

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”  
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ  
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

## Отчет по лабораторной работе №7

Специальность ПО-9

Выполнил  
А. С. Мисюк,  
студент группы ПО-9  
Проверил  
А. А. Крощенко,  
ст. преп. кафедры ИИТ,  
«\_\_» \_\_\_\_\_ 2024 г.

Брест 2024

Цель работы: освоить возможности языка программирования Java в построении графических приложений.

## Вариант №2

**Задание 1.** Определить класс Rectangle и класс Point. Объявить массив из n объектов класса Point. Написать функцию, определяющую, какая из точек лежит снаружи, а какая – внутри прямоугольника.

### Код программы:

```
package com.labrab7.task1;

import ...

/** ... */
public class Main
{
    public static void main(String[] args)
    {
        List<Point> points = new ArrayList<>();

        try (BufferedReader reader = new BufferedReader(new FileReader("task1_input.txt")))
        {
            // Чтение координат верхнего левого и нижнего правого углов прямоугольника
            String line = reader.readLine();
            String[] rectangleCoords = line.split(" ");
            int rectangleX1 = Integer.parseInt(rectangleCoords[0]);
            int rectangleY1 = Integer.parseInt(rectangleCoords[1]);
            int rectangleX2 = Integer.parseInt(rectangleCoords[2]);
            int rectangleY2 = Integer.parseInt(rectangleCoords[3]);
            Rectangle rectangle = new Rectangle(new Point(rectangleX1, rectangleY1), new
Point(rectangleX2, rectangleY2));

            // Чтение количества точек
            line = reader.readLine();
            int n = Integer.parseInt(line);

            // Чтение координат точек
            for (int i = 0; i < n; i++)
            {
                line = reader.readLine();
                String[] pointCoords = line.split(" ");
                int pointX = Integer.parseInt(pointCoords[0]);
                int pointY = Integer.parseInt(pointCoords[1]);
                points.add(new Point(pointX, pointY));
            }

            for (Point point : points)
            {
                if (rectangle.containsPoint(point))
                {
                    System.out.println("Точка " + point + " находится внутри прямоугольника.");
                } else
                {
                    System.out.println("Точка " + point + " находится снаружи прямоугольника.");
                }
            }

            // Рисование
            SwingUtilities.invokeLater(() -> {
```

```

        JFrame frame = new JFrame("Drawing");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        DrawingPanel panel = new DrawingPanel(points, rectangle);
        panel.setPreferredSize(new Dimension(800, 600));

        frame.getContentPane().add(panel);
        frame.pack();
        frame.setVisible(true);
    });

    } catch (IOException e)
    {
        System.out.println("Ошибка при чтении файла: " + e.getMessage());
    }
}

```

```
package com.labrab7.task1;
```

```

@SuppressWarnings("FieldMayBeFinal")
public class Point
{
    private int x;
    private int y;

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

    public int getX()
    {
        return x;
    }

    public int getY()
    {
        return y;
    }

    @Override
    public String toString()
    {
        return "(" + this.getX() + ", " + this.getY() + ")";
    }
}

```

```
package com.labrab7.task1;
```

```

@SuppressWarnings("FieldMayBeFinal")
public class Rectangle
{
    private Point topLeft;
    private Point bottomRight;

    public Rectangle(Point topLeft, Point bottomRight)
    {
        this.topLeft = topLeft;
        this.bottomRight = bottomRight;
    }

    public Point getTopLeft()
    {
        return topLeft;
    }
}

```

```

public Point getBottomRight()
{
    return bottomRight;
}

public boolean containsPoint(Point point)
{
    int x = point.getX();
    int y = point.getY();

    int topLeftX = topLeft.getX();
    int topLeftY = topLeft.getY();
    int bottomRightX = bottomRight.getX();
    int bottomRightY = bottomRight.getY();

    return x >= topLeftX && x <= bottomRightX && y >= topLeftY && y <= bottomRightY;
}
}

package com.labrab7.task1;

import ...

@SuppressWarnings("FieldMayBeFinal")
class DrawingPanel extends JPanel
{
    private List<Point> points;
    private Rectangle rectangle;

    public DrawingPanel(List<Point> points, Rectangle rectangle) {
        this.points = points;
        this.rectangle = rectangle;
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);

        // Рисование прямоугольника
        g.drawRect(
            rectangle.getTopLeft().getX(),
            rectangle.getTopLeft().getY(),
            rectangle.getBottomRight().getX() - rectangle.getTopLeft().getX(),
            rectangle.getBottomRight().getY() - rectangle.getTopLeft().getY()
        );

        // Рисование точек
        for (Point point : points) {
            int x = point.getX();
            int y = point.getY();
            g.fillOval(x - 2, y - 2, 5, 5);
        }
    }
}

```

## Пример

```

>java.exe com.labrab7.task1.Main
Точка (20, 20) находится снаружи прямоугольника.
Точка (60, 90) находится снаружи прямоугольника.
Точка (120, 150) находится внутри прямоугольника.
Точка (150, 450) находится внутри прямоугольника.
Точка (400, 350) находится снаружи прямоугольника.

