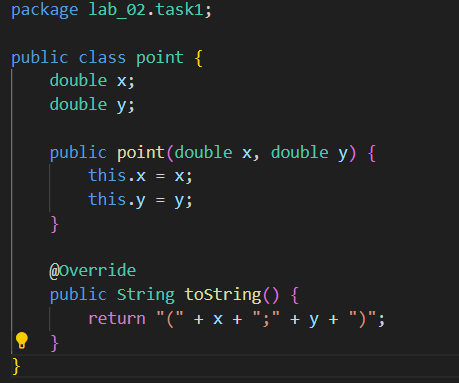
Лабораторна робота №2

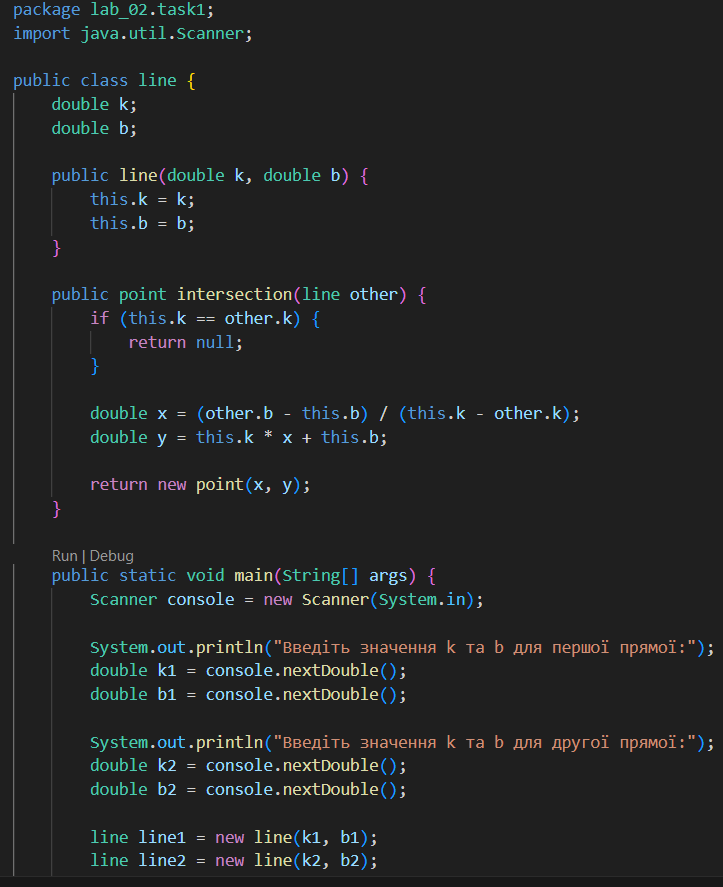
Звіт

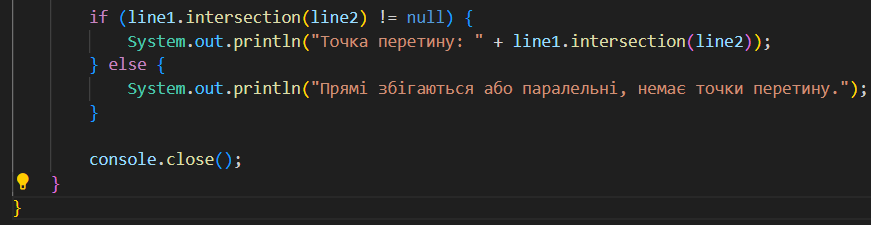
1.

**Point**

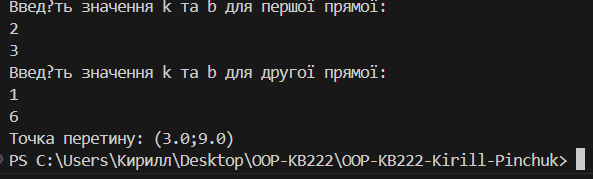


Line



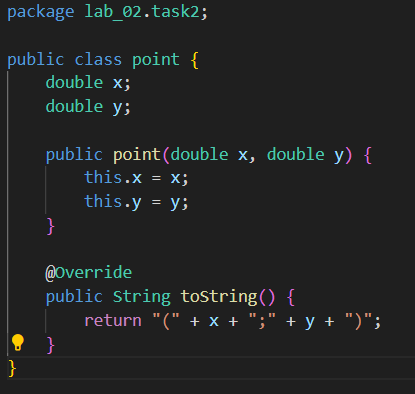


Результат:



