

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”  
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

**Отчёт**  
по лабораторной работе №7

Выполнила:  
студентка группы ПО-9  
Матюшик Е.П.

Проверил:  
Крощенко А. А.

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

## Вариант 13

### Задание 1

Построение графических примитивов и надписей

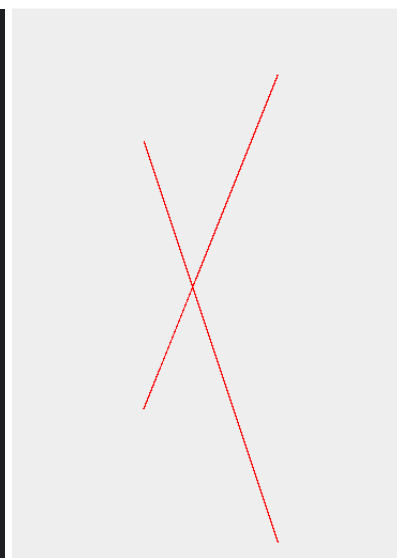
Требования к выполнению:

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

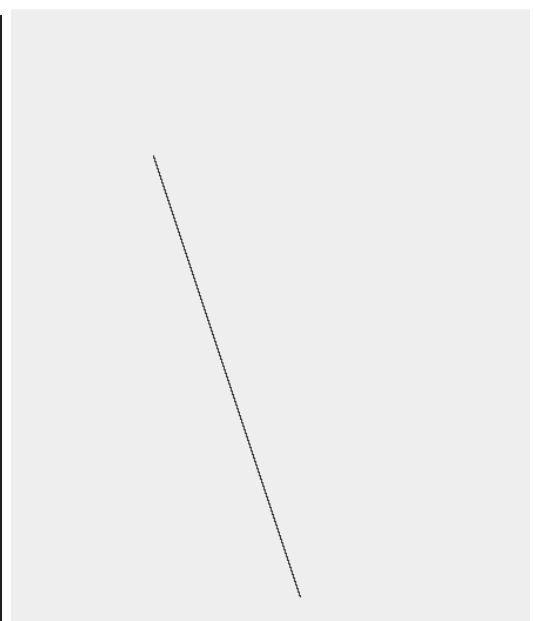
Определить класс Line для прямых линий, проходящих через точки A(x1, y1) и B(x2, y2). Создать массив объектов класса Line. Определить, используя функции, какие из прямых линий пересекаются, а какие совпадают. Нарисовать все пересекающиеся прямые.

**Результат программы:**

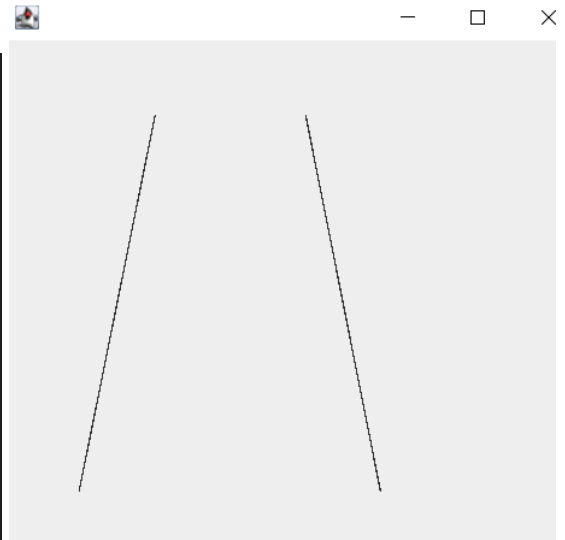
```
Введите количество линий: 2
Введите координаты точек для линии 1:
Точка A (x1): 100
Точка A (y1): 100
Точка B (x2): 200
Точка B (y2): 400
Введите координаты точек для линии 2:
Точка A (x1): 200
Точка A (y1): 50
Точка B (x2): 100
Точка B (y2): 300
Линия 1 и линия 2 пересекаются.
```



```
Введите количество линий: 2
Введите координаты точек для линии 1:
Точка A (x1): 100
Точка A (y1): 200
Точка B (x2): 100
Точка B (y2): 400
Введите координаты точек для линии 2:
Точка A (x1): 100
Точка A (y1): 200
Точка B (x2): 100
Точка B (y2): 400
Линия 1 и линия 2 совпадают.
```



```
Введите количество линий: 2
Введите координаты точек для линии 1:
Точка A (x1): 50
Точка A (y1): 50
Точка B (x2): 100
Точка B (y2): 300
Введите координаты точек для линии 2:
Точка A (x1): 250
Точка A (y1): 50
Точка B (x2): 200
Точка B (y2): 300
Линия 1 и линия 2 не пересекаются и не совпадают.
Ни одна из линий не пересекается и не совпадает.
```



