

# Asocijativne i neuređene strukture podataka

---

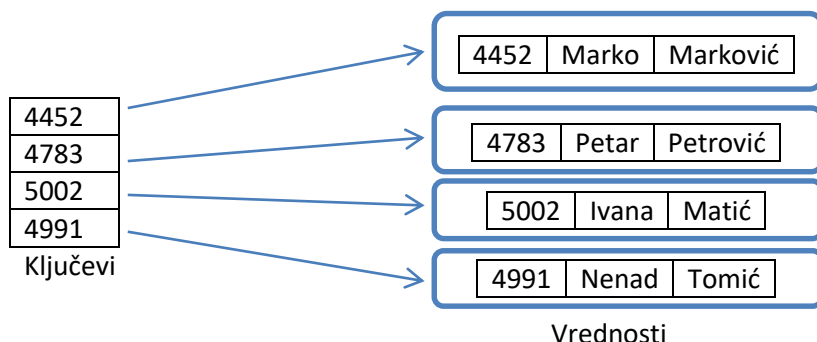
**Autori:**  
**Goran Savić**  
**Milan Segedinac**

# Strukture podataka

## 1. Asocijativni niz

Asocijativni niz predstavlja strukturu podataka koja skladišti kolekciju parova {*ključ*, *vrednost*}. Ova struktura se ne bazira na sekvenci kao prethodno opisane strukture. Umesto putem indeksa u sekvenci, elementu se pristupa putem njegovog ključa koji je mapiran na njemu odgovarajuću vrednost. Zbog ovakve strukture, asocijativni niz se često naziva i mapa ili rečnik. Iako se elementima ne pristupa putem indeksa, zavisno od implementacije, sama struktura može biti realizovana i kao uređena i kao neuređena. U uređenoj varijanti asocijativnog niza, postoji definisan redosled između ključeva elemenata.

Na slici je dat primer asocijativnog niza koji skladišti studente. Ključ za studenta je broj indeksa.



Operacije koje standardno asocijativni niz omogućuje su:

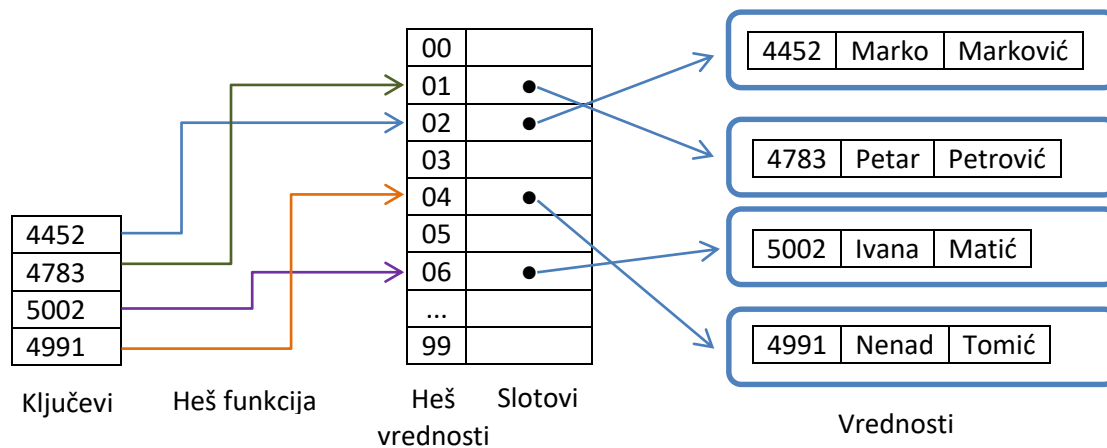
- Dodavanje para {*ključ*, *vrednost*} u kolekciju
- Izbacivanje para iz kolekcije (na osnovu ključa)
- Pronalaženje vrednosti u kolekciji na osnovu njenog ključa
- Izmena postojećeg para - vezivanje nove vrednosti za postojeći ključ

Najvažniji aspekt implementacije asocijativnog niza je realizacija mapiranja ključa na vrednost. Dakle, potrebno je omogućiti pronalaženje vrednosti na osnovu ključa. Ovo je moguće realizovati na više načina. Najjednostavniji način je uvezivanje elemenata u sekvencijalnu listu. Tada se element po ključu pronalazi linearnim prolaskom kroz elemente. Obzirom da u takvoj realizaciji vreme potrebno za pronalaženje elementa linearno raste sa povećanjem broja elemenata u asocijativnom nizu, ovakav pristup je prikladan samo za nizove sa malo elemenata. Drugi način se mnogo češće koristi i on podrazumeva realizaciju preko heš tabele. Konačno, treći način koji se često koristi u praksi je korišćenjem strukture tipa stablo. Obzirom da ćemo se stablom kao strukturom baviti kasnije, ovde će biti objašnjena implementacija asocijativnog niza korišćenjem heš tabele.

Heš tabela koristi heš funkciju koja vrši transformaciju ključa u broj koji predstavlja indeks elementa u nizu slotova (eng. *buckets* ili *slots*). Slotovi sadrže vrednosti iz asocijativnog niza. Heš funkcijom nazivamo bilo koju funkciju koja mapira ulazne podatke proizvoljne dužine u izlazne podatke fiksne dužine. Vrednost koju heš funkcija vraća se naziva heš vrednost ili heš kod.

