

10. Dialogkomponenten

10.1 Allgemeine Eigenschaften von Dialogen

Abhängigkeit vom Frame-Parent

- Wenn man mehrere JFrame-Objekte erzeugt, so können diese separat iconifiziert werden
- Im Gegensatz dazu hängen Dialoge immer von ihrem Parent-Frame ab
 - wird dieser iconifiziert, so verschwindet der Dialog mit ihm
 - ohne ein eigenes Icon

Modalität

- Gibt an, ob ein Dialog die Hoheit über die Eingabe einer Anwendung hat
- Modale Dialoge sind im Vordergrund, der Rest der Anwendung ist deaktiviert

10.1 Allgemeine Eigenschaften von Dialogen

Allgemeine Formen von Modalität

- Nicht-Modal
- Anwendungs-Modal
- System-Modal

Java kennt keine System-Modalität.

Dialog-Klassen

- | | |
|---------------|--|
| • Dialog | Dialog-Klasse aus AWT |
| • JDialog | Allgemeine Dialog-Klasse aus JFC |
| • JOptionPane | Convenience-Klasse mit Standard-Dialogen |

10.2 Class java.awt.Dialog

Beschreibung:

A Dialog is a top-level window with a title and a border that is typically used to take some form of input from the user. The default layout for a dialog is BorderLayout. A dialog must have either a frame or another dialog defined as its owner when it's constructed. When the owner window of a visible dialog is hidden or minimized, the dialog will automatically be hidden from the user. When the owner window is subsequently re-opened, then the dialog is made visible to the user again. A dialog can be either modeless (the default) or modal. A modal dialog is one which blocks input to all other toplevel windows in the app context, except for any windows created with the dialog as their owner.

10.2 Class java.awt.Dialog

Konstruktoren:

```
JDialog ( Dialog owner )
```

```
JDialog ( Frame owner )
```

```
JDialog ( Dialog owner, String title )
```

```
JDialog ( Frame owner, String title )
```

```
JDialog ( Dialog owner, String title, boolean modal )
```

```
JDialog ( Frame owner, String title, boolean modal )
```

Eigenschaft: Modalität

```
void setModal ( boolean modal )
```

```
boolean isModal()
```

10.2 Class java.awt.Dialog

Eigenschaft: Resizable

```
void setResizable ( boolean resizable )  
boolean isResizable()
```

Eigenschaft: Title

```
void setTitle ( String title )  
String getTitle()
```

Anzeige:

```
void show()           void hide()
```

10.3 Class javax.swing.JDialog

Beschreibung:

The main class for creating a dialog window. You can use this class to create a custom dialog, or invoke the many class methods in JOptionPane to create a variety of standard dialogs.

JDialog wird in der Vorlesung nicht besprochen.

