JP, Wtorek, 15.10.2012, 16:15-17:45
Zadanie 2

Zadanie pokazuje użycie struktur i klas w języku C++ - podstawowych elementów programowania obiektowego.

1. Należy stworzyć plik main.cpp zawierający strukturę `Punkt` oraz funkcję `main`. Struktura `Punkt` ma zawierać cztery pola – `string fNazwa`, `double fX`, `double fY` (współrzędne), oraz dwie funkcje (które będziemy nazywać od dzisiaj metodami):
   `void UstawPunkt(string Nazwa, int x, int y)` - ustawia elementy `fNazwa` oraz współrzędne `fX` i `fY`, oraz oblicza i ustawia odległość
   `void WypiszPunkt()` - wypisuje na ekran elementy obiektu `Punkt` (czyli `fNazwa` oraz `fX`, `fY`).
   W funkcji `main` należy stworzyć obiekt `Punkt`, ustawić podaną metodą jego nazwę oraz współrzędne i wypisać go. **1.5 p.**
   Uwaga!
   String (działa podobnie jak tablica `char*`) wchodzi w skład standardowej biblioteki języka C++. Należy go użyć przez załadowanie odpowiedniego nagłówka:
   `#include <string>`
2. Należy zmienić strukturę na klasę (użyć słowa kluczowego `class` zamiast `struct`, patrz uwaga niżej). **0.5 p.**
   Uwaga! W "pierwszym przybliżeniu" klasa jest tym samym co struktura – z jednym wyjątkiem (oprócz słowa kluczowego `class` zamiast `struct`) - elementy struktury są domyślnie publiczne, tzn. są widoczne na zewnątrz struktury. Innymi słowy, konstrukcja typu:
   `cout<<punkt.fNazwa<<endl;` - gdzie `punkt` jest obiektem typu struktury `Punkt`, jest poprawna i program się skompiluje. Elementy klasy są z kolei niewidoczne na zewnątrz, czyli program z tego typu wywołaniem nie skompiluje się. W tym celu, te elementy klasy, które chcemy uczynić widocznymi na zewnątrz, należy opatrzyć tzw. specyfikatorem dostępności `public`, np.:
   ```
   class Punkt
   {
    public:
       string fNazwa;
   };
   ```
   Taka definicja klasy powoduje, że dostęp do elementu `fNazwa` jest możliwy w dalszej częsci programu, czyli jego wypisanie np. w funkcji `main` powyższą konstrukcją byłoby możliwe. Więcej o specyfikatorach dostępu będzie na następnym wykładzie.
3. Klasę `Punkt` należy podzielić na pliki Punkt.h z deklaracją klasy `Punkt` i Punkt.cpp z definicjami metod tejże klasy. Należy stworzyć plik main.cpp, który będzie wykorzystywał klasę `Punkt` (ustawi obiekt typu `Punkt` oraz go wypisze). **1 p.**
4. Należy stworzyć w pliku main.cpp funkcję:
   `void WypiszPunkt(Punkt *pkt)` – która robi dokładnie to samo co metoda klasy `Punkt` o tej samej nazwie, jednak nie jest jej elementem składowym i przyjmuje wskaźnik do obiektu typu `Punkt`. **1 p.**
5. Ustawić przeciążenie operatora `+` - użycie operatora dodawania ma dodać do siebie współrzędne punktów, sprawdzając, czy dodajemy do siebie punkty o tej samej nazwie. **1 p.**

Zadanie dodatkowe:

1. Stworzyć jeszcze jedną funkcję (w pliku main.cpp):
   `void WypiszPunkt2(Punkt &pkt)` – przyjmującą referencję do obiektu typu

`Punkt`.

Stworzyć tablicę elementów typu `Punkt`, ustawić ich elementy oraz wypisać wszystkimi trzema metodami. **0.5 p.**

Uwagi:

1. Do wypisywania elementów na ekran używamy metod języka C++ (**nie używamy metod języka C**), tzn. strumieni wyjścia/wejścia `cout` i `cin`. Należy załączyć nagłówek:
   ```
   #include <iostream>
   ```
2. Do kompilacji używamy kompilatora g++:
   ```
   g++ -Wall main.cpp klasa1.cpp klasa2.cpp -o main
   ```