

Języki Programowania, Zadanie 2, 16.10.2011

Należy stworzyć strukturę wektor reprezentującą wektor. Struktura ta powinna zawierać dwa pola typu double (składowe x i y) oraz zawierać następujące metody (funkcje składowe):

- umożliwiającą zapisanie danego wektora `zapisz(double xx, double yy)` (1 p)
- wypisującą dany wektor na ekran w formacie `[x,y]` `wypisz()` (0.5 p)

Struktura powinna być podzielona na dwa oddzielne pliki: `wektor.h` i `wektor.cpp`.

W pliku `*.h` powinna się znaleźć definicja struktury (słowo kluczowe `struct`, składniki struktury, oraz deklaracje funkcji), natomiast w pliku `*.cpp` – ciało funkcji składowych (metod). Dodatkowo główny plik programu zawierający funkcję `main()` - razem 3 pliki. **(Patrz: wykład!)**

Wskazówki:

- 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::"`
- By dostać się do składników struktury „na zewnątrz” używamy `"."`, np. `cout<<A.x<<endl;`

W programie głównym należy stworzyć pojedynczy obiekt typu Wektor ($x = 2, y = 3$) i wypisać go na ekran. (1 p)
Następnie zgłosić ukończenie pierwszej części zadania prowadzącemu.

Należy dopisać następującą funkcję składową:

- zwracającą długość wektora `dlugosc()` (0.5 p)

Oraz funkcje zewnętrzne:

- zrealizować operator mnożenia (*) - iloczyn skalarny wektorów (1 p)
- `wektor dodaj(wektor w1, wektor w2)` (0.5 p)
- `wektor odejmij(wektor w1, wektor w2)` (0.5 p)

Wszystkie napisane funkcje muszą zostać przetestowane w programie głównym.

```
#include <iostream>
```

```
#include <math.h>
```

```
g++ -Wall Wektor.cpp program.cpp -o program
```

Zadanie dodatkowe

Należy przygotować „bazę danych” z możliwością dodawania, usuwania oraz wypisywania wektorów używając w tym celu tablicy statycznej na 30 wektorów.

W głównym programie (funkcji `main`) wymagana jest instrukcja typu switch - case, pytająca w pętli o podanie liczby od 1-4.

- Wpisanie 1 - wypisanie wszystkich zapisanych wektorów na ekran. (`cout<<zmienna`)
- Wpisanie 2 - dodanie nowego wektora do „bazy danych”. (`cin>>zmienna`)
- Wpisanie 3 - usuwanie wektora o konkretnym indeksie z „bazy danych”
- Wpisanie 4 - powinno przerwać pętlę oraz spowodować opuszczenie programu.
- Wpisanie liczby innej niż dozwolone - program zakomunikuje, iż została wprowadzona błędna wartość.