

## Kolokwium Testowe 1, gr. A, Języki Programowania, 19.11.2014

### Treść zadania

Należy stworzyć klasę Owoc zawierającą trzy pola składowe:

- nazwa(char \*) //zakładamy, że nazwa nie przekroczy 25 znaków, ale pamięć powinna być alokowana dynamicznie
- ilosc (liczba całkowita)
- cena (liczba zmiennoprzecinkowa)

Oraz zestaw metod: metody zwracające i ustawiające składniki klasy (metody typu “get” i “set”), konstruktor domyślny, konstruktor z 3 parametrami oraz konstruktor kopiujący i destruktor. Należy zewnętrzną funkcję Wypisz() która przyjmuje obiekt Owoc i wypisuje go na ekran . Ponadto należy stworzyć metodę SetAll ustawiającą wszystkie trzy parametry jednocześnie (jeśli podamy dwa parametry, wtedy cena=1.0, a jeśli tylko jeden parametr, wtedy cena=1.0, a ilosc = 0), oraz przeciążyć operator przekierowania: (“<<”).

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/2013/kolokwium/mainowoc.cpp> )

```
#include <iostream>
#include <cstring>
#include "owoc.h"

using namespace std;

int main()
{
    char* gruszka = new char[8];    strcpy(gruszka,"gruszka");
    char* jablko= new char[7];     strcpy(jablko,"jablko");
    char* banan= new char[6];      strcpy(banan,"banan");

    Owoc o1;
    o1.SetNazwa(gruszka);
    cout<<"Nazwa: "<<o1.GetNazwa()<<" Ilosc: "<<o1.GetIlosc()<<" Cena: "<<o1.GetCena()<<endl;
    delete gruszka;
    Wypisz(o1);

    Owoc *o2 = new Owoc(jablko,10,1.1);
    cout<<"Nazwa: "<<o2->GetNazwa()<<" Ilosc: "<<o2->GetIlosc()<<" Cena: "<<o2->GetCena()<<endl;
    cout<<o1<<" , "<<*o2<<endl;
    delete o2;

    Owoc o3(o1); Wypisz(o3);
    o3.SetAll(banan);
    cout<<o3<<endl;
    o3.SetAll(banan,2);
    cout<<o3<<endl;
    o3.SetAll(banan,2,4.5);
    return 0;
}
```

Co po uruchomieniu skutkowało by pojawieniem się na ekranie:

```
$ ./program
Nazwa: gruszka Ilosc: 0 Cena: 0
Owoc: (gruszka, 0, 0)
Nazwa: jablko Ilosc: 10 Cena: 1.1
Owoc: (gruszka, 0, 0), Owoc: (jablko, 10, 1.1)
Jestem w destruktorze...
Owoc: (gruszka, 0, 0)
Owoc: (banan, 0, 1)
Owoc: (banan, 2, 1)
Jestem w destruktorze...
Jestem w destruktorze...
```

Plik nagłówkowy powinien zostać otoczony strukturą „ifndef”. Program powinien być napisany w 3 plikach (**owoc.cpp**, **owoc.h** 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](mailto:majanik@if.pw.edu.pl).

## Kolokwium Testowe 1, gr. B, Języki Programowania, 19.11.2014

### Treść zadania

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

- x, y, z (double)
- notatka (std::string)
- 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 punkty 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 między 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 między 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 bez żadnych ostrzeżeń (flaga -Wall).

Program należy wysłać przed końcem zajęć na adres [majanik@if.pw.edu.pl](mailto:majanik@if.pw.edu.pl)