Selected files

3 printable files

Triangle.java
TestTriangle.java
Point2D.java

Triangle.java

```java
public class Triangle {
    private Point2D p1;
    private Point2D p2;
    private Point2D p3;

    // Initialization constructor
    public Triangle(Point2D p1, Point2D p2, Point2D p3) {
        this.p1 = new Point2D(p1);
        this.p2 = new Point2D(p2);
        this.p3 = new Point2D(p3);
    }

    // Calculate the perimeter of the triangle
    // public double perimeter() {
    // return Point2D.distance(p1, p2) + Point2D.distance(p2, p3) +
    // Point2D.distance(p3, p1);
    // }

    // // Calculate the area of the triangle
    // public double area() {
    // double a = Point2D.distance(p1, p2);
    // double b = Point2D.distance(p2, p3);
    // double c = Point2D.distance(p3, p1);
    // double s = (a + b + c) / 2; // semi-perimeter
    // return Math.sqrt(s * (s - a) * (s - b) * (s - c));
    // }

    // Getters for vertices
    public Point2D getP1() {
        return p1;
    }

    public Point2D getP2() {
        return p2;
    }

    public Point2D getP3() {
        return p3;
    }
}

```

TestTriangle.java

```java
public class TestTriangle {
    public static void main(String[] args) {
        // Create points for vertices of the triangle
        Point2D p1 = new Point2D(0, 0);
        Point2D p2 = new Point2D(3, 0);
```

```
        Point2D p3 = new Point2D(0, 4);

        // Create a triangle using the points
        Triangle triangle = new Triangle(p1, p2, p3);

        // Display the vertices of the triangle
        System.out.println("Vertices of the triangle:");
        System.out.println("Point 1: " + triangle.getP1());
        System.out.println("Point 2: " + triangle.getP2());
        System.out.println("Point 3: " + triangle.getP3());

        // Calculate and display the perimeter of the triangle
        System.out.println("Perimeter of the triangle: " + triangle.perimeter());

        // Calculate and display the area of the triangle
        System.out.println("Area of the triangle: " + triangle.area());
    }
}

```

Point2D.java

```
import java.util.Scanner;

public class Point2D {
    private int x;
    private int y;

    // Default constructor initializes the point at the origin (0,0)
    public Point2D() {
        this.x = 0;
        this.y = 0;
    }

    // Constructor with coordinates
    public Point2D(int x, int y) {
        this.x = x;
        this.y = y;
    }

    // Copy constructor
    public Point2D(Point2D p) {
        this.x = p.x;
        this.y = p.y;
    }

    // Method to input coordinates from keyboard
    public void input() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter x coordinate: ");
        this.x = scanner.nextInt();
        System.out.print("Enter y coordinate: ");
        this.y = scanner.nextInt();
    }

    // String representation of coordinates
    @Override
    public String toString() {
        return "(" + x + ", " + y + ")";
```

```
    }

    // Move point to new coordinates
    public void move(int x, int y) {
        this.x = x;
        this.y = y;
    }

    // Check if point is at origin
    public boolean isOrigin() {
        return (x == 0 && y == 0);
    }

    // Calculate distance from this point to another point p
    public double distance(Point2D p) {
        int dx = this.x - p.x;
        int dy = this.y - p.y;
        return Math.sqrt(dx * dx + dy * dy);
    }

    // Static method to calculate distance between two points
    public static double distance(Point2D p1, Point2D p2) {
        int dx = p1.x - p2.x;
        int dy = p1.y - p2.y;
        return Math.sqrt(dx * dx + dy * dy);
    }

    // Getter methods for x and y coordinates
    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }
}

```