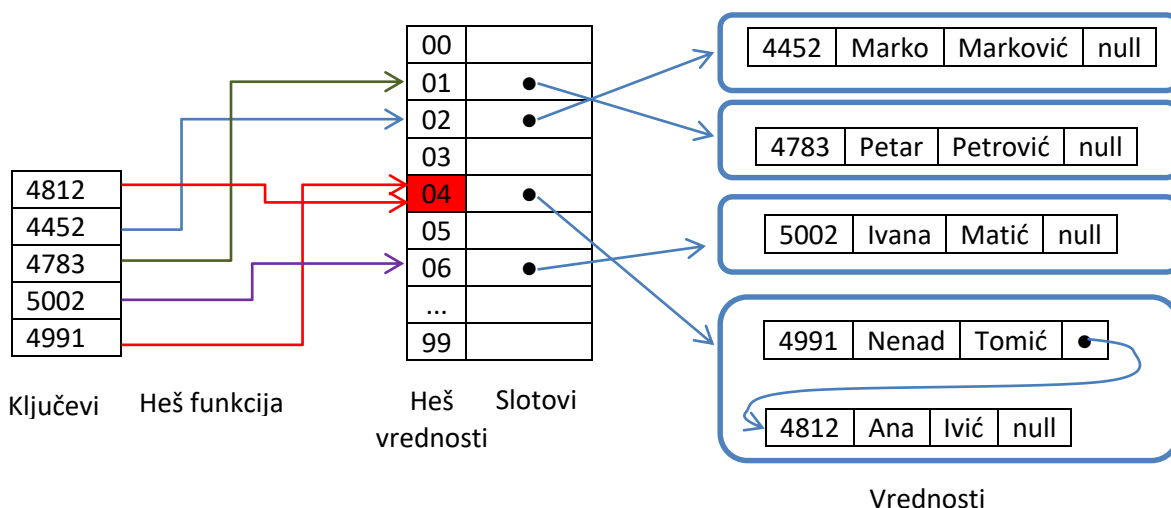
Implementacija asocijativnog niza iz prethodnog primera korišćenjem heš tabele je prikazana na slici.

## Strukture podataka



Vidimo da prikazana heš funkcija mapira brojeve indeksa studenta na cele brojeve između 0 i 99. Ovi brojevi predstavljaju indeks u nizu slotova. U svakom slotu je upisana vrednost onog elementa asocijativnog niza koji odgovara ključu koji je mapiran na indeks tog elementa.

U idealnom slučaju, heš funkcija svaki ključ mapira na različit indeks u nizu slotova. Međutim, ovo uglavnom nije slučaj. Umesto toga, različiti ključevi se nekad mapiraju na isti indeks. Tada slot sadrži više vrednosti. Dakle, indeksom elementa u nizu slotova nije jednoznačno određena vrednost kojoj se pristupa za taj ključ. Ovo se naziva kolizija i može se rešiti na različite načine. Jedan način je da se svi elementi u slotu (to su elementi koji se mapiraju na isti indeks) uvežu u spregnutu listu. Tada slot ne sadrži vrednost određenog elementa nego listu elemenata čiji se ključevi mapiraju na isti indeks. Za pronalaženje elementa, neophodno je putem heš funkcije doći do indeksa odgovarajućeg slota, a zatim sekvencijalnim prolaskom kroz listu vezanu za taj slot pronaći odgovarajući element. Ako je heš funkcija dobro definisana, tada će elementi biti prilično ravnomerno raspoređeni po slotovima, pa sekvencijalni prolazak kroz listu neće značajno smanjiti performanse, jer lista neće biti prevelika. Naredna slika prikazuje organizaciju heš table u situaciji kada heš funkcija može različite ključeve mapirati na isti indeks slota.



Vidimo da su vrednosti sada organizovane u spregnutu listu i imaju dodatno polje koje predstavlja referencu na naredni element u listi. U slučaju kada postoji kolizija (u primeru se brojevi indeksa 4812 i 4991 preslikavaju na isti indeks slota), vrednosti vezane za ključeve koji su uzrok kolizije se

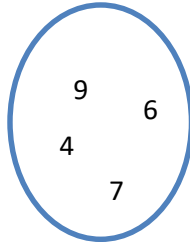
# Strukture podataka

---

organizuju tako da formiraju sekvencu time što elementi sadrže referencu na naredni element u tom slotu.

## 2. Skup

Pomenućemo ovde još i strukturu podataka pod nazivom skup. Ova struktura podataka je računarska predstava matematičkog koncepta konačnog skupa. Skup je neuređena kolekcija elemenata koja ne sadrži duplikate. Dakle, određeni element se samo jednom može dodati u skup. Iz tog razloga, kada radimo sa skupovima važno je definisati kako se određuje da li su dva elementa jednaka. Skup je ilustrovan na narednoj slici.



Ovde treba pomenuti i da postoji neuređena kolekcija koje dozvoljava duplikate. Takva kolekcija se naziva multiset ili beg (eng. *bag*).