

9. Basiskomponenten

9.1 Überblick

- JLabel
- JButton
- JTextField
- JPasswordField
- JTextArea
- JSeparator
- JList
- JSlider
- JCheckbox
- JComboBox

9.2 Class javax.swing.JLabel

Beschreibung:

A display area for a short text string or an image, or both. A label does not react to input events. A JLabel object can display either text, an image, or both. You can specify where in the label's display area the label's contents are aligned by setting the vertical and horizontal alignment.

Konstruktoren:

```
JLabel ()  
JLabel ( String text )  
JLabel ( Icon icon, int horizontalAlignment )  
JLabel ( String text, Icon icon ,  
        int horizontalAlignment )
```

9.2 Class javax.swing.JLabel

Eigenschaft: Label-Text, Label-Icon

```
void setText ( String text );
```

```
String getText();
```

```
void setIcon ( Icon icon );
```

```
Icon getIcon();
```

Eigenschaft: Ausrichtung

```
void setHorizontalAlignment ( int alignment );
```

```
int getHorizontalAlignment();
```

```
void setVerticalAlignment ( int alignment );
```

```
int getVerticalAlignment();
```

9.2 Class javax.swing.JLabel

Eigenschaft: Ausrichtung

```
void setHorizontalTextPosition ( int textPosition );  
int getHorizontalTextPosition();
```

```
void setVerticalTextPosition ( int textPosition );  
int getVerticalTextPosition();
```

9.2 Class javax.swing.JLabel

Eigenschaft: Ausrichtung

Konstante für Ausrichtungen und Alignment, definiert im Interface `javax.swing.SwingConstants`:

| | |
|--------------------------------------|--|
| <code>SwingConstants.TOP</code> | <code>SwingConstants.BOTTOM</code> |
| <code>SwingConstants.LEFT</code> | <code>SwingConstants.RIGHT</code> |
| <code>SwingConstants.CENTER</code> | |
| <code>SwingConstants.VERTICAL</code> | <code>SwingConstants.HORIZONTAL</code> |
| <code>SwingConstants.NORTH</code> | <code>SwingConstants.SOUTH</code> |
| <code>SwingConstants.EAST</code> | <code>SwingConstants.WEST</code> |

... und einige mehr, siehe JDK-Dokumentation.

9.2 Class javax.swing.JLabel

Demo: LabelDemoFrame

```
public class LabelDemoFrame extends JFrame {
    public LabelDemoFrame() {
        getContentPane().setLayout (new BorderLayout(5,5));
        ((JComponent) getContentPane()).setBorder
            (new EmptyBorder(5,5,5,5));

        ImageIcon icon =  new ImageIcon("up.gif");

        JLabel l1 =
            new JLabel("North",icon,SwingConstants.CENTER);
        getContentPane().add(l1,BorderLayout.NORTH);
        l1.setVerticalTextPosition(SwingConstants.BOTTOM);
        l1.setHorizontalTextPosition(SwingConstants.CENTER);

        ...
    }
}
```

9.2 Class javax.swing.JLabel

```
... JLabel l2 =
    new JLabel("South",icon,SwingConstants.CENTER) ;
getContentPane().add(l2,BorderLayout.SOUTH) ;
l2.setVerticalTextPosition(SwingConstants.TOP) ;
l2.setHorizontalTextPosition(SwingConstants.CENTER) ;
JLabel l3 =
    new JLabel("West",icon,SwingConstants.CENTER) ;
getContentPane().add(l3,BorderLayout.WEST) ;
l3.setVerticalTextPosition(SwingConstants.TOP) ;
l3.setHorizontalTextPosition(SwingConstants.RIGHT) ;
JLabel l4 = new
    JLabel("East",icon,SwingConstants.CENTER) ;
getContentPane().add(l4,BorderLayout.EAST) ;
l4.setVerticalTextPosition(SwingConstants.TOP) ;
l4.setHorizontalTextPosition(SwingConstants.LEFT) ;
pack() ;
}
```


9.2 Class javax.swing.AbstractButton

Basisklasse

```
javax.swing.AbstractButton
|
+-- JButton
|
+-- JMenuItem
|   |
|   +-- JMenu
|       |
|       +-- JCheckBoxMenuItem
|           |
|           +-- JRadioButtonMenuItem
|
+-- JToggleButton
|
+-- JCheckBox
|
+-- JRadioButton
```

9.3 Class javax.swing.AbstractButton

Beschreibung:

Metaklasse für alle Button-Klassen.

