

Министерство образования Республики Беларусь  
Учреждение образования  
“Брестский государственный технический университет”  
Кафедра ИИТ

**Отчёт**  
**По лабораторной работе №7**  
**По дисциплине СПП**

**Выполнил**

Студент группы ПО-3  
3-го курса  
Куликович И. Т.

**Проверил**

Крощенко А. А.

# Лабораторная работа №7

## ВАРИАНТ 13

- Прочитать задания, взятые из каждой группы.
- Определить паттерн проектирования, который может использоваться при реализации задания.  
Пояснить свой выбор.
- Реализовать фрагмент программной системы, используя выбранный паттерн.  
Реализовать все необходимые дополнительные классы.

**Задание 1.** Проект «Бургер-закусочная». Реализовать возможность формирования заказа из определенных позиций (тип бургера (веганский, куриный и т.д.)), напиток (холодный – пепси, кока-кола и т.д.; горячий – кофе, чай и т.д.), тип упаковки – с собой, на месте. Должна формироваться итоговая стоимость заказа.

**Задание 2.** Проект «IT-компания». В проекте должен быть реализован класс «Сотрудник» с субординацией (т.е. должна быть возможность определения кому подчиняется сотрудник и кто находится в его подчинении). Для каждого сотрудника помимо сведений о субординации хранятся другие данные (ФИО, отдел, должность, зарплата). Предусмотреть возможность удаления и добавления сотрудника.

## Код программы

live.ilyusha.spp7.task1.Line

```
package live.ilyusha.spp7.task1;
```

```
public class Line {

    private 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 boolean intersects(Line line) {
        double a1 = (double) (y1 - y2) / (x1 - x2);
        double a2 = (double) (line.y1 - line.y2) / (line.x1 - line.x2);
        double b1 = y1 - a1 * x1;
        double b2 = line.y1 - a2 * line.x1;
        double xa = (b2 - b1) / (a1 - a2);
        return a1 != a2 && xa > Math.max(x1, line.x1) && xa < Math.min(x2,
line.x2);
    }

    public boolean equals(Line line) {
        return x1 == line.x1 && x2 == line.x2 && y2 == line.y2 && y1 ==
line.y1;
    }

    /* codegen */

    public int getX1() {
        return x1;
    }
}
```

```

    public void setX1(int x1) {
        this.x1 = x1;
    }

    public int getY1() {
        return y1;
    }

    public void setY1(int y1) {
        this.y1 = y1;
    }

    public int getX2() {
        return x2;
    }

    public void setX2(int x2) {
        this.x2 = x2;
    }

    public int getY2() {
        return y2;
    }

    public void setY2(int y2) {
        this.y2 = y2;
    }
}

```

## live.ilyusha.spp7.task1.LinesPanel

```

package live.ilyusha.spp7.task1;

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

class LinesPanel extends JPanel {

    ArrayList<Line> lines;

    public LinesPanel(ArrayList<Line> lines) {
        super();
        this.lines = lines;
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2D = (Graphics2D) g;
        g2D.setStroke(new BasicStroke(5));
        g2D.setColor(Color.blue);
        for (Line l: lines) {
            g.drawLine(l.getX1(), l.getY1(), l.getX2(), l.getY2());
        }
    }
}

```

## live.ilyusha.spp7.task1.Main

```
package live.ilyusha.spp7.task1;

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

public class Main {

    public static void main(String[] args) {
        ArrayList<Line> lines = new ArrayList<Line>();
        ArrayList<Line> intersectingLines = new ArrayList<Line>();

        lines.add(new Line(25, 25, 275, 275));
        lines.add(new Line(50, 100, 200, 250));
        lines.add(new Line(75, 100, 150, 25));
        lines.add(new Line(100, 25, 300, 225));
        lines.add(new Line(250, 25, 200, 110));
        lines.add(new Line(200, 225, 300, 175));

        for (Line i: lines) {
            for (Line j : lines) {
                if (i.intersects(j)) {
                    intersectingLines.add(i);
                    intersectingLines.add(j);
                }
            }
        }

        for (Line i: lines) {
            for (Line j : lines) {
                if (i.equals(j)) intersectingLines.remove(j);
            }
        }

        JFrame window = new JFrame();
        window.setSize(350, 350);

        LinesPanel linesPanel = new LinesPanel(intersectingLines);

        window.add(linesPanel);
        window.setVisible(true);
    }
}
```

## live.ilyusha.spp7.task2.LevyPanel

```
package live.ilyusha.spp7.task2;

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

public class LevyPanel extends JPanel {

    private final int iterations;
    private final Color color;

    LevyPanel(int n, Color col) {
```

```

        iterations = n;
        color = col;
    }

    public void paintComponent(Graphics g){
        super.paintComponents(g);
        drawLevy(150, 350, 450, 350, iterations, color, g);
    }

    private void drawLevy(int x1, int y1, int x2, int y2, int n, Color color,
Graphics g) {
        Graphics2D g2D = (Graphics2D) g;
        g2D.setStroke(new BasicStroke(5));
        g2D.setColor(color);

        if (n == 0) {
            g.drawLine(x1, y1, x2, y2);
            return;
        }

        int x = (x1 + x2) / 2 + (y2 - y1) / 2;
        int y = (y1 + y2) / 2 - (x2 - x1) / 2;
        drawLevy(x1, y1, x, y, n - 1, color, g);
        drawLevy(x, y, x2, y2, n - 1, color, g);
    }
}

