

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

Лабораторная работа №7  
По дисциплине: «СПП»

Выполнила:  
студент 3 курса  
группы ПО-8  
Гордейчук М.В.  
Проверил:  
Крощенко А.А.

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

### Вариант 7

**Задание 1:** построение графических примитивов и надписей.

Изобразить в окне приложения (апплета) отрезок, вращающийся в плоскости фрейма вокруг точки, движущейся по отрезку.

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

Снежинка Коха.

### Ход работы:

**Задание 1:**

**Текст программы:**

```
import java.awt.geom.*;
import java.awt.*;
import java.util.*;
import javax.swing.JFrame;
public class Main {
    public static void main(String[] args) {
        javax.swing.JFrame frame = new JFrame("7.1"); //создание окна приложения
        frame.setDefaultCloseOperation(javax.swing.JFrame.EXIT_ON_CLOSE); //событие закрытия окна
        frame.setBounds(0, 0, 1000, 500);
        frame.setContentPane(new MyPanel()); //заменяет панель содержимого JFrame
        frame.setVisible(true);
    }
}
class MyPanel extends javax.swing.JPanel {
    private AffineTransform transformer;
    private AffineTransform transformerrotate;
    private class MyTimerTask extends TimerTask {
        int i = 0;
        @Override
        public void run() {
            transformer = AffineTransform.getTranslateInstance(i*10,0); //смещение на 10 по оси икс (по у -
0)
            transformerrotate = AffineTransform.getRotateInstance(i* 0.1, 150, 150); //поворачивает
координаты вокруг точки привязки на угол в i*0.1 радиан
            transformer.concatenate(transformerrotate); //объединение для выполнения комбинированного
преобразования
            i++;
            repaint();
        }
    }
}
```

@Override

```
protected void paintComponent(Graphics g) {
```

```
    super.paintComponent(g);
```

```
    Graphics2D g2 = (Graphics2D) g; //фундаментальный класс для того, чтобы представить 2-мерные  
    формы, текст и изображения
```

```
    g2.transform(transformer);
```

```
    BasicStroke pen = new BasicStroke(3);
```

```
    g2.setStroke(pen); //устанавливаем ширину пера в 3
```

```
    g2.draw(new Line2D.Float(150, 150, 50, 50)); //создает и инициализирует линию по указанным  
    координатам
```

```
}
```

```
public MyPanel() {
```

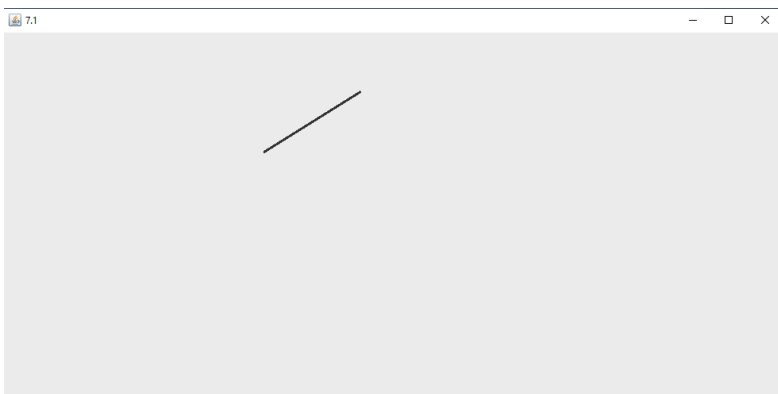
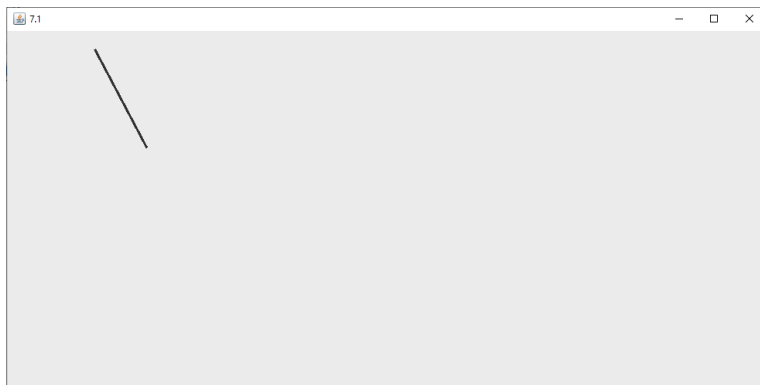
```
    Timer t = new Timer();
```

