

Języki Programowania, Wtorek 13.10.2015, 8:15-9:45

Zadanie 1

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

1. Należy stworzyć pusty plik tekstowy – powinien mieć rozszerzenie `.cpp` (np. `main.cpp`). W pliku należy dołączyć biblioteki wejścia/wyjścia:

```
#include <iostream>
```

oraz zdefiniować używaną przestrzeń nazw:

```
using namespace std;
```

oraz napisać funkcję `int main()`. Funkcja powinna wypisać na ekran sekwencję „(0,1)”. Program należy skompilować wpisując w linii komend polecenie:

g++ -Wall UtworzonyPlik.cpp -o NazwaProgramu

Jeśli program działa, należy zmodyfikować go w taki sposób, aby wypisał wartości podawane z klawiatury oraz je wypisywał w takiej samej formie jak wyżej. Po skończeniu tej części należy pokazać prowadzącemu działający program. **(0.5 pkt)**

2. Należy stworzyć strukturę `Wektor` reprezentującą wektor w dwuwymiarowej przestrzeni kartezjańskiej. Struktura powinna zawierać dwa pola składowe typu `double`:

- `double fX;`

- `double fY;` **(0.5 pkt)**

oraz następujące metody (funkcje składowe struktury):

- `void UstawWektor(double x, double y);` - ustawia pola `fX` oraz `fY` na zadane wartości x oraz y

- `void WypiszWektor();` - wypisuje na ekran elementy składowe obiektu typu `Wektor` **(1 pkt)**

W funkcji głównej programu (`main`) należy stworzyć obiekt typu `Wektor` o wartościach $x=10$, $y=90$ i wypisać go na ekran. Następnie należy stworzyć drugi obiekt typu `Wektor` o współrzędnych zadanych z klawiatury i również wypisać go na ekran. **(1 pkt.)**

Łącznie 3 pkt.

3. Należy zamienić strukturę na klasę – w tym celu słowo kluczowe `struct` zmieniamy na słowo kluczowe `class`. Sprawdzamy, czy program dalej działa (dlaczego nie działa?). **(0.5 pkt)**

Umieścić słowo kluczowe `public` w odpowiednim miejscu w kodzie.

4. Stworzyć metodę `Dlugosc()`, która policzy długość wektora (należy użyć znanej z C biblioteki `math.C` poprzez `#include <math.h>`)

(0.5 pkt)

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

Wskazówki:

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

Za niekompilujący się program można otrzymać maksymalnie 2 pkt!