

```

import java.awt.Rectangle;
import java.awt.Shape;
import java.awt.geom.GeneralPath;
import java.awt.geom.AffineTransform;

/**
 * Ein Stuhl, der manipuliert werden kann und sich selbst auf
 * einer Leinwand zeichnet.
 * Er erbt von der Klasse Moebelstueck.
 *
 * @author Hauke Morisse
 * nach einer Vorlage von Uwe Debacher,
 * Michael Kölling und David J. Barnes und Axel Schmoltzky
 * @version 10.11.2012
 */
public class Stuhl extends Moebelstueck
{

    /**
     * Erzeuge einen neuen Stuhl mit einer Standardfarbe an einer
     * Standardposition.
     */
    public Stuhl()
    {
        xPosition = 160;
        yPosition = 80;
        farbe = "blau";
        orientierung = 0;
        istSichtbar = false;
        breite = 40;
        tiefe = 40;
    }

    /**
     * Berechnet das zu zeichnende Shape anhand der gegebenen Daten
     * [ Zum Anzeigen der Attributwerte über Inspect muss hier die
     * Sichtbarkeit
     * auf public gesetzt werden. ]
     */
    private Shape gibAktuelleFigur()
    {
        GeneralPath stuhl = new GeneralPath();
        stuhl.moveTo(xPosition, yPosition);
        stuhl.lineTo(xPosition+breite, yPosition);
        stuhl.lineTo(xPosition+breite+3, yPosition+tiefe);
        stuhl.lineTo(xPosition-3, yPosition+tiefe);
        stuhl.lineTo(xPosition, yPosition);
        //Das ist die Umrandung. Das Stuhl bekommt noch eine Lehne
        stuhl.moveTo(xPosition, yPosition + 5 );
    }
}

```

```

        stuhl.lineTo(xPosition + breite        , yPosition + 5);

        AffineTransform t1 = new AffineTransform();

t1.rotate(Math.toRadians(orientierung),gibMitteX(),gibMitteY());

        return  t1.createTransformedShape(stuhl);
    }

    /**
     * Hole die X-Koordinate des Mittelpunktes
     * [ Hilfsfunktion für das Drehen. ]
     */
    private int gibMitteX()
    {
        return xPosition+breite/2;
    }

    /**
     * Hole die Y-Koordinate des Mittelpunktes
     * [ Hilfsfunktion für das Drehen. ]
     */
    private int gibMitteY()
    {
        return yPosition+tiefe/2;
    }

    /**
     * Mache dieses Objekt sichtbar. Wenn es bereits sichtbar ist,
tue
     * nichts.
     */
    public void zeige()
    {
        istSichtbar = true;
        zeichne();
    }

    /**
     * Mache dieses Objekt unsichtbar. Wenn es bereits unsichtbar
ist, tue
     * nichts.
     */
    public void verberge()
    {
        loesche();
        istSichtbar = false;
    }

    /**
     * Drehe auf den angegebenen Winkel
     */

```

```

public void dreheAuf(int neuerWinkel)
{
    loesche();
    orientierung = neuerWinkel;
    zeichne();
}

/**
 * Bewege dieses Objekt horizontal um 'entfernung'
Bildschirmpunkte.
 */
public void bewegeHorizontal(int entfernung)
{
    loesche();
    xPosition += entfernung;
    zeichne();
}

/**
 * Bewege dieses objekt vertikal um 'entfernung'
Bildschirmpunkte.
 */
public void bewegeVertikal(int entfernung)
{
    loesche();
    yPosition += entfernung;
    zeichne();
}

/**
 * Ändere die Farbe dieses Objektes in 'neueFarbe'.
 * Gültige Angaben sind "rot", "gelb", "blau", "gruen",
 * "lila" und "schwarz".
 */
public void aendereFarbe(String neueFarbe)
{
    farbe = neueFarbe;
    zeichne();
}

/**
 * Zeichne dieses Objekt mit seinen aktuellen Werten auf den
Bildschirm.
 */
private void zeichne()
{
    if (istSichtbar)
    {
        Shape figur = gibAktuelleFigur();
        Leinwand leinwand = Leinwand.gibLeinwand();
        leinwand.zeichne (

```

```

        this,
        farbe,
        figur);
    leinwand.warte(10);
}
}

/*
 * Lösche dieses Objekt vom Bildschirm.
 */
private void loesche()
{
    if (istSichtbar)
    {
        Leinwand leinwand = Leinwand.gibLeinwand();
        leinwand.entferne(this);
    }
}
}

```