

Java - Part 5

Katja Wegner

katja.wegner@eml-r.villa-bosch.de

Überblick

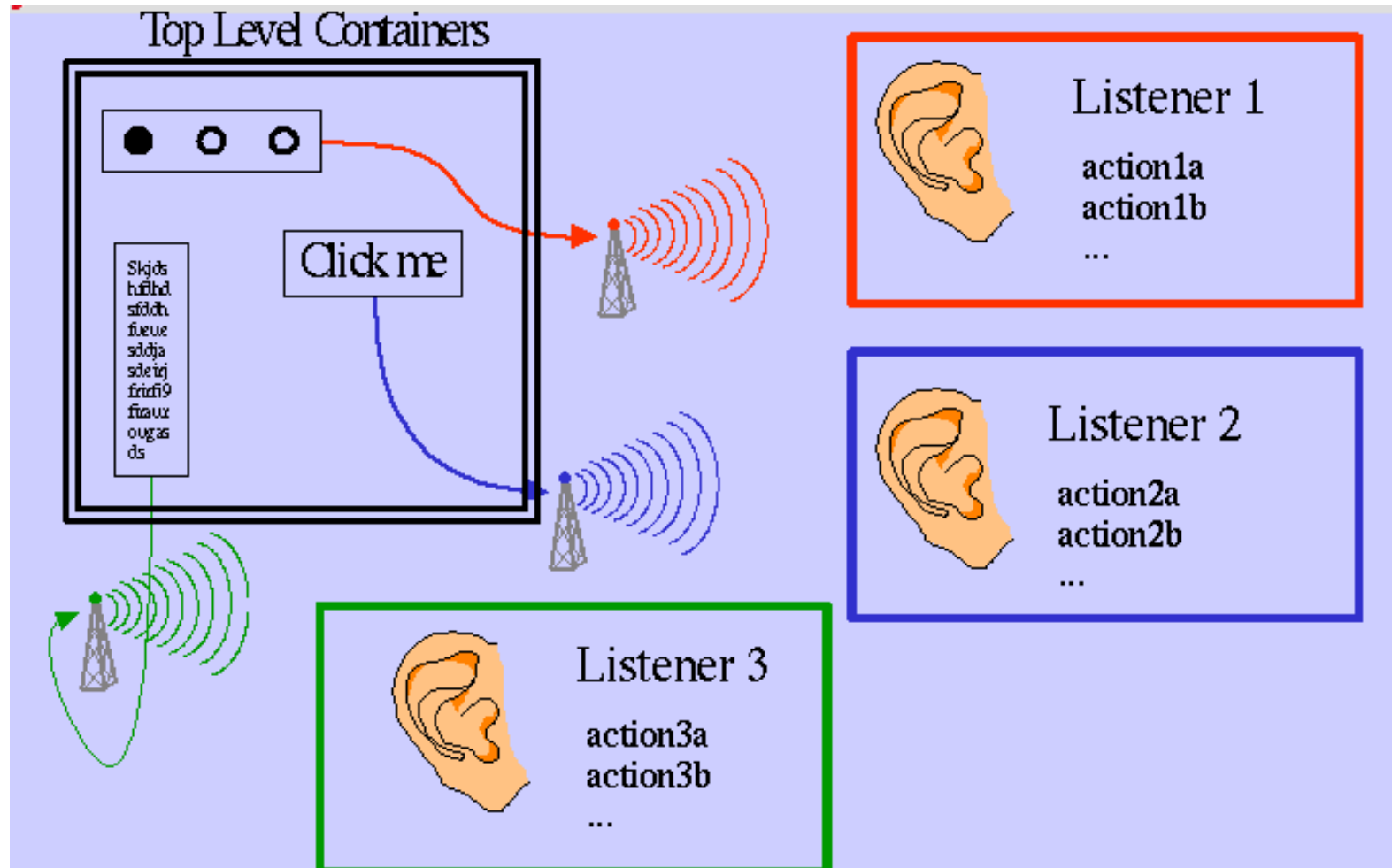
⇒ Swing

- Was ist Swing?
- Komponenten
- Praktische Anwendung:
 - * Hinzufügen von Komponenten in ein Fenster
 - * Layout Manager
 - * Event listeners

Was ist Swing?

