Praktikum iz programiranja 3

2023/24

Python
Rečnici

- Rečnici su generalizovana verzija liste. Posmatrajmo listu koja sadrži broj dana u mesecima godine

```
dana = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
```

- Ako hoćemo broj dana u mesecu januaru koristimo `dana[0]`, ili u decembru `dana[11]` ili `dana[-1]`
- Rečnik dana u mesecima godine:

```
dani ={'Januar':31, 'Februar':28, 'Mart':31, 'April':30,
       'Maj':31, 'Jun':30, 'Jul':31, 'Avgust':31,
       'Septembar':30, 'Oktobar':31, 'Novembar':30, 'Decembar':31}
```

- Sada pristupamo sa:

```
dani['Januar']
```

- Prednost: kod je čitljiv i ne treba razmišljati o indeksu

Python
Osnovne operacije sa rečnicima

- Da označimo da je nešto rečnik koristimo zagrade `{}`
- Svaki element u rečniku je par podataka razvojen sa dve tačke
- Prvi deo u paru se naziva `ključ (key)`, drugi je `vrednost (value)`
- Ključ igra ulogu sličnu indeksu

```
d = {'A':100, 'B':200}
```

ključ

vrednost

```
d['A'] = 100
```

- Ključevi su često stringovi, mogu biti i celi i realni brojevi
- U istom rečniku mogu se naći ključevi različitog tipa

Python
Osnovne operacije sa rečnicima

```
d = {'A':100, 'B':200}
```

- Promena vrednosti:

```
d['A'] = 400
```

- Dodavanje novog elementa u rečnik:

```
d['C'] = 500
```

Ovo nije moguće sa listama. Ako napišem L[2]=500 za listu L koja ima samo dva elementa L[0] i L[1], dobićemo grešku: „index out of range“

- Brisanje elementa iz rečnika:

```
del d['A']
```

Python
Osnovne operacije sa rečnicima

- Prazan rečnik je `{}`, čemu je analogno `[]` za praznu listu, ili `''` za prazan string.
- Redosled elemenata u rečniku ne mora biti isti kao redosled dodavanja elemenata u rečnik
- Python rearanžira elemente rečnika u cilju optimizacije performansi.

Python
Osnovne operacije sa rečnicima

- U igri Scrabble, svakom slovu je pridružena vrednost. Možemo koristiti sledeći rečnik za vrednost slova u njemu:

```
points = {'A':1, 'B':3, 'C':3, 'D':2, 'E':1, 'F':4, 'G':2,
          'H':4, 'I':1, 'J':8, 'K':5, 'L':1, 'M':3, 'N':1,
          'O':1, 'P':3, 'Q':10, 'R':1, 'S':1, 'T':1, 'U':1,
          'V':4, 'W':4, 'X':8, 'Y':4, 'Z':10}
```

- Da izračunamo skor za neku reč koristimo kod

```
skor = sum([points[c] for c in word])
```

Python
Osnovne operacije sa rečnicima

Rečnik omogućava lepu varijantu za prikaz špila karata:

```
špil = [{'vrednost':i, 'boja':c}
        for c in ['pik', 'tref', 'herc', 'karo']
        for i in range(2,15)]
```

Špil je lista sa 52 rečnika. Metod `shuffle` se može iskoristiti za mešanje špila:

```
shuffle(špil)   //  from random import shuffle
```

Prva karta u špilu je `špil[0]`. Boja i vrednost karte:

```
špil[0]['vrednost']
špil[0]['boja']
```

Python
Rad sa rečnicima

```
d = {'A':100, 'B':200}
```

- Kopiranje rečnika:

```
d2=d.copy()
```

- Operator `in` se koristi da nam kaže da li je neki ključ u rečniku ili nije.
- Ako pokušamo da dobijemo vrednost nekog ključa koji nije u rečniku dobićemo grešku.
- `print(d['C'])  - greška`

- Možemo da sprečimo grešku korišćenjem operatora `in`. Pre upotrebe nekog ključa, možemo da proverimo da li je u rečniku.

