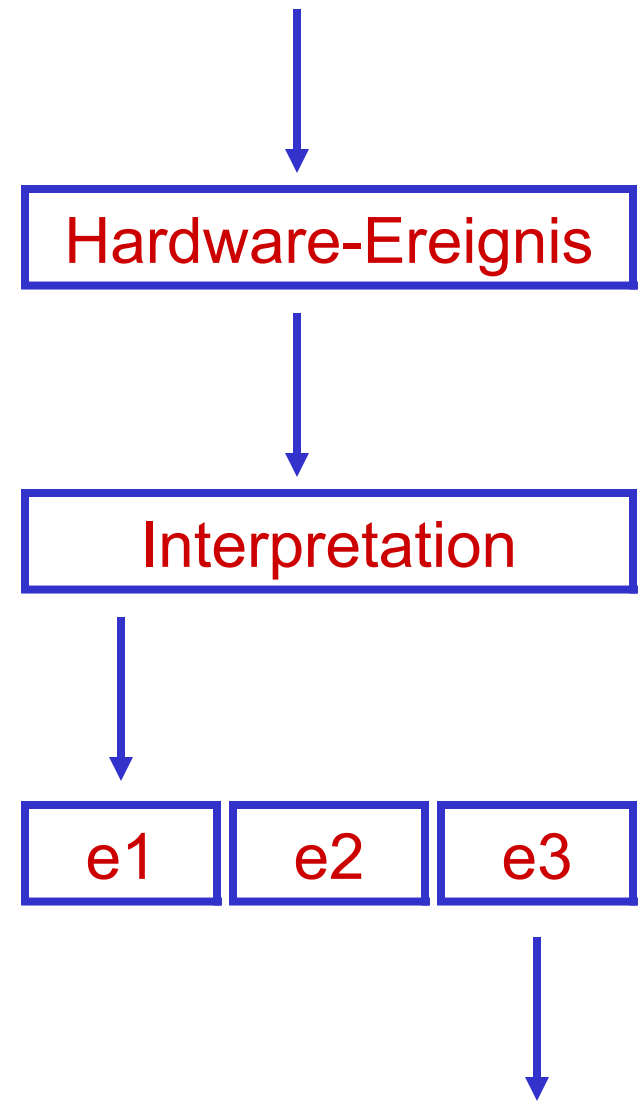


4. Ereignisbehandlung

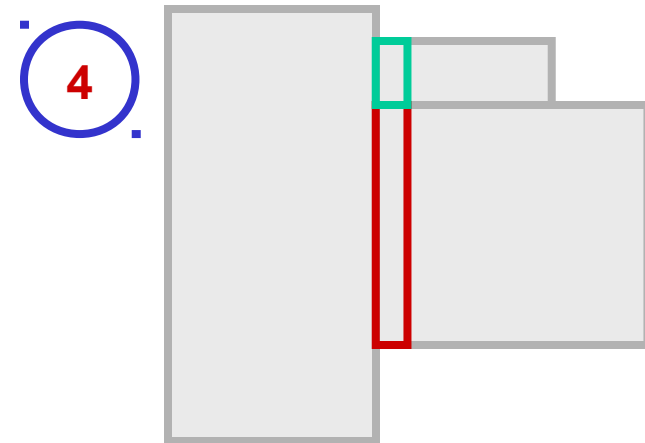
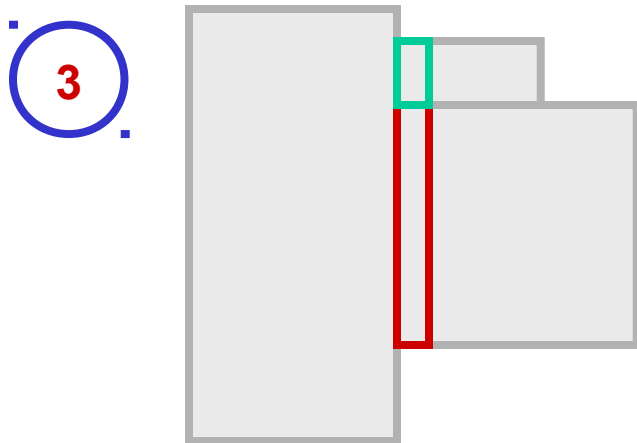
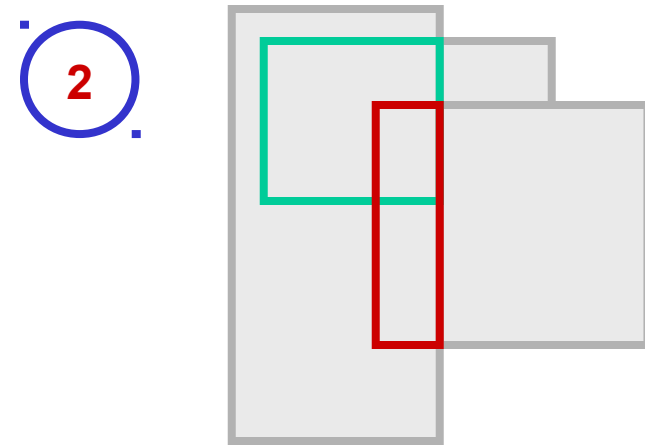
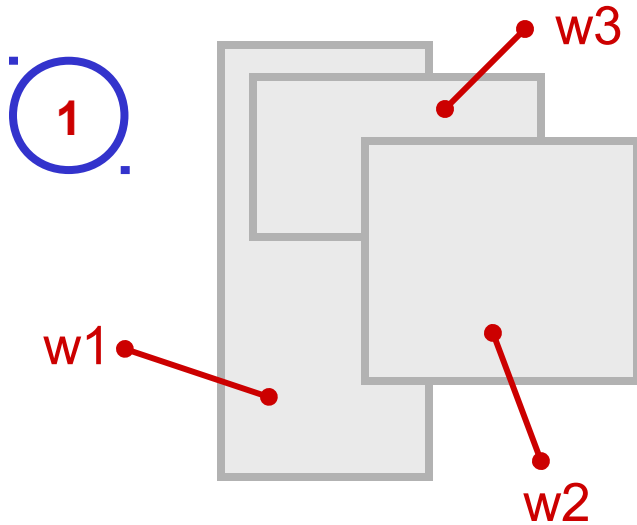
4.1 Event Queue

Event Queue

- Das Graphik-System generiert aus den Hardware-Ereignissen (Interrupts) Software-Ereignisse und verwaltet diese in einer Warteschlange, der **event queue**
- Eine Benutzeraktion kann dabei u.U. mehrere Ereignisse, auch für mehrere Windows, zur Folge haben
- Die Ereignisse in der queue werden nacheinander an die einzelnen Prozesse (bzw. deren Windows) ausgeliefert



4.1 Event Queue



4.1 Event Queue

Ausgelöste Ereignisse

Übergang	Ereignisse
(1) → (2)	Redraw (w1, Rechteck1) Redraw (w1, Rechteck2)
(2) → (3)	ComponentEvent (w1, Neue Position) → löst folgende weiteren Events aus Redraw (w2, Rechteck 3) Redraw (w3, Rechteck 4)
(3) → (4)	ComponentEvent (w1, Neue Position) → löst folgende weiteren Events aus Redraw (w2, Rechteck 5) Redraw (w3, Rechteck 6)

4.2 Ereignistypen

Zwei Haupttypen von Ereignissen

- Low level Ereignisse (Basisereignisse)
 - direkt aufgrund der Benutzeraktionen entstehende Ereignisse
- High level Ereignisse (semantische Ereignisse)
 - aufgrund des Zusammenhangs schon interpretierte Ereignisse (oder Ereigniskombinationen)

Dieser Abschnitt behandelt low level Ereignisse.

Semantische Ereignisse spielen bei GUI-Klassen eine Rolle (siehe spätere Kapitel).

4.2 Ereignistypen

Hierarchie-Ereignisse

- Veränderungen innerhalb eines Containers

`java.awt.event.ContainerEvent`

- Veränderungen in der Window-Hierarchie

`java.awt.event.HierarchyEvent`

Komponenten-Ereignis

- Veränderung der Geometrie (x, y, w, h, ...)

`java.awt.event.ComponentEvent`

Paint-Ereignis

- Redraw

`java.awt.event.PaintEvent`

4.2 Ereignistypen

Window-Ereignis

- Veränderungen des Window-Status (iconified, active, closed, closing, ...)

