

Zadatak:

Napišite sučelje i implementaciju za strukture **Osoba** i **Imenik**. Struktura **Osoba** predstavlja osobu (ime, prezime, prebivalište, godina rođenja). Struktura **Imenik** predstavlja kolekciju podataka tipa **Osoba**. Ukupan broj osoba u kolekciji nikad neće biti veći od 100. Sučelje za obje strukture spremite u datoteku **imenik.h**, a implementaciju u datoteku **imenik.cpp**.

Osoba – Konstruktori i funkcije članice

- **Osoba(string ime, string prezime, string grad, int godRodjenja)**

Konstruktor za strukturu Osoba.

ime – ime osobe

prezime – prezime osobe

grad – prebivalište osobe

godRodjenja – godina rođenja osobe

- **string Ime(void)**

Vraća ime osobe.

- **string Prezime(void)**

Vraća prezime osobe.

- **string Grad(void)**

Vraća prebivalište osobe.

- **int GodinaRodjenja(void)**

Vraća godinu rođenja osobe.

Imenik – Konstruktori i funkcije članice

- **Imenik()**

Konstruktor za strukturu Imenik. Stvara praznu kolekciju osoba.

- **Imenik(Osoba o)**

*Konstruktor za strukturu **Imenik**. Stvara kolekciju koja se sastoji samo od osobe **o**.*

- **void Dodaj(Osoba o)**

*Dodaje osobu **o** u kolekciju. Osoba **o** se uvijek dodaje na kraj kolekcije. Moguće je da se u kolekciji nalazi više osoba s istim podacima (imenom, prezimenom, prebivalištem i godinom rođenja).*

- **int Izbrisi(Osoba o)**

*Briše sve osobe iz kolekcije s istim podacima kao i osoba **o**. Ostale osobe zadržavaju isti poredak u kolekciji. Funkcija vraća broj izbrisanih osoba.*

- **int BrojOsoba(void)**

Vraća ukupan broj osoba u imeniku.

- **Osoba ItaOsoba(int i)**

*Vraća **i**-tu osobu u imeniku. Ovdje je **i** broj između 0 i BrojOsoba()-1. Prva osoba ima redni broj 0.*

- **Imenik NadjiPrezime(string prezime)**

*Vraća novi imenik koji sadrži samo osobe s prezimenom **prezime**. Redoslijed osoba u novom imeniku mora odgovarati redoslijedu kojeg su te osobe imale u imeniku koji je pozvao funkciju.*

- **Imenik NadjiGrad(string grad)**

*Vraća novi imenik koji sadrži samo osobe koje žive u gradu **grad**. Redoslijed osoba u novom imeniku mora odgovarati redoslijedu kojeg su te osobe imale u imeniku koji je pozvao funkciju.*

- **Imenik NadjiGodinuRodjenja(int god)**

*Vraća novi imenik koji sadrži samo osobe koje su rođene **god** godine. Redoslijed osoba u novom imeniku mora odgovarati redoslijedu kojeg su te osobe imale u imeniku koji je pozvao funkciju.*

- **int NadjiNajmanjuGodinuRodjenja(void)**

Vraća najmanju godinu rođenja koja se pojavljuje u imeniku. Ukoliko je imenik prazan treba vratiti -1.

- **void SortirajPoPrezimenu(void)**

Sortira osobe u imeniku po prezimenu (od slova A prema slovu Z). Prvo slovo prezimena će uvijek biti veliko, a ostala slova mala. Ako postoji više osoba s istim prezimenom tada njihov međusobni redoslijed odgovara redoslijedu kojeg su imali prije poziva funkcije.

Primjer klijentskog programa

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

void ispisOsobe(Osoba o)
{
    cout << o.Ime() << " " << o.Prezime() << " " << o.Grad() << " " << o.GodinaRodjenja();
}

void ispisImenika(Imenik im)
{
    for(int i=0; i<im.BrojOsoba(); i++)
    {
        ispisOsobe(im.ItaOsoba(i));
        cout << endl;
    }
}

int main(void)
{
    Imenik testImenik(Osoba("Mate", "Matic", "Rijeka", 1984));
    testImenik.Dodaj(Osoba("Ivica", "Ivic", "Zagreb", 1990));
    testImenik.Dodaj(Osoba("Josip", "Josipovic", "Split", 1980));
    testImenik.Dodaj(Osoba("Ante", "Matic", "Osijek", 1991));
    testImenik.Dodaj(Osoba("Igor", "Matijasevic", "Rijeka", 1950));
    testImenik.Dodaj(Osoba("Karlo", "Matijasevic", "Split", 1970));
    testImenik.Dodaj(Osoba("Ivica", "Ivic", "Rijeka", 1966));
    testImenik.Dodaj(Osoba("Josip", "Josipovic", "Split", 1980));

    ispisImenika(testImenik);
    // Mate Matic Rijeka 1984
    // Ivica Ivic Zagreb 1990
    // Josip Josipovic Split 1980
    // Ante Matic Osijek 1991
    // Igor Matijasevic Rijeka 1950
    // Karlo Matijasevic Split 1970
    // Ivica Ivic Rijeka 1966
    // Josip Josipovic Split 1980

    cout << testImenik.NadjiNajmanjuGodinuRodjenja() << endl;
    // 1950

    testImenik.SortirajPoPrezimenu();
    ispisImenika(testImenik);
    // Ivica Ivic Zagreb 1990
    // Ivica Ivic Rijeka 1966
    // Josip Josipovic Split 1980
    // Josip Josipovic Split 1980
    // Mate Matic Rijeka 1984
    // Ante Matic Osijek 1991
    // Igor Matijasevic Rijeka 1950
    // Karlo Matijasevic Split 1970

    cout << testImenik.Izbrisi(Osoba("Josip", "Josipovic", "Split", 1980)) << endl;
    // 2

    ispisImenika(testImenik);
    // Ivica Ivic Zagreb 1990
    // Ivica Ivic Rijeka 1966
    // Mate Matic Rijeka 1984
    // Ante Matic Osijek 1991
    // Igor Matijasevic Rijeka 1950
    // Karlo Matijasevic Split 1970

    ispisImenika(testImenik.NadjiGrad("Rijeka").NadjiPrezime("Ivic"));
    // Ivica Ivic Rijeka 1966

    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.