10.4 Class javax.swing.JOptionPane

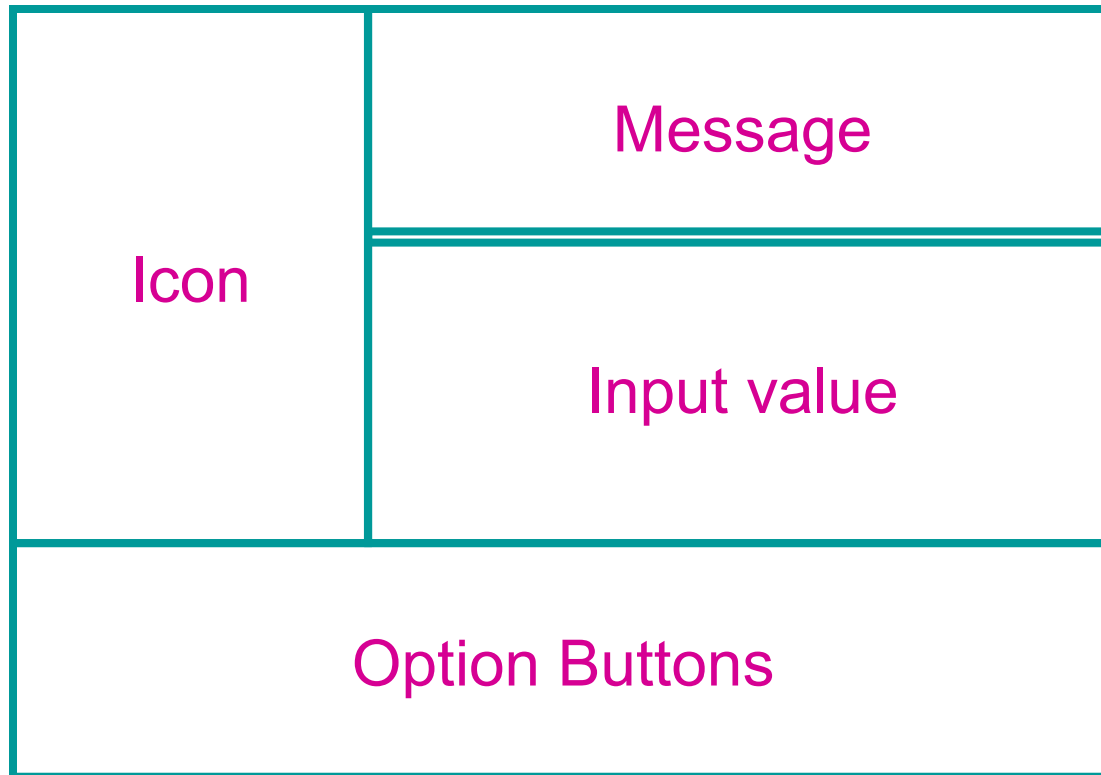
Beschreibung:

JOptionPane makes it easy to pop up a standard dialog box that prompts users for a value or informs them of something. While the JOptionPane class may appear complex because of the large number of methods, almost all uses of this class are one-line calls to one of the static showXxxDialog methods shown below:

showConfirmDialog	Asks a confirming question, like yes/no/cancel.
showInputDialog	Prompt for some input.
showMessageDialog	Tell the user about something that has happened.
showOptionDialog	The Grand Unification of the above three.

10.4 Class javax.swing.JOptionPane

Aufbau:



10.4 Class javax.swing.JOptionPane

Beispiel: Fehlermeldungen

```
try
{
    ... Versuche ne URL zu öffnen, wirft
        eine exception ...
}
catch ( MalformedURLException url_exception )
{
    JOptionPane.showMessageDialog
        ( parent, "Malformed URL", "Resources Error",
          JOptionPane.INFORMATION_MESSAGE, null );
}
```

parent ist der **owner** und muß ein Frame/JFrame sein

10.4 Class javax.swing.JOptionPane

Setzen/Übergabe von Parametern an JOptionPane

- Parameter werden unterschiedlich interpretiert, je nach dem, welchen Datentyp sie besitzen

Parameter Message

Component	Darstellung der Komponente
Icon	Darstellung des Icons
Others	Konversion mittels toString() und Erzeugung eines Labels
Object[]	Ein Array wird als Feld von Meldungen interpretiert, die eine nach der andere interpretiert wird

