDOK. NÁSLEDNE VYKONÁ LOGICKÚ KONTROLU NA ZISTENIE, ČI JE PRIEMER VÄČŠÍ AKO STANOVENÁ HODNOTA. ULOŽTE PROGRAM A PRINESTE HO NA ĎALŠIU HODINU, ABY STE HO MOHLI PREZENTOVAŤ V TRIEDE.



PLÁN HODINY



PREDMET: PROGRAMOVANIE A KÓDOVANIE

TÉMA: PODMIENENÉ PRÍKAZY (IF-ELSE) V PROGRAMOVACOM JAZYKU PYTHON – LEKCIA 5

ZAMERANIE A CIELE LEKCIE:

TÁTO LEKCIA OBOZNÁMI ŠTUDENTOV S VYUŽITÍM PODMIENENÝCH PRÍKAZOV V PROGRAMOVACOM JAZYKU PYTHON, VRÁTANE KLÚČOVÝCH SLOV IF, ELIF A ELSE.

NA KONCI HODINY BY ŠTUDENTI MALI BYŤ SCHOPNÍ IMPLEMENTOVAŤ ZÁKLADNÚ PODMIENENÚ LOGIKU VO SVOJICH PROGRAMOCH A ROZUMIEŤ ROZHODOVACIEMU PROCESU V KÓDE.

POTREBNÉ MATERIÁLY:

- POČÍTAČE S NAINŠTALOVANÝM PYTHONOM (NAPRÍKLAD VS CODE, PYCHARM ALEBO INÉ VÝVOJOVÉ PROSTREDIE)
- PROJEKTOR ALEBO PROJEKČNÉ PLÁTNO
- TABUĽA A POPISOVAČE
- PRÍLOHA ZOBRAZUJÚCA PODMIENENÉ INŠTRUKCIE

ÚVOD DO PODMIENENÝCH PRÍKAZOV (15 MINÚT)

DISKUSIA (5 MINÚT): ZAČNITE OBJASNENÍM, PREČO SÚ PODMIENENÉ PRÍKAZY NEVYHNUTNÉ PRI PROGRAMOVANÍ. ROZOBERTE VÝZNAM ROZHODOVANIA A SPÔSOB, AKÝM PROGRAMY VYUŽÍVAJÚ LOGICKÉ VÝRAZY NA URČENIE VYKONÁVANÝCH AKCIÍ.

PREZENTÁCIA (10 MINÚT): NAČRTNITE ŠTRUKTÚRU PODMIENENÝCH PRÍKAZOV V PYTHONE. VYSVETLITE ÚLOHU KLÚČOVÝCH SLOV IF, ELIF A ELSE A PREDVEĎTE PRÍKLADY Z KAŽDODENNÉHO ŽIVOTA, KDE JE POTREBNÉ ROBIŤ ROZHODNUTIA. NÁSLEDNE PREDLOŽTE PRÍKLADY KÓDU PRE TIETO KONŠTRUKCIE V PYTHONE.

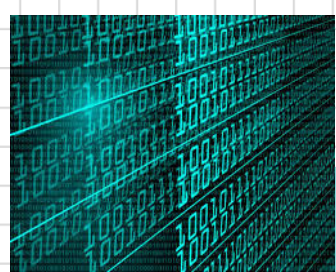
POUŽÍVANIE PODMIENENÝCH PRÍKAZOV V PYTHONE (20 MINÚT)

VYSVETLENIE (10 MINÚT): PREDSTAVTE, AKO JE MOŽNÉ APLIKOVAŤ PODMIENENÉ PRÍKAZY V PYTHONE. OBJAVTE FUNGOVANIE ŠTRUKTÚRY IF-ELSE A SPÔSOB, AKÝM SA LOGICKÉ VÝRAZY VYUŽÍVAJÚ NA OVERENIE PODMIENOK. UVEĎTE PRÍKLADY RÔZNYCH SITUÁCIÍ, AKO JE URČENIE DENNEJ DOBY NA ZÁKLADE ČASU ALEBO ROZHODOVANIE O TOM, ČI NIEKTO ÚSPEŠNE ZLOŽIL SKÚŠKU NA ZÁKLADE DOSIAHNUTÉHO SKÓRE.

PRÍKLAD (10 MINÚT): UVEĎTE JEDNODUCHÝ PRÍKLAD KÓDU NA TABUĽU ALEBO PLÁTNO PROJEKTORA, KTORÝ ILUSTRUJE POUŽITIE ŠTRUKTÚRY IF-ELSE. NAPRÍKLAD PROGRAM, KTORÝ URČUJE, ČI MÁ NIEKTO NÁROK NA ZĽAVU, ALEBO ČI JE ČÍSLO Kladné, záporné, alebo nulové.



