

Laboratorijska vježba 1: Osnove rada u programskom jeziku Python

Za izradu laboratorijske vježbe treba koristiti odgovarajuću Jupyter Notebook datoteku. Urađenu vježbu je potrebno konvertirati u PDF format, a zatim je PDF datoteku potrebno predati do postavljenog roka koristeći platformu Zamger.

Ime i prezime studenta, broj indeksa:

Amar Hasečić, 2116/18673

Datum izrade izvještaja:

13.03.2024.

Zadatak 1.

Implementirati funkciju `sortiraj` koja kao argument prima Python listu realnih brojeva i sortira je od najvećeg ka najmanjem koristeći algoritam *Bubble Sort*.

Rješenje:

```
def swap(a, b):  
    return b, a  
  
def sortiraj(lista):  
    for i in range(1, len(lista)):  
        for j in range(0, len(lista)-1):  
            if lista[j] > lista[j+1]:  
                lista[j], lista[j+1] = swap(lista[j], lista[j+1])  
  
    return lista
```

Nakon implementacije funkcije, potrebno je biti moguće izvršiti programski kod ispod tako da daje prikazani ispis. Osim toga, potrebno je dodati još primjera kako bi se implementacija adekvatno testirala.

```
niz = [10, 9, -1, 2, 5, 6]
sortirani_niz = sortiraj(niz)
print(", ".join(map(str, sortirani_niz)))

-1, 2, 5, 6, 9, 10

niz = [2, -1, -1, 5, 8, 4, 2, 1]
sortirani_niz = sortiraj(niz)
print(", ".join(map(str, sortirani_niz)))

-1, -1, 1, 2, 2, 4, 5, 8

niz = [-1, -2, -3, -4]
sortirani_niz = sortiraj(niz)
print(", ".join(map(str, sortirani_niz)))

-4, -3, -2, -1
```

Zadatak 2.

Implementirati funkciju `max_na_intervalu(f, a, b, step)` koja kao argument prima neku funkciju f i vraća maksimalnu vrijednost (rezultat poziva funkcije) na segmentu $[a, b]$. Koristiti korak diskretizacije `step` koji se prosljeđuje kao argument funkcije. Za provjeru koristiti dvije funkcije koje su definisane u kodu ispod.

```
def f1(x):
    return x

def f2(x):
    return x-2
```

Rješenje:

```
def max_na_intervalu(f, a, b, step):
    lista = []
    for i in range(a, b+1, step):
        lista.append(f(i))

    return max(lista)
```

Nakon implementacije funkcije, potrebno je biti moguće izvršiti programski kod ispod tako da daje prikazani ispis. Osim toga, potrebno je dodati još primjera kako bi se implementacija adekvatno testirala.

```
maksimum = max_na_intervalu(f1, 0, 10, 1)
```

```
print("Maksimalna vrijednost poziva funkcija f(x) = x na segmentu [0, 10] sa korakom 1 je: " + str(maksimum))
```

```
maksimum = max_na_intervalu(f2, 0, 10, 1)
```

```
print("Maksimalna vrijednost poziva funkcija f(x) = x-2 na segmentu [0, 10] sa korakom 1 je: " + str(maksimum))
```

Maksimalna vrijednost poziva funkcija f(x) = x na segmentu [0, 10] sa korakom 1 je: 10

Maksimalna vrijednost poziva funkcija f(x) = x-2 na segmentu [0, 10] sa korakom 1 je: 8

```
maksimum = max_na_intervalu(f1, 10, 15, 1)
```

```
print("Maksimalna vrijednost poziva funkcija f(x) = x na segmentu [0, 10] sa korakom 1 je: " + str(maksimum))
```

```
maksimum = max_na_intervalu(f2, 18, 24, 1)
```

```
print("Maksimalna vrijednost poziva funkcija f(x) = x-2 na segmentu [0, 10] sa korakom 1 je: " + str(maksimum))
```

Maksimalna vrijednost poziva funkcija f(x) = x na segmentu [0, 10] sa korakom 1 je: 15

Maksimalna vrijednost poziva funkcija f(x) = x-2 na segmentu [0, 10] sa korakom 1 je: 22

#negativan step

```
maksimum = max_na_intervalu(f1, 15, 10, -1)
```

```
print("Maksimalna vrijednost poziva funkcija f(x) = x na segmentu [0, 10] sa korakom 1 je: " + str(maksimum))
```

Maksimalna vrijednost poziva funkcija f(x) = x na segmentu [0, 10] sa korakom 1 je: 15

Zadatak 3.

Definisati skup od 5 slova u Pythonu kao listu (['A', 'B', 'C', 'D', 'E']). Nakon toga, korištenjem random.choice() funkcije u Pythonu implementirati funkciju generiši za generisanje liste od proizvoljno mnogo znakova slučajno odabranih iz skupa od 5 slova. Implementirati funkciju analiziraj koja vraća listu sa frekvencijama pojavljivanja svakog slova u generisanoj listi i ispisuje slovo koje se najviše puta pojavljuje.

Napomena:

Obavezno prvo izvršiti import biblioteke random.