`java.awt.event.WindowEvent`

Eingabe-Ereignisse

- Tastatur-Eingabe

`java.awt.event.KeyEvent`

- Mouse-Eingabe (press, release, click, motion, ...)

`java.awt.event.MouseEvent`

Focus-Ereignis

- Komponente erhält oder verliert den Keyboard-Focus

`java.awt.event.FocusEvent`

4.3 Event-Basisklassen

Java-Basisklassen für Ereignisse

```
java.lang.Object
|
+-- java.util.EventObject
    |
    +-- java.awt.AWTEvent
```


4.3.1 Class java.util.EventObject

Beschreibung

The root class from which all event state objects shall be derived.

All Events are constructed with a reference to the object, the "source", that is logically deemed to be the object upon which the Event in question initially occurred upon.

Methoden

```
Object getSource() ;  
String toString() ;
```

4.3.2 Class java.awt.AWTEvent

Beschreibung

The root event class for all AWT events. (...)

Öffentliche Klassenkonstante

Die Klasse beinhaltet alle Event-Masken des AWT.

Methoden

`String paramString();`

Returns a string representing **the state of this event**. This method is intended to be used only for debugging purposes, and the content and format of the returned string may vary between implementations. The returned string may be empty but may not be null.

4.4 Event-Selektion

Mechanismen für die Auswahl von Ereignissen

- **Event Listener** Interfaces für den Ereignistyp **E**
- **Adapter**-Klasse für den Ereignistyp **E**
- **addEListener()**-Methoden der verschiedenen AWT-Komponentenklassen

Basismechanismus

- Anwendung konstruiert ein Listener-Objekt **L** für die Komponente **K**
- Mittels **K.addEListener(L)** wird das Listener-Objekt der Komponente zugewiesen
- Ab diesem Zeitpunkt wird das Listener-Objekt **L** informiert, wenn auf der Komponente **K** ein Ereignis **E** eintritt

4.4 Event-Selektion

Beispiel: ComponentEvent

- Wird ausgelöst, wenn sich die Geometrie und die Sichtbarkeit einer Komponente verändert
- Event-Klasse: `java.awt.event.ComponentEvent`
- Listener-Interface: `java.awt.event.ComponentListener`
- Zu implementierende Methoden des Interfaces
 - `componentShown()`
 - `componentHidden()`
 - `componentResized()`
 - `componentMoved()`
- Methode zum Registrieren eines Listener-Objektes, definiert in der Klasse `java.awt.Component`
 - `addComponentListener(ComponentListener l)`
- Adapter-Klasse `java.awt.event.ComponentAdapter`

4.4 Event-Selektion

Informationen im Event-Objekt der Klasse `ComponentEvent`

- **source**: Quell-Objekt des Events
- **id**: eine Integer-Id, die den Event-Typ eindeutig identifiziert
- **component**: die Komponente, die das Event ausgelöst hat

Methoden

`Component getComponent();`

Quell-Komponente

`String paramString();`

Text-Repräsentation

→ geerbt von `AWTEvent`

`int getID();`

Event-Typ

`String toString();`

Text-Repräsentation

→ geerbt von `EventObject`

`Object getSource();`

Quell-Objekt

4.4 Event-Selektion

Beispiel: `ComponentListenerPrinter`

Demo: `Frame2Demo2`

Listener-Klasse:

```
public class ComponentListenerPrinter
    implements ComponentListener {
    private int count=0;

    public void componentResized(ComponentEvent e) {
        System.out.println(count+": Component Resized");
        System.out.println("  New size = (" +
            e.getComponent().getWidth()+" "+
            e.getComponent().getHeight()+"") );

        count++;
    }
    ...
}
```

4.4 Event-Selektion

Listener-Klasse (Fortsetzung)

```
public void componentMoved(ComponentEvent e) {  
    System.out.println(count+": Component Moved");  
    System.out.println("    New position = (" +  
        e.getComponent().getX()+" , "+  
        e.getComponent().getY()+" )");  
  
    count++; }  
  
public void componentShown(ComponentEvent e) {  
    System.out.println(count+": Component Shown");  
    count++; }  
  
public void componentHidden(ComponentEvent e) {  
    System.out.println(count+": Component Hidden");  
    count++; }  
}
```

