

KURS JĘZYKA C++

FIGURY NA PŁASZCZYŹNIE

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Prolog.

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.

Przekształcenie płaszczyzny euklidesowej jest nazywane *przekształceniem izometrycznym*, gdy zachowuje odległości dowolnej pary punktów — innymi słowy, jeśli jakąś figurę przekształcimy za pomocą izometrii, to nie zmieni ona ani kształtu ani rozmiaru. W geometrii euklidesowej istnieją cztery zasadnicze przekształcenia płaszczyzny:

- (i) przesunięcie (translacja) wszystkich punktów płaszczyzny na tą samą odległość w ustalonym kierunku (o zadany wektor);
- (ii) obrót wokół ustalonego punktu wszystkich punktów płaszczyzny (o zadany kąt);
- (iii) symetria (odbicie) względem punktu;
- (iv) symetria (odbicie) względem osi;

Dwie figury definiuje się jako *przystające*, jeżeli jedna z nich może być przekształcona w drugą za pomocą przesunięć, obrotów i symetrii. Przesunięcia, obroty i symetrie 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. Pamiętaj o hermetyzacji, aby ukryć stan każdego obiektu.

W wymienionych klasach zdefiniuj konstruktory (w tym konstruktor kopiujący), przypisania kopiujące oraz funkcje składowe do wykonywania przekształceń izometrycznych na tych obiektach geometrycznych (przesunięcia, obroty oraz symetrie środkowe i osiowe).

Ponadto zdefiniuj funkcję globalną, która będzie wyznaczać odległość pomiędzy parą punktów.

W klasie `odcinek` zdefiniuj funkcję składową obliczającą długość odcinka oraz funkcję składową sprawdzającą, czy zadany punkt należy do odcinka. Dodatkowo zdefiniuj dwie funkcje globalne, które będą sprawdzać czy dwa odcinki są równoległe i czy są prostopadłe.

W klasie `trojkat` zdefiniuj funkcje składowe obliczające obwód trójkąta i pole trójkąta oraz funkcję składową sprawdzającą, czy zadany punkt leży wewnątrz trójkąta. Dodatkowo zdefiniuj dwie funkcje globalne, które będą sprawdzać czy dwa trójkąty są rozłączne i czy jeden zawiera się w drugim.

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.

Wskazówka.

Do wykonania translacji będziesz potrzebować wektora — zdefiniuj go w postaci odrębnej klasy. Do wykonania symetrii osiowej będziesz potrzebować prostej — zdefiniuj ją w postaci odrębnej klasy.

Podpowieź.

Zawsze, gdy napotykamy w programie jakieś błędy, niejednoznaczności czy sprzeczności należy to sygnalizować na pomocą wyjątków. Sytuacje wyjątkowe zgłaszamy instrukcją `throw`. Na przykład w konstruktorze klasy `odcinek` należy zasygnalizować wyjątek, gdy oba końce odcinka będą miały takie same współrzędne. Niech wyjątkami będą obiekty typu `invalid_argument` zadeklarowane w pliku nagłówkowym `<stdexcept>`:

```
if (p.wspix() == q.wspix() and p.wspiy() == q.wspiy())
    throw invalid_argument("nie można utworzyć odcinka o długości 0");
```

Uwaga.

Podziel program na pliki nagłówkowe i źródłowe. Funkcję `main()` umieść w osobnym pliku źródłowym.

Istotne elementy programu.

- Definiowanie klas w pliku nagłówkowym i funkcji składowych w pliku źródłowym.
- Definicja konstruktorów kopiujących i operatorów przypisania.
- Ukrywanie stanu obiektów (hermetyzacja).
- Wykorzystanie klasy `punkt` przy budowie odcinka i trójkąta (kompozycja).
- Implementacja algorytmów geometrii analitycznej.
- Zgłaszanie i wyłapywanie wyjątków.