



JAVA

Oberflächen mit Swing

Grafik von <http://javafx.com>



AGENDA



- Grundlagen GUI
- Swing
- JavaFX

GRAPHISCHE BENUTZEROBERFLÄCHEN

- AWT = Abstract Window Toolkit
 - Verwendet die jeweiligen GUI-Komponenten des OS
 - Bietet nur eine Grundmenge von Elementen aller OS
- Swing
 - Swing werden direkt von Java gerendert und ist OS frei
 - Mehr Elemente und bequemere API
 - Oft kein natives OS-Verhalten und etwas langsamer

GRAPHISCHE BENUTZEROBERFLÄCHEN

- JavaFX
 - Ab Java 7 Teil vom JDK
 - Aktuelle Version 8
 - Portabel
 - Direkt von Java gerendert



SWING

- Alle Swing UI Komponenten sind `java.awt.Container` und können andere Komponenten beinhalten
- Ermöglicht ein einheitliches Konzept zum Rendering
- Scope dieses Abschnittes: Grundkonzept kennen

SWING JCOMPONENT

- JComponent ist die Basisklasse aller Swing UI Komponenten
- Erbt von `java.awt.Container` -> Swing-Elemente sind Container
- JComponent unterstützt viele Features
 - ToolTips, Border, Pluggable Look & Feel, Drag & Drop

HALLO SWING

```
package ba.java.swing;

import javax.swing.JFrame;
import javax.swing.JLabel;

public class SwingApp {
    ... public SwingApp() {
        ... JFrame frame = new JFrame("Erste Swing Applikation");
        ... frame.getContentPane().add(new JLabel("Hallo!"));
        ... frame.setSize(400, 200);
        ... // frame.pack(); für automatische Größenberechnung
        ... frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        ... frame.setVisible(true);
    }
    ... public static void main(String[] args) {
        ... new SwingApp();
    }
}
```



SWING KOMPONENTEN

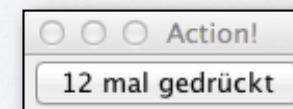
- JFrame
- JComponent
- JPanel
- JScrollPane
- JButton
- JLabel
- JTabbedPane
- JCheckbox
- JComboBox
- JTextField, JTextArea
- JPasswordField
- JSlider, JMenuBar, ...

SWING

EVENTS & LISTENER

- Package java.awt.event bzw. java.swing.event
- java.awt.ActionListener ist die Mutter aller EventListener
- Meist anonyme Klasse, die Interface implementiert

```
public class DrueckMich {  
    ...  
    private int count = 0;  
    private final String GEDRUECKED = "mal gedrückt";  
    ...  
    public DrueckMich() {  
        ... JFrame derFrame = new JFrame("Action!");  
        ... final JButton derButton = new JButton(count + GEDRUECKED);  
        ... derButton.addActionListener(new ActionListener() {  
            ...  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                count++;  
                derButton.setText(count + GEDRUECKED);  
            }  
        });  
        ... derFrame.getContentPane().add(derButton);  
        ... derFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        ... derFrame.pack();  
        ... derFrame.setVisible(true);  
    }  
    ...  
    public static void main(String[] argv) {  
        ... new DrueckMich();  
    }  
}
```



SWING

LISTENER UND ADAPTER

- Für jeden Typ von Listener existiert ein Interface, das den Listener definiert. Namensschema: `<Typ>Listener`
- Dazu gibt es jeweils eine passende Klasse, die die Event Informationen hält. Namensschema: `<Typ>Event`
- Auf Komponenten existiert Methode `add<Typ>Listener(..)`
- Meist existiert eine Adapterklasse, Schema: `<Typ>Adapter`