- ⇒ Grafikbibliothek
- ⇒ Komponenten zum Bilden von grafischen Benutzerschnittstellen
 - Fenster, Knöpfe, Menüs
 - Event listeners zum Behandeln von Benutzereingaben
- ⇒ Java 2D: Hinzufügen von 2D Grafiken, Bildern oder Text
- ⇒ Drag & Drop Unterstützung

Hauptprinzip

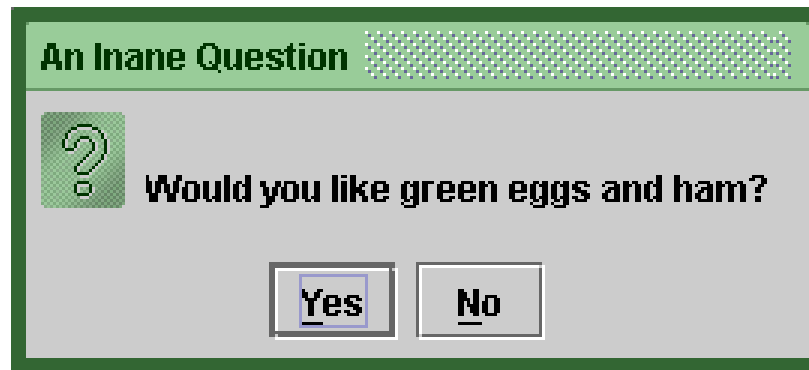


Top Level Container

⇒ Container, die immer an der obersten Stelle der Swing Hierarchie stehen



Applet

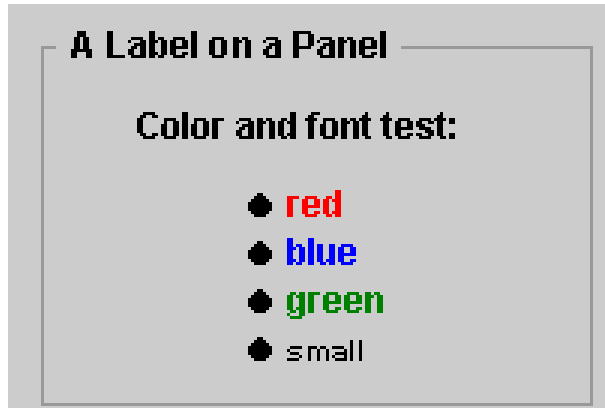


Dialogfenster
(JDialog)



Hauptfenster
(JFrame)

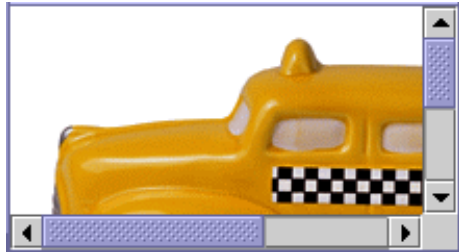
Zwischenlevel Container



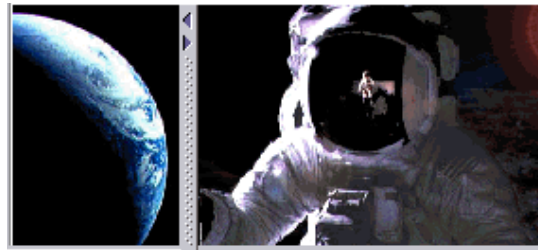
JPanel



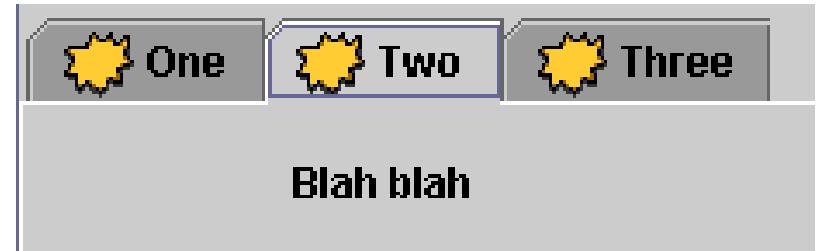
JToolBar



JScrollPane



JSplitPane



JTabbedPane

Kontrollelemente



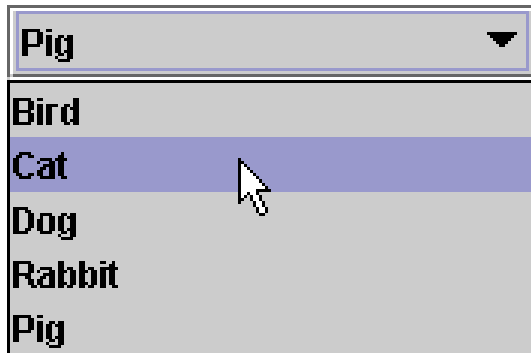
Knöpfe (JButton)



