

Колоквиумски задачи - Група 4

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

- име на претплатникот (динамички алоцирана низа од знаци),
- матичен број (ЕМБГ) (низа од 13 знаци)
- број на различни списанија на кои што е претплатен (позитивен 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 == (два претплатника се исти доколку имаат ист ЕМБГ) (5 поени),
- операторот за проследување на излезен тек << кој ќе го печати ЕМБГ, името и бројот на различни списанија на кои е претплатен претплатникот. (5 поени)

Дополнително да се дефинира класа **Spisanie** за списание на кое се претплатуваат читатели, во која се чуваат:

- име на списанието (низа од 50 знаци),
- низа од **Pretplatnik** објекти (динамички алоцирана низа од објекти од класата **Pretplatnik**),
- вкупен број на претплатници (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 **Pretplatnik** на списание. (10 поени)

Во оваа класа да се напишат методи **najdiPretplatnik(char* embg)** кој ќе го отпечати на екран претплатникот со матичен број еднаков на матичниот број проследен како аргумент на методот и **najgolemFanNaSpisanija()** кој како резултат ќе го враќа претплатникот кој е претплатен на најмногу списанија. (10 поени)

Сите атрибути во класата се приватни.

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

- име (низа од максимум 40 знаци),
- година на раѓање (цел број),
- годишна заработувачка во долари (цел број). (5 поени)

Од оваа класа да се изведат две нови класи **Teniser** и **Rakometar**. За класата **Teniser** дополнително се чува пол (машки или женски) и процент на даночна стапка (цел број). За класата **Rakometar** дополнително се чува бројот на одиграни натпревари (цел број). За секоја од класите да се напише соодветен конструктор. (10 поени)

Секој спортист плаќа данок на заработувачка на годишно ниво. Да се напишат соодветни методи за пресметување данок:

- за тенисер се пресметува како процентот на даночната стапка од вкупната годишна заработувачка
пр. $200\,000\$ * 5\%$ (даночна стапка) = $10\,000\$$ (5 поени)
- за ракометар се пресметува како 8% од годишната заработувачка ако бројот на одиграни натпревари е над 30 или како 10% од годишната заработувачка ако бројот на одиграни натпревари е под 30 (вклучително и 30)
пр. за ракометар кој одиграл 25 натпревари: $10\,000\$ * 10\% = 1000\$$ (10 поени)

Да се преоптовари операторот проследување на излезен тек << за спортист кој ќе ги печати на екран името, годината на раѓање, заработувачката, како и данокот што треба да го плати спортистот. (10 поени)

Да се напише функција `najgolemDolzchnik` која ќе прима како аргументи низа од покажувачи од класата спортист, вкупен број на елементи и ќе го отпечати спортистот што треба да плати највисок данок. (10 поени)

ПРОГРАМИ ЗА ТЕСТИРАЊЕ

1 задача

```
int main() {
    Pretplatnik p1("Emil Stankov", "0305982450011", 5);
    Pretplatnik p2("Tomche Delev", "0508985450022", 10);
    Pretplatnik p3("Vesna Kirandzhiska", "0201986455101", 7);
    Spisanie s("Enter");
    s += p1;
    s += p2;
    s += p3;
    cout << "Najgolem fan na spisanija e "<< s.
         najgolemFanNaSpisanija()<< endl;
    cout << "Pretplatnik so matichen broj 0508985450022"<< endl;
    s.najdiPretplatnik("0508985450022");
    return 0;
}
```

2 задача

```
int main() {
    Sportist *pole[] = {new Teniser("Roger Federer", 1981,
                                   1000000, 'm', 7),
                       new Rakometar("Kiril Lazarov", 1980, 60000, 40),
                       new Teniser("Marija Sharapova", 1987, 800000, 'z', 5)};
    najgolemDolzchnik(pole, 3);
    for(int i = 0; i < 3; i++) {
        delete pole[i];
    }
    return 0;
}
```
