

Języki Programowania, wtorek 11.10.2016, 8:15-9:45

Zadanie 1

Zadanie polega na zaimplementowaniu w języku C++ przykładowej struktury `Vector` zawierającej dwa elementy składowe – współrzędne kartezjańskie x oraz y wektora dwuwymiarowego, oraz stworzeniu dwóch najprostszych metod (funkcji) służących do operowania polami składowymi struktury. Na samym końcu zmienimy strukturę na klasę (podstawy konceptu programistycznego w języku C++), która jest rozwinięciem idei struktur znanych z języka C.

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 wypisywał dwie wartości podawane z klawiatury, w takiej samej formie jak wyżej. Po skończeniu tej części należy pokazać prowadzącemu działający program. **(0.5 pkt)**

2. Należy stworzyć strukturę `Vector` 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 SetVector(double x, double y);` - ustawia pola `fX` oraz `fY` na zadane wartości `x` oraz `y`
- `void PrintVector();` - wypisuje na ekran elementy składowe obiektu typu `Vector` **(1 pkt)**

W funkcji głównej programu (`main`) należy stworzyć obiekt typu `Vector` o wartościach `x=10`, `y=90` i wypisać go na ekran. Następnie należy stworzyć drugi obiekt typu `Vector` 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)**

Podpowiedź: należy umieścić słowo kluczowe `public` w odpowiednim miejscu w kodzie.

4. Stworzyć metodę `Length()`, która policzy długość wektora (należy użyć znanej z C biblioteki `math.h` poprzez `#include <math.h>`) **(0.5 pkt)**

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

Wskazówki:

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