

7. Überblick über die Java Foundation Classes

7.1 Begriffe

Widget / Widget Class / GUI-Objekt

- „Window Gadget“ - allgemein verwendeter Begriff für abgeschlossene, interaktive Graphikobjekte
- analog verwendete Begriffe: „Component“, „GUI-Objekt“

Widget Set

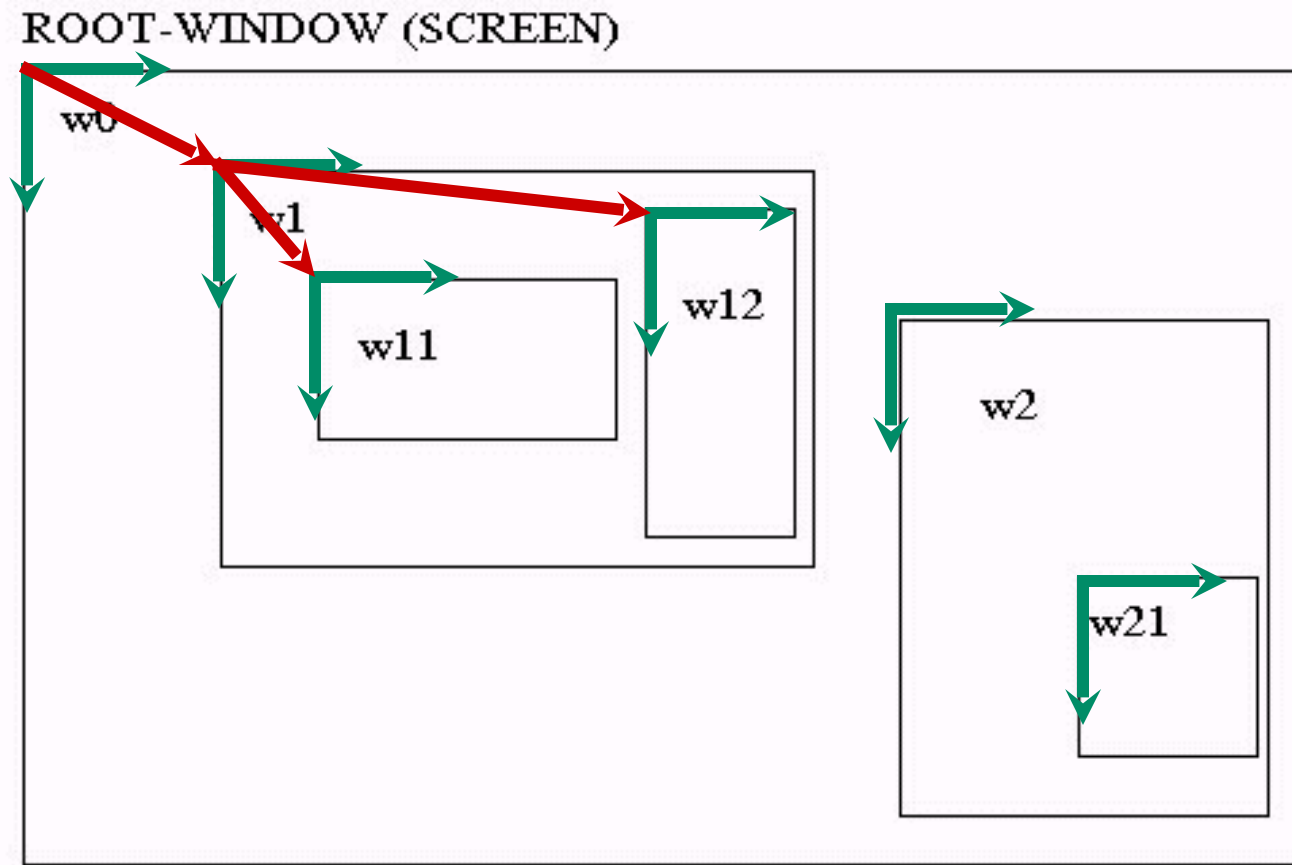
- GUI-Bibliothek mit gemeinsamem, einheitlichen L&F (look and feel)
- Gebräuchliche GUI's: Windows-Produkte, Java Swing (oder JFC), X/Intrinsics Widget Sets, z.B. Motif (eher aussterbend)

Koordinatensysteme

- analog zu Window-Koordinaten hat jede Komponente ein eigenes Koordinatensystem

7.1 Begriffe

Widget- und Window-Koordinatensysteme



7.1 Begriffe

L&F, Look & Feel:

- Art der Präsentation (Look) und
- Art der Interaktion (Feel)

L&F beinhaltet:

