1.

#include <iostream>

using namespace std;

class Proizvod {

private: char \*Naziv;

private: char \*Proizvodjac;

private: float Cijena;

private: int Kolicina;

public: void Unos() {

char temp[100];

cout << "Unesite naziv proizvoda: ";

cin.getline(temp, 100);

int duzina = strlen(temp) + 1;

Naziv = new char[duzina];

strcpy\_s(Naziv, duzina, temp);

cout << "Unesite proizvođača: ";

cin.getline(temp, 100);

duzina = strlen(temp) + 1;

Proizvodjac = new char[duzina];

strcpy\_s(Proizvodjac, duzina, temp);

cout << "Unesite cijenu proizvoda: ";

cin >> Cijena;

cout << "Unesite količinu: ";

cin >> Kolicina;

cin.ignore();

}

public: void Ispis() {

cout << "Proizvod: " << Naziv << "\nProizvođač: " << Proizvodjac << endl;

cout << "Cijena: " << Cijena << " KM\nKoličina: " << Kolicina << endl;

}

};

void main() {

//Unijeti podatke o tri proizvoda, te ih ispisati.

int n = 3;

Proizvod \*proizvodi = new Proizvod[n];

for (int i = 0; i < n; i++)

{

proizvodi[i].Unos();

}

system("cls");

for (int i = 0; i < n; i++)

{

proizvodi[i].Ispis();

}

system("pause>0");

}

2.

#include <iostream>

using namespace std;

class Apartman {

private: char \*Naziv;

private: char \*Mjesto;

private: float Cijena;

private: int BrojKreveta;

public: void Unos() {

char temp[100];

cout << "Unesite naziv apartmana: " << endl;

cin.getline(temp, 100);

int duzina = strlen(temp) + 1;

Naziv = new char[duzina];

strcpy\_s(Naziv, duzina, temp);

cout << "Unesite mjesto: " << endl;

cin.getline(temp, 100);

duzina = strlen(temp) + 1;

Mjesto = new char[duzina];

strcpy\_s(Mjesto, duzina, temp);

cout << "Unesite cijenu: ";

cin >> Cijena;

cout << "Unesite broj kreveta: ";

cin >> BrojKreveta;

cin.ignore();

}

public: void Info() {

cout << "Apartman: " << Naziv << endl;

cout << "Mjesto: " << Mjesto << endl;

cout << "Cijena " << Cijena << endl;

cout << "Broj kreveta: " << BrojKreveta << endl;

}

public: float GetCijena() {

//vraća cijenu apartmana

return Cijena;

}

public: bool IsFromMostar() {

//funkcija koja vraća informaciju da li je apartman iz Mostara

return (strcmp(Mjesto, "Mostar") == 0);

}

};

float GetProsjecnaCijena(Apartman \*niz, int vel) {

//Funkcija koja računa prosječnu cijenu apartmana

float prosjecna = 0;

if (vel == 0)

return 0;

for (int i = 0; i < vel; i++)

prosjecna += niz[i].GetCijena();

return prosjecna / float(vel);

}

int GetBrojApartmanaInMostar(Apartman \*niz, int vel) {

//Funkcija koja vraća koliko je apartmana iz Mostara

int br = 0;

for (int i = 0; i < vel; i++)

{

if (niz[i].IsFromMostar())

br++;

}

return br;

}

void GetNajjeftinijiApartman(Apartman niz[], int vel) {

int min = INT\_MAX, pozicija = -1;

for (int i = 0; i < vel; i++)

{

if (niz[i].GetCijena() < min)

{

pozicija = i;

min = niz[i].GetCijena();

}

}

if (pozicija == -1)

cout << "Ne postoji nijedan apartman.\n";

else {

cout << "Najjeftiniji apartman je: \n";

niz[pozicija].Info();

cout << endl;

}

}

void main() {

int n;

cout << "Koliko apartmana želite dodati? ";

cin >> n;

cin.ignore();

Apartman \*apartmani = new Apartman[n];

for (int i = 0; i < n; i++)

apartmani[i].Unos();

system("cls");

for (int i = 0; i < n; i++) {

apartmani[i].Info();

cout << "===========================================\n\n";

}

cout << "Prosječna cijena apartmana iznosi: " << GetProsjecnaCijena(apartmani, n) << " KM\n";

cout << "Broj apartmana u Mostaru: " << GetBrojApartmanaInMostar(apartmani, n) << endl;

GetNajjeftinijiApartman(apartmani, n);

system("pause>0");

}

3.

