

GUI – Programmierung

mit

Swing



Stephanie Böhning

Programmieren 2 - Media Systems

Hochschule für Angewandte Wissenschaften Hamburg

Motivation

„Es muss schön sein und sich gut anfühlen“

- Usability ist sehr wichtig
- Programme müssen übersichtlich und schön/cool sein, damit sie genutzt werden
- Konsole ist für Nerds ;-)

Nach welchen Kriterien wählt ihr eure Programme/Spiele aus?

- ...

Möglichkeiten der graphischen Programmierung in Java

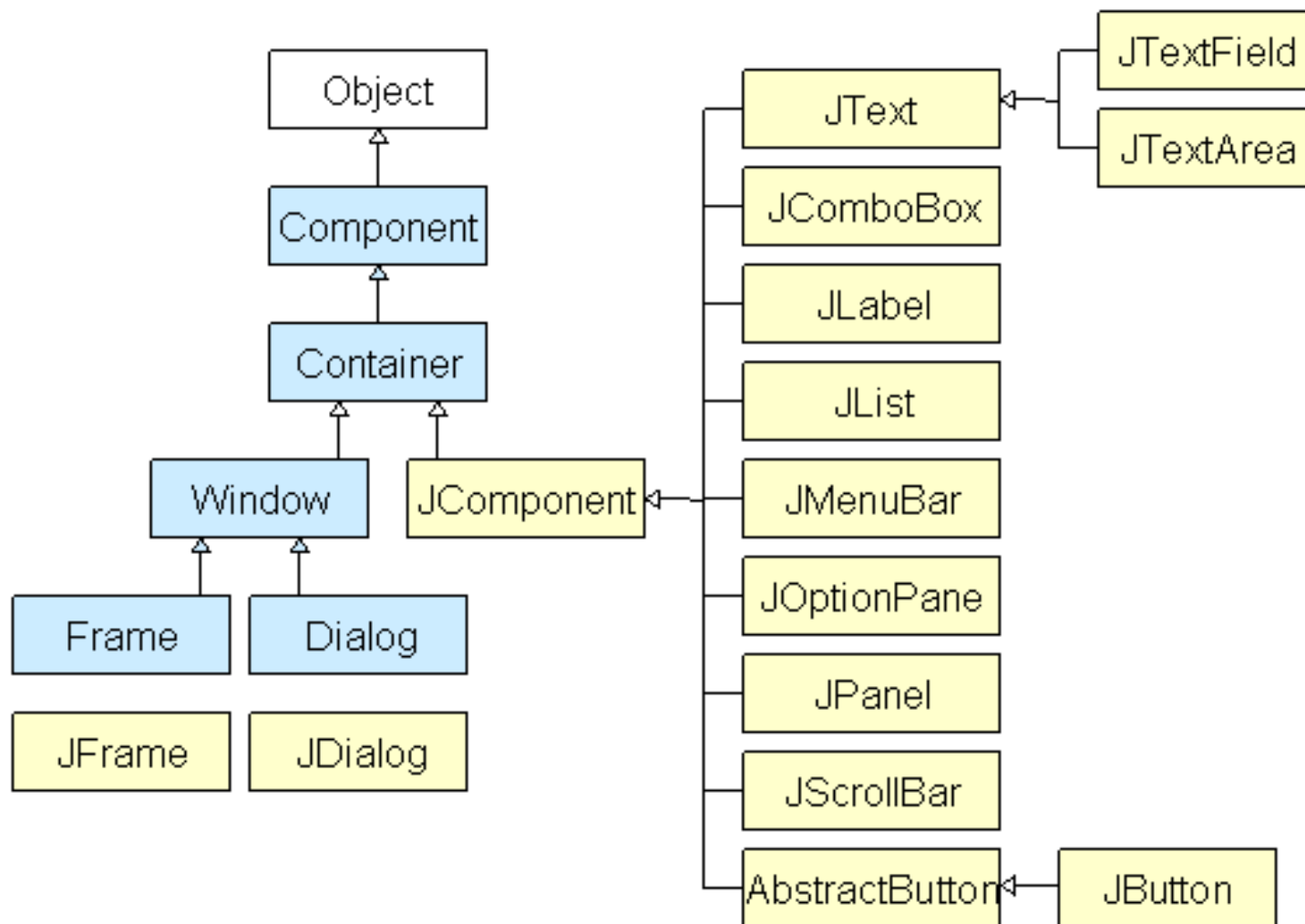
- Man nutzt Graphiken aus vorhandenen Bibliotheken
- Man zeichnet Formen mit Computer
- Man entwirft Sachen per Hand und bindet Sie als Bilder ein

