
# Objektorientierte Programmierung

## Studiengang Medieninformatik

Hans-Werner Lang

Hochschule Flensburg

Vorlesung 13
14.06.2017

Hochschule Flensburg
University of Applied Sciences

# Heute:

- Anonyme Objekte
- Lokale Klassen
- Anonyme Klassen
- Anonyme Funktionen

"Anonym" = ohne Namen

# Anonyme Objekte (1)

Hochschule Flensburg
University of Applied Sciences

Objekt mit Namen *p*:

```
Position p=new Position(2, 3);
System.out.println(p);
```

Objekt ohne Namen (anonymes Objekt):

```
System.out.println(new Position(2, 3));
```

# Anonyme Objekte (2)

Hochschule Flensburg
University of Applied Sciences

Anderes Beispiel:

Objekte mit Namen *p* und *q*:

```
Position p=new Position(2, 3);
Position q=new Position(1, 1);
System.out.println(p.add(q));
```

Objekte ohne Namen (anonyme Objekte):

```
System.out.println(new Position(2, 3).add(new Position(1, 1)));
```

# Anonyme Objekte (3)

Hochschule Flensburg
University of Applied Sciences

Beispiel aus dem Laborprojekt:

```
public static void main(String[] args)
{
    new GameFrame();
}
```

# Lokale Klassen

Hochschule Flensburg
University of Applied Sciences

Beispiel aus dem Laborprojekt: lokale Klasse *MyThread*

```
@Override
public void update(Observable arg0, Object arg1)
{
    if (config.turn==mycolor && !config.gameOver())
    {
        class MyThread extends Thread    // lokale Klasse
        {
            @Override
            public void run()
            {
                makeMove(findMove());
            }
        }
        new MyThread().start();          // anonymes Objekt
    }
}
```

# Lokale Klassen

Hochschule Flensburg
University of Applied Sciences

Eine lokale Klasse ist nur in dem Block erreichbar, in dem sie deklariert worden ist. Außerhalb dieses Blocks ist sie nicht definiert (genauso wie lokale Variablen).

## Anonyme Klassen

Oft benötigt man nur ein einziges Objekt der lokalen Klasse. Dann bietet Java die Möglichkeit, sogar ohne Vergabe eines Klassennamens ein Objekt einer lokalen Klasse zu erzeugen. Die Klasse bleibt sozusagen anonym.

# Anonyme Klassen (1)

Hochschule Flensburg
University of Applied Sciences

Anonyme Klassen haben keinen Namen. Daher haben anonyme  Klassen auch keinen eigenen Konstruktor (der Konstruktor heißt ja immer so wie die Klasse).

Wie erzeugen wir dann ein Objekt  einer anonymen Klasse?

Durch Aufruf des Standardkonstruktors einer Oberklasse, im äußersten Fall der Klasse *Object,* der "Mutter aller Klassen".

Von einer anonymen Klasse lässt sich immer nur ein einziges Objekt erzeugen.

# Anonyme Klassen (2)

Hochschule Flensburg
University of Applied Sciences

Beispiel aus dem Laborprojekt: anonyme Klasse

```
@Override
public void update(Observable arg0, Object arg1)
{
    if (config.turn==mycolor && !config.gameOver())
    {
        Thread th=new Thread()
        {                           // anonyme Klasse, abgeleitet von Thread
            @Override
            public void run()
            {
                makeMove(findMove());
            }
        };
        th.start();
    }
}
```

# Anonyme Klassen (3)

Hochschule Flensburg
University of Applied Sciences

Beispiel aus dem Laborprojekt: anonyme Klasse und anonymes Objekt

```
@Override
public void update(Observable arg0, Object arg1)
{
    if (config.turn==mycolor && !config.gameOver())
    {
        new Thread()                    // anonymes Objekt
        {                               // anonyme Klasse, abgeleitet von Thread
            @Override
            public void run()
            {
                makeMove(findMove());
            }
        }.start();
    }
}
```

# Anonyme Klassen (4)

Hochschule Flensburg
University of Applied Sciences

Häufig angewendet: anonyme Klassen, die Listener erweitern

```
JButton button = new JButton("Klick mich");
button.addActionListener(new ActionListener()     // anonymes Objekt
{                                                 // anonyme Klasse
    public void actionPerformed(ActionEvent evt)
    {
        System.out.println("Button angeklickt");
    }
});
```

# Anonyme Funktionen (1)

Hochschule Flensburg
University of Applied Sciences

Funktion mit Namen *increment*:

```
int increment(int a)
{
    int b=a+1;
    return b;
}
```

Anonyme Funktion in Java 8:

```
(int a) -> {int b=a+1; return b;}
```

Einfacher:

```
(int a) -> {return a+1;}
```

Noch einfacher (mit Typ-Inferenz):

```
(a) -> a+1
```

# Anonyme Funktionen (2)

Hochschule Flensburg
University of Applied Sciences

Anonyme Funktion in der Programmiersprache Python:

```
lambda a : a+1        # bedeutet: a -> a+1
```

Lamda-Kalkül (Church, Kleene 1932) Schreibweise: λa.a+1

Anwendung:

```
print (lambda a : a+1) (3)
4
```

Einfacher:

```
print 3+1
4
```

Anonyme Funktion in der funktionalen Programmiersprache Haskell:

```
(\a -> a + 1) 3
4
```

Einfacher:

```
3+1
4
```

# Anonyme Funktionen (3)

Hochschule Flensburg
University of Applied Sciences

Aber:

Anonyme Funktionen werden  für die Anwendung auf Listen gebraucht, als Parameter für Funktionen höherer Ordnung, z.B. *map*:

Python:

```
print map(lambda a : a+1, [1, 2, 3])
[2, 3, 4]
```

Haskell:

```
map (\a -> a + 1) [1, 2, 3]
[2, 3, 4]
```

# Anonyme Funktionen (4)

Hochschule Flensburg
University of Applied Sciences

Oder *filter*:

Python:

```
print filter(lambda a : a%2==0, [1, 2, 3, 4, 5, 6])
[2, 4, 6]
```

Haskell:

```
filter (\a -> a 'mod' 2==0) [1, 2, 3, 4, 5, 6]
[2, 4, 6]
```

Hochschule Flensburg
University of Applied Sciences

# Die gute Nachricht:

Vorlesung 12 (Threads) und Vorlesung 13 (anonyme Klassen) sind nicht klausurrelevant.

Hochschule Flensburg
University of Applied Sciences

# Das war's!

- Dies war meine letzte Vorlesung an der Hochschule!
- Ihnen viel Erfolg bei den Klausuren und schöne Semesterferien!