

# Grafische Programmierung mit Java

## Modul JAV02

Daniel Falkner

AKAD University

05/06.September.2014



## Module JAV02

### Aufgabe 2

- Erläutern Sie ausführlich die elementaren Steuerelemente von Swing wie JLabel, JButton, JComboBox usw.

## Über mich

Daniel Falkner

- AKAD Student - Bachelor of Science (Wirtschaftsinformatik)
- T-Systems International GmbH - Telekom IT
- IT-Architekt - System Analyst
- Projektleiter
- Proof of Concept Engineer
- Debian Linux Administrator

## JLabel

### statischer Text, nicht editierbar

```
1 // Wir holen uns ein Icon aus dem dem Java Look and Feel
2 Icon icon = MetalIconFactory.
   getFileChooserHomeFolderIcon();
3
4 // Wir erstellen ein JLabel mit einem Text und
   unserem Icon
5 // Die horizontale Ausrichtung setzen wir auf "
   CENTER"
6 JLabel label = new JLabel ("My 127.0.0.1 is my
   castle",
7 icon, JLabel.CENTER);
8
9 // Die vertikale Ausrichtung des JLabels setzen wir
   auf "TOP"
10 label.setVerticalAlignment(JLabel.TOP);
11
12 // Die relative Ausrichtung des Textes zum Icon
   setzen wir auf "LEFT"
13 label.setHorizontalTextPosition(JLabel.LEFT);
```

## Demo

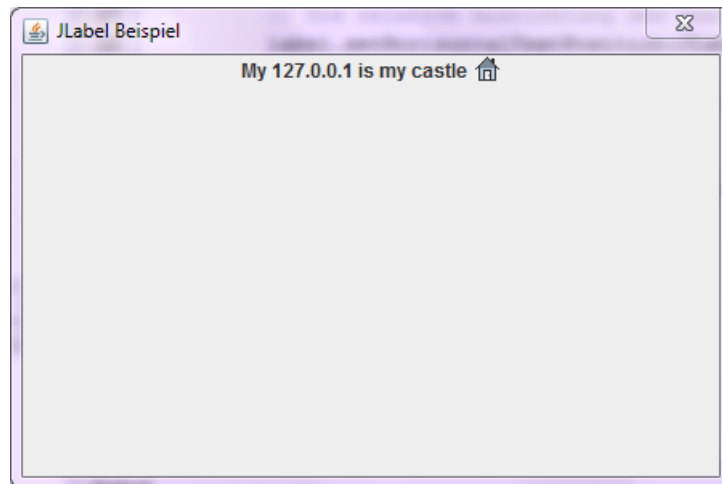


Abb.: JLabel

<http://http://java-tutorial.org/>

## Demo

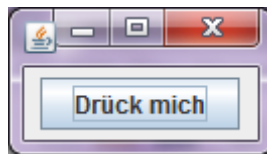


Abb.: JButton

<http://http://java-tutorial.org/>

## JButton

## Schaltfläche (Button)

- Vergleichbar mit einem Taster

```
1 // JButton mit Text "Drueck mich" wird erstellt
2     JButton button = new JButton("Drueck mich");
3
4     // JButton wird dem Panel hinzugefuegt
5     panel.add(button);
```

## JToggleButton

## Schaltfläche, welche zwei Zustände kennt.

- gedrückt und nicht gedrückt
- Vergleichbar einem Schalter

```
1
2 // JToggleButton mit Text "Drueck mich" wird erstellt
3     JToggleButton toggleButton = new JToggleButton("
4         Drueck mich", true);
5
6     // JToggleButton wird dem Panel hinzugefuegt
7     panel.add(toggleButton);
```