4.4 Event-Selektion

Hinzufügen des Listeners zu der Komponente

```
public class Frame2Demo2
{

    public static void main(String[] args)
    {
        Frame2 demo = new Frame2("Frame2Demo2",300,200,100,100);
        ComponentListenerPrinter myListener =
            new ComponentListenerPrinter();
        demo.addComponentListener(myListener);
        demo.setVisible(true);
    }
}
```


4.4 Event-Selektion

Beispiel: `ComponentListenerExtendedPrinter`

Demo: `Frame2Demo3`

Listener-Klasse: - Ausführliche Ausgabe des Events

```
private void printAll(ComponentEvent e) {  
    System.out.println("    Quell-Objekt:"  
        +e.getSource().toString()+"\n"+  
        "    Quell-Komponente:"  
        +e.getComponent().toString()+"\n"+  
        "    Event-Typ:"  
        +e.getID()+"\n"+  
        "    ToString:"  
        +e.toString()+"\n"+  
        "    ParamString:"  
        +e.paramString()+"\n\n    ");  
}
```

4.4 Event-Selektion

0: Component Shown

```
-----  
Quell-Objekt:  
  org.enviromatics.teaching.awt.Frame2[frame0,100,100,300x200,layout=java.awt.BorderLayo  
  ut,resizable,title=Frame2Demo2 - (c) by Snoopy Brown]  
Quell-Komponente:  
  org.enviromatics.teaching.awt.Frame2[frame0,100,100,300x200,layout=java.awt.BorderLayo  
  ut,resizable,title=Frame2Demo2 - (c) by Snoopy Brown]  
Event-Typ:      102  
ToString:      java.awt.event.ComponentEvent[COMPONENT_SHOWN] on frame0  
ParamString:    COMPONENT_SHOWN  
paint(): Paint-Ereignis  
paint(): Paint-Ereignis  
paint(): Paint-Ereignis
```

1: Component Resized

```
-----  
Quell-Objekt:  
  org.enviromatics.teaching.awt.Frame2[frame0,100,100,323x200,layout=java.awt.BorderLayo  
  ut,resizable,title=Frame2Demo2 - (c) by Snoopy Brown]  
Quell-Komponente:  
  org.enviromatics.teaching.awt.Frame2[frame0,100,100,323x200,layout=java.awt.BorderLayo  
  ut,resizable,title=Frame2Demo2 - (c) by Snoopy Brown]  
Event-Typ:      101  
ToString:      java.awt.event.ComponentEvent[COMPONENT_RESIZED (100,100 323x200)] on  
  frame0  
ParamString:    COMPONENT_RESIZED (100,100 323x200)
```

Adapter-Klassen

- Oft interessiert man sich nur für wenige der zu implementierenden Ereignisse eines Ereignis-Interfaces
 - Beispiel: `ComponentEvent`, i.d.R. ist nur `componentResized()` von Interesse
- Dennoch muß man alle Methoden des Interfaces implementieren
- Abhilfe: **Adapter-Klassen**
 - Abstrakte Klassen, die alle Methoden des Interfaces implementieren; diese Methoden beinhalten keine Anweisungen
 - Von diesen Adapter-Klassen leitet man seine Listener-Klasse ab und überlädt die Ereignis-Methoden, die von Interesse sind

4.4 Event-Selektion

Beispiel: ComponentAdapter

Quellcode der Klasse `java.awt.event.ComponentAdapter`

```
public abstract class ComponentAdapter
    implements ComponentListener
{
    public void componentResized(ComponentEvent e) {}
    public void componentMoved(ComponentEvent e) {}
    public void componentShown(ComponentEvent e) {}
    public void componentHidden(ComponentEvent e) {}
}
```

4.4 Event-Selektion

Beispiel: `ResizeListener`

Demo: `Frame2Demo4`

```
public class ResizeListener extends ComponentAdapter
{
    public void componentResized(ComponentEvent e)
    {
        System.out.println("  Component Resized");
        System.out.println("  New size = (" +
                           e.getComponent().getWidth()+" "+
                           e.getComponent().getHeight()+"") );
    }
}
```

4.5 Event-Klassen

```
java.awt.event.AWTEvent
|
+-- java.awt.event.ComponentEvent
    |
    +-- java.awt.event.WindowEvent
    |
    +-- java.awt.event.ContainerEvent
    |
    +-- java.awt.event.FocusEvent
    |
    +-- java.awt.event.InputEvent
        |
        +-- java.awt.event.MouseEvent
        |
        +-- java.awt.event.KeyEvent
    |
    +-- java.awt.event.PaintEvent
```

4.5.1 Class java.awt.AWTEvent

Beschreibung:

Metaklasse für alle AWT-Ereignisse.

