

Języki Programowania+, Wtorek 18.10.2018, 8-10

Zadanie 2

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

Cały program należy napisać w pięciu plikach.

Należy stworzyć klasę:

Rectangle (1.0 p)

Pola prywatne: double x,y,a,b; //(x,y) – wybrany róg prostokąta, a,b – długości boków

Metody publiczne:

- konstruktor domyślny

- konstruktor główny

Print(), double Area() //zwraca pole powierzchni

oraz metody set i get do ustawiania i pobierania pól prywatnych, jeśli potrzebne

Domyślnymi wartościami dla wymiarów prostokąta powinny być A = B = 30

W funkcji głównej należy stworzyć obiekt typu Rectangle(x=4, y =10) , wypisać go na ekran oraz wypisać na ekranie jego pole, następnie zgłosić ukończenie części zadania prowadzącemu.

Należy przeciążyć operator dodawania (operator+) (0.5 p)

Dla klasy prostokąt (zwracający prostokąt o współrzędnych pierwszego z podanych prostokątów i długości boków będących sumą długości boków prostokątów dodawanych). **Operator należy zaprzyjaźnić z klasą Rectangle, oraz odwoływać się bezpośrednio do prywatnych pól składowych klasy.**

Ponadto należy stworzyć funkcje (nie metody składowe!): (1.0 p)

Rectangle CopyRectangle(Rectangle &p) – zwracającą nowy prostokąt poprzez skopiowanie długości boków prostokąta p, ale ustawiając współrzędne x oraz y na 0.

void ChangeRectangle(Rectangle *p, double X, double Y, double A, double B)

– zmieniającą wartości (a,b,x,y) istniejącego obiektu Rectangle.

Oraz napisać klasę okrąg:

Circle (1.0 p)

Pola prywatne: double x,y,R; //(x,y) – środek okręgu, R- promień

Metody publiczne: konstruktory

Print()

double Area()

metody set i get do ustawiania i pobierania pól prywatnych, jeśli potrzebne

oraz metodę double* Center() zwracającą tablicę dwóch liczb: na pierwszym miejscu składową x klasy okrąg, na drugim miejscu tablicy: składową y.

Domyślną wartością dla promienia okręgu powinno być 10.

Funkcje zewnętrzne (z tymi samymi nazwami – przeciążanie nazw): (1.0 p)

`bool IsInside` – argumentami funkcji powinien być obiekt klasy `Rectangle` oraz współrzędne `x` oraz `y`, zwraca „true” jeśli punkt `(x,y)` znajduje się wewnątrz prostokąta.

`bool IsInside` – argumentami funkcji powinien być obiekt klasy `Circle`, oraz współrzędne `x` oraz `y`, zwraca „true” jeśli punkt `(x,y)` znajduje się wewnątrz okręgu.

`bool IsInside` – argumentami funkcji powinny być dwie tablice: prostokątów i okręgów, ilość prostokątów w tablicy, ilość okręgów w talibcy, oraz współrzędne punktu `x`, `y`, zwraca „true” jeśli punkt `(x,y)` znajduje się wewnątrz **któregokolwiek** z prostokątów bądź okręgów zapisanych w podanych tablicach.

Jeśli starczy czasu, stworzyć nową strukturę `Point` (opisującą punkt w przestrzeni 2D) i przerobić klasy `Rectangle` i `Circle` w ten sposób, by zamiast wartości `double x,y` przyjmowały obiekt klasy `Point`.

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

Szablon dla zawartości funkcji głównej (`main`):

```
//utworzenie pojedynczego prostokąta
//wypisanie prostokąta oraz jego pola
--- 1 pkt ---
//utworzyć prostokąt p2 (9,9,5,7)
//utworzyć prostokąt p3 dodając prostokaty p1 oraz p2
//wypisać p3
--- 1.5 pkt ---
//utworzyć prostokąt p4 kopiując prostokąt p2 przy użyciu funkcji CopyRectangle
//zmienić prostokąt p4 tak, by jego składowe wynosiły (1,1,1,1)
//wypisać p4
--- 2.5 pkt ---
//utworzenie i wypisanie pojedynczego okręgu
--- 3 pkt ---
//utworzyć tablice (a) dwóch okręgów i (b) trzech prostokątów
//jako pierwszy prostokąt wpisać obiekt p1, jako drugi: p2 jako trzeci (2,3)
//jako pierwszy okrąg wpisać obiekt o1, jako drugi okrąg: (13,14,5)
--- 4 pkt ---
//wypróbować funkcje IsInside na utworzonych tablicach figur
//użyć strukturę Punkt zamiast pól double x oraz double y w klasach Circle oraz Rectangle
--- 5 pkt ---
```

Zadanie dodatkowe:

1. Stworzyć funkcje zewnętrzne:

`Circle* CopyCircle(Circle *o)` – zwracającej nowy okrąg poprzez kopiowanie długości boków okręgu `o`, zerując współrzędne `x` oraz `y`, ale działającą na wskaźnikach.

2. 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.

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