```

```
(task1_input.txt)
50 100 200 500
5
20 20
60 90
120 150
150 450
400 350
```

**Задание 2.** Реализовать построение заданного типа фрактала по варианту. Везде, где это необходимо, предусмотреть ввод параметров, влияющих на внешний вид фрактала. Н-фрактал

### Код программы:

```
package com.labrab7.task2;

import ...

/** ... */
public class Main extends JFrame
{
    public static void main(String[] args)
    {
        SwingUtilities.invokeLater(() -> new Main(args));
    }

    public Main(String[] args)
    {
        setTitle("H-Fractal");
        setSize(600, 615);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        int recursionDepth = Integer.parseInt(args[0]); // глубина рекурсии

        FractalPanel fractalPanel = new FractalPanel(recursionDepth);
        fractalPanel.setBackground(Color.BLACK);
        getContentPane().add(fractalPanel);

        setVisible(true);
    }
}

package com.labrab7.task2;

import ...

@SuppressWarnings("FieldMayBeFinal")
public class FractalPanel extends JPanel
{
    private int recursionDepth;
    private static Color[] colors = {
        Color.red, Color.blue, Color.green,
        Color.orange, Color.cyan, Color.gray
    };

    public FractalPanel(int recursionDepth)
    {
        super();
    }
}
```

```

        this.recursionDepth = recursionDepth;
    }

    @Override
    protected void paintComponent(Graphics g)
    {
        super.paintComponent(g);
        drawFractal(g, getWidth() / 2, getHeight() / 2, 140, 0);
    }

    private void drawFractal(Graphics g, int x, int y, int size, int depth)
    {
        if (depth >= recursionDepth) return;

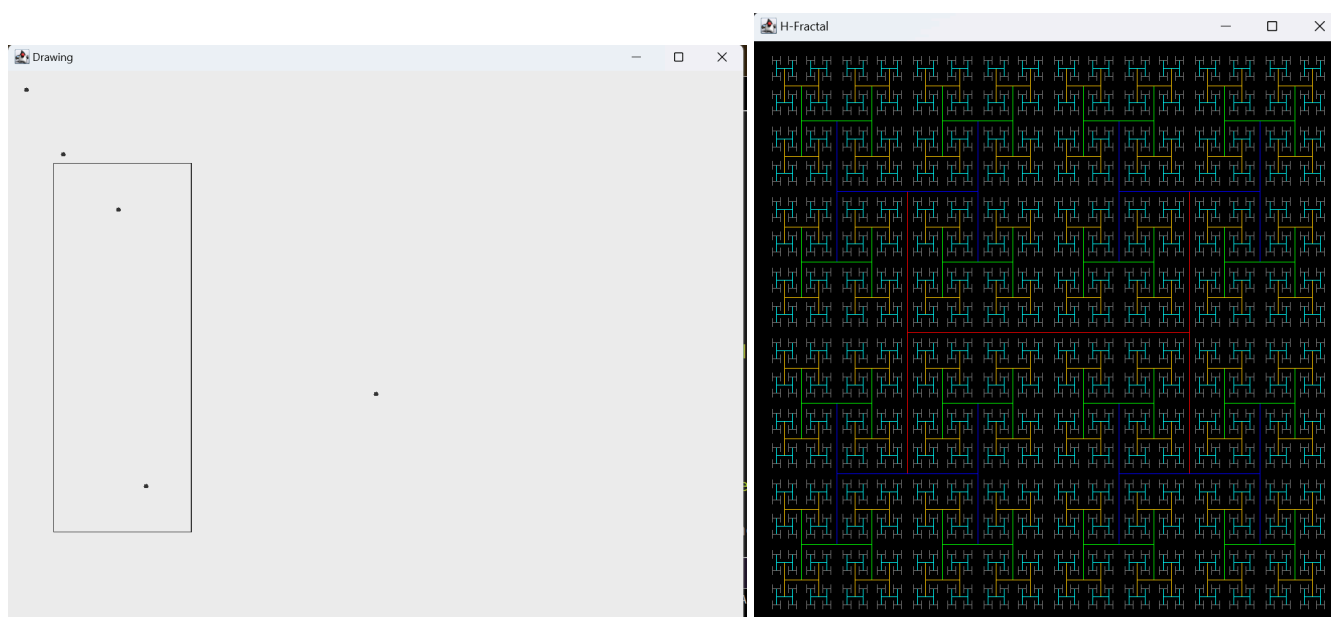
        g.setColor(colors[depth % colors.length]);

        // рисуем основную фигуру
        g.drawLine(x - size, y, x + size, y);
        g.drawLine(x - size, y - size, x - size, y + size);
        g.drawLine(x + size, y - size, x + size, y + size);

        // рекурсивно рисуем дочерние фракталы
        drawFractal(g, x - size, y - size, size / 2, depth + 1);
        drawFractal(g, x + size, y - size, size / 2, depth + 1);
        drawFractal(g, x - size, y + size, size / 2, depth + 1);
        drawFractal(g, x + size, y + size, size / 2, depth + 1);
    }
}

```

## Рисунки с результатами работы программы



<C:\Users\misij\Documents\SPP\labrab7\out\production\labrab7> com.labrab7.task1.Main

Точка (20, 20) находится снаружи прямоугольника.

Точка (60, 90) находится снаружи прямоугольника.

Точка (120, 150) находится внутри прямоугольника.

Точка (150, 450) находится внутри прямоугольника.

Точка (400, 350) находится снаружи прямоугольника.

**Вывод:** исследовал создание графических примитивов и их использование в языке программирования Java, приобрел практические навыки в области графических возможностей Java.