#include <iostream>

using namespace std;

class Izuzetak :public exception {

char\* \_datum;

int \_linija;

public:

//konstruktor, destruktor

// operator <<

};

template<class T1, class T2, int max>

class Kolekcija {

T1 \* \_elementi1[max];

T2 \* \_elementi2[max];

int \* \_trenutno;

public:

//konstruktor, destruktor

// operator =

void AddElement(T1 prvi, T2 drugi) {

}

int GetMax() { return max; }

int GetTrenutno() { return \*\_trenutno; }

T1 &GetElement1(int i) {

}

T2 &GetElement2(int i) {

}

//operator <<

};

class Datum {

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

public:

//konstruktor, destruktor

//operator=

//operator ==

//operator <

//operator <<

};

class Kurs {

char\* \_naziv;

Datum \* \_pocetak;

Datum \* \_kraj;

char \* \_imePredavaca;

public:

//konstruktor, destruktor

// operator =

// operator ==

// operator <

//operator <<

Datum getPocetak() { return \*\_pocetak; }

Datum getKraj() { return \*\_kraj; }

};

class Polaznik {

char \* \_imePrezime;

Kurs\* \_pohadjaniKursevi;

int trenutnoKurseva;

int max;

public:

//konstruktor, destruktor funkcije

//operator =

void AddKurs(Kurs novi) {

}

void RemoveKurs(Kurs ukloni)

{

}

// operator ==

//operaotor <<

bool polagaoKodPredavaca(char \*predavac) {

//vraća true ako je polaznik pohadjao barem jedan kurs kod proslijedjenog predavača

}

};

class SkillsCentar {

char\* \_nazivCentra;

Kolekcija<Kurs, Polaznik, 150> \_aplikanti;

public:

//konstruktor, destruktor funkcije

//operator =

void AddAplikaciju(Kurs k, Polaznik p)

{

//ONEMOGUCITI APLICIRANJE ZA ISTI KURS

}

void DodajUspjesnoOkoncanKurs(Polaznik p, Kurs k)

{

//Polazniku u njegov niz kurseva \_pohadjaniKursevi, dodaje proslijeđeni kurs

}

//operator <<

};

void main()

{

//NAPOMENA: Komentarisati pojedine dijelove main-a kako bi se testirali

//različiti slučajevi bacanja izuzetaka

Datum d1(12, 12, 2018), d2(12, 2, 2018), d3(14, 5, 2019), d4(16, 6, 2018);

Kurs k1("Programiranje 1", "Test Test", d1, d2 );

Kurs k2("Programiranje 3", "Test Test", d2, d3);

Kurs k3("Grafički dizajn", "Niko Nikić", d3, d4);

Polaznik p1("Miro Miric",5);

Polaznik p2("Kralj Fahd", 2);

SkillsCentar KCKF("Kulturni Centar Kralj Fahd");

try {

KCKF.AddAplikaciju(k1, p1);

KCKF.AddAplikaciju(k2, p1);

//KCKF.AddAplikaciju(k2, p1);//bacanje izuzetka jer je polazniku p1 već dodat kurs k2

KCKF.AddAplikaciju(k1, p2);

KCKF.AddAplikaciju(k3, p2);

KCKF.AddAplikaciju(k2, p2);

//KCKF.DodajUspjesnoOkoncanKurs(p1, k3);//treba da baci izuzetak jer p1 nije pohadjao kurs k3

KCKF.DodajUspjesnoOkoncanKurs(p1, k1);

//KCKF.DodajUspjesnoOkoncanKurs(p2, k3);

//KCKF.DodajUspjesnoOkoncanKurs(p2, k2);

KCKF.DodajUspjesnoOkoncanKurs(p2, k1);//bacanje izuzetka jer p2 moze imati max 2 kursa

cout << KCKF << endl;

}

catch (Izuzetak &ex) {

cout << ex << endl;

}

catch (exception &ex) {

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

}

system("pause");

}