

Objektorientiertes Programmieren 1

Grafical User Interface 1

Stephan Kessler, BSc in Systems Engineering

Kontakt: stephan.kessler@edu.teko.ch

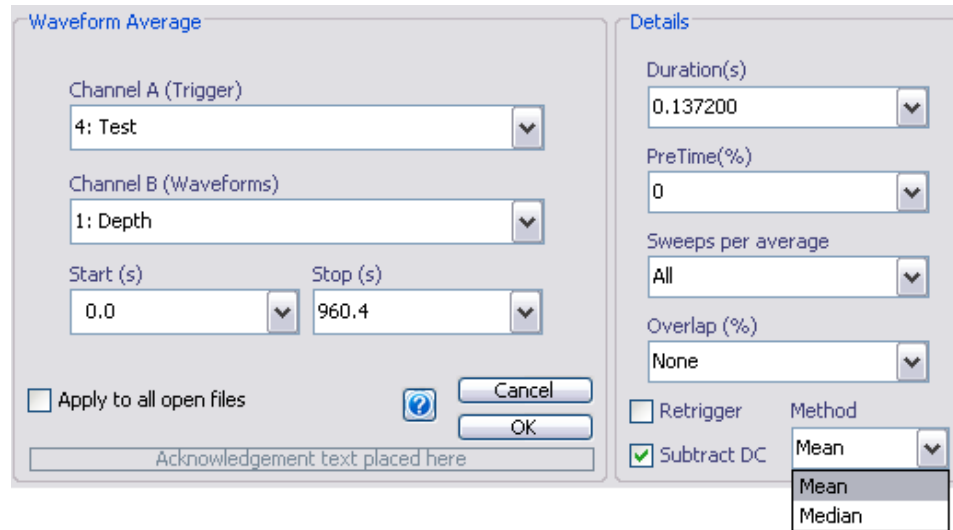


Vorbereitung

Lesen Sie folgende Abschnitte im Buch ***Sprechen Sie Java?*** als Vorbereitung für dieses Thema. Beachten Sie, dass bei Angaben von Unterkapiteln nur diese auch gelesen werden müssen:

- Grafische Benutzeroberfläche
(S. 303-308)

Grafische Benutzerschnittstelle



- Die Grafische Benutzerschnittstelle (Graphical User Interface → GUI) ist eine weit verbreitete Form für die Ein- und Ausgabe von Programmen.
- Die Programme erhalten grafische Elemente wie Knöpfe, Textfelder und Auswahlfelder.
- Bis anhin haben Sie die klassische Konsole-Variante für die Ein- und Ausgabe verwendet (Textform).
- Unerfahrene Benutzer kommen üblicherweise in einem GUI-Programm besser zurecht.
- In Java gibt es mehrere Bibliotheken für die Darstellung von einem GUI. Wir verwenden für den Kurs die *swing*-Bibliothek.

Wie bisher...

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

public class MeinFenster extends JFrame
{

}
```

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

Für die 2D-Grafiken haben wir bis jetzt die *swing*-Bibliothek verwendet. Wir können diesen Aufbau auch für die GUI-Programmierung benutzen.

Zuerst erstellen Sie wie gewohnt ein *JFrame*-Fenster (in diesem Beispiel vererbt auf *MeinFenster*).

Erstes Beispiel

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

public class MeinFenster extends JFrame {
    public JButton button_1;
    public JButton button_2;
    public JTextField text_1;

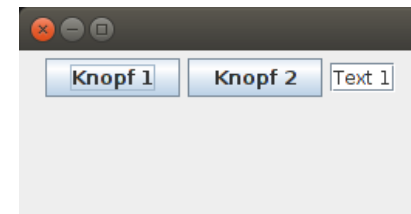
    public MeinFenster(){
        getContentPane().setLayout(new FlowLayout());

        button_1 = new JButton();
        button_2 = new JButton();
        text_1 = new JTextField();

        button_1.setText("Knopf 1");
        button_2.setText("Knopf 2");
        text_1.setText("Text 1");

        getContentPane().add(button_1);
        getContentPane().add(button_2);
        getContentPane().add(text_1);
    }
}
```

Nun haben wir die Möglichkeit unser Fenster mit GUI-Elementen zu erweitern. In diesem Beispiel wurde das Fenster mit 2 Knöpfe und einem Textfeld erweitert.



