```
/**
 * Die Kochkurve
 * Erstellungdatum 12.03.1999
 * Überarbeitet am 19.10.2007
 * Rolf Krüger
 */

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

class Interpret {
    private String sCode;
    private Graphics g;
    private double dir = 0.0;
    private double len = 0.0;
    private double xStart = 30.0;
    private double yStart = 200.0;

    public Interpret(String s, Graphics graph) {
        sCode = new String(s);
        g = graph;
    }

    public void reFormat(double l, int xS, int yS)
    {
        len = l;
        xStart = xS;
        yStart = yS;
    }

    public void show() {
        for(int i = 0; i<sCode.length(); i++)
        {
            char z = sCode.charAt(i);
            switch(z) {
                case &apos;s&apos;:
                    // Linie der Länge len in Richtung dir zeichnen
                    double dx = len * Math.cos(dir);
                    double dy = len * Math.sin(dir);
                    g.drawLine((int)xStart, (int)yStart, (int)(xStart+dx), (int)(yStart-dy));
                    xStart += dx;
                    yStart -= dy;
                    break;
                case &apos;L&apos;:
                    // Richtung um 120 Grad nach links drehen
                    dir += 2.0 / 3.0 * Math.PI;
                    // g.drawString(String.valueOf(dir), 100, 100);
                    break;
                case &apos;l&apos;:
                    // Richtung um 60 Grad nach links drehen
                    dir += 1.0 / 3.0 * Math.PI;
                    break;
                    // Richtung um 60 Grad nach rechts drehen
                case &apos;r&apos;:
                    dir -= 1.0 / 3.0 * Math.PI;
                    break;
            }
        }
    }
}

 public class Koch extends Applet implements ActionListener {

     private Button fein;
     private Button grob;

     private int max = 0;               // Zunächst die Ausgangsfigur anzeigen
     private StringBuilder sInit;       // dient als Vorlage für Ersetzungen
     private String sRep = "srsLsrs";   // Der Ersetzungstring für die koch&apos;sche
                                        // Schneeflocke
     private StringBuilder sNext;       // enthält die nächste Ersetzung

      public void paint(Graphics g) {
```

```
        setBackground(Color.WHITE);
        // Ein dunkles Blau als Zeichenfarbe auswählen
        g.setColor(new Color(0, 85, 180));

        // Den Anfangsstring initialisieren
        sInit = new StringBuilder("sLsLs");
        // Der Ausgabestring wird für die Ausgabe ohne Ersetzungen vorbereitet
        sNext = new StringBuilder(sInit);

        // Wenn Ersetzungen vorgenommen werden sollen
        if(max > 0)
           for(int j=0; j<max; j++) {
               // den Ausgabestring löschenlöschen
               sNext.replace(0, sNext.length(), "");

               // Den Ausgangsstring zeichenweise durchsuchen und den
               // Ausgabestring entsprechend aufbauen
               for(int i=0; i<sInit.length(); i++) {
                   // In Abhängigkeit vom gefundenen Zeichen
                   char c = sInit.charAt(i);
                   switch(c) {
                       case &apos;s&apos;:
                           // Den Ersetzungsstring anhängen
                           sNext.append(sRep);
                           break;
                       default:
                           // Alle anderen Zeichen einfach anhängen
                           sNext.append(c);
                   }
               }
               // Für weitere Iterationen wird der Ausgabestring in den
               // Ausgangsstring kopiert.
               sInit.replace(0, sInit.length(), sNext.toString());
           }
        // Ein Interpreterobjekt erzeugen
        Interpret n = new Interpret(sNext.toString(), g);
        // Die Koch&apos;sche Schneeflocke hat eine fraktale Dimension von 3.
        // Die Länge der Basisstrecke wird entsprechend der Iterationsstufe
        // so angepasst, dass die Gesamtgröße der Figur erhalten bleibt.
        n.reFormat(180.0/Math.pow(3.0, max), 30, 210);
        n.show();
        if(0 == max)
            g.drawString("Ausgangsfigur", 90, 45);
        else
            g.drawString(String.valueOf(max) + ". Iteratonsstufe", 80, 45);
    }

    public void init() {
        // Die Button-Objekte werden instanziert und initialisiert
        fein  = new Button("fein");
        grob = new Button("grob");

        fein.setBounds (10,30,100,40);
        add(fein);
        grob.setBounds (10,70,100,40);
        add(grob);
        fein.addActionListener(this);
        grob.addActionListener(this);
    }

    public void actionPerformed(ActionEvent event) {
        // Die Button-Events ändern die Anzahl der Iterationen
        if(event.getSource() == fein) {
            if(max<8)
                max++;
        }
        else {
            if(max>0)
                max--;
        }
        repaint();
    }
}
```