- Graphische Präsentation des Zustands und/oder der Aktion
- Interne Zustände
- Internes Ereignismanagement
- Externe Ereignisse („Actions“/„Callbacks“)

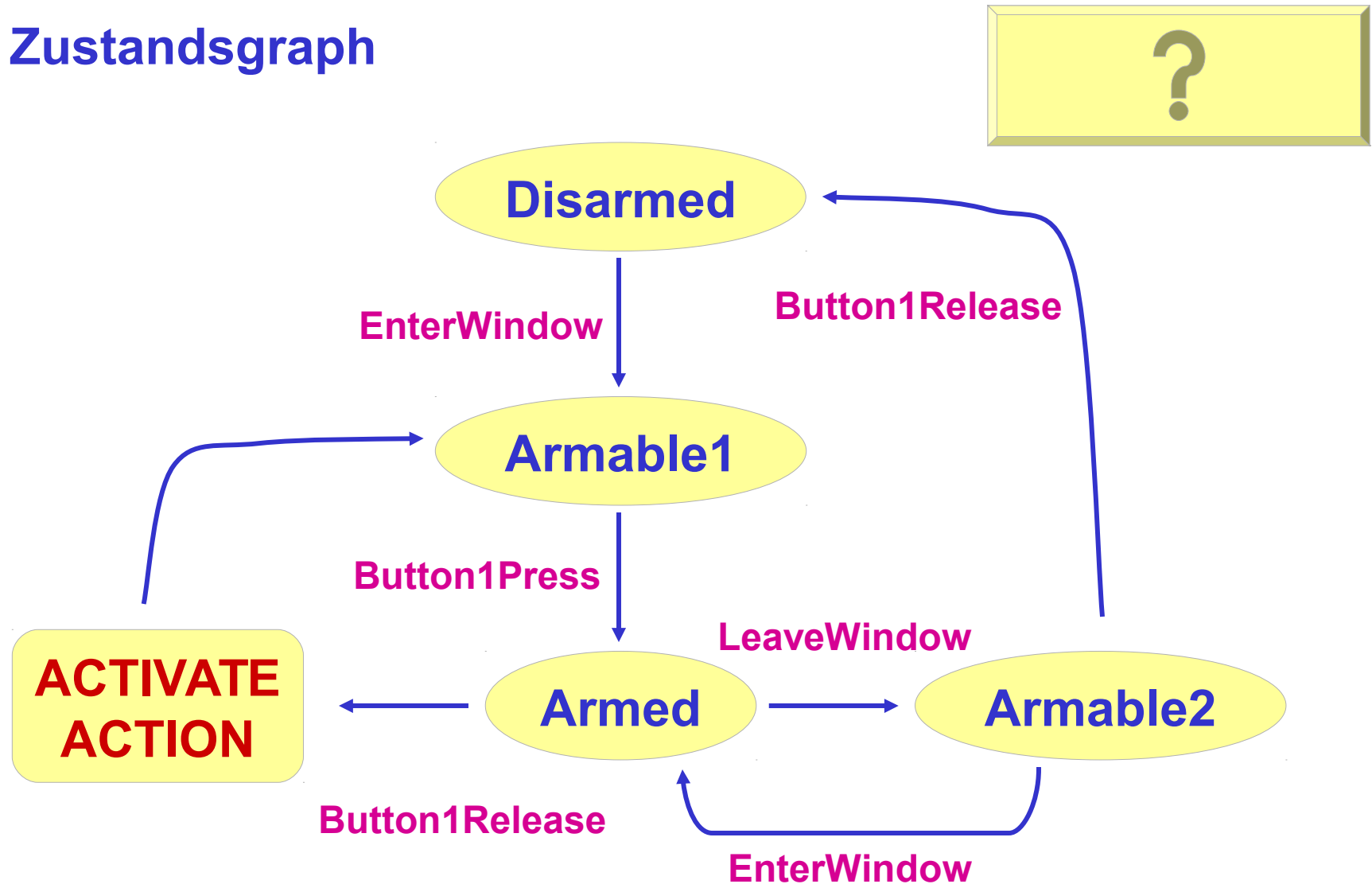
7.2 Look & Feel

Beispiel PushButton in Powerpoint:

- Präsentation Beschriftung, Rahmen
Darstellung abhängig vom Zustand, Cursor abhängig von Mouse-Position
- Zustände: ARMED, DISARMED
- Interne Ereignisse: ButtonPress, ButtonRelease, WindowEnter, WindowLeave
- Externe Ereignisse: ACTIVATE ACTION
(Aktionen)



