

**Министерство образования Республики Беларусь
Учреждение Образования
«Брестский Государственный Технический Университет»
Кафедра ИИТ**

**Лабораторная работа №7
По дисциплине СПП за 5 семестр
Тема: «Java»**

Выполнил:
Студент 3-го курса
Группы ПО-5
Крощук В.В.
Проверил:
Крощенко А.А.

Брест 2021

Лабораторная работа №7

Цель работы:

освоить возможности языка программирования Java в построении графических приложений.

Вариант: 8

Задания и выполненные решения:

1. Построение графических примитивов и надписей

Требования к выполнению

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

Задать движение по экрану строк (одна за другой) из массива строк. Направление движения по апплету и значение каждой строки выбираются случайным образом.

Код:

Main.java

```
package com.company;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.Random;

public class Main {

    public static void main(String[] args) {
        final ArrayList<String> strings = new ArrayList<String>();
        strings.add("Задать движение по экрану строк");
        strings.add("(одна за другой)");
        strings.add("из массива строк");
        strings.add("направление движения по апплету");
        strings.add("и значение каждой строки");
        strings.add("выбирается случайным образом");

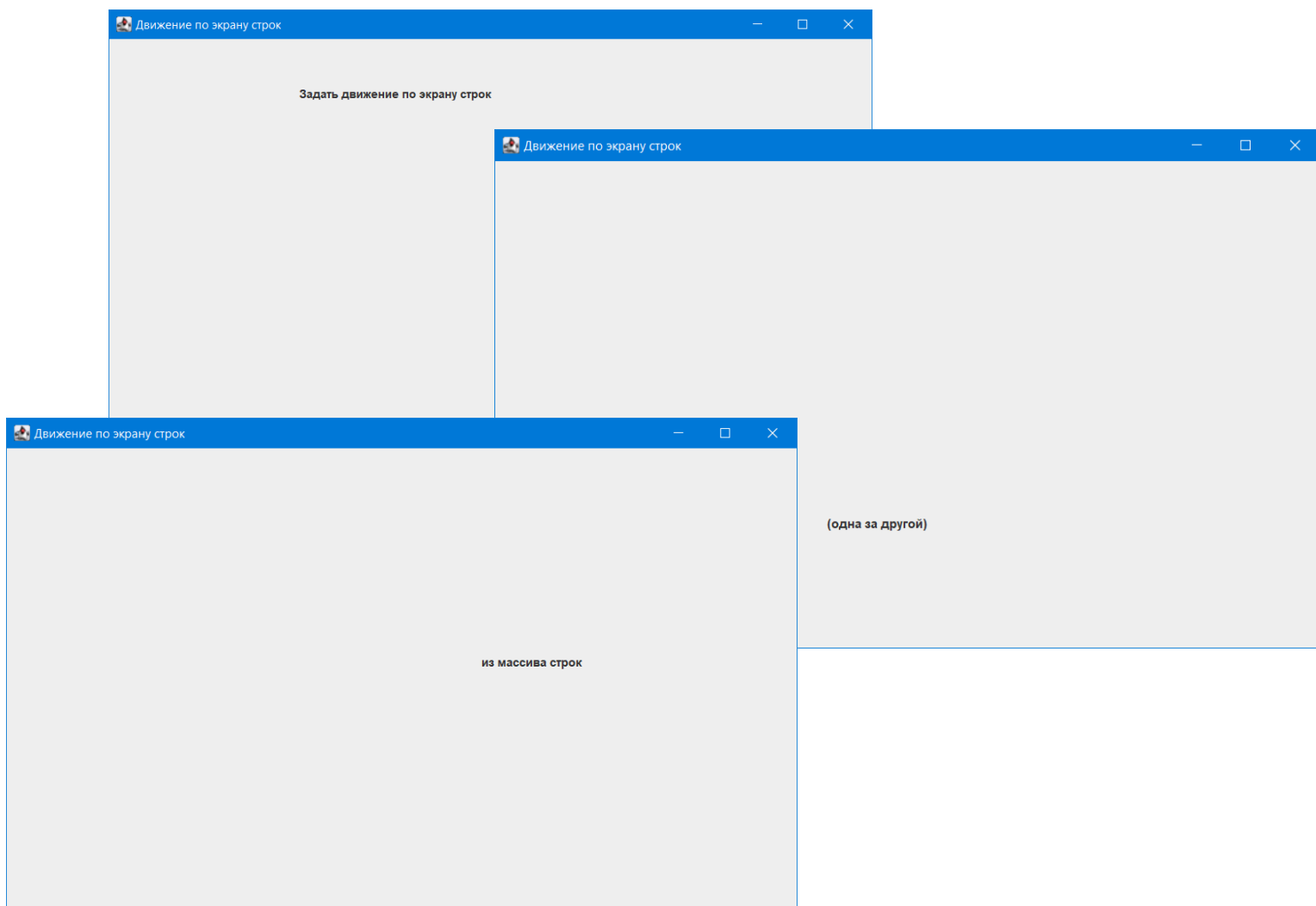
        final JFrame frame = new JFrame("Движение по экрану строк");
        frame.setPreferredSize(new Dimension(800, 500));
        frame.setVisible(true);
        frame.setLayout(null);
        final JLabel lbl = new JLabel();
        lbl.setLocation(-1, 0);
        lbl.setSize(300, 20);
```