Swing

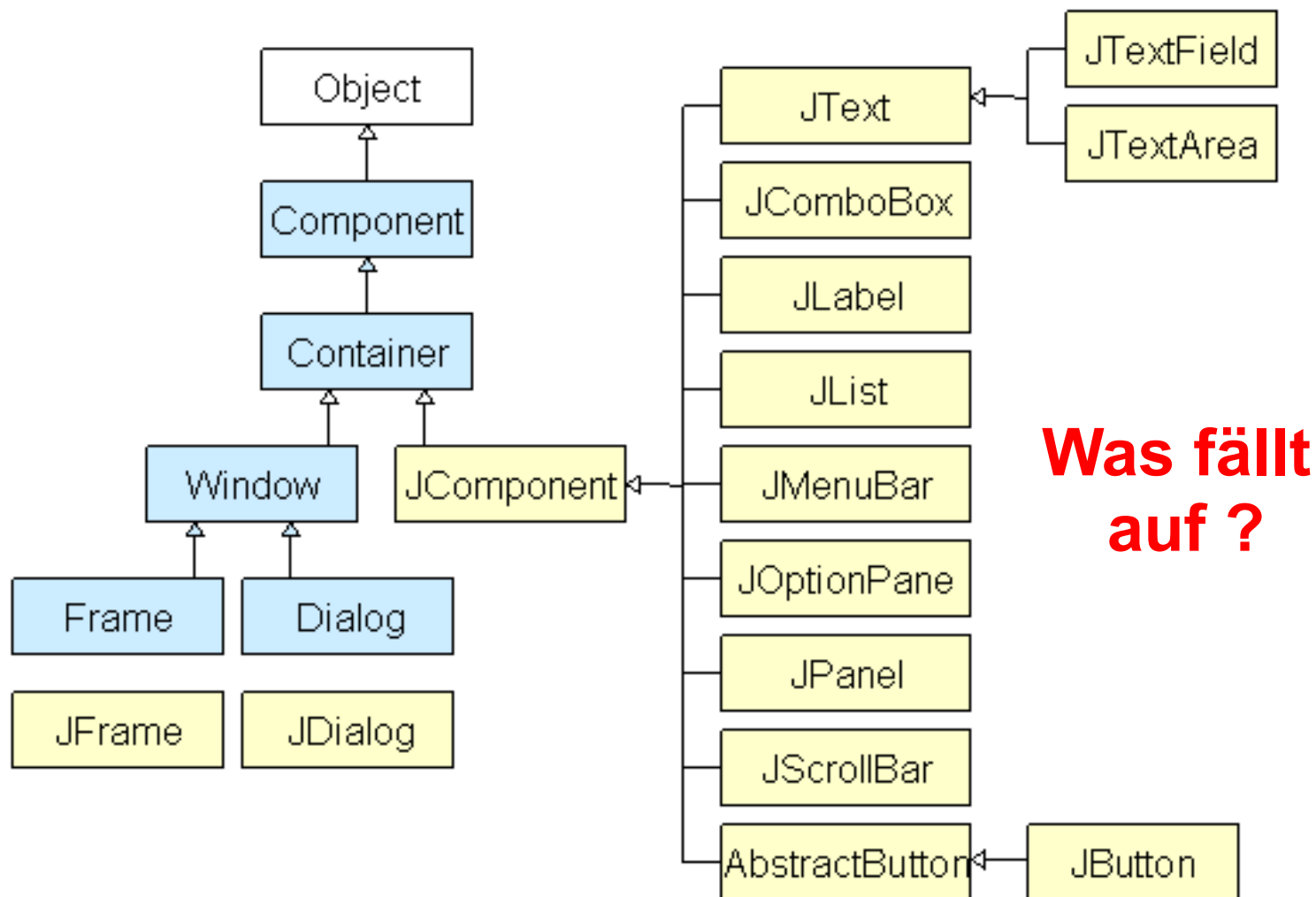
Graphische Bibliotheken Java Foundation Classes (JFC):

- AWT (Vorgänger von Swing)
- Swing
- Java2D

Swing



Swing



Unsere erste kleine GUI

```
import javax.swing.JFrame;

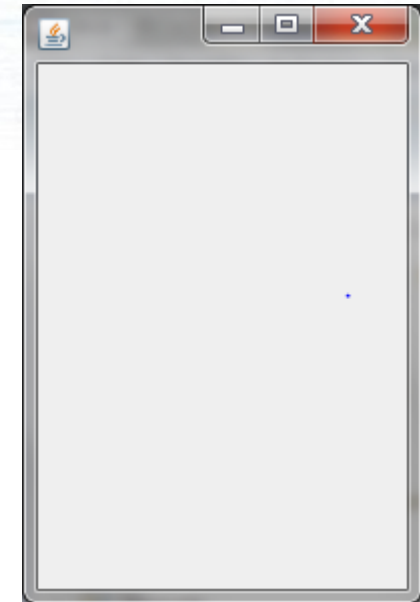
public class MyFirstGUI {

    public static void main(String[] args) {

        JFrame myFrame = new JFrame();

        myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        myFrame.setSize(200, 300);
        myFrame.setVisible(true);

    }
}
```

