```
//@author Dörte Haftendorn
//Anregung von  www.jjam.de - Tetraeder - Version 18.11.2003
//nun Oktader Haftendorn Lüneburg 29.4.50

import java.awt.*;
import java.applet.*;

public class oktaeder extends Applet {
    // 8 Eckpunkte 1-8
    // mit je 3 Koordinaten 1,2,3
    double p[][] = new double[7][7];
    int x=1, y=2, z=3;
    public void init() {
        setBackground(new Color(255,255,100));

        // 8 Eckpunkte im lokalen Oktaeder-Koordinatensystem
        // Nullpunkt = Mittelpunkt
        p[1][x] = 150; p[1][y] = 0;    p[1][z] =0;
        p[2][x] =    0; p[2][y] =150;    p[2][z] = 0;
        p[3][x] = -150; p[3][y] = 0;    p[3][z] = 0;
        p[4][x] =    0; p[4][y] =-150; p[4][z] =   0;
        p[5][x] =    0; p[5][y] = 0; p[5][z] =150;
        p[6][x] =    0; p[6][y] = 0; p[6][z] =-150;

        //         5
        //       / | \
        //      /  |  \
        //     // 4| \  \
        //    1    |   3
        //      `  2  ´
        //         |
        //         6
    }

    // Rotationswinkel in rad
    double angle_x =0.01;
    double angle_y = 0.007;
    double angle_z = 0.001;

    Image buffer;
    Graphics2D gBuffer;

    public void paint(Graphics g) {

        // Double-Buffering
        if (buffer==null) {
            buffer=createImage(this.getSize().width, this.getSize().height);
            gBuffer=(Graphics2D)buffer.getGraphics();
        }
        gBuffer.clearRect(0,0,this.getSize().width, this.getSize().height);

        // Antialiasing
        gBuffer.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);

        gBuffer.setColor(new Color(70,0,0));

        // Lokale Oktaeder-Koordinaten
        // in Welt-Koordinaten: +200
        gBuffer.drawLine((int)(p[1][x])+150,(int)(p[1][y])+150,(int)(p[2][x])+150,(int)
(p[2][y])+150);
        gBuffer.drawLine((int)(p[2][x])+150,(int)(p[2][y])+150,(int)(p[3][x])+150,(int)
(p[3][y])+150);
        gBuffer.drawLine((int)(p[3][x])+150,(int)(p[3][y])+150,(int)(p[4][x])+150,(int)
(p[4][y])+150);
```

```java
        gBuffer.drawLine((int)(p[4][x])+150,(int)(p[4][y])+150,(int)(p[1][x])+150,(int)
(p[1][y])+150);
        gBuffer.drawLine((int)(p[1][x])+150,(int)(p[1][y])+150,(int)(p[5][x])+150,(int)
(p[5][y])+150);
        gBuffer.drawLine((int)(p[2][x])+150,(int)(p[2][y])+150,(int)(p[5][x])+150,(int)
(p[5][y])+150);
        gBuffer.drawLine((int)(p[3][x])+150,(int)(p[3][y])+150,(int)(p[5][x])+150,(int)
(p[5][y])+150);
        gBuffer.drawLine((int)(p[4][x])+150,(int)(p[4][y])+150,(int)(p[5][x])+150,(int)
(p[5][y])+150);

        gBuffer.drawLine((int)(p[1][x])+150,(int)(p[1][y])+150,(int)(p[6][x])+150,(int)
(p[6][y])+150);
        gBuffer.drawLine((int)(p[2][x])+150,(int)(p[2][y])+150,(int)(p[6][x])+150,(int)
(p[6][y])+150);
        gBuffer.drawLine((int)(p[3][x])+150,(int)(p[3][y])+150,(int)(p[6][x])+150,(int)
(p[6][y])+150);
        gBuffer.drawLine((int)(p[4][x])+150,(int)(p[4][y])+150,(int)(p[6][x])+150,(int)
(p[6][y])+150);
        g.drawImage (buffer, 0, 0, this);

        // Verzögerung
        try {Thread.sleep(10);}
        catch (InterruptedException e) {}

        double px, py, pz;

        for (int i=1;i<7;i++) {

            px = p[i][x];
            py = p[i][y];
            pz = p[i][z];

            // Rotation um x-Achse
            p[i][y] = py*Math.cos(angle_x)-pz*Math.sin(angle_x);
            p[i][z] = py*Math.sin(angle_x)+pz*Math.cos(angle_x);

            py = p[i][y];
            pz = p[i][z];

            // Rotation um y-Achse
            p[i][x] = px*Math.cos(angle_y)+pz*Math.sin(angle_y);
            p[i][z] =-px*Math.sin(angle_y)+pz*Math.cos(angle_y);

            px = p[i][x];

            // Rotation um z-Achse
            p[i][x] = px*Math.cos(angle_z)-py*Math.sin(angle_z);
            p[i][y] = py*Math.cos(angle_z)+px*Math.sin(angle_z);
        }

        repaint();
    }

    public void update(Graphics g) {paint(g);}
}
```