

Kolokwium Testowe 1, Języki Programowania

Należy stworzyć klasę Czlowiek, zawierającą następujące pola składowe:

Czlowiek:

- std::string imie
- std::string nazwisko
- int* pesel //przechowująca numer PESEL (11 liczb, że każdy int jest cyfrą 0-9)
- publiczne pole statyczne ilosc (int)

Oraz zestaw metod:

konstruktor domyślny (domyślny właściciel to "John Smith, 000000000"), konstruktor z 3 parametrami, konstruktor kopiujący i destruktor, oraz:

- metody ustawiające i zwracające składniki klasy (metody typu "set" i „get”)
- operator<<, pozwalający na wypisanie wszystkich pól składowych na ekran
- operator[], umożliwiający zmianę/odzyskanie zadanej cyfry numeru pesel
- operator=, zgodnie ze zdrowym rozsądkiem
- do tego napisać funkcję (zewnętrzna) WypiszLudzi przyjmującą tablicę obiektów Czlowiek oraz ich liczbę, służącą do wypisywania owej podanej tablicy na ekran

Klasy powinny być skonstruowane w taki sposób, by działały z podanym niżej programem (do ściągnięcia ze strony:

http://www.if.pw.edu.pl/~majanik/data/JP/2017/Kolokwium/program_testowe1.cpp):

```
#include <iostream>
#include "Czlowiek.h"

using namespace std;

int main()
{
    cout<<Czlowiek::ilosc<<endl;
    Czlowiek cz0;
    cout<<cz0<<endl;

    int pesel[]={0,0,0,0,0,0,1,1,0,0,7};
    Czlowiek cz1("James", "Bond", pesel);
    cout<<cz1<<endl;

    Czlowiek *cz2 = new Czlowiek(cz1);
    cz2->SetNazwisko("Black");
    cout<<*cz2<<endl;

    int pesel2[]={2,2,2,2,2,2,5,5,3,3,3};
    cz1.SetPesel(pesel2);
    cout<<cz1[2]<<" "<<cz1[7]<<" "<<cz1[10];
    cout<<endl;

    Czlowiek tab[]={cz0, cz1, *cz2};
    WypiszLudzi(tab,3);
    cout<<Czlowiek::ilosc<<endl;

    return 0;
}
```

Co po uruchomieniu skutkowało by pojawieniem się na ekranie:

```
$ ./program
0
Czlowiek: John Smith: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
Czlowiek: James Bond: 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 7
Czlowiek: James Black: 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 7
2 5 3
Czlowiek: John Smith: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
Czlowiek: James Bond: 2, 2, 2, 2, 2, 2, 5, 5, 3, 3, 3
Czlowiek: James Black: 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 7
6
```

Plik nagłówkowy powinien zostać otoczony strukturą „ifndef”. Program powinien być napisany w 2 plikach (czlowiek.h, czlowiek.cpp oraz program.cpp z funkcją main()) oraz powinien się kompilować bez żadnych ostrzeżeń (flaga -Wall).

Program, oraz wszystkie pliki należy wysłać przed końcem zajęć na adres majanik@if.pw.edu.pl! Uwaga, za niekompilujący się program można uzyskać co najwyżej 40% punktów!

Kolokwium Testowe 1, Języki Programowania

Należy stworzyć klasę point, reprezentującą punkt w przestrzeni trójwymiarowej i zawierającą pola składowe:

- x, y, z (double)
- notatka (char*)
- pole statyczne ilosc (int)

Oraz zestaw metod: metodę zwracającą i ustawiającą notatkę (metodę typu "get" i "set"), konstruktor domyślny, konstruktor z 3 parametrami (x,y,z), konstruktor kopiujący i destruktor. Konstruktory prócz przypisywania wartości odpowiednim składnikom powinny zwiększać wartość pola ilość o 1, destruktor powinien zmniejszać wartość tego pola.

Należy również stworzyć metodę statyczną zwracającą wartość pola ilosc, oraz przeciążyć operator+ (zwracający point, będący sumą współrzędnych), a notatka powinna zostać ustawiona na "wynik sumy"). Oraz metodę Odleglosc(point& p) zwracającą odległość danego punktu od punktu podanego jako argument. Ponadto należy stworzyć metodę SetXYZ ustawiającą trzy parametry (x, y, z) jednocześnie oraz stworzyć metodę WypiszPunkt() która wypisuje na ekran informację o składnikach danego obiektu. Do tego napisać funkcję (nie funkcję składową) WypiszPunkty przyjmującą tablicę punktów oraz ich liczbę, służącą do wypisywania owej podanej tablicy na ekran.

Klasy powinny być skonstruowane w taki sposób, by działały z podanym niżej programem :
(program można ściągnąć ze strony: <http://www.if.pw.edu.pl/~majanik/data/JP/2014/kolokwium/mainpunkt.cpp>)

```
#include <iostream>
#include "point.h"
using namespace std;

int main()
{
    point p;
    p.WypiszPunkt(); cout<<endl;
    point *p2 = new point(2,3,4);
    p2->WypiszPunkt(); cout<<endl;
    p.SetXYZ(1,2,3);
    p.SetNotatka("punkt pierwszy");
    p2->SetNotatka("punkt drugi");
    p.WypiszPunkt(); cout<<endl;
    p2->WypiszPunkt(); cout<<endl;

    point p3 = p;
    p3 = p3 + *p2;
    p3.WypiszPunkt(); cout<<" , odleglosc miedzy punktami: "<<p3.Odleglosc(p)<<endl;
    cout<<"Liczba punktow: "<<point::GetIlosc()<<endl;
    delete p2;
    cout<<"Liczba punktow: "<<point::GetIlosc()<<endl;

    point tabP[2]; tabP[0].SetXYZ(6,6,6); tabP[1].SetXYZ(2,2,2);
    WypiszPunkty(tabP,2);
    return 0;
}
```

Co po uruchomieniu skutkowało by pojawieniem się na ekranie:

```
$ ./program
(0,0,0)
(2,3,4)
(1,2,3) punkt pierwszy
(2,3,4) punkt drugi
(3,5,7) wynik sumy, odleglosc miedzy punktami: 5.38516
Liczba punktow: 3
Liczba punktow: 2
(6,6,6) (2,2,2)
```

Plik nagłówkowy powinien zostać otoczony strukturą „ifndef”. Program powinien być napisany w 3 plikach (**point.cpp**, **point.h** oraz **program.cpp** z funkcją main()) oraz powinien się kompilować przy użyciu polecenia make (należy napisać odpowiedni Makefile).

Program należy wysłać przed końcem zajęć na adres majanik@if.pw.edu.pl