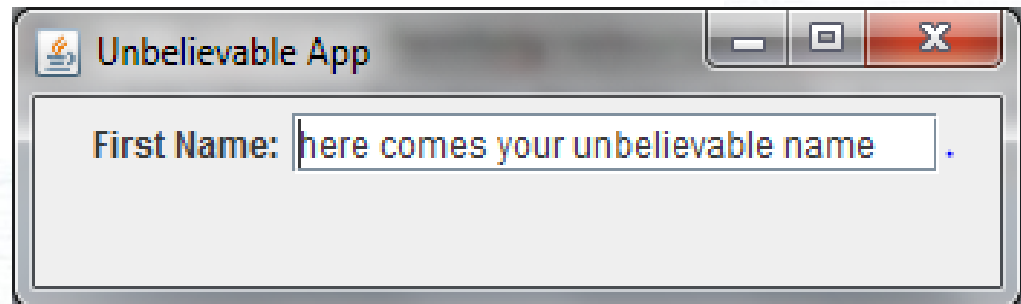


Container

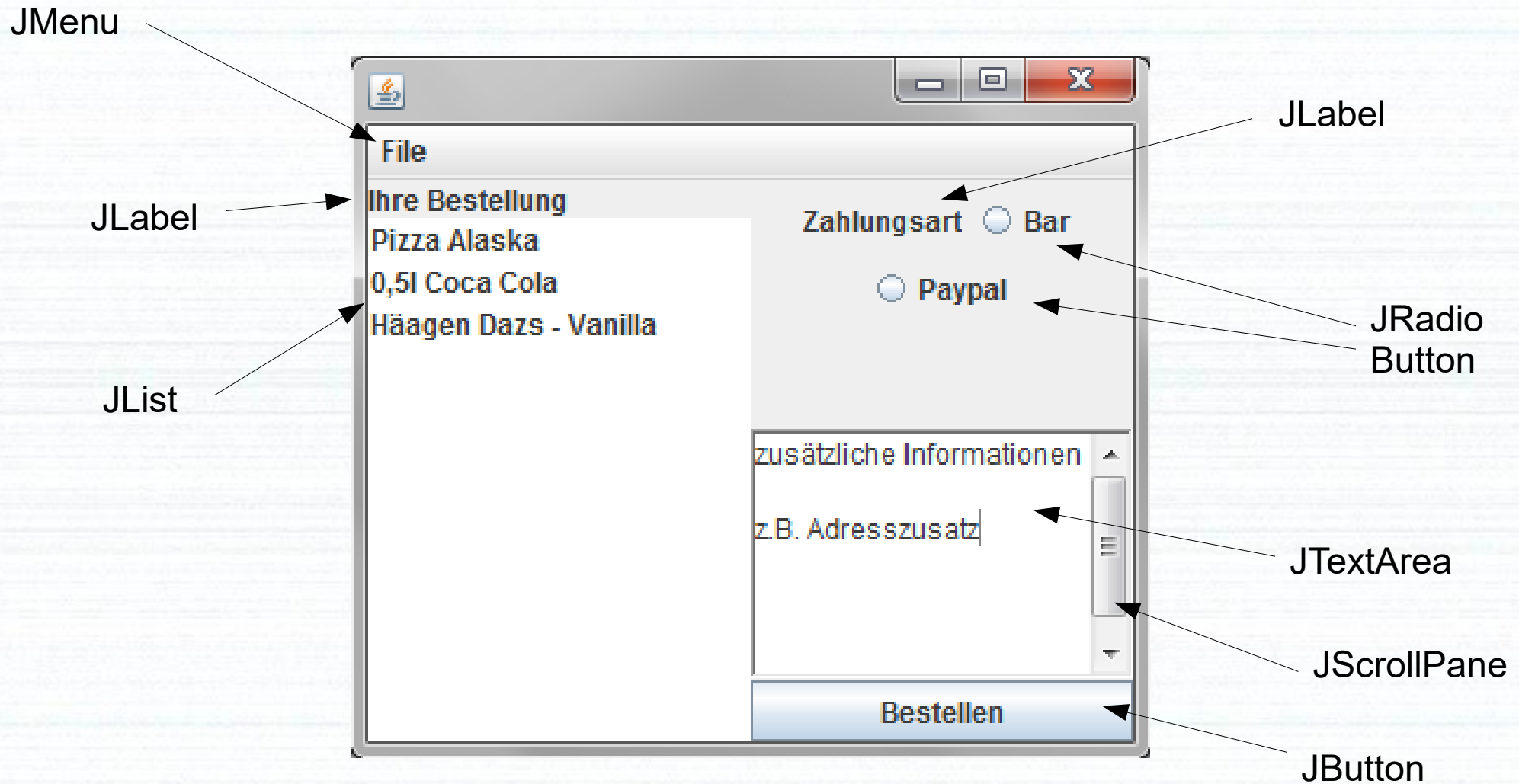
- Top-Level Container: JFrame, JDialog, JApplet, JWindow
- Light-weight-Container: JPanel
- Layouts: BoxLayout, BorderLayout, FlowLayout, GridLayout