Rješenje:

```

import random
from collections import Counter

slova = ['A', 'B', 'C', 'D', 'E']

def generiši(n):
    lista = []
    for i in range(n):
        lista.append(random.choice(slova))
    return lista

def analiziraj(lista):
    frekvencije = {char: lista.count(char) for char in slova}
    return list(frekvencije.values()), max(frekvencije, key =
    frekvencije.get)

```

Nakon implementacije funkcije, potrebno je biti moguće izvršiti programski kod ispod tako da daje prikazani ispis. Osim toga, potrebno je dodati još primjera kako bi se implementacija adekvatno testirala.

```

lista = generiši(10)

print("Lista generisanih slova je: " + "".join(map(str, lista)))

(frekvencije, najveći) = analiziraj(lista)

for i in range(0, 5):
    print("Karakter " + slova[i] + " se u generisanoj listi ponavlja "
+ str(frekvencije[i]) + " puta.")
    # za prethodni primjer brojevi ponavljanja: (A, B, C, D, E): (1,
4, 3, 1, 1)

print("Najveći broj ponavljanja u listi frekvencija je " +
str(najveći) + ".")

```

```

Lista generisanih slova je: AECADCCDAD
Karakter A se u generisanoj listi ponavlja 3 puta.
Karakter B se u generisanoj listi ponavlja 0 puta.
Karakter C se u generisanoj listi ponavlja 3 puta.
Karakter D se u generisanoj listi ponavlja 3 puta.
Karakter E se u generisanoj listi ponavlja 1 puta.
Najveći broj ponavljanja u listi frekvencija je A.

```

Zadatak 4.

Implementirati Python klasu *Matrica* koja sadržava jedan atribut *elementi* koji čuva elemente matrice. Ovaj atribut predstaviti kao listu listi, npr. `[[1, 2, 3], [4, 5, 6], [7, 8, 9]]`. U konstruktoru

klase postaviti atribut *elementi* na praznu listu ako konstruktor nema argumente ili na listu listi koja se prosleđuje kao argument konstruktora.

Implementirati sljedeće funkcije članice:

- `at(i, j)` Vraća element na poziciji (i,j)
- `zeros(n, m)` Atribut *elementi* popunjava matricom dimenzija n x m koja je popunjena nulama
- `ones(n, m)` Atribut *elementi* popunjava matricom dimenzija n x m koja je popunjena jedinicama
- `eye(n)` Atribut *elementi* popunjava matricom dimenzija n x n koja se sastoji od nula osim na dijagonali (gdje se nalaze jedinice)

Rješenje:

```
class Matrica:
    elementi = []

    def __init__(self, elementi=None):
        if elementi is not None:
            Matrica.elementi = elementi

    @classmethod
    def at(cls, i, j):
        return cls.elementi[i][j]

    @classmethod
    def zeros(cls, n, m):
        cls.elementi = [[0] * n for _ in range(m)]
        return cls()

    @classmethod
    def ones(cls, n, m):
        cls.elementi = [[1] * n for _ in range(m)]
        return cls()

    @classmethod
    def eye(cls, n):
        cls.elementi = [[0] * n for _ in range(n)]
        for i in range(n):
            cls.elementi[i][i] = 1
        return cls()
```

Nakon implementacije funkcije, potrebno je biti moguće izvršiti programski kod ispod tako da daje prikazani ispis. Osim toga, potrebno je dodati još primjera kako bi se implementacija adekvatno testirala.

```
matrica = Matrica()

matrica.zeros(2, 3)
```

```

print("Nakon inicijalizacije nul-matrice 2x3:")
for i in range(0, len(matrica.elementi)):
    print(" ".join(map(str, matrica.elementi[i])))

matrica.ones(3, 3)

print("\nElement na poziciji 1 u matrici je: " + str(matrica.at(0, 0))
+ "\n")

matrica.eye(5)

print("Nakon inicijalizacije jedinične matrice 5x5:")
for i in range(0, len(matrica.elementi)):
    print(" ".join(map(str, matrica.elementi[i])))

Nakon inicijalizacije nul-matrice 2x3:
0 0
0 0
0 0

Element na poziciji 1 u matrici je: 1

Nakon inicijalizacije jedinične matrice 5x5:
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

matrica = Matrica()

matrica.zeros(10, 1)

print("Nakon inicijalizacije nul-matrice 2x3:")
for i in range(0, len(matrica.elementi)):
    print(" ".join(map(str, matrica.elementi[i])))

matrica.ones(1, 1)

print("\nElement na poziciji 1 u matrici je: " + str(matrica.at(0, 0))
+ "\n")

matrica.eye(4)

print("Nakon inicijalizacije jedinične matrice 5x5:")
for i in range(0, len(matrica.elementi)):
    print(" ".join(map(str, matrica.elementi[i])))

Nakon inicijalizacije nul-matrice 2x3:
0 0 0 0 0 0 0 0 0 0

```

Element na poziciji 1 u matrici je: 1

Nakon inicijalizacije jedinične matrice 5x5:

```
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
```