Erstes Beispiel

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

public class MeinFenster extends JFrame {
    public JButton button_1;
    public JButton button_2;
    public JTextField text_1;

    public MeinFenster(){
        getContentPane().setLayout(new FlowLayout());

        button_1 = new JButton();
        button_2 = new JButton();
        text_1 = new JTextField();

        button_1.setText("Knopf 1");
        button_2.setText("Knopf 2");
        text_1.setText("Text 1");

        getContentPane().add(button_1);
        getContentPane().add(button_2);
        getContentPane().add(text_1);
    }
}
```

Für die Knöpfe wird die Klasse *JButton* verwendet. Das Textfeld benutzt die Klasse *JTextField*.

In der Markierten Zeile werden die Objekt-Variablen entsprechend deklariert.

Erstes Beispiel

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

public class MeinFenster extends JFrame {
    public JButton button_1;
    public JButton button_2;
    public JTextField text_1;

    public MeinFenster(){
        getContentPane().setLayout(new FlowLayout());

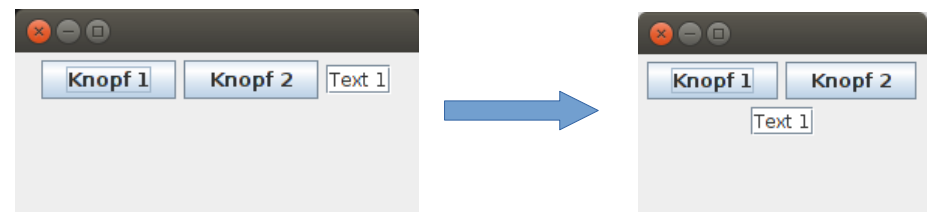
        button_1 = new JButton();
        button_2 = new JButton();
        text_1 = new JTextField();

        button_1.setText("Knopf 1");
        button_2.setText("Knopf 2");
        text_1.setText("Text 1");

        getContentPane().add(button_1);
        getContentPane().add(button_2);
        getContentPane().add(text_1);
    }
}
```

Für die Darstellung der GUI-Elemente können verschiedene Verhaltensregeln verwendet werden. In diesem Beispiel wird die Regel bzw. das Layout *FlowLayout* verwendet. Dadurch werden die Elemente nacheinander dargestellt.

Ist das Ende einer „Zeile“ erreicht wird das nächste Element auf der nächsten „Zeile“ dargestellt.



Erstes Beispiel

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

public class MeinFenster extends JFrame {
    public JButton button_1;
    public JButton button_2;
    public JTextField text_1;

    public MeinFenster(){
        getContentPane().setLayout(new FlowLayout());

        button_1 = new JButton();
        button_2 = new JButton();
        text_1 = new JTextField();

        button_1.setText("Knopf 1");
        button_2.setText("Knopf 2");
        text_1.setText("Text 1");

        getContentPane().add(button_1);
        getContentPane().add(button_2);
        getContentPane().add(text_1);
    }
}
```

getContentPane() gibt den gesetzten „ContentPain“ (Pain → Scheibe), der die Formatierung des Fensters regelt, zurück.

Dieses Objekt besitzt eine Methode *setLayout()* womit die Layout-Regel gesetzt werden kann.

Erstes Beispiel

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

public class MeinFenster extends JFrame {
    public JButton button_1;
    public JButton button_2;
    public JTextField text_1;

    public MeinFenster(){
        getContentPane().setLayout(new FlowLayout());

        button_1 = new JButton();
        button_2 = new JButton();
        text_1 = new JTextField();

        button_1.setText("Knopf 1");
        button_2.setText("Knopf 2");
        text_1.setText("Text 1");

        getContentPane().add(button_1);
        getContentPane().add(button_2);
        getContentPane().add(text_1);
    }
}
```

In Java haben Sie die Möglichkeit direkt bei einer Rückgabe des Objektes mit dem Punkt auf das entsprechende Objekt zuzugreifen.

Daher kann in diesem Beispiel direkt die Methode *setLayout()* aufgerufen werden.

