

Beleške sa časa – 6.nedelja

Terminologija

delimiter = separator = razdvajač = graničnik (između elemenata koje želimo da izdvojimo)

redundantnost = ponavljanje (naredbi u kontekstu programskog kôda; podataka u opštem kontekstu)

Placeholder promenljiva

- placeholder promenljiva predstavlja simboličku promenljivu koja se može iskoristiti na bilo kom mestu u kôdu umesto stvarne promenljive ukoliko nam vrednost te promenljive nije od značaja
- kao što joj naziv kaže, placeholder promenljiva služi kao čuvar mesta gde bi u kôdu trebalo da stoji stvarna promenljiva koja prihvata neku vrednost (kao što je npr. povratna vrednost funkcije, vrednost brojača u brojačkoj petlji, itd.)
- u programskom jeziku Python se čuvari mesta označavaju podvlakom _
- placeholder promenljiva se koristi za prihvatanje vrednosti koje nisu od značaja za dalje izvršavanje programa (irelevantni podaci)

Prihvatanje većeg broja povratnih vrednosti funkcije

- funkcije u suštini uvek mogu da vrate samo jednu povratnu vrednost tj. jedan objekat
- kada funkcija treba da vrati više od jedne vrednosti, tada se te vrednosti pakuju u neku kolekciju (listu, torku, skup ili rečnik) i kao takve se prosleđuju u vidu jedne povratne vrednosti funkcije
- prilikom prihvatanja povratne vrednosti funkcije koja predstavlja neku kolekciju, moguće je izvršiti tzv. raspakivanje kolekcije korišćenjem mehanizma višestrukog dodeljivanja vrednosti
- kada znamo koliko kolekcija ima elemenata, možemo da elemente kolekcije dodelimo odgovarajućem broju promenljivih:

```
kolekcija = f(ar_1, ..., ar_x)
```

```
el_0, el_1, ..., el_n = kolekcija
```

ili direktno, bez pomoćne promenljive koja prihvata celu kolekciju:

```
el_0, el_1, ..., el_n = f(ar_1, ..., ar_x)
```

Redundantnost u kôdu

- redundantnost u kôdu se odnosi na prisustvo dupliranih (ponovljenih) ili suvišnih naredbi ili čak čitavih segmenata kôda

- primer redundantnosti u kôdu:

```
x = int(input())
```

```
y = int(input())
```

```
if x > y:
```

```
    ...
```

```
else:
```

```
    ...
```

```
if x > y:
```

```
    ...
```

```
else:
```

```
    ...
```

```
x = int(input())
```

```
if x%2 == 0:
```

```
    parnost = "paran"
```

```
    kolicnik = x // 2
```

```
else:
```

```
    parnost = "neparan"
```

```
    kolicnik = x // 2
```

- redundantnost u kôdu je neophodno izbegavati zato što značajno otežava čitljivost kôda (razumevanje ideje programa, "šta je pisac htio da kaže"), kao i menjanje i održavanje koda (ažuriranje programa, dodavanje novih funkcionalnosti programa, itd.)

- redundantnost čini kôd podložnijim nastajanju grešaka prilikom menjanju i održavanju koda jer je teško ispratiti gde će se sve unete izmene odraziti u celom programu

- redundantnost u kôdu se uklanja objedinjavanjem ponovljenih naredbi u blokove kôda i/ili pomoćne funkcije