

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

**Лабораторная работа №7**

По дисциплине: «Современные платформы программирования»

**Выполнил:**

студент 3 курса  
группы ПО-8  
Таразевич Н.А.

**Проверил:**

Крощенко А.А.

Брест, 2024

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

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

## 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.

Изобразить четырехугольник, вращающийся в плоскости апплета вокруг своего центра тяжести.

#### Код программы:

```
class Ugolnik extends JFrame {
    private int[] __x =new int[4];
    private int[] __y =new int[4];
    double angle=0;

    private final int CENTER_X;
    private final int CENTER_Y;
    private final int __windowX;
    private final int __windowY;

    public Ugolnik(int windowX, int windowY, int[] __x, int[] __y) {
        super("rect");

        setSize(windowX, windowY);
        setVisible(true);
        setResizable(false);
        this.__x=__x;
        this.__y=__y;

        __windowX = windowX;
        __windowY = windowY;
        int xA=__x[0];

        int yA=__y[0];

        int xB=__x[1];int yB=__y[1];
        int xC=__x[2];int yC=__y[2];
```

```

int xD=__x[3];int yD=__y[3];

int S1=(int) (0.5*Math.abs((xB-xA)*(yC-yA)-(xC-yA)*(yB-yA)));
int S2=(int) (0.5*Math.abs((xC-xA)*(yD-yA)-(xD-yA)*(yC-yA)));
int xm1=(xA+xB+xC)/3; int ym1=(yA+yB+yC)/3;

int xm2=(xA+xC+xD)/3;int ym2=(yA+yC+yD)/3;

int x1=(S1*xm1+S2*xm2)/(S1+S2);int y1=(S1*ym1+S2*ym2)/(S1+S2);
CENTER_X=x1;

CENTER_Y=y1;

}

public void paint(Graphics g) {
    g.fillRect(0, 0, __windowX, __windowY);
    g.setColor(Color.BLUE);

    int temp_x1,temp_y1,temp_x2,temp_y2;
    for (int i = 0; i < 3; i++) {

        temp_x1 = (int) (CENTER_X+(__x[i]-
CENTER_X)*Math.cos(angle) -
                (__y[i]-CENTER_Y)*Math.sin(angle));

        temp_x2 = (int) (CENTER_X+(__x[i+1]-
CENTER_X)*Math.cos(angle) -
                (__y[i+1]-CENTER_Y)*Math.sin(angle));

        temp_y1 = (int) (CENTER_Y+(__x[i]-
CENTER_X)*Math.sin(angle) +
                (__y[i]-CENTER_Y)*Math.cos(angle));

        temp_y2 = (int) (CENTER_Y+(__x[i+1]-
CENTER_X)*Math.sin(angle) +
                (__y[i+1]-CENTER_Y)*Math.cos(angle));

        g.drawLine(temp_x1, temp_y1, temp_x2, temp_y2);
    }

    temp_x1 = (int) (CENTER_X+(__x[3]-CENTER_X)*Math.cos(angle) -
        (__y[3]-CENTER_Y)*Math.sin(angle));

    temp_x2 = (int) (CENTER_X+(__x[0]-CENTER_X)*Math.cos(angle) -
        (__y[0]-CENTER_Y)*Math.sin(angle));

    temp_y1 = (int) (CENTER_Y+(__x[3]-CENTER_X)*Math.sin(angle) +
        (__y[3]-CENTER_Y)*Math.cos(angle));

    temp_y2 = (int) (CENTER_Y+(__x[0]-CENTER_X)*Math.sin(angle) +
        (__y[0]-CENTER_Y)*Math.cos(angle));

```

```

        g.drawLine(temp_x1, temp_y1, temp_x2, temp_y2);
        this.angle = this.angle + 0.01;
    }

    public void run(){
        while(true){

            repaint();

            try{
                Thread.sleep(5);
            }

            catch(InterruptedException ignored){
            }

            setDefaultCloseOperation(EXIT_ON_CLOSE);
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter width : ");
        int windowX = sc.nextInt();
    }

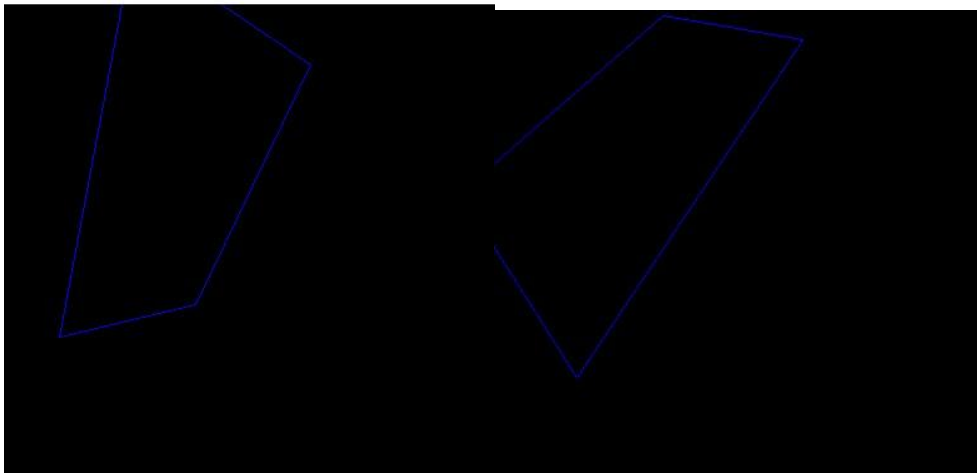
```

### Результат работы программы:

```

Enter width : 1000
Enter height : 600
Enter point(x y) : 120 50
Enter point(x y) : 231 34
Enter point(x y) : 145 342
Enter point(x y) : 12 234

```



## Задание 2

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

Треугольная салфетка Серпинского.

