Лабораторна робота №4

Circle

package lab\_04;

class Circle extends Figure {

    private Point center;

    private double radius;

    Circle(Point center, double radius) {

        if (radius <= 0) {

            throw new IllegalArgumentException("Invalid radius");

        }

        this.center = center;

        this.radius = radius;

    }

    @Override

    double area() {

        return Math.PI \* radius \* radius;

    }

    @Override

    Point centroid() {

        return center;

    }

    @Override

    public String toString() {

        return "Circle[" + center + " Radius " + radius + "]";

    }

}

Figure

package lab\_04;

abstract class Figure {

    abstract double area();

    abstract Point centroid();

}

Main

package lab\_04;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Введіть тип фігури, яку ви хочите обсилити (Трикутник(1) / Чотирикутник(2) / Коло(3)):");

        int choice = scanner.nextInt();

        switch (choice) {

        case 1:

            System.out.println("Введіть координати трикутника (x1 y1 x2 y2 x3 y3):");

            double x1 = scanner.nextDouble();

            double y1 = scanner.nextDouble();

            double x2 = scanner.nextDouble();

            double y2 = scanner.nextDouble();

            double x3 = scanner.nextDouble();

            double y3 = scanner.nextDouble();

            Point a = new Point(x1, y1);

            Point b = new Point(x2, y2);

            Point c = new Point(x3, y3);

            Triangle triangle = new Triangle(a, b, c);

            System.out.println("Трикутник: " + triangle);

            System.out.println("Площа: " + triangle.area());

            System.out.println("Центроїд: " + triangle.centroid());

            break;

        case 2:

            System.out.println("Введіть координати чотирикутника (x1 y1 x2 y2 x3 y3 x4 y4):");

            double x1q = scanner.nextDouble();

            double y1q = scanner.nextDouble();

            double x2q = scanner.nextDouble();

            double y2q = scanner.nextDouble();

            double x3q = scanner.nextDouble();

            double y3q = scanner.nextDouble();

            double x4q = scanner.nextDouble();

            double y4q = scanner.nextDouble();

            Point Aq = new Point(x1q, y1q);

            Point Bq = new Point(x2q, y2q);

            Point Cq = new Point(x3q, y3q);

            Point Dq = new Point(x4q, y4q);

            Quadrilateral quadrilateral = new Quadrilateral(Aq, Bq, Cq, Dq);

            System.out.println("Чотирикутник: " + quadrilateral);

            System.out.println("Площа: " + quadrilateral.area());

            System.out.println("Центроїд: " + quadrilateral.centroid());

            break;

        case 3:

            System.out.println("Введіть координати центра кола (x y) та радіус:");

            double xc = scanner.nextDouble();

            double yc = scanner.nextDouble();

            double radius = scanner.nextDouble();

            Point center = new Point(xc, yc);

            Circle circle = new Circle(center, radius);

            System.out.println("Коло: " + circle);

            System.out.println("Площа: " + circle.area());

            System.out.println("Центроїд: " + circle.centroid());

            break;

        default:

            System.out.println("Invalid choice!");

    }

    scanner.close();

    }

}

Point

package lab\_04;

public class Point {

    double x;

    double y;

    public Point(double x, double y) {

        this.x = x;

        this.y = y;

    }

    @Override

    public String toString() {

        return "(" + x + ", " + y + ")";

    }

}

Quadrilateral

package lab\_04;

import java.util.ArrayList;

class Quadrilateral extends Figure {

    private Point a;

    private Point b;

    private Point c;

    private Point d;

    Quadrilateral(Point a, Point b, Point c, Point d) {

        if (!isValidQuadrilateral(a, b, c, d)) {

            throw new IllegalArgumentException("Invalid quadrilateral vertices");

        }

        this.a = a;

        this.b = b;

        this.c = c;

        this.d = d;

    }

    private boolean isValidQuadrilateral(Point a, Point b, Point c, Point d) {

        ArrayList<Point> points = new ArrayList<>();

        points.add(a);

        points.add(b);

        points.add(c);

        points.add(d);

        for (int i = 0; i < 4; i++) {

            Point p1 = points.get(i);

            Point p2 = points.get((i + 1) % 4);

            Point p3 = points.get((i + 2) % 4);

            if (isCollinear(p1, p2, p3)) {

                return false;

            }

        }

        return true;

    }

    private boolean isCollinear(Point a, Point b, Point c) {

        return Math.abs((a.x \* (b.y - c.y) + b.x \* (c.y - a.y) + c.x \* (a.y - b.y))) < 1e-9;

    }

    @Override

    double area() {

        double areaTriangle1 = triangleArea(a, b, c);

        double areaTriangle2 = triangleArea(c, b, d);

        return areaTriangle1 + areaTriangle2;

    }

    private double triangleArea(Point A, Point B, Point C) {

        return Math.abs((A.x\*(B.y - C.y) + B.x\*(C.y - A.y) + C.x\*(A.y - B.y)) / 2);

    }

    @Override

    Point centroid() {

        double centroidX = (a.x + b.x + c.x + d.x) / 4;

        double centroidY = (a.y + b.y + c.y + d.y) / 4;

        return new Point(centroidX, centroidY);

    }

    @Override

    public String toString() {

        return "Quadrilateral[A" + a + " B" + b + " C" + c + " D" + d + "]";

    }

}

Triangle

package lab\_04;

class Triangle extends Figure {

    private Point a;

    private Point b;

    private Point c;

    Triangle(Point a, Point b, Point c) {

        if (!isValidTriangle(a, b, c)) {

            throw new IllegalArgumentException("Invalid triangle vertices");

        }

        this.a = a;

        this.b = b;

        this.c = c;

    }

    private boolean isValidTriangle(Point a, Point b, Point c) {

        double ab = distance(a, b);

        double bc = distance(b, c);

        double ac = distance(a, c);

        return (ab + bc > ac) && (ab + ac > bc) && (ac + bc > ab);

    }

    private double distance(Point p1, Point p2) {

        return Math.sqrt(Math.pow(p2.x - p1.x, 2) + Math.pow(p2.y - p1.y, 2));

    }

    @Override

    double area() {

        return Math.abs((a.x \* (b.y - c.y) + b.x \* (c.y - a.y) + c.x \* (a.y - b.y)) / 2);

    }

    @Override

    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" + a + " B" + b + " C" + c + "]";

    }

}

Результат:

