

# JĘZYK PROGRAMOWANIA C++

## FIGURY NA PŁASZCZYŹNIE

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Geometria euklidesowa to klasyczna odmiana geometrii opisana po raz pierwszy przez Euklidesa w dziele *Elementy* z IV wieku p.n.e. (zebrał on w tym dziele całą ówczesną wiedzę matematyczną znaną Grekom). Pierwotnie uprawiano ją jedynie na płaszczyźnie i w przestrzeni trójwymiarowej, wiążąc ją jednocześnie ze światem fizycznym, który miała opisywać.

W geometrii Euklidesowej występują pojęcia pierwotne, przyjmowane bez definicji, do których należą punkt, prosta, płaszczyzna oraz należenie punktu do prostej i należenie prostej do płaszczyzny. Euklides przyjął, że punkt, prosta, płaszczyzna i przestrzeń mają wymiary kolejno: zero, jeden, dwa i trzy.

W geometrii Euklidesa istnieją trzy zasadnicze przekształcenia płaszczyzny:

- (i) przesunięcie (translacja), polegające na przemieszczeniu wszystkich punktów płaszczyzny o tę samą odległość w ustalonym kierunku;
- (ii) obrót wokół ustalonego punktu wszystkich punktów płaszczyzny;
- (iii) symetria względem osi.

Dwie figury definiuje się jako równoważne (przystające), jeżeli jedna z nich może być przekształcona w drugą za pomocą przesunięć, obrotów i odbić. Obroty, przesunięcia i translacje tworzą grupę przekształceń.

### Zadanie.

Zdefiniuj klasy **punkt**, **odcinek** i **trojkat**, które będą reprezentowały odpowiednio punkt, odcinek i trójkąt na płaszczyźnie euklidesowej z kartezjańskim układem współrzędnych. Klasa **punkt** powinna zawierać dwa pola `x` i `y` typu `double` do pamiętania współrzędnych. Klasa **odcinek** ma reprezentować odcinek na płaszczyźnie ograniczony dwoma różnymi punktami. Klasa **trojkat** ma reprezentować trójkąt na płaszczyźnie wyznaczony przez trzy niewspółliniowe punkty.

W klasach tych zdefiniuj konstruktory (w tym konstruktor kopiujący), przypisania kopiujące oraz metody przesuwające obiekty geometryczne o zadany wektor i dokonujące ich obrotu dookołaadanego punktu o zadany kąt.

Ponadto zdefiniuj funkcję globalną, która będzie liczyć dystans pomiędzy dwoma punktami.

W klasie **odcinek** zdefiniuj metodę obliczającą długość odcinka, metodę badającą czy zadany punkt leży na odcinku i metodę zwracającą środek odcinka. Dodatkowo zdefiniuj funkcje globalne, które będą sprawdzać czy dwa odcinki są równoległe, czy są prostopadłe i metodę wyznaczającą punkt przecięcia odcinków (jeśli odcinki się nie przecinają to należy zgłosić wyjątek).

W klasie **trojkat** zdefiniuj metodę obliczającą obwód trójkąta, pole trójkąta, metodę badającą czy zadany punkt leży wewnątrz trójkąta i metodę zwracającą środek trójkąta. Dodatkowo zdefiniuj funkcje globalne, które będą sprawdzać czy dwa trójkąty są rozłączne, czy jeden zawiera drugi i metodę sprawdzającą czy punkt leży wewnątrz trójkąta.

Na koniec napisz program rzetelnie testujący działanie obiektów tych klas. Wszystkie obiekty w tym programie powinny być utworzone na stosie. Każda zdefiniowana metoda w klasach **punkt**, **odcinek** i **trojkat** powinna być w programie wywołana a jej wyniki wypisane na standardowe wyjście.

#### **Uwaga.**

Podziel program na pliki nagłówkowe i źródłowe.

#### **Podpowiedź.**

Sytuację wyjątkową zgłaszamy instrukcją **throw**. Na przykład w konstruktorze klasy **odcinek** należy zasygnalizować błąd za pomocą wyjątku, gdy oba końce odcinka będą miały takie same współrzędne. Niech wyjątkami będą obiekty typu **string**:

```
if (p.wspix()==q.wspix() or p.wspiy()==q.wspiy())  
    throw string("nie można utworzyć odcinka o zerowej długości");
```