Python
Rad sa rečnicima

```
slovo = input('Unesite slovo: ')
if slovo in d:
    print('Vrednost je', d[slovo])
else:
    print('Nije u rečniku')
```

Može se koristiti i `not in`

Python
Rad sa rečnicima

```
tabela = {1 : 'Srbija', 2 : 'Italija', 3 : 'Francuska', 4 : 'Spanija’}
```

Metoda `pop()` briše pridruživanje za zadati ključ i tom prilikom vraća vrednost koja se izbacuje iz rečnika.

```
print(tabela.pop(2), tabela)
Italija {1: 'Srbija', 3: 'Francuska', 4: 'Spanija’}
tabela.pop(5)
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    tabela.pop(5)
KeyError: 5
```

Python
Rad sa rečnicima

```
tabela = {1 : 'Srbija', 2 : 'Italija', 3 : 'Francuska', 4 : 'Spanija’}
```

- Kada se u metodi navede opcioni parameter greška se ne pojavljuje:

```
tabela.pop(5, None)
None
```

- Brisanje svih elemenata iz rečnika:

```
tabela.clear()
print(tabela)
{}
```

Python
Rad sa rečnicima

```
d = {'A':100, 'B':200}
```

- Može se napraviti petlja kroz rečnik na sličan način kao i kod liste.
- Štampanje svih ključeva rečnika:

```
for kljuc in d:
    print(kljuc)
```

- Štampanje vrednosti iz rečnika:

```
for kljuc in d:
    print(d[kljuc])
```

Python
Rad sa rečnicima

```
d = {'A':100, 'B':200}
```

- Kako od rečnika dobiti liste ključeva i vrednosti:

```
list(d)   -  ['A','B'], ključevi iz rečnika d
list(d.values())  -  [100,200], vrednosti iz rečnika d
list(d.items()) -  [('A',100),('B',200)], parovi (ključ,vrednost)
```

- Parovi koje vraća `d.items()` nazivaju se `torke (tuples)`. Torke su slične listama.

Python
Rad sa rečnicima

- Svi ključevi rečnika kojima odgovara vrednost 100.

```
d = {'A':100, 'B':200, 'C':100}
L = [x[0] for x in d.items() if x[1]==100] #filtriranje liste ključeva
```

```
['A', 'C']
```

Python
Rad sa rečnicima

- Funkcija `dict` je dodatan način kreiranja rečnika.

```
d = dict([('A',100),('B',300)])
{'A':100, 'B':300}
```

- `Sastavljanje rečnika` je slično sastavljanju liste. Sledeći jednostavan primer kreira rečnik od liste reči u kojem će vrednosti u rečniku biti dužine odgovarajućih reči:

```
reči=['jabuka','banana','jagoda']
d = {s : len(s) for s in reči}
{'jabuka': 6, 'banana': 6, 'jagoda': 6}
```

Primer

Na programskom jeziku Python sastaviti funkciju koja vrši obradu and spiskom nadimaka:

- u jednom redu sadrži podatke o nadimku, polu(0–ženski,1–muški) i imenima.
- Izlaz treba da grupiše imena po nadimcima.

Ako je nadimak za imena oba pola koristiti oznaku 2.

Ulaz	Izlaz
Tea 0 Dorotea Teodora Saša 0 Aleksandra Fića 1 Filip Jela 0 Jelisaveta Saša 1 Aleksandar Jela 0 Jelica Jelena	Tea 0 Dorotea, Teodora Saša 2 Aleksandra, Aleksandar Fića 1 Filip Jela 0 Jelisaveta, Jelica, Jelena

Primer

```
def obrada():
    nadimci = {}
    while True:
        try:
            linija = input()
            nadimak, pol, *imena = linija.split()
            imena = ", ".join(imena)
            if nadimak not in nadimci:
                nadimci[nadimak] = [pol, imena]
            else:
                nadimci[nadimak][1] += ", " + imena
            if pol != nadimci[nadimak][0]:
                nadimci[nadimak][0] = 2
        except EOFError:
            break
    for k, v in nadimci.items():
        print("{:6}{:<3}{}".format(k, v[0], v[1]))

obrada()
```