Eigenschaften: Text, Icon

```
void setText ( String text );
```

```
String getText();
```

```
void setIcon ( Icon icon );
```

```
void setPressedIcon ( Icon icon );
```

```
void setRolloverIcon ( Icon icon );
```

```
Icon getIcon();
```

```
Icon getPressedIcon();
```

```
Icon getRolloverIcon();
```

9.3 Class javax.swing.AbstractButton

Eigenschaft: Ausrichtung

```
void setHorizontalAlignment ( int alignment );  
int getHorizontalAlignment();
```

```
void setVerticalAlignment ( int alignment );  
int getVerticalAlignment();
```

```
void setHorizontalTextPosition ( int textPosition );  
int getHorizontalTextPosition();
```

```
void setVerticalTextPosition ( int textPosition );  
int getVerticalTextPosition();
```

9.3 Class javax.swing.AbstractButton

Action-Listener:

```
addActionListener ( ActionListener listener );
```

Interface ActionListener:

```
actionPerformed ((ActionEvent e)
```

ActionEvent:

```
int getModifiers();
```

```
Geerbt (!):
```

```
Object getSource();
```

9.3 Class javax.swing.AbstractButton

Change-Listener:

```
addChangeListener ( ChangeListener listener );
```

Interface ChangeListener:

```
stateChanged ( ChangeEvent e )
```

ChangeEvent:

```
Geerbt (!) :
```

```
Object getSource () ;
```

9.3 Class javax.swing.AbstractButton

Item-Listener:

```
addItemListener ( ItemListener listener );
```

Interface ItemListener:

```
itemStateChanged ( ItemEvent e )
```

ChangeEvent:

```
Object getItem();  
int getStateChange();
```

```
Geerbt (!):
```

```
Object getSource();
```

9.4 Class javax.swing.JButton

Beschreibung:

An implementation of a "push" button.

Konstruktoren:

```
JButton()
```

```
JButton ( String text )
```

```
JButton ( Icon icon )
```

```
JButton ( String text, Icon icon )
```