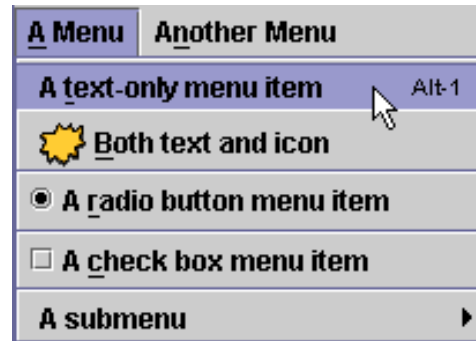
Schieber (JSlider)



Listen (JList)



Auswahlbox (JComboBox)



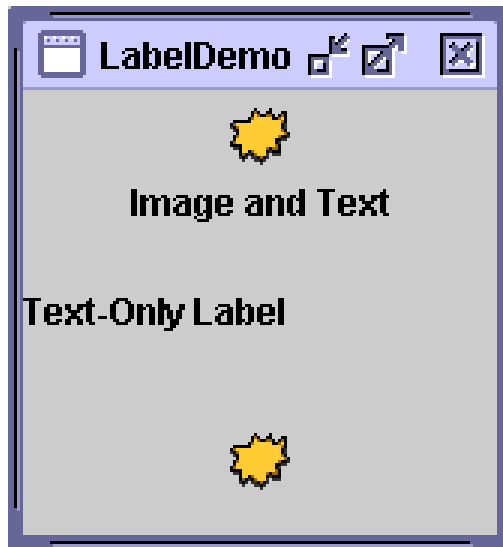
Menü (JMenu)



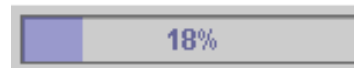
Textfeld (JTextField)

Unveränderliche Komponenten

⇒ Präsentieren von Informationen (one way)



Textausgabe (JLabel)



Prozessverlauf
(JProgressBar)



Hinweis (JToolTip)

Interaktive Komponenten

⇒ Formatierte Ausgaben, die durch Benutzer verändert werden können

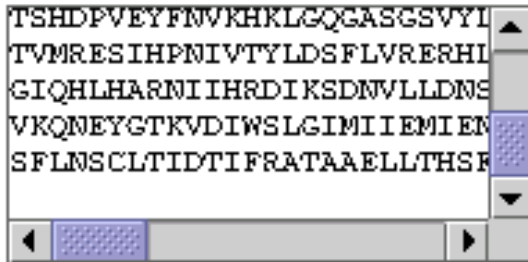


Dateidialog (JFileChooser)



Farbauswahl (JColorChooser)

Interaktive Komponenten (2)



Textfeld (JTextArea)




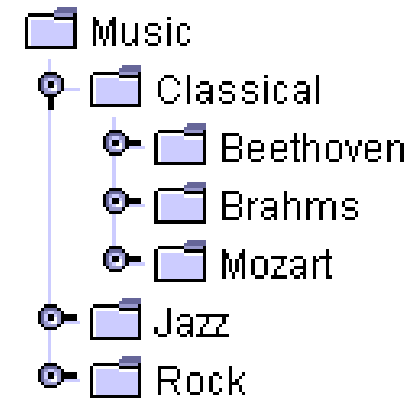
First Name	Last Name	Favorite Food
Jeff	Dinkins	
Ewan	Dinkins	
Amy	Fowler	
Hania	Gajewska	
David	Geary	

Tabelle (JTable)



Baumstruktur (JTree)

