```
import java.awt.Dimension;
import java.awt.Graphics;
import javax.swing.JPanel;
import javax.swing.JFrame;
import java.awt.Color; // Required for setting line 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;

      private static int width = 980;
      private static int height = 630;

    private static final int numSteps = 50;
    private static final int XStart = 10;
    private static final int YStart = 10;

    private static final int innerWidth = 490;
    private static final int innerHeight = 315;
    private static final int innerXStart = XStart + (width - innerWidth) / 2;
    private static final int innerYStart = YStart + (height - innerHeight) / 2;


      // 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");
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.add(panel);
            frame.pack();
            frame.setVisible(true);
      }
    public void drawLineArt(Graphics g, int startX, int startY, int rectWidth, int
rectHeight) {

      //Calculates spacing
        final double WIDTH_STEP = (double)rectWidth / numSteps;
        final double HEIGHT_STEP = (double)rectHeight / numSteps;

        // Coordinates of the rectangle sides
        int xLeft = startX;
        int xRight = startX + rectWidth;
        int yTop = startY;
        int yBottom = startY + rectHeight;

        // Draw the bounding rectangle
        g.drawRect(startX, startY, rectWidth, rectHeight);

        for (int i = 0; i <= numSteps; i++) {
            // Calculated distances for the standard pattern (distance i from
corner)
            int dx = (int)(i * WIDTH_STEP);
            int dy = (int)(i * HEIGHT_STEP);
```

```
            // Calculated distances for the reflected pattern (distance N-i from
corner)
            int i_opposite = numSteps - i;
            int dx_ref = (int)(i_opposite * WIDTH_STEP);
            int dy_ref = (int)(i_opposite * HEIGHT_STEP);


            g.drawLine(xRight, yBottom - dy_ref, xRight - dx, yBottom);
            g.drawLine(xLeft, yBottom - dy_ref, xLeft + dx, yBottom);
            g.drawLine(xRight, yTop + dy_ref, xRight - dx, yTop);
            g.drawLine(xLeft, yTop + dy_ref, xLeft + dx, yTop);
        }
    }

      public void drawLineArt(Graphics g) {
        g.setColor(Color.BLACK);
        drawLineArt(g, XStart, YStart, width, height);
        drawLineArt(g, innerXStart, innerYStart, innerWidth, innerHeight);
      }

      /**
       * 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);
      }

}
```