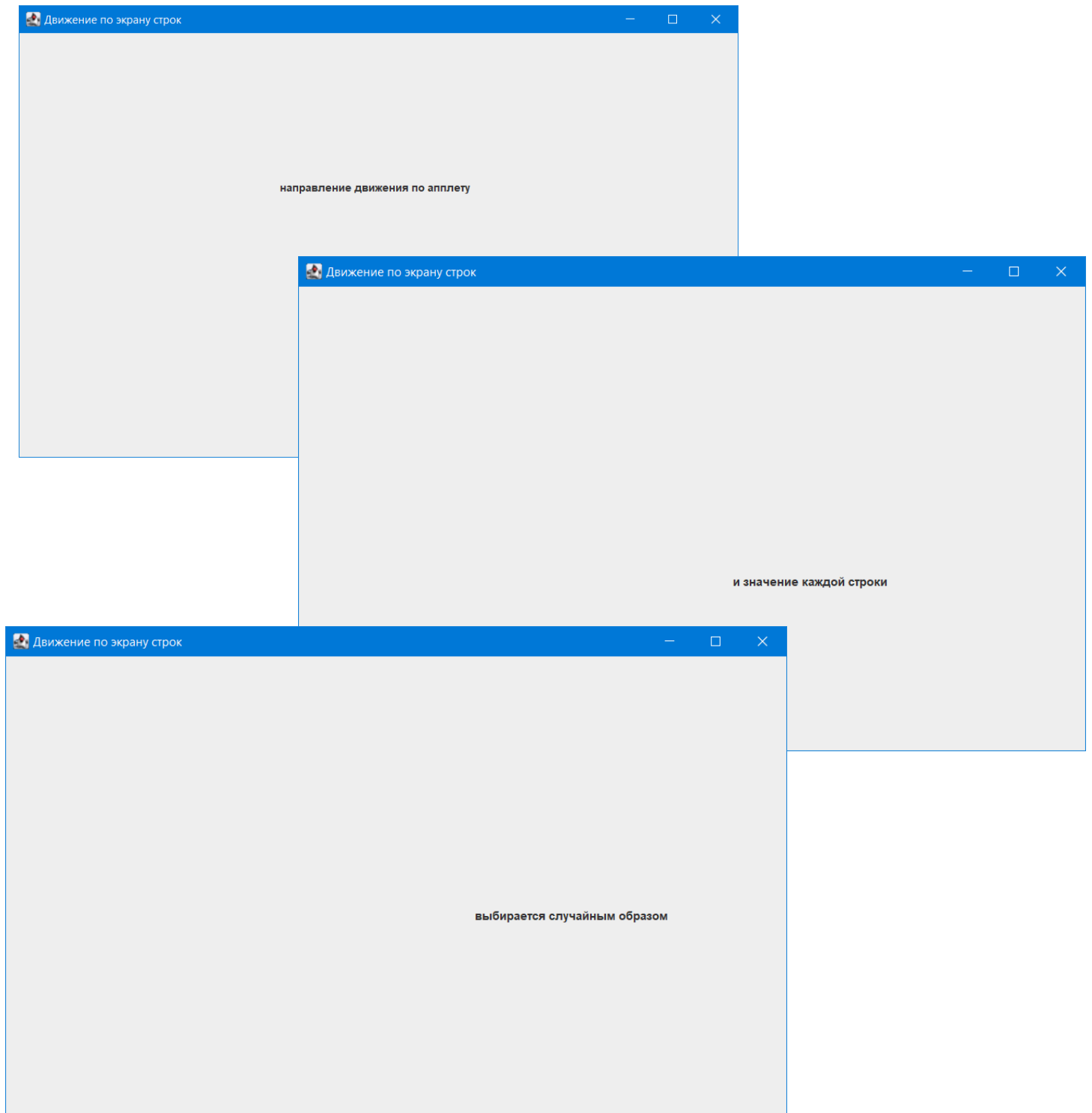
```

frame.add(lbl);
frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
frame.pack();
Timer timer = new Timer(50, new ActionListener() {
    int speedX, speedY;
    Random rnd = new Random();
    @Override
    public void actionPerformed(ActionEvent arg0) {
        Point loc = lbl.getLocation();
        if (loc.x > frame.getWidth() || loc.y > frame.getHeight() ||
loc.x < 0
        || loc.y < 0) {
            lbl.setLocation(frame.getWidth()/2, frame.getHeight()/2);
            speedX = -5+rnd.nextInt(10);
            speedY = -5+rnd.nextInt(10);
            lbl.setText(strings.get(rnd.nextInt(strings.size()-1)));
        } else {
            lbl.setLocation(loc.x + speedX, loc.y + speedY);
        }
    }
});
timer.start();
}
}