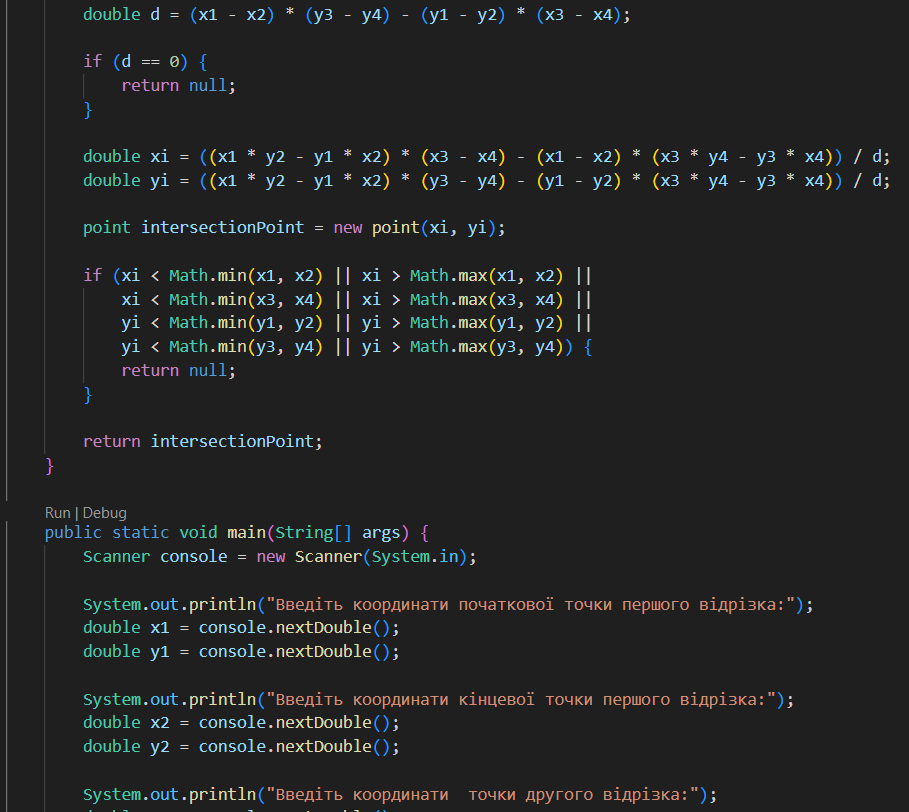
2.

