



**FH MÜNSTER**  
University of Applied Sciences

# **Wahlpflichtmodul**

## **Fortgeschrittenes Software Engineering**

Prof. Dr.-Ing. Claus Grewe

Fachbereich Wirtschaft

Master of Science Wirtschaftsinformatik

Sommersemester 2023

Methoden und  
Werkzeuge

Software Reverse  
Engineering

Funktionale  
Programmierung

Polyglotte  
Programmierung

Aspektorientierte  
Programmierung

Kritische Systeme

Ansätze verlässlicher  
Programmierung

Bewertung von  
Zuverlässigkeit

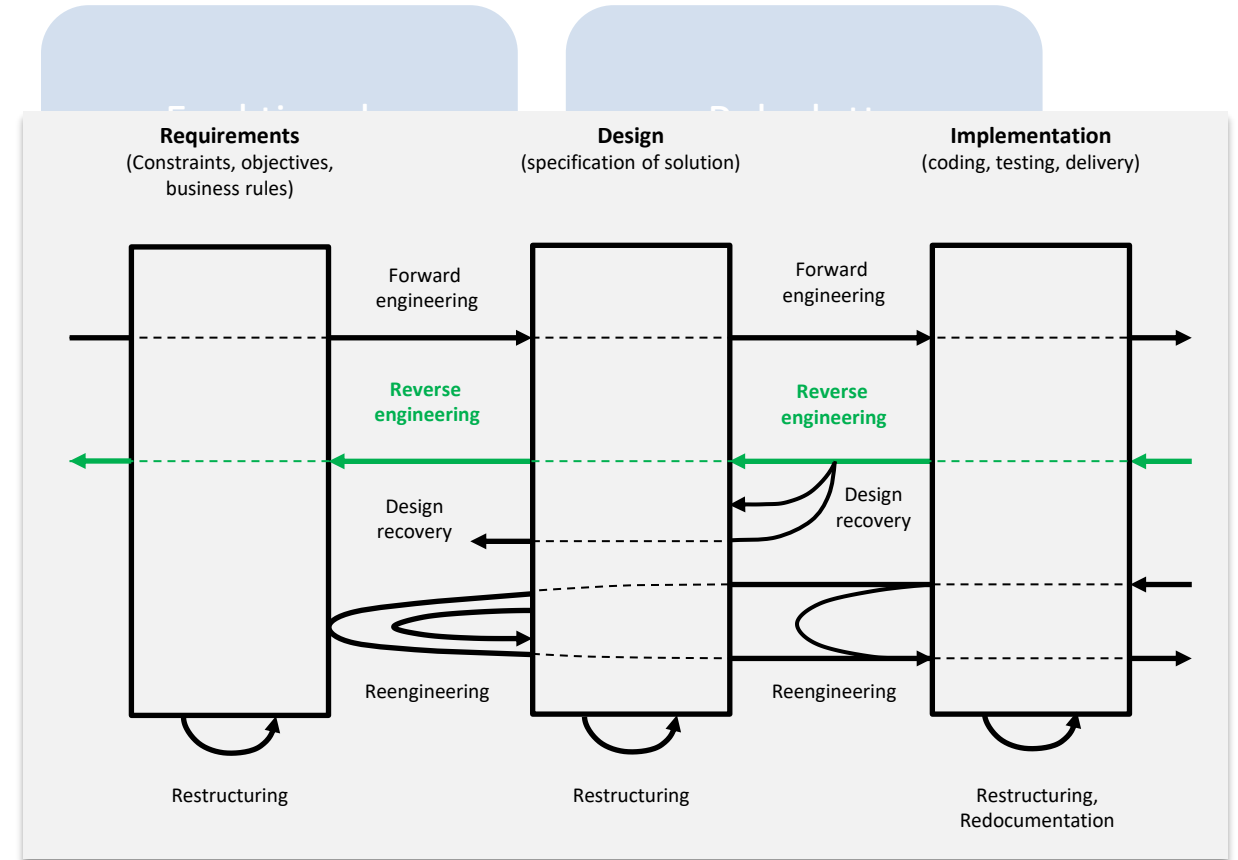
Methoden und  
Werkzeuge

Software Reverse  
Engineering

Aspe  
Prog



the Systeme



Methoden und  
Werkzeuge

Software Reverse  
Engineering

Funktionale  
Programmierung

Polyglotte  
Programmierung

```
List<String> items = List.of("green", "*yellow", "red", "*blue", "black", "white");
```

