

Zadanie 2, Języki Programowania,

Proszę stworzyć program umożliwiający zapamiętywanie zbioru prostokątów (ich położenia na płaszczyźnie oraz długości boków). Następnie możliwe powinno być sprawdzenie, czy podany punkt o współrzędnych (x,y) znajduje się wewnątrz którejkolwiek z figur. Wszystkie podawane współrzędne mają być dodatnie.

Cały program należy napisać w trzech plikach: nagłówkowym (*.h), implementacyjnym (*.cpp), oraz zawierającym funkcję główną (*.cpp).

Należy stworzyć klasy:

Punkt

- Pola publiczne: • `double x,y; // (x,y) – współrzędne punktu`
Metody publiczne: • `void Zapisz(double X, double Y),`
• `void Wypisz()`
Domyślnymi wartościami powinny być `x = y = 30`.

Prostokat

- Pola prywatne: • `Punkt p, double a,b; // p – lewy dolny róg prostokąta, a,b – długości boków`
Metody publiczne: • `void Zapisz(double X, double Y, double A, double B),`
• `void Wypisz(),`
• `double Pole() //zwraca pole powierzchni`
• metody `set` i `get` do ustawiania i pobierania pola `x` oraz `a`
np. `void SetA(double A), double GetX()`
Domyślnymi wartościami dla wymiarów prostokąta powinny być `A = B = 30`.
• `void Wylicz(Punkt p1, Punkt p2) //ustawia p, a i b, jeśli p1 i p2 to kolejno lewy dolny i prawy górny róg prostokąta)`

Szablon funkcji głównej znajduje się z drugiej strony kartki.

Funkcje **zaprzyjaźnione** – nie funkcje składowe (z tymi samymi nazwami – **przeciążanie nazw**):

bool Wewnatrz (Prostokat P, double x, double y) – zwraca „true” jeśli punkt (x,y) znajduje się wewnątrz prostokąta P.

bool Wewnatrz (Prostokat P, Punkt pkt)– zwraca „true” jeśli punkt (pkt) znajduje się wewnątrz prostokąta P.

bool Wewnatrz(Prostokat *P, int n, double x, double y) – zwraca „true” jeśli punkt (x,y) znajduje się wewnątrz **któregokolwiek** z n prostokątów zapisanych w podanej tablicy P.

Działanie wszystkich funkcji powinno zostać przetestowane dla przykładowych obiektów w funkcji main.

Szablon dla zawartości funkcji głównej (main):

```
//utworzenie pojedynczego punktu pkt1 (x=3,y=3)
//utworzenie pojedynczego prostokąta p1 (x=4,y=10) – a, b: domyślne

//wypisanie punktu pkt1 na ekran
//wypisanie prostokąta na ekran
//wypisanie pola prostokąta na ekran
    ---2 pkt---
//stworzenie drugiego prostokąta p2 (x=1, y =10, a = 10, b = 5)
//utworzenie drugiego punktu (x=6,y=6)
//stworzenie trzeciego prostokąta na podstawie punktów pkt1 i pkt2 (fun. Wylicz)
//wypisanie trzeciego prostokąta na ekran
    ---3 pkt---
//sprawdzenie funkcji Wewnatrz (wypisanie na ekranie "0" lub "1") dla prostokąta
p1 i punktów: (1,1) oraz (5,11) – zapisanych jako dwie wielkości typu double
//sprawdzenie funkcji Wewnatrz (wypisanie na ekranie "0" lub "1") dla prostokąta
p1 i punktów: (2,2) oraz (6,12) – zapisanych jako obiekty typu Punkt
    ---4 pkt---
//utworzenie 3-elementowej tablicy prostokątów, wpisanie w kolejne komórki
prostokątów p1, p2, p3.
// sprawdzenie funkcji Wewnatrz (wypisanie na ekranie "0" lub "1") dla tablicy
prostokątów
    ---5 pkt----
```

Zadania dodatkowe:

1. `double* Boki()` zwracającą tablicę dwóch liczb: na pierwszym miejscu długość boku a, na drugim miejscu tablicy: długość boku b.

2. Należy stworzyć funkcje (nie funkcje składowe!):

Prostokat SkopiujProstokat(Prostokat &p) – zwracającej nowy prostokąt poprzez kopiowanie długości boków prostokąta p, ale zerując współrzędne x oraz y. **Do czego tutaj służy „&”?**

Prostokat* SkopiujProstokatWsk(Prostokat *p) – zwracającej nowy prostokąt poprzez kopiowanie długości boków prostokąta p, zerując współrzędne x oraz y, ale działającą na wskaźnikach. **void ZmienProstokat(Prostokat &p, double x, double y, double a, double b)** – zmieniająca wartości (a,b,x,y) istniejącego obiektu Prostokat p.

```
//utworzyć prostokat p3 kopiując prostokat p2 przy użyciu funkcji SkopiujProstokat
//utworzyć wkaźnik na prostokat p4 kopiując prostokat p3 przy użyciu funkcji
SkopiujProstokatWsk. Wypisać prostokat p4 na ekran.
//zmienić prostokat p3 tak, by jego składowe wynosiły (1,1,1,1), wypisać na ekran prostokat
p3
//wypisać na ekran boki prostokata p4 uzyskane przy użyciu funkcji Boki()
```

3. W przyszłości chcielibyśmy nasze figury wyświetlać na ekranie. By prezentowały się odpowiednio będziemy, prócz współrzędnych i wymiarów, przyporządkowywać im kolor. Lista dostępnych kolorów to: czerwony, zielony, niebieski, pomarańczowy.

Proszę stworzyć typ wyliczeniowy przechowujący informację o kolorze dla klasy prostokąt, oraz dodać dodatkowy składnik: kolor do tej klasy.