

ZBIERKA ÚLOH - ZADANIA

Úloha 1 Upravte program **16_automat.py** tak, aby sme mohli zadávať päťfarebný vzor.

Úloha 2 Upravte program **16_automat.py** tak, aby vykreslil vzor osovo súmerný podľa pomyselnej osi x.

Úloha 3 Vytvorte program **16_gravitacna_sila.py**, ktorý vypočíta a do konzoly vypíše veľkosť sily, ktorou sa navzájom priťahujú dve telesá.

Vstupnými údajmi programu sú tri kladné reálne čísla: hmotnosti oboch telies (v kilogramoch) a vzdialenosť telies (v metroch). Vzorec a ďalšie potrebné údaje vyhľadajte na internete. Ošetrite prípadné výnimky, spracujte ich pomocou chybového hlásenia do konzoly.

ZBIERKA ÚLOH – RIEŠENIA

Úloha 4 Upravte program `16_automat.py` tak, aby sme mohli zadávať päťfarebný vzor.

Riešenie:

Program zatiaľ pracuje s dvoma farbami (znak 'x' pre červenú farbu, znak 'o' pre modrú farbu). Definujte nové farby a ich zástupné znaky, napr. žltá = 'z', zelená = 'e' a čierna = 'c'.

Je potrebné vykonať niekoľko zmien vo funkciách programu:

- Vo funkcii `kresli_riadok()` musíme zabezpečiť, aby sa spracovali nové zástupné znaky pre nové farby:

```
def kresli_riadok(riadok, d):
    for znak in riadok:
        if znak == 'x':
            p.dot(d, 'red')
        elif znak == 'o':
            p.dot(d, 'blue')
        elif znak == 'z':
            p.dot(d, 'yellow')
        elif znak == 'e':
            p.dot(d, 'green')
        elif znak == 'c':
            p.dot(d, 'black')
        p.forward(d)
    p.backward(len(riadok) * d)
```

- Vo funkcii `je_ok()` rozšírime množinu prípustných znakov:

```
def je_ok(vzor): # riesenie ulohy 3
    if vzor[len(vzor) - 1] != '/': # ak riadok nekonci znakom /
        return False
    for znak in vzor:
        if not znak in 'nxozec/': # ak znak nie je z povolenej množiny znakov
            return False
    return True
```

- Vo funkcii `pocet_koralikov()` doplníme množinu zástupných znakov (aby sme započítali aj koráliky nových farieb):

```
def pocet_koralikov(vzor): # riesenie ulohy 5
    if not je_ok(vzor):
        raise ValueError('Zadaný vzor nemá požadovaný tvar.')
    pocet = 0
    for znak in 'xozec':
        pocet = pocet + vzor.count(znak)
    return pocet
```

Potom už len stačí zdefinovať nový vzor – aj s novými farbami, a program bude pracovať správne.

Úloha 5 Upravte program **16_automat.py** tak, aby vykreslil vzor osovo súmerný podľa pomyselnej osi x.

Riešenie:

Nová funkcia `os_x()` je takmer kópiou už definovanej funkcie `zrkadlo()` – najprv však musíme upraviť pôvodný vzor: prevrátime ho (aby sme ho mohli vykresľovať od konca). Prevrátením sa však koncový znak `'/'` dostane na začiatok a bude chýbať na konci – upravíme to pomocou výrezu a spojenia reťazcov:

```
def os_x(vzor):
    vzor = vzor[::-1]          # vzor prevratime
    vzor = vzor[1:] + '/'      # zbavime sa znaku / na zaciatku, pridame ho na koniec
    novy = ''
    pocet_riadkov = vzor.count('/')      # zistime pocet riadkov vzoru
    for riadok in range(pocet_riadkov):  # prejdeme postupne vsetky riadky vzoru
        koniec_riadku = vzor.find('/')    # najdeme koniec aktualneho riadku
        riadok = vzor[:koniec_riadku]     # aktualny riadok prepojime s premennou riadok
        novy = novy + riadok[::-1] + '/'  # novy spojime s prevratenym aktualnym riadkom
        vzor = vzor[koniec_riadku + 1:]   # zo retazca odstraníme spracovany riadok
    return novy
```

Ak využijeme to, že funkcie `os_x()` a `zrkadlo()` majú podstatnú časť kódu rovnakú, môžeme upraviť funkciu `os_x()` takto:

```
def os_x(vzor):
    vzor = vzor[::-1]
    vzor = vzor[1:] + '/'
    return zrkadlo(vzor)
```

Úloha 6 Vytvorte program **16_gravitacna_sila.py**, ktorý vypočíta a do konzoly vypíše veľkosť sily, ktorou sa navzájom priťahujú dve telesá.

Vstupnými údajmi programu sú tri kladné reálne čísla: hmotnosti oboch telies (v kilogramoch) a vzdialenosť telies (v metroch). Vzorec a ďalšie potrebné údaje vyhľadajte na internete. Ošetrte prípadné výnimky, spracujte ich pomocou chybového hlásenia do konzoly.

Riešenie:

```
def vypocet_sily(h1, h2, v):
    kapa = 6.67 * 10 ** -11
    sila = kapa * h1 * h2 / v ** 2
    return sila

hmotnost1 = input('Hmotnosť 1. telesa: ')
hmotnost2 = input('Hmotnosť 2. telesa: ')
vzdialenost = input('Vzájomná vzdialenosť telies: ')

try:
    print(f'Telesá sa navzájom priťahujú silou veľkosti {vypocet_sily(hmotnost1, hmotnost2, vzdialenost)} N.')
except ValueError as chyba:
    print(chyba)
```