Basisprogrammstruktur

```
import javax.swing.*;

public class MyGui extends JFrame {
    public MyGui() {
        super(''Name meiner GUI ''); // call constructor of JFrame
    }

    public static void main(String[] args) {
        MyGui gui = new MyGui(); // Objekt von MyGui
        gui.initFrame(); // Aufruf der Initialisierungsmethode
    }

    public void initFrame() {
        ... // Komponenten hinzufügen
        this.setVisible(true); // Anzeigen der Oberfläche
    }
}
```

Hinzufügen von Komponenten

```
import java.awt.Container;

...

public initFrame() {
    Container contentPane = this.getContentPane();
    contentPane.setLayout(new FlowLayout());

    JLabel label = new JLabel("Test");           // Erzeugt Textausgabe
    JButton button = new JButton("Click me");    // Erzeugt Knopf
    JTextField text = new JTextField("Edit me"); // Erzeugt Texteingabefeld

    contentPane.add(label);                       // fügt Textausgabe hinzu
    contentPane.add(button);                      // fügt Knopf hinzu
    contentPane.add(text);                       // fügt Texteingabefeld hinzu
    this.setBounds(50, 50, 400, 200);           // legt Größe des Fensters fest
    ...
}
```

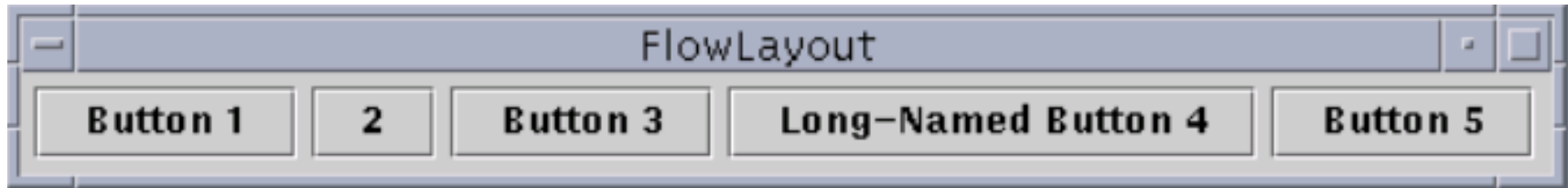
Layout Manager

⇒ Bestimmen Position der Komponenten in einem Container

⇒ Layout Klasse: ([AWT](#))

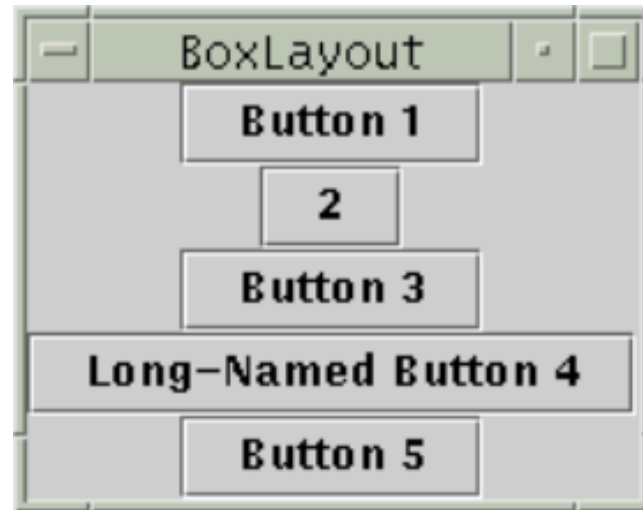
- FlowLayout
- BorderLayout
- GridLayout
- GridBagLayout

FlowLayout



```
Container contentPane = getContentPane();  
contentPane.setLayout(new FlowLayout);  
  
contentPane.add(new JButton("Button 1"));  
contentPane.add(new JButton("2"));  
contentPane.add(new JButton("Button 3"));  
contentPane.add(new JButton("Long-Named Button 4"));  
contentPane.add(new JButton("Button 5"));
```