Zustandsgraph



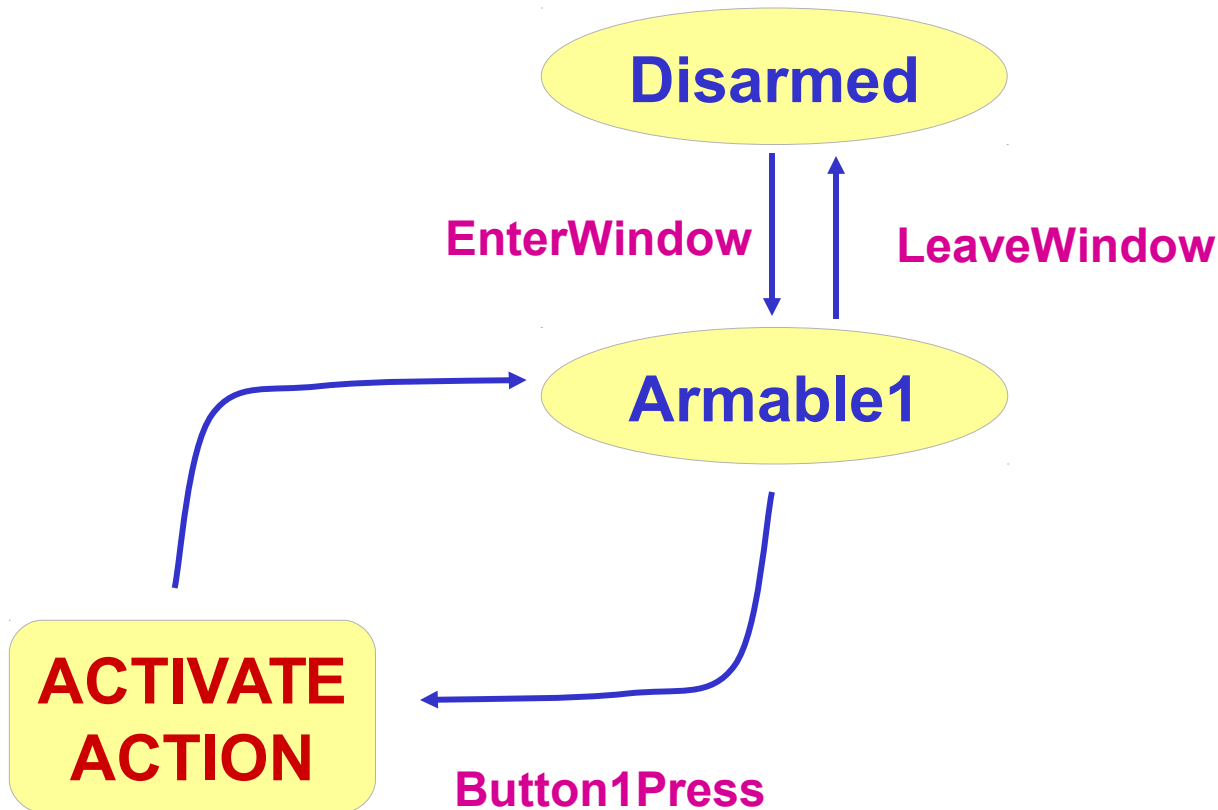
7.2 Look & Feel

Anforderungen an L&F / UI-Toolkits

- möglichst intuitiv
- möglichst leicht zu merken / wieder zu erkennen
- Konsistenz der Ereignisse innerhalb des Widgets
- Konsistenz innerhalb eines Widget Sets (sowohl in der Präsentation als auch in der Interaktion)
- Unterstützung unterschiedlicher Eingabe-Mechanismen (Tastatur/Mouse)
- Aufgabenangepaßtheit
- Minimalismus
- Feedback (visuell/Audio)
- Anbieten sinnvoller Default-Werte durch das Toolkit (z.B. bei Farbwahl)