### Код программы:

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

class Line {
    int x1, y1, x2, y2;

    public Line(int x1, int y1, int x2, int y2) {
        this.x1 = x1;
        this.y1 = y1;
        this.x2 = x2;
        this.y2 = y2;
    }

    public void draw(Graphics g) {
        g.drawLine(x1, y1, x2, y2);
    }

    public boolean intersects(Line other) {
        // Проверка на пересечение двух отрезков
        int s1_x, s1_y, s2_x, s2_y;
        s1_x = x2 - x1;
        s1_y = y2 - y1;
        s2_x = other.x2 - other.x1;
        s2_y = other.y2 - other.y1;

        double s, t;
        double denominator = (-s2_x * s1_y + s1_x * s2_y);

        if (denominator == 0) {
            // Линии параллельны или совпадают
            return false;
        }
    }
}
```

```

        s = (-s1_y * (x1 - other.x1) + s1_x * (y1 - other.y1)) /
denominator;
        t = (s2_x * (y1 - other.y1) - s2_y * (x1 - other.x1)) /
denominator;

        return (s >= 0 && s <= 1 && t >= 0 && t <= 1);
    }
}

class DrawingPanel extends JPanel {
    Line[] lines;

    public DrawingPanel(Line[] lines) {
        this.lines = lines;
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        for (Line line : lines) {
            line.draw(g);
        }

        // Проверяем пересечения и рисуем пересекающиеся линии
        for (int i = 0; i < lines.length; i++) {
            for (int j = i + 1; j < lines.length; j++) {
                if (lines[i].intersects(lines[j])) {
                    g.setColor(Color.RED);
                    g.drawLine(lines[i].x1, lines[i].y1, lines[i].x2,
lines[i].y2);
                    g.drawLine(lines[j].x1, lines[j].y1, lines[j].x2,
lines[j].y2);
                }
            }
        }
    }
}

public class Main1 {
    public static void main(String[] args) {
        // Получаем количество линий от пользователя
        Scanner scanner = new Scanner(System.in);
        System.out.print("Введите количество линий: ");
        int numLines = scanner.nextInt();

        // Создаем массив для хранения линий
        Line[] lines = new Line[numLines];

        // Получаем координаты точек для каждой линии
        for (int i = 0; i < numLines; i++) {

```

```

        System.out.println("Введите координаты точек для линии " +
(i+1) + " :");
        System.out.print("Точка A (x1): ");
        int x1 = scanner.nextInt();
        System.out.print("Точка A (y1): ");
        int y1 = scanner.nextInt();
        System.out.print("Точка B (x2): ");
        int x2 = scanner.nextInt();
        System.out.print("Точка B (y2): ");
        int y2 = scanner.nextInt();
        lines[i] = new Line(x1, y1, x2, y2);
    }

    // Закрываем Scanner
    scanner.close();

    // Определяем, какие линии пересекаются или совпадают
    boolean intersectFound = false;
    for (int i = 0; i < lines.length; i++) {
        for (int j = i + 1; j < lines.length; j++) {
            if (lines[i].intersects(lines[j])) {
                System.out.println("Линия " + (i+1) + " и линия "
+ (j+1) + " пересекаются.");
                intersectFound = true;
            } else if (lines[i].x1 == lines[j].x1 && lines[i].y1
== lines[j].y1 &&
                lines[i].x2 == lines[j].x2 && lines[i].y2 ==
lines[j].y2) {
                System.out.println("Линия " + (i+1) + " и линия "
+ (j+1) + " совпадают.");
                intersectFound = true;
            } else {
                System.out.println("Линия " + (i+1) + " и линия "
+ (j+1) + " не пересекаются и не совпадают.");
            }
        }
    }

    if (!intersectFound) {
        System.out.println("Ни одна из линий не пересекается и не
совпадает.");
    }

    // Создаем окно для отображения
    JFrame frame = new JFrame();
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(400, 400);

    // Добавляем панель для рисования
    DrawingPanel panel = new DrawingPanel(lines);

```

```

        frame.add(panel);

