

Колоквиумски задачи - Група 5

1. Да се дефинира класа **Pacient** за кој ќе се чуваат информации за:

- име на пациентот (динамички алоцирана низа од знаци),
- матичен број - ЕМБГ (низа од 13 знаци),
- број на посети на ординација (ненегативен цел број). (5 поени)

За потребите на оваа класа да се преоптоварат потребните конструктори и да се напише соодветен деструктор. (10 поени)

Дополнително за оваа класа да се преоптоварат:

- операторот за споредување $<$ (пациентите се споредуваат според бројот на посети на ординација) (5 поени),
- операторот за проследување на излезен тек « кој ќе го печати ЕМБГ, името и бројот на посети на ординацијата од страна на пациентот. (5 поени)

Дополнително да се дефинира класа **Ordinacija** за ординација која има свои редовни пациенти, во која се чуваат:

- име на ординацијата (низа од 30 знаци),
- низа од **Pacient** објекти (динамички алоцирана низа од објекти од класата **Pacient**),
- вкупен број на пациенти (ненегативен цел број) (5 поени).

За потребите на оваа класа да се напишат соодветен конструктор и деструктор. (10 поени)

Да се преоптовари операторот $+=$ за додавање нов **Pacient** на ординација. (10 поени)

Во оваа класа да се напишат методи **najdiPacient(char* embg)** кој ќе го отпечати на екран пациентот со матичен број еднаков на матичниот број проследен како аргумент на методот и **najredovenPacient()** кој како резултат ќе го враќа пациентот кој најмногу пати ја посетил ординацијата. (10 поени)

Сите атрибути во класата се приватни.

2. Да се дефинира класа **Prostorija**, во која се чуваат информации за:

- шифра (низа од максимум 10 знаци),
- кат на кој се наоѓа просторијата (ненегативен цел број),
- површина во m^2 (децимален број). (5 поени)

Од оваа класа да се изведат две нови класи **Predavalna** и **Laboratorija**. За класата **Predavalna** дополнително се чува цена на изнајмување на квадратен метар за еден месец (цел број). За класата **Laboratorija** дополнително се чува бројот на компјутери (позитивен цел број). За секоја од класите да се напише соодветен конструктор. (10 поени)

Еден факултет има потреба да изнајмува простории. Да се напишат соодветни методи за пресметување на цената на изнајмување на просториите:

- за предавалните се пресметува како производ од површината на просторијата и цената по квадратен метар,
пр. $20m^2 * 50$ (цена по квадратен метар) = 1000 (5 поени)
- за лабораториите се пресметува како $200 \times$ бројот на компјутери ако бројот на компјутери е под 20 или како $300 \times$ бројот на компјутери ако бројот на компјутери е над 20 (вклучително и 20).
пр. за лабораторија со 15 сметачи: $200 * 15 = 3000$ (10 поени)

Да се преоптовари операторот проследување на излезен тек << за просторија кој ќе ги печати на екран шифрата, катот на кој се наоѓа просторијата, површината, како и цената на изнајмување. (10 поени)

Да се напише функција `najskapaProstorija` која ќе прима како аргументи низа од покажувачи од класата просторија и вкупен број на елементи, а ќе ја отпечати просторијата со највисока цена на изнајмување. (10 поени)

ПРОГРАМИ ЗА ТЕСТИРАЊЕ

1 задача

```
int main() {
    Pacient p1("Mitre Mitreski", "0305982450011", 15);
    Pacient p2("Petre Petreski", "0508985450022", 10);
    Pacient p3("Marina Marinova", "0201986455101", 9);
    Ordinacija o("Neuromedika");
    o += p1;
    o += p2;
    o += p3;
    cout << "Najredoven pacient e " << o.najredovenPacient() <<
        endl;
    cout << "Pretplatnik so matichen broj 0201986455101" << endl;
    o.najdiPacient("0201986455101");
    return 0;
}
```

2 задача

```
int main() {
    Prostorija *pole[] = { new Predavalna("P-117", 1, 49.5, 80),
        new Laboratorija("Lab-200", 2, 62.6, 45),
        new Predavalna("P-315", 3, 60.8, 70) };
    najskapaProstorija(pole, 3);
    for(int i = 0; i < 3; i++) {
        delete pole[i];
    }
    return 0;
}
```