```

Результат:





Выполнение происходит согласно поставленной задачи. Работает корректно!

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

«Кривая дракона»

Код:

Main.java

```
package com.company;

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

import static javax.swing.WindowConstants.EXIT_ON_CLOSE;

public class Main {

    public static void main(String[] args) {
        JFrame window = new JFrame("Dragons");
        window.setSize(600, 600);
        for(int i = 0; i < 5; i++){
            System.out.print("Enter number of iterations: ");
            Scanner in = new Scanner(System.in);
            int n = in.nextInt();
            window.setContentPane(new DragonView(n));
            window.setResizable(false);
            window.setDefaultCloseOperation(EXIT_ON_CLOSE);
            window.setVisible(true);
        }
        System.out.println("End");
    }
}
```

DragonView.java

```
package com.company;

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

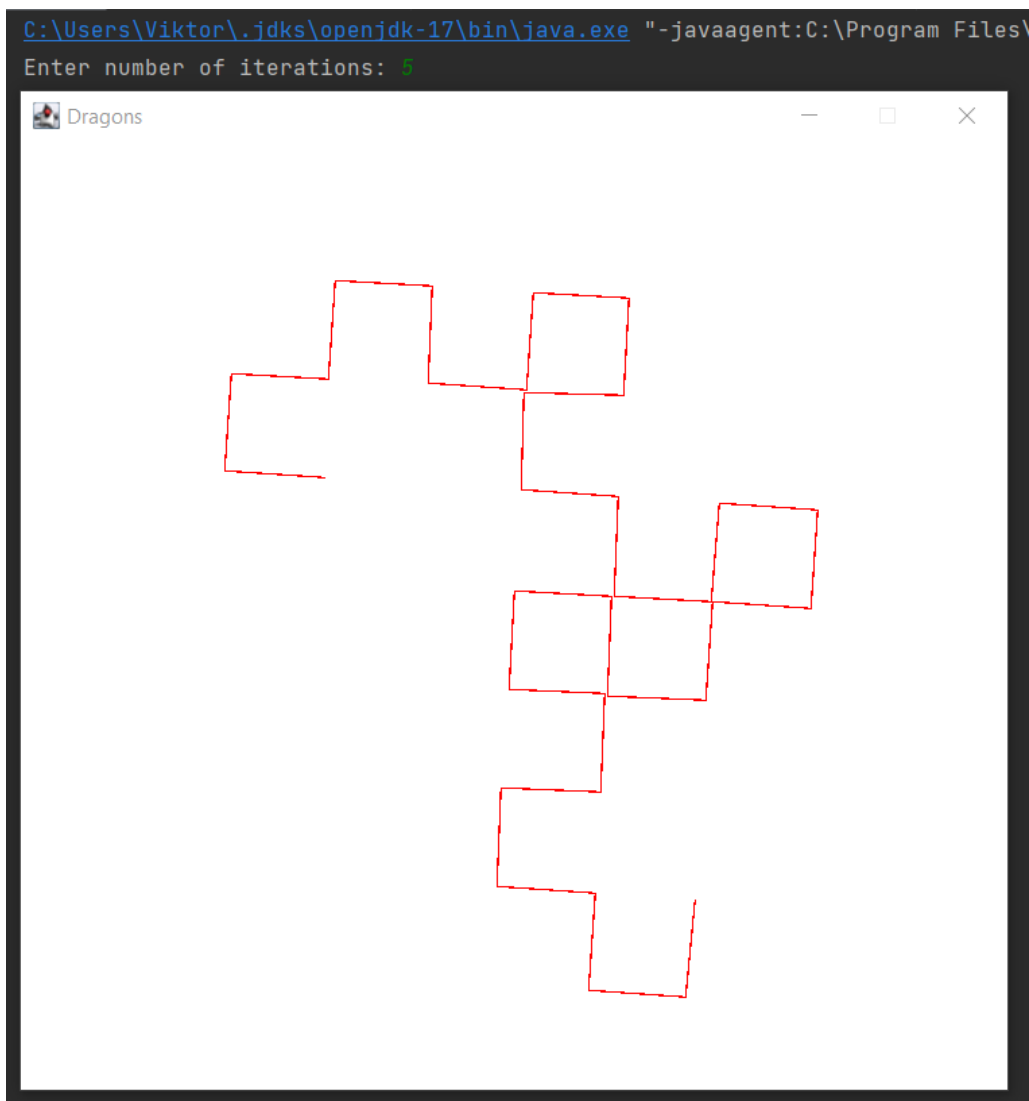
public class DragonView extends JPanel {
    private int numberOfIterations;
    public DragonView(int numberOfIterations) {
        this.numberOfIterations = numberOfIterations;
    }
    @Override
    public void paintComponent(Graphics g) {
        super.paintComponents(g);
        g.setColor(Color.RED);
    }
}
```

