

Hochschule Harz	FB Automatisierung und Informatik
Programmierung2	Dipl.-Inf. Dipl.-Ing. (FH) M. Wilhelm
Tutorial 02:	„Programmierung 2“ für MI / WI Thema: Swing und GridBagLayout

Versuchsziele

Kenntnisse in der Anwendung von:

- Erstellen einer Swing-Anwendung
- Benutzung von Klassen
 - Definition
 - Erzeugung
 - Verwendung von Arrays
 - Sortierung

Tutorial02

Swing: Erstellen eines JFrames mit GridBagLayout

In dieser Aufgabe soll ein einfaches Fenster erstellt werden und drei Schalter mit einfachen Aktionen verknüpft werden.

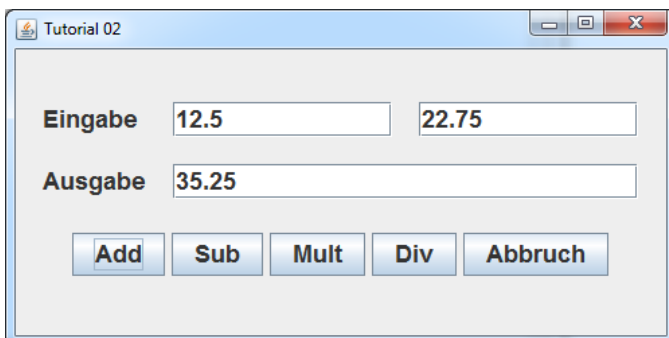


Abbildung 1 Musterlösung

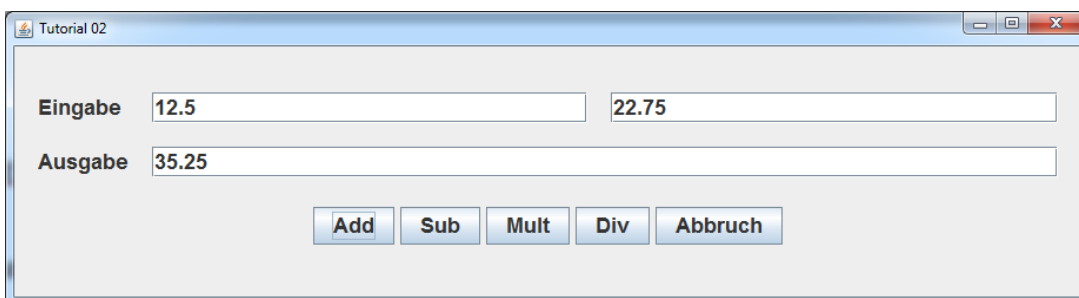


Abbildung 2 Musterlösung mit größerer Breite

1. Teilaufgaben: Projekt erstellen

- Erstellen Sie ein neues Eclipse-Projekt:
 - Projektname: Tutorial02
 - Klassenname: JFrame02

- Erstellen Sie eine neue Klasse
 - Menü File, Eintrag New, Eintrag class
 - Name: JFrame02
- Erstellen Sie aus dem vorgegebene JFrame-Rahmen folgendes Programm
 - Zwei JLabel-Elemente
 - Drei JTextField als Editorzeile
 - Fünf Schalter mittels JPanel und FlowLayout eingetragen

2. Teilaufgaben: GridBagLayout erstellen

- Fügen Sie die einzelnen GUI-Elemente in das JFrame
 - JLabel labelEingabe
 - .gridx
 - .gridy
 - .gridwidth
 - .weightx
 - .anchor
 - .fill
 - JTextField input1
 - .gridx
 - .gridy
 - .gridwidth
 - .weightx
 - .anchor
 - .fill
 - JTextField input1
 - .gridx
 - .gridy
 - .gridwidth
 - .weightx
 - .anchor
 - .fill
 - JLabel labelAusgabe
 - .gridx
 - .gridy
 - .gridwidth
 - .weightx
 - .anchor
 - .fill
 - JTextField output
 - .gridx
 - .gridy
 - .gridwidth
 - .weightx
 - .anchor
 - .fill
 - JPanel panelBn
 - .gridx
 - .gridy
 - .gridwidth
 - .weightx
 - .anchor
 - .fill
 - JButton bnAdd
 - JButton bnSub
 - JButton bnMult
 - JButton bnDiv
 - JButton bnEsc

- Verknüpfen Sie die fünf Schalter mit einem ActionListener und jeweils einer Methode
 - Add:
 - Ausgabe im JTextField „output“ den Inhalt von input1
 - Sub:
 - Ausgabe im JTextField „output“ den Inhalt von input2
 - Mult:
 - „Auslesen“ des ersten Strings
 - Umwandeln in eine double-Zahl
 - Eine Eins addieren
 - Ausgabe in „output“
 - Div:
 - „Auslesen“ des ersten Strings
 - Umwandeln in eine double-Zahl
 - Multiplizieren mit zwei
 - Ausgabe in „output“
 - Abbrechen:
 - Beenden des Programms

