

Java

Prof. Dr. Stefan Enderle
NTA Isny

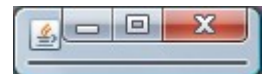
7. Oberflächen- programmierung

JFrame

- Die Klasse `JFrame` bietet eine Basis für Programme mit graphischer Benutzeroberfläche.
- Das folgende Programm legt ein Objekt der Klasse `JFrame` an und zeigt dieses mittels `setVisible(true)` an.

```
import javax.swing.*;
import java.awt.*;

public class Frame
{
    public static void main(String[] args)
    {
        JFrame Rahmen = new JFrame();
        Rahmen.setVisible(true);
    }
}
```

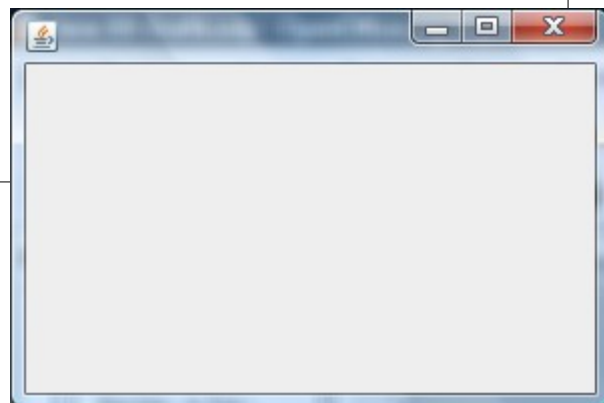


JFrame – Größe festlegen

- Die Größe eines JFrame Objektes kann mit `setSize(x-Size, y-Size)` verändert werden.
- Das folgende Programm zeigt einen JFrame mit 300x200 Pixeln Größe

```
import javax.swing.*;
import java.awt.*;

public class Frame
{
    public static void main(String[] args)
    {
        JFrame Rahmen = new JFrame();
        Rahmen.setSize(300,200);
        Rahmen.setVisible(true);
    }
}
```

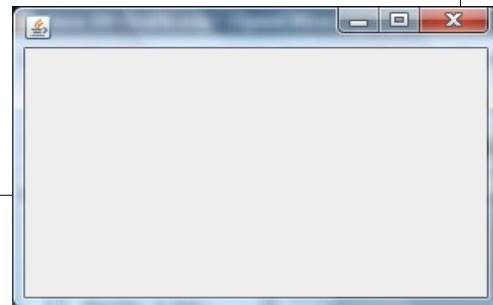


JFrame – Größe fixieren

- Die Größe eines JFrame Objektes kann mit `setResizable(false)` fixiert werden. Das Fenster ist dann nicht vergrößern/kleinern. Default ist `setResizable(true)`.
- Das folgende Programm zeigt einen JFrame mit fester Größe von 300x200 Pixeln

```
import javax.swing.*;
import java.awt.*;

public class Frame
{
    public static void main(String[] args)
    {
        JFrame Rahmen = new JFrame();
        Rahmen.setSize(300,200);
        Rahmen.setResizable(false);
        Rahmen.setVisible(true);
    }
}
```



JFrame – Position

- Die Position eines JFrame Objektes kann mit `setLocation(x-Pos, y-Pos)` verändert werden. Die Position (0,0) ist hierbei links oben.
- Das folgende Programm zeigt einen JFrame an der Stelle 500,500.

```
import javax.swing.*;
import java.awt.*;

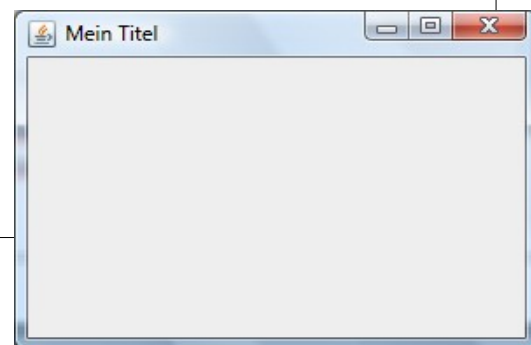
public class Frame
{
    public static void main(String[] args)
    {
        JFrame Rahmen = new JFrame();
        Rahmen.setSize(300,200);
        Rahmen.setLocation(500,500);
        Rahmen.setVisible(true);
    }
}
```

JFrame – Titel

- Der Titel eines JFrame Objektes kann mit `setTitle(titel)` verändert werden. Der Titel wird als Fenstertitel angezeigt.
- Das folgende Programm zeigt einen JFrame mit Titel.

```
import javax.swing.*;
import java.awt.*;

public class Frame
{
    public static void main(String[] args)
    {
        JFrame Rahmen = new JFrame();
        Rahmen.setSize(300,200);
        Rahmen.setLocation(500,500);
        Rahmen.setTitle("Mein Titel");
        Rahmen.setVisible(true);
    }
}
```