SWING

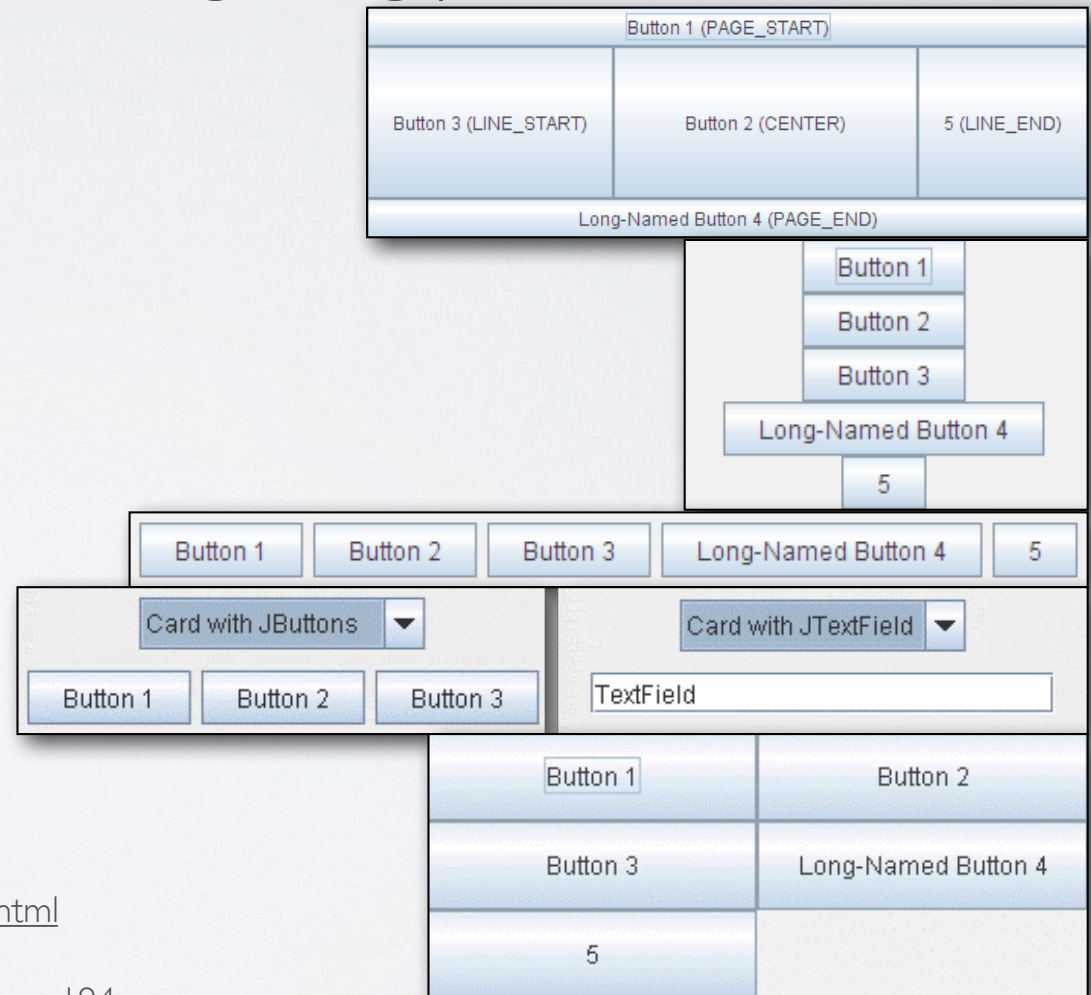
PROMINENTE EVENTLISTENER

- WindowListener
- ActionListener
- KeyListener
- MouseListener
- MouseMotionListener

SWING LAYOUTMANAGER

- `Container#setLayout(LayoutManager mgr)`
- `BorderLayout`
- `BoxLayout`
- `FlowLayout`
- `CardLayout`
- `GridLayout`

