

Języki Programowania+, Wtorek 15.10.2013, 8:15-9:45

Zadanie 1

Zadanie polega na zaimplementowaniu w języku C++ przykładowej klasy `Punkt` zawierającej dwa elementy składowe – współrzędne kartezjańskie x oraz y położenia punktu, oraz stworzeniu dwóch najprostszych metod służących do operowania polami składowymi klasy.

1. Należy stworzyć strukturę `Punkt` reprezentującą punkt w dwuwymiarowej przestrzeni kartezjańskiej. Struktura powinna zawierać dwa pola składowe typu `double`:

- `double fx;`
- `double fy;`

2. Napisać zewnętrzną funkcję `Zeruj`, przyjmującą referencję do obiektu klasy `Punkt`, która będzie przypisywała wartość 0 polom fx oraz fy dla danego punktu. Należy stworzyć instancję obiektu `Punkt` w funkcji głównej programu (`main`), wyzerować jego pola używając napisanej funkcji oraz następnie wypisać te wartości na ekran. **(0.5 pkt)**

3. Dopisać do struktury `Punkt` następujące metody (funkcje składowe struktury):

- `void UstawPunkt(double x, double y);` - ustawia pola fx oraz fy na zadane wartości x oraz y
- `void WypiszPunkt();` - wypisuje na ekran elementy składowe obiektu `Punkt`

W funkcji głównej programu należy stworzyć obiekt `Punkt` o wartościach $x=10$, $y=90$ i wypisać go na ekran. Następnie należy stworzyć drugi obiekt `Punkt` o współrzędnych zadanych z klawiatury i również wypisać go na ekran. **(1.0 pkt.)**

4. Należy zamienić strukturę na klasę. Wprowadzamy odpowiednie poprawki. Należy umieścić słowo kluczowe `public` w odpowiednim miejscu w kodzie. **(1 pkt)**

5. Dopisać metodę `double Odleglosc(double x, double y)`, która policzy odległość punktu od danego punktu (x,y) (należy użyć znanej z C biblioteki `math.C` poprzez `#include <math.h>`) **(0.5 pkt)**

6. Stworzyć metodę `bool Wewnatrz(double r, double x, double y)` – sprawdzającą czy punkt mieści się w kole o danym promieniu r oraz środku w punkcie (x,y) – metoda zwraca `true`, jeśli punkt znajduje się oraz `false` jeśli nie. Sprawdzić działanie metody dla wybranego obiektu typu `Punkt`. **(1 pkt.)**

5. Podzielić klasę `Punkt` na dwa pliki – `Punkt.h` z definicją klasy (słowo kluczowe `class` oraz jej składniki) oraz `Punkt.cpp` z metodami klasy (ciała metod klasy). Funkcja `main` zostaje w osobnym trzecim pliku. **(1 pkt.)**

Wskazówki:

- By dostać się do składników struktury „na zewnątrz” używamy `."`, np. `cout<<A.x<<endl;`
- Należy pamiętać o załączeniu biblioteki `Punkt.h` w plikach `.cpp` (`#include "Punkt.h"`).
- W pliku `Punkt.cpp` przy definicjach funkcji należy pamiętać o dodaniu operatora zakresu `"Punkt : :"`