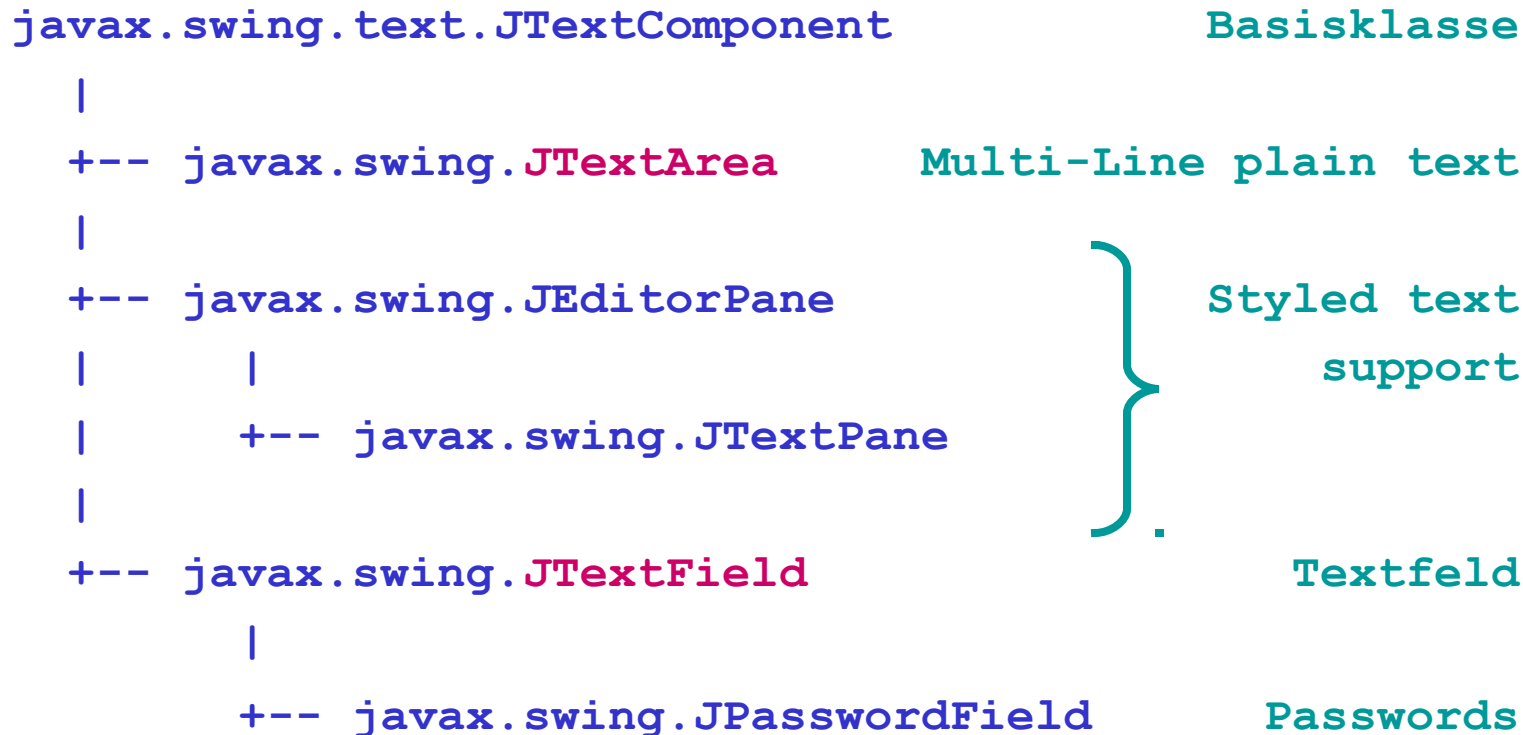
Eigenschaft: Default-Button (<ENTER>)

```
void setDefaultCapable ( int defaultcapable );
```

9.5 Text-Klassen

Textklassen in JFC

- Basis-Textklassen
- Dokumentenklassen (siehe MVC)



9.6 Class javax.swing.JTextField

Beschreibung:

JTextField is a lightweight component that allows the editing of a single line of text.

Konstruktoren:

```
JTextField ()  
JTextField ( String text )  
JTextField ( int columns )  
JTextField ( String text, int columns )
```

Action-Listener (<Enter>-Taste):

```
addActionListener ( ActionListener listener );
```

9.6 Class javax.swing.JTextField

Eigenschaften (z.T. geerbt):

```
setColumns ( int columns );
```

```
int getColumns();
```

```
setFont ( Font font );
```

```
Font getFont();
```

```
setHorizontalAlignment ( int alignment );
```

```
int getHorizontalAlignment();
```

```
void setText(String s);
```

```
String getText();
```

9.7 Class javax.swing.JTextArea

Beschreibung:

A TextArea is a multi-line area that displays plain text. It

Konstruktoren:

```
JTextArea ( int rows, int columns )
```

```
JTextArea ( String text )
```

```
JTextArea ( String text, int rows, int columns )
```

Eigenschaften (z.T. geerbt):

```
void setText ( String text );
```

```
String getText();
```

9.7 Class javax.swing.JTextArea

Eigenschaften (z.T. geerbt):

```
void setText ( String text );  
String getText();
```

```
void append ( String str );  
void insert ( String pos );
```

```
void setFont ( Font f );  
Font getFont();
```

```
void setLineWrap ( boolean wrap );  
boolean getLineWrap();
```

```
void setTabSize ( int size );  
int getTabSize();
```