# Лабораторна робота № 4

**Абстрактні класи та інтерфейси**

**Мета роботи**: Використовуючи теоретичне підґрунтя абстракті класи та інтерфейси у мові Java, виконати дії що будуть вказано в завданні до лабораторної роботи.

## Завдання до лабораторної роботи

Для виконання лабораторної роботи необхідно використовувати репозиторій на github, що був створений при виконанні завдання до першої лабораторної роботи. Результати роботи необхідно буде завантажити на github для перевірки.

При використанні IDE створити проект під назвою lab04 для розміщення результатів виконання завдань. У випадку використання звичайного текстового редактору створити окрему директорію lab04 для виконання завдань.

Виконайте наступне завдання:

|  |  |
| --- | --- |
| 1 | Створіть три класи Triangle, Quadrilateral, Circle, що розширяють абстрактний клас Figure.  Реалізуйте **конструктор** для кожного з класів.  Для трикутника конструктор у якості параметра приймає три вершини класу Point.  Для чотирикутника конструктор у якості параметра приймає чотири вершини класу Point.  Для кола конструктор у якості параметра приймає координати центу класу Point та радіус типу double.  Конструктор кожної з фігур має перевіряти, що фігура є не виродженою.  Реалізуйте методи **підрахунку площі** для кожної фігури.  Реалізуйте метод **toString()** для кожної з фігур який виводить текст у вказаному вигляді:  Трикутник - Triangle[A(x,y) B(x,y) C(x,y)]  Чотирикутник - Quadrilateral[A(x,y) B(x,y) C(x,y) D(x,y)]  Коло - Circle[(x,y) Radius]  Реалізуйте метод Point centroid()  Повернути центроїд фігури.  Іншими словами, це має бути "центроїд площі". |
|  |  |
|  |  |
|  |  |

Сформуйте звіт який містить код програм та результати тестування.

Надайте посилання на github для перевірки.

Код:

package lab4;  
  
// Клас точки  
class Point {  
 double x, y;  
  
 public Point(double x, double y) {  
 this.x = x;  
 this.y = y;  
 }  
  
 public String toString() {  
 return "(" + x + "," + y + ")";  
 }  
}  
  
// Абстрактний клас фігури  
abstract class Figure {  
 public abstract double area();  
 public abstract Point centroid();  
 public abstract String toString();  
}  
  
// Клас Трикутник  
class Triangle extends Figure {  
 private final Point a, b, c;  
  
 public Triangle(Point a, Point b, Point c) {  
 if (areColinear(a, b, c)) {  
 throw new IllegalArgumentException("Points are colinear - not a valid triangle");  
 }  
 this.a = a;  
 this.b = b;  
 this.c = c;  
 }  
  
 private boolean areColinear(Point p1, Point p2, Point p3) {  
 double area = p1.x \* (p2.y - p3.y) +  
 p2.x \* (p3.y - p1.y) +  
 p3.x \* (p1.y - p2.y);  
 return area == 0;  
 }  
  
 @Override  
 public double area() {  
 return Math.abs(  
 a.x \* (b.y - c.y) +  
 b.x \* (c.y - a.y) +  
 c.x \* (a.y - b.y)  
 ) / 2.0;  
 }  
  
 @Override  
 public Point centroid() {  
 double x = (a.x + b.x + c.x) / 3;  
 double y = (a.y + b.y + c.y) / 3;  
 return new Point(x, y);  
 }  
  
 @Override  
 public String toString() {  
 return "Triangle[" + a + " " + b + " " + c + "]";  
 }  
}  
  
// Клас Чотирикутник  
class Quadrilateral extends Figure {  
 private final Point a, b, c, d;  
  
 public Quadrilateral(Point a, Point b, Point c, Point d) {  
 if (areColinear(a, b, c) || areColinear(b, c, d) || areColinear(c, d, a) || areColinear(d, a, b)) {  
 throw new IllegalArgumentException("Invalid quadrilateral - three points are colinear");  
 }  
 this.a = a;  
 this.b = b;  
 this.c = c;  
 this.d = d;  
 }  
  
 private boolean areColinear(Point p1, Point p2, Point p3) {  
 double area = p1.x \* (p2.y - p3.y) +  
 p2.x \* (p3.y - p1.y) +  
 p3.x \* (p1.y - p2.y);  
 return area == 0;  
 }  
  
 @Override  
 public double area() {  
 // Площа чотирикутника = площа двох трикутників  
 Triangle t1 = new Triangle(a, b, c);  
 Triangle t2 = new Triangle(a, c, d);  
 return t1.area() + t2.area();  
 }  
  
 @Override  
 public Point centroid() {  
 double x = (a.x + b.x + c.x + d.x) / 4;  
 double y = (a.y + b.y + c.y + d.y) / 4;  
 return new Point(x, y);  
 }  
  
 @Override  
 public String toString() {  
 return "Quadrilateral[" + a + " " + b + " " + c + " " + d + "]";  
 }  
}  
  
// Клас Коло  
class Circle extends Figure {  
 private final Point center;  
 private final double radius;  
  
 public Circle(Point center, double radius) {  
 if (radius <= 0) {  
 throw new IllegalArgumentException("Radius must be positive");  
 }  
 this.center = center;  
 this.radius = radius;  
 }  
  
 @Override  
 public double area() {  
 return Math.PI \* radius \* radius;  
 }  
  
 @Override  
 public Point centroid() {  
 return center;  
 }  
  
 @Override  
 public String toString() {  
 return "Circle[" + center + " " + radius + "]";  
 }  
}  
  