Kleine GUI mit Komponenten

```
public class MyFirstFrameWithComponents extends JFrame{  
  
    public MyFirstFrameWithComponents() {  
  
        //create components  
        JTextField nameFld = new JTextField("here comes your unbelievable name", 20);  
  
        //set title and layout  
        setTitle("Unbelievable App");  
        setLayout(new FlowLayout());  
  
        //add components  
        add(new JLabel("First Name:"));  
        add(nameFld);  
  
        setDefaultCloseOperation(EXIT_ON_CLOSE);  
        setSize(350, 100);  
        setVisible(true);  
    }  
}
```



Wichtige Komponenten

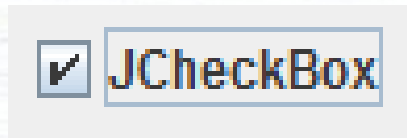


Sonstige Komponenten

- JComboBox, ausklappbares Auswahlmenü



- JCheckBox



- JTextField, einzeliliges Textfeld

- JProgressBar

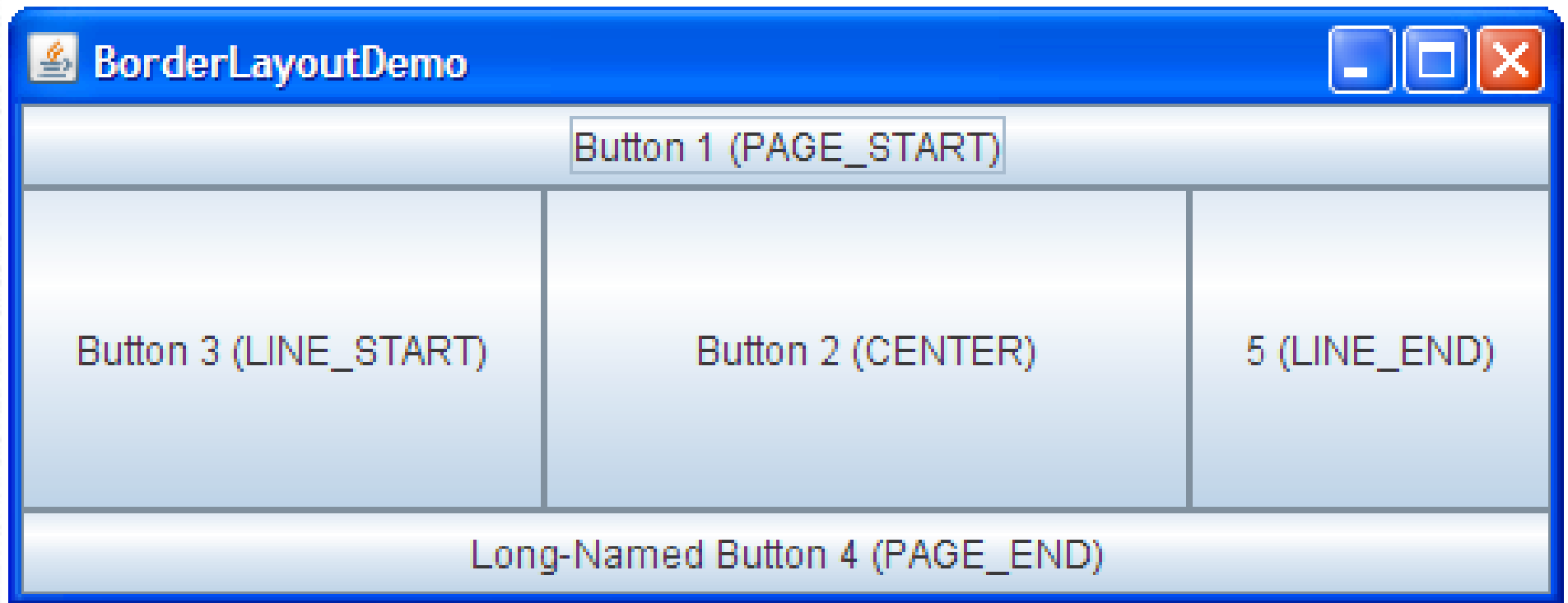
- JSpinner



Layouts

- Gibt die Struktur der GUI vor
- Komponenten werden entsprechend einem Layout angeordnet
- Können beliebig mit einander kombiniert werden, dadurch kann man schicke und komplexe GUIs erstellen

