

Smijete koristiti pisane materijale ili vlastite bilješke spremljene u HOME direktoriju, web-stranicu kolegija i linkove koji vode s nje.

Strogo je zabranjeno korištenje ChatGPT-a te će se isto provjeravati za svaki kod. Ne smijete se koristiti nikakvim sredstvima komunikacije (papirići, mobitel, mail, chat, messenger i sl.). Svaki pokušaj prepisivanja i dogovaranja rezultirat će trenutnim udaljavanjem sa kolokvija. Strogo je zabranjeno fotografiranje ili pretipkavanje teksta zadataka.

Iznimno, studenti koji kolokvij pišu u čitaonici, smiju upaliti hotspot na mobitelu, nakon čega mobitel moraju okrenuti licem prema stolu na prikladnoj udaljenosti. Mobitel se nakon toga više ne smije dirati do kraja izvođenja kolokvija.

Prvi zadatak (15 bodova)

U ovome zadatku potrebno je napisati strukturu u kojoj pamtimo podatke o sudioniku nekog događaja (primjerice, natjecanja, kongresa, izleta i sl.) te strukturu u kojoj pamtimo podatke o sudionicima kongresa (upotrebom ranije spomenute strukture za pojedinog sudionika).

Napišite parametriziranu strukturu *sudionik* koja sadrži sljedeće informacije o sudioniku nekog događaja:

- *std::string* za ime sudionika pod kojim se prijavljuje na događaj (poput *Bob* ili *prof. dr. Ann i prof. Bob*)
- generički tip **T** za id sudionika (koji jedinstveno određuje sudionika na određenom događaju)
- generički tip **U** za preostali iznos kotizacije koju sudionik još treba platiti (u EUR) za sudjelovanje na određenom događaju (to je neki numerički tip poput *int*, *float*, *double* i sl.).

Sudionika trebamo moći definirati naredbom poput `sudionik<unsigned,double> a;` (u kom slučaju taj sudionik ima ime *nepoznat*), `sudionik<unsigned,double> j("mr. John");` ili `sudionik<unsigned,double> j("mr. John", 0, 24.99);`. Uočite da smo samo u posljednjem primjeru propisali id i iznos kotizacije (u ostalima nam nisu važni).

Zatim, toj strukturi dodajte funkciju članicu naziva *ocisti* (povratnog tipa *void* i bez dodatnih argumenata) koja iz imena sudionika uklanja sljedeće titule: *dr.*, *mr.* i *prof.* (drugim riječima, uklanja sve pojave stringova *dr.*, *mr.* i *prof.* iz imena sudionika).

Napišite (neparametriziranu!) strukturu *kongres* koja sadrži sljedeće informacije o kongresu:

- dva *unsigned* broja - za trenutni broj sudionika i maksimalan broj sudionika tog kongresa
- *sudionik<unsigned,double>** za polje sudionika tog kongresa
- *double* za iznos kotizacije (početno kotizacija ima isti iznos za svakog sudionika tog kongresa).

Kongres trebamo moći definirati naredbom oblika `kongres Math25(100.99, 4);` (pritom je iznos kotizacije 100.99 EUR, maksimalan broj sudionika je 4 i početno nemamo sudionika na tom kongresu).

Za strukturu *kongres* implementirajte sljedeće funkcije:

- `unsigned prijavi (sudionik<unsigned,double> s)` koja dodaje sudionika u polje sudionika tog kongresa,
- `unsigned prijavi (std::string ime)` koja dodaje sudionika s imenom *ime* u polje sudionika tog kongresa.

IME I PREZIME:

MATIČNI BROJ STUDENTA:

BODOVI:

Svaki dodani sudionik mora imati jedinstveni id (na način da prvi prijavljeni na kongres ima id 1, drugi prijavljeni ima id 2, itd.) te početno ima kotizaciju jednaku kotizaciji kongresa. Pritom pri dodavanju iz imena sudionika moraju biti uklonjene sve titule (kako je ranije objašnjeno u ovome zadatku). Navedene funkcije vraćaju 0 ako više nema mjesta na tom kongresu za novog sudionika, a inače vraćaju id dodanog sudionika. Može postojati više sudionika s istim imenom.

Kako bi sva dinamički alocirana memorija bila dealocirana, dodajte **destruktor** za strukturu kongres. Također implementirajte **copy-konstruktor** za tu strukturu.

Naposlijetku, napišite (globalnu) funkciju `void ispis (kongres s)` koja ispisuje podatke o svim trenutno prijavljenim sudionicima kongresa. Pritom se u svakoj liniji ispisuju podaci za jednog sudionika - njegov id, njegovo ime i iznos kotizacije (samo brojka, bez *EUR*). Podaci za pojedinog sudionika ispisuju se međusobno odvojeni zarezom.

Napomena: Rješenje rastavite na tri datoteke:

- `main.cpp` datoteku s implementacijom funkcije `ispis` i `main` funkcijom za testiranje (v. ispod!),
- `sudionik.h` datoteku s implementacijom strukture za podatke o pojedinom sudioniku (i pripadnim funkcijama članicama te strukture),
- `kongres.h` datoteku s implementacijom strukture za podatke o kongresu (i pripadnim funkcijama članicama te strukture).

