Programiranje za umetnike 1

~ 10 ~

Staša Vujičić Stanković

Teme

Složeni tipovi podataka
– kolekcije –
torke, skupovi, rečnici

Kolekcije

Uređene

- stringovi
  - liste
  - torke
- rečnici (počev od verzije Python 3.7)

Neuređene

- skupovi

Kolekcije

Nepromenljivi tipovi

- **stringovi**
- **torke**

Promenljivi tipovi

- skupovi
- liste
- rečnici

Torke

- Torke (tip `tuple`) se upotrebljavaju da se smesti veći broj vrednosti u jednu promenljivu
- Torke predstavljaju uređene, nepromenljive kolekcije!
- Za razliku od lista ne mogu se menjati nakon kreiranja.
- Mogu da sadrže elemente različitih tipova.
- Elementi se prilikom navođenja u zagradama () razdvajaju zarezima.
- Ako torka sadrži samo jedan element, iza njega mora da stoji zarez

Torke

- Elementi torki su:
  - uređeni,
  - nepromenljivi i
  - dozvoljene su ponovljene vrednosti.
- Elementi torki su indeksirani, počev od 0.
- Za određivanje broja elemenata torke, koristi se funkcija `len()`

Torke

```
torka = ()        # prazna torka ili
torka2 = tuple()  # prazna torka
torka3 = ('jedini element', )
torka4 = ('jedini element') # nije torka vec string!

# kao argument konstruktora tuple()
# navodi se validna sekvenca
torka = tuple(['crvena', 'zelena', 'zuta'])

print(torka[0])
# torka[1] = 'bela' # greska jer je torka nepromenljiv tip
torka_nova = torka[1:2] + ('bela',)
print(torka_nova)
```

Skupovi

- Skupovi su neuređene, neindeksirane kolekcije objekata
  - Ne pamti se redosled ubacivanja elemenata u skup
  - Elementi skupa su nepromenljivi, ali se mogu ukloniti i dodati novi
  - Nisu dozvoljeni duplikati
- Za određivanje broja elemenata skupa, koristi se funkcija `len()`

```
drugovi = {'Pera', 'Mika', 'Laza'}
print(drugovi)
```

Skupovi

- Elementima skupa se ne može pristupiti upotrebom indeksa ali se:
  - za pristup elementima može iskoristiti for petlja
  - za proveru da li je određena vrednost element skupa može upotrebiti ključna reči `in`.

```
drugovi = set(['Pera','Mika','Laza', 'Pera'])

drugovi.add('Andrija')

for drug in drugovi:
    print(drug)

drugovi = set(['Pera','Mika','Laza', 'Pera'])

novi_drugovi = ['Mirko', 'Slavko', 'Aca']

drugovi.update(novi_drugovi)

for drug in drugovi:
    print(drug)
```

Metode za rad sa skupovima

Metoda	Opis
A.add(element)	Ubacuje element u skup A
A.discard(element)	Uklanja element iz skupa A
A.remove(element)	Uklanja element iz skupa A pri čemu prijavljuje grešku ako element nije u skupu
A.clear()	Uklanja sve članove skupa
A.pop()	Vraća slučajno odabran element skupa i uklanja ga iz skupa
A.copy()	Vraća kopiju skupa
del	ključna reč del u potpunosti eliminiše skup

Operatori nad skupovima

Operator	Metoda	Opis
A | B	A.union(B)	Unija skupova A i B
A |= B	A.update(B)	Dodaje sve elemente iz B u A
A & B	A.intersection(B)	Presek skupova A i B
A &= B	A.intersection_update(B)	U skupu A ostaje presek A i B
A – B	A.difference(B)	Elementi skupa A koji nisu u skupu B
A -= B	A.difference_update(B)	U skupu A ostaju oni koji nisu u B
A ^ B	A.symmetric_difference(B)	Elementi koji su samo u A ili samo u B
A ^= B	A.symmetric_difference_update(B)	Elementi koji su samo u A ili samo u B ostaju u skupu A

Operatori nad skupovima

Operator	Metoda	Opis
A == B		Vraća True ako je skup A jednak B
A != B		Vraća True ako je skup A različit od B
A <= B	A.issubset(B)	Vraća True ako je A podskup B
A >= B	A.issuperset(B)	Vraća True ako je A nadskup B
A < B		Vraća True ako je A pravi podskup B
A > B		Vraća True ako je A pravi nadskup B
	A.isdisjoint(B)	Vraća True ako je presek A i B prazan