## Demo

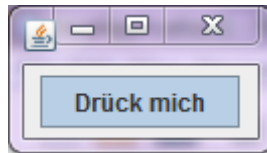


Abb.: JToggleButton Selektiert

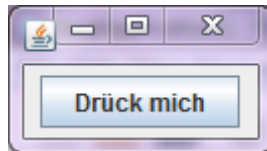
<http://http://java-tutorial.org/>

Abb.: JToggleButton Nicht Selektiert

<http://http://java-tutorial.org/>

## Demo

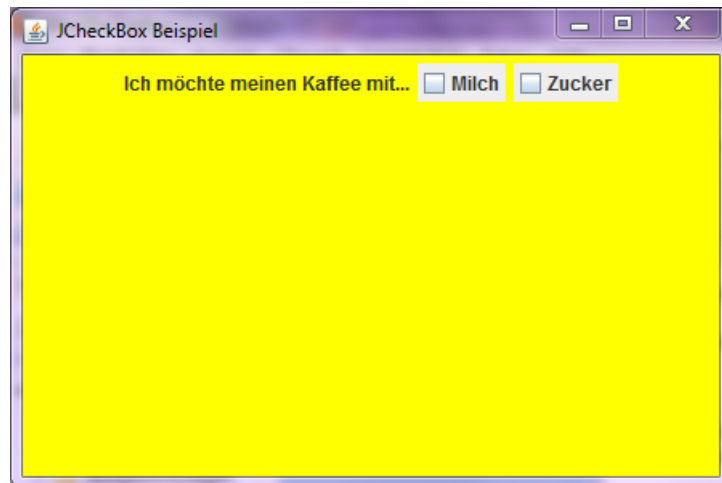


Abb.: JCheckBox

<http://http://java-tutorial.org/>

## JCheckBox

Auswahlkästchen, das, wenn es ausgewählt wurde, mit einem Häkchen oder Kreuz versehen wird.

```
1
2 //JCheckBoxen werden erstellt
3 JCheckBox checkBoxMilch = new JCheckBox("Milch");
4 JCheckBox checkBoxZucker = new JCheckBox("Zucker");
```

## JRadioButton

Schaltfläche zur Auswahl zwischen mehreren Optionen, in der Regel sind sie in einer ButtonGroup angeordnet. Im Gegensatz zur JCheckBox kann nur maximal eine Option selektiert werden.

```
1
2 // Hier wird ein selektierter JRadioButton erstellt
3 JRadioButton radio = new JRadioButton ("Ich bin ein
   RadioButton", true);
```

## Demo

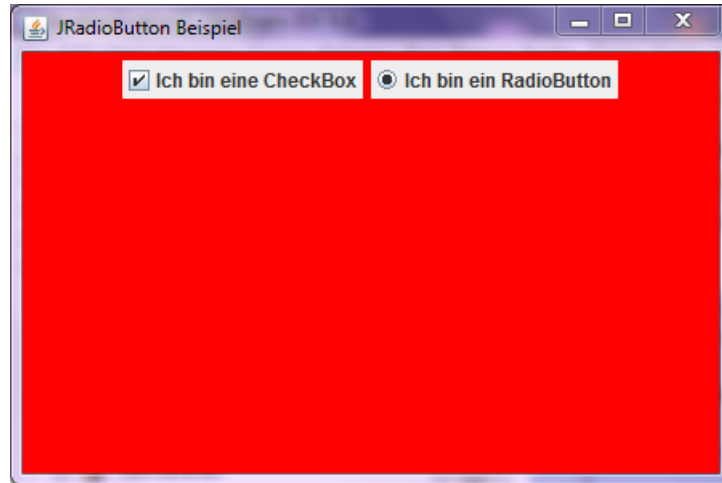


Abb.: JRadioButton

<http://http://java-tutorial.org/>

## Demo

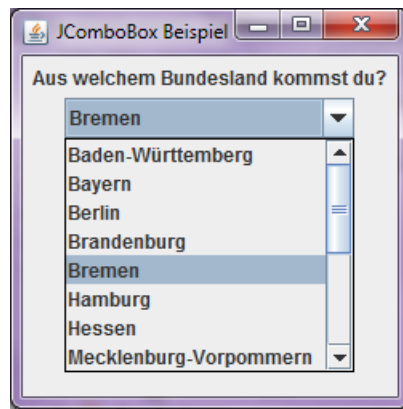


Abb.: JComboBox

<http://http://java-tutorial.org/>

## JComboBox

Dropdown-Liste (auch als Auswahlliste oder Listbox bezeichnet), die zur Auswahl eines Elementes aufgeklappt wird. Wenn die JComboBox editierbar ist, kann über ein Textfeld der ausgewählte Wert auch vom Anwender gesetzt werden.

```

1
2 // Array fuer unsere JComboBox
3 String comboBoxListe[] = {"Baden-Wuerttemberg", "
  Bayern",
4   "Berlin", "Brandenburg", "Bremen",
5   "Hamburg", "Hessen", "Mecklenburg-Vorpommern",
6   "Niedersachsen", "Nordrhein-Westfalen", "
  Rheinland-Pfalz",
7   "Saarland", "Sachsen", "Sachsen-Anhalt",
8   "Schleswig-Holstein", "Thueringen"};
9
10 //JComboBox mit Bundeslaender-Eintraegen wird
  erstellt
11 JComboBox bundeslandAuswahl = new JComboBox(
  comboBoxListe);