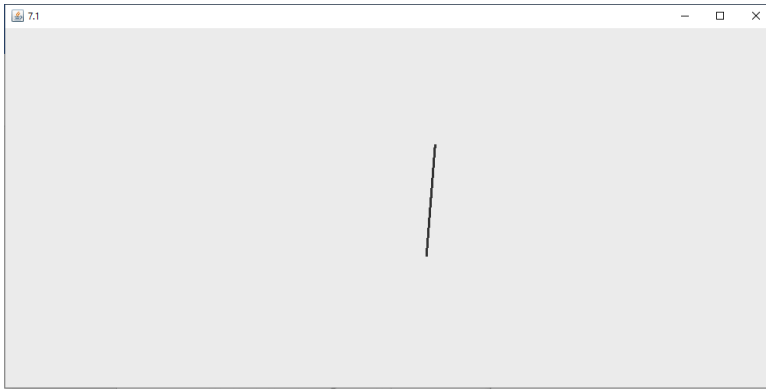
```
    t.schedule(new MyTimerTask(), 100, 100); //до начала выполнения MyTimerTask() пройдет 100 мс  
    (задержка) и он будет выполняться с периодом в 100 мс
```

```
}
```

```
}
```

**Результат:**





## Задание 2:

### Текст программы:

```
package sample;
```

```
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.awt.geom.Line2D;
import java.awt.geom.Point2D;
import javax.swing.JApplet;
import javax.swing.JFrame;

public class Main extends JApplet {
    private boolean drawn = false;
    private Graphics2D g2 = null;
    public void init() {
        setBackground(Color.black);
    }
    @Override
    public void paint(Graphics g) {
        if (drawn)
            return;
        drawn = true;
        super.paint(g);
        g2 = (Graphics2D) g;
        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
            RenderingHints.VALUE_ANTIALIAS_ON);
        g2.setPaint(Color.blue);
        double a = 300;
        double p1x = 250;
        double p1y = 400;
```

```

double p2x = p1x + a;
double p2y = p1y;
double h = Math.sqrt(Math.pow(a, 2) - Math.pow((a / 2), 2) / 4);
double pmx = (p1x + p2x) / 2;
double pmy = (p1y + p2y) / 2;
double p3x = pmx + (h * (p1y - pmy)) / (a / 2);
double p3y = pmy + (h * (p1x - pmx)) / (a / 2);
drawCurveKochRecur(new Line2D.Double(p1x, p1y, p2x, p2y), 5);
drawCurveKochRecur(new Line2D.Double(p3x, p3y, p1x, p1y), 5);
drawCurveKochRecur(new Line2D.Double(p2x, p2y, p3x, p3y), 5);
}

private void drawCurveKochRecur(Line2D line, int maxIter, int curIter) {
    if (curIter == maxIter)
        drawLine(line);
    if (curIter <= maxIter) {
        double a = line.getP1().distance(line.getP2());
        a = a / 3;
        double h = Math.sqrt(Math.pow(a, 2) - Math.pow((a / 2), 2) / 4);
        Point2D ps = line.getP1();
        Point2D pe = line.getP2();
        Point2D pm = new Point2D.Double((ps.getX() + pe.getX()) / 2,
            (ps.getY() + pe.getY()) / 2);
        Point2D p1 = new Point2D.Double((2 * ps.getX() + pe.getX()) / 3,
            (2 * ps.getY() + pe.getY()) / 3);
        Point2D p2 = new Point2D.Double((2 * pe.getX() + ps.getX()) / 3,
            (2 * pe.getY() + ps.getY()) / 3);
        Point2D p3 = new Point2D.Double(
            pm.getX() + (h * (-p2.getY() + pm.getY())) / (a / 2),
            pm.getY() + (h * (p2.getX() - pm.getX())) / (a / 2)
        );
        curIter++;
        drawCurveKochRecur(new Line2D.Double(ps, p1), maxIter, curIter);
        drawCurveKochRecur(new Line2D.Double(p1, p3), maxIter, curIter);
        drawCurveKochRecur(new Line2D.Double(p3, p2), maxIter, curIter);
        drawCurveKochRecur(new Line2D.Double(p2, pe), maxIter, curIter);
    }
}

private void drawCurveKochRecur(Line2D line, int maxIter) {
    drawCurveKochRecur(line, maxIter, 0);
}

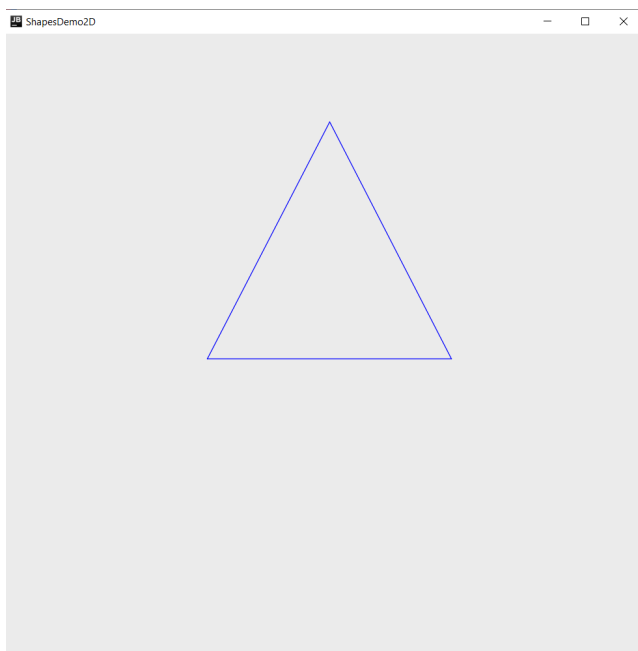
public void drawLine(Line2D line) {
    g2.draw(new Line2D.Double(line.getP1(), line.getP2()));
}

