1.

#include<iostream>

using namespace std;

class Predmet

{

char\* \_naziv;

int \_godinaStudija;

public:

//Potrebne konstruktor i destruktor funkcije

// operator =

// operator ==

//operator <<

};

class Prisustvo

{

char\* \_student; //ne smije sadržavati brojeve, regex: [0-9]

char\* \_brojIndeksa;

int \_brojSati;

public:

//Potrebne konstruktor i destruktor funkcije

// operator =

//operator ==

// operator <<

};

class Nastava {

//Oznaka prostorije

char\* \_prostorija;

char \*\_datum;

char\* \_satnica;

char\* \_predavac;

//P, V ili W

Predmet\* \_predmet;

char \_tipNastave;

int\* \_odrzanoSati;

Prisustvo \_prisutni[50];

public:

int \_trenutnoPrisutnih;

//Potrebne konstruktor i destruktor funkcije

// operator <<

//Preklopljeni operator+= za dodavanje objekata tipa Prisustvo u niz \_prisutni

//Onemogućiti dodavanje istog prisustva

//Preklopljeni operator-= za uklanjanje objekata tipa Prisustvo iz niza \_prisutni

//Preklopljeni operator [] koji na osnovu parametra broj indeksa provjerava da li je

//odreðeni student prisustvovao nastavi

bool operator [](const char brIndeksa[]) {

}

};

void main()

{

Prisustvo p1("Test Test", "IB111", 2);

Prisustvo p2("Niko Nikic", "IB222", 4);

Prisustvo p3("Miro Miric", "IB333", 6);

Predmet predmet("Programiranje 3", 2);

Nastava n("AMF1", "15.02.2019", "14:00-17:00", "Denis Music", predmet, 'P', 3);

n += p1;

n += p2;

n += p1;

n -= p3;

bool daLiJePrisustvovao = n["IB4444"];

if (daLiJePrisustvovao)

cout << "IB4444 je prisustvovao nastavi\n";

else

cout << "IB4444 nije prisustvovao nastavi\n";

cout << n << endl;

system("pause>0");

}

2.

#include <iostream>

using namespace std;

class PolozeniPredmet {

char\* Naziv;

char\* Profesor;

int ocjena;

public:

//potrebne konstruktor i destruktor funkcije

//operator =

//operator ==

//operator <<

void PredmetInfo() {

}

int getOcjena() { }

};

class Clanstvo {

char brojKartice[10];

bool ClanarinaPlacena;

int dug;

public:

//potrebne konstruktor i destruktor funkcije

//operator =

int uplataClanarine(int vrijednost) {

//umanjuje dug za proslijeđenu vrijednost

//ispisati korisniku poruku o tome koliko je duga ostalo,

//a ukoliko je citav dug isplacen takodjer ispisati adekvatnu poruku

//ukoliko je vrijednost veća od duga, ispisati iznos kusura

}

void ClanstvoInfo() {

}

};

class Osoba {

char\* imePrezime;

char JMBG[13];

char Pol[1];

public:

//potrebne konstruktor i destruktor funkcije

virtual void Info()

{

cout << "Ime i prezime: " << imePrezime << endl;

cout << "JMBG: " << JMBG << endl;

cout << "Pol: " << Pol << endl;

}

};

class Student : public Osoba {

int index;

int GodStudija;

PolozeniPredmet\* polozeni[40];

int BrojPolozenih;

public:

//potrebne konstruktor i destruktor funkcije

void dodajPolozeni(char\* prof, char\* n, int ocj)

{

//voditi racuna da se ne dodaju dva ista polozena predmeta

}

void ukloniPolozeni(int lokacija)

{

}

//operator[](int index) koji uklanja polozeni predmet sa proslijeđene pozicije

//voditi racuna o tome da li je proslijeđena pozicija validna

int pretraga(int ocjena)

{

//vraća broj poozenih predmeta čija je ocjena jednaka proslijeđenoj

}

};

void main()

{

Student\* s = new Student(1489, 2, "Test Test", "1921680244", "M", "12345");

s->dodajPolozeni("Niko", "SOS", 10);

s->dodajPolozeni("Miro", "Pro", 10);

s->dodajPolozeni("Neko", "SOS", 10);

s->pretraga(10);

s->Info();

s->pretraga(8);

PolozeniPredmet\* p = (\*s)[0];

if(p!=nullptr)

p->PredmetInfo();

s->ukloniPolozeni(1);

s->ukloniPolozeni(6);

s->Info();

system("pause>0");

}

2.

#include <iostream>

#include <vector>

#include <iterator>

#include <string>

using namespace std;

class Datum {

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

public:

//konstruktor, konstruktor kopije, destruktor

//operator =

//operator ==

//operator <<

};

class ObavezeNaPredmetu {

char \*\_nazivObaveze; //seminarski rad; 1.parcijalni; 2.parcijalni; integralni

Datum \* \_datumIzvrsenja;

int \_ocjena; // 5 – 10

public:

//konstruktor, konstruktor kopije, destruktor

//operator =

//operator == (dvije verzije - jedna prima objekat tipa ObavezeNaPredmetu, druga samo char \*nazivObaveze)

//operator <<

};

class Profesor {

char \*imePrezime;

char \*titula;

public:

//konstruktor, konstruktor kopije, destruktor

//operator =

//operator == (dvije verzije - jedna prima objekat tipa ObavezeNaPredmetu, druga samo char \*nazivObaveze)

//operator <<

//operator >>

char \*GetImePrezime() { return imePrezime; }

};

class Predmet {

protected:

char \*\_nazivPredmeta;

Profesor \*\_profesoriNaPredmetu;

public:

int \_max;

int \_trenutno;

//konstruktor, konstruktor kopije, destruktor

//operator =

//operator == (dvije verzije - jedna prima objekat tipa ObavezeNaPredmetu, druga samo char \*nazivObaveze)

/\* 1. operator+= :: omogucava dodavanje novog profesora na predmet.

Prilikom dodavanja onemoguciti da se jednom predmetu dodaju dva profesora sa istim imenom. U

zavisnosti od uspjesnosti operacije funckija vraca true ili false.

Omogućiti proširenje niza, ukoliko je broj profesora dostigao max

2. operator-= :: na osnovu parametra tipa char\* uklanjanja ime profesora sa liste profesora na

predmetu. U zavisnosti od uspjesnosti operacije funckija vraca true ili false. Unutar funkcije koristiti iteratore\*/

bool PredavaoProfesor(char \*prof) {

//vraća true ili false u zavisnosti od toga da li je profesor sa proslijedjenim imenom i prezimenom

//predavao na trenutnom predmetu

}

virtual void Info() {

}

};

class PolozeniPredmet : public Predmet {

Datum \_datumPolaganja;

ObavezeNaPredmetu \_listaIzvrsenihObaveza[10];

int \_konacnaOcjena; //formira se na osnovu ocjena izvrsenih obaveza

public:

int \_trenutno;

//konstruktor, konstruktor kopije, destruktor

//operator =

//operator == (dvije verzije - jedna prima objekat tipa ObavezeNaPredmetu, druga samo char \*nazivObaveze)

/\*1. FormirajKonacnuOcjenu :: je zaduzena za formiranje konacne ocjene na polozenom predmetu.Svi

predmeti trebaju imati najmanje tri izvrsene obaveze, od kojih jedna mora biti seminarski rad.

Funkcija vraca novoformiranu ocjenu.

2. DodajIzvrsenuObavezu :: na osnovu primljenog parametra dodaje obavezu u listu izvresnih

obaveza. Onemoguciti dodavanje identicnih obaveza.

3. ProfesorNaPredmetu :: funkcija koja vraca vrijednost true ili false u zavisnosti od toga da li

je trazeni profesor predavao na polozenom predmetu ili ne\*/

int FormirajKonacnuOcjenu() {

if (\_trenutno < 3) {

\_konacnaOcjena = 5;

return \_konacnaOcjena;

}

bool provjera = false;

for (int i = 0; i < \_trenutno; i++) {

if (\_listaIzvrsenihObaveza[i] == "seminarski")

provjera = true;

}

if (provjera == false) {

\_konacnaOcjena == 5;

return \_konacnaOcjena;

}

for (int i = 0; i < \_trenutno; i++)

\_konacnaOcjena += \_listaIzvrsenihObaveza[i].getOcjena();

\_konacnaOcjena /= \_trenutno;

return \_konacnaOcjena;

}

void DodajIzvrsenuObavezu(ObavezeNaPredmetu obaveza) {

if (\_trenutno >= 10) {

cout << "Nemoguće dodati više od 10 obaveza na predmetu.\n";

return;

}

for (int i = 0; i < \_trenutno; i++)

{

if (\_listaIzvrsenihObaveza[i] == obaveza) {

cout << "Obaveza već postoji!"<<endl;

return;

}

}

\_listaIzvrsenihObaveza[\_trenutno] = obaveza;

\_trenutno++;

cout << "Obaveza uspješno dodata!\n";

}

bool ProfesorNaPredmetu(char \*prof) {

for (int i = 0; i < \_trenutno; i++)

{

if (this->PredavaoProfesor(prof) == true)

return true;

}

return false;

}

};

void main() {

PolozeniPredmet prvi("PR3", Datum(27, 11, 2015));

ObavezeNaPredmetu prva("seminarski", Datum(29, 12, 2015), 9);

ObavezeNaPredmetu druga("I parcijalni", Datum(23, 12, 2015), 7);

ObavezeNaPredmetu treca("II parcijalni", Datum(1, 1, 2016), 8);

Profesor p1("Denis Music", "prof");

Profesor p2("Marija Herceg", "asistent");

Profesor p3("Indira Hamulic", "asistent");

prvi += p1;

prvi += p2;

prvi += p2;

prvi.DodajIzvrsenuObavezu(prva);

prvi.DodajIzvrsenuObavezu(druga);

if (prvi.ProfesorNaPredmetu("Marija Herceg"))

cout << "Marija je predavala na PR3" << endl;

prvi -= ("Indira Hamulic");

if (prvi.ProfesorNaPredmetu("Indira Hamulic") == false)

cout << "Indira nije predavala na PR3" << endl;

prvi.Info();

system("pause>0");

1. }