<http://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>



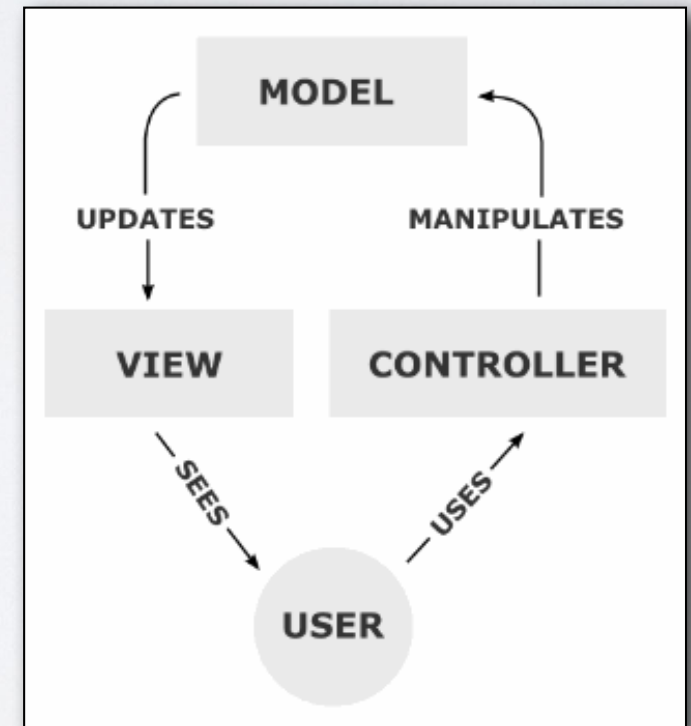
SWING ZEICHNEN

```
public PaintBeispiel() {  
    ... JFrame frame = new JFrame("Malen mit Swing");  
    ... frame.setSize(400, 200);  
    ... JPanel zeichenFläche = new JPanel() {  
        ... // jede Swing.JComponent hat eine methode paint(),  
        ... // welche zum Zeichnen überschrieben werden kann.  
        ... @Override  
        ... public void paint(Graphics g) {  
            ... super.paintComponents(g);  
            ... g.setColor(Color.RED);  
            ... g.drawRect(50, 50, 100, 100);  
            ... g.setColor(Color.BLACK);  
            ... g.fillOval(50, 50, 100, 100);  
            ... }  
    ... };  
    ... frame.getContentPane().add(zeichenFläche);  
    ... frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    ... frame.setVisible(true);  
}
```

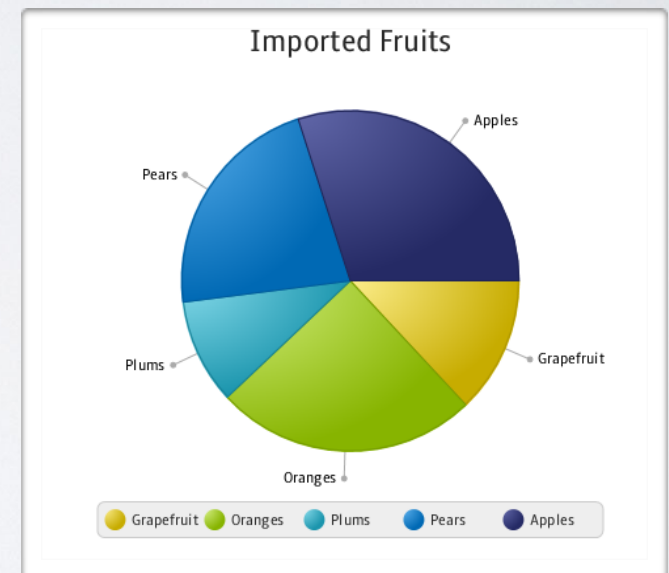


JAVAFX

- Ähnlich zu Swing, aber komfortabler
- Superklasse Node bzw. Control
- MVC
- Sourcen (sind bei Java 8 enthalten)



JAVAFX GUI ELEMENTE



JAVAFX SCENEBuilder

- Grafischer Editor
- .fxml und @FXML
- <http://www.oracle.com/technetwork/java/javafxscenebuilder-1x-archive-2199384.html>

WICHTIGE DINGE ÜBER JAVAFX

- .fxml und @FXML
- javafx.fxml.Initializable
- ObservableList und FXCollections
- Inhalt eines Packages
- Dokumentation ist sehr gut

ZU INSTALLIEREN

- Aktuelles Oracle JDK
- IntelliJ oder [efxclipse.org](https://www.eclipse.org/efxclipse/) -> Vollständige Eclipse-Distribution
- SceneBuilder

HAPPY HACKING

