

# Schueler-Organisiertes Lernen am Beispiel von Grafischen Benutzer-Schnittstellen in Java Tag der Offenen Tür - GTS 2008

<http://worgtsone.scienceontheweb.net/worgtsone/> - [mailto: worgtsone @ hush.com](mailto:worgtsone@hush.com)

Sa 8. Nov 2008 – 13. Oktober 2011

## Inhaltsverzeichnis

1	Intro	1
2	Objektorientierung	2
3	Listener sind willkommene Sklaven	4



## 1 Intro

Morgens um 1000 standen wir da mit einem funktionsfähigen Java-Editor und ein paar Java-Konzepten.

Nachmittags um 1600 hatte die Hälfte der Schüler ihren Taschenrechner fertig.

## 2 Objektorientierung

nimmt einem der Java-Editor fast völlig ab, daher nur kommentierter Quelltext.

```
// lets import awt stuff. pupils and i dislike unreliable swing bullshit.
import java.awt.*;
import java.awt.event.*;

// Frame is father (or parent, if you like).
// ActionListener is easy, but capable Event-Model of Java 1.0, 1992.
class TaRe01 extends Frame implements ActionListener {
//   worgtsone, TaRe01, Sa 8. Nov 20:33:57 CET 2008

// name all the members which happen to occur in 2 methods
  CheckboxGroup cbg1;  TextField tf1; TextField tf2; TextField tf3;

// first, a short main method. We call the constructor by constructing...
  public static void main (String[]args) {   TaRe01 tare01 = new TaRe01 (); }

// ... and put the rest into constructor ( except Event Treatment)
  TaRe01 () {
    setTitle ("Taschenrechner01");
    setLayout (new GridLayout (0, 1));          // quick and nice.
                                                // means: rows as you like, one column

    setSize (300, 350);
    setLocation (50, 50);

// define GUI elements
// browse
// http://www.dpunkt.de/java/Referenz/Das_Paket_java.awt/1.html
// for reference
    Button b1 = new Button ("Quit");
    add (b1);
    b1.addActionListener (this);

// there is a checkbox-group since java 1.1
// it MAKES radioButton functionality without saying so.
    cbg1 = new CheckboxGroup ();
    Checkbox cb1 = new Checkbox ("Addieren", true, cbg1);
    Checkbox cb2 = new Checkbox ("Subtrahieren", true, cbg1);
    Checkbox cb3 = new Checkbox ("Multizieren", true, cbg1);
    Checkbox cb4 = new Checkbox ("Dividieren", true, cbg1);
    add (cb1);
    add (cb2);
    add (cb3);
    add (cb4);

    tf1 = new TextField ("4", 8);
    add (tf1);
    tf2 = new TextField ("5", 8);
    add (tf2);
```

```

tf3 = new TextField ("Ergebnis", 16);
this.add (tf3);

Button b2 = new Button ("Rechne!");
this.add (b2);
b2.addActionListener (this);
Button b3 = new Button ("pack()");    // pack() is good and useful
this.add (b3);
b3.addActionListener (this);
// pack and show
this.pack ();                        // pack() is good and useful
this.setVisible (true);
}

public void actionPerformed (ActionEvent e) {
String c = e.getActionCommand ();
// System.out.println (c);
if (c.equals ("Quit"))    System.exit (0);

if (c.equals ("pack()"))    this.pack ();

if (c.equals ("Rechne!")) {
double z1 = 0;
double z2 = 0;
String result = "";
try {
z1 = Double.parseDouble (tf1.getText ().replace (',', '.'));
} catch (Exception nfe) {    // number format exception, maybe???
result = "evil";
}
try {
z2 = Double.parseDouble (tf2.getText ().replace (',', '.'));
} catch (Exception nfe) {    // number format exception, maybe???
result = "evil";
}
double r = 0;                // result
if (result.equals ("evil"))
tf3.setText ("BöseZahl!");
else {
String label = cbg1.getSelectedCheckbox ().getLabel ();
if (label.equals ("Addieren"))    r = z1 + z2;
if (label.equals ("Subtrahieren"))    r = z1 - z2;
if (label.equals ("Multizieren"))    r = z1 * z2;
if (label.equals ("Dividieren"))    r = z1 / z2;
tf3.setText (r + "");
}    }    }}

```

### 3 Listener sind willkommene Sklaven

bei Einsparen von Knöpfen. Hallo Erik ;>).



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

class Tare02 extends Frame
    implements ActionListener, ItemListener, TextListener {
//    wortgsone, Tare02, Mo 11. Nov 2008
    CheckboxGroup cbg1;
    TextField tf1;
    TextField tf2;
    TextField tf3;

    public static void main (String[]args) { Tare02 Tare02 = new Tare02 (); }

    Tare02 () {
        this.setTitle ("Taschenrechner01");
        this.setLayout (new GridLayout (0, 1));
        this.setSize (300, 350);
        this.setLocation (50, 50);

        Button b1 = new Button ("Quit");
        this.add (b1);
        b1.addActionListener (this);

        cbg1 = new CheckboxGroup ();
        Checkbox cb1 = new Checkbox ("Addieren", true, cbg1);
        Checkbox cb2 = new Checkbox ("Subtrahieren", true, cbg1);
        Checkbox cb3 = new Checkbox ("Multizieren", true, cbg1);
        Checkbox cb4 = new Checkbox ("Dividieren", true, cbg1);
```

```

this.add (cb1);    cb1.addItemListener (this);
this.add (cb2);    cb2.addItemListener (this);
this.add (cb3);    cb3.addItemListener (this);
this.add (cb4);    cb4.addItemListener (this);

tf1 = new TextField ("4", 8);
tf2 = new TextField ("5", 8);
tf3 = new TextField ("Ergebnis", 16);
this.add (tf1);    tf1.addTextListener (this);
this.add (tf2);    tf2.addTextListener (this);
this.add (tf3);
this.pack ();      // pack() is good and useful
this.setVisible (true);
}

public void actionPerformed (ActionEvent e) {
    String c = e.getActionCommand ();    if (c.equals ("Quit")) System.exit (0);
}

public void itemStateChanged (ItemEvent ie) {    rechne (); }

public void textValueChanged (TextEvent te) {    rechne (); }

public void rechne () {
    double z1 = 0;
    double z2 = 0;
    String result = "";
    try { z1 = Double.parseDouble (tf1.getText ().replace (',', '.'));    }
    catch (Exception nfe) {    result = "evil";    }
    try { z2 = Double.parseDouble (tf2.getText ().replace (',', '.'));
    } catch (Exception nfe) {    result = "evil";    }
    double r = 0;    // result
    if (result.equals ("evil"))
        tf3.setText ("BöseZahl!");
    else {
        String label = cbg1.getSelectedCheckbox ().getLabel ();
        if (label.equals ("Addieren"))
            r = z1 + z2;
        if (label.equals ("Subtrahieren"))
            r = z1 - z2;
        if (label.equals ("Multizieren"))
            r = z1 * z2;
        if (label.equals ("Dividieren"))
            r = z1 / z2;
        tf3.setText (r + "");
    } }}

```