

Panel als Container für GUI Elemente

Im folgenden Beispiel wird der Button nicht direkt dem Frame Objekt hinzugefügt, sondern einem "Container", dem Panel-Objekt. Das Panel Objekt wird dann am Ende im Frame angezeigt. Damit hat man die Möglichkeit, zunächst dem Panel alle GUI Elemente hinzuzufügen, die man anzeigen möchte, und dann das Panel mit allen Elementen im Frame auszugeben.



Frame



Frame

HalloWelt.java

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

public class HalloWelt extends Frame
{
    // Ein Panel, um die Elemente aufzunehmen
    Panel panel= new Panel();
    // Das Button Objekt
    Button button = new Button("Schaltfläche");

    // Konstruktor
    public HalloWelt ()
    {
        setTitle("Hallo Welt");

        // Interaktionen mit dem Fenster behandeln
        addWindowListener(new TestWindowListener());

        // Einstellungen für den Button
        button.setForeground(Color.RED); //
        // Vordergrundfarbe auf "rot" setzen
        button.setBackground(Color.WHITE); //
        // Hintergrundfarbe auf "weiß" setzen
        // Interaktionen mit dem Button behandeln
        button.addActionListener(new TestActionListener()); //
        // ActionListener für Schaltfläche hinzufügen

        // Button zum Panel-Container hinzufügen
    }
}
```

```
panel.add(button);
// Dem Panel können jetzt weitere GUI Elemente zugefügt werden...

// Panel im Fenster anzeigen
add(panel);

pack(); // Fenstergröße
auf die benötigte Größe //

"zusammenpacken"
setVisible(true);
}

class TestWindowListener extends WindowAdapter
{
    public void windowClosing(WindowEvent e)
    {
        e.getWindow().dispose();
        System.exit(0);
    }
}

class TestActionListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        System.out.println("Schaltfläche wurde gedrückt");
    }
}

public static void main (String args[])
{
    new HalloWelt();
}
}
```

Aufgaben

A1

- Teste das neue Programm
- Füge einen weiteren Button hinzu, der einen anderen Text auf der Konsole ausgibt, wenn er angeklickt wird.

A2

Mit

```
TextField name = new TextField("Name");
```

kann man ein Textfeld mit dem Namen name definieren.

Mit der Methode `getText` kann man den Inhalt des Textfelds später auslesen, z.B. wir durch folgenden Ausdruck der Inhalt des Textfelds in der Variablen `myname` abgelegt:

```
String myname = name.getText();
```

- Definiere eine globales Textfeld, füge das dem Panel hinzu und modifiziere die Handlerfunktion für deinen Button so, dass ein Klick den Benutzer mit dem Text "Hallo <NAME>" auf der Konsolenausgabe begrüßt, wobei <NAME> durch den Inhalt des Textfelds ersetzt werden soll.

From:

<https://info-bw.de/> -

Permanent link:

<https://info-bw.de/faecher:informatik:oberstufe:java:awt:panel:start>

Last update: **11.03.2020 14:12**

