

```
import java.awt.*;
import java.awt.event.*;
import java.awt.image.*;
import javax.swing.*;
import javax.swing.border.*;
import java.util.*;

/**
 * FractalTriangle - 1030 GUI Demonstration.
 * (The Sierpinski Triangle)
 *
 * @author William Soukoreff
 */
public class FractalTriangle extends JFrame
{
    public static void main(String[] args)
    {
        FractalTriangle jframe = new FractalTriangle();
        jframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        jframe.setTitle("Fractal");
        jframe.pack();
        jframe.setVisible(true);
    }

    /*
     * some constants
     */
    static final int PANEL_WIDTH = 500;
    static final int PANEL_HEIGHT = 500;

    // this is the JPanel where we'll draw the fractal
    DrawPanel drawpanel;

    /*
     * the constructor for our JFrame object
     */
    public FractalTriangle()
    {
        /*
         * construct and configure GUI components
         */
        drawpanel = new DrawPanel();
        drawpanel.setBackground(Color.WHITE);
        drawpanel.setPreferredSize(new Dimension(PANEL_WIDTH, PANEL_HEIGHT));
        drawpanel.setMaximumSize(new Dimension(PANEL_WIDTH, PANEL_HEIGHT));

        // make paint panel this JFrame's content pane
        setContentPane(drawpanel);
    }

    /*
     * This inner class extends JPanel
     *
     * this is where we do our drawing...
     */
    class DrawPanel extends JPanel
    {
```

```

// constructor
public DrawPanel()
{
    super();
}

void drawTriangle(Graphics g,
    double x1, double y1,
    double x2, double y2,
    double x3, double y3, double size)
{
    if(size < 0.5)
        return;

    g.drawLine((int)x1, (int)y1, (int)x2, (int)y2);
    g.drawLine((int)x2, (int)y2, (int)x3, (int)y3);
    g.drawLine((int)x3, (int)y3, (int)x1, (int)y1);

    drawTriangle(g,
        x1 + size/4.0, y1 - (y3-y1)/2.0,
        x2 - size/4.0, y2 - (y3-y1)/2.0,
        x3, y1, (x2-x1)/2.0);
    drawTriangle(g,
        x1 - size/4.0, y1 + (y3-y1)/2.0,
        x1 + size/4.0, y2 + (y3-y1)/2.0,
        x1, y3, (x2-x1)/2.0);
    drawTriangle(g,
        x2 - size/4.0, y1 + (y3-y1)/2.0,
        x2 + size/4.0, y2 + (y3-y1)/2.0,
        x2, y3, (x2-x1)/2.0);
}

// do our drawing...
public void paintComponent(Graphics g)
{
    super.paintComponent(g);

    final double width = 450;
    final double height = 450;

    g.setColor(Color.BLACK);
    drawTriangle(g,
        width/4.0 + 25.0, height/2.0 + 10.0,
        width*0.75 + 25.0, height/2.0 + 10.0,
        width/2.0 + 25.0, height + 10.0, width/2.0);
}
}
}

```