```java
// Lagerkräfte eines Durchlaufträgers mit vier Stützen

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class DLT4FR extends JFrame implements ActionListener {
    private static final long serialVersionUID = 0320;
    private JTextField[] distanzen, forces;
    private JButton buttonF;
    private JPanel hauptRahmen, lRahmen, oRahmen, uRahmen;
    private JLabel[] input;
    // Konstruktor
    public DLT4FR() {
        super("Durchlauftr\u00e4ger");
        int index;
        Container fenster = getContentPane();
        // Button, Textfelder
        buttonF = new JButton("Lagerkr\u00e4fte / (qL)");
        buttonF.addActionListener(this);
        distanzen = new JTextField[2];
        forces = new JTextField[4];
        input = new JLabel[4];
        // Rahmen
        hauptRahmen = new JPanel();
        hauptRahmen.setLayout(new GridLayout(4,1,5,5));
        lRahmen = new JPanel();
        lRahmen.setLayout(new GridLayout(1,3,0,0));
        lRahmen.setBackground(Color.LIGHT_GRAY);
        oRahmen = new JPanel();
        oRahmen.setLayout(new GridLayout(1,2,20,0));
        uRahmen = new JPanel();
        uRahmen.setLayout(new GridLayout(1,4,5,5));
        input[0] = new JLabel("a/L>0",JLabel.CENTER);
        input[1] = new JLabel("(a+c)/L<1",JLabel.CENTER);
        input[1].setForeground(Color.WHITE);
        input[2] = new JLabel("c/L>0",JLabel.CENTER);
        input[3] = new JLabel("\u00a9 pwill",JLabel.CENTER);
        input[3].setForeground(Color.LIGHT_GRAY);
        for (index = 1; index <=3; index++) {
                lRahmen.add(input[index-1]);
        }
        for (index = 1; index <=2; index++) {
                distanzen[index-1] = new JTextField();
                distanzen[index-1].setBackground(Color.WHITE);
                oRahmen.add(distanzen[index-1]);
                if(index < 2) {
                    oRahmen.add(input[3]);
                }
        }
        for (index = 1; index <=4; index++) {
```

```
                forces[index-1] = new JTextField();
                forces[index-
1].setBackground(Color.LIGHT_GRAY);
                uRahmen.add(forces[index-1]);
        }
        hauptRahmen.add(lRahmen);
        hauptRahmen.add(oRahmen);
        hauptRahmen.add(buttonF);
        hauptRahmen.add(uRahmen);
        hauptRahmen.setBackground(Color.LIGHT_GRAY);
        fenster.add(hauptRahmen);
        // Erscheinungsbild: Nimbus
        try {
     UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusL
ookAndFeel");
              }
              catch (InstantiationException e) {
              }
              catch (ClassNotFoundException e) {
              }
              catch (UnsupportedLookAndFeelException e) {
              }
              catch (IllegalAccessException e) {
              }
              SwingUtilities.updateComponentTreeUI(fenster);
           fenster.setVisible(true);
        }
        // Initialisierung
        public static void main(String[] args) {
          int xPos,yPos;
          JFrame frame = new DLT4FR();
          ExitWindow abbrechen = new ExitWindow();
          frame.addWindowListener(abbrechen);
          // Abfrage Bildschirmabmessungen
          Dimension dim =
Toolkit.getDefaultToolkit().getScreenSize();
          // Abmessungen des Applikationsfensters
          frame.setSize(320,160);
          // Positionierung des Applikationsfensters auf dem
Bildschirm
          xPos = (dim.width-320)/2;
          yPos = (dim.height-160)/2;
          frame.setLocation(xPos,yPos);
          // Anzeige des Rahmenfensters auf dem Desktop
          frame.setVisible(true);
        }
        // Berechnung der Lagerkraefte
        public void actionPerformed(ActionEvent event) {
          double a = 0., c = 0., c1 = 0., c2 = 0., c3 = 0., d1
= 0., d2 = 0.;
          double sd, fa, fb, f1, f2;
```

```
            int ih, index;
            boolean flag = false;
            if (event.getSource() == buttonF) {
                    for (index = 1; index <=2; index++) {
                        distanzen[index-
1].setBackground(Color.WHITE);
                    }
                    flag = false;
                    input[0].setForeground(Color.BLACK);
                    input[1].setForeground(Color.WHITE);
                    input[2].setForeground(Color.BLACK);
                   a = Double.parseDouble(distanzen[0].getText());
                   c = Double.parseDouble(distanzen[1].getText());
                        if (a+c>=0.99998) {

      input[1].setForeground(Color.RED);
                                    flag = true;
                        }
                        if (a<0.0) {

      input[0].setForeground(Color.RED);
                                flag = true;
                        }
                   if (c<0.0) {

      input[2].setForeground(Color.RED);
                                    flag = true;
                   }
            }
            // innenliegende Lager
            c1 = 2.*Math.pow(a,2.)*Math.pow(1.-a,2.);
            c2 = a*c*(1-Math.pow(a,2.)-Math.pow(c,2.));
            c3 = 2.*Math.pow(c,2.)*Math.pow(1.-c,2.);
            d1 = 0.25*a*(1.-2.*Math.pow(a,2.)+Math.pow(a,3.));
            d2 = 0.25*(1.-c)*(1.-2.*Math.pow(1.-
c,2.)+Math.pow(1.-c,3.));
            sd = c2*c2-c1*c3;
            fa = (c2*d2-c3*d1)/sd;
            fb = (c2*d1-c1*d2)/sd;
            if(a==0. & c==0.) {
                 f1 = 0.5;
                 f2 = 0.5;
            }
            else {
                 // Randkraefte
                 f2 = 0.5-a*fa-(1.-c)*fb;
                 f1 = 0.5-(1.-a)*fa-c*fb;
            }
            // Rundung
            ih = (int) Math.round(1000.* f1);
            f1 = ih/1000.;
```

```
            forces[0].setText(f1 + "");
            ih = (int) Math.round(1000.* fa);
            fa = ih/1000.;
            forces[1].setText(fa + "");
            ih = (int) Math.round(1000.* fb);
            fb = ih/1000.;
            forces[2].setText(fb + "");
            ih = (int) Math.round(1000.* f2);
            f2 = ih/1000.;
            forces[3].setText(f2 + "");
                 // Ausnahmebehandlung
                 if (flag) {
                      for (index = 1; index <=4; index++) {
                           forces[index-1].setText("");
                      }
                 }
        }
}

class ExitWindow extends WindowAdapter {
        public void windowClosing(WindowEvent e) {
          System.exit(0);
        }
        // Aufruf leerer WindowListener-Methoden
        public void windowIconified(WindowEvent we) {
        }
        public void windowOpened(WindowEvent we) {
        }
        public void windowClosed(WindowEvent we) {
        }
        public void windowDeiconified(WindowEvent we) {
        }
        public void windowActivated(WindowEvent we) {
        }
}
```