# Programiranje 3 – Ispit

Kreirati na desktopu folder sa svojim imenom i prezimenom. Zadatak kopirajte u tekstualni dokument pod nazivom **ImePrezime** i spremite u prethodno kreirani folder. Pristup internetu nije dozvoljen. Komuniciranje sa drugim polaznicima koji rade ispit nije dozvoljeno. Ukoliko polaznik prekrši neko od navedenih pravila biti će diskvalifikovan sa ispita bez mogućnosti izlaska na popravni ispit**. Za sva dodatna pitanja i nejasnoće obratiti se prisutnom osoblju.**

**Napomena: Dozvoljeno je mijenjati pojedine funkcije i operatore, kao i dodavati svoje. Main funkciju imate gotovu ali je smijete mijenjati.**

#include<iostream>

using namespace std;

template<class T1, class T2, int max>

class Kolekcija {

public:

T1 \* \_elementi1;

T2 \* \_elementi2;

int \_trenutnoElemenata;

//konstruktor i destruktor funkcije

//operator =

//1. AddElement :: omogucava dodavanje novog elementa u kolekciju. Onemoguciti

// ponavljanje elemenata (ukoliko se pojavljuju isti elementi baciti izuzetak tipa exception).

//Voditi računa ako je \_trenutnoElemenata dostiglo max, te i u tom slučaju baciti izuzetak.

//2. RemoveElement :: na osnovu lokacije uklanja element iz kolekcije.

// Prilikom uklanjanja elemenata ocuvati redoslijed njihovog dodavanja. Ukoliko lokacija nije validna, baciti

//izuzetak tipa exception

//operator <<

int getTrenutno() { }

T2 &getDrugi(int i) { }

T1 &getPrvi(int i) { }

};

class Datum {

int \*\_dan, \*\_mjesec, \*\_godina;

public:

//konstruktor i destruktor funkcije

//operator =

//operator <

//operator ==

//operator <<

};

class Pregled {

char \* \_dijagnoza;

char\* \_terapija;

Datum \_datumPregleda;

public:

//konstruktor i destruktor funkcije

//operator =

//operator == (dvije verzije, jedna verzija prima objekat tipa Pregled, druga prima terapiju)

//operator <<

Datum getDatum() {

}

};

class Pacijent {

protected:

char \*\_imePrezime;

//tip char\* je za podatke o imenu i prezimenu ljekara koji je obavio pregled, a

//objekat tipa Pregled za podatke o pregledu

Kolekcija<char\*, Pregled , 2> \_preglediPacijenta;

public:

//konstruktor i destruktor funkcije

//operator =

//operator [] vraca pokazivač na zadnji Pregled kod koga je postavljena terapija proslijedjena kao parametar.

//Ako terapija ne postoji niti za jedan pregled, baciti izuzetak tipa exception.

//funkciju AddPregled koja dodaje podatke o novom pregledu pacijenta. Dakle prima dva parametra: pregled i ljekar. Nije dozvoljeno dodati dva

//identicna pregleda. U zavisnosti od uspjesnosti operacije, funkcija vraca true ili false

//operator -= na osnovu primljenog parametra (tipa Datum) uklanja podatke o prvom pregledu

//koji je obavljen tog datuma. U zavisnosti od uspjesnosti operacije, funkcija vraca true

// ili false

//operator<<

};

void main() {

try {

Datum d1(1, 1, 2018), d2(2, 2, 2019), d3(15, 3, 2019);

Pregled p1("Dijagnoza 1", "Terapija 1", d1);

Pregled p2("Dijagnoza 2", "Terapija 2", d2);

Pregled p3("Dijagnoza 3", "Terapija 3", d3);

Pacijent pacijent("Test Test");

char lj1[] = "Ljekar1 Ljekar1";

char lj2[] = "Ljekar2 Ljekar2";

pacijent.AddPregled(p1,lj1);

pacijent.AddPregled(p2,lj1);

pacijent.AddPregled(p3,lj2);//izuzetak (max dostigao trenutnoElemenata)

pacijent.AddPregled(p1, lj1);//izuzetak isti elementi

cout << "\n::::::: PACIJENT 1. PUT :::::::::::::\n";

cout << pacijent << endl;

char terapija[] = "Terapija 1";

Pregled \*trazeniPregled=pacijent[terapija];

cout << "TRAŽENI PREGLED JE: \n";

cout << \*trazeniPregled << endl;

pacijent -= d1;//uklanjanje pregleda p1

cout << "::::::: PACIJENT 2. PUT :::::::::::::\n";

cout << pacijent << endl;

trazeniPregled = pacijent[terapija];//pokusaj ispisa pregleda sa terapijom koji smo sada uklonili

cout << "\nTRAŽENI PREGLED JE: \n";

cout << \*trazeniPregled << endl;

}

catch (exception &e) {

cout << e.what() << endl;

}

system("pause>0");

}