U .h datotekama funkcije članice pojedine strukture moraju biti implementirane izvan deklaracije pripadne strukture te nije u .h datotekama dozvoljena upotreba `using namespace std`! Također je preporučljivo na početak sadržaja .h datoteka dodati: `#pragma once` Nije dozvoljena upotreba struktura iz STL-a kao ni ključne riječi `auto`.

main funkcija za testiranje:

```
int main(void) {
    sudionik<unsigned,double> a;
    sudionik<unsigned,double> j("mr. John");
    sudionik<unsigned,double> b("Bob");
    kongres Math25(100.99, 4);
    Math25.prijavi("mr. John");
    Math25.prijavi(a);
    Math25.prijavi(sudionik<unsigned,double>("prof. dr. Ann i prof. Bob"));
    cout << Math25.prijavi(b) << endl;
    cout << Math25.prijavi(j) << endl;
    ispis(Math25);

    return 0;
}
```

IME I PREZIME:

MATIČNI BROJ STUDENTA:

BODOVI:

Drugi zadatak (20 bodova)

U knjižnici je potrebno organizirati knjige te pratiti koliko zakasnine treba platiti pojedini član knjižnice. Podaci o posuđenoj knjizi spremljeni su u strukturu `knjiga` koja sadrži točno sljedeće podatke:

- `string naslov` - naslov knjige,
- `string ID_clana` - identifikacijska oznaka člana koji je posudio tu knjigu,
- `int zaduzenje` - broj dana koliko je do sada zadužena knjiga.

Knjižnica je podijeljena u odjele prema žanru knjiga. Motivirani time, organizaciju knjižnice provodimo preko globalne varijable

```
multimap<string, string> knjiznica.
```

Ključevi ove multimape su stringovi koji označavaju žanr knjige, a vrijednosti stringovi koji označavaju naslov knjige.

(i) Napišite funkciju s prototipom

```
queue<knjiga> azuriraj(multimap<string, pair<string,int> > podaci, string rijec),
```

gdje je varijabla `podaci` multimapa koja za ključeve ima identifikacijske oznake članova knjižnice, a za vrijednosti uređene parove naslova knjige koju su posudili i trenutnog broja dana zaduženja za taj naslov. Funkcija kreira i vraća red struktura `knjiga` svih posuđenih knjiga čiji se naslovi nalaze u multimapi `podaci`, a koje pripadaju žanru koji ima `rijec` kao podstring. Svakoj strukturi `knjiga` u kreiranom redu također upisuje i vrijednosti varijabli `ID_clana` i `zaduzenje`.

(ii) Napišite funkciju s prototipom

```
void zakasnine(queue<knjiga> posudjene, int cijena),
```

gdje je varijabla `posudjene` red struktura `knjiga`, a cjelobrojna varijabla `cijena` je iznos u eurima za jedan dan zakasnine. Zakasninu za jednog člana knjižnice računamo kao sumu brojeva $cijena * (zaduzenje - 7)$ po svim knjigama koje je zadužio strogo više od 7 dana. Funkcija ispisuje identifikacijske oznake članova knjižnice koji imaju zaduženu barem jednu knjigu strogo više od 7 dana, naslove svih takvih knjiga i ukupan iznos zakasnine za pojedinog člana. Ispis treba biti u sljedećem formatu:

```
ID_clana1: naslov11 naslov12 ... naslov1n1 iznos1
```

```
ID_clana2: naslov21 naslov22 ... naslov2n2 iznos2
```

```
⋮
```

IME I PREZIME:

MATIČNI BROJ STUDENTA:

BODOVI:

Kod gornjeg ispisa, identifikacijske oznake članova knjižnice, kao i naslovi posuđenih knjiga za svakog člana, trebaju biti uzlazno sortirani.

Možete pretpostaviti da će se funkcija zakasnine pozivati s varijablom tipa `queue<knjiga>` dobivenom jedino preko povratne vrijednosti funkcije `azuriraj` iz (i).

Napomena: Rješenje spremite u **jednu** datoteku `knjiznica.cpp`. **Dozvoljeno je korištenje struktura iz STL-a.**

Primjer: Za funkciju `main()` u nastavku, ispis treba biti:

BK789: Alisa u zemlji cudesca 6

MH123: Harry Potter Kralj prstenova 12

```
int main()
{
    knjiznica.insert(make_pair("Djecji,Fantastika","Harry Potter"));
    knjiznica.insert(make_pair("Znanost","Matematika 1"));
    knjiznica.insert(make_pair("Fantastika,Pustolovni","Kralj prstenova"));
    knjiznica.insert(make_pair("Bajka,Fantastika","Alisa u zemlji cudesca"));
    knjiznica.insert(make_pair("Priroda","Tropske biljke"));
    knjiznica.insert(make_pair("Bajka,Fantastika","Snjeguljica"));

    string rijec("Fantastika");

    multimap<string, pair<string,int> > podaci;

    podaci.insert(make_pair("MH123",make_pair("Harry Potter",8)));
    podaci.insert(make_pair("SS456",make_pair("Matematika 1",2)));
    podaci.insert(make_pair("MH123",make_pair("Kralj prstenova",12)));
    podaci.insert(make_pair("BK789",make_pair("Alisa u zemlji cudesca",10)));

    queue<knjiga> posudjene=azuriraj(podaci,rijec);

    zakasnine(posudjene, 2);

    return 0;
}
```

IME I PREZIME:

MATIČNI BROJ STUDENTA:

BODOVI: