

Колоквиумски задачи - Група 6

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

- име на клиентот (динамички алоцирана низа од знаци),
- број на сметка (цел број),
- број на трансакции (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 $<$ (клиентите се споредуваат според бројот на трансакции) (5 поени),
- операторот за проследување на излезен тек $<<$ кој ќе го печати името, бројот на сметка и бројот на трансакции на клиентот. (5 поени)

Дополнително да се дефинира класа **Banka** за банка која има свои редовни клиенти, во која се чуваат:

- име на банката (низа од 30 знаци)
- низа од **Klient** објекти (динамички алоцирана низа од објекти од класата **Klient**),
- вкупен број на клиенти (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 **Klient** на банка. (10 поени)

Во оваа класа да се напишат методи **noviKlienti(int broj)** кој ќе ги печати на екран сите клиенти чии што број на сметка е поголем од бројот проследен како аргумент на методот и **najdobarKlient()** кој како резултат ќе го враќа клиентот кој извршил најголем број на трансакции. (10 поени)

Сите атрибути во класата се приватни.

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

- шифра (низа од максимум 10 знаци),
- професор (низа од максимум 40 знаци),
- учебна година (позитивен цел број поголем од 2000). (5 поени)

Од оваа класа да се изведат две нови класи **NRS** и **ProfesionalniVeshtini**. За класата **NRS** дополнително се чува број на кредити (децимален број). За класата **ProfesionalniVeshtini** дополнително се чува фонд на часови за предавања (позитивен цел број). За секоја од класите да се напише соодветен конструктор. (10 поени)

За секој курс се одредува цена за запишување (и слушање). Да се напишат соодветни методи за пресметување на цената на запишување на курсевите:

- за курсевите по НРС се пресметува како број на кредити $\times 300$ ако курсот е запишан пред 2005 година, односно како број на кредити $\times 200$ ако курсот е запишан по 2005 година (вклучувајќи ја и 2005 година)
пр. За курс по НРС запишан во 2004 година: 300×8 (број на кредити) = 2400 (5 поени)

- за курсевите по Професионални Вештини се пресметува како фонд на часови $\times 250$ ако курсот е запишан пред 2008 (вклучувајќи ја и 2008 година), односно 20% повеќе ако курсот е запишан по 2008 година (во однос на курс запишан пред 2008 година)
пр. За курс по Професионални Вештини запишан во 2007 година: 250×8 (фонд на часови) = 2000 (10 поени)

Да се преоптовари операторот проследување на излезен тек `<<` за курс кој ќе ги печати на екран шифрата, професорот кој го предава курсот, учебната година, како и цената на запишување. (10 поени)

Да се напише функција `najevtinKurs` која ќе прима како аргументи низа од покажувачи од класата курс и вкупен број на елементи, а ќе го отпечати курсот со најниска цена на запишување. (10 поени)

ПРОГРАМИ ЗА ТЕСТИРАЊЕ

1 задача

```
int main() {
    Klient k1("Mitre Mitreski", 1560, 10);
    Klient k2("Petre Petreski", 1950, 15);
    Klient k3("Marina Marinova", 1960, 9);
    Banka b("Stopanska Banka");
    b += k1;
    b += k2;
    b += k3;
    cout << "Najdobar klient e " << b.najdobarKlient() << endl;
    cout << "Novi klienti se:" << endl;
    b.noviKlienti(1500);
    return 0;
}
```

2 задача

```
int main() {
    Kurs *pole[] = { new NRS("NRS-2003", "A. Madevska", 2003, 8),
                    new ProfesionalniVeshtini("PRV-2010", "K. Zdravkova",
                    2010, 6),
                    new NRS("NRS-2012", "I. Chorbev", 2012, 8)};
    najevtinKurs(pole, 3);
    for(int i = 0; i < 3; i++) {
        delete pole[i];
    }
    return 0;
}
```