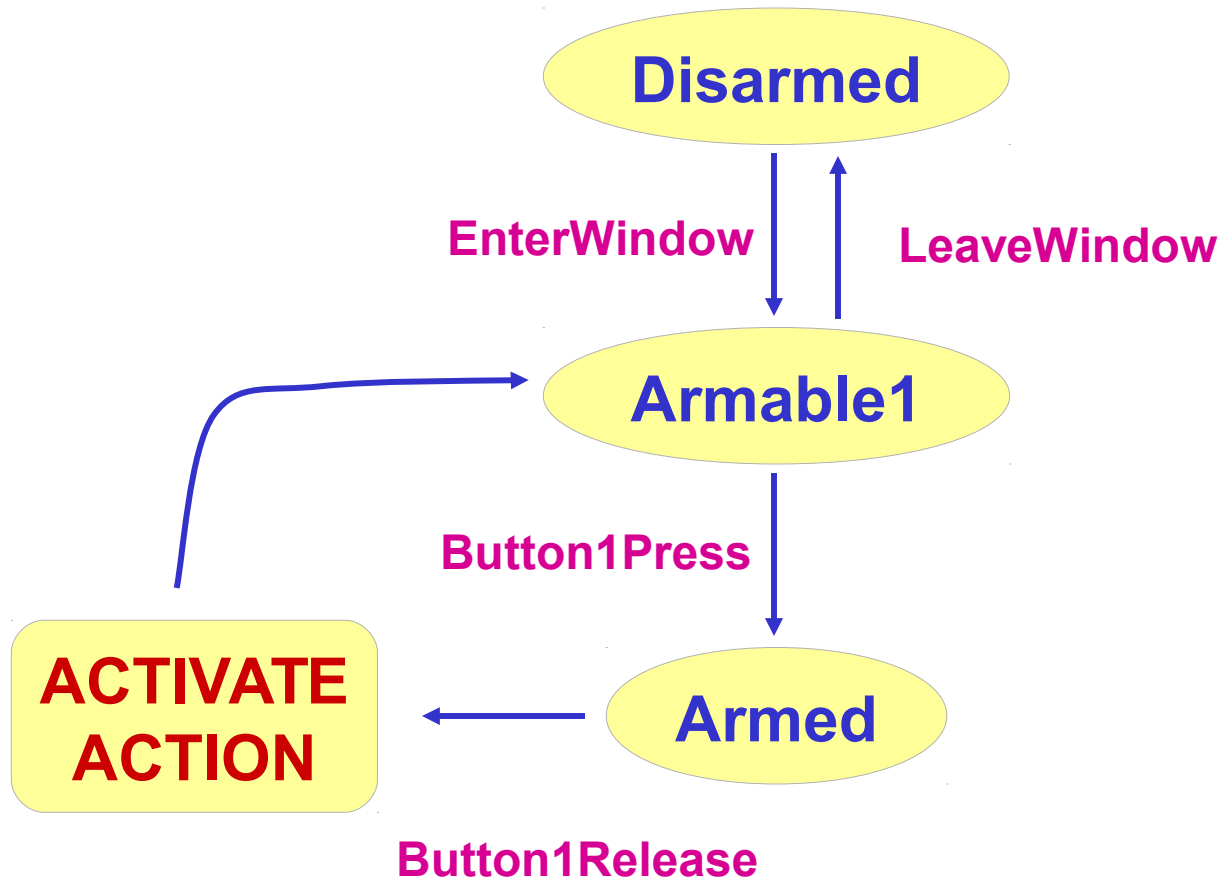
Frage: Warum nicht direkt Activate?

?



Frage: Warum der rechte Zweig?

?



Beispiel

- rechte Maustaste bei ARMED bzw. bei DISARMED
- rechte Maustaste bei DISARMED, dann Window verlassen, dann linke Maustaste
- d.h. rechte Maustaste bringt in einem Fall ein Kontextmenü, im anderen Fall ein Cancel der linken Maustaste



7.3 Ereignisse

Jedes Toolkit verwendet leicht unterschiedliche Begriffe:

<u>Begriff</u>	<u>Xwindows/Motif</u>	<u>Java</u>
Ereignis (Window)	Event	Event
Internes Ereignis	Action	-
Externes Ereignis (Aktion)	Callback Reason	Action
Aufgerufene Funktion	Callback	Listener

7.3 Ereignisse

Viele Komponentenklassen besitzen

- mehrere wichtige Ereignisse/Aktionen
- von denen man oft nicht alle benötigt

Diese können implementiert sein als

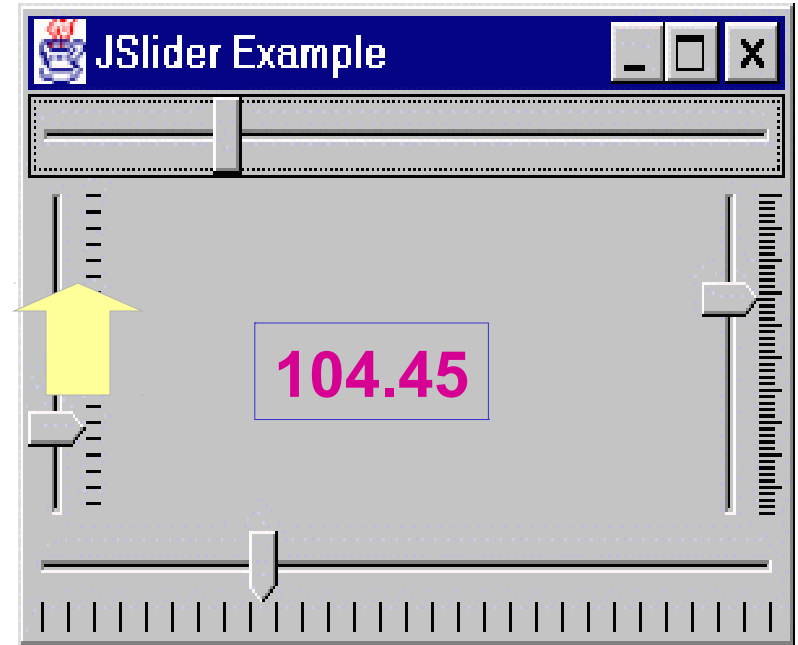
- unterschiedliche Aktionen oder
- unterschiedliche Methoden eines Listeners
- unterschiedliche Methoden eines MVC-Models