Listener-Interface: -

Methoden: -

Adapter-Klasse: -

Informationen im Ereignis:

Ereignis-ID	int getID()
Parameter-String (für Debugging)	String paramString()

4.5.2 Class java.awt.ComponentEvent

Beschreibung:

Wird ausgelöst bei Änderungen der Geometrie und wenn eine Komponente sichtbar oder unsichtbar wird.

Listener-Interface:

`ComponentListener`

Methoden:

`componentShown()`

`componentHidden()`

`componentMoved()`

`componentResized()`

Adapter-Klasse:

`ComponentAdapter`

4.5.2 Class java.awt.ComponentEvent

Add-Methode:

`addComponentListener()` in `java.awt.Component`

Informationen im Ereignis:

Quellkomponente	<code>Component</code> <code>getComponent()</code>
Parameter-String	<code>String</code> <code>paramString()</code>

Ereignis-Ids (int):

- `ComponentEvent.COMPONENT_SHOWN`
- `ComponentEvent.COMPONENT_HIDDEN`
- `ComponentEvent.COMPONENT_MOVED`
- `ComponentEvent.COMPONENT_RESIZED`

4.5.3 Class java.awt.WindowEvent

Beschreibung:

Wird ausgelöst, wenn ein Toplevel-Window (`Frame` oder `Window`) seinen Status ändert (d.h. geöffnet, geschlossen, iconifiziert, deiconifiziert, aktiviert oder deaktiviert wird).

Listener-Interface: `WindowListener`

Methoden: `windowActivated()`

`windowDeactivated()`

`windowIconified()`

`windowDeiconified()`

`windowOpened()`

`windowClosed()`

`windowClosing()`

4.5.3 Class java.awt.WindowEvent

Adapter-Klasse:

WindowAdapter

Add-Methode:

addWindowListener() in java.awt.Window

Informationen im Ereignis:

Quellwindow

Window getWindow()

Parameter-String

String paramString()

4.5.3 Class java.awt.WindowEvent

Ereignis-Ids (int):

WindowEvent.WINDOW_ACTIVATED

WindowEvent.WINDOW_DEACTIVATED

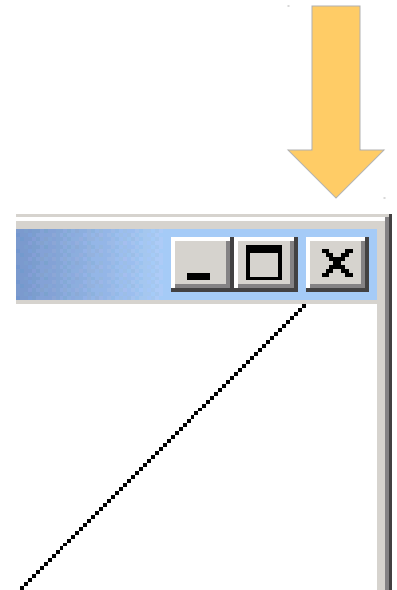
WindowEvent.WINDOW_OPENED

WindowEvent.WINDOW_CLOSED

WindowEvent.WINDOW_CLOSING

WindowEvent.WINDOW_ICONIFIED

WindowEvent.WINDOW_DEICONIFIED



Anwendung des WindowEvent:

- Beenden des Programms durch den Window-Manager → WINDOW_CLOSING Event

4.5.3 Class java.awt.WindowEvent

Beispiel: ClosingListener

Demo: Frame2Demo5

```
public class ClosingListener extends WindowAdapter {
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
}

public class Frame2Demo5 {
    public static void main(String[] args) {
        Frame2 demo = new Frame2("Frame2Demo5",300,200,100,100);
        ClosingListener myListener = new ClosingListener();
        demo.addWindowListener(myListener);
        demo.setVisible(true);
    }
}
```

4.5.4 Class java.awt.InputEvent

Beschreibung:

Basisklasse für alle Eingabe-Ereignisse.

