

// Squares.java: This program draws 50 squares inside each other.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Squares {

    private Frame mainFrame;
    private Label headerLabel;
    private Label subheaderLabel;
    private JPanel controlPanel;
    private JButton enterButton;
    private CvSquares cvs;

    private double n = 8, k = 10, q = 0.2;

    public Squares() {
        prepareGUI();
    }

    public static void main(String[] args) {
        Squares squares = new Squares();
        squares.showContent();
    }

    private void prepareGUI() {
        mainFrame = new Frame("Squares: 50 squares inside each other");
        mainFrame.setSize(800, 600);
        mainFrame.setLayout(new FlowLayout());
        mainFrame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent windowEvent) {
                System.exit(0);
            }
        });
        headerLabel = new Label();
        headerLabel.setAlignment(Label.LEFT);
        subheaderLabel = new Label();
        subheaderLabel.setAlignment(Label.LEFT);

        controlPanel = new JPanel();
        controlPanel.setLayout(new FlowLayout());

        mainFrame.add(headerLabel);
        mainFrame.add(subheaderLabel);
        mainFrame.add(controlPanel);
        mainFrame.setVisible(true);
    }

    private void showContent() {
        headerLabel.setText(
            "Input an n number of normal squares, k number of inside squares, and q value to"
            "divide the squares.");
        subheaderLabel.setText("Click 'Enter' to see results. Default setting is: n = 8, k = 10, q"
            "= 0.2");
        Label nLabel = new Label("n: ", Label.RIGHT);
        Label kLabel = new Label("k: ", Label.RIGHT);
        Label qLabel = new Label("q: ", Label.RIGHT);
        final TextField nText = new TextField(5);
        final TextField kText = new TextField(5);
        final TextField qText = new TextField(5);
        nText.setText("8");
        kText.setText("10");
        qText.setText("0.2");

        enterButton = new JButton("Enter");
        cvs = new CvSquares(n, k, q);

        enterButton.addActionListener(new ActionListener() {
```

```

        public void actionPerformed(ActionEvent e) {
            if (nText.getText().isEmpty()) {
                n = 8;
                nText.setText("8");
                nText.getEchoChar();
            }

            if (kText.getText().isEmpty()) {
                k = 10;
                kText.setText("10");
                nText.getEchoChar();
            }

            if (qText.getText().isEmpty()) {
                q = 0.2;
                qText.setText("0.2");
                nText.getEchoChar();
            }

            n = Double.parseDouble(nText.getText());
            k = Double.parseDouble(kText.getText());
            q = Double.parseDouble(qText.getText());

            cvs.setn(n);
            cvs.setk(k);
            cvs.setq(q);
            controlPanel.add(cvs);
        }
    });

    controlPanel.add(nLabel);
    controlPanel.add(nText);
    controlPanel.add(kLabel);
    controlPanel.add(kText);
    controlPanel.add(qLabel);
    controlPanel.add(qText);
    controlPanel.add(enterButton);
    controlPanel.add(cvs);
    mainFrame.setVisible(true);
}
}

```

```

class CvSquares extends Canvas {
    private static final long serialVersionUID = 1L;

    double n, k, q;

    public void setn(double n) {
        this.n = n;
    }

    public void setk(double k) {
        this.k = k;
    }

    public void setq(double q) {
        this.q = q;
    }

    public CvSquares(double n, double k, double q) {
        setBackground(Color.LIGHT_GRAY);
        setSize(401, 401);

        setn(n);
        setk(k);
        setq(q);
    }

    int maxX, maxY, incX, incY;
}

```

```

void initgr() {
    Dimension d = getSize();
    maxX = d.width - 1;
    maxY = d.height - 1;
    incX = (int) (maxX / n);
    incY = (int) (maxY / n);
}

public void paint(Graphics g) {
    initgr();

    double xA, yA, xB, yB, xC, yC, xD, yD;
    double xA1, yA1, xB1, yB1, xC1, yC1, xD1, yD1;
    double p = 1 - q;

    for (int row = 0; row < maxX; row += incX) {
        for (int col = 0; col < maxY; col += incY) {
            xA = row;
            xB = row + incX;
            xC = row + incX;
            xD = row;
            yA = col;
            yB = col;
            yC = col + incY;
            yD = col + incY;
            for (int i = 0; i < k; i++) {
                g.drawLine((int) xA, (int) yA, (int) xB, (int) yB);
                g.drawLine((int) xB, (int) yB, (int) xC, (int) yC);
                g.drawLine((int) xC, (int) yC, (int) xD, (int) yD);
                g.drawLine((int) xD, (int) yD, (int) xA, (int) yA);
                if (((row / incX) % 2) == ((col / incY) % 2)) {
                    xA1 = p * xA + q * xB;
                    yA1 = p * yA + q * yB;
                    xB1 = p * xB + q * xC;
                    yB1 = p * yB + q * yC;
                    xC1 = p * xC + q * xD;
                    yC1 = p * yC + q * yD;
                    xD1 = p * xD + q * xA;
                    yD1 = p * yD + q * yA;
                    xA = xA1;
                    xB = xB1;
                    xC = xC1;
                    xD = xD1;
                    yA = yA1;
                    yB = yB1;
                    yC = yC1;
                    yD = yD1;
                } else {
                    xA1 = q * xA + p * xB;
                    yA1 = q * yA + p * yB;
                    xB1 = q * xB + p * xC;
                    yB1 = q * yB + p * yC;
                    xC1 = q * xC + p * xD;
                    yC1 = q * yC + p * yD;
                    xD1 = q * xD + p * xA;
                    yD1 = q * yD + p * yA;
                    xA = xA1;
                    xB = xB1;
                    xC = xC1;
                    xD = xD1;
                    yA = yA1;
                    yB = yB1;
                    yC = yC1;
                    yD = yD1;
                }
            }
        }
    }
}

```