

Praktikum iz programiranja 3



2023/24



O kursu

- **Model 70+30**
 - Redovna izrada domaćih zadataka = 4 poena
 - 2 kolokvijuma = 26 + 40 poena
 - Završni deo ispita = 30 poena
- **Uslov za izlazak na završni ispit – više od polovine poena na predispitnim obavezama**

Najvažnije teme

- Osnove Python-a – podsećanje
 - Aritmetičke operacije, upravljačke komande, standardni ulaz/izlaz, ugrađene funkcije, stringovi, liste, funkcije
- Strukture - liste, skupovi, torke, rečnici, manipulacija podacima
- Biblioteke
- Standardni algoritamski problemi

Python

Promenljive

- Ime promenljive sme sadržati samo
 - Brojeve
 - Velika i mala slova
 - Donju crticu _
- Ime ne sme početi brojem
- Python razlikuje velika i mala slova

Neka od dozvoljenih imena:

`jabuka, slova123, A4, ime_psa`

Neka od nedozvoljenih imena:

`ime psa, 123slova, jabuka#`

Python

Aritmetičke operacije

- Python kao kalkulator

Znak	Operacija
+	Sabiranje
-	Oduzimanje
*	Množenje
/	Deljenje
%	Ostatak pri deljenju
//	Celobrojno deljenje
**	Stepenovanje

```
>>> 2 + 3
5
>>> 2 - 3
-1
>>> 2 * 3
6
>>> 9 / 4
2.25
```

```
>>> 2 + 3 * 6
20
>>> 2 + 4 * 9 - 3
35
>>> 27 // 10
2
>>> 27 / 10
2.7
>>> 27 % 10
7
```

Python

Tipovi i vrednosti

- Dodelom vrednosti se određuje tip neke promenljive. Ukoliko nismo sigurni kog je tipa neki podatak, interpreter nam to može reći

```
>>> poruka = "Programiranje je lepo"
```

```
>>> pi = 3.14
```

```
>>> n = 5
```

```
>>> poruka
```

```
'Programiranje je lepo'
```

```
>>> pi
```

```
3.14
```

```
>>> n
```

```
5
```

```
>>> type(poruka)
```

```
<class 'str'>
```

```
>>> type(pi)
```

```
<class 'float'>
```

```
>>> type(n)
```

```
<class 'int'>
```

tip	oznaka	domen vrednosti	primeri vrednosti
celobrojni	int	celi brojevi	0, 1, -1243
realni	float	realni brojevi	1.25, .02, -145.25
kompleksi	comlex	kompleksi brojevi	-3j, -0.5 – 4.25j
logički	bool	{ False, True }	False, True
tekstualni	str	niz znakova	'Ucim programiranje'
torka	tuple	nepromenljiv niz proizvoljnih objekata	(), (1.25, True, 'abc')
lista	list	promenljiv niz proizvoljnih objekata	[], [1, 2], ['A', 1, 1 + 2j]
skup	set	neuređen skup proizvoljnih nepromenljivih objekata. Duplikati nisu dozvoljeni	{}, {1, 2}, {'A', 1, {1, 2}}
rečnik	dict	parovi ključ-vrednost	{11: 'Beograd', 21: 'Novi Sad', 34: 'Kragujevac'}
nedefinisano	NoneType	{ None }	None

Python

Logičke operacije

Znak	Relacija
And	Logičko „i“
Or	Logičko „ili“
Not	Logičko „ne“

x		y		x and y		x or y
True		True		True		True
True		False		False		True
False		True		False		True
False		False		False		False

- Tabela istinitosti logičkih operacija **and** i **or**

```
print(' x      | y      | x and y | x or y')
print('True   | True   |', True and True, '   |', True or True)
print('True   | False  |', True and False, '  |', True or False)
print('False  | True   |', False and True, '  |', False or True)
print('False  | False  |', False and False, ' |', False or False)
```


Python

Operatori

Znak	Relacija
>	Veće
<	Manje
>=	Veće i jednako
<=	Manje i jednako
==	Jednako
!=	Različito

- Ispisati veći od uneta dva realna broja

```
x = float(input('x= '))  
y = float(input('y= '))  
print('veći broj je', x * (y < x) + y * (x <= y) )
```