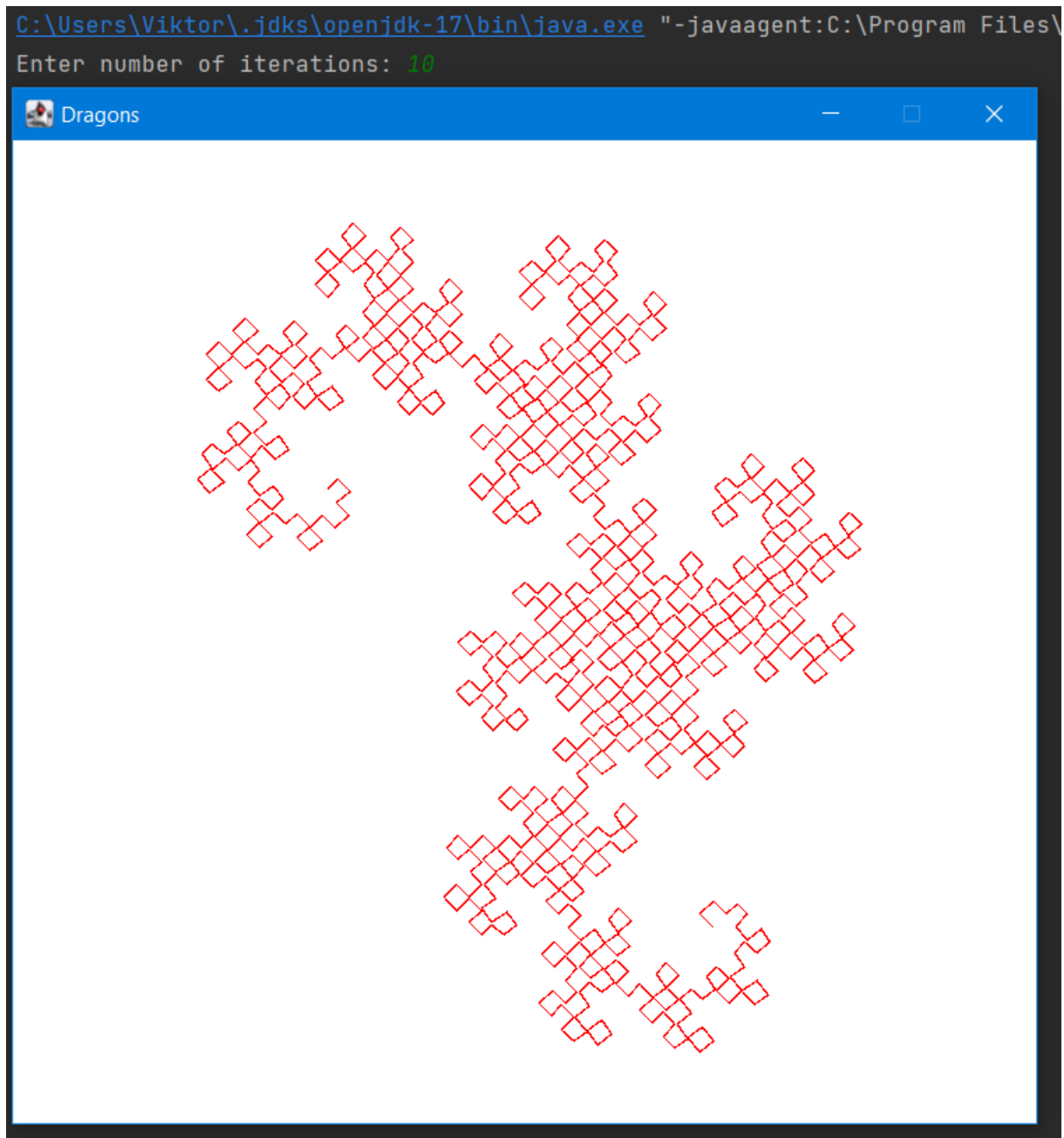
```

        drawDragon(180, 200, 400, 450, numberOfIterations, g);
        repaint();
    }
    private void drawDragon(int x1, int y1, int x2, int y2, int n, Graphics
g) {
        int xx, yy;
        if (n > 0) {
            xx = (x1 + x2) / 2 + (y2 - y1) / 2;
            yy = (y1 + y2) / 2 - (x2 - x1) / 2;
            drawDragon(x2, y2, xx, yy, n - 1, g);
            drawDragon(x1, y1, xx, yy, n - 1, g);
        }
        if (n == 0)
            g.drawLine(x1, y1, x2, y2);
    }
}

```

Результаты:





Исходя из полученных результатов, все работает корректно, согласно поставленному заданию.

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