BorderLayout



BorderLayout

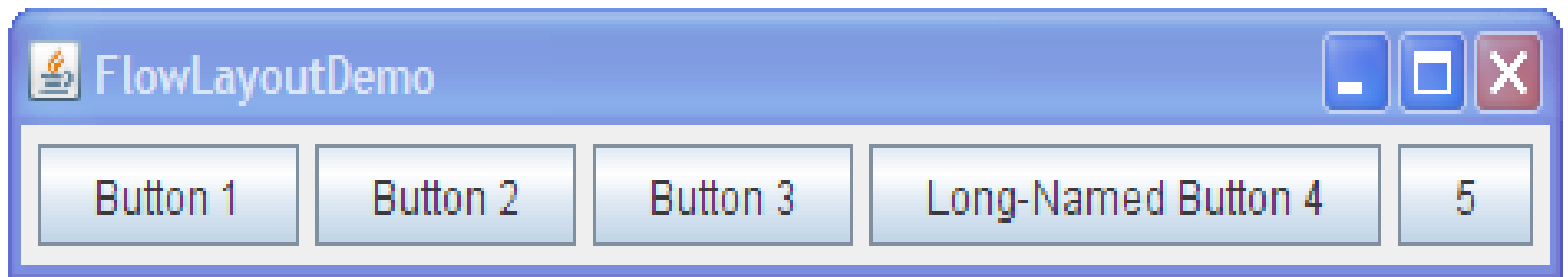
- DefaultLayout für JFrames und JDialogs
- NORTH, SOUTH, CENTER, EAST, WEST

- Beispiel

```
setLayout(new BorderLayout());  
add(btn, BorderLayout.SOUTH);           oder  
add(btn, BorderLayout.PAGE_END);
```

- Achtung, gibt man der Komponente im BorderLayout keinen Platz, wird sie im CENTER platziert und überschreibt ggf. andere Komponenten

FlowLayout



FlowLayout

- Horizontale aneinander Reihung von Komponenten
- Komponenten werden defaultmäßig mittig angeordnet
- Was passiert, wenn nicht alle Komponenten in eine Zeile passen?
- Beispiel:

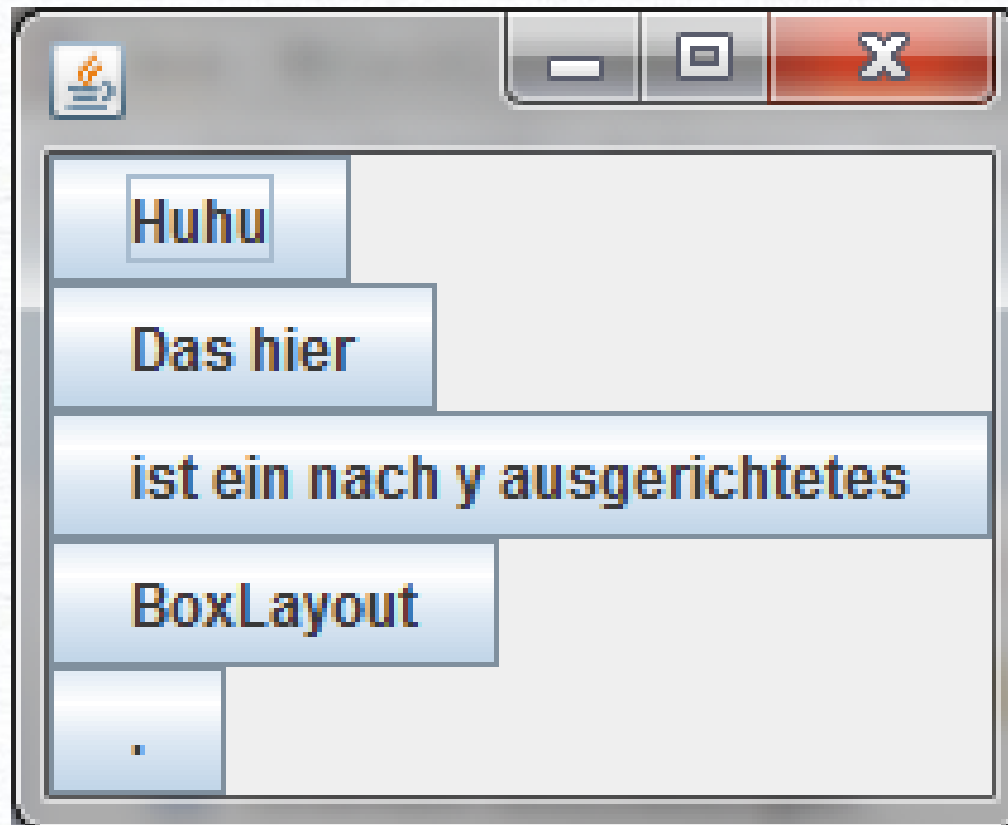
```
setLayout(new FlowLayout(FlowLayout.RIGHT));  
add(btn);
```


FlowLayout

- Horizontale aneinander Reihung von Komponenten
- Komponenten werden defaultmäßig mittig angeordnet
- Was passiert, wenn nicht alle Komponenten in eine Zeile passen? → **es geht in der nächsten Zeile weiter**
- Beispiel:

```
setLayout(new FlowLayout(FlowLayout.RIGHT));  
add(btn);
```