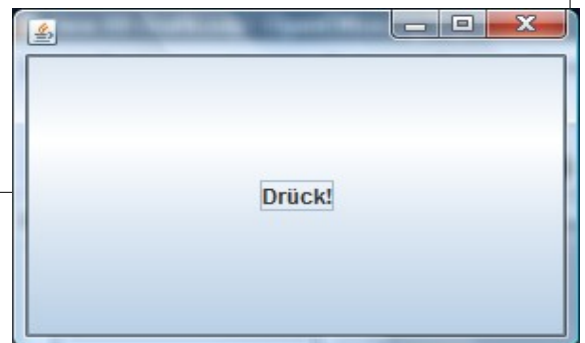
JFrame – Objekte einfügen

- Ein `JFrame` Objekt enthält einen "Container", in den weitere Objekte eingefügt werden können.
- Das folgende Programm zeigt, wie das Container-Objekt geholt und ein neues `JButton` Objekt eingefügt wird.

```
import javax.swing.*;
import java.awt.*;

public class Frame
{
    public static void main(String[] args)
    {
        JFrame Rahmen = new JFrame();
        Rahmen.setSize(300,200);

        JButton button = new JButton("Drück!");
        Container Inhalt = Rahmen.getContentPane();
        Inhalt.add(button);
        Rahmen.setVisible(true);
    }
}
```



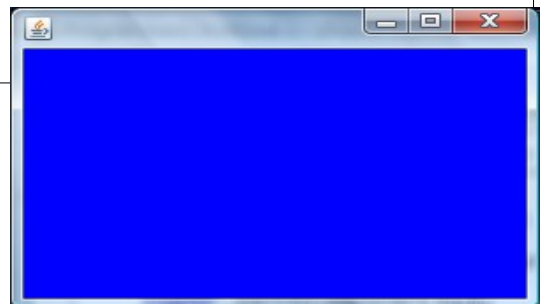
JPanel – als Leinwand

- Die Klasse `JPanel` kann als Leinwand benutzt werden, auf die beliebige graphische Objekte (Linien, Kreise, Rechtecke) gemalt werden können.
- Ein Objekt der Klasse `JPanel` muss wie oben in ein `JFrame` Objekt eingefügt werden.
- Das folgende Programm zeigt, wie ein `JPanel` Objekt eingefügt wird und `setBackground(color)` aufgerufen wird, um die Hintergrundfarbe zu ändern.

```
import javax.swing.*;
import java.awt.*;

public class Frame
{
    public static void main(String[] args)
    {
        JFrame Rahmen = new JFrame();
        Rahmen.setSize(300,200);

        Container Inhalt = Rahmen.getContentPane();
        JPanel panel = new JPanel();
        panel.setBackground(Color.blue);
        Inhalt.add(panel);
        Rahmen.setVisible(true);
    }
}
```



Color – Farben

•Die Klasse `Color` stellen folgende Standard-Farben zur Verfügung:

- `black` (schwarz)
- `green` (grün)
- `red` (rot)
- `blue` (blau)
- `lightGray` (hellgrau)
- `white` (weiß)
- `cyan` (cyan)
- `magenta` (magenta)
- `yellow` (gelb)
- `darkGray` (dunkelgrau)
- `orange` (orange)
- `gray` (grau)
- `pink` (pink)

•Die Standard-Farben werden folgendermaßen verwendet (Beispiel):

```
Color myColor1 = Color.blue;
```

•Eigene Farben können als RGB-Wert definiert werden:

```
Color myColor2 = new Color(17, 123, 240);
```

JPanel – malen

- Um auf einem Panel zu malen, muss eine eigene Klasse definiert werden, die von `JPanel` ableitet:

```
class MeinPanel extends JPanel
{
    ...
}
```

- Diese Klasse muss dann die Methode `paintComponent(Graphics g)` zur Verfügung stellen. Diese wird automatisch immer dann aufgerufen, wenn das Fenster neu gemalt werden soll.
- Das folgende Programm zeigt, wie eine eigene Klasse `MeinPanel` von `JPanel` Objekt abgeleitet wird und die Methode `paintComponent` überschrieben wird, um eigene Graphik-Objekte darzustellen.

