

UNIVERZITET U TUZLI  
FAKULTET ELEKTROTEHNIKE



# Strukture podataka

Liste

Zadaća 2

Tuzla, april/travanj 2020.

# Sadržaj

<b>Sadržaj</b>	<b>2</b>
<b>Zadatak 1:</b>	<b>3</b>
<b>Zadatak 2:</b>	<b>4</b>
2.1. Dohvatanje informacija za cijelu godinu	4
2.2. Dohvatanje informacija za određeni mjesec	5
2.3. Dodavanje informacija	5
2.4. Napomene	5

## Zadatak 1:

Implementirati **sortiranu dvostruko linkanu listu** koja treba da zadovolji sljedeće funkcionalnosti:

- Lista mora biti generička te sadržavati sve neophodne konstruktore, operatore jednakosti, te destruktora.
- Lista mora sadržavati iterator klasu koja zadovoljava osobine bidirekcionog iteratora, skupa sa `begin` i `end` metodama.
- Metod `add` koji dodaje element u listu tako da lista bude uzlazno sortirana (npr. 1, 2, 3, ...). Ovaj metod nema povratne vrijednosti, a sadrži dva overloada, jedan koji prima `const T&` i drugi koji prima `T&&`,
- Metode `find()` i `find_if()` koji vraćaju iterator na pronađeni element, ili iterator na kraj liste u slučaju da nema traženog elementa. Metod `find` prima konstantnu referencu na neku vrijednost, dok `find_if` prima predicate funkciju.
- Metod `remove()` koji uzima iterator i briše element u listi sa te pozicije (Obratiti pažnju na granične slučajeve!),
- Metode `size`, `empty` i `clear`,
- Implementirati dodatne metode potrebne za implementaciju narednog zadatka.

U prilogu zadaće se nalazi file `zadatak1.cpp` koji predstavlja primjer korištenja prethodno opisane klase.

## Zadatak 2:

Napraviti strukturu `ToDoTask`, koja predstavlja obavezu koju korisnik treba uraditi. `ToDoTask` se sastoji od:

- `id`,
- `naslova` (title),
- `opisa` (description),
- `prioriteta` (priority).

Implementirati potrebne operatore za dodavanje strukture u prethodno implementiranu sortiranu listu. Prilikom kreiranja objekta `ToDoTask`, od korisnika tražiti unos naslova, opisa i prioriteta, a za vrijednost `id`-a iskoristiti sljedeću generator funkciju:

```
int generateId(){
    static int counter = 0;
    return ++counter;
}
```

Napraviti glavni program koji implementira funkcionalnost notesa. Korisniku je potrebno prikazati meni za unos nove obaveze, prikaz trenutnih obaveza te označavanje obaveze kao završene. Obaveze se unose u sortiranu listu, tako da se obaveze sa najvećim prioritetom nalaze na početku liste. Prilikom označavanja obaveze kao završene, korisniku ponuditi

unos id-a obaveze, te na osnovu toga izvršiti brisanje obaveze iz liste (koristeći metode find() i remove()).

Dodatno, napraviti opciju za ispis 3 obaveze sa najvećim prioritetom, te 3 obaveze sa najmanjim prioritetom (ukoliko notes sadrži manje od 3 obaveze, ispisati sve obaveze).

Napomena:

Za izradu ovog zadatka potrebno je koristiti listu kreiranu iz prvog zadatka. U folderu za zadatak2, uz nove hpp i cpp datoteke, možete kopirati potrebne datoteke iz prvog zadatka.

## Zadatak 3:

Napisati program koji korisniku daje historijske meteorološke informacije za određenu meteorološku stanicu. File stations.csv sadrži listu meteoroloških stanica za koje su podaci dostupni. Za svaku meteorološku stanicu dat je njen jedinstven identifikator, geografska širina, geografska dužina, nadmorska visina, grad/mjesto u kojem se stanica nalazi te dodatan csv file u kojem se nalaze svi zabilježeni podaci o minimalnoj i maksimalnoj temperaturi koju je ta meteorološka stanica zabilježila za određeni mjesec.

