

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”
КАФЕДРА ИИТ

ОТЧЁТ
по лабораторной работе №7

Выполнил:
студент 3 курса
группы ПО-9
Тусюк Т.В.

Проверил:
Крощенко А.А.

Брест 2024

Цель работы: приобрести навыки применения паттернов проектирования при решении практических задач с использованием языка Java.

Задание 1:

Реализовать соответствующие классы, указанные в задании;

- Организовать ввод параметров для создания объектов (можно использовать файлы);
- Осуществить визуализацию графических примитивов, решить поставленную задачу.

Создать классы Point и Line. Объявить массив из n объектов класса Point. Для объекта класса Line определить, какие из объектов Point лежат на одной стороне от прямой линии и какие на другой. Реализовать ввод данных для объекта Line и случайное задание данных для объекта Point.

Выполнение задания:

```
package com.example.sppa;

import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Line;
import javafx.stage.Stage;

import java.util.ArrayList;
import java.util.List;
import java.util.Random;

class Point {
    private double x;
    private double y;

    public Point(double x, double y) {
        this.x = x;
        this.y = y;
    }

    public double getX() {
        return x;
    }

    public double getY() {
        return y;
    }
}

class LineSegment {
    private Point point1;
    private Point point2;

    public LineSegment(Point point1, Point point2) {
        this.point1 = point1;
        this.point2 = point2;
    }
}
```

```

    public Point getPoint1() {
        return point1;
    }

    public Point getPoint2() {
        return point2;
    }

    public boolean isPointOnSameSide(Point point) {
        double dx1 = point2.getX() - point1.getX();
        double dy1 = point2.getY() - point1.getY();
        double dx2 = point.getX() - point1.getX();
        double dy2 = point.getY() - point1.getY();
        return (dx1 * dy2 - dy1 * dx2) >= 0;
    }
}

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) {
        int n = 10; // Количество точек

        List<Point> points = generateRandomPoints(n);
        LineSegment lineSegment = generateRandomLineSegment();

        Group root = new Group();
        Scene scene = new Scene(root, 600, 400, Color.WHITE);

        drawLineSegment(root, lineSegment);
        drawPoints(root, points);

        primaryStage.setTitle("Points on Line Segment");
        primaryStage.setScene(scene);
        primaryStage.show();

        // Определение, на какой стороне от линии находятся точки
        for (Point point : points) {
            boolean onSameSide = lineSegment.isPointOnSameSide(point);
            System.out.println("Point (" + point.getX() + ", " + point.getY() + ")
is on the same side as line segment: " + onSameSide);
        }

        private List<Point> generateRandomPoints(int n) {
            Random random = new Random();
            List<Point> points = new ArrayList<>();
            for (int i = 0; i < n; i++) {
                double x = random.nextDouble() * 600; // случайная координата x в
пределах окна
                double y = random.nextDouble() * 400; // случайная координата y в
пределах окна
                points.add(new Point(x, y));
            }
            return points;
        }

        private LineSegment generateRandomLineSegment() {
            Random random = new Random();
            Point point1 = new Point(random.nextDouble() * 600, random.nextDouble() *

```

```

400);
    Point point2 = new Point(random.nextDouble() * 600, random.nextDouble() *
400);
    return new LineSegment(point1, point2);
}

private void drawLineSegment(Group root, LineSegment lineSegment) {
    Point point1 = lineSegment.getPoint1();
    Point point2 = lineSegment.getPoint2();
    Line lineUI = new Line(point1.getX(), point1.getY(), point2.getX(),
point2.getY());
    lineUI.setStroke(Color.RED);
    root.getChildren().add(lineUI);
}

private void drawPoints(Group root, List<Point> points) {
    for (Point point : points) {
        Circle circle = new Circle(point.getX(), point.getY(), 3, Color.BLUE);
        root.getChildren().add(circle);
    }
}

public static void main(String[] args) {
    launch(args);
}
}