BoxLayout



BoxLayout Variante 1

- Vertikale oder horizontale Ausrichtung
- Man kann Platzhalter zwischen Komponenten hinzufügen
- Beispiel:

```
setLayout(new BoxLayout(getContentPane(),  
                          BoxLayout.Y_AXIS));  
add(btn);
```


BoxLayout Variante 2 mit Struts

//legt eine horizontale Box an

```
Box box = Box.createHorizontalBox();
```

//fügt Komponenten zu der Box hinzu

```
box.add(saveBtn);
```

```
box.add(Box.createHorizontalStrut(5)); oder
```

```
box.add(Box.createHorizontalGlue());
```

```
box.add(cancelBtn);
```

//fügt die Box dem Layout hinzu

```
add (box);
```

GridLayout

Button 1	Button 2
Button 3	Long-Named Button 4
5	

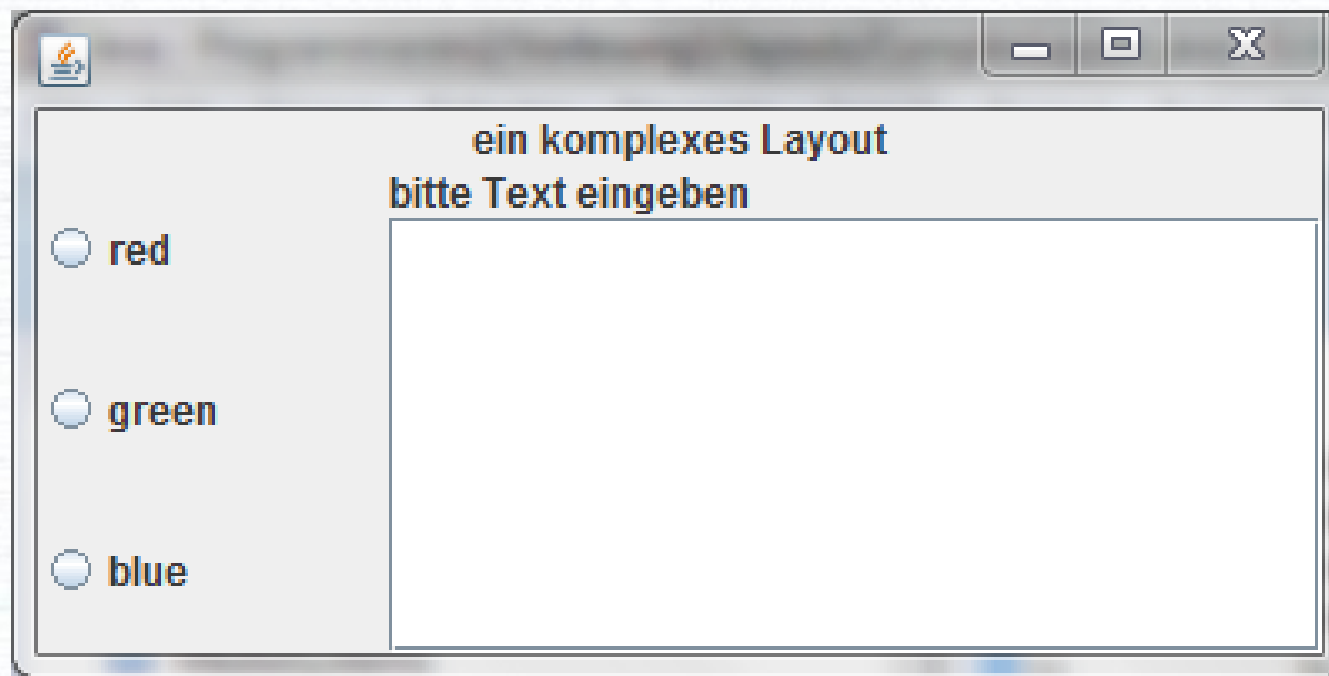
GridLayout

- GridLayout mit festgelegter Anzahl Zeilen und Spalten
- Beispiel:

```
setLayout(new GridLayout(3, 3)); //row, columns  
add(btn);
```

