



Swing Basics

Grundlagen der GUI-Programmierung mit Swing

Daniel Hiller (dbap GmbH, Münster)

- Historie
- Standard-Komponenten
- EDT
- Layout Manager
- Event Listener
- Actions
- SwingWorker
- Usability

- Zunächst AWT
 - Schnittmenge der Komponenten aller Plattformen, für die eine JRE existiert
 - Interface, welches die nativen Komponenten der Plattformen steuert
 - „fette“ Komponenten

- Dann Swing
 - Komponenten sind vollständig in Java erstellt
 - „leichte“ Komponenten
 - Vorteil: volle Kontrolle über das Aussehen und die Menge der Komponenten, z.B.
 - Verwendung des Platform Look And Feels zur Erhaltung des „nativen“ Gefühls
 - Erstellung eines eigenen Look And Feels
 - Diverse Look And Feels verfügbar (z.B. Infonode)

Standard-Komponenten

Existierende Standard-Komponenten in Java 6

Demo

Es kann nur einen geben

Der EventDispatch-Thread und warum es wichtig ist,
Komponenten **nur innerhalb dieses Threads** zu
aktualisieren

Warum?

- Die meisten Swing-Objekt-Methoden sind nicht thread-sicher
- Aufrufe außerhalb des EDT können zu
 - Thread-Interferenz oder
 - Speicher-Inkonsistenzen führen
- Fehler, die durch Zugriffe außerhalb des EDT entstehen, sind
 - schwer zu identifizieren und
 - noch schwerer zu reproduzieren



Demo

InvokeLater oder invokeAndWait

```
SwingUtilities.invokeLater(new Runnable() {  
  
    @Override  
    public void run() {  
        // TODO Manipulation von Komponenten hier  
    }  
});
```

SwingWorker

```
new SwingWorker<Object, Object>() {  
    protected Object doInBackground() throws Exception {  
        // TODO Datenermittlung oder lang andauernde Berechnungen  
        return null;  
    }  
    protected void done() {  
        // TODO Manipulation von Komponenten hier  
    }  
}.execute();
```