10.4 Class javax.swing.JOptionPane

Parameter Message

- Einfachster Fall: Übergabe eines Strings oder einer String-Konstante

Meldungstypen

- Über den Meldungstyp kann man einen der Standard-Icons steuern

`JOptionPane.INFORMATION_MESSAGE`

`JOptionPane.ERROR_MESSAGE`

`JOptionPane.WARNING_MESSAGE`

`JOptionPane.QUESTION_MESSAGE`

`JOptionPane.PLAIN_MESSAGE`

10.4 Class javax.swing.JOptionPane

Beispiel

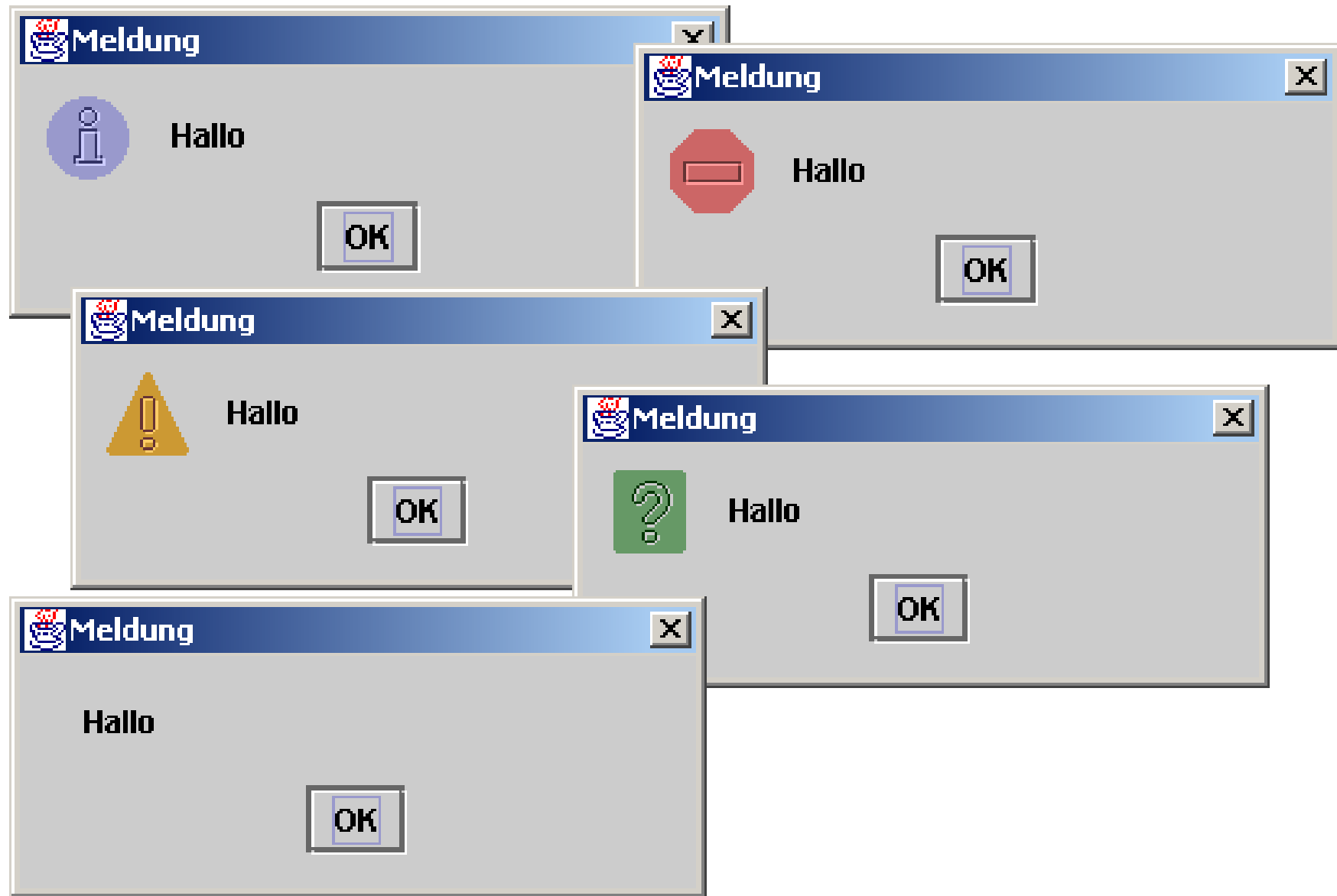
- Benutzer aktiviert einen Button, das führt zur Darstellung eines Dialogs

```
public void actionPerformed(ActionEvent e) {  
    JOptionPane.showMessageDialog(null, "Hallo",  
        "Meldung", JOptionPane.INFORMATION_MESSAGE );  
}
```

Bemerkungen:

