

```

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

public class PrincipalStresses extends JFrame
                                implements ActionListener,
WindowListener {

/**
 * Berechnung von Hauptspannungen im räumlichen Spannungsfeld
 * pwill@hs-mittweida.de
 */

    private static final long serialVersionUID = 012010;
    private JTextField[] spannungen, spannungenHS;
    private JButton buttonHS;
    private JPanel spannungstensor, vektorHS;
    private JPanel hauptRahmen=null;
    private float s1, s2, s3;

// Konstruktor
public PrincipalStresses() {
    super("Spannungstensor");
    int index;
    // Button, Textfelder
    buttonHS = new JButton("Hauptspannungen");
    buttonHS.addActionListener(this);
    spannungen = new JTextField[9];
    spannungenHS = new JTextField[3];

// ContentPane Hauptrahmen
    hauptRahmen = new JPanel();
    hauptRahmen.setLayout(new BorderLayout(5,5));

// Panel Spannungstensor
    spannungstensor = new JPanel();
    spannungstensor.setLayout(new GridLayout(3,3,2,2));
    spannungstensor.setBackground(Color.WHITE);
    for (index = 1; index <=9; index++) {
        spannungen[index-1] = new JTextField();
        if(index == 2 | index == 3 | index == 6) {
            spannungen[index-
1].setBackground(Color.LIGHT_GRAY);
        }

        spannungstensor.add(spannungen[index-1]);
    }
}

```

```

// Panel Hauptspannungen
vektorHS = new JPanel();
vektorHS.setLayout(new GridLayout(1,3,2,2));
vektorHS.setBackground(Color.WHITE);

for (index = 1; index <=3; index++) {
    spannungenHS[index-1] = new JTextField();
    spannungenHS[index-
1].setBackground(Color.LIGHT_GRAY);
    vektorHS.add(spannungenHS[index-1]);
}

// GUI
hauptRahmen.add("North", spannungsTensor);
hauptRahmen.add("Center", buttonHS);
hauptRahmen.add("South", vektorHS);
hauptRahmen.setBackground(Color.WHITE);
setContentPane(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(hauptRahmen);
}

// Initialisierung
public static void main(String[] args) {
    int xPos,yPos;
    JFrame frame = new PrincipalStresses();
    ExitWindow abbrechen = new ExitWindow();
    frame.addWindowListener(abbrechen);
    // Abfrage Bildschirmabmessungen
    Dimension dim =
Toolkit.getDefaultToolkit().getScreenSize();
    // Abmessungen des Applikationsfensters
    frame.setSize(270,190);
    // Positionierung des Applikationsfensters auf dem
Bildschirm
    xPos = (dim.width-270 )/2;
    yPos = (dim.height-190)/2;
}

```

```

        frame.setLocation(xPos,yPos);
        // Anzeige des Rahmenfensters auf dem Desktop
        frame.setVisible(true);
    }

    // Aufbereitung von Daten, Berechnung der Hauptspannungen
    public void actionPerformed(ActionEvent event) {
        double sxx = 0, syy = 0, szz = 0, txy = 0, txz = 0,
        tyz = 0;
        double invariante1, invariante2, invariante3, p, q,
        phi;
        int is;
        if (event.getSource() == buttonHS) {
            sxx =
            Double.parseDouble(spannungen[0].getText());

            txy =
            Double.parseDouble(spannungen[3].getText());
            spannungen[1].setText(txy + "");
            syy =
            Double.parseDouble(spannungen[4].getText());

            txz =
            Double.parseDouble(spannungen[6].getText());
            spannungen[2].setText(txz + "");
            tyz =
            Double.parseDouble(spannungen[7].getText());
            spannungen[5].setText(tyz + "");
            szz =
            Double.parseDouble(spannungen[8].getText());
        }

        // Hilfsgrößen
        invariante1 = sxx + syy + szz;
        invariante2 = sxx*syy + syy*szz + sxx*szz - (txy*txy
+ txz*txz + tyz *tyz);
        invariante3 = sxx*syy*szz + 2*txy*txz*tyz-
sxx*tyz*tyz - syy*txz*txz - szz*txy*txy;
        p = (Math.pow(invariante1,2)/3 - invariante2)/3;
        q = (-Math.pow(invariante1,3)/27
+invariante1*invariante2/6 - invariante3/2);
        phi = Math.acos(-q/(Math.pow(p,1.5)));

        // Hauptspannungen (zwei Nachkommastellen)
        s1 = (float) (invariante1/3 +
2*Math.sqrt(p)*Math.cos(phi/3));
        is = Math.round(s1*100f);
        s1 = is/100f;
        spannungenHS[0].setText(s1 + "");
        s2 = (float) (invariante1/3 +
2*Math.sqrt(p)*Math.cos(phi/3 - 2*Math.PI/3));

```

```

        is = Math.round(s2*100f);
        s2 = is/100f;
        spannungenHS[1].setText(s2 + "");
        s3 = (float) (invariante1/3 +
2*Math.sqrt(p)*Math.cos(phi/3 + 2*Math.PI/3));
        is = Math.round(s3*100f);
        s3 = is/100f;
        spannungenHS[2].setText(s3 + "");
    }

}

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) {
    }
}

```