

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

```
public class App extends JPanel implements ActionListener {
```

```
    private int xc, yc, r;
```

```
    public App() {
        setBackground(Color.WHITE);
        setPreferredSize(new Dimension(300, 300));
    }
```

```
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
```

```

        g.setColor(Color.RED);
        drawCircle(g, xc, yc, r);
    }
```

```
    private void drawCircle(Graphics g, int xc, int yc, int r) {
        int x = 0, y = r;
        int d = 3 - 2 * r;
```

while (y >= x) {
 // **Düzeltilme:** `drawCirclePoints()` yöntemi, her bir x ve y değeri için sekiz pikseli çizer. Bu, dairenin keskin ve pürüzsüz görünmesini sağlar.

```
            drawCirclePoints(g, xc, yc, x, y);
```

```
            x++;
```

// **Düzeltilme:** `d` değeri, dairenin çevresini çizerken hangi y değerinin kullanılacağını belirler. Bu değeri hesaplamak için aşağıdaki formül kullanılır:

```
            //
            // d = 3 - 2 * (x + y)
```

```
            d = 3 - 2 * (x + y);
```

// **Düzeltilme:** `d` değeri pozitif ise, y değeri bir azaltılır. Bu, dairenin çevresini çizerken, dairesel bir eğri oluşturmak için y değerini azaltır.

```
            if (d > 0) {
                y--;
```

```
            }
```

```
        }
```

```
    }
```

```
    private void drawCirclePoints(Graphics g, int xc, int yc, int x, int y) {
```

// **Düzeltilme:** `drawCirclePoints()` yöntemi, her bir x ve y değeri için sekiz pikseli çizer. Bu, daireni n keskin ve pürüzsüz görünmesini sağlar.

```
        g.fillRect(xc + x, yc + y, 1, 1);
        g.fillRect(xc - x, yc + y, 1, 1);
        g.fillRect(xc + x, yc - y, 1, 1);
        g.fillRect(xc - x, yc - y, 1, 1);
        g.fillRect(xc + y, yc + x, 1, 1);
        g.fillRect(xc - y, yc + x, 1, 1);
```

```
        g.fillRect(xc + y, yc - x, 1, 1);
        g.fillRect(xc - y, yc - x, 1, 1);
    }

    public void actionPerformed(ActionEvent e) {
        xc = Integer.parseInt(JOptionPane.showInputDialog("Enter x-coordinate of the center:"));
        yc = Integer.parseInt(JOptionPane.showInputDialog("Enter y-coordinate of the center:"));
        r = Integer.parseInt(JOptionPane.showInputDialog("Enter radius:"));

        repaint();
    }

    public static void main(String[] args) {
        JFrame frame = new JFrame("Circle Drawing");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().add(new App());
        frame.pack();
        frame.setVisible(true);
    }
}
```