

## Übung 2 – Medientechnik

### **Inhalt:** Swing (Teil 2)

- Controller als anonyme innere Klassen
- Layout-Manager
- Look-And-Feel

### **Aufgaben**

Laden Sie sich das Archiv *swing2.zip* von der Übungswebseite herunter und entpacken Sie es in ein geeignetes Verzeichnis in Ihrem Home-Ordner. Das Archiv enthält die Pakete *swing2adder* und *swing2layout*.

#### *Aufgabe 1: Controller als anonyme innere Klassen*

Für diese Aufgabe benötigen Sie das Paket *swing2adder*. Dieses enthält den Code für den Addierer aus der letzten Übung. Allerdings fehlt der *Controller*. Für jeden Button (also *Add*- und *Exit*) soll jeweils ein *ActionListener* als anonyme innere Klassen der *View*-Klasse implementiert werden. Der *ActionListener* des *Add*-Buttons soll als Controller dienen, um das Model zu ändern. Dabei können Teile des Codes des *Controllers* aus Übung 1 wieder verwendet werden.

#### *Aufgabe 2: Layout-Manager*

Für diese Aufgabe wird das Paket *swing2layout* benötigt. Dieses Paket enthält die Klassen *FlowLayoutDemo*, *BorderLayoutDemo*, *GridLayoutDemo* und *GridBagLayoutDemo*. Der Code in den Klassen enthält Lücken, die ausgefüllt werden müssen. Entsprechende Stellen im Code sind mit Kommentaren versehen. Vervollständigen Sie die Klassen unter Verwendung der in den Kommentaren vorgegebenen Layout-Manager. Nutzen Sie die Java-API – Dokumentation (<http://java.sun.com/j2se/1.5.0/docs/api/>), um an die notwendigen Informationen zu gelangen. Kompilieren Sie die Klassen und testen Sie die fertigen Anwendungen. Verändern Sie jeweils die Fenstergrößen und beobachten Sie, was passiert. Experimentieren Sie mit entsprechenden Parametern, um das Layout zu verändern. Welche Wirkung hat beim *GridLayout* die gleichzeitige Angabe von Zeilen und Spalten? Wodurch wird beim *GridBagLayout* die Anzahl der Zeilen und Spalten festgelegt?

#### *Aufgabe 3: Look-And-Feel*

Es sollen verschiedene Look-And-Feel-Varianten getestet werden. Ohne spezielle Angaben entspricht das Look-And-Feel i. d. R. der jeweiligen Plattform. Fügen Sie in einer der Beispielanwendungen zu Beginn der Main-Methode den folgenden Code hinzu:

```
try {
    javax.swing.UIManager.setLookAndFeel( LookAndFeelName );
}
catch (Exception e) {
    System.out.println("Look and Feel not found!");
}
```

Dabei ist *LookAndFeelName* durch den String für das entsprechende Look-And-Feel zu ersetzen. Gültige Werte sind u. a.:

```
"javax.swing.plaf.metal.MetalLookAndFeel"
"com.sun.java.swing.plaf.motif.MotifLookAndFeel"
"com.sun.java.swing.plaf.gtk.GTKLookAndFeel"
```

Das plattformspezifische Look-And-Feel entspricht auf dieser Plattform dem *CrossPlatformLookAndFeel*, so dass hierbei keine Unterschiede zu sehen sind.

*Zusatzaufgabe: Praktische Anwendung der erworbenen Kenntnisse*

Programmieren Sie einen Taschenrechner. Die grafische Oberfläche des Rechners soll das unten skizzierte Layout besitzen. Verwenden Sie dafür die zuvor besprochenen Layout-Manager. Achten Sie darauf, dass zwischen den einzelnen Elementen Abstände existieren. Das Look-And-Feel soll dem *GTKLookAndFeel* entsprechen. Verwenden Sie anonyme innere Klassen, um die GUI-Elemente an das *Model* des Rechners anzubinden.

