```
// www.jjam.de - Lindenmayer System - Version 21.03.2003
//Ergänzt durch Haftendorn :
//Anzeige der Stufen, ausgelöst durch Javascript 1.05.05

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

public class Lindenmayer extends Applet {

    Point a, b;               // Verbindungspunkte eines Einzelschritts

    int lengthF = 3*3*3*3*3*3*2;   // Schrittlänge max 6 Stufen Ha
    int stufe=6;              //  Stufe  max 6         Ha
    int j;                    //Zähler                Ha
    double direction;         // Richtung in Grad
    double rotation = 40;   // Drehung in Grad

    Graphics g;
    Graphics2D g2;

    public void init() {
        setBackground(new Color(255,255,200));   //Ha
    }

    public void paint(Graphics g) {

        g2 = (Graphics2D) g; // Anti-Aliasing
        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
        g2.setColor(new Color(100,200,80)); // Farbe  Ha
        a = new Point(200,500); // Start-Punkt    Ha
        direction = -90;        // Start-Richtung in Grad    Ha
        lengthF = 3*3*3*3*3*3*2;                          //Ha
        for (j=1;j<stufe+1;j++){      // Schrittlänge    Ha
            lengthF= lengthF/3;
        }

        turtleGraphic(g2, "F", stufe);             ///Ha
       //for (j=0;j<7,j++){

        //turtleGraphic(g2, "F", j);  // Axiom und Iterationstiefe
        //repaint(50L);

    }

    public void setStufe(int stufeNr)                          //Ha eingefügt
    /** soll im JavaSrcipt verwendet werden **/
        {
            stufe=stufeNr;
            repaint(50L);  //L fuer Long, 50 Millisek
        }
    public void turtleGraphic(Graphics g2, String instruction, int depth) {

        if (depth==0) return;
        depth -= 1;

        Point aMark = new Point(0,0);
        double directionMark = 0;
        // Dummy-Werte

        int i;
        char c;

        for (i=0;i<instruction.length();i++) {
```

```
            c = instruction.charAt(i);

            // Schritt Vorwärts
            if (c=='F') {

                // Produktionsregel iterieren, solange Tiefe nicht erreicht ist
                turtleGraphic(g2, "F[+F]F[-F]F", depth);

                // Zeichnen: Ab 'a' in Richtung 'direction' einen Schritt der Länge
'lengthF'
                if (depth==0) {
                    double rad = 2*Math.PI/360 * direction; // Grad -> Radiant

                    int p = (int) (lengthF * Math.cos(rad));
                    int q = (int) (lengthF * Math.sin(rad));

                    b = new Point(a.x+p, a.y+q);

                    g2.drawLine(a.x, a.y, b.x ,b.y);

                    a = b; // Neuer Startpunkt
                }
            }

            // Drehung links herum
            else if (c=='+') direction += rotation;

            // Drehung rechts herum
            else if (c=='-') direction -= rotation;

            // Position und Richtung speichern
            else if (c=='[') {
                aMark = a;
                directionMark = direction;
            }

            // Zurück zu gespeicherter Position und Richtung
            else if (c==']') {
                a = aMark;
                direction = directionMark;
            }
        }
    }
}
```