Informationen im Ereignis:

Event-Maske	<code>int getModifiers()</code>
Event-Modifier	<code>boolean isAltDown()</code>
	<code>boolean isAltGraphDown()</code>
	<code>boolean isControlDown()</code>
	<code>boolean isMetaDown()</code>
	<code>boolean isShiftDown()</code>
Timestamp	<code>long getWhen()</code>

4.5.4 Class java.awt.InputEvent

Event-Masken:

InputEvent.GRAPH_MASK

InputEvent.ALT_GRAPH_MASK

InputEvent.CONTROL_MASK

InputEvent.META_MASK

InputEvent.SHIFT_MASK

InputEvent.BUTTON1_MASK

InputEvent.BUTTON2_MASK

InputEvent.BUTTON3_MASK

4.5.5 Class java.awt.MouseEvent

Beschreibung:

Event-Klasse für **Mouse Events**: mouse button pressed, mouse button released, mouse button clicked, mouse cursor enters component, mouse cursor exits component sowie **Mouse Motion Events**: mouse moved, mouse dragged.

Informationen im Ereignis:

Anzahl Clicks	<code>int getClickCount()</code>
Position	<code>Point getPoint()</code>
X-Position	<code>int getX()</code>
Y-Position	<code>int getY()</code>

4.5.5 Class java.awt.MouseEvent

Listener-Interface:

MouseListener

Methoden:

mousePressed()

mouseReleased()

mouseClicked()

mouseEntered()

mouseExited()

Adapter-Klasse:

MouseAdapter

Add-Methode:

addMouseListener() in java.awt.Component

4.5.5 Class java.awt.MouseEvent

Listener-Interface:

MouseMotionListener

Methoden:

mouseMoved()

mouseDragged()

Adapter-Klasse:

MouseMotionAdapter

Add-Methode:

addMouseMotionListener() in java.awt.Component

4.5.6 Class java.awt.KeyEvent

Beschreibung:

Event-Klasse für Key Events, die sowohl sichtbare als auch Kontrollzeichen verarbeitet.

Listener-Interface:

KeyListener

Methoden:

keyPressed()

keyReleased()

keyTyped()

Adapter-Klasse:

KeyAdapter

4.5.6 Class java.awt.KeyEvent

Informationen im Ereignis:

Zeichen	<code>char getKeyChar()</code>
Keycode	<code>int getKeyCode()</code>
Beschreibung	<code>String getKeyText()</code>

Virtual Key Codes:

- Tastatur-Eingaben werden über **Virtual Keys** beschrieben
- Diese ordnen jedem lesbaren Zeichen und jedem Steuerzeichen einen **Key Code** zu
- Die Keycodes sind als Klassenkonstante in KeyEvent deklariert

4.5.6 Class java.awt.KeyEvent

Virtual Key Codes:

<code>KeyEvent.VK_0</code>	<code>...</code>	<code>KeyEvent.VK_9</code>
<code>KeyEvent.VK_A</code>	<code>...</code>	<code>KeyEvent.VK_Z</code>
<code>KeyEvent.VK_SPACE</code>	<code>...</code>	<code>KeyEvent.VK_UNDERSCORE</code>
<code>KeyEvent.VK_ADD</code>	<code>...</code>	<code>KeyEvent.VK_DIVIDE</code>
<code>KeyEvent.VK_F1</code>	<code>...</code>	<code>KeyEvent.VK_F24</code>
<code>KeyEvent.VK_LEFT</code>	<code>...</code>	<code>KeyEvent.VK_DOWN</code>

etc.

4.5.7 Class java.awt.ContainerEvent

Beschreibung:

Wird ausgelöst, wenn eine Komponente zu einem Container hinzugefügt oder entfernt wird.

- siehe JDK API-Dokumentation

4.5.8 Class java.awt.FocusEvent

Beschreibung:

Wird ausgelöst, wenn eine Komponente den Keyboard-Focus erhält oder verliert.

- siehe JDK API-Dokumentation