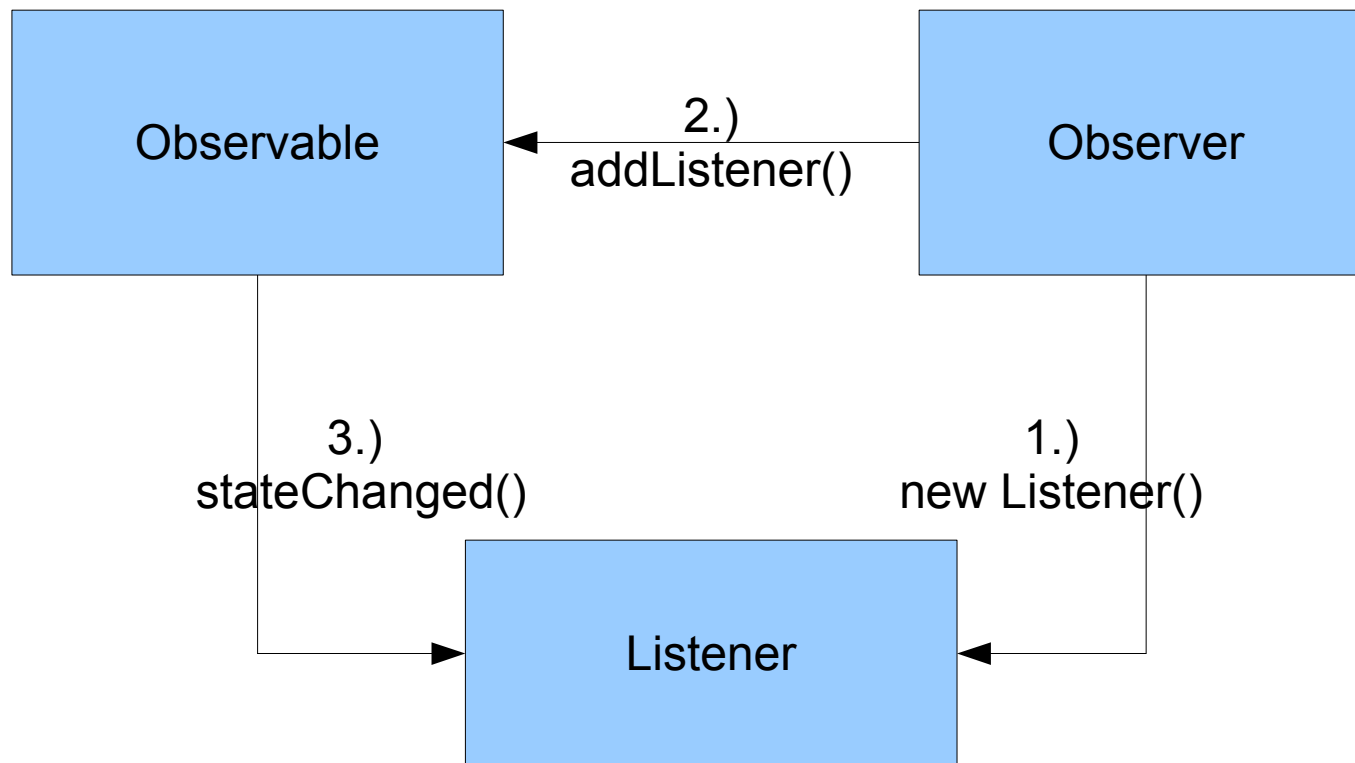
Immer skalierbar bleiben

Automatisches Organisieren von Komponenten durch
Layout Manager

Demo

Wiring it together

Listener zur Verarbeitung von Benutzer-
Ereignissen



```
class Observable {  
    final List<StateListener> listeners = new ArrayList<StateListener>();  
    void addStateListener(StateListener l) {}  
    void removeStateListener(StateListener l) {}  
    void fireStateChanged(StateEvent stateEvent) {}  
}
```



```
class Observer {  
  
    private final String name;  
  
    Observer(String name) {  
        this.name = name;  
    }  
  
    final StateListener myStateListener = new StateListener() {  
  
        @Override  
        public void stateChanged(StateEvent e) {  
            handleStateChanged(e);  
        }  
    };  
  
    void handleStateChanged(StateEvent e) {  
        System.out.println("\n" + name + "\n.handleStateChanged( " + e.toString() + " |");  
        // TODO: Event behandeln  
    }  
}
```

```
final Observable observable = new Observable();

final Observer firstObserver = new Observer("First");
observable.addStateListener(firstObserver.myStateListener);

final Observer secondObserver = new Observer("Second");
observable.addStateListener(secondObserver.myStateListener);

observable.fireStateChanged(new StateEvent("ON"));
observable.fireStateChanged(new StateEvent("OFF"));

observable.removeStateListener(firstObserver.myStateListener);

observable.fireStateChanged(new StateEvent("ON"));
observable.fireStateChanged(new StateEvent("OFF"));

observable.removeStateListener(secondObserver.myStateListener);
```

DEMO

```
public class ComboBoxListenerDemo extends AbstractListenerDemo {

    public ComboBoxListenerDemo() {

    }

    @Override
    protected Component createDemoComponents() {
        final JPanel panel = new JPanel();
        final JComboBox jComboBox = new JComboBox(new DefaultComboBoxModel(
            new String[] { "First", "Second", "Third" }));
        panel.add(jComboBox);

        jComboBox.addItemListener(new ItemListener() {

            @Override
            public void itemStateChanged(ItemEvent arg0) {
                appendEvent(arg0);
            }
        });

        return panel;
    }
}
```

DEMO

Uuuuund – ACTION!

Actions als Startobjekte für Benutzer-Aktionen



DEMO

SwingWorker, der Lastenträger

Daten ermitteln außerhalb des EDT


```
new SwingWorker<Object, Object>() {

    @Override
    protected Object doInBackground() throws Exception {
        // TODO Lang andauernde Berechnung hier
        return null;
    }

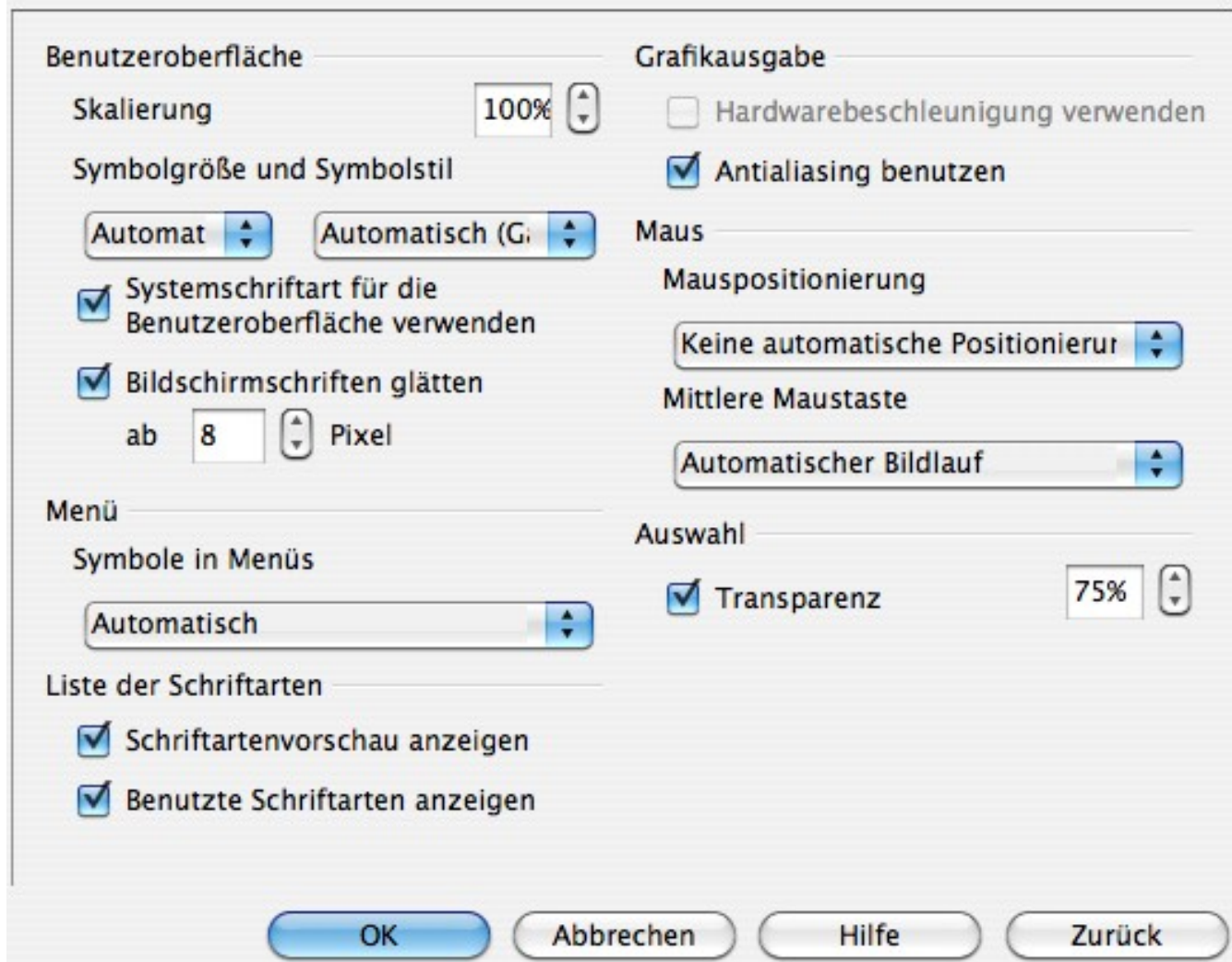
    protected void done() {
        try {
            Object resultFromDoInBackground = get();
        } catch (InterruptedException e) {
            // Thread, der doInBackground() ausgeführt hat, wurde abgebrochen
            return;
        } catch (ExecutionException e) {
            // Es ist während der Ausführung ein Fehler aufgetreten
            e.printStackTrace();
            // TODO Fehlerbehandlung
        }
        // TODO Aktualisierung der GUI hier
    }

}.execute();
```