Mehrere Listener/Callbacks können sich auf das gleiche Ereignis beziehen.

7.3 Ereignisse

Beispiel: Slider

Aktion:
neue Größe
einstellen



7.3 Ereignisse

Beispiel: Slider

Aktion:

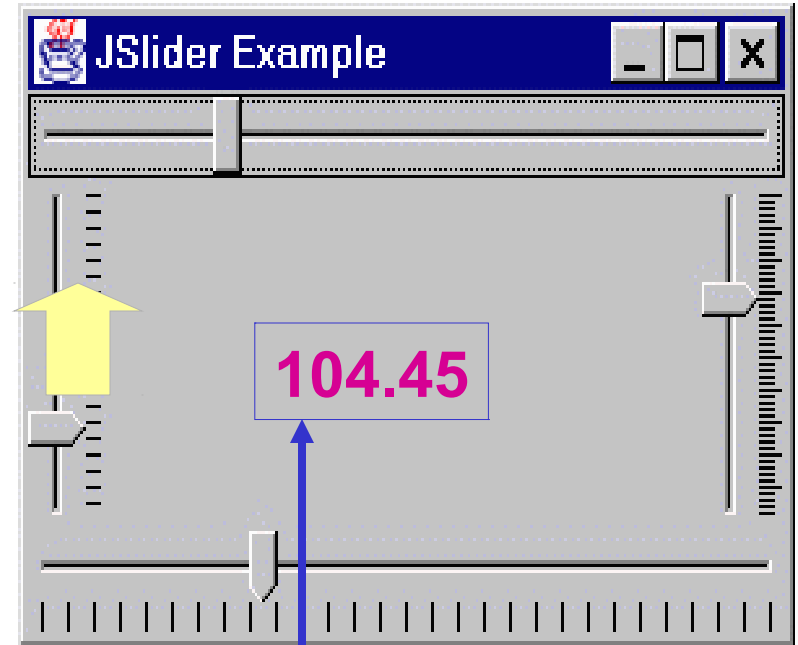
neue Größe
einstellen

EndMoveSlider

Aktion:

Graphisches
Feedback in
Textfeld

MoveSlider



7.3 Ereignisse

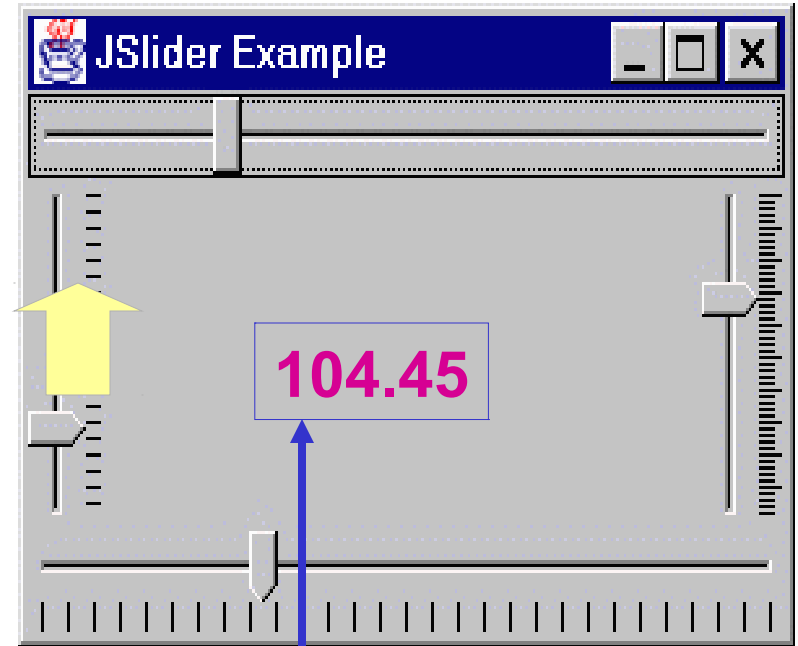
Ein Listener

Aktion:

