

## Zadatak:

Napišite sučelje i implementaciju za strukture **Hotel** i **SkupHotela**. Struktura **Hotel** predstavlja hotel (ime, grad, cijena noćenja, broj slobodnih mjesta). Struktura **SkupHotela** predstavlja kolekciju podataka tipa **Hotel**. Ukupan broj hotela u kolekciji nikad neće biti veći od 100. Sučelje za obje strukture spremite u datoteku **hoteli.h**, a implementaciju u datoteku **hoteli.cpp**.

### Hotel – Konstruktori i funkcije članice

- **Hotel(string ime, string grad, int cijena, int brojSlobodnihMjesta)**

*Konstruktor za strukturu Hotel.*

*ime – ime hotela*

*grad – grad u kojem se nalazi hotel*

*cijena – cijena po noćenju*

*brojSlobodnihMjesta – broj slobodnih mjesta u hotelu*

- **string Ime(void)**

*Vraća ime hotela.*

- **string Grad(void)**

*Vraća grad u kojem se nalazi hotel*

.

- **int Cijena(void)**

*Vraća cijenu noćenja u hotelu.*

- **int BrojSlobodnihMjesta()**

*Vraća broj slobodnih mjesta u hotelu.*

### SkupHotela – Konstruktori i funkcije članice

- **SkupHotela()**

*Konstruktor za strukturu SkupHotela. Stvara praznu kolekciju hotela.*

- **SkupHotela(Hotel h)**

Konstruktor za strukturu **SkupHotela**. Stvara kolekciju koja se sastoji samo od hotela **h**.

- **void Dodaj(Hotel h)**

Dodaje hotel **h** u kolekciju. Hotel **h** se uvijek dodaje na kraj kolekcije. Moguće je da se u kolekciji nalazi više hotela s identičnim podacima (imenom, gradom, cijenom i brojem slobodnih soba).

- **int Izbaci(Hotel h)**

Briše sve hotele iz kolekcije s istim podacima kao i hotel **h**. Ostali hoteli zadržavaju isti poredak kao u kolekciji. Funkcija vraća broj izbrisanih hotela.

- **int BrojHotela(void)**

Vraća ukupan broj hotela u kolekciji.

- **Hotel ItiHotel(int i)**

Vraća **i**-ti hotel u kolekciji. Ovdje je **i** između 0 i **BrojHotela()** -1. Prvi hotel ima redni broj 0.

- **SkupHotela NadjiHoteleUGradu(string grad)**

Vraća novi skup hotela koji sadrži samo hotele koji se nalaze u gradu **grad**. Redoslijed hotela u novoj kolekciji mora odgovarati redoslijedu kojeg su ti hoteli imali u skupu koji je pozvao funkciju.

- **SkupHotela NadjiHotele(string grad, int maxCijena, int minSlobodnihMj)**

Vraća novi skup hotela koji sadrži samo hotele koji se nalaze u gradu **grad**, cijena noćenja im je manja ili jednaka od **maxCijena** i imaju minimalno **minSlobodnihMj** slobodnih mjesta. Redoslijed hotela u novoj kolekciji mora odgovarati redoslijedu kojeg su ti hoteli imali u skupu koji je pozvao funkciju.

- **int NajmanjaCijenaUGradu(string grad)**

Vraća najmanju cijenu noćenja u gradu **grad**. Ukoliko je skup hotela prazan treba vratiti -1.

- **void SortirajPoCijeni(void)**

Sortira hotele u skupu prema cijeni noćenja (od manje prema većoj). Ako postoji više hotela s istom cijenom, tada njihov međusobni redoslijed odgovara redoslijedu kojeg su imali prije poziva funkcije.