```

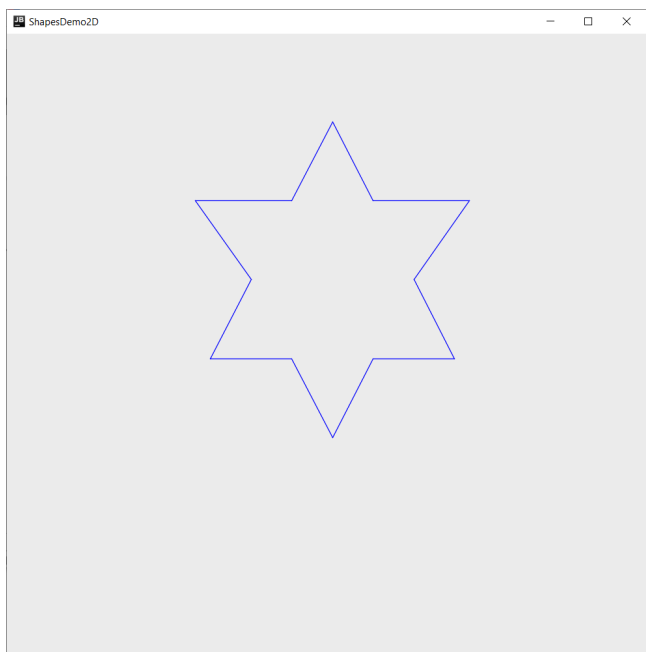
```
public static void main(String s[]) {  
    JFrame f = new JFrame("ShapesDemo2D");  
    f.addWindowListener(new WindowAdapter() {  
        public void windowClosing(WindowEvent e) {  
            System.exit(0);  
        }  
    });  
    JApplet applet = new Main();  
    f.getContentPane().add("Center", applet);  
    applet.init();  
    f.pack();  
    f.setSize(new Dimension(800, 800));  
    f.show();  
}  
}
```

### Результат:

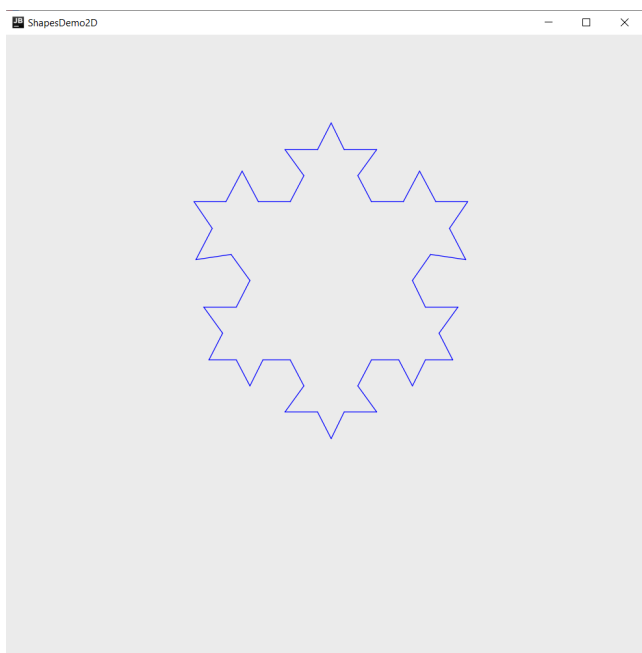
1) глубина рекурсии 0



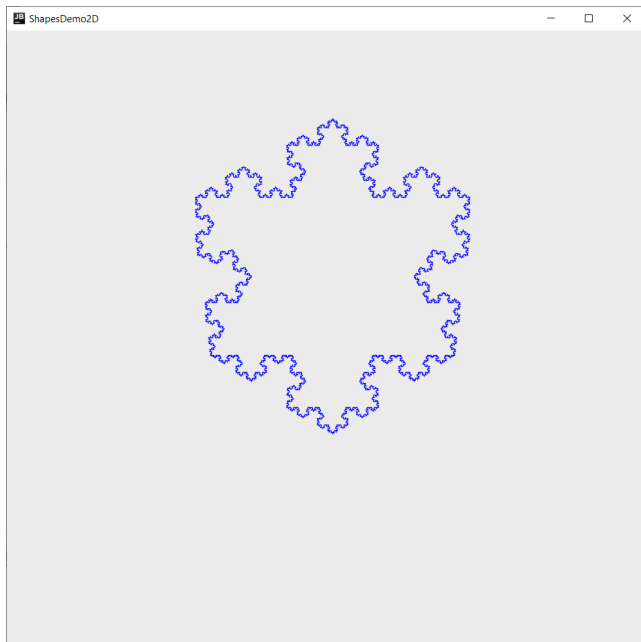
2) глубина рекурсии 1



3) глубина рекурсии 2



4) глубина рекурсии 5



**Вывод:** освоила возможности языка программирования Java в построении графических приложений.