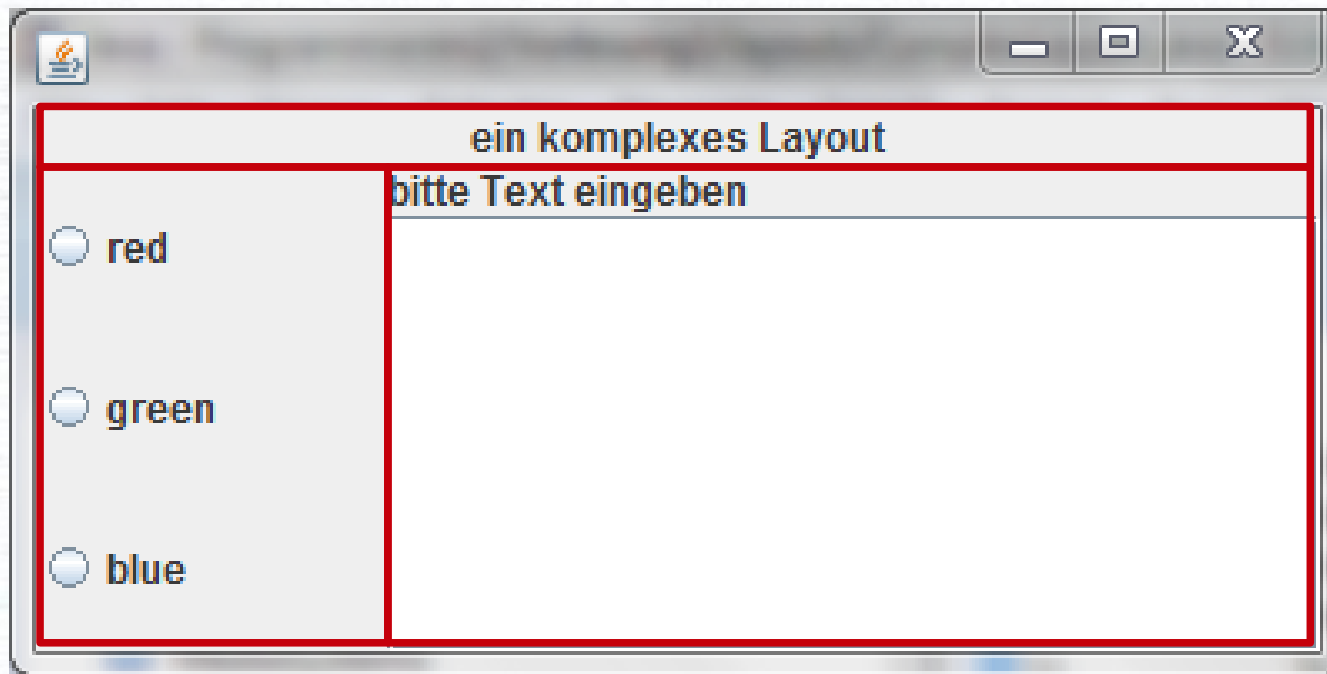

Aufgabe

Wie ist folgendes Layout aufgebaut? Setzt euch in 2-3er Gruppen zusammen und diskutiert.



Aufgabe

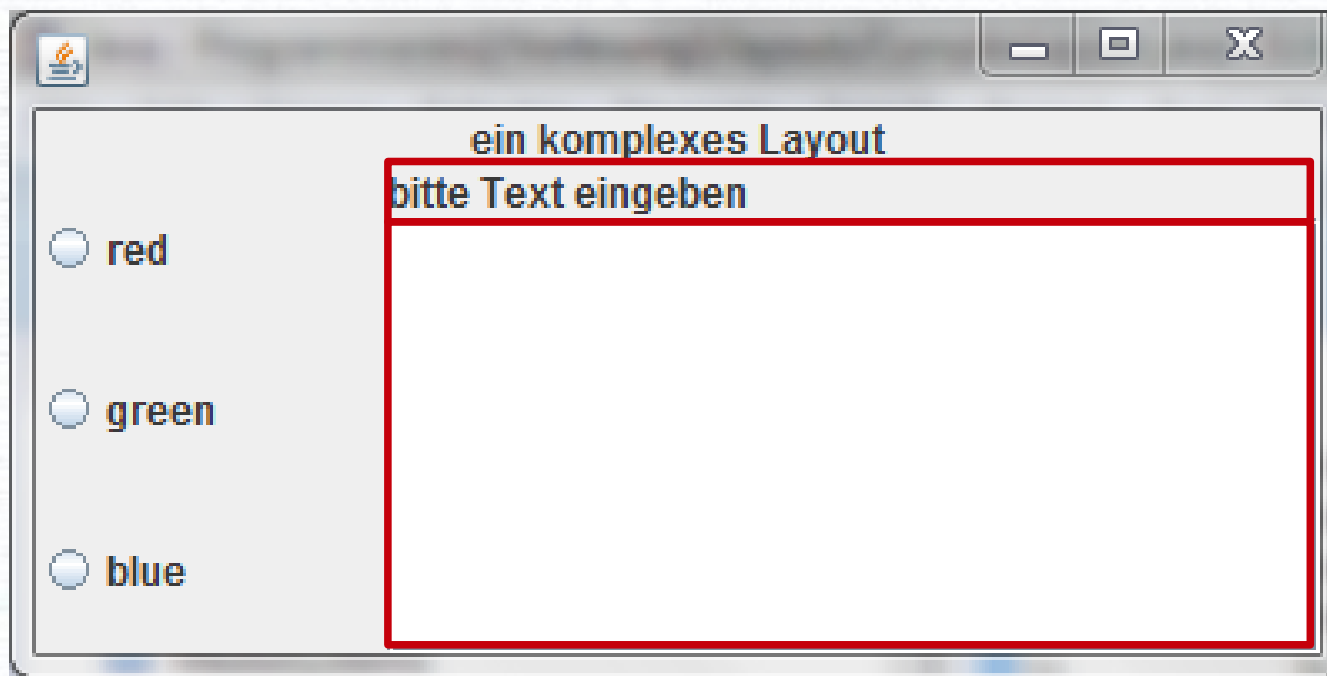
Wie ist folgendes Layout aufgebaut? Setzt euch in 2-3er Gruppen zusammen und diskutiert.



