

Sveučilište u Zagrebu
Fakultet strojarstva i brodogradnje

Vježbe iz kolegija Objektno programiranje

Operacije s listama, rječnik, formatiranje ispisa,
upravljanje programom

Mario Essert, Tihomir Žilić, Vladimir Milić, Jakov Topić

Zagreb, 2020./2021.

1. Operacije s listama

1.1. ZADATAK Kreirajte liste `lista1` i `lista2`.

```
>>> print(lista1, lista2)
['jedan', 'dva', 'tri'] ['sest', 'sedam', 8, 9]
```

1.2. ZADATAK Na kraj liste `lista1` dodajte element 4.

```
>>> print(lista1)
['jedan', 'dva', 'tri', 4]
```

1.3. ZADATAK Na početak liste `lista2` dodajte element 'pet'.

```
>>> print(lista2)
['pet', 'sest', 'sedam', 8, 9]
```

1.4. ZADATAK Iz liste `lista2` izbrišite element 8.

```
>>> print(lista2)
['pet', 'sest', 'sedam', 9]
```

1.5. ZADATAK Primjenom *slicing*-a izdvojite iz liste `lista1` element 'dva' i spremite ga u varijablu `x`.

```
>>> print(x)
['dva']
```

1.6. ZADATAK Kreirajte listu `lista3` tako da u listi `lista1` prva tri elementa zamijenite podlistom `[1, 2, 3]`. Iz `lista3` izdvojite element 3 i spremite ga u varijablu `y`.

```
>>> print(lista3)
[[1, 2, 3], 4]
>>> print(y)
3
```

1.7. ZADATAK Kreirajte listu `niz1`.

```
>>> print("Neparni brojevi do 10: ", niz1)
Neparni brojevi do 10: [1, 3, 5, 7, 9]
```

1.8. ZADATAK Koristeći sažetu `for` petlju, generirajte liste `niz2` i `niz3` iz liste `niz1`.

```
>>> print("Parni brojevi do 10: ", niz2)
```

```
Parni brojevi do 10: [2, 4, 6, 8, 10]
>>> print("Neparni brojevi od 10 do 20: ", niz3)
Neparni brojevi od 10 do 20: [11, 13, 15, 17, 19]
```

1.9. ZADATAK Generirajte listu niz iz liste niz1 i niz2.

```
>>> print("Sortirani niz", niz)
Sortirani niz [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

1.10. ZADATAK Koristeći sažetu for petlju, pomoću liste niz ispišite sljedeće:

```
Desetice do 70: [10, 20, 30, 40, 50, 60, 70]
```

2. Rječnik

2.1. ZADATAK Kontakti osoba predstavljeni su kao rječnik u rječniku tako da je ime ključ (*key*), a adresa stanovanja i telefon su vrijednosti (*values*):

```
kontakti={
    'Ivo Ivic':{
        'tel': '332-721',
        'adr': 'Slavka Jankovica 4'
    },
    'Pero Peric':{
        'tel':'227-321',
        'adr':'Kralja Zvonimira 21'
    },
    'Mia Mijic':{
        'tel':'309-299',
        'adr':'Vijenac Murse 3'
    }
}

oznake={
    'tel':'telefon',
    'adr':'adresa'
}
```

Napišite program koji će na zahtjev korisnika ispisivati telefon ili adresu osobe u kontakti.

Uputa: Definirajte varijablu ime kojoj ćete pridružiti željenu osobu, te varijablu zahtjev kojoj ćete pridružiti 'A' ili 'T' ovisno da li tražite adresu stanovanja ili telefon. Na primjer, ako su ime='Mia Mijic' i zahtjev='T' tada se ispisuje poruka: telefon od Mia Mijic je 309-299. Također, uzmite u obzir i slučaj ako se tražena osoba ne nalazi u kontakti.

3. Formatiranje stringa i ispisa

3.1. ZADATAK Napišite naredbu kojom ćete na zaslonu računala dobiti poruku:

```
>>>
Hello Objektno.
```

3.2. ZADATAK Definirajte varijablu `x="Hello Objektno programiranje"`. Koristeći varijablu `x`, ispišite na zaslon:

```
>>>
Vrijednost od x je Hello Objektno programiranje
```

3.3. ZADATAK Formatirajte string koristeći varijablu `x` i metodu `format()`, tako da ispis na zaslon bude u više linija:

```
>>>
Vrijednost od x
je Hello Objektno programiranje
```

3.4. ZADATAK Formatirajte string koristeći varijablu `x` i formatiranje `f""` (od ver.3.6), tako da ispis na zaslon bude u više linija:

```
>>>
Vrijednost od x
je Hello Objektno programiranje
```

3.5. ZADATAK Definirajte varijable `i=123`, `j=99`, `y=123.456789123`, `z=1234567.5`. Koristeći navedene varijable i formatiranje pomoću `%`, ispišite sljedeće poruke u različitim formatima ispisa.

Napomena: `%g` formatira ispis koristeći eksponent, ako je broj veći od 10^6 ili manji od 10^{-4} :

```
>>>
Vrijednosti su i,j,y,z: 123 99 123.45678912345 1234567.5
>>>
d - cijeli broj i: 123
>>>
f - broj klizeceg zareza y: 123.456789
>>>
f - broj klizeceg zareza konacnog formata na 2 decimale y: 123.46
>>>
f - broj klizeceg zareza konacnog formata na 4 decimale y: 123.4568
>>>
g - realni broj u eksponencijalnom zapisu z: 1.23457e+06
>>>
g - realni broj u eksp. zapisu u konacnom formatu z: 1.23e+06
>>>
s - Vrijednost mog stringa je: Hello Objektno programiranje
>>>
int s 4 znaka razmaka: 123 99
```

```
>>>
int s 4 znaka razmaka: 99 123
```

4. Upravljanje programom (petlje, grananje)

4.1. ZADATAK Primjenom `for` petlje odredite prvi broj koji je manji od 100 i ima cijeli korijen. Uputa: Koristite ugrađene funkcije `range`, `sqrt`, `int`, te za izlazak iz petlje `break`. Ispišite poruku:

```
>>>
Prvi broj koji je manji od 100 i ima cijeli korijen je: 81
```

4.2. ZADATAK Pomoću `for` petlje (u punom i sažetom zapisu) napišite program kojim ćete dobiti "tablicu" u kojoj će prvi stupac biti mjerene temperature u Celsiousovim stupnjevima, a drugi stupac njihove vrijednosti u Farenheitovim stupnjevima. Pretvaranje Celsiousovih u Farenheitove stupnjeve provodi se prema formuli $F = 9 \cdot C / 5 + 32$. Formatirajte ispis (pomoć: koristite `print('%5.1f %5.1f' % ...)`) tako da rješenje dobijete u sljedećem obliku :

```
>>>
-5.0 23.0
-4.5 23.9
-4.0 24.8
-3.5 25.7
-3.0 26.6
-2.5 27.5
-2.0 28.4
-1.5 29.3
-1.0 30.2
-0.5 31.1
0.0 32.0
0.5 32.9
1.0 33.8
1.5 34.7
2.0 35.6
2.5 36.5
```

Uputa: Krenite sa sljedećim kodom:

```
C_start = -5    #pocetna temperatura
C_step = 0.5    #korak
n = 16         #koliko temperatura
Cdegrees = [0.0]*n
Fdegrees = [0.0]*n
```

4.3. ZADATAK Pomoću `while` petlje kreirajte tablicu za problem kao u 4.2., ako su temperature u Celsiousovim stupnjevima -20, -15, -10, -5, ..., 40. Koristite ispis bez formatiranja.