DEMO

Keep it simple

Usability durch Einfachheit und Integration



The screenshot shows the 'Benutzeroberfläche' (User Interface) settings dialog box. It is divided into several sections with various controls:

- Benutzeroberfläche**
 - Skalierung**: A numeric input field set to '100%' with up/down arrows.
 - Symbolgröße und Symbolstil**: Two dropdown menus, the first set to 'Automat' and the second to 'Automatisch (G...'.
 - ☒ **Systemschriftart für die Benutzeroberfläche verwenden**
 - ☒ **Bildschirmschriften glätten**
 - ab **Pixel**
- Menü**
 - Symbole in Menüs**: A dropdown menu set to 'Automatisch'.
- Liste der Schriftarten**
 - ☒ **Schriftartenvorschau anzeigen**
 - ☒ **Benutzte Schriftarten anzeigen**
- Grafikausgabe**
 - ☐ **Hardwarebeschleunigung verwenden**
 - ☒ **Antialiasing benutzen**
- Maus**
 - Mauspositionierung**: A dropdown menu set to 'Keine automatische Positionierung'.
 - Mittlere Maustaste**: A dropdown menu set to 'Automatischer Bildlauf'.
- Auswahl**
 - ☒ **Transparenz** with up/down arrows.

At the bottom, there are four buttons: **OK**, **Abbrechen**, **Hilfe**, and **Zurück**.

- Wenige Elemente
- Plattform-Besonderheiten
 - z.B. OS X
 - Menüs am oberen Bildschirmrand
 - Einstellungs-Dialog immer direkt im ersten Untermenü (!= Extras/Optionen bei Windows)
- Tastatur-Bedienbarkeit

Anschrift

Firma	<input type="text" value="Max Mustermann"/>		
Vor-/Name/Kürzel	<input type="text" value="Max"/>	<input type="text" value="Mustermann"/>	<input type="text" value="IM"/>
Straße	<input type="text" value="Teststraße 123"/>		
PLZ/Ort	<input type="text" value="12345"/>	<input type="text" value="Berlin"/>	
Land/Region	<input type="text"/>		
Titel/Position	<input type="text"/>	<input type="text"/>	
Tel. (Priv/Ge.)	<input type="text"/>	<input type="text"/>	
Fax/E-Mail	<input type="text"/>	<input type="text"/>	

Danke für's Zuhören!

- <http://download.oracle.com/javase/tutorial/uiswing/index.html>
- <http://download.oracle.com/javase/tutorial/uiswing/concurrency/dispatch.html>
- <http://download.oracle.com/javase/tutorial/uiswing/layout/visual.html>
- <http://download.oracle.com/javase/6/docs/api/javax/swing/SwingWorker.html>
- <http://www.infonode.net>
- <http://download.java.net/javadesktop/swingset3/SwingSet3.jnlp>
- <http://www.miglayout.com/>
- <http://www.jgoodies.com/>
- <http://www.dbap.de>