## Primjer klijentskog programa

```
#include "hoteli.h"
#include <iostream>
using namespace std;

void ispisHotel(Hotel h)
{
    cout << h.Ime() << " " << h.Grad() << " " << h.Cijena() << " " << h.BrojSlobodnihMjesta();
}

void ispisSkupHotela(SkupHotela hoteli)
{
    for(int i=0; i<hotel.BrojHotela(); i++)
    {
        ispisHotel(hoteli.ItiHotel(i));
        cout << endl;
    }
}

int main(void)
{
    SkupHotela test(Hotel("Hotel_1", "Zagreb", 500, 20));
    test.Dodaj(Hotel("Hotel_2", "Zagreb", 300, 30));
    test.Dodaj(Hotel("Hotel_3", "Zagreb", 250, 100));
    test.Dodaj(Hotel("Hotel_4", "Split", 350, 200));
    test.Dodaj(Hotel("Hotel_5", "Rijeka", 100, 150));
    test.Dodaj(Hotel("Hotel_6", "Split", 120, 140 ));
    test.Dodaj(Hotel("Hotel_3", "Zagreb", 250, 100));
    test.Dodaj(Hotel("Hotel_7", "Zagreb", 250, 25));

    ispisSkupHotela(test);
    // Hotel_1 Zagreb 500 20
    // Hotel_2 Zagreb 300 30
    // Hotel_3 Zagreb 250 100
    // Hotel_4 Split 350 200
    // Hotel_5 Rijeka 100 150
    // Hotel_6 Split 120 140
    // Hotel_3 Zagreb 250 100
    // Hotel_7 Zagreb 250 25

    cout << test.Izbaci(Hotel("Hotel_3", "Zagreb", 250, 100)) << endl;
    // 2

    ispisSkupHotela(test);
    // Hotel_1 Zagreb 500 20
    // Hotel_2 Zagreb 300 30
    // Hotel_4 Split 350 200
    // Hotel_5 Rijeka 100 150
    // Hotel_6 Split 120 140
    // Hotel_7 Zagreb 250 25

    test.SortirajPoCijeni();
    ispisSkupHotela(test);
    // Hotel_5 Rijeka 100 150
    // Hotel_6 Split 120 140
    // Hotel_7 Zagreb 250 25
    // Hotel_2 Zagreb 300 30
    // Hotel_4 Split 350 200
    // Hotel_1 Zagreb 500 20

    cout << test.NajmanjaCijenaUGradu("Zagreb") << endl;
    // 250

    ispisSkupHotela(test.NadjiHotele("Zagreb", 300, 25));
    // Hotel_7 Zagreb 250 25
    // Hotel_2 Zagreb 300 30

    return 0;
}
```

## Opće napomene

- Struktura, funkcije i datoteke koje šalžete moraju se zvati točno onako kako je zadano u zadatku. Pazite na mala i velika slova!
- Trebate poslati samo sučelje i implementaciju. U datotekama koje šalžete ne smije se nalaziti funkcija `main()`!
- nijedna funkcija ne smije ništa učitavati s tipkovnice ili neke datoteke, niti išta ispisivati na ekran ili u neku datoteku.
- Svaki od main-ova pomoću kojih testiramo ispravnost vašeg programa neće pozivati sve gore navedene funkcije. Stoga, ako neku od funkcija ne znate napisati ipak možete dobiti koji bod (u tom slučaju tu funkciju nemojte navesti niti u `.h` niti u `.cpp` datoteci ili napravite neku trivijalnu implementaciju).

Ispravnost implementacija koje napišete bit će provjerena tako da ćemo mi napisati razne klijentske programe koji će deklarirati nekoliko varijabli zadane strukture, i na njima pozivati funkcije koje ste trebali napisati. Ako se poslani programi ne budu uspješno povezivali (linkali) s našim klijentskim programima, smatrat će se neispravnima.

Neki klijentski programi provjeravat će samo neke jednostavnije funkcije, dok će neki provjeravati sve funkcije koje trebate napisati. Provjera je potpuno automatska, tako da je od presudne važnosti da se pridržavate specifikacije. Nepridržavanje lako može uzrokovati osvojenih 0 bodova iz zadaće!

Naravno, za provjeru radi li implementacija prije nego što je pošaljete, preporučuje se da je testirate pomoću nekog klijentskog programa. No taj klijentski program ne šalžete!

Za sva pitanja vezana uz ovu zadaću javite se asistentu Vujčiću na [jvujcic+rp1@gmail.com](mailto:jvujcic+rp1@gmail.com).