#include<iostream>

using namespace std;

class Pregled {

public: char \*Dijagnoza;

private: double cijenaPregleda;

public: void Unos() {

char temp[100];

cout << "Unesite dijagnozu: ";

cin.getline(temp, 100);

int duzina = strlen(temp) + 1;

Dijagnoza = new char[duzina];

strcpy\_s(Dijagnoza, duzina, temp);

cout << "Unesite cijenu pregleda: ";

cin >> cijenaPregleda;

cin.ignore();

}

public: void Ispis() {

cout << Dijagnoza << " (" << cijenaPregleda << " KM)\n";

}

public: void Dealociraj() {

delete[]Dijagnoza;

Dijagnoza = NULL;

}

public: double GetCijenaPregleda() {

return cijenaPregleda;

}

};

class Pacijent {

private: char \*imePrezime;

private: Pregled \* preglediPacijenata[10];

private: int brojPregleda;

public:void Unos() {

char temp[100];

cout << "Unesite ime i prezime pacijenta: ";

cin.getline(temp, 100);

int duzina = strlen(temp) + 1;

imePrezime = new char[duzina];

strcpy\_s(imePrezime, duzina, temp);

brojPregleda = 0;

}

public: void Ispis() {

//Ispis osnovnih informacija o pacijentu, te o njegovim pregledima

cout << "Pacijent: " << imePrezime << endl;

for (int i = 0; i < brojPregleda; i++)

{

preglediPacijenata[i]->Ispis();

}

}

public: void DodajPregled() {

//Dodaje novi pregled za pacijenta

if (brojPregleda >= 10) {

cout << "Ne možete dodati više od 10 pregleda!" << endl;

}

else {

preglediPacijenata[brojPregleda] = new Pregled;

preglediPacijenata[brojPregleda]->Unos();

brojPregleda++;

}

}

public: double GetUkupneTroskove() {

//Funkcija vraća ukupnu cijenu svih pregleda pacijenta

double sum = 0;

for (int i = 0; i < brojPregleda; i++)

{

sum += preglediPacijenata[i]->GetCijenaPregleda();

}

return sum;

}

public:void Dealociraj() {

delete[]imePrezime;

imePrezime = NULL;

for (int i = 0; i < brojPregleda; i++)

{

preglediPacijenata[i]->Dealociraj();

delete preglediPacijenata[i];

preglediPacijenata[i] = NULL;

}

}

public: bool HasDijagnoza(char \*dijagnoza) {

//vraća informaciju da li pacijent ima dijagnozu jednaku proslijeđenoj

for (int i = 0; i < brojPregleda; i++)

{

if (strcmp(preglediPacijenata[i]->Dijagnoza, dijagnoza) == 0)

return true;

}

return false;

}

};

void PretragaPoDijagnozi(Pacijent \*pacijenti, int vel) {

//Vrši pretragu pregleda svih pacijenata po dijagnozi

//Ispisuje podatke pacijenta koji ima unesenu dijagnozu

//Unutar funkcije tražiti od korisnika da unese dijagnozu koju pretražuje

//te prema tome ispisati podatke pacijenata koji imaju unesenu dijagnozu

char temp[100];

cout << "Unesite dijagnozu koju pretražujete: ";

cin.getline(temp, 100);

cout << "Pacijenti sa dijagnozom: " << temp << "su:\n\n";

for (int i = 0; i < vel; i++)

{

if (pacijenti[i].HasDijagnoza(temp))

pacijenti[i].Ispis();

}

}

void main() {

int vel;

cout << "Unesite broj pacijenata koje zelite dodati:";

cin >> vel;

cin.ignore();

Pacijent \*pacijenti = new Pacijent[vel];

for (int i = 0; i < vel; i++)

{

pacijenti[i].Unos();

for (int j = 0; j < 2; j++)//za svakog pacijenta se dodaju po dva pregleda

pacijenti[i].DodajPregled();

}

PretragaPoDijagnozi(pacijenti, vel);

cout << "Ukupni troškovi: \n";

for (int i = 0; i < vel; i++)

{

cout<<i+1<<". pacijent: "<<pacijenti[i].GetUkupneTroskove()<<endl;

}

for (int i = 0; i < vel; i++)

{

pacijenti[i].Dealociraj();

}

delete[]pacijenti;

pacijenti = NULL;

system("pause>0");

}