JPanel – malen

```
import javax.swing.*;
import java.awt.*;

class MeinPanel extends JPanel
{
    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);
        setBackground(Color.blue);
        g.setColor(new Color (255,0,255));
        g.fillRect(70,50,80,80);
        g.setColor(Color.yellow);
        g.drawLine(0,0,100,100);
        g.setColor(Color.green);
        g.drawOval(50,50,30,30);
    }
}

public class Frame
{
    public static void main(String[] args)
    {
        JFrame Rahmen = new JFrame();
        Rahmen.setSize(300,200);

        Container Inhalt = Rahmen.getContentPane();
        MeinPanel a = new MeinPanel();
        Inhalt.add(a);
        Rahmen.setVisible(true);
    }
}
```

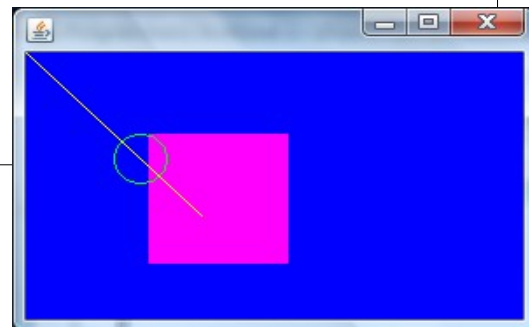


Image – Bilder anzeigen

```
import javax.swing.*;
import java.awt.*;

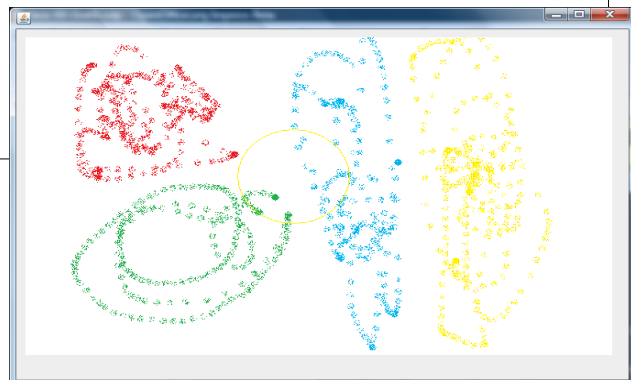
class MeinPanel extends JPanel
{
    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);

        Image img = Toolkit.getDefaultToolkit().
            getImage("test.gif");
        g.drawImage(img,10,10,this);
    }
}

public class Test
{
    public static void main(String[] args)
    {
        JFrame rahmen = new JFrame();
        rahmen.setSize(500,400);

        MeinPanel panel = new MeinPanel();
        rahmen.add(panel);

        rahmen.setVisible(true);
    }
}
```



Menü

- JMenuBar: Leiste mit Hauptmenü
- JMenu: Hauptmenü-Einträge (Datei, Bearbeiten,...)

```
import javax.swing.*;
import java.awt.*;

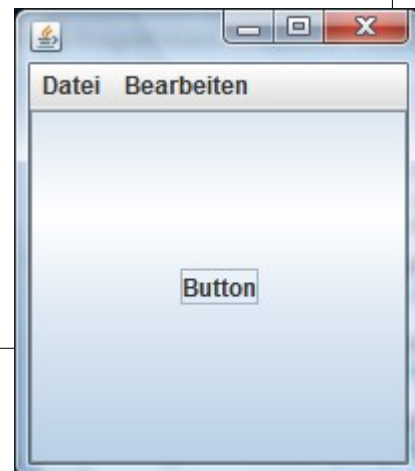
public class Test
{
    public static void main(String[] args)
    {
        JFrame rahmen = new JFrame();
        rahmen.setSize(500,400);
        rahmen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JMenuBar menuBar = new JMenuBar();
        rahmen.setJMenuBar(menuBar);

        JMenu datei      = new JMenu("Datei");
        JMenu bearbeiten = new JMenu("Bearbeiten");
        menuBar.add(datei);
        menuBar.add(bearbeiten);

        JButton button = new JButton("Button");
        rahmen.add(button);

        rahmen.setVisible(true);
    }
}
```



Menü

- JMenuItem: Wie Button, aber im Menü eingesetzt

```
import javax.swing.*;
import java.awt.*;

public class Test
{
    public static void main(String[] args)
    {
        JFrame rahmen = new JFrame();
        rahmen.setSize(500,400);
        rahmen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JMenuBar menuBar = new JMenuBar();
        rahmen.setJMenuBar(menuBar);

        JMenu datei      = new JMenu("Datei");
        JMenu bearbeiten = new JMenu("Bearbeiten");
        menuBar.add(datei);
        menuBar.add(bearbeiten);

        datei.add(new JMenuItem("Neu"));
        datei.add(new JMenuItem("Öffnen"));
        datei.addSeparator();
        datei.add(new JMenuItem("Ende"));

        JButton button = new JButton("Button");
        rahmen.add(button);

        rahmen.setVisible(true);
    }
}
```