        // Отображаем окно
        frame.setVisible(true);
    }
}

```

## Задание 2

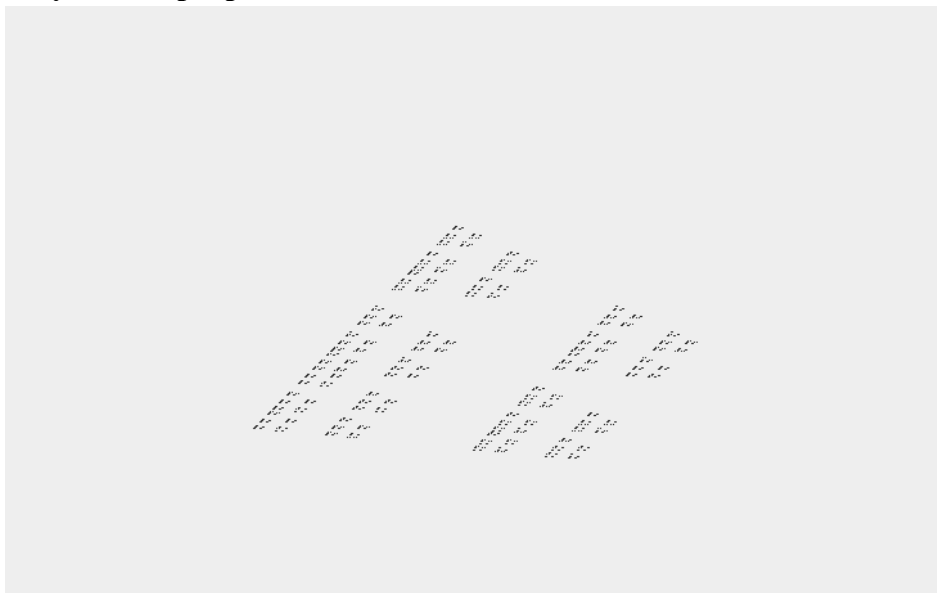
Реализовать построение заданного типа фрактала по варианту

Везде, где это необходимо, предусмотреть ввод параметров, влияющих на внешний вид фрактала.

Фрактал Леви

**Входные данные:**

**Результат программы:**



**Код программы:**

```

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

public class LeviFractal extends JPanel {
    private int level;
    private Point startPoint;

```

```

private int length;

public LeviFractal(int level, Point startPoint, int length) {
    this.level = level;
    this.startPoint = startPoint;
    this.length = length;
}

@Override
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    drawLeviFractal(g, level, startPoint, length);
}

private void drawLeviFractal(Graphics g, int level, Point
startPoint, int length) {
    if (level == 0) {
        // Рисуем отрезок
        int endX = startPoint.x + length;
        int endY = startPoint.y;
        g.drawLine(startPoint.x, startPoint.y, endX, endY);
    } else {
        // Находим координаты точки для нового отрезка
        int newX = startPoint.x + length / 4;
        int newY = startPoint.y - length / 4;

        // Рисуем новые отрезки
        drawLeviFractal(g, level - 1, startPoint, length / 3);
        drawLeviFractal(g, level - 1, new Point(newX, newY),
length / 3);
        drawLeviFractal(g, level - 1, new Point(newX + length / 3,
newY - length / 3), length / 3);
        drawLeviFractal(g, level - 1, new Point(newX + 2 * length
/ 3, newY + length / 3), length / 3);
        drawLeviFractal(g, level - 1, new Point(newX + length,
newY), length / 3);
    }
}

public static void main(String[] args) {
    // Ввод параметров
    int level =
Integer.parseInt(JOptionPane.showInputDialog("Введите уровень
фрактала:"));
    int startX =
Integer.parseInt(JOptionPane.showInputDialog("Введите начальную
координату X:"));
    int startY =
Integer.parseInt(JOptionPane.showInputDialog("Введите начальную
координату Y:"));
}

```

```

        int length =
Integer.parseInt(JOptionPane.showInputDialog("Введите длину
отрезка:"));

        // Создание окна для отображения фрактала
JFrame frame = new JFrame("Levi Fractal");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(800, 600);

        // Добавление панели для отображения фрактала
LeviFractal fractalPanel = new LeviFractal(level, new
Point(startX, startY), length);
frame.add(fractalPanel);

        // Отображение окна
frame.setVisible(true);
    }
}

```

**Вывод:** в ходе выполнения лабораторной были освоены возможности языка программирования Java в построении графических приложений.