

PROGRAMSKI JEZICI 1 – 1.2.2016.

- ❶ (60 bodova) U prostoru imena **Kolokvijum** definisati generičku klasu **List<T>** koja predstavlja jednostruko uvezanu listu. Potrebno je obezbijediti mogućnost daljeg nasljeđivanja iste klase. Potrebno je implementirati konstruktore kopije i pomjeranja, operatore dodjele (sa pomjeranjem i kopiranjem) i destruktora. Omogućiti optimizaciju koda garantovanjem da konstruktori neće baciti izuzetke. Potrebno je enkapsulirati klasu **Node** koja predstavlja čvorove liste. Realizovati:

- Metodu **print(std::ostream&)** koja treba da bude vidljiva samo izvedenim klasama. Ova metoda treba da štampa elemente liste s lijevim poravnanjem i konstantnom širinom od pet karaktera (koristiti **<iomanip>**)
- Operator za ispis na izlazni tok
- Operator **+=** za dodavanje novog elementa u listu
- Enkapsulirati javnu klasu **Iterator** koja predstavlja iterator za listu i implementira operatore inkrementa (prefiksni i postfixni **operator++** i **operator+=**), operator dereferenciranja, **operator !=** koji poredi dva iteratora prema indeksu elementa na koji pokazuju i operator dodjele (sa kopiranjem i pomjeranjem) za element
- Operator **[]** za pristup i-tom elementu liste (implementirati operator korištenjem iteratora)
- Izvršiti specijalizaciju šablona za **std::hash**, pri čemu se vrijednost heša računa kao bitska XOR operacija nad heševima svih elemenata liste. Zaglavlje za **std::hash** dato je ispod.

```
struct std::hash<Klasa>
{
    size_t operator()(const Klasa& x) const
    {
        return izracunHesa(x);
        /*npr.:
        return std::hash<int>()(x.Vrijednost);*/
    }
};
```

- ❷ (40 bodova) U prostoru imena **Kolokvijum** implementirati klasu **HashTable<T, hash>** koja predstavlja heš¹ tabelu elemenata tipa **T** sa podrazumijevanim heš generatorom **std::hash<T>** i javno je naslijeđena iz klase **List<T>**. Konstruktor ove klase treba da prihvata parametar koji određuje kapacitet tabele. Potrebno je implementirati konstruktore kopije i pomjeranja, operatore dodjele (sa pomjeranjem i kopiranjem) i destruktora. Omogućiti optimizaciju koda garantovanjem da konstruktori neće baciti izuzetke. Potom realizovati:

- Metodu **insert(T&)** za umetanje elemenata. Elemente umetati na indeks određen heš vrijednošću preslikanom na opseg niza na proizvoljan način (koristiti niz pokazivača na umetnute elemente), a ako element već postoji na tom indeksu u naslijeđenu listu. Ako je u tabeli već maksimalan broj elemenata, baciti izuzetak tipa **char*** sa porukom „Hash table full.“
- Metodu **exists(const T&)** koja određuje da li dati element postoji u tabeli. Ova funkcija ne smije da baci izuzetak.
- Operator za ispis na izlazni tok (redosljed elemenata nije bitan, ali svi moraju biti ispisani)
- U glavnom programu ilustrovati rad heš tabele lista cjelobrojnih vrijednosti, sa hvatanjem potencijalnih izuzetaka i upisom tabele u datoteku pod nazivom „out.txt“.

Napomene:

Razdvojiti interfejs i implementaciju.

Izbjeći dupliranje koda.

U zaglavljima klasa koristiti punu specifikaciju imena objekata i metoda iz standardne biblioteke.

Koristiti standardnu biblioteku.

¹Heš je generisana cjelobrojna vrijednost kojom se predstavlja objekat neke klase. Heš ne mora da bude jedinstven za svaki objekat, ali se obično konstruiše tako da različiti objekti iste klase imaju različite heševe, kako bi se pojednostavilo poređenje. Ako dva objekta imaju različit heš, oni su različiti. Ako dva objekta imaju isti heš, onda ih je neophodno porediti odgovarajućim operatorom poređenja.