Erstes Beispiel

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

public class MeinFenster extends JFrame {
    public JButton button_1;
    public JButton button_2;
    public JTextField text_1;

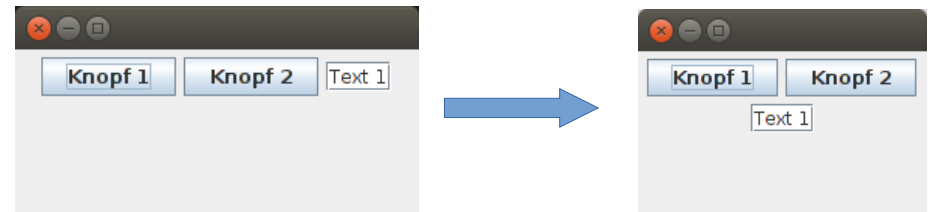
    public MeinFenster(){
        getContentPane().setLayout(new FlowLayout());

        button_1 = new JButton();
        button_2 = new JButton();
        text_1 = new JTextField();

        button_1.setText("Knopf 1");
        button_2.setText("Knopf 2");
        text_1.setText("Text 1");

        getContentPane().add(button_1);
        getContentPane().add(button_2);
        getContentPane().add(text_1);
    }
}
```

Ein neues Objekt der Klasse *FlowLayout* wird instanziiert und direkt als Parameter übergeben. Dieses Objekt regelt die entsprechende Formatierung.



Erstes Beispiel

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

public class MeinFenster extends JFrame {
    public JButton button_1;
    public JButton button_2;
    public JTextField text_1;

    public MeinFenster(){
        getContentPane().setLayout(new FlowLayout());

        button_1 = new JButton();
        button_2 = new JButton();
        text_1 = new JTextField();

        button_1.setText("Knopf 1");
        button_2.setText("Knopf 2");
        text_1.setText("Text 1");

        getContentPane().add(button_1);
        getContentPane().add(button_2);
        getContentPane().add(text_1);
    }
}
```

Eine Übersicht der Layouts kann unter folgendem Link aufgerufen werden:

<https://docs.oracle.com/javase/tutorial/ui/swing/layout/visual.html>

Erstes Beispiel

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

public class MeinFenster extends JFrame {
    public JButton button_1;
    public JButton button_2;
    public JTextField text_1;

    public MeinFenster(){
        getContentPane().setLayout(new FlowLayout());

        button_1 = new JButton();
        button_2 = new JButton();
        text_1 = new JTextField();

        button_1.setText("Knopf 1");
        button_2.setText("Knopf 2");
        text_1.setText("Text 1");

        getContentPane().add(button_1);
        getContentPane().add(button_2);
        getContentPane().add(text_1);
    }
}
```

Im nächsten Schritt können nun die GUI-Elemente instanziiert werden.

Erstes Beispiel

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

public class MeinFenster extends JFrame {
    public JButton button_1;
    public JButton button_2;
    public JTextField text_1;

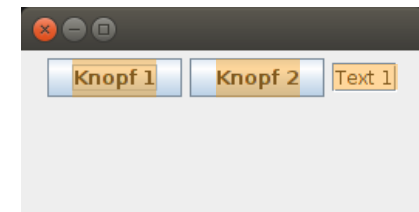
    public MeinFenster(){
        getContentPane().setLayout(new FlowLayout());

        button_1 = new JButton();
        button_2 = new JButton();
        text_1 = new JTextField();

        button_1.setText("Knopf 1");
        button_2.setText("Knopf 2");
        text_1.setText("Text 1");

        getContentPane().add(button_1);
        getContentPane().add(button_2);
        getContentPane().add(text_1);
    }
}
```

Eigenschaften werden über die entsprechenden Methoden gesetzt.



Erstes Beispiel

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

public class MeinFenster extends JFrame {
    public JButton button_1;
    public JButton button_2;
    public JTextField text_1;

    public MeinFenster(){
        getContentPane().setLayout(new FlowLayout());

        button_1 = new JButton();
        button_2 = new JButton();
        text_1 = new JTextField();

        button_1.setText("Knopf 1");
        button_2.setText("Knopf 2");
        text_1.setText("Text 1");

        getContentPane().add(button_1);
        getContentPane().add(button_2);
        getContentPane().add(text_1);
    }
}
```

z.B. Können Sie die Grösse eines Elementes mit der Methode *setPreferredSize* setzen. Diese Methode benötigt als Parameter die Klasse *Dimension* welche sie direkt an dieser Stelle instanziiieren können. Als Parameter nimmt *Dimension* wie gewohnt *width* und *height* auf:

setPreferredSize(new Dimension(30, 50));

Weitere Eigenschaften können Sie der Java-Dokumentation der jeweiligen Klasse entnehmen.

Erstes Beispiel

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

public class MeinFenster extends JFrame {
    public JButton button_1;
    public JButton button_2;
    public JTextField text_1;

    public MeinFenster(){
        getContentPane().setLayout(new FlowLayout());

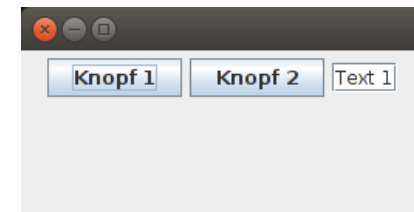
        button_1 = new JButton();
        button_2 = new JButton();
        text_1 = new JTextField();

        button_1.setText("Knopf 1");
        button_2.setText("Knopf 2");
        text_1.setText("Text 1");

        getContentPane().add(button_1);
        getContentPane().add(button_2);
        getContentPane().add(text_1);
    }
}
```

Am Schluss werden die GUI-Elemente dem Content-Pane hinzugefügt.

