

Graphische Oberflächen erstellen mit Java-GUI (Graphical User Interface)

1) AWT, SWT und Swing

AWT:

- plattformunabhängig
- einfache Operationen zum Zeichnen und zur Ereignisbehandlung
- optische Darstellung vom Betriebssystem abhängig

SWT:

- Erweiterung von AWT

Swing:

- zudem unabhängig vom Betriebssystem
- bietet mehr Optionen

2) GUI -Komponenten

Viele GUI komponenten sind als AWT und Swing verfügbar.
Unterschied erkennbar an ihren Namen
Bsp.: java.awt.Button , javax.swing.JButton

3) Typen von Komponenten

„heavyweight“- Komponenten

z.B. Rahmen (JFrame), Applets (JApplet), Dialoge (JDialog), etc.

- Bilden den äußeren Rahmen eines Programms
- Sie können keine anderen heavyweight components enthalten.

„lightweight“- Komponenten

z.B: Schaltflächen (JButton), Beschriftungen (JLabel), Bildlaufleisten (JScrollbar), etc.

- lightweight-Komponenten müssen in heavyweight-Komponenten platziert werden, da sie über kein eigenes Frame verfügen.

4) Befehle/ Methoden:

1) .add():

fügt Komponente direkt zur Hauptkomponente hinzu

2) .getContentPane().add():

manchmal benötigt man eine Zwischenkomponente („content pane“)

3) .setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE):

Zustand, sodass die Anwendung mit einem Klick auf das X sofort beendet wird.

4) .pack():

regelt die Größe des Fensters frame, sodass das Fenster mit seinem Inhalt eine optimale Größe hat

- 5) `.setSize(int width, int height):`
die gewünschte Größe wird manuell angegeben
- 6) `.setVisible(true):`
das Fenster wird mit seinem Inhalt sichtbar gemacht.
- 7) `JFrame("String Titel"):`
erzeugt ein Fenster/Rahmen mit einem Titel
- 8) `JLabel("String Name"):`
erzeugt eine Beschriftung
- 9) `.setForeground(Color.RED):`
Ändert die Schriftfarbe

5. Action Listener

- Swing- Komponenten arbeiten mit Ereignissen
- Dient zur Auslösung eines Ereignisses durch eine Aktion z.B per Maus klick
- Das Programm erkennt ihn durch Action Listener
- Überprüft ihn mit der Methode `actionPerformed()`

6. Beispiel

1) *Den Rahmen Erstellen:*

```
public static void main( String[] args ){  
    JFrame frame = new JFrame("Good And Evil");  
    frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );  
    frame.pack();  
    frame.setVisible( true );  
}
```

2) *Den Button in den Frame erstellen:*

```
final JButton button1 = new JButton("Ende" );  
frame.add( button1 );  
button2.addActionListener( new ActionListener() {  
    public void actionPerformed((ActionEvent e ) {  
        System.exit( 0 );           // mit Klick auf „Ende“ schließt sich dies  
    }  
});
```

6. ImageIcon()

Hiermit wird ein Bild auf dem Button erzeugt:

```
final Icon icon1 = new ImageIcon(".. .png" );
final JButton button1 = new JButton( icon1 );
frame.add( button1 );
ActionListener al = new ActionListener() {
    public void actionPerformed((ActionEvent e) {
        System.exit( 0 );
    }
};
button1.addActionListener( al );
```

- Achtung: die Bilder müssen vorher im Workspace – im jeweiligen ordner – gespeichert werden.
- Nur noch das Fenster und den Button in einer Klasse vereinen und die jeweiligen Packages importieren lassen.