```

## JList

Einfache Liste, die mehrere Elemente enthalten kann. Einfach- und Mehrfachauswahl möglich.

```

1
2 // Array fuer unsere JList
3 String interessen[] = {"Politik", "Autos", "Mode",
4   "Film- und Fernsehen", "Computer", "Tiere", "
  Sport"};
5
6 //JList mit Eintraegen wird erstellt
7 JList themenAuswahl = new JList(interessen);

```

## Demo

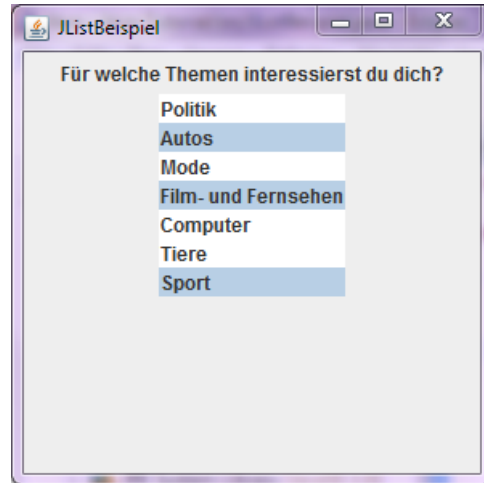


Abb.: JList

<http://http://java-tutorial.org/>

## Demo

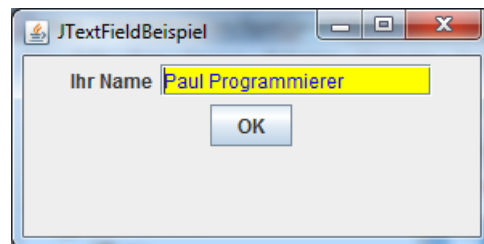


Abb.: JTextField

<http://http://java-tutorial.org/>

## JTextField

## einfache einzeilige Texteingabe

```

1
2 // Textfeld wird erstellt
3     // Text und Spaltenanzahl werden dabei direkt
    gesetzt
4     JTextField tfName = new JTextField("Paul
    Programmierer", 15);
5     // Schriftfarbe wird gesetzt
6     tfName.setForeground(Color.BLUE);
7     // Hintergrundfarbe wird gesetzt
8     tfName.setBackground(Color.YELLOW);

```

## JTextArea

## einfache mehrzeilige Texteingabe

```

1
2 //5-zeiliges und 20-spaltiges Textfeld wird erzeugt
3     JTextArea textfeld = new JTextArea(5, 20);
4
5     //Text fuer das Textfeld wird gesetzt
6     textfeld.setText("Lorem ipsum dolor sit amet, " +
7         "consetetur sadipscing elitr, sed diam nonumy " +
8         "eirmod tempor invidunt ut labore et " +
9         "dolore magna aliquyam erat, sed diam voluptua. " +
10        "At vero eos et accusam et justo duo dolores et " +
11        "ea rebum.");
12    //Zeilenumbruch wird eingeschaltet
13    textfeld.setLineWrap(true);
14
15    //Zeilenumbrueche erfolgen nur nach ganzen Woertern
16    textfeld.setWrapStyleWord(true);
17
18    //Ein JScrollPane, der das Textfeld beinhaltet, wird
    erzeugt
19    JScrollPane scrollpane = new JScrollPane(textfeld);

```

## Demo

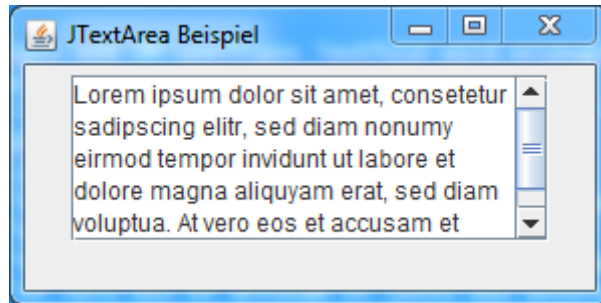


Abb.: JTextArea

<http://http://java-tutorial.org/>

## Demo

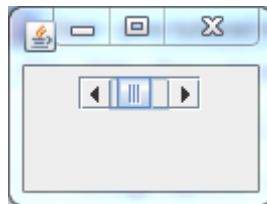


Abb.: JScrollbar

<http://http://java-tutorial.org/>

## JScrollbar

Schieberegler zum Scrollen.



```

1
2      //JScrollbar wird erzeugt
3      JScrollbar scrollbar = new JScrollbar
4          (JScrollbar.HORIZONTAL, 30, 10, 0, 100);

```

## JSlider

Schieberegler, der mit einer Skala versehen werden kann.

