```java
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import javax.swing.*;

class DrawPanel extends JPanel {

    private static final long serialVersionUID = 78856138825254339 87L;
    private int failures;
    private boolean win;

    public DrawPanel() {
        super();
        this.failures = 0;
        this.win = false;
        this.setBackground(new Color(210));
        this.setPreferredSize(new Dimension(310, 220));
    }
```

```java
@Override
protected void paintComponent( Graphics g ) {
    super.paintComponent(g);

    // draw head
    int thickness = 6;
    int lineOffset = -(thickness / 2);
    int startPosX = 120;
    int startPosY = 50;
    int radius = 25;
    int diameter = radius * 2;
    for (int i = 0; i <= thickness; i++) {
        switch (this.failures) {
        case 10: // Right leg
            g.drawLine(startPosX + radius, startPosY + diameter + 2
                    * radius - i, startPosX + radius + (diameter * 2 / 3),
                    startPosY + 2 * diameter + radius - i);
            g.drawString("Verloren", 175, 150);
        case 9: // Left leg
            g.drawLine(startPosX + radius, startPosY + diameter + 2
                    * radius - i, startPosX + radius - (diameter * 2 / 3),
                    startPosY + 2 * diameter + radius - i);
        case 8: // Right hand
            g.drawLine(startPosX + radius, startPosY + diameter + radius
                    + lineOffset + i, startPosX + radius
                    + (diameter * 2 / 3), startPosY + diameter + lineOffset
                    + i);
        case 7: // Left hand
            g.drawLine(startPosX + radius, startPosY + diameter + radius
                    + lineOffset + i, startPosX + radius
                    - (diameter * 2 / 3), startPosY + diameter + lineOffset
                    + i);
```

```
            case 6: // Back
                g.drawLine(startPosX + radius + lineOffset + i, startPosY
                        + diameter, startPosX + radius + lineOffset + i,
                        startPosY + 4 * radius);
            case 5: // Head
                g.drawOval(startPosX + i, startPosY + i, diameter - 2 * i,
                        diameter - 2 * i);
            case 4:
                g.drawLine(startPosX + radius, startPosY / 3, startPosX
                        + radius, startPosY / 3 + radius + i);
            case 3:
                g.drawLine(startPosX / 3, startPosY / 3 + diameter + i,
                        startPosX / 3 + diameter + i, startPosY / 3);
            case 2:
                g.drawLine(startPosX / 3, startPosY / 3 + i,
                        startPosX + radius, startPosY / 3 + i);
            case 1:
                g.drawLine(startPosX / 3 + i, startPosY / 3, startPosX / 3 + i,
                        startPosY / 3 + 8 * radius);
                break;
            default:
            }
        }

        if (this.win) {
            g.drawString("Gewonnen", 175, 150);
        }
    }
```

```
    public void draw( int numFailures, boolean win2 ) {
        if (this.failures != numFailures || this.win != win2) {
            this.failures = numFailures;
            this.win = win2;
            repaint();
        }
    }

    public int getFailures() {
        return failures;
    }

}
```