Podskupovi

- Za generisanje podskupova se koristi metod `combinations`, modula `itertools`:

```
import itertools as it

def podskupovi(skup, velicina_podskupa):
    return [set(i)
    for i in it.combinations(skup, velicina_podskupa)]

A = {1, 2, 3, 4, 5}
n = 4

print(podskupovi(A, n))
```

Rečnici

- Rečnici (tip `dict`) su uređene, promenljive kolekcije
  (počev od Python 3.7 verzije – do nje su bili neuređene kolekcije)
- Nazivaju se još i heš mape ili heš tabele
- Služe za čuvanje parova objekata ključ-vrednost
- Omogućavaju dobijanje vrednosti na osnovu ključa
- Pri navođenju se ključ i vrednost u paru razdvajaju znakom dvotačka (:)
- Parovi se navode u { } i razdvajaju zarezom

```
imenik = {'Ime':'Pera', 'Prezime':'Peric', 'Broj':'063333333'}
```

Rečnici

Vrednosti u elementima rečnika mogu biti bilo kog tipa podataka

```
imenik1 = {} # prazan ili
imenik2 = dict() # prazan
imenik1 = {'Ime':'Pera', 'Prezime':'Peric', 'Broj': 33}

# Eksplicitno ubacivanje parova
imenik2['Ime'] = 'Jovan'
imenik2['Prezime'] = 'Jovanovic'
imenik2['Broj'] = 44
```

Rečnici

- Rečnici ne mogu sadržati duplikate
  (ne mogu imati dva elementa sa istim ključem)

```
imenik1 = {
    'Ime':'Pera',
    'Prezime':'Peric',
    'Broj':'063333333',
    'Broj':'065555555',
    }

print(imenik1)
```

Rečnici – dodavanje elemenata

- Dodavanje elementa u rečnik se vrši korišćenjem novog indeksnog ključa i dodeljivanjem vrednosti

```
imenik = {
    'Ime':'Pera',
    'Prezime':'Peric',
    'Broj':'063333333'
    }

kljucevi = imenik.keys()

print("Pre promene", kljucevi)

imenik['Inicijali']='PP'

print("Posle promene", kljucevi)
```

Rečnici – dodavanje elemenata

- Metod update() ažurira rečnik elementima iz datog argumenta.
  Ako element ne postoji, stavka će biti dodata.
- Argument mora biti rečnik ili
  iterativni objekat sa parovima ključ-vrednost.

```
imenik = {
    'Ime':'Pera',
    'Prezime':'Peric',
    'Broj': 33
    }

imenik_izmena = {
    'Ime': 'Mika',
    'Grad': 'Novi Sad'
}

print("Pre promene", imenik)

imenik.update(imenik_izmena)

print("Posle promene", imenik)
```

Rečnici – uklanjanje elemenata

```
# Metod pop() uklanja element sa navedenim ključem
imenik = {
    'Ime':'Pera',
    'Prezime':'Peric',
    'Broj': 33,
    'Grad': 'Novi Sad'
    }

print("Pre promene", imenik)

imenik.pop('Grad')

print("Posle promene", imenik)
```

Rečnici – uklanjanje elemenata

```
# Metod popitem() uklanja poslednji element
imenik = {
    'Ime':'Pera',
    'Prezime':'Peric',
    'Broj': 33,
    'Grad': 'Novi Sad'
    }

print("Pre promene", imenik)

imenik.popitem()

print("Posle promene", imenik)
```

Rečnici – uklanjanje elemenata

```
# Kljucna rec del uklanja element sa navedenim kljucem
imenik = {
    'Ime':'Pera',
    'Prezime':'Peric',
    'Broj': 33,
    'Grad': 'Novi Sad'
    }

print("Pre promene", imenik)

del imenik['Grad']

print("Posle promene", imenik)
```

Rečnici – uklanjanje elemenata

```
# ...ili kompletan recnik ukoliko nije naveden kljuc
imenik = {
    'Ime':'Pera',
    'Prezime':'Peric',
    'Broj': 33,
    'Grad': 'Novi Sad'
    }

print("Pre promene", imenik)

del imenik

print("Posle promene", imenik)
```

