Zadanie 3 18-21 marca 2024 r.

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). 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:

1. przesunięcie/translacja – przesunięcie wszystkich punktów płaszczyzny/figury o zadany wektor;
2. obrót – obrót wszystkich punktów płaszczyzny/figury wokół ustalonego punktu o zadany kąt;
3. symetria środkowa – odbicie wszystkich punktów płaszczyzny/figury względem ustalonego punktu;
4. symetria osiowa – odbicie wszystkich punktów płaszczyzny/figury względem ustalonej 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 nie współliniowe 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 – jedna ma sprawdzać czy dwa odcinki są równoległe a druga 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. W programie testującym uruchom wszystkie funkcje globalne oraz wszystkie publiczne funkcje składowe. Rezultaty wywołanych funkcji zaprezentuj na standardowym wyjściu. Wszystkie obiekty w tym programie powinny być utworzone na stosie.

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

Uwaga

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 (w postaci równania ogólnego Ax + By + C = 0, przy czym A i B nie mogą być jednocześnie równe 0).

Wskazówka

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 zgłosić wyjątek, gdy oba końce odcinka będą miały takie same współrzędne. Niech wyjątkami będą obiekty typu `invalid_argument` (deklaracje tej klasy wyjątku znajduje się w pliku nagłówkowym `<stdexcept>`).

Ważne elementy programu

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