- Erster Parameter ist immer der Parent-Frame; kann null sein, dann wird ein Default-Frame erzeugt
- Von den showXxxDialog gibt es eine ganze Reihe von Varianten, die nicht alle hier besprochen werden (Selbststudium)

10.4 Class javax.swing.JOptionPane

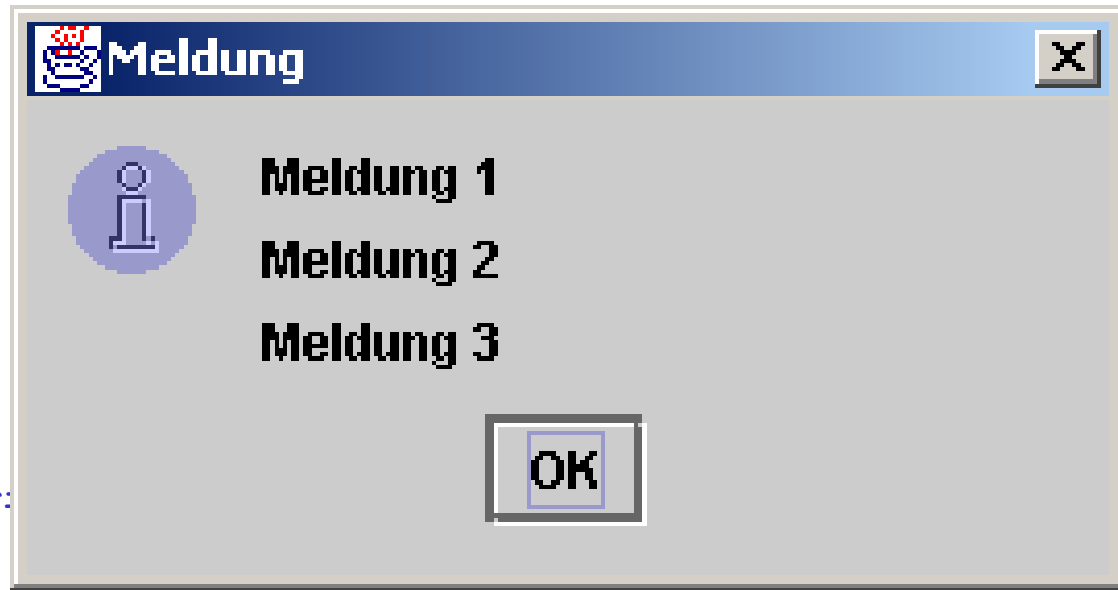


10.4 Class javax.swing.JOptionPane

Parameter Message

- Übergabe eines Objekt-Arrays

```
String[] messages = new String[3];  
messages[0] = "Meldung 1";  
messages[1] = "Meldung 2";  
messages[2] = "Meldung 3";  
JOptionPane.showMessageDialog(null, messages,  
    "Meldung", JOptionPane.INFORMATION_MESSAGE );
```

