

Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №7

По дисциплине «Современные платформы программирования»

Выполнила:

Студентка 3 курса

Группы ПО-3

Гаврилкович Е. В.

Проверил:

Крощенко А.А.

Брест 2020 г.

## Цель работы:

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

## Ход работы

### Задание 1:

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.

### Текст программы:

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

public class Main extends JPanel {
    private final static int AMT_OF_LINES = 10;
    private final static int WIDTH = 613;
    private final static int HEIGHT = 550;
    private Line[] lines;
    private Main() {
        Random random = new Random();
        lines = new Line[AMT_OF_LINES];
        for (int i = 0; i < AMT_OF_LINES; i++) {
            lines[i] = new Line(random.nextInt(WIDTH), random.nextInt(HEIGHT),
                                random.nextInt(WIDTH), random.nextInt(HEIGHT));
        }
    }
    public void paintComponent(Graphics graphics) {
        for (Line line : lines) {
            if (checkIfCross(line)) {
                graphics.drawLine(line.getX1(), line.getY1(), line.getX2(), line.getY2());
            }
        }
    }
    public static void main(String[] args) {
        JFrame frame = new JFrame("Cross Lines");
        Main panel = new Main();
        frame.add(panel);
        frame.setSize(WIDTH, HEIGHT);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

```

        frame.setVisible(true);
    }
    private boolean checkIfCross(Line line) {
        for (Line l : lines) {
            if (line.cross(l) && !line.equals(l))
                return true;
        }
        return false;
    }
}

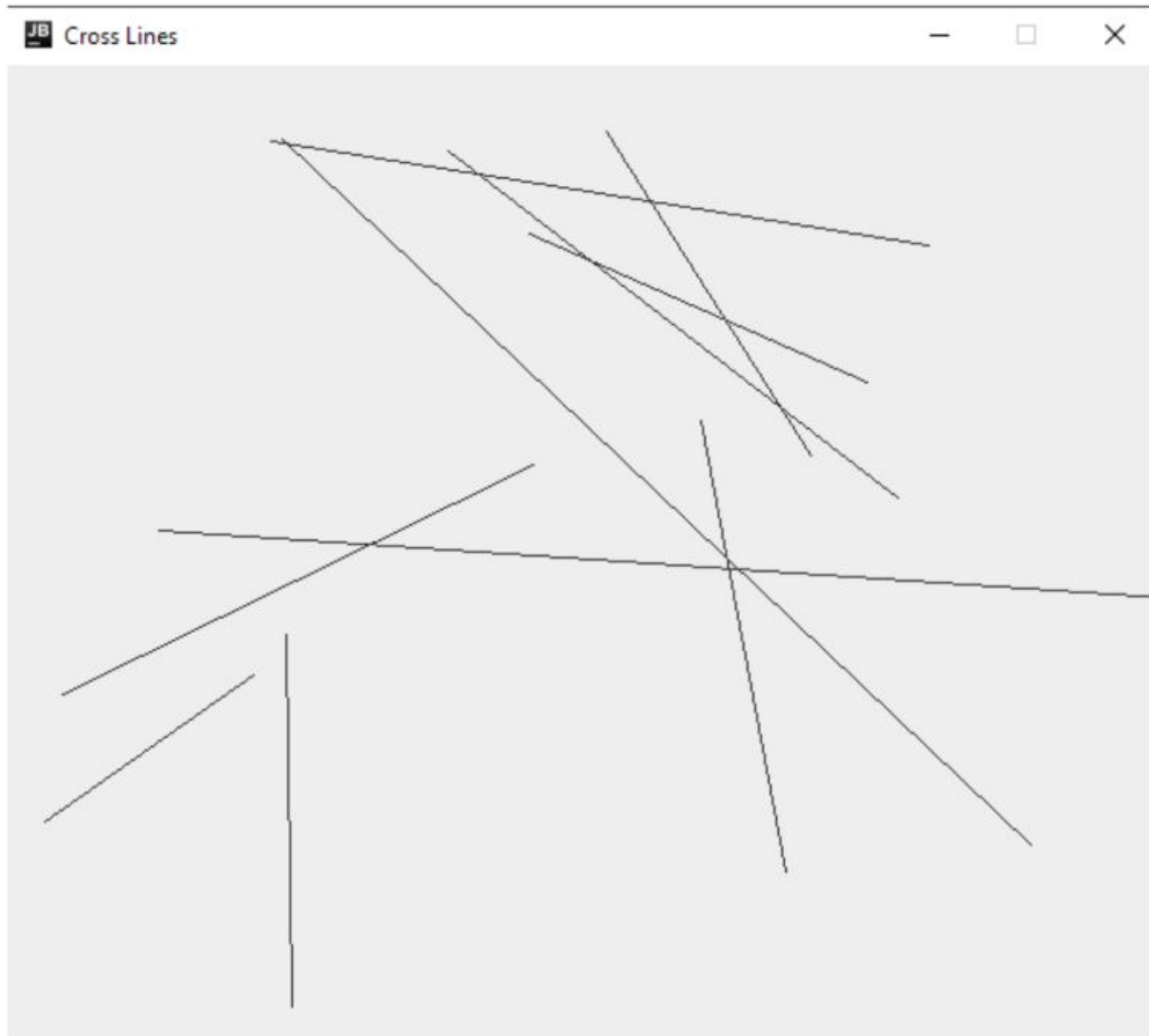
class Line {
    private int x1;
    private int y1;
    private int x2;
    private int y2;
    public Line(int x1, int y1, int x2, int y2) {
        super();
        this.x1 = x1;
        this.y1 = y1;
        this.x2 = x2;
        this.y2 = y2;
    }
    public int getX1() {
        return x1;
    }
    public int getY1() {
        return y1;
    }
    public int getX2() {
        return x2;
    }
    public int getY2() {
        return y2;
    }
    public boolean cross(Line line) {
        int common = (this.x2 - this.x1) * (line.y2 - line.y1) - (this.y2 - this.y1) * (line.x2 - line.x1);
        if (common == 0)
            return false;
        int rH = (this.y1 - line.y1) * (line.x2 - line.x1) - (this.x1 - line.x1) * (line.y2 - line.y1);
        int sH = (this.y1 - line.y1) * (this.x2 - this.x1) - (this.x1 - line.x1) * (this.y2 - this.y1);
        int r = rH / common;
        int s = sH / common;
        if (r >= 0 && r <= 1 && s >= 0 && s <= 1)
            return true;
        else
            return false;
    }
}

