

JAVA 08

Exkurs AWT

JAVA 08 - Exkurs AWT

- Das **Abstract Window Toolkit (AWT)** von Java bietet grundlegende Möglichkeiten eine Fenster-Anwendung zu entwickeln.
- **Unabhängig vom Betriebssystem (Windows, MAC, Linux)** können hiermit also erste grafische Benutzeroberflächen erstellt werden, um nicht mehr nur auf **System.out.println()** und den **Scanner()** angewiesen zu sein.
- Es gibt eine Reihe an Weiterentwicklungen (wie zum Beispiel **Swing**), die auf AWT aufbauen und weitere Funktionalitäten bieten.

1. Das Frame Element

- Mit dem Frame Element können Fenster erstellt werden.

```
public class AwtMain {  
    public static void main (String args[]) {  
        Frame myWindow = new Frame();  
        myWindow.setTitle („Hello AWT-World");  
        myWindow.setSize(400,100);           // Fenstergröße einstellen  
        myWindow.setVisible(true);           // Fenster sichtbar machen  
    }  
}
```

1. Das Frame Element

- Wie schließen wir das Fenster (und beenden das Programm)?

```
Frame myWindow = new Frame();  
addWindowListener(new TestWindowListener()); // so...  
myWindow.setVisible(true);
```

- Der hinzugefügte ***Listener*** muss noch deklariert werden (Einzelheiten -> JAVA II):

```
class TestWindowListener extends WindowAdapter {  
    public void windowClosing(WindowEvent e) {  
        e.getWindow().dispose(); // Fenster "killen", = schließen  
        System.exit(0);          // VM "killen" = Programm beenden  
    }  
}
```

2. Das Label Element

- Mit dem Label Element können einzeilige Texte ausgegeben werden.

```
Frame myWindow = new Frame();
```

```
Label label1 = new Label();
```

```
label1.setText(„Ich bin ein einzeiliger Text“);
```

```
myWindow.add(label1);
```

```
Label label2 = new Label(„Eine Konstruktor-Beschriftung“);
```

```
myWindow.add(label2);
```

```
myWindow.add(new Label(„ganz kurz noch...“));
```

```
myWindow.setVisible(true);
```

3. Das Button Element

- Auf einen Klick auf einen Button können wir reagieren.
- Das ist ähnlich aufwendig wie das schließen des Frames/Windows
- Dank lambda (JAVA II) geht es aber auch sehr kompakt:

```
Button button = new Button(„Klick Mich!“);
```

```
button.addActionListener(  
    // fun with java 8 - lambda expression  
    e -> System.out.println(„Danke!“);  
);
```

```
myWindow.add(button);           // nicht vergessen, den Button ins Fenster
```

4. Das TextField Element

- Für Eingaben des Benutzers bei Laufzeit des Programms bietet AWT das *TextField*
- Bei diesem einfachen Beispiel würde bei einem Klick die Eingabe des Benutzers in der Konsole erscheinen:

```
TextField textField = new TextField();  
Button button = new Button(„Klick Mich!“);  
  
button.addActionListener(  
    e -> System.out.println(textField.getText())  
);
```

5. Weitere Grundelemente

- Es gibt weitere Elemente wie
 - **TextArea** eine erweiterte Form von ***TextField***, mit mehr Möglichkeiten
 - **Checkbox** sinnvoll für boolean-Werte
 - **Choice** Ein Dropdown für die Auswahl aus mehreren Möglichkeiten
 - **List** ist quasi ein ausgeklapptes Dropdown, eher eine Auswahlliste
 - ... jede Menge mehr – was den Rahmen der Veranstaltung sprengen würde.
- Des Weiteren können auch Grundformen farbig gezeichnet werden...

*Bei den Code-Beispielen gibt es mehrere Beispiele für die „**List**“...*

6. Das Layout anpassen

- Innerhalb des Frames können **Panels** definiert werden, die als Container für weitere Elemente verwendet werden können:

```
myWindow = new Frame();
```

```
myPanel = new Panel();
```

```
myPanel.add(new Label („Hello“);
```

```
myPanel.add(new Label („World“);
```

```
myWindow.add(myPanel);
```

```
myWindow.setVisible(true);
```

6. Das Layout anpassen

- Um diese anzuordnen, kann ein LayoutManager verwendet werden.
- Ein interessantes (unter mehreren) ist das GridLayout...

```
myPanel = new Panel();
```

```
myPanel.setLayout(new GridLayout(2, 3));
```

- ...das folgendes Layout erschafft:

(Bei der Nutzung von add(); werden Zellen in der angegebenen Reihenfolge befüllt)

1	2	3
4	5	6