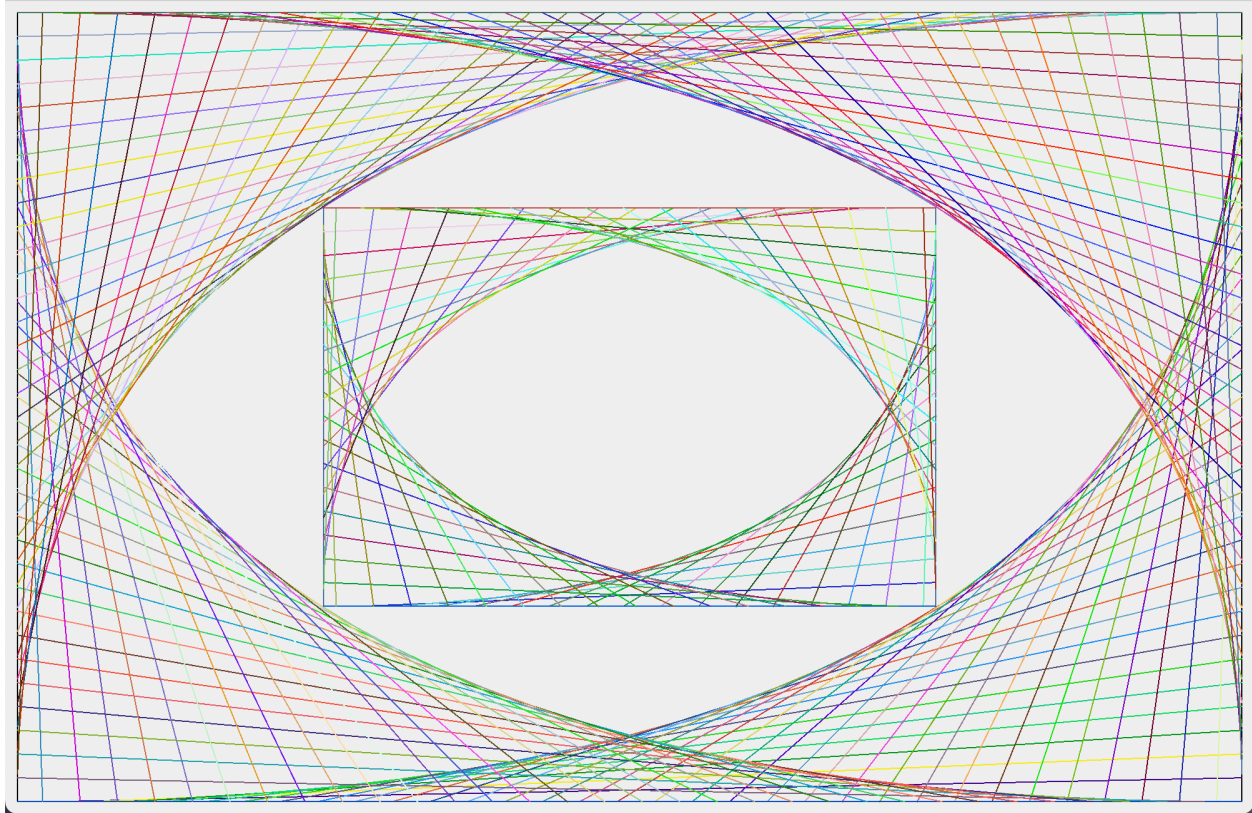




```

import java.awt.Dimension;
import java.awt.Graphics;
import javax.swing.JPanel;
import javax.swing.JFrame;
import java.util.Random;
import java.awt.Color;
public class LineArt extends JPanel {
    // Unique version ID for this class to ensure saved objects can be loaded safely
    private static final long serialVersionUID = 1L;
    // Initial width of height of the starting rectangle
    private static int width = 980;
    private static int height = 630;
    // main method to launch the program as a standalone application - no need to
    // modify
    public static void main(String[] args) {
        LineArt panel = new LineArt();
        panel.setPreferredSize(new Dimension(width + 20, height + 20)); //
content size window dimensions
        JFrame frame = new JFrame("Line Art"); // Title of frame
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