// Основний клас Main  
public class lab1 {  
 public static void main(String[] args) {  
 // Створюємо трикутник  
 Triangle triangle = new Triangle(new Point(0, 0), new Point(4, 0), new Point(2, 3));  
 System.out.println(triangle);  
 System.out.println("Площа: " + triangle.area());  
 System.out.println("Центроїд: " + triangle.centroid());  
  
 System.out.println();  
  
 // Створюємо чотирикутник  
 Quadrilateral quad = new Quadrilateral(new Point(0, 0), new Point(4, 0), new Point(4, 3), new Point(0, 3));  
 System.out.println(quad);  
 System.out.println("Площа: " + quad.area());  
 System.out.println("Центроїд: " + quad.centroid());  
  
 System.out.println();  
  
 // Створюємо коло  
 Circle circle = new Circle(new Point(2, 2), 5);  
 System.out.println(circle);  
 System.out.println("Площа: " + circle.area());  
 System.out.println("Центроїд: " + circle.centroid());  
 }  
}

Цей код описує **геометричні фігури** через об'єктно-орієнтовану модель:

Спільна абстракція Figure.

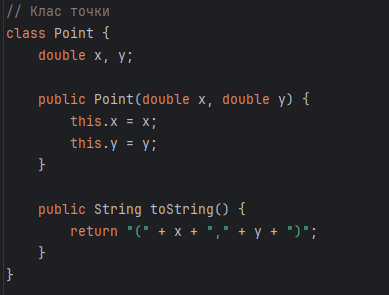
Реалізації для **трикутника**, **чотирикутника** та **кола**.

Для кожної фігури можна обчислити **площу**, **центроїд** (середину мас) та отримати опис через toString().

1)Клас Point

Представляє точку в 2D-просторі з координатами x, y.

Має toString(), щоб гарно виводити координати, наприклад: (2.0,3.0).



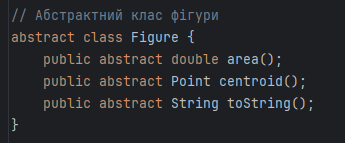
2) Figure (абстрактний клас)

Усі фігури повинні реалізовувати:

**area()** — площа,

**centroid()** — координати центру фігури,

**toString()** — опис фігури у вигляді рядка



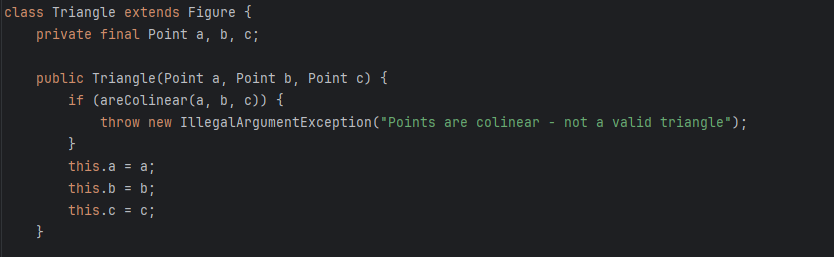
3)Клас Triangle

Створюється з 3 точок a, b, c.

Перевіряє, що точки не лежать на одній прямій (**не колінеарні**).

Площа обчислюється за формулою через координати.

Центроїд — середнє значення координат усіх вершин.



4)Клас Quadrilateral

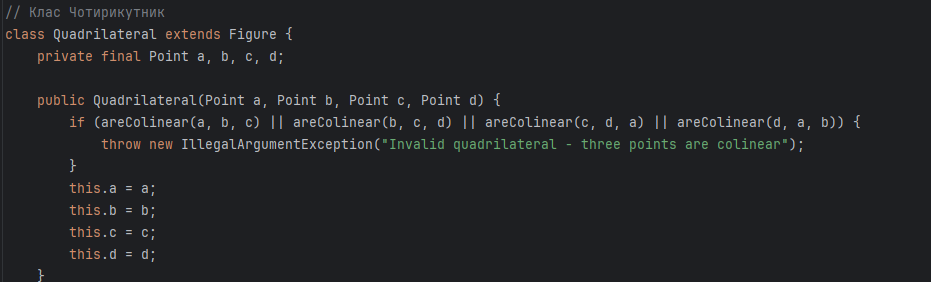
Створюється з 4 точок a, b, c, d.

Перевіряє, що **жодні 3 послідовні точки не лежать на одній прямій**.

Розбивається на два трикутники:

Площа = площа Triangle(a,b,c) + Triangle(a,c,d).

Центроїд — середнє арифметичне координат усіх вершин.

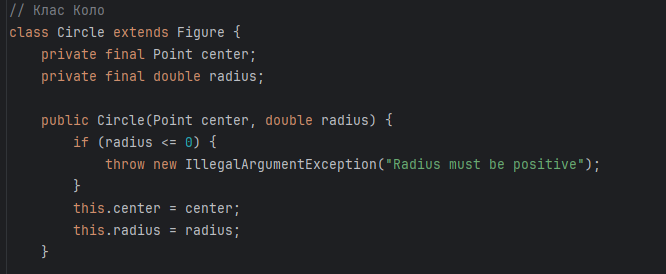


5)Клас Circle Створюється з центру (Point) і радіуса (double).

Радіус перевіряється > 0.

Площа: πr2

Центроїд — це сам центр.



6) Main (метод main)

У main() створюються 3 фігури:

1. **Трикутник** (0,0)-(4,0)-(2,3)
2. **Чотирикутник** (0,0)-(4,0)-(4,3)-(0,3)
3. **Коло** з центром (2,2) і радіусом 5

Для кожної виводиться:

* опис фігури (toString()),
* площа (area()),
* центроїд (centroid())