**Border-
Layout**

Aufgabe

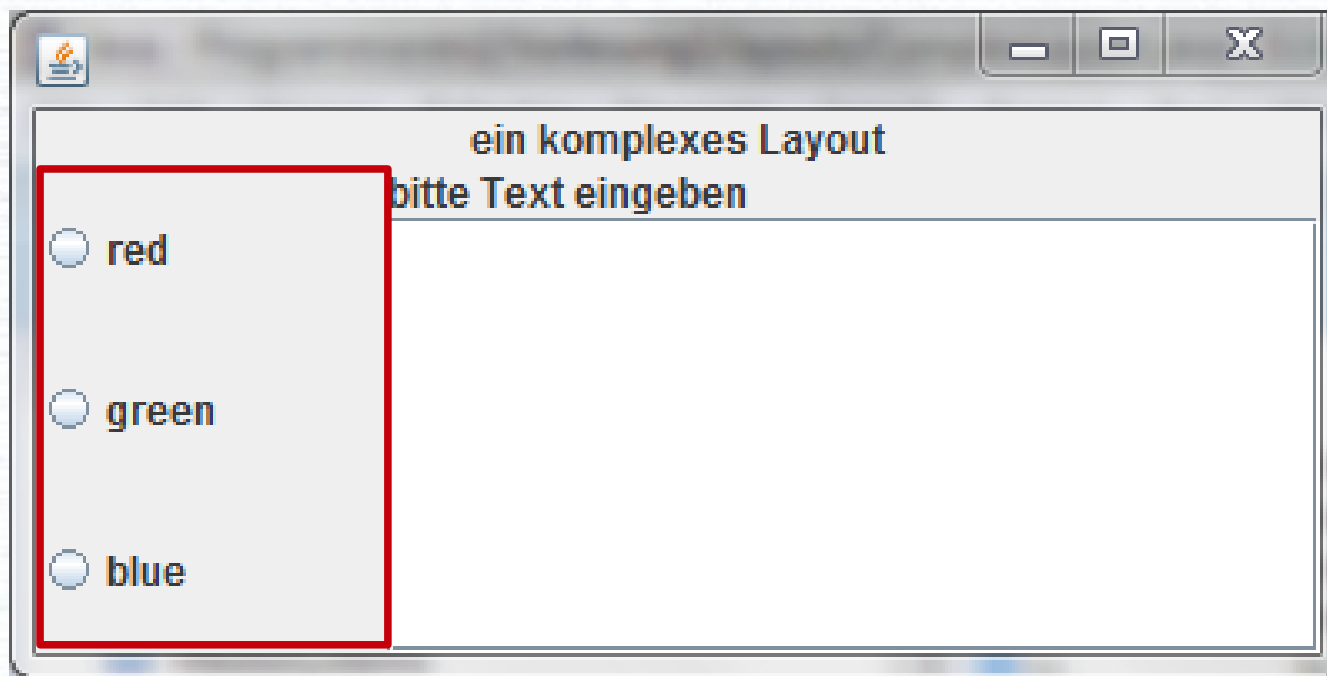
Wie ist folgendes Layout aufgebaut? Setzt euch in 2-3er Gruppen zusammen und diskutiert.



**Border-
Layout**

Aufgabe

Wie ist folgendes Layout aufgebaut? Setzt euch in 2-3er Gruppen zusammen und diskutiert.



**Grid-
Layout**

```

public ComplexLayoutExample(String text) {
    super (text);

    setLayout(new BorderLayout());

    //left side with 3 radio buttons
    Box box = Box.createVerticalBox();
    JRadioButton radio1 = new JRadioButton("red");
    JRadioButton radio2 = new JRadioButton("green");
    JRadioButton radio3 = new JRadioButton("blue");
    box.add(radio1);
    box.add(radio2);
    box.add(radio3);

    //center panel
    JPanel panel = new JPanel();
    panel.setLayout(new BorderLayout());
    JLabel demand = new JLabel("bitte Text eingeben");
    JTextArea textField = new JTextArea(5, 10);
    panel.add(demand, BorderLayout.NORTH);
    panel.add(textField, BorderLayout.CENTER);

    JLabel title = new JLabel("ein komplexeres Layout");
    title.setHorizontalAlignment(JLabel.CENTER);
    add(panel, BorderLayout.CENTER);
    add(box, BorderLayout.WEST);
    add(title, BorderLayout.NORTH);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    pack();
    setVisible(true);
}

```

Gibt es

Fragen ? ? ?