Rečnici – uklanjanje elemenata

```
# Metod clear() prazni recnik
imenik = {
    'Ime':'Pera',
    'Prezime':'Peric',
    'Broj': 33,
    'Grad': 'Novi Sad'
    }

print("Pre promene", imenik)

imenik.clear()

print("Posle promene", imenik)
```

Rečnici – pristup elementima

```
imenik = {
    'Ime':'Pera',
    'Prezime':'Peric',
    'Broj': 33,
    'Grad': 'Novi Sad'
    }
# stampa sve kljuceve
for el in imenik:
    print(el)
print("--------------------")
# ili - stampa sve kljuceve
for el in imenik.keys():
    print(el)
print("--------------------")

# stampa sve vrednosti
for el in imenik:
    print(imenik[el])
print("--------------------")
# ili - stampa sve vrednosti
for vr in imenik.values():
    print(vr)
print("--------------------")
# prolazi kroz sve parove kljuc-vrednost
for klj,vr in imenik.items():
    print(klj,vr)
```

Rečnici – kopiranje

```
imenik = {
    'Ime':'Pera',
    'Prezime':'Peric',
    'Broj': 33,
    'Grad': 'Novi Sad'
    }

# pogresan nacin
drugi_imenik = imenik
print(drugi_imenik)
drugi_imenik['Ime'] = 'Mika'
print(drugi_imenik)
print(imenik)
```

Rečnici – kopiranje

```
imenik = {
    'Ime':'Pera',
    'Prezime':'Peric',
    'Broj': 33,
    'Grad': 'Novi Sad'
    }

# ispravno - upotrebom metoda copy()
drugi_imenik = imenik.copy()
print(drugi_imenik)
drugi_imenik['Ime'] = 'Mika'
print(drugi_imenik)
print(imenik)
```

Rečnici – kopiranje

```
imenik = {
    'Ime':'Pera',
    'Prezime':'Peric',
    'Broj': 33,
    'Grad': 'Novi Sad'
    }

# ispravno - upotrebom metoda dict()
drugi_imenik = dict(imenik)
print(drugi_imenik)
drugi_imenik['Ime'] = 'Mika'
print(drugi_imenik)
print(imenik)
```

Metode za rad sa rečnicima

Metoda	Opis
A.clear()	Uklanjanje svih elemenata iz rečnika
A.copy()	Vraća kopiju rečnika
dict.fromkeys(seq [, v])	Vraća novi rečnik sa ključevima iz seq i vrednostima v (ako vrednost nije data onda None)
A.get(key[, d])	Vraća vrednost za ključ key. Ako ne postoji, onda vraća d, a ako nije dato d vraća None
A.items()	Vraća pregled svih elemenata rečnika u obliku ključ, vrednost
A.keys()	Vraća pregled svih ključeva rečnika
A.pop(key[,d])	Uklanja element sa ključem key i vraća njegovu vrednost, ili vraća d ako element nije pronađen (ako d nije navedeno vraća KeyError)
A.popitem()	Vraća i uklanja poslednji uneti element. Za prazan rečnik vraća KeyError.

Metode za rad sa rečnicima

Metoda	Opis
A. setdefault(key[, d])	Ako je key u rečniku, vraća vrednost. Ako nije, dodaje key sa vrednošću d i vraća tu vrednost
A.update([other])	Ažurira rečnik parovima ključ-vrednost iz other. a Ako neki ključ postoji, upisuje preko postojećeg
A.values()	Vraća pregled svih ključeva rečnika
len(A)	Vraća broj elemenata u rečniku
sorted(A)	Vraća novu, sortiranu listu ključeva u rečniku
del	ključna reč del briše element na osnovu zadatog ključa ili u potpunosti eliminiše rečnik

Literatura

- Python 3.10.0 documentation
- Wentworth, Peter, Elkner, Jeffrey, Downey, Allen B. and Meyers, Chris. How to Think Like a Computer Scientist: Learning with Python 3. free online book
- Lutz, Mark. Learning python: Powerful object-oriented programming. O'Reilly Media, Inc., 2013.
- Beazley, David, and Jones, Brian. Python Cookbook: Recipes for Mastering Python 3. O'Reilly Media, Inc., 2013.
- Python Cheatsheet
- Website Setup Python cheat sheet
- Learn Python, basic tutorial

Hvala

Staša Vujičić Stanković

stasa.vujicic.stankovic@math.rs

www.matf.bg.ac.rs/~stasa.vujicic.stankovic