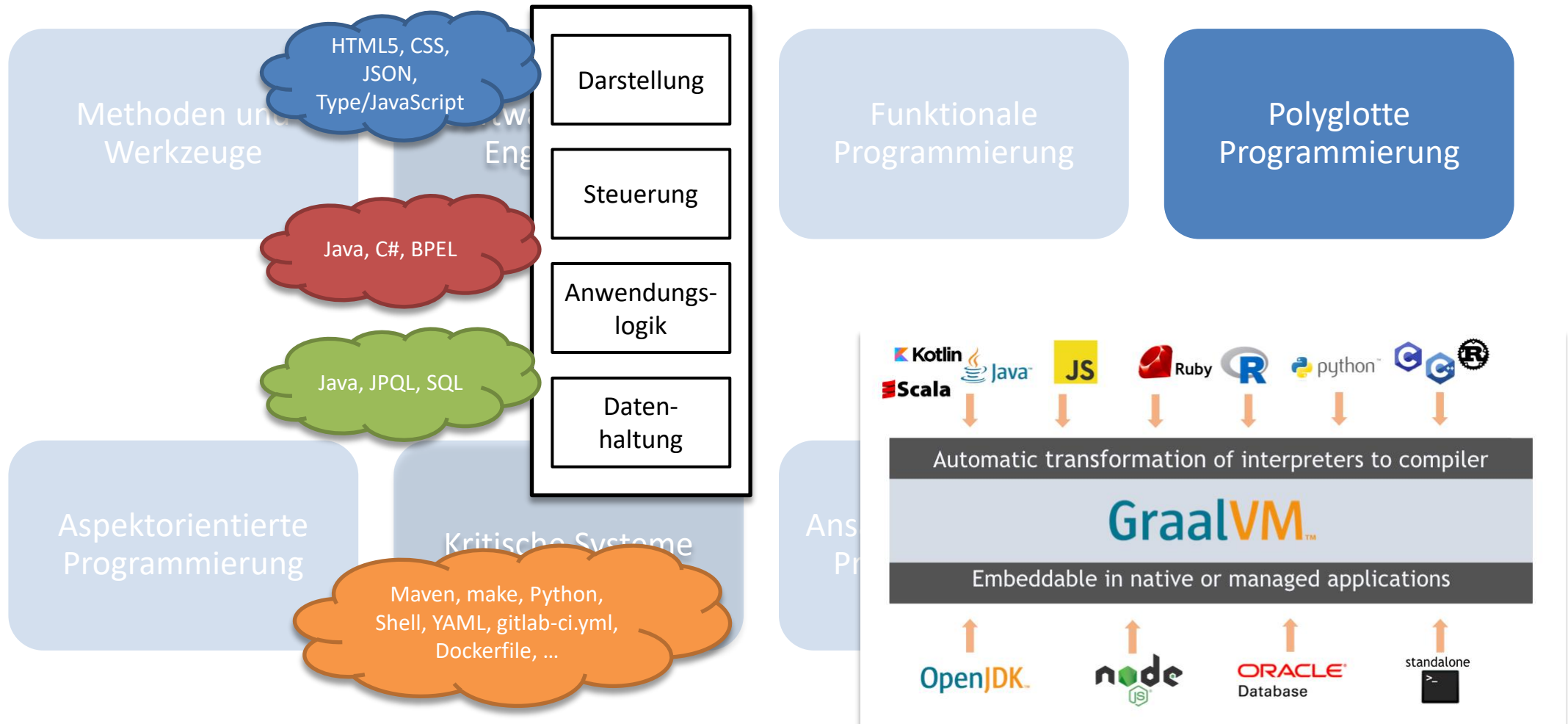
Aspekte  
Program

```
String tmp, result = "";  
for (String item : items) {  
    if (item.startsWith("*")) {  
        tmp = item.replace("*", "");  
        result += tmp + ",";  
    }  
}  
System.out.println(result);
```

Imperativer Stil

```
String result = null;  
result = items.stream()  
    .filter(item -> item.startsWith("*"))  
    .map(item -> item.replace("*", ""))  
    .reduce("", (t, u) -> t + u + ",");  
System.out.println(result);
```

