

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №7

По дисциплине: «Современные платформы программирования»

Выполнил:

студент 3 курса

группы ПО-8

Бубен С.О.

Проверил:

Крощенко А.А

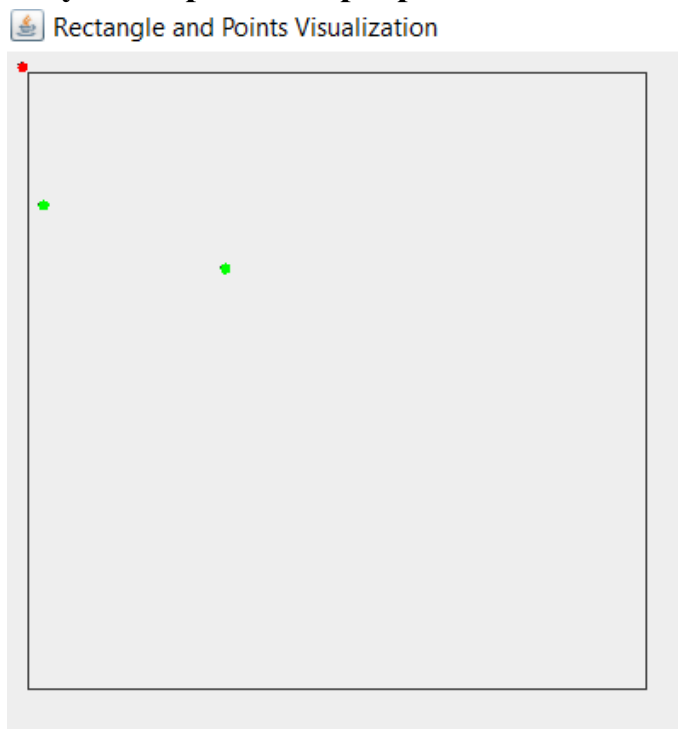
Цель работы: освоить возможности языка программирования Java в построении графических приложений.

Вариант 2

Задание 1

Реализовать соответствующие классы, указанные в задании; Организовать ввод параметров для создания объектов (можно использовать файлы); Осуществить визуализацию графических примитивов. 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.

Результат работы программы:



```
Точки, которые находятся внутри прямоугольника:  
(100.0, 100.0)  
(15.0, 70.0)  
Точки, которые не находятся внутри прямоугольника:  
(5.0, 5.0)
```

Код:

Point.java

```
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;  
    }  
}
```

Rectangle.java

```
class Rectangle {  
    private Point topLeft; private Point bottomRight;  
  
    public Rectangle(Point topLeft, Point bottomRight) { this.topLeft = topLeft;  
        this.bottomRight = bottomRight;  
    }  
  
    public boolean contains(Point point) { double x = point.getX();  
        double y = point.getY();  
  
        double topLeftX = topLeft.getX(); double topLeftY = topLeft.getY();  
        double bottomRightX = bottomRight.getX(); double bottomRightY = bottomRight.getY();  
  
        return x >= topLeftX && x <= bottomRightX && y >= topLeftY && y <= bottomRightY;  
    }  
  
    public double getTopLeftX() { return topLeft.getX();  
    }  
  
    public double getTopLeftY() { return topLeft.getY();  
    }  
  
    public double getBottomRightX() {  
        return bottomRight.getX();  
    }  
  
    public double getBottomRightY() { return bottomRight.getY();  
    }  
}
```

