

## Priprema za vježbu 6

*Cilj vježbe je upoznavanje sa apstraktnim tipom podataka "Mapa".*

*Rok je naveden u informacionom sistemu zamger.*

### Zadatak 1.

Napraviti apstraktnu generičku klasu Mapa. Mapa je vrlo često korišten apstraktni tip podataka u programiranju. Mapa uvezuje podatak jednog proizvoljnog tipa (ključ) sa podatkom drugog proizvoljnog i ne nužno istog tipa (vrijednost). Programer često doživljava mapu kao jednu posebnu vrstu niza / vektora kod kojeg ključ ne mora biti cjelobrojnog tipa niti mora biti u nekom fiksnom opsegu. Npr. mogli biste pisati:

```
NizMapa<string, string> glavni_gradovi;  
glavni_gradovi["Bosna i Hercegovina"] = "Sarajevo";  
glavni_gradovi["Hrvatska"] = "Zagreb";  
glavni_gradovi["Sjedinjene Američke Države"] = "Washington";
```

Potrebno je napraviti apstraktnu klasu kako bismo mogli eksperimentisati sa raznim strategijama implementacije Mape. Apstraktna klasa Mapa treba podržavati sljedeće metode:

- Konstruktor bez parametara koji kreira praznu mapu. Ovdje treba istaći da, pošto je Mapa generička klasa sa dva tipa (TipKljuča i TipVrijednosti), a konstruktor nema parametara, prilikom kreiranja objekta tipa Mapa morate navesti oba tipa na sljedeći način:

```
NizMapa<string, float> m;
```

- Preklopljeni operator uglaste zagrade [] (konstantna i nekonstantna varijanta), kako bi se mapa mogla koristiti u svim situacijama kao i niz (vidjeti primjer iznad). Default vrijednost člana mape (ako ključ ne postoji u mapi) treba biti TipVrijednosti()
- Metodu **int brojElementata()** koja vraća broj elemenata trenutno pohranjenih u mapi.
- Metodu **void obrisi()** koja briše sve elemente iz mape. Nakon poziva ove metode mapa mora i dalje biti upotrebljiva (trebaju se moći dodavati novi elementi).
- Metodu **void obrisi(const TipKljuča& kljuc)** koja briše samo jedan element iz mape sa ključem *kljuc*. Razlog za ovu metodu je što obrisati element nije isto što i pridružiti mu nulu ili prazan string, jer se brisanjem oslobađa memorija. U slučaju da ključ ne postoji baciti izuzetak.

Da bi se ovakva Mapa ikako mogla implementirati, potrebno je da TipKljuča bude uporedivi tip (mora podržavati operatore poređenja). Za TipVrijednosti se traži samo da posjeduje default konstruktor, kao što je slučaj i prilikom kreiranja niza. Na primjer: za klasu *std::vector* je definisan default konstruktor ali nije definisan operator poređenja.

### Zadatak 2.

Napraviti klasu NizMapa izvedenu iz klase Mapa, kod koje je mapa implementirana sa dinamički alociranim nizom koji koristi Pair (niz ključeva i niz vrijednosti). Voditi računa da veličina mape ne bude ograničena ničim osim dostupnom memorijom na računaru, ali i da dodavanje novog elementa

u mapu ima optimalne performanse. Omogućiti da se NizMapa može slati funkcijama, vraćati iz funkcija, da se nad konstantnom referencom na NizMapu mogu vršiti sve operacije koje ima smisla vršiti, te da pri svemu tome ne dolazi do curenja memorije. Napravite i minimalno pet testnih funkcija u kojim testirate osobine NizMape sa različitim tipovima ključa i vrijednosti.