### Код программы:

```
class Triangles extends JPanel {
    private int __n;
    private int __depth;

    Triangles(int n, int depth) {
        __n = n;
        __depth = depth;
    }

    public void setColor(Graphics g){
        Random rand = new Random();
        int randNum = rand.nextInt(5);
        if(randNum == 0){
            g.setColor(Color.RED);
        }
        else if(randNum == 1){
            g.setColor(Color.MAGENTA);
        }
        else if(randNum == 2){
            g.setColor(Color.GREEN);
        }
        else if(randNum == 3){
            g.setColor(Color.BLUE);
        }
        else if(randNum == 4){
            g.setColor(Color.YELLOW);
        }
    }

    public void paintComponent(Graphics g){
        super.paintComponents(g);
        drawTri(600, 350, __n, g, 0);
    }

    private void drawTri(int x, int y, int length, Graphics g, int
cur_depth) {
```

```

        setColor(g);

        int xA = (int) Math.round(x- 0.5*length);
        int xB= x;

        int xC= (int) Math.round(x+ 0.5*length);

        int yA =(int)Math.round(y - Math.sqrt(0.75*length*length/3));
        int yB= (int)Math.round(y + Math.sqrt(0.75*length*length/3));
        int yC= (int)Math.round(y - Math.sqrt(0.75*length*length/3));

        g.drawLine(xA, yA, xB, yB);
        g.drawLine(xB, yB, xC, yC);
        g.drawLine(xC, yC, xA, yA);

        if(cur_depth < __depth) {

            drawTri((int) Math.round((x+xA)/2),
(int)Math.round((y+yA)/2),length/2, g,cur_depth+1);

            drawTri((int) Math.round((x+xB)/2),
(int)Math.round((y+yB)/2),length/2, g,cur_depth+1);

            drawTri((int) Math.round((x+xC)/2),
(int)Math.round((y+yC)/2),length/2, g,cur_depth+1);

        }
    }
}

```

```

class Lab2 extends JFrame {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter length");
        int n = sc.nextInt();

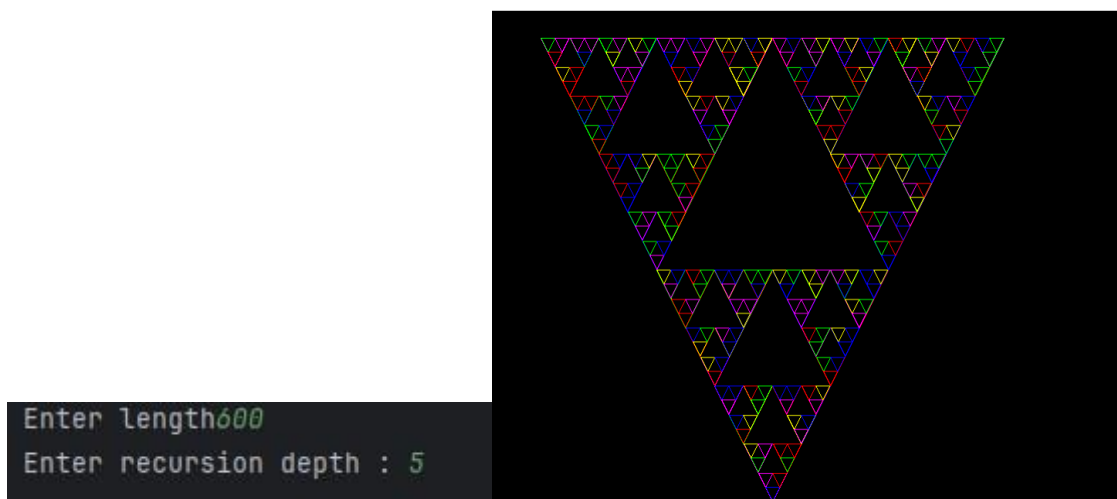
        System.out.print("Enter recursion depth : ");
        int depth = sc.nextInt();

        JFrame window = new JFrame("serpinsky");
        window.setSize(1520, 700);
        window.setBackground(Color.BLACK);
        window.setContentPane(new Triangles(n, depth));
        window.setResizable(false);
        window.setDefaultCloseOperation(EXIT_ON_CLOSE);
        window.setVisible(true);

    }
}

```

**Результат работы программы:**



Вывод: я изучил основы работы с графическими примитивами.