Menü

•AbstractAction:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;

public class Test
{
    public static void main(String[] args)
    {
        final JFrame rahmen = new JFrame();
        rahmen.setSize(500,400);
        rahmen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

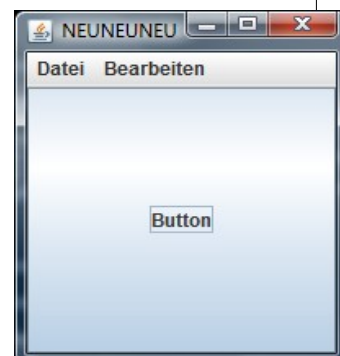
        JMenuBar menuBar = new JMenuBar();
        rahmen.setJMenuBar(menuBar);

        JMenu datei      = new JMenu("Datei");
        JMenu bearbeiten = new JMenu("Bearbeiten");
        menuBar.add(datei);
        menuBar.add(bearbeiten);

        datei.add(new JMenuItem("Neu"));
        datei.add(new JMenuItem("Öffnen"));
        datei.addSeparator();
        datei.add(new AbstractAction("Titel ändern") {
            public void actionPerformed(ActionEvent e) {
                rahmen.setTitle("NEUNEUNEU");
            }
        });

        JButton button = new JButton("Button");
        rahmen.add(button);

        rahmen.setVisible(true);
    }
}
```



GridLayout

- Alle Grid-Zellen gleich groß!

```
import javax.swing.*;
import java.awt.*;

public class Test
{
    public static void main(String[] args)
    {
        JFrame rahmen = new JFrame();
        rahmen.setSize(500,400);
        rahmen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        rahmen.setLayout(new GridLayout(4, 1));

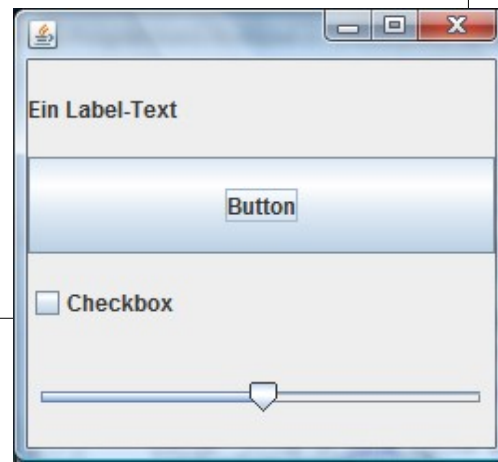
        JLabel label = new JLabel("Ein Label-Text");
        rahmen.add(label);

        JButton button = new JButton("Button");
        rahmen.add(button);

        JCheckBox checkbox = new JCheckBox("Checkbox");
        rahmen.add(checkbox);

        JSlider slider = new JSlider(0,100);
        rahmen.add(slider);

        rahmen.setVisible(true);
    }
}
```



GridLayout

- Anstatt 4 Zeilen jetzt 4 Spalten!

```
import javax.swing.*;
import java.awt.*;

public class Test
{
    public static void main(String[] args)
    {
        JFrame rahmen = new JFrame();
        rahmen.setSize(500,400);
        rahmen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        rahmen.setLayout(new GridLayout(1, 4));

        JLabel label = new JLabel("Ein Label-Text");
        rahmen.add(label);

        JButton button = new JButton("Button");
        rahmen.add(button);

        JCheckBox checkbox = new JCheckBox("Checkbox");
        rahmen.add(checkbox);

        JSlider slider = new JSlider(0,100);
        rahmen.add(slider);

        rahmen.setVisible(true);
    }
}
```



GridLayout

- Oder 2 Spalten a 2 Zeilen...

```
import javax.swing.*;
import java.awt.*;

public class Test
{
    public static void main(String[] args)
    {
        JFrame rahmen = new JFrame();
        rahmen.setSize(500,400);
        rahmen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        rahmen.setLayout(new GridLayout(2, 2));

        JLabel label = new JLabel("Ein Label-Text");
        rahmen.add(label);

        JButton button = new JButton("Button");
        rahmen.add(button);

        JCheckBox checkbox = new JCheckBox("Checkbox");
        rahmen.add(checkbox);

        JSlider slider = new JSlider(0,100);
        rahmen.add(slider);

        rahmen.setVisible(true);
    }
}
```