neue Größe
einstellen

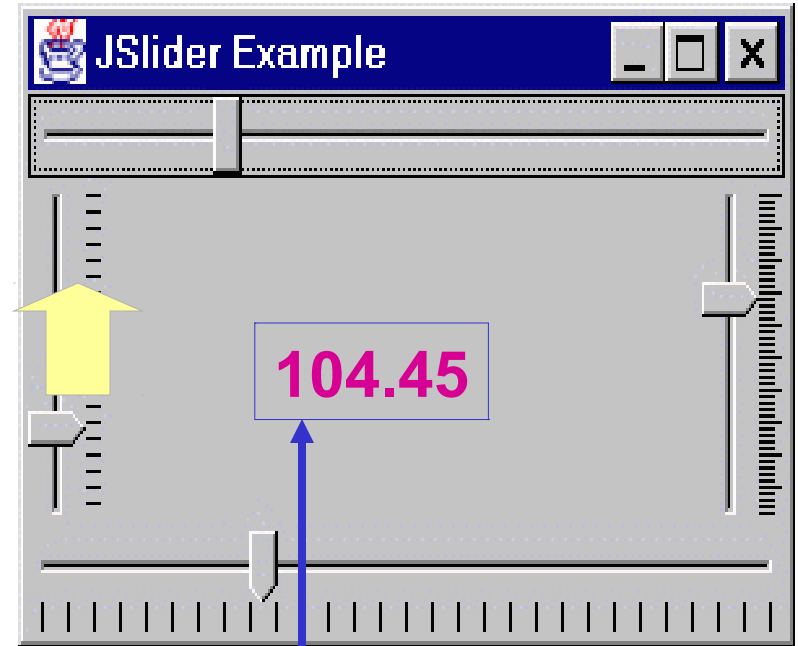
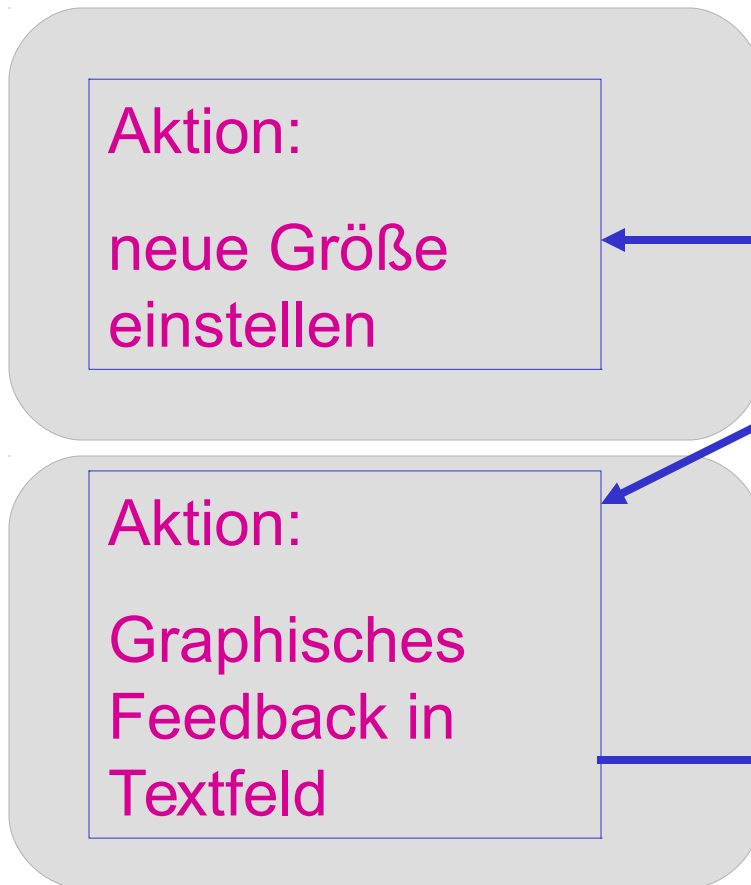
Aktion:

Graphisches
Feedback in
Textfeld



7.3 Ereignisse

Zwei Listener



7.4 Java Foundation Classes

JFC (Swing)

- Seit JDK 1.1.x eingeführt als separates Package (com.sun...)
- Seit JDK 1.2 enthalten (javax.swing...)