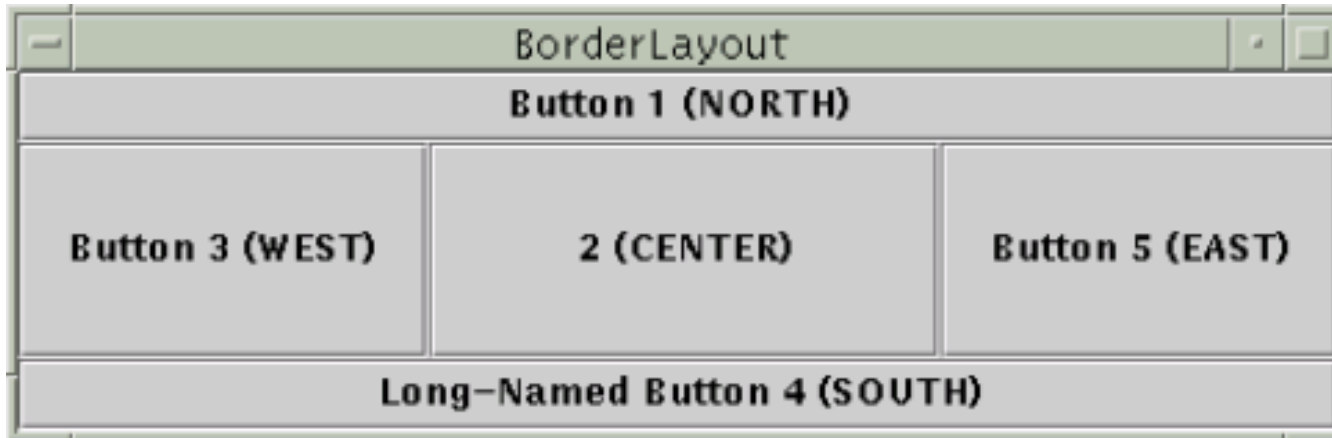
BoxLayout



```
Container contentPane = getContentPane();
contentPane.setLayout(new BorderLayout(contentPane, BorderLayout.Y_AXIS));

contentPane.add(new JButton("Button 1"));
contentPane.add(new JButton("2"));
contentPane.add(new JButton("Button 3"));
contentPane.add(new JButton("Long-Named Button 4"));
contentPane.add(new JButton("Button 5"));
```

BorderLayout



```
Container contentPane = getContentPane();  
contentPane.setLayout(new BorderLayout());
```

```
contentPane.add(new JButton("Button 1 (North)"), BorderLayout.NORTH);  
contentPane.add(new JButton("2 (Center)"), BorderLayout.CENTER);  
contentPane.add(new JButton("Button 3 (West)"), BorderLayout.WEST);  
contentPane.add(new JButton("B4 (South)"), BorderLayout.SOUTH);  
contentPane.add(new JButton("Button 5 (East)"), BorderLayout.EAST);
```