PLÁN HODINY



3. PRAKTICKÁ ČINNOSŤ (20 MINÚT)

AKTIVITA (15 MINÚT): POŽIADAJTE ŠTUDENTOV, ABY VYTVORILI JEDNODUCHÝ PROGRAM V PYTHONI, KTORÝ OBSAHUJE ŠTRUKTÚRU IF-ELSE. NAPRÍKLAD, NECH NAPÍŠU PROGRAM, KTORÝ OVERÍ VEK POUŽÍVATEĽA A ROZHODNE, ČI MÁ POUŽÍVATEĽ PRÁVO ZÚČASTNIŤ SA KONKRÉTNEJ UDALOSTI. POMÔŽTE IM S NASTAVENÍM VÝVOJOVÉHO PROSTREDIA A ZABEZPEČTE, ABY KAŽDÝ MOHOL ÚSPEŠNE SPUSTIŤ SVOJ PROGRAM.

PRÁCA VO DVOJICIACH (5 MINÚT): ROZDELTE ŠTUDENTOV DO DVOJÍC, ABY SI NAVZÁJOM SKONTROLOVALI SVOJ KÓD. POVZBUDZUJTE ICH, ABY DISKUTOVALI O AKÝCHKOL'VEK OTÁZKACH ALEBO PROBLÉMOCH, KTORÉ SA OBJAVIA, A VZÁJOMNE SI POMOHLI ZLEPŠIŤ KÓD.

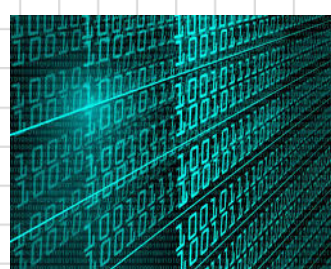
4. OTÁZKY, ODPOVEDE A SÚHRN (5 MINÚT)

OTÁZKY A ODPOVEDE (3 MINÚTY): UMOŽNITE ŠTUDENTOM KLÁŠŤ OTÁZKY. PREBERTE BEŽNÉ PROBLÉMY ALEBO NEJASNOSTI, KTORÉ SA MOHLI POČAS HODINY OBJAVIŤ. POSKYTNITE ŠTUDENTOM PRÍLEŽITOSŤ VYJADRIŤ SPÄTNÚ VÄZBU O TOM, AKO POROZUMELI LEKCII.

ZHRNUTIE A DOMÁCA ÚLOHA (2 MINÚTY): ZHRŇTE KLÚČOVÉ BODY LEKCIE A ZDÔRAZNITE VÝZNAM PODMIENENÝCH PRÍKAZOV. AKO DOMÁCU ÚLOHU POŽIADAJTE ŠTUDENTOV, ABY VYTVORILI PROGRAM V PYTHONI, KTORÝ SIMULUJE JEDNODUCHÝ ROZHODOVACÍ PROCES SO ŠTRUKTÚROU IF-ELSE.



PLÁN HODINY



PREDMET: PROGRAMOVANIE A KÓDOVANIE

TÉMA: PODMIENENÉ PRÍKAZY (IF-ELSE) V PROGRAMOVACOM JAZYKU PYTHON - LEKCIA 5

HLAVNÉ BODY:

PODMIENENÉ PRÍKAZY UMOŽŇUJÚ PROGRAMU PRIJÍMAŤ ROZHODNUTIA NA ZÁKLADE SPLNENIA URČITÝCH PODMIENOK.

KLÚČOVÉ SLOVÁ IF, ELIF A ELSE SA VYUŽÍVAJÚ NA SPRACOVANIE RÔZNYCH SITUÁCIÍ VO VAŠOM KÓDE.

BOOLEOVSKÉ VÝRAZY URČUJÚ, KTORÁ ČASŤ KÓDU SA VYKONÁ.

PRÍKLAD PROGRAMU:

```
if_else_examples.py > ...  
1  # Example of using if-else statements to determine the time of day  
2  hour = int(input("Enter the current hour (0-23): "))  
3  
4  if 5 <= hour < 12:  
5      print("Good morning!")  
6  elif 12 <= hour < 18:  
7      print("Good afternoon!")  
8  elif 18 <= hour < 22:  
9      print("Good evening!")  
10 else:  
11     print("Good night!")
```

2. PRAKTICKÁ ÚLOHA

ÚLOHA: NAPIŠTE PROGRAM V PYTHONI, KTORÝ VYUŽÍVA ŠTRUKTÚRU IF-ELSE NA OVERENIE RÔZNYCH PODMIENOK. ROZHODNITE SA NAPRÍKLAD, ČI JE ČÍSLO PÁRNE ALEBO NEPÁRNE, ALEBO SKONTROLUJTE, ČI MÁ OSOBA NÁROK NA ZĽAVU NA ZÁKLADE VEKU.

3. DOMÁCA ÚLOHA

NAPIŠTE PROGRAM, KTORÝ OVERÍ TEPLotu A URČÍ, ČI JE POČASIE STUDENÉ, TEPLÉ ALEBO HORÚCE. VYUŽITE ŠTRUKTÚRU IF-ELSE NA ROZHODOVANIE A ULOŽTE PROGRAM NA PREZENTÁCIU V NASLEDUJÚCEJ TRIEDE.



01. 01. 2024 –31. 12. 2025



**Funded by
the European Union**



**Spojená škola
Kollárova 17, Sečovce**

**Móricz Zsigmond
Református Kollégium,
Gimnázium, Technikum,
Általános Iskola és Óvoda**



**SOLSKI CENTER
SLOVENJ GRADEC**

2023-2-SK01-KA210-VET-000184496