Kada se program pokrene treba korisniku da izlista meni kao na slici 1.

```
***** Historical temperature information *****
1. Get information for entire year
2. Get information for specific month
3. Insert new historical temperature information

4. Exit
Your choice?: █
```

Slika 1. Inicijalni meni

### 3.1. Dohvatanje informacija za cijelu godinu

Ukoliko je odabrana opcija 1, korisnik dalje treba da unese lokaciju stanice koja ga zanima nakon čega program ispisuje informacije o toj stanici kao što je to prikazano na slici 2.

```
Enter location where station is placed: Tuzla
StationID: BKM00014557
Latitude: 44.55
Longitude: 18.7
Elevation: 307

Enter year: █
```

Slika 2. Izbor 1

Nakon unošenja godine program treba da ispiše sve mjesece te godine zajedno sa minimalnom i maksimalnom temperaturom tog mjeseca. Ukoliko unutar file-a, taj mjesec nije dostupan program treba da ispiše N/A (not available) za minimalnu i maksimalnu temperaturu (Slika 3.).

```

Enter year: 2002

Month   Max temp   Min temp
Jan     5.4       -4.83
Feb     13.78    0.38
Mar     15.66    2.73
Apr     17.01    5.22
May     N/A      10.9
Jun     N/A      N/A
Jul     N/A      15.36
Aug     N/A      N/A
Sep     N/A      N/A
Oct     19.08    6.34
Nov     15.86    N/A
Dec     N/A      N/A

```

Slika 3. Konačan ispis prvog dijela programa

### 3.2. Dohvatanje informacija za određeni mjesec

Opcija 2 treba također da korisniku ponudi unošenje lokacije, te da ispiše informacije o toj stanici (Slika 2.). Nakon toga korisnik treba da unese godinu te mjesec za koji želi dohvatiti informacije te program ispisuje minimalnu i maksimalnu temperaturu za traženi mjesec (Slika 4.).

```

Enter location where station is placed: Tuzla
StationID: BKM00014557
Latitude: 44.55
Longitude: 18.7
Elevation: 307

Enter year: 2008
Enter month: 5
Max temp   Min temp
24.24      8.94

```

Slika 4. Konačan ispis drugog dijela programa

Kao i u prvom dijelu, ukoliko informacije nisu dostupne program treba da ispiše N/A.

### 3.3. Dodavanje informacija

Potrebno je implementirati i opciju kojom dodajemo novu informaciju o temperaturi za određeni mjesec. Od korisnika se traži unos jedinstvenog identifikatora stanice, godina, mjesec te minimalna i maksimalna temperatura. Ukoliko stanica sa unešenim identifikatorom ne postoji, program treba da ispiše da stanica ne postoji i vrati se na početni meni. Nakon unošenja podataka, korištenjem opcija 1 i 2 testirati da li smo pravilno unijeli podatke u korištene strukture podataka. U slučaju da informacija za unešeni mjesec već postoji, potrebno je prebrisati staru vrijednost. Nije potrebno ove informacije ponovno snimati u odgovarajuće fajlove već dodane informacije postoje samo dok program živi.

### 3.4. Napomene

Za učitavanje fajlova koristiti `std::ifstream` objekat iz standardne biblioteke. Za organizaciju podataka nije potrebno implementirati nikakve kolekcije već odabrati najefikasnije strukture podataka za pretraživanje (prvi i drugi dio programa) iz standardne biblioteke (od onih koje smo do sada radili). CSV file-ovi se nalaze u prilogu zadaće (zadatak3.zip).

## Način predavanja

Zadaću je potrebno predati u obliku arhive **ime\_prezime\_zadaca2.zip** pri čemu sadržaj arhive treba da bude sljedeći (nakon otpakivanja arhive na disku mora biti sljedeći sadržaj):

```
ime_prezime_zadaca1/  
├── zadatak1  
│   ├── file1  
│   ├── file2  
│   └── file3  
└── zadatak2  
    ├── file1  
    ├── file2  
    └── file3
```

Pored toga, potrebno je da se studenti strogo pridržavaju naziva metoda, njihovih parametara i povratnih vrijednosti.

Studenti koji se ne budu strogo pridržavali uputa neće biti adekvatno ocijenjeni. Svaki pokušaj prepisivanja će biti prijavljen i adekvatno sankcionisan.

**Uslov za izlazak na provjeru je uraditi minimalno 50% zadaće.**