```

```
@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;
    Line other = (Line) obj;
    if (x1 != other.x1 && x1 != other.x2)
        return false;
    if (x2 != other.x2 && x2 != other.x1)
        return false;
    if (y1 != other.y1 && y1 != other.y2)
        return false;

    if (y2 != other.y2 && y2 != other.y1)
        return false;
    return true;
}
}
```

**Рисунок с результатом работы программы:**



## **Задание 2:**

Реализовать построение фрактала Леви.

### **Текст программы:**

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

public class Fractal extends JPanel {
    private final static int X1 = 150;
    private final static int Y1 = 350;
    private final static int X2 = 450;
    private final static int Y2 = 350;
    private final static int AMT_OF_ITERATIONS = 20;
    public void paintComponent(Graphics graphics) {
        super.paintComponents(graphics);
```

```

        graphics.setColor(Color.ORANGE);
        draw(X1, Y1, X2, Y2, AMT_OF_ITERATIONS, graphics);
    }
    private void draw(int x1, int y1, int x2, int y2, int amtOfIterations, Graphics graphics) {
        if (amtOfIterations == 0) {
            graphics.drawLine(x1, y1, x2, y2);
        } else {
            int xx = (x1 + x2) / 2 + (y2 - y1) / 2;
            int yy = (y1 + y2) / 2 - (x2 - x1) / 2;
            draw(x1, y1, xx, yy, amtOfIterations - 1, graphics);
            draw(xx, yy, x2, y2, amtOfIterations - 1, graphics);
        }
    }
}

class Main {
    public final static int WIDTH = 613;
    public final static int HEIGHT = 490;
    public static void main(String[] args) {
        JFrame frame = new JFrame("Фрактал Леви");
        frame.setSize(WIDTH, HEIGHT);
        frame.setContentPane(new Fractal());
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}

```

**Рисунок с результатом работы программы:**