```

## live.ilyusha.spp7.task2.Main

```

package live.ilyusha.spp7.task2;

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

public class Main {

    public static void main(String[] args) {
        System.out.println("Enter iterations: ");
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        System.out.println("Choose color: 1-red, 2-cyan, 3-pink, 4-blue");
        Color[] color = { Color.RED, Color.CYAN, Color.MAGENTA, Color.BLUE };

        JFrame window = new JFrame("Levy");
        window.setSize(615, 605);
        window.setContentPane(new LevyPanel(n, color));
        window.setResizable(false);
        window.setVisible(true);
    }
}

```

## Спецификация ввода

### Для задачи 2:

```
>java Main
```

```
Enter iterations: <количество итераций>
```

```
Choose color: [1] red, [2] cyan, [3] pink, [4] blue <номер цвета>
```

## Пример

Для задачи 2:

```
>java Main
```

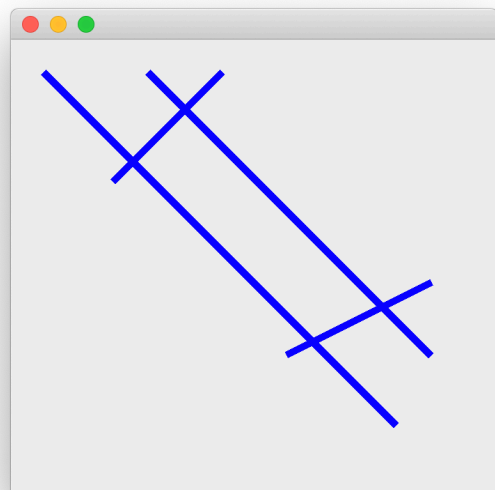
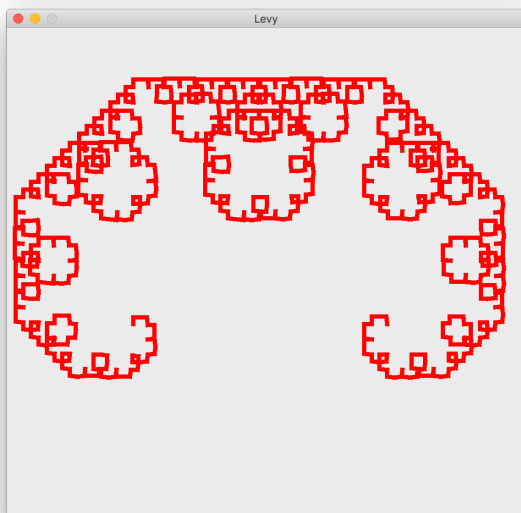
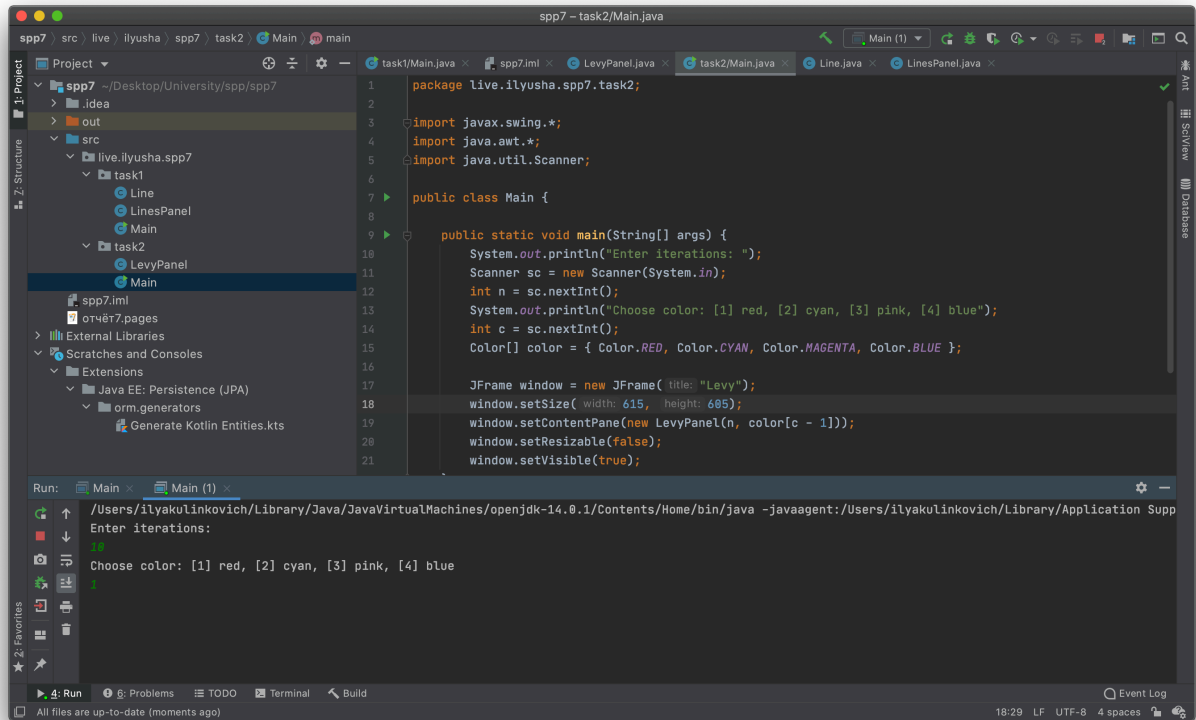
```
Enter iterations: 10
```

```
Choose color: [1] red, [2] cyan, [3] pink, [4] blue 1
```

## Спецификация вывода

<графическое приложение>

## Рисунки с результатами работы программы



## **Вывод**

В данной лабораторной работе я освоил возможности языка программирования Java в построении графических приложений.