Eigenschaften

- plattformunabhängig
- pure Java
- lightweight
- implementiert MVC-Paradigma (Model View Controller)
- pluggable L&F
- leicht zu internationalisieren
- umfangreiche Sammlung von Managern

7.4 Java Foundation Classes

Erinnerung: Klassenhierarchie der AWT-Klassen

`java.lang.Object`

|

+-- `java.awt.Component`

allgemeine Eigenschaften

|

+-- `java.awt.Container`

kann Kinder besitzen

|

+-- `java.awt.Window`

Toplevel ohne Dekoration

|

|

| + `java.awt.Frame`

Toplevel mit Dekoration

|

+-- `java.awt.Panel`

Container mit Layout

|

+ `java.awt.Applet`

Objekt für Browser

7.4 Java Foundation Classes

```
java.awt.Component           allgemeine Eigenschaften
+-- java.awt.Container       kann Kinder besitzen
    |
    +-- javax.swing.JComponent Swing Metaklasse
        |
        +-- javax.swing.JPanel  Swing Container
            |
            +-- java.awt.Window  Toplevel ohne Dekoration
                |
                +-- javax.swing.JWindow Swing Toplevel
                    |
                    +-- java.awt.Frame  Toplevel mit Dekoration
                        |
                        +-- javax.swing.JFrame Swing Toplevel
                            |
                            +-- java.awt.Panel      Container mit Layout
                                +-- java.applet.Applet      Applet
                                    +-- javax.swing.JApplet  Swing-Applet
```

Abgrenzung AWT/JFC

AWT

verwendet native Graphikobjekte
des Windowsystems

die meisten Klasse haben
einen nativen peer

heavyweight

immer opaque (deckend)

JFC/Swing

verwendet nur sehr
wenige native Objekte

die meisten Objekte haben
keinen nativen peer

lightweight

transparenz möglich

7.4 Java Foundation Classes

AWT Button

AWT Button

**Window System
Independence**

„Windows“ Button

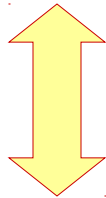
„Motif“ Button

„Windows“ Window

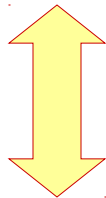
„XWindows“ Window

7.4 Java Foundation Classes

AWT Button



Native Button



Native Window

AWT-Button Eigenschaften:

- Font
- Foreground
- ...

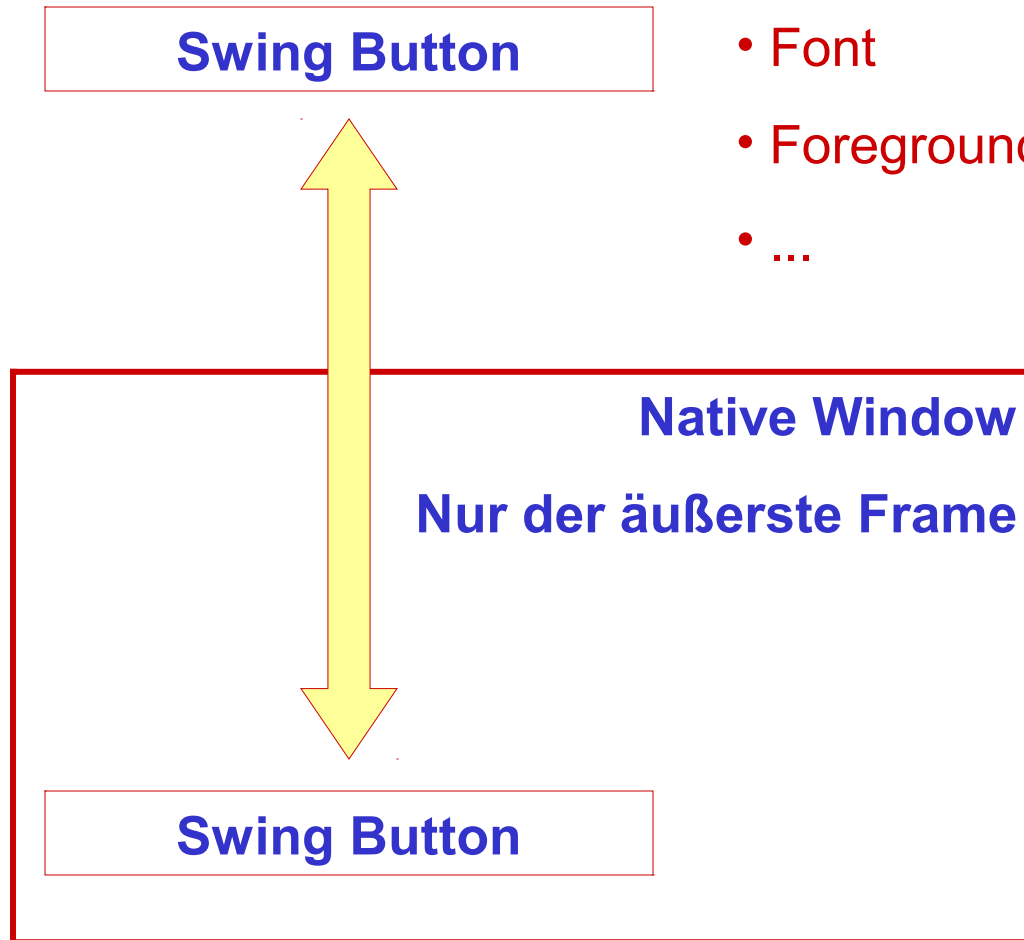
Windows-Button Eigenschaften:

- Font
- Foreground
- ...