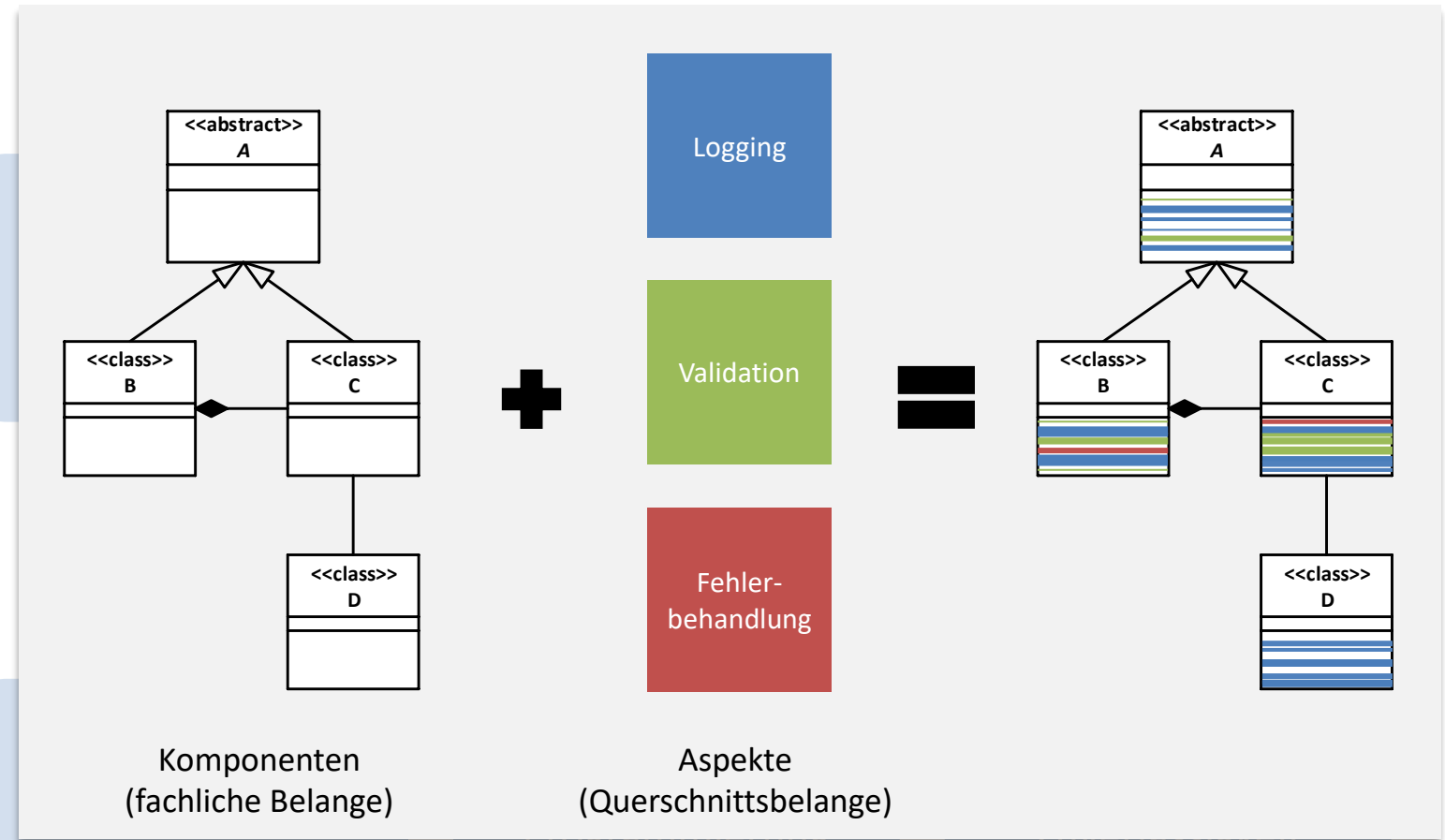
Funktionaler Stil



# Überblick

Methoden und  
Werkzeuge

Aspektorientierte  
Programmierung





Softw  
Eng



unktio  
nprogramm



Aspektorientierte  
Programmierung

Kritische Systeme

Ansätze verlässlicher  
Programmierung

Bewertung von  
Zuverlässigkeit

Java

```
String[] strs = new String[] { "Hund", "Katze", "Maus" };  
for (int i = strs.length; i >= 0; i--) {  
    System.out.println "[" + (i - 1) + "]: " + strs[i - 1]);  
}
```

```
[2]: Maus  
[1]: Katze  
[0]: Hund  
...java.lang.ArrayIndexOutOfBoundsException: -1
```

C

```
char *strs[3] = { "Hund", "Katze", "Maus" };  
int i, len;  
  
len = sizeof(strs) / sizeof(char *);  
for (i = len; i >= 0; i--) {  
    printf("[%d]: %s\n", i-1, strs[i-1]);  
}
```

```
[2]: Maus  
[1]: Katze  
[0]: Hund  
[-1]: [?][?][?][?]
```

Groovy

```
def strs = [ "Hund", "Katze", "Maus" ];  
for (int i = strs.size; i >= 0; i--) {  
    System.out.println "[" