

Praktikum iz programiranja 1



2024/25



Greške u kodu

- Bitan deo programiranja čini ispravno tumačenje poruka koje Python interpreter pruža usled nemogućnosti izvršavanja datog koda. Postoji nekoliko vrsta grešaka koje se mogu pojaviti u Python-u. Svaka vrsta ukazuje na različitu vrstu problema u kodu, a razumevanje ovih tipova grešaka je ključno za kreiranje efikasnih Python programa. Greške u Python-u mogu biti:
 - sintaksne greške (Syntax Errors),
 - greške u vremenu izvršavanja (Runtime Errors),
 - logičke greške (Logical Errors).

Sintaksna greška

Nedovršen aritmetički izraz:

```
>>> 3+2*  
File "<stdin>", line 1  
    3+2*  
      ^
```

SyntaxError: invalid syntax

Uvlačanje u okviru linijskih komandi dovodi do greške:

```
>>> print('Програмирање')  
File "<stdin>", line 1  
    print('Програмирање')  
IndentationError: unexpected indent
```

Nedovršena dodela vrednosti:

```
>>> a='danas  
File "<stdin>", line 1  
    a='danas  
      ^
```

SyntaxError: unterminated string literal (detected at line 1)

Greška u vremenu izvršavanja

NameError:

```
# Покушај коришћења променљиве пре него што је дефинисана
```

```
print(b) # Грешка: b није дефинисана
```

```
# Исправка: дефинисање променљиве пре употребе
```

```
b = 10
```

```
print(b) # Исписује: 10
```

Poruka:

```
>>> Traceback (most recent call last):
```

```
File "D:/XXX/Zadatak.py", line 2, in <module>
```

```
print(b) # Грешка: b није дефинисана
```

```
NameError: name 'b' is not defined
```

```
b, c = 10, 5
```

```
a = maksimum(b, c)
```

Poruka:

```
NameError: name 'maksimum' is not defined
```

Greška u vremenu izvršavanja

TypeError :

```
x = "10"  
y = 5  
z = x + y  
print(z)
```

Poruka:

```
>>> Traceback (most recent call last):  
  File "D:/XXX/Zadatak.py", line 3, in <module>  
    z = x + y  
TypeError: can only concatenate str (not "int") to str
```

```
e1 = int("123.56")- ValueError: invalid literal for int() with base 10: '123.56'  
e2 = float("123.b")- ValueError: could not convert string to float: '123.b'  
e3 = complex("2+3a")- ValueError: complex() arg is a malformed string  
e4 = float(1+2j)- TypeError: can't convert complex to float  
e5 = int(1+2j)- TypeError: can't convert complex to int  
e6 = eval("2 + 'text'")- TypeError: unsupported operand type(s) for +: 'int' and  
'str'
```

Logička greška

- Logička greška nastaje kada se kod izvršava bez sintaksnih ili grešaka u vremenu izvršavanja, ali proizvodi netačne rezultate zbog pogrešne logike u kodu.
- Ove vrste grešaka su često uzrokovane netačnim pretpostavkama, nepotpunim razumevanjem problema ili nepravilnom upotrebom algoritama ili formula.
- Za razliku od sintaksnih ili grešaka u vremenu izvršavanja, logičke greške mogu biti teške za otkrivanje i ispravljanje jer se kod izvršava bez prikazivanja poruka o grešci.

```
duzina = 5
sirina = 3
# Нетачна формула за обим
obim = duzina * sirina # Грешка
print(obim) # Исписује: 15 (уместо очекиваних 16)
```

Grananje



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
```

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

Grananje

- Šta je rezultat datog koda?

```
a=0  
if a:  
    print(a)
```

```
a=0  
if a==0:  
    print(a)
```

Uslovno i alternativno izvršavanje

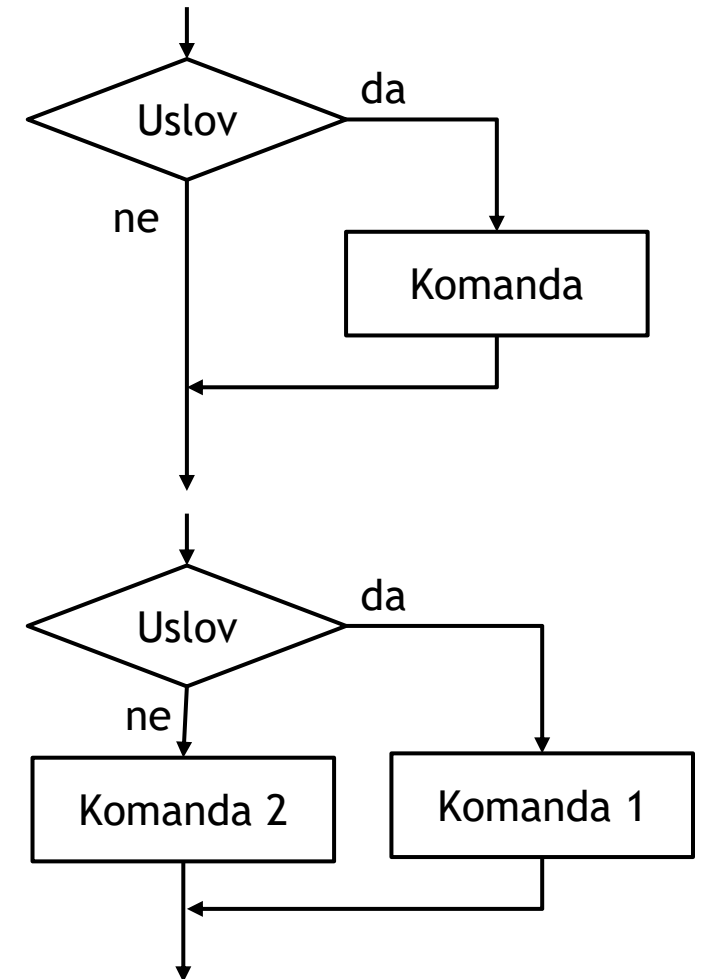
- Za uneti broj ispisati da li je manji od 10

```
a = int(input("Unesi broj "))
```

```
if a < 10:  
    print ('Broj je manji od 10')
```

- Za uneti broj ispisati da li je paran ili neparan

```
if a%2 == 0:  
    print ('Broj je paran')  
else:  
    print ('Broj je neparan')
```



Blok komandi

- Za unete dužine stranica pravougaonika ili kvadrata (ukoliko su unete vrednosti jednake) izračunati površinu.

```
a=int(input("Unesite stranicu a: "))  
b=int(input("Unesite stranicu b: "))
```

```
if a==b:
```

```
    p=a**2  
    print("Povrsina kvadrata iznosi: ", p)
```

```
else:
```

```
    p=a*b  
    print("Povrsina pravougaonika iznosi: ", p)
```

Više od
jedne
naredbe

Primer

- Napisati program koji za uneto x izračunava vrednosti funkcija $f(x)$ i $g(x)$, koje su date formulama

$$f(x) = \begin{cases} -1, & \text{ako je } x < 0 \\ 2x + 3, & \text{ako je } x \geq 0 \end{cases}$$

i

$$g(x) = \begin{cases} \ln(-x), & \text{ako je } x < 0 \\ e^x, & \text{ako je } x \geq 0 \end{cases}$$

Primer

```
import math
x=float(input('Unesi x '))
if x<0:
    f=-1.0
    g=math.log(-x)
else:
    f=2*x+3
    g=math.exp(x)
print(f,g)
```

Primer

- Ako se od odsečaka sa dužinama x , y , z može konstruisati trougao, izračunati njegovu površinu po formuli:

$$P = \sqrt{s * (s - x) * (s - y) * (s - z)}$$

Ugnježdjeni uslovi

- Odrediti odnos dva uneta broja

```
x = int(input("Unesi broj x "))
y = int(input("Unesi broj y "))

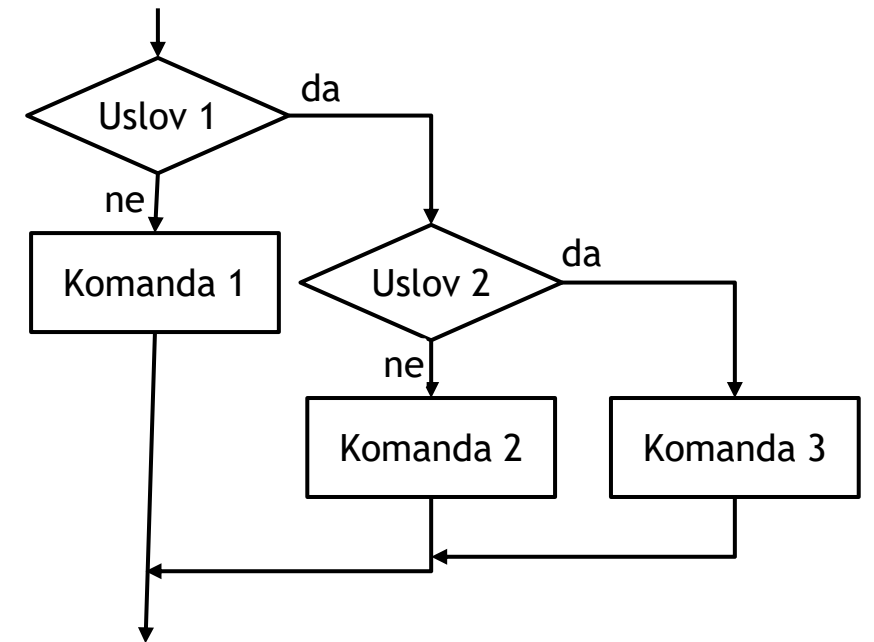
if x == y:
    print ('x i y su jednaki')
else:
    if x < y:
        print ('x je manje od y')
    else:
        print ('x je vece od y')
```

Lančani uslovi

- Odrediti odnos dva uneta broja

```
x = int(input("Unesi broj x "))
y = int(input("Unesi broj y "))

if x < y:
    print ('x je manje od y')
elif x > y:
    print ('x je vece od y')
else:
    print ('x i y su jednaki')
```



Primer

- Unetu operaciju primeniti na zadate vrednosti promenljivih **a** i **b**

```
a = 25
b = 5
c = input('uneti operaciju: ')
if (c == '+'):
    print (a + b)
elif (c == '*'):
    print (a * b)
elif (c == '-'):
    print (a - b)
elif (c == '/'):
    print (a / b)
```

Primer

- Napisati program koji na osnovu broja bodova osvojenih na ispitu, određuje ocenu prema sledećoj tabeli

Broj bodova	Ocena
51 - 60	6
61 - 70	7
71 - 80	8
81 - 90	9
91 - 100	10

Primer

```
bodovi = float(input('Unesi broj bodova '))
if bodovi >= 51:
    if bodovi >= 61:
        if bodovi >= 71:
            if bodovi >= 81:
                if bodovi >= 91:
                    ocena = 10
                else:
                    ocena = 9
            else:
                ocena = 8
        else:
            ocena = 7
    else:
        ocena = 6
else:
    ocena = 5
print(ocena)
```

|||

```
bodovi = float(input('Bodovi '))
if bodovi >= 91:
    ocena = 10
else:
    if bodovi >= 81:
        ocena = 9
    else:
        if bodovi >= 71:
            ocena = 8
        else:
            if bodovi >= 61:
                ocena = 7
            else:
                if bodovi >= 51:
                    ocena = 6
                else:
                    ocena = 5
print(ocena)
```

Vezivanje više uslova logičkim operatorima

Kako bi napisali date uslove:

- $1 < x < 2$
- $x = y = z$
- X ne pripada interval $-a$ do a

Vezivanje više uslova logičkim operatorima

Stanari jedne zgrade treba da renoviraju jedan sprat i radnicima su dali svoje uslove kada smeju da buše i grade:

- rekli su im da mogu da rade od 9 sati izjutra ali
- da moraju da naprave pauzu u vreme odmora od 14-17
- ne smeju posle 22 časa da buše

```
sati = int(input('Koliko je sati: '))
if ((sati>=9 and sati<14) or (sati>=17 and sati <22)):
    print("moze da se radi")
else:
    print("ne moze da se radi")
```