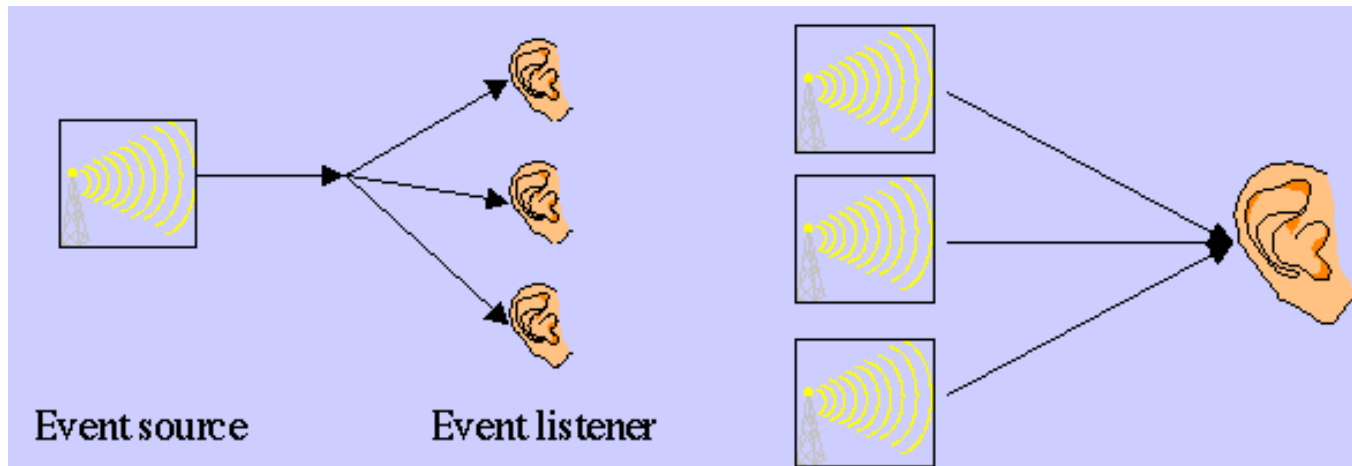

Event Listener

⇒ wird einer Komponente hinzugefügt

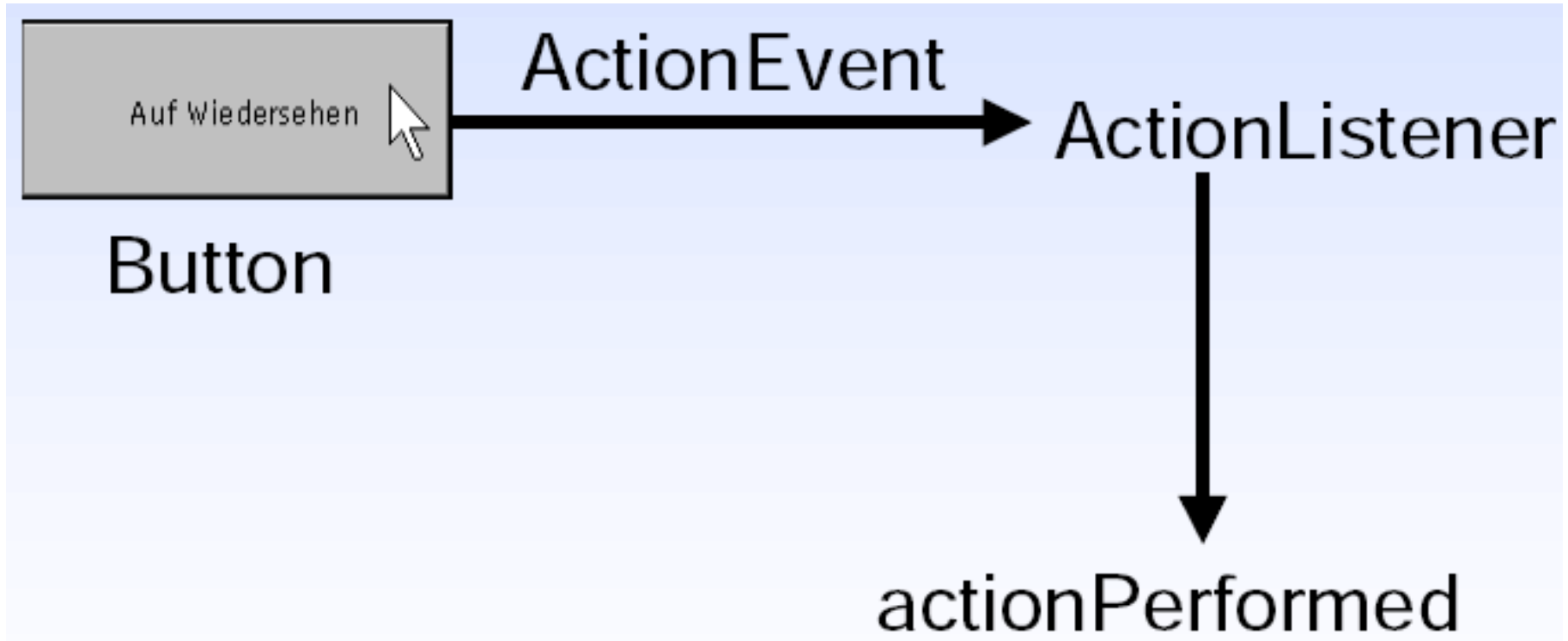
⇒ wartet auf Ereignisse von dieser Komponente, z.B.