- Lautstärkeregler

```

1
2      // JSlider-Objekt wird erzeugt
3      JSlider meinSlider = new JSlider();
4
5      // Mindestwert wird gesetzt
6      meinSlider.setMinimum(0);
7      // Maximalwert wird gesetzt
8      meinSlider.setMaximum(20);
9
10     // Die Abstände zwischen den
11     // Teilmarkierungen werden festgelegt
12     meinSlider.setMajorTickSpacing(5);
13     meinSlider.setMinorTickSpacing(1);
14
15     // Standardmarkierungen werden erzeugt
16     meinSlider.createStandardLabels(1);
17
18     // Zeichnen der Markierungen wird aktiviert
19     meinSlider.setPaintTicks(true);
20
21     // Zeichnen der Labels wird aktiviert
22     meinSlider.setPaintLabels(true);
23
24     // Schiebepalken wird auf den Wert 9 gesetzt
25     meinSlider.setValue(9);

```

## Demo

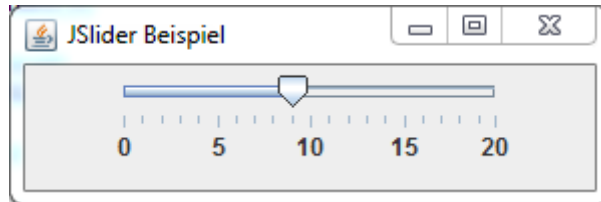


Abb.: JSlider

<http://http://java-tutorial.org/>

## Demo

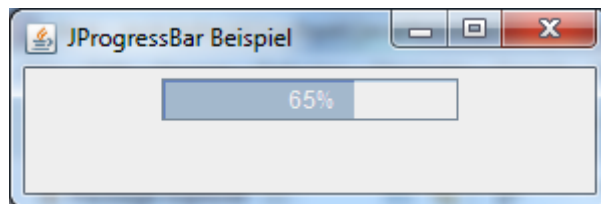


Abb.: JProgressBar

<http://http://java-tutorial.org/>

## JProgressBar

## Fortschrittsbalken

- für Download oder Update
- Bearbeitung von längeren Berechnungen

```
1
2 // JProgressBar-Objekt wird erzeugt
3 JProgressBar meinLadebalken = new JProgressBar(0, 100);
4
5 // Wert fuer den Ladebalken wird gesetzt
6 meinLadebalken.setValue(0);
7
8 // Der aktuelle Wert wird als
9 // Text in Prozent angezeigt
10 meinLadebalken.setStringPainted(true);
```

## JSpinner

Ähnlich der JComboBox, allerdings klappt die Liste nicht auf, sondern die Navigation durch die Liste erfolgt über Pfeiltasten.

- Warenkorb

```
1
2 //JSpinner wird erzeugt
3 JSpinner spinner = new JSpinner();
```

## Demo

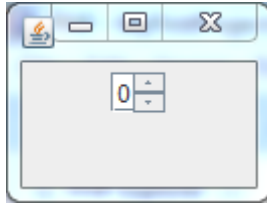


Abb.: JSpinner  
<http://http://java-tutorial.org/>

## Demo

Abb.: JSeparator  
<http://http://java-tutorial.org/>

## JSeparator

## einfache Trennlinie

- optische Abgrenzung zwischen Blöcken

```
1
2 // Erzeugung eines Objektes der Klasse JSeparator
3 JSeparator sep = new JSeparator();
```

## Anhang

Vielen Dank für Ihre Aufmerksamkeit.  
Fragen?

## Link zur Präsentation

- <https://github.com/derdanu/akad-jav02-beamer>



## Quellen

- <http://docs.oracle.com/javase/7/docs/>
- <http://java-tutorial.org/bedienelemente.html>