```

Результат:

 Points on Line Segment



```

"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:D:\IntelliJ IDEA 2024.1.1\lib\idea_
Point (507.1695790038705, 237.47714592179022) is on the same side as line segment: false
Point (293.71881379042486, 58.43176089369546) is on the same side as line segment: true
Point (530.298679603715, 172.36086785988957) is on the same side as line segment: true
Point (536.3858785326236, 25.255438548599507) is on the same side as line segment: true
Point (557.6364784765786, 229.97485750867574) is on the same side as line segment: false
Point (595.1491302704627, 196.9402020005727) is on the same side as line segment: true
Point (377.13941794386466, 286.4683516394331) is on the same side as line segment: false
Point (96.08730287358946, 188.7032249336299) is on the same side as line segment: false
Point (371.9210355539255, 266.1671217513981) is on the same side as line segment: false
Point (595.0046075549673, 184.56917079856998) is on the same side as line segment: true

```

Задание 2:

9) Остров Минковского

Выполнение задания:

```
package com.example.sppa;

import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.shape.Rectangle;
import javafx.stage.Stage;

public class Task_02 extends Application {

    @Override
    public void start(Stage primaryStage) {
        int width = 600;
        int height = 600;
        int depth = 4; // Глубина рекурсии

        Group root = new Group();
        Scene scene = new Scene(root, width, height);
        primaryStage.setScene(scene);
        primaryStage.setTitle("Minkowski Island");

        drawMinkowskiIsland(root, width / 4, height / 4, width / 2, height / 2,
depth);

        primaryStage.show();
    }

    private void drawMinkowskiIsland(Group group, double x, double y, double width,
double height, int depth) {
        if (depth == 0) {
            Rectangle square = new Rectangle(x, y, width, height);
            square.setFill(Color.BLACK);
            group.getChildren().add(square);
        } else {
            double newWidth = width / 2;
            double newHeight = height / 2;

            // Отрисовка внешнего квадрата
            Rectangle outerSquare = new Rectangle(x, y, width, height);
            outerSquare.setFill(Color.WHITE);
            group.getChildren().add(outerSquare);

            // Вычисление координат внутренних квадратов
            double innerX = x + newWidth / 2;
            double innerY = y + newHeight / 2;

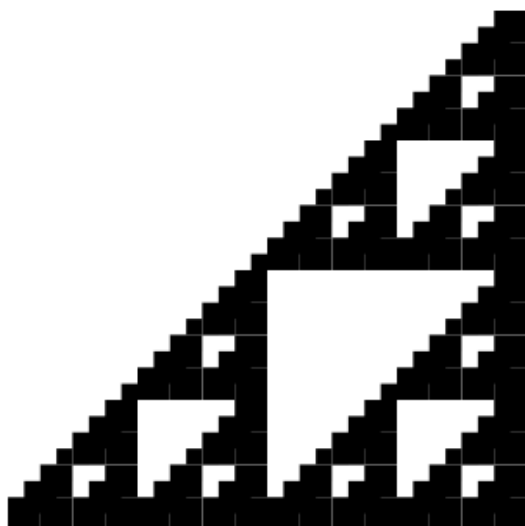
            // Рекурсивное построение внутренних квадратов
            drawMinkowskiIsland(group, innerX, innerY, newWidth, newHeight, depth -
1);
            drawMinkowskiIsland(group, x, y + newHeight, newWidth, newHeight, depth
- 1);
            drawMinkowskiIsland(group, x + newWidth, y + newHeight, newWidth,
newHeight, depth - 1);
            drawMinkowskiIsland(group, x + newWidth, y, newWidth, newHeight, depth -
1);
        }
    }
}
```

```
    }  
}  
  
public static void main(String[] args) {  
    launch(args);  
}  
}
```

Результат:

 Minkowski Island

— □ ×



Вывод: приобрел навыки применения паттернов проектирования при решении практических задач с использованием языка Java.