Main.java

```
import javax.swing.*;  
import java.awt.*;  
import java.util.Scanner;  
  
public class Main {  
    private static Rectangle rectangle; private static Point[] points;  
  
    public static void main(String[] args) { Scanner scanner = new Scanner(System.in);  
  
        System.out.println("Введите координаты верхнего левого угла прямоугольника (x, y):");  
        double topLeftX = scanner.nextDouble(); double topLeftY = scanner.nextDouble();  
        Point topLeftPoint = new Point(topLeftX, topLeftY);
```

```

System.out.println("Введите координаты нижнего правого угла прямоугольника (x, y):");
double bottomRightX = scanner.nextDouble(); double bottomRightY = scanner.nextDouble();
Point bottomRightPoint = new Point(bottomRightX, bottomRightY);
rectangle = new Rectangle(topLeftPoint, bottomRightPoint); System.out.println("Введите количество точек:");
int n = scanner.nextInt();
points = new Point[n];
System.out.println("Введите координаты точек (x, y):"); for (int i = 0; i < n; i++) {
double x = scanner.nextDouble(); double y = scanner.nextDouble(); points[i] = new Point(x, y);
}

scanner.close();

System.out.println("Точки, которые находятся внутри прямоугольника:");
for (Point point : points) {
if (rectangle.contains(point)) {
System.out.println("(" + point.getX() + ", " + point.getY() +
");");
}
}

System.out.println("Точки, которые не находятся внутри прямоугольника:");
for (Point point : points) {
if (!rectangle.contains(point)) {
System.out.println("(" + point.getX() + ", " + point.getY() +
");");
}
}

JFrame frame = new JFrame("Rectangle and Points Visualization"); frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JPanel panel = new JPanel() { @Override
protected void paintComponent(Graphics g) { super.paintComponent(g);
draw(g);
}
};

frame.add(panel); frame.setSize(400, 400); frame.setVisible(true);
}

private static void draw(Graphics g) {
int topLeftX = (int) rectangle.getTopLeftX(); int topLeftY = (int) rectangle.getTopLeftY();
int bottomRightX = (int) rectangle.getBottomRightX(); int bottomRightY = (int) rectangle.getBottomRightY();
g.drawRect(topLeftX, topLeftY, bottomRightX - topLeftX, bottomRightY
- topLeftY);

for (Point point : points) { int x = (int) point.getX(); int y = (int) point.getY(); g.fillOval(x, y, 3, 3);
}

for (Point point : points) {
if (rectangle.contains(point)) { g.setColor(Color.GREEN); int x = (int) point.getX(); int y = (int) point.getY();
g.fillOval(x, y, 5, 5);
} else {
g.setColor(Color.RED);
int x = (int) point.getX(); int y = (int) point.getY(); g.fillOval(x, y, 5, 5);
}
}
}
}
}

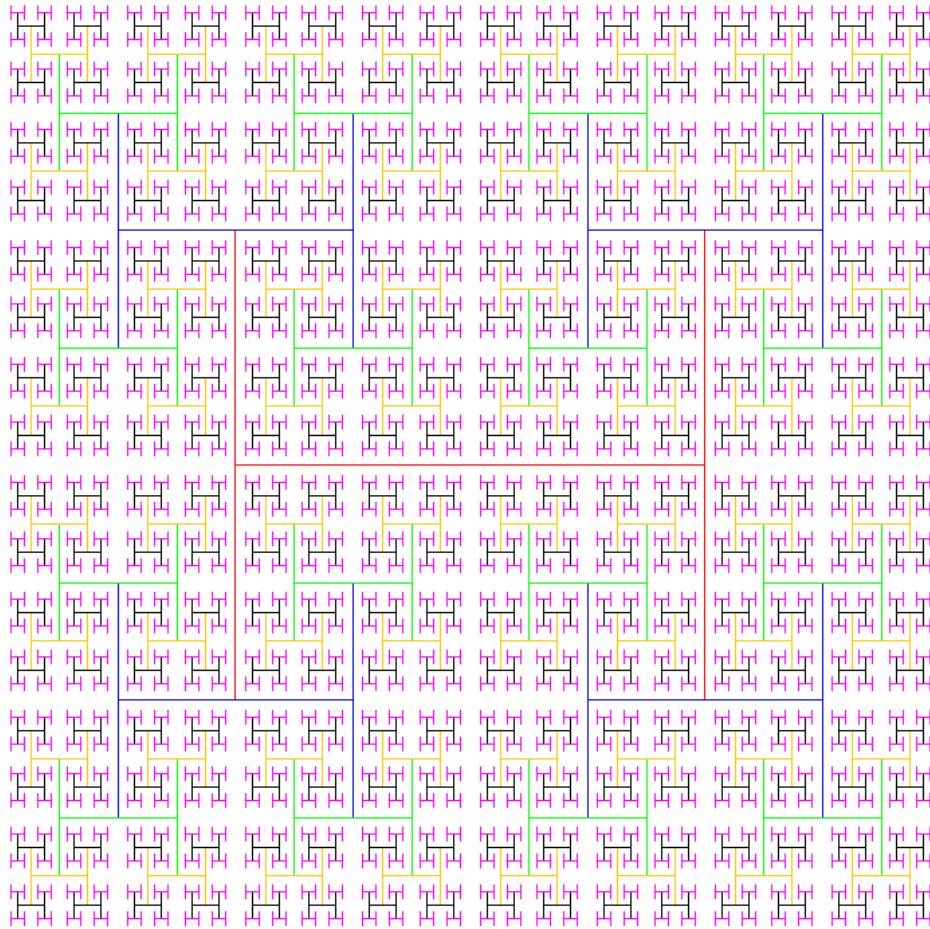
```

Задание 2

Реализовать построение заданного типа фрактала. Везде, где это необходимо, предусмотреть ввод параметров, влияющих на внешний вид фрактала: Н-фрактал.

Результат работы программы:

Fractal



Код программы:

Task2.java

```
import javax.swing.*;
import java.awt.*;
import java.util.Scanner;

public class Task2 extends JFrame { public Task2() {
setTitle("H-Fractal"); setSize(700, 700);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); setLocationRelativeTo(null);

Scanner scanner = new Scanner(System.in); System.out.print("Enter recursion depth: "); int recursionDepth =
scanner.nextInt();

Fractal fractal = new Fractal(recursionDepth); fractal.setBackground(Color.WHITE); getContentPane().add(fractal);
setVisible(true);
}

public static void main(String[] args) { SwingUtilities.invokeLater(Task2::new);
}
}
```

Fractal.java

```
import javax.swing.*;
import java.awt.*;

public class Fractal extends JPanel
{
private int recDepth;
private static Color[] colors = { Color.red, Color.blue, Color.green,
```

```
Color.orange, Color.black, Color.magenta, Color.darkGray
};

public Fractal(int recDepth)
{
    super();
    this.recDepth = recDepth;
}

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

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

    graphics.setColor(colors[depth % colors.length]); graphics.drawLine(x - size, y, x + size, y); graphics.drawLine(x - size, y
    - size, x - size, y + size); graphics.drawLine(x + size, y - size, x + size, y + size);

    drawFractal(graphics, x - size, y - size, size / 2, depth + 1); drawFractal(graphics, x + size, y - size, size / 2, depth +
    1); drawFractal(graphics, x - size, y + size, size / 2, depth + 1); drawFractal(graphics, x + size, y + size, size / 2,
    depth + 1);
}
}
```

Вывод: Освоил возможности языка программирования Java в построении графических приложений.