

# Objektorientierte Programmierung

## Studiengang Medieninformatik

Hans-Werner Lang

Hochschule Flensburg

Vorlesung 1

15.03.2017

## Objektorientierte Programmierung (Studiengang Medieninformatik)

Form: Vorlesung und Labor, 2+2 SWS, 5cp

Prüfung: Klausur (2 Std.), benotet, Prüfungsleistung

Labor: Pflicht, Voraussetzung für die Anerkennung des Moduls,  
Labortestate, nicht benotet

# Vorlesung

- Inhalt:
- Prinzipien der Objektorientierung  
(Klassen und Objekte, Felder und Methoden, Vererbung und Assoziation, Überschreiben von Methoden, abstrakte Klassen und Interfaces, Klassenmethoden, Typ-Parameter)
  - GUI-Programmierung (Graphical User Interface)
  - Datenstrukturen (z.B. Listen)
  - Design Patterns (z.B. Iterator, Observer)

Programmiersprache: Java

Entwicklungsumgebung: Eclipse

# Vorlesung

Voraussetzungen: Strukturierte Programmierung

- Primitive Datentypen (int, double, boolean)
- Variablen, Ausdrücke, Operatoren
- Deklaration, Wertzuweisung
- Strings, Arrays
- Kommentare
- Steuerkonstrukte (if, while, for)
- Funktionen, Parameterübergabe, Rückgabewert
- ~~Rekursion~~

## Labor

- Durchführung:
- Sie erhalten wöchentlich Aufgabenzettel mit Programmieraufgaben
  - Sie bearbeiten die Aufgaben im Labor (Zweiergruppen zulässig)
  - Alle 2-3 Wochen Abnahme der bearbeiteten Aufgaben (Testat)
- Abnahme:
- Einzel, auch wenn Sie in einer Gruppe arbeiten
  - Aufgabe gelöst, Programm läuft
  - Programm ist selbst programmiert
  - Programm kann erklärt und ggf. modifiziert werden

## Labor

Ziel 1: Spiel *Vier gewinnt* programmieren

- GUI
- KI (Computerspieler)

Ziel 2: Spiel *Einfach Genial* programmieren

- KI (Computerspieler)
- Laborgruppen gegeneinander

# Klassen und Objekte

... und nun geht's los!

# Klassen und Objekte

Klassen legen die Eigenschaften (Attribute) von Objekten fest  
(hier Zähler und Nenner):

```
public class Bruch
{
    int zaehler, nenner;

    // Konstruktor (erzeugt ein Objekt)
    public Bruch(int z, int n)
    {
        zaehler=z;
        nenner=n;
    }
    ...
}
```



## Klassen und Objekte

Objekte sind Elemente der Klasse und besitzen entsprechende Eigenschaften (hier Zähler und Nenner):

```
Bruch a, b, c;  
a=new Bruch(1, 2); // 1/2  
b=new Bruch(2, 3); // 2/3
```

Mit `new` werden neue Objekte der Klasse erzeugt (Aufruf des Konstruktors).

Andere Bezeichnungen für *Objekt*: Instanz (engl. *instance*, Exemplar);  
andere Bezeichnung für *Klasse*: Objekttyp.

# Klassen und Objekte

Klassen legen die anwendbaren Operationen (Methoden) der Objekte fest (hier z.B. Multiplikation)

```
public class Bruch
{
    ...
    // multipliziert den Bruch this mit dem Bruch other
    public Bruch mul(Bruch other)
    {
        int z=this.zaehler*other.zaehler;
        int n=this.nenner*other.nenner;
        return new Bruch(z, n);
    }
}
```

# Klassen und Objekte

Methoden werden auf Objekte angewendet:

```
Bruch a, b, c;
```

```
a=new Bruch(1, 2); // 1/2
```

```
b=new Bruch(2, 3); // 2/3
```

```
c=a.mul(b); // c=a*b
```

# GUI-Klassen und -Objekte

```
import javax.swing.*;

public class Main
{
    public static void main(String[] args)
    {
        JFrame myframe;           // Deklaration
        myframe = new JFrame();   // Objekt erzeugen
        myframe.setSize(500, 300); // Methoden-
        myframe.setVisible(true); // aufrufe
    }
}
```

# Bedeutung

```
JFrame myframe;           // Deklaration  
myframe = new JFrame();   // Objekt erzeugen
```



(Aufruf des Konstruktors der Klasse JFrame)

## Bedeutung

Methodenaufruf:

```
myframe.setSize(500, 300);
```

*setSize* ist eine Methode der Klasse, die auf ein Objekt der Klasse angewendet wird (hier auf das Objekt *myframe*).

In diesem Fall ändert dieser Aufruf der Methode *setSize* den Zustand des Objekts *myframe* in der Weise, dass die Eigenschaften Breite und Höhe neue Werte erhalten.

Breite und Höhe sind im Objekt als sogenannte Attribute gespeichert.

Deklaration von Attributen in der Klasse:

```
private int width, height;
```

Ein Zugriff von außerhalb der Klasse ist nicht möglich, da `private`.

```
myframe.width=400;
```

# Attribute und Methoden

## Attribute

```
private int width, height;  
private boolean visible;
```

## Methoden

```
public void setSize(int w, int h)  
{  
    width=w;  
    height=h;  
}
```

```
public void setVisible(boolean v)  
{  
    visible=v;  
}
```

## Attribute (meistens `private`)

- bestimmen den Zustand des jeweiligen Objekts

## Methoden (meistens `public`)

- werden auf ein Objekt angewandt,
- verändern den Zustand des Objekts,
- geben Auskunft über den Zustand des Objekts oder
- machen irgendetwas anderes mit dem Objekt

Beim nächsten Mal:

Eine Klasse von einer anderen Klasse ableiten:

```
public class MyFrame extends JFrame
```