Python

Grananje

- Neke naredbe se izvršavaju samo ako je neki uslov ispunjen. Da bi se opisalo uslovno izvršavanje nekih naredbi koristi se naredba `if`

```
if uslov:  
    naredba_1  
    ...  
    naredba_k
```

```
if uslov:  
    naredba_1  
    ...  
    naredba_m  
else:  
    naredba_1  
    ...  
    naredba_n
```

```
if uslov1:  
    naredba_1  
    ...  
    naredba_m  
elif uslov2:  
    naredba_1  
    ...  
    naredba_n  
else:  
    naredba_1  
    ...  
    naredba_k
```

- Zgrade u uslovu nisu OBAVEZNE.
- Dvotačka je obavezna nakon navođenja uslova
- Naredba u uslovu će vratiti tačno/netačno
- Ako je tačan uslov koji se zadaje izvršiće se naredba_1 ...

Python

Ponavljjanje

- Python pruža mehanizam da se jedna ili više naredbi ponovi:

- određen broj puta

```
for brojač in opseg :  
    telo_petlje
```

```
for i in range(a,b,k) :  
    naredba 1  
    naredba 2
```

početna vrednost

završna vrednost koja se ne uzima u obzir

korak

- dok je neki uslov ispunjen

```
while uslov :  
    telo_petlje
```

Python

Standardni ulaz/izlaz

- Upisivanje tekstualne ili brojčane vrednosti je moguće pomoću naredbe `input()`.
- Rezultat izvršenja naredbe `input()` mora se pridružiti nekoj promenljivoj.
- Tip prihvaćenih podataka je uvek `string`, ukoliko se ne zahteva drugačije.

`a=tip_podatka(input("Unesi vrednost"))`

naziv promenljive

tip promenljive: int, float, bool ili string

naredba za unos

tekstualni deo kao poruka, ne mora da postoji

- Funkcija za ispis je `print()`:

```
print("Ovo je tekst ", "a ovo je broj ", 30)
```

Python

Ugrađene funkcije

- min, max, abs

```
print(min(2.3,5.8),max(8,12))
```

```
broj = -3
```

```
print(max (abs (broj), min (2, 4)))
```

- Matematičke funkcije

```
import math as m
```

```
print(m.factorial(4))
```

pow, sqrt, floor, trunc, exp, log, sin, cos,...

Python

Stringovi

- String je niz karaktera ograničen jednostrukim ili dvostrukim navodnicima, ali mora početi i završiti se istim navodnicima

```
s = 'On je rekao, "Zdravo, svete!"'
```

- Specijalni karakteri

- `\n` – završava trenutnu liniju i nastavlja u sledećoj
- `\t` – ubacuje tab u string
- `\'` – ubacuje `'` u string
- `\"` – ubacuje `"` u string
- `\\` – ubacuje `\` u string

- String se može štampati u više linija

- Ako se započne sa tri navodnika (jednostruka ili dvostruka) i tako i završi
- Ako se u string umetne specijalni karakter `\n`

```
s1 = """Ovo  
je string koji ima  
više linija."""
```

```
s2 = 'Ovo\ntakodje'
```

Python

Stringovi

- Dva stringa se spajaju u jedan korišćenjem operatora +
- Više stringova se spajaju u jedan, na isti način, korišćenjem operatora +

```
s0 = "Spajanje"
```

```
s1 = "više"
```

```
s3 = "zajedno"
```

```
s = s0 + " " + s1 + " " + s2 + " " + s3
```

- Množenje stringa nulom i negativnim brojem ne daje nikakav rezultat
- Nije dozvoljeno stringa množenje razlomljenim brojem i sabiranje sa celim brojem

```
s = 'Ha'
```

```
print (s * 10)
```

```
HaHaHaHaHaHaHaHaHaHa
```

```
a = 'Program'
```

```
ProgramProgramProgram
```

```
print(a * 3)
```

Python

Deljenje stringova

- Delovima stringa može se pristupiti preko indeksa navedenog u zagradama []

```
s = 'programiranje'
print ('s = ', s)
print ('s[0] = ', s[0], ' s[3] = ', s[3])
print ('s[-1] = ', s[-1])
print ('s[1:5] = ', s[1:5])
print ('s[5:-2] = ', s[5:-2])
```

```
s = programiranje
s[0] = p  s[3] = g
s[-1] = e
s[1:5] = rogr
s[5:-2] = amiran
```