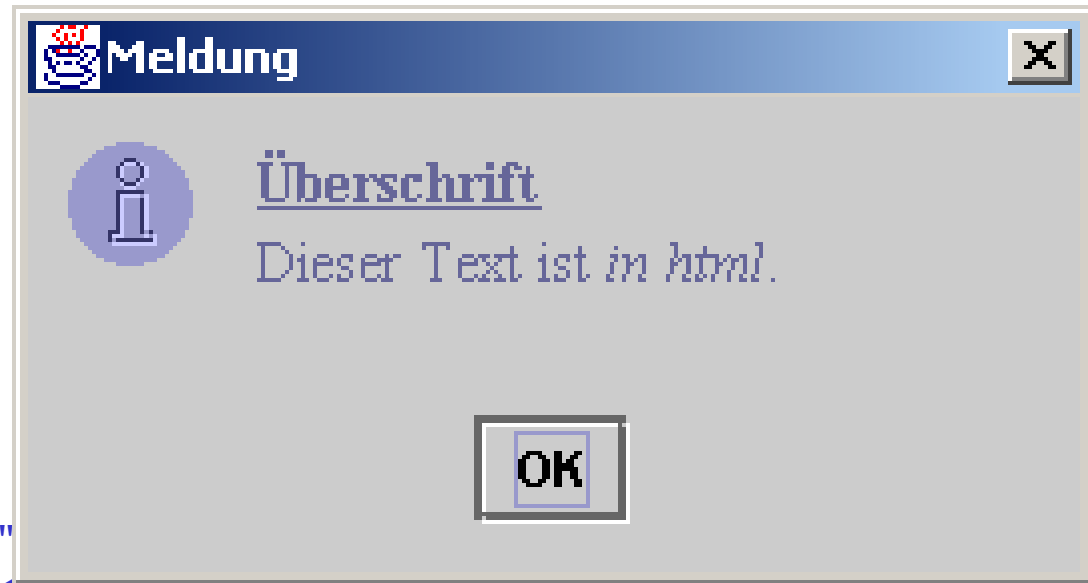


10.4 Class javax.swing.JOptionPane

Parameter Message

- Übergabe eines HTML-String

```
String message = new String("  
    "<b><u>Überschrift</u></b><p>"+  
    "Dieser Text ist <i>in html</i>.<p>"+  
    "</html>");  
JOptionPane.showMessageDialog(null,message,  
    "Meldung",JOptionPane.INFORMATION_MESSAGE );
```



10.4 Class javax.swing.JOptionPane

Parameter Message

- Gemischte
Übergabe

```
Object[] messages = new Object[2];
JLabel message1 =
    new JLabel("And the message is ...");
messages[0] = message1;
String message2 = new String(" ... ");
messages[1] = message2;
```



10.4 Class javax.swing.JOptionPane

Lokale Event-Loop

- im Gegensatz zum normalen Ablauf wird bei den showXxxDialog-Methoden die Event-Loop mit dem Code synchronisiert
- d.h. es erfolgt kein Sprung in einen ActionListener, wenn einer der Buttons aktiviert wird

```
-> JOptionPane.showMessageDialog(null, "Hallo",  
|     "Meldung", JOptionPane.INFORMATION_MESSAGE );  
|
```

```
-> Fortsetzung der Anwendung  
    nach Schließen des Dialogs
```

10.4 Class javax.swing.JOptionPane

Parameter Option-Type

- Parameter für Standard-Optionen

`JOptionPane.DEFAULT_OPTION`

`JOptionPane.YES_NO_OPTION`

`JOptionPane.YES_NO_CANCEL_OPTION`

`JOptionPane.OK_CANCEL_OPTION`

```
int s = JOptionPane.showConfirmDialog(null,  
    "Question", "Title",  
    JOptionPane.YES_NO_CANCEL_OPTION,  
    JOptionPane.QUESTION_MESSAGE );  
if ( s == JOptionPane.YES_OPTION ) ...  
else if ( selection == JOptionPane.NO_OPTION ) ...  
else if ( selection == JOptionPane.CANCEL_OPTION )  
    ...
```

10.4 Class javax.swing.JOptionPane

Einfache Eingabe-Dialoge - Input-Dialog

- Einfachste Version liefert einen String als Ergebnis des Aufrufs

```
String s =  
    JOptionPane.showInputDialog("Filename:");  
if ( s == null )  
    System.out.println("Cancelled");  
else if ( s.length()==0 )  
    System.out.println("String Empty");  
else  
    System.out.println("Filename = "+s);
```

10.4 Class javax.swing.JOptionPane

Abschlußbemerkung zu diesem Kapitel

- Die Behandlung der Dialogkomponenten in der Vorlesung geht nur so weit, daß ein weiteres Selbst-Studium in den Übungen vorbereitet ist
- Nicht behandelt werden insbesondere
 - Wie geht man vor, wenn man nicht showXxxDialog verwendet, sondern spezifische Modifikationen vornehmen will, bevor der Dialog angezeigt wird
 - Wie baut man Custom Dialogs auf
 - Überblick über Methoden der Klasse JOptionPane

Demo:

DialogDemo1