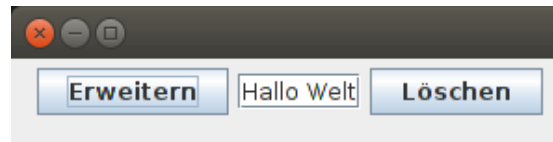
Dabei entspricht die Reihenfolge beim Hinzufügen der Reihenfolge der Darstellung.




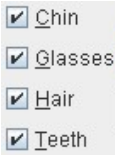
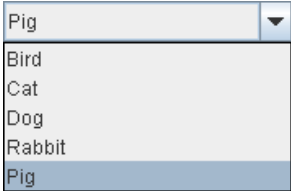
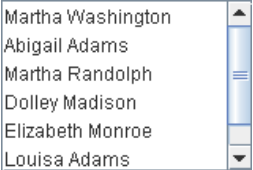
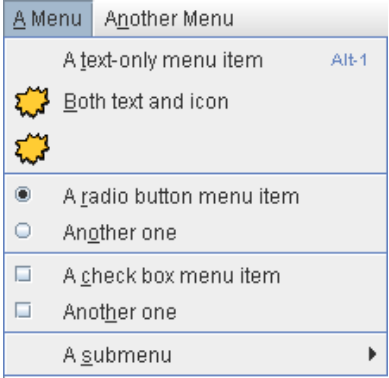
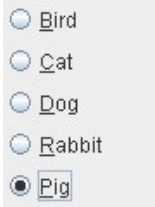







Aufgabe

- Programmieren Sie folgendes GUI-Fenster:



Weitere Swing Elemente

- JButton → 
- JCheckbox → 
- JComboBox → 
- JList → 
- JMenu → 
- JRadioButton → 
- JSlider → 
- JSpinner → 
- JTextField → 
- JPasswordField → 
- JLabel → 

Ereignisse

```
import java.awt.event.ActionEvent;

public class Button1Action extends SimplifiedListener{
    public void actionPerformed(ActionEvent e) {
        System.out.println("Knopf wurde gedrückt");
    }
}
```

Um Ereignisse (z.B. Knopf wird gedrückt) Abfangen zu können, können Sie die aktuelle Version des `SimplifiedListener` verwenden und eine Klasse von dieser erben lassen.

Dadurch können Sie die Methode *`actionPerformed(ActionEvent e)`* überschreiben. Beachten Sie hier, dass Sie *`java.awt.event.ActionEvent`* für den Parameter *`ActionEvent`* importieren.

In diesem Beispiel wurde eine Klasse für den *`button_1`* erstellt. Die Methode *`actionPerformed`* soll nun ausgeführt werden, sobald *`button_1`* gedrückt wird.

Ereignisse

```
import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JTextField;

public class MeinFenster extends JFrame {
    public JButton button_1;
    public JButton button_2;
    public JTextField text_1;

    public MeinFenster(){
        getContentPane().setLayout(new FlowLayout());

        button_1 = new JButton();
        button_2 = new JButton();
        text_1 = new JTextField();
        button_1.setText("Erweitern");
        button_2.setText("Löschen");
        text_1.setText("Hallo Welt");

        button_1.addActionListener(new Button1Action());

        getContentPane().add(button_1);
        getContentPane().add(text_1);
        getContentPane().add(button_2);
    }
}
```

Die Swing-Elemente besitzen jeweils die Methode *addActionListener*. Hier können Sie die erstellte Klasse einfügen.

In diesem Beispiel wird nach dem Drücken von *button_1* der Text „Knopf wurde gedrückt“ ausgeführt.



Aufgabe

- Erweitern Sie Ihr Programm aus der letzten Aufgabe so, dass der Knopf „Löschen“ die Zahl 0 in Ihr Textfeld einfügt.
- Durch drücken auf „Erweitern“ soll sich der Wert im Textfeld um 1 erhöhen (0, 1, 2, 3 usw.)