7.4 Java Foundation Classes

Swing-Button Eigenschaften:

- Font
- Foreground
- ...

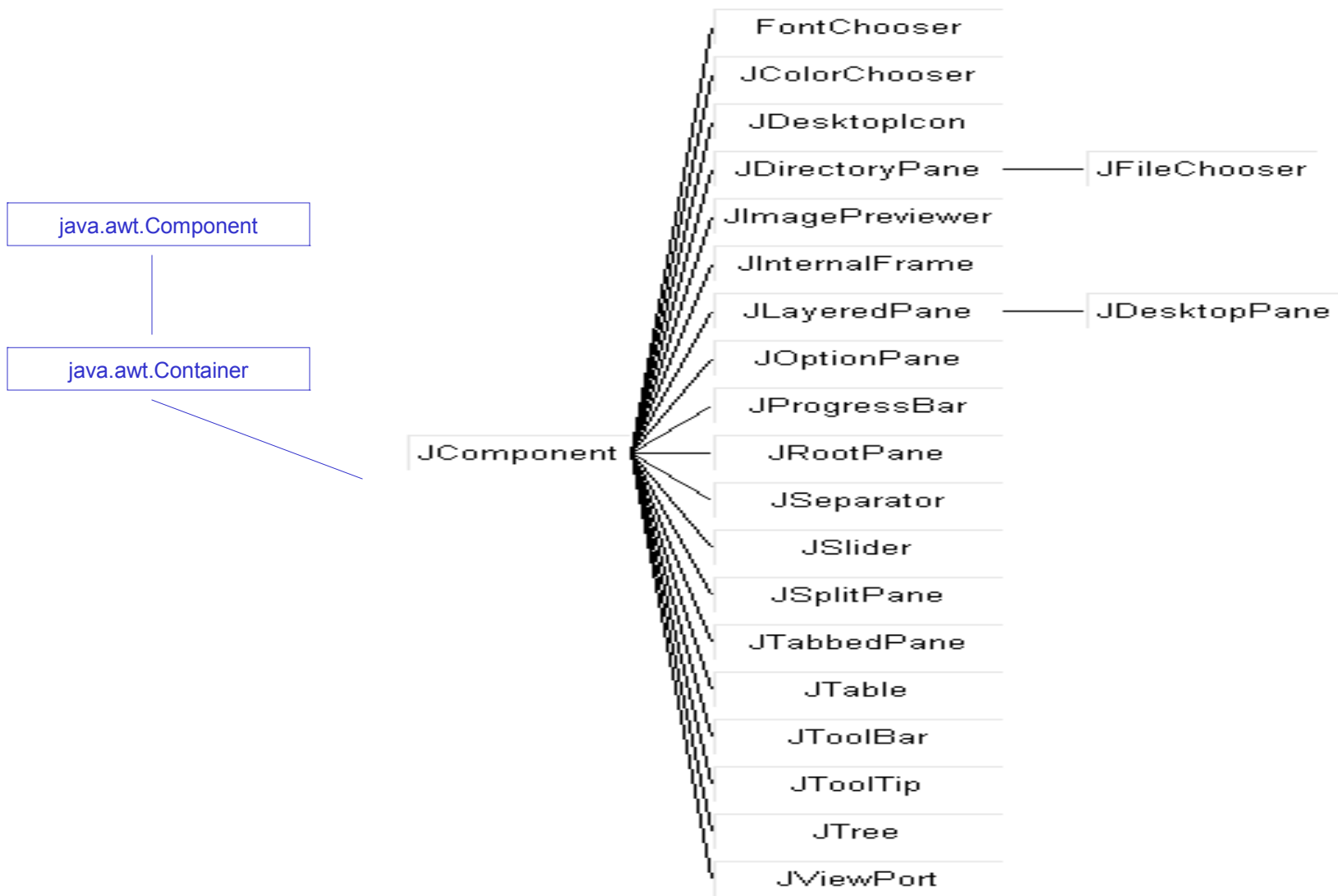


Softwaretechnik 2



724

7.4 Java Foundation Classes



7.4 Java Foundation Classes

SwingSet Demo

- Überblick über Swing-Klassen und deren Eigenschaften
- Siehe
 - jdk1.x/demo/jfc