```

```

        frame.add(panel);
        frame.pack();
        frame.setVisible(true);
    }
    /**
     * Draw the four corners of line art. Line art displays straight lines inside a
     * rectangle from one side to a perpendicular side. The lines must be drawn in
     * such a way that both the starting points of the lines on one side and the
     * ending points on the other side are equi-distant along the sides. The size of
     * the rectangle is 980 pixels wide by 630 pixels high.
     *
     * @param g the Graphics object used for drawing shapes, text, and images
     */
    public void drawLineArt(Graphics g) {
        // Draw the initial rectangle
        g.drawRect(10, 10, width, height);
        // Draw bottom-left corner
        // int y=640;
        // for(int x=100;x<990;x+=20) {
        //     y=(int)(y-(20*(.65*x)));
        //     g.drawLine(x, 640, 990, y);
        // }
        Random random = new Random();
        int y = 640;
        for (int x = 10; x < 990; x += 30) {
            int red = random.nextInt(256);
            int green = random.nextInt(256);
            int blue = random.nextInt(256);
            Color randomColor = new Color(red, green, blue);
            g.setColor(randomColor);
            g.drawLine(x, 640, 990, y);
            y -= 30*65/100;
        }
        // Draw bottom-right corner
        int yy = 640;
        for (int xx = 990; xx > 10; xx -= 30) {
            int red = random.nextInt(256);
            int green = random.nextInt(256);
            int blue = random.nextInt(256);
            Color randomColor = new Color(red, green, blue);

```

```

        g.setColor(randomColor);
        g.drawLine(10, yy, xx, 640);
        yy -= 30*65/100;
    }
    // Draw top-right corner
    int yyy = 10;
    for (int xxx = 990; xxx > 10; xxx -= 30) {
        int red = random.nextInt(256);
        int green = random.nextInt(256);
        int blue = random.nextInt(256);
        Color randomColor = new Color(red, green, blue);
        g.setColor(randomColor);
        g.drawLine(10, yyy, xxx, 10);
        yyy = yyy + 30*65/100;
    }
    // Draw top-left corner
    int yyyy = 10;
    for (int xxxx = 10; xxxx < 990; xxxx += 30) {
        int red = random.nextInt(256);
        int green = random.nextInt(256);
        int blue = random.nextInt(256);
        Color randomColor = new Color(red, green, blue);
        g.setColor(randomColor);
        g.drawLine(xxxx, 10, 990, yyyy);
        yyyy = yyyy + 30*65/100;
    }
    // g.drawRect(240, 161, 520, 329);
    int j = 490;
    g.drawRect(500 - j / 2, 325 - j / 2 * 65 / 100, j, j * 65 / 100);
    int yx = 484;
    for (int xy = 255; xy <= 745; xy += 30) {
        int red = random.nextInt(256);
        int green = random.nextInt(256);
        int blue = random.nextInt(256);
        Color randomColor = new Color(red, green, blue);
        g.setColor(randomColor);
        g.drawLine(xy, 484, 745, yx);
        yx = yx - 30*65/100;
    }
    int yyx = 484;

```

```

    for (int xxy = 745; xxy >= 255; xxy -= 30) {
        int red = random.nextInt(256);
        int green = random.nextInt(256);
        int blue = random.nextInt(256);
        Color randomColor = new Color(red, green, blue);
        g.setColor(randomColor);
        g.drawLine(255, yyx, xxy, 484);
        yyx -= 30*65/100;
    }
    int yyyx = 166;
    for (int xxxy = 745; xxxy >= 255; xxxy -= 30) {
        int red = random.nextInt(256);
        int green = random.nextInt(256);
        int blue = random.nextInt(256);
        Color randomColor = new Color(red, green, blue);
        g.setColor(randomColor);
        g.drawLine(255, yyyx, xxxy, 166);
        yyyx = yyyx + 30*65/100;
    }
    int yyyyx = 166;
    for (int xxxxy = 255; xxxxy <= 745; xxxxy += 30) {
        int red = random.nextInt(256);
        int green = random.nextInt(256);
        int blue = random.nextInt(256);
        Color randomColor = new Color(red, green, blue);
        g.setColor(randomColor);
        g.drawLine(xxxxy, 166, 745, yyyyx);
        yyyyx = yyyyx + 30*65/100;
    }
}

/**
 * Overrides JPanel's paintComponent method to perform custom drawing.
 *
 * @param g the Graphics object used for drawing shapes, text, and images
 */
@Override
protected void paintComponent(Graphics g) {
    super.paintComponent(g); // Clears the panel before drawing
    drawLineArt(g);
}

```

} }