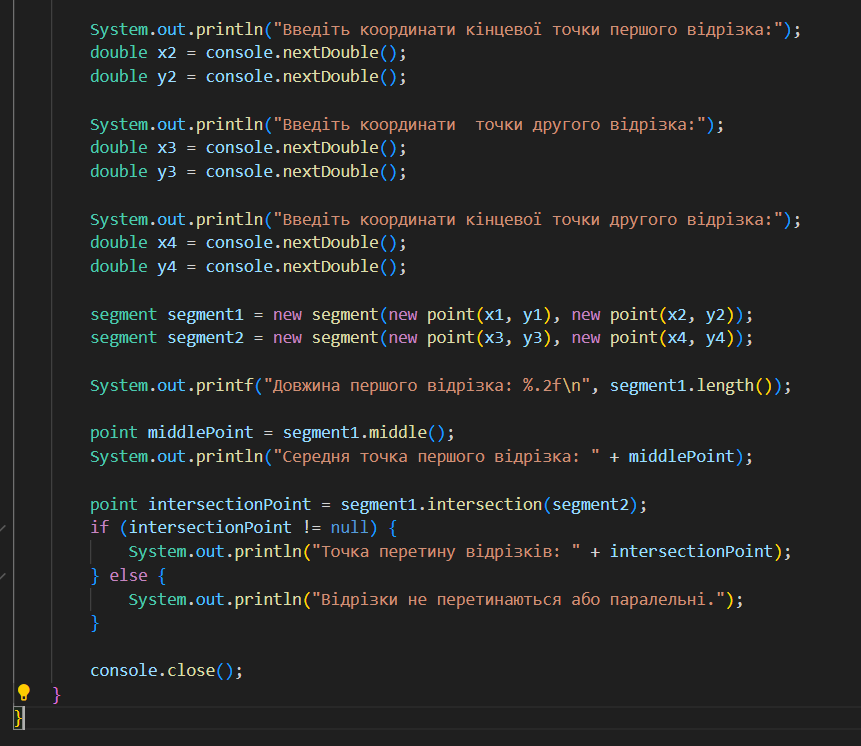
**Point**

****

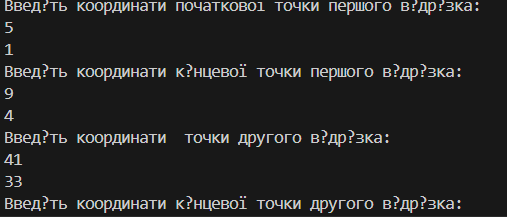
**Segment**

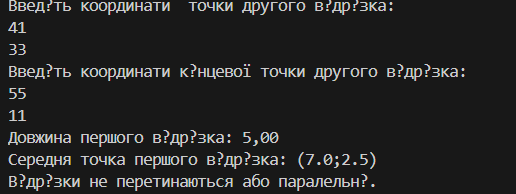
****

****

****

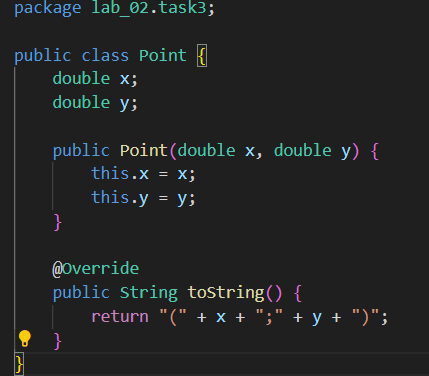
Результат:





3.

**Point**

****

**Triangle**

package lab\_02.task3;

import java.util.Scanner;

public class Triangle {

    private Point vertex1;

    private Point vertex2;

    private Point vertex3;

    public Triangle(Point vertex1, Point vertex2, Point vertex3) {

        if (areVerticesCollinear(vertex1, vertex2, vertex3)) {

            throw new IllegalArgumentException("Три точки не можуть бути на одній прямій");

        }

        if (!isValidTriangle(vertex1, vertex2, vertex3)) {

            throw new IllegalArgumentException("Три точки не утворюють трикутник");

        }

        this.vertex1 = vertex1;

        this.vertex2 = vertex2;

        this.vertex3 = vertex3;

    }

    private boolean areVerticesCollinear(Point vertex1, Point vertex2, Point vertex3) {

        double area = 0.5 \* Math.abs((vertex2.x - vertex1.x) \* (vertex3.y - vertex1.y) - (vertex3.x - vertex1.x) \* (vertex2.y - vertex1.y));

        return area == 0;

    }

    private boolean isValidTriangle(Point vertex1, Point vertex2, Point vertex3) {

        double side1 = calculateDistance(vertex1, vertex2);

        double side2 = calculateDistance(vertex1, vertex3);

        double side3 = calculateDistance(vertex2, vertex3);

        return side1 + side2 > side3 && side1 + side3 > side2 && side2 + side3 > side1;

    }

    private double calculateDistance(Point p1, Point p2) {

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

    }

    public double area() {

        double side1 = calculateDistance(vertex1, vertex2);

        double side2 = calculateDistance(vertex1, vertex3);

        double side3 = calculateDistance(vertex2, vertex3);

        double p = (side1 + side2 + side3) / 2;

        return Math.sqrt(p \* (p - side1) \* (p - side2) \* (p - side3));

    }

    public Point centroid() {

        double centroidX = (vertex1.x + vertex2.x + vertex3.x) / 3;

        double centroidY = (vertex1.y + vertex2.y + vertex3.y) / 3;

        return new Point(centroidX, centroidY);

    }

    public static void main(String[] args) {

        Scanner console = new Scanner(System.in);

        System.out.println("Введіть координати трьох вершин трикутника:");

        System.out.println("Вершина A:");

        double x1 = console.nextDouble();

        double y1 = console.nextDouble();

        System.out.println("Вершина B:");

        double x2 = console.nextDouble();

        double y2 = console.nextDouble();

        System.out.println("Вершина C:");

        double x3 = console.nextDouble();

        double y3 = console.nextDouble();

        Triangle triangle = new Triangle(new Point(x1, y1), new Point(x2, y2), new Point(x3, y3));

        System.out.printf("Площа трикутника: %.2f\n", triangle.area());

        Point centroid = triangle.centroid();

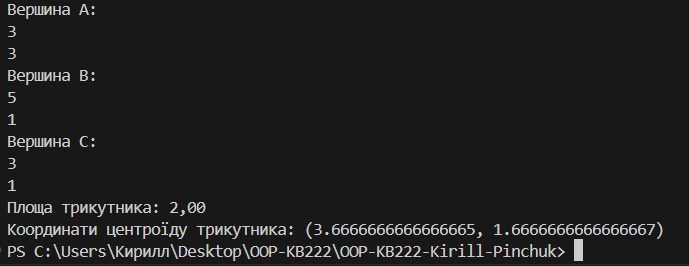
        System.out.println("Координати центроїду трикутника: (" + centroid.x + ", " + centroid.y + ")");

        console.close();

    }

}

**Результат:**

****