

Swing 101: Basics

Carsten Gips (FH Bielefeld)

Unless otherwise noted, this work is licensed under CC BY-SA 4.0.

Wiederholung GUI in Java

- **AWT:** `abstract window toolkit`
 - Älteres Framework ("Legacy")
 - "Schwergewichtig": plattformangepasst
 - Paket `java.awt`
- **Swing**
 - Nutzt AWT
 - "Leichtgewichtig": rein in Java implementiert
 - Paket `javax.swing`
- **JavaFX**
 - Soll als Ersatz für Swing dienen
 - Community eher verhalten
 - Weiterentwicklung immer wieder unklar
 - Nicht mehr im JDK/Java SE Plattform enthalten
 - Vergleichsweise komplexes Framework, auch ohne Java programmierbar (Skriptsprache FXML)

Graphische Komponenten einer GUI

- Top-Level Komponenten
 - Darstellung direkt auf Benutzeroberfläche des Betriebssystems
 - Beispiele: Fenster, Dialoge
- Atomare Komponenten
 - Enthalten i.d.R. keine weiteren Komponenten
 - Beispiele: Label, Buttons, Bilder
- Gruppierende Komponenten
 - Bündeln und gruppieren andere Komponenten
 - Beispiele: JPanel

Achtung: Unterteilung nicht im API ausgedrückt: Alle Swing-Bausteine leiten von Klasse `javax.swing.JComponent` ab!

=> Nutzung “falscher” Methoden führt zu Laufzeitfehlern.

Ein einfaches Fenster

```
public class FirstWindow {  
    public static void main(String[] args) {  
        JFrame frame = new JFrame("Hello World :");  
  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
        frame.pack();  
        frame.setVisible(true);  
    }  
}
```

- Swing baut auf AWT auf und nutzt dieses
- JavaFX ist moderner, aber kein Swing-Ersatz geworden
- Basics:
 - Swing-Fenster haben Top-Level-Komponenten: `JFrame`, ...
 - Atomare Komponenten wie Buttons, Label, ... können gruppiert werden
 - Fenster müssen explizit sichtbar gemacht werden
 - Nach Schließen des Fensters läuft die Applikation weiter (Default)
 - Swing-Events werden durch den *Event Dispatch Thread* (EDT) verarbeitet
=> Aufpassen mit Multithreading!

LICENSE



Unless otherwise noted, this work is licensed under CC BY-SA 4.0.