

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

Versuchsziele

Kenntnisse in der Anwendung von:

- Erstellen einer Swing-Anwendung
- Benutzung von
 - GridbagLayout
 - Enums

Aufgabe 02

1) Mathematische Aktionen implementieren

In dieser Teilaufgabe sollen die mathematischen Funktionen vollständig implementiert werden:

Teilaufgaben:

- Erstellen einer enum-Klasse
 - Name: „Mathematik“
 - Elemente:
 - Add
 - Sub
 - Mult
 - Div
- Erstellen einer neuen Methode:
 - Name: „calc“
 - Parameter: Mathematik „operator“
 - Rückgabe: keine
 - Quellcode:
 - Umwandeln der beiden Eingabewerte in zwei double-Variablen a und b
 - Je nach Wert des Parameter „operator“ wird die Berechnung ausgeführt und das Ergebnis in das JTextField ausgegeben
 - Entweder mehrere if-Anweisungen oder eine switch/case-Anweisung
 - Beispiel: `output.setText("Addition: "+a+ " + "+b+ " = "+(a+b));`

Methoden eines JTextField:

- `String s = getText()`
- `setText(String)`

Umwandlung eines String in einem double-Wert:

```
String sValue="123.44";
Double d;
try {
    d= Double.parseDouble(sValue);
}
catch ( NumberFormatException e ){
    System.out.println( "Konvertierungsfehler: "+sValue );
}
```



Abbildung 1 Ergebnis einer Addition

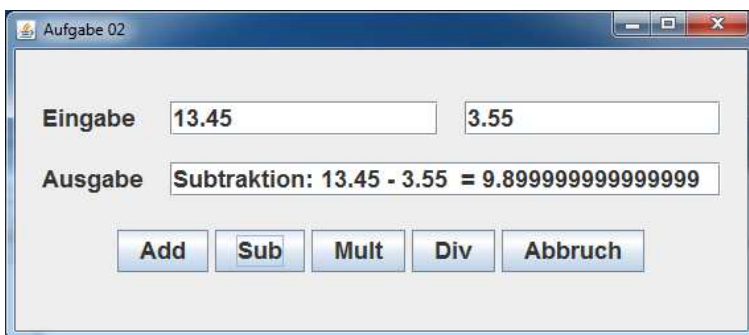


Abbildung 2 Ergebnis einer Subtraktion

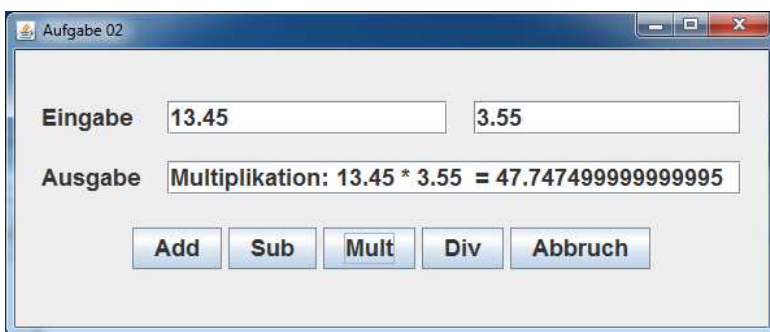


Abbildung 3 Ergebnis einer Multiplikation

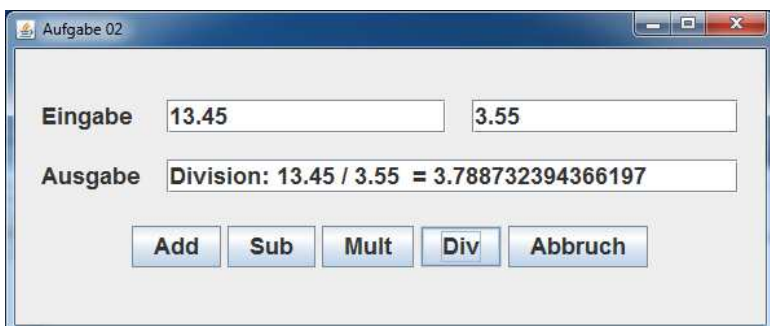


Abbildung 4 Ergebnis einer Division

2) Swing: Umbau des JFrames

In dieser Aufgabe soll ein Texteditor, JTextArea, eingebaut werden. Dazu benötigt man noch einen Schalter „clear“, der den Editor löscht.

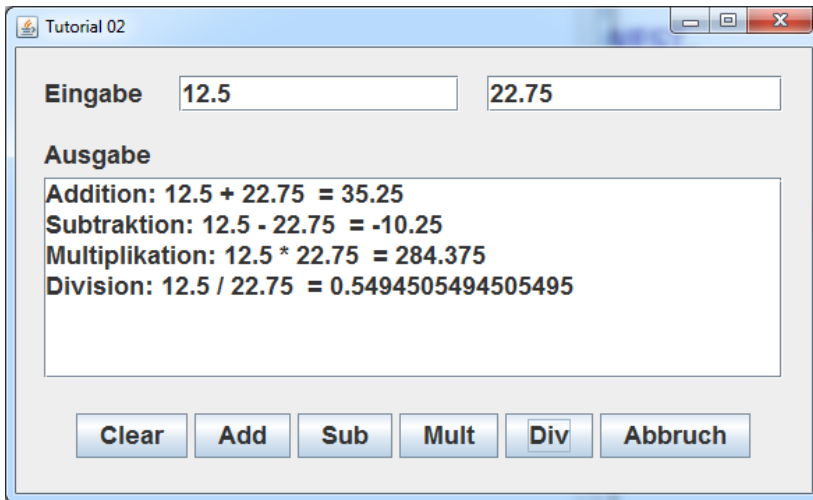


Abbildung 5 Musterlösung

Teilaufgaben:

- Ersetzen des Ausgabe-JTextField durch einen mehrzeiligen Editor, JTextArea.
 - JTextArea output
 - gridx
 - gridy
 - gridwidth
 - weightx
 - weighty ändern
 - anchor
 - fill ändern
- Einbau des Schalters „clear“
 - Deklarieren und Erzeugen
 - ActionPerformed zuweisen
 - Methode implementieren

Methoden der Klasse JTextArea:

- **setText(String)**
- String s = **getText()**
- **append(String)**
- **setFont(new Font("Arial", Font.BOLD,18))**

Hinweis:

- Zeilenumbruch:
 - "\n"

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(jLabell1,  
    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));
```



Hinweise:

- **JLabel erhalten immer FILL=NONE**
- double weightx Wieviel Prozent der zusätzlichen Pixel bekommt diese Zelle
- double weighty Wieviel Prozent der zusätzlichen Pixel bekommt diese Zelle
- int fill Soll das Element in der Zelle ausgefüllt werden
 - NONE, weightx
 - HORIZONTAL weighty
 - VERTICAL weightx und weighty
 - BOTH