Rahmen:

```
//Titel:          2. Tutorial, Programmierung 2
//Version:       1,0
//Copyright:     Copyright (c) 2013
//Autor:         M. Wilhelm
//Organisation: HS-Harz
//Beschreibung: GridBagLayout
```

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

```
public class Tutorial02 extends JFrame {

    JLabel jLabel1 = new JLabel("Eingabe");
    JTextField input1 = new JTextField("input1");
    JButton btnAdd = new JButton("Add");

    public Tutorial02() {
        setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        setGUI();
    }

    private void setGUI() {
        setSize(500, 250);
        setTitle("Tutorial 02");

        btnAdd.setFont( new Font("Arial", Font.BOLD,18) );

        this.getContentPane().setLayout( new GridBagConstraints() );

        /*
        this.getContentPane().add(jLabel1,
            new GridBagConstraints(
                0, 0, 1, 1, 0.0, 0.0
                ,GridBagConstraints.WEST,
                GridBagConstraints.NONE,
                new Insets(0,0,0,0), 0, 0));
        */

        JPanel panelBn = new JPanel();
        panelBn.setLayout( new FlowLayout() );
```

```

        panelBn.add(btnAdd);
    } // setGUI

    // siehe Hausaufgabe
    public enum Mathematik {
        Add, Sub, Mult, Div
    }

    // siehe Hausaufgabe
    private void calc(int op) {
    }

    private void btnAdd_Click() {
        // Hier aufruf von calc mit einem Parameter (Hausaufgabe)
    }

    private void btnEsc_Click() {
        setVisible(false);
        dispose(); // System.exit(0);
    }

    public static void main(String[] args) {
        Tutorial02 frame = new Tutorial02();
        frame.setVisible(true);
    }
} // Tutorial02

```

Hinweise:

- Methoden der Klassen JLabel
 - setText(String)
 - setFont
- setFont Methoden der Klassen JTextField
 - setText(String)
 - getText()
 - setFont

Umwandlung eines String in einem Integer:

```

String sValue="123";
int n;
try {
    n= Integer.parseInt(sValue);
}
catch ( NumberFormatException e ){
    System.out.println( "Konvertierungsfehler: "+sValue );
}

```

Umwandlung eines String in einem Double:

```

String sValue="123";
double d;
try {
    d= Double.parseDouble(sValue);
}
catch ( NumberFormatException e ){
    System.out.println( "Konvertierungsfehler: "+sValue );
}

```

Anhang:

Klasse GridBagConstraints:

- int gridx Element in der gridx-te Spalte
- int gridy Element in der gridy-te Zeile
- int gridwidth Anzahl der Spalten, die das Element benötigt
- int gridheight Anzahl der Zeilen, die das Element benötigt
- double weightx Verteilung von Platzänderung (Prozentual)
- double weighty Verteilung zusätzlichen Platzes (Prozentual)
- int anchor Verknüpfung (NORTHEAST, EAST, SOUTHEAST, CENTER, SOUTH, SOUTHWEST, WEST, NORTH, NORTHWEST)
- int fill Ausfüllen (NONE, HORIZONTAL, BOTH, VERTICAL)
- Insets insets äußerer Rand, **Top, Left, Bottom, Right**
- int ipadx vergrößert das GUI-Element, Breite des GUI-Elements
- int ipady vergrößert das GUI-Element, Höhe des GUI-Elements

Beispiel:

```
this.getContentPane().add(jLabel1,  
    new GridBagConstraints(0, 0, 1, 1, 0.0, 0.0,GridBagConstraints.WEST,  
        GridBagConstraints.NONE, new Insets(4,20,4,0), 00, 0));  
  
this.getContentPane().add(tName,  
    new GridBagConstraints(1, 0, 1, 1, 1.0, 0.0,GridBagConstraints.WEST,  
        GridBagConstraints.HORIZONTAL, new Insets(4,0,4,20), 00, 0));  
  
JPanel panelBn = new JPanel();  
panelBn.setLayout( new FlowLayout() );  
panelBn.add(btnOk);  
panelBn.add(btnEsc);  
this.getContentPane().add(panelBn,  
    new GridBagConstraints(0, 1, 2, 1, 0.0, 0.0,GridBagConstraints.CENTER,  
        GridBagConstraints.HORIZONTAL, new Insets(4,0,4,0), 0, 0));
```