Python

String funkcije

- Dužina stringa – `len`
- String u stringu – `find`, `index`
- Broj ponavljanja stringa u stringu – `count`
- Konvertovanje svih slova u velika, odnosno mala – `upper`, `lower`
- Promena dela stringa novim stringom – `replace`
- Za određivanje tipa podataka koristi se funkcija `type`
- Konverzija u string – `str`
- Konverzija u ceo broj – `int`
- Konverzija u realan broj – `float`

Python

Liste

- Lista je niz podataka proizvoljnog tipa. Svaki podataka u listi se naziva **element**

`[1, 4, 17, 256, 3, 59, 45]` - lista celih brojeva

`['april', 'jun', 'septembar', 'novembar']` - lista stringova

`[]` - prazna lista

`['Marina', 28, 101, 'PMF']` - lista elemenata različitog tipa

`[1, 4, [7, 6, [13]], [9, 5]]` - lista čiji elementi su liste

- Elementim liste se pristupa preko rednog broja pozicije – **indeksa**
- Indeks prvog elementa u listi je **0**

Python

Liste

- Spajanje listi

```
A = [1, 2, 3, 4, 5]
```

```
B = [6, 7, 8, 9]
```

```
C = A + B
```

- Direktna promena vrednosti elemenata liste

```
lista = [2, 4, 6, 8]
```

```
lista[1] = 0
```

```
lista[-1] = 12
```

Python

Liste

- Elementi liste se mogu ponavljati određen broj puta operatorom *

```
C = [0]
A = [1,2,3]
B = 3 * A
D = 4 * C
print(B, D)
```

```
[1, 2, 3, 1, 2, 3, 1, 2, 3] [0, 0, 0, 0]
```

Python

Izdvajanje elemenata liste

```
lista = [10, 9, 8, 7, -1, -2, -3]
print (lista[2], lista[-1])
print (lista[1:5])
print (lista[1:5:2])
print (lista[12])
```

8 -3

[9, 8, 7, -1]

[9, 7]

IndexError: list index out of range

Python

Funkcije za rad sa listama

- Dužina liste, odnosno broj elemenata liste – `len`
- Dodavanje elemenata na kraj liste – `append`, `extend`
- Dodavanje elemenata na željenu poziciju – `insert`
- Minimalni i maksimalni element liste – `min`, `max`
- Broj ponavljanja elementa – `count`
- Brisanje elementa iz liste – `del`, `pop`, `remove`
- Sortiranje i okretanje elemenata liste – `sort`, `reverse`

Python

Filtriranje listi

```
temperature = [2, -1, 0, -8, -10, -1, 4, 5, 8, 6]
pozitivne_temperature = [t for t in temperature if t > 0]
print(pozitivne_temperature)
[2, 4, 5, 8, 6]
```

- Zadati tekst ispisati tako što bi se izostavila pojava samoglasnika

```
tekst = "Praktikum iz programiranja"
novi = [a for a in tekst if a not in {'a', 'e', 'i', 'o', 'u'}]
for slovo in novi:
    print(slovo, end='')
```

Python

Funkcije

- Python pruža mogućnost definisanja korisničkih funkcija

```
def ime_funkcije(parametri_funkcije) :  
    _____telo funkcije
```

- Funkcija **mora** biti definisana pre prvog korišćenja (poziva)

```
def stampaj_izjavu():  
    print ("Moje ime je Ana i OK sam.")  
    print ("Spavam celu noc i radim ceo dan.")
```

```
stampaj_izjavu()
```

```
print (stampaj_izjavu)  
print (type(stampaj_izjavu))
```

```
Moje ime je Ana i OK sam.  
Spavam celu noc i radim ceo dan.  
<function stampaj_izjavu at 0x0341FE40>  
<class 'function'>.
```


Python

Funkcije sa parametrima

```
def stampaj_2puta(param):  
    print(param)  
    print(param)
```

```
stampaj_2puta('Spam')  
stampaj_2puta(math.pi)  
stampaj_2puta('Spam'*4)  
stampaj_2puta(17)
```

```
Spam  
Spam  
3.141592653589793  
3.141592653589793  
SpamSpamSpamSpam  
SpamSpamSpamSpam  
17  
17
```

Python

Funkcije

- Funkcija može da ima jednu ili više povratnih vrednosti
- Rezultat izvršenja funkcija koje imaju povratne vrednosti mora biti dodeljen promenljivima na mestu poziva

```
def saberi (a,b):  
    c = a + b  
    return c
```

```
def deli (a,b):  
    c = a // b  
    d = a % b  
    return (c, d)
```