Języki Programowania+, wtorek 11.10.2016, 12:15-13:45

Zadanie 1

Zadanie polega na zaimplementowaniu w języku C++ przykładowej struktury `Punkt` zawierającej dwa elementy składowe – współrzędne `wiersz` oraz `kolumna` położenia punktu, oraz stworzeniu metod (funkcji) służących do operowania polami składowymi struktury. Na samym końcu zmienimy strukturę na klasę (podstawy koncept programistyczny w języku C++), która jest rozwinięciem idei struktur znanych z języka C.

1. Należy stworzyć **strukturę** `Punkt` reprezentującą punkt. Struktura powinna zawierać dwa pola składowe typu `int` (całkowite): `wier` oraz `kol`.

2. Napisać **zewnętrzną funkcję** `Losuj`, przyjmującą referencję do obiektu klasy `Punkt`, która będzie przypisywała losową wartość z zakresu `0-9` polom `wier` oraz `kol` dla zadanego punktu. Należy stworzyć instancję obiektu `Punkt` w funkcji głównej programu (`main`), wylosować pola używając napisanej funkcji oraz następnie wypisać te wartości na ekran. Wypisując wartości należy odwołać się bezpośrednio do składników klasy. **(0.5 pkt)**

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

- `void UstawPunkt(double x, double y);` - ustawia pola `row` oraz `col` na zadane wartości `x` oraz `y`
- `void WypiszPunkt();` - wypisuje na ekran elementy składowe obiektu `Punkt` oddzielone spacją.

W funkcji głównej programu należy stworzyć obiekt `Punkt` i kolejno nadać mu wartości `wier=3`, `kol=4` i narysować na ekranie w wierszu i kolumnie odpowiadajacym składowym klasy. Następnie należy stworzyć drugi obiekt `Punkt` o współrzędnych zadanych z klawiatury i wypisać go na ekran przy użyciu funkcji `WypiszPunkt`. **(1.0 pkt.)**

4. Należy zamienić strukturę na klasę. Wprowadzamy odpowiednie poprawki, aby program się nadal kompilował. **Podpowiedź:** Należy umieścić słowo kluczowe `public` w odpowiednim miejscu w kodzie. **(1 pkt)**

5. Dopisać metodę `double Odleglosc(int r, int c)`, która (zakładając, że pola `wier` i `kol` odpowiadają wartościom w przestrzeni kartezjańskiej) policzy odległość od punktu, na którym wywoływana jest metoda, do punktu zadanego jako argumenty metody (`r,c`) (by uzyskać pierwiastek należy użyć znanej z C biblioteki `math.C` poprzez `#include <math.h>`). **(0.5 pkt)**

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

7. W funkcji głównej main tworzymy tablicę kilku punktów. Następnie piszemy funkcję (nie będącą metodą składową klasy) `NarysujPunkty(Punkt* tab, int iloscPunktow, int n, int m)` – która w prostokącie (należy wypisać ramkę ze znaków '=') o bokach `n x m` wypisze punkty z tablicy `tab` w zależności od ich położnia (`row`, `col`). **(1 pkt.)**

Dodatkowe:

Stworzyć metodę `bool Wewnatrz(double r, double x, double y)` – sprawdzającą czy punkt mieści się w kole o zadanym 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`. **(0.5 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::”`
- W C++ programy kompilujemy za pomocą komendy g++:

g++ -Wall plik1.cpp plik2.cpp main.cpp -o NazwaProgramu

Za program, który się nie kompiluje można otrzymać maksymalnie 2 pkt!