- mouse events
- window events

⇒ aufgrund dieser Ereignisse können Anweisungen ausgeführt werden



Ereignis-Behandlung am Beispiel Button-Klick



Listener - Beispiele

Aktion, die das Ereignis auslöst	Listener Typ
Benutzer drückt Knopf oder Return nach der Eingabe in ein Textfeld oder nach Auswahl eines Menüpunktes	<i>ActionListener</i>
Benutzer schließt ein Fenster	<i>WindowListener</i>
Benutzer drückt Mausknopf während der Mauszeiger sich über einer Komponente befindet	<i>MouseListener</i>
Benutzer bewegt die Maus über eine Komponente	<i>MouseMotionListener</i>
Komponente wird angezeigt	<i>ComponentListener</i>
Komponente bekommt Tastaturfokus	<i>FocusListener</i>
Auswahl einer Tabelle oder Liste wird geändert	<i>ListSelectionListener</i>

Listener - Beispiele (2)

Komponente	Event	Listener
Window	WindowEvent	WindowListener
TextField, TextArea	TextEvent	TextListener
Checkbox, Choice, List	ItemEvent	ItemListener
Button, TextField, List, MenuItem	ActionEvent	ActionListener
Component	ComponentEvent, FocusEvent, KeyEvent, MouseEvent	ComponentListener, FocusListener, KeyListener, MouseListener, MouseMotionListener

ActionListener - Beispiel

```
public void initFrame() {  
    Container contentPane = getContentPane();  
    contentPane.setLayout(new BorderLayout(contentPane, BorderLayout.Y_AXIS));  
  
    JButton b1 = new JButton("Button 1");  
    JButton b2 = new JButton("2");  
  
    b1.setActionCommand("but 1");  
    b2.setActionCommand("but 2");  
  
    MyButtonListener myButLis = new MyButtonListener();  
    b1.addActionListener(myButLis);  
    b2.addActionListener(myButLis);  
  
    contentPane.add(b1);  
    contentPane.add(b2);  
}
```

ActionListener - Beispiel (2)

```
private class MyButtonListener implements ActionListener {
    String s = '';

    public MyButtonListener() { }

    public void actionPerformed(ActionEvent e) {
        String entry = e.getActionCommand();

        if (entry.equals("but 1"))
            s = "button 1";
        else if (entry.equals("but 2"))
            s = "button 2";

        System.out.println(s);
    }
}
```

WindowListener - Beispiel

```
import java.awt.event.*;

public void initFrame() {
    ...
    this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    this.addWindowListener(new WindowAdapter() {
        public void windowClosed(WindowEvent e) {
            System.exit(0);
        }
    });
}
```

5.1. Übung

Implementieren Sie eine Oberfläche *MyGui* mit fünf Knöpfe. Probieren Sie verschiedenen Layouts aus (Flow / Box / BorderLayout).

Fügen Sie ein *JPanel* hinzu, das drei *JLabel* und drei *JButton* enthält.

Fügen Sie den drei Knöpfen einen Listener hinzu, der folgendes tun soll:

Wenn der erste (zweite/dritte) Knopf gedrückt wurde, soll das Label des zweiten (ersten/ersten) und des dritten (dritten/zweiten) ausgegeben werden.

Tipps:

Der Inhalt eines *JLabel label* kann mit `label.setText("xxx")` geändert werden. Ein *JPanel panel* wird mit `new JPanel()` erzeugt und kann ebenso mit `add` dem *ContentPane* hinzugefügt werden.

5.2. Übung

Erzeugen Sie ein Fenster *MeineGeheimnisse*, das ein *JLabel* "Authentifizierung" enthält, ein *JPanel* und ein *JPasswordField* enthält. Definieren Sie eine Feld vom Typ *String*, das Name/Passwort Einträge enthält. Es soll mindestens eine Größe von drei Einträgen haben. Fügen Sie einen *ActionListener* für *JPasswordField* hinzu. Dieser liest den eingegebenen Text der beiden Textfelder und wenn ein passendes Paar im Feld gefunden wurde, wird ein neues *JLabel* mit "geheimen Informationen" und ein *JButton* *remove* mit dem Namen "Entfernen" dem Fenster hinzugefügt. Wenn dieser Knopf gedrückt wird, werden das Label und der Knopf wieder entfernt. Ausserdem sollte in diesem Fall das Passwortfeld gelöscht werden.

Tipps:

Hinzufügen einer Komponente: `contentPane.add(c);` und entfernen einer Komponente: `contentPane.remove(c);`

Lesen des Inhalts eines Textfeldes: `getText();` und des Passwortfeldes: `getPassword();` (gibt `char[]` zurück → benutzen Sie `new String(passwordField.getPassword())`)

Swing Tutorial

<http://java.sun.com/docs/books/tutorial/uiswing/>