

```

import java.awt.Rectangle;
import java.awt.Shape;
import java.awt.geom.GeneralPath;
import java.awt.geom.AffineTransform;
import java.awt.geom.Ellipse2D;
import java.awt.Graphics2D;

/**
 * Die Klasse Moebelstueck generalisiert die bisherigen Klassen. Im zweiten
 * Schritt sind
 * die Attribute von "Stuhl" und "Tisch" übertragen, die beiden gemeinsam sind,
 * sowie die
 * Hilfsfunktionen für das Drehen: Die Methoden "gibMitteX()" und "gibMitteY()".
 *
 * @author Hauke Morisse
 * @version 10.11.2012
 */
public class Moebelstueck
{
    protected int xPosition;
    protected int yPosition;
    protected int orientierung;
    protected String farbe;
    protected boolean istSichtbar;
    protected int breite;
    protected int tiefe;

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

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

    /**
     * Berechnet das zu zeichnende Shape anhand der gegebenen Daten
     * [ Zum Anzeigen der Attributwerte über Inspect muss hier die Sichtbarkeit
     * auf public gesetzt werden. ]
     */
    protected Shape gibAktuelleFigur()
    {
        Shape tisch = new Ellipse2D.Double(xPosition , yPosition, breite, tiefe);

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

        return t1.createTransformedShape(tisch);
    }
}

```

```

/**
 * 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();
}

/**
 * Aendere die Farbe dieses Objektes in 'neueFarbe'.
 * Gueltige 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.
 */
protected void zeichne()
{
    if (istSichtbar)
    {
        Shape figur = gibAktuelleFigur();
        Leinwand leinwand = Leinwand.gibLeinwand();
        leinwand.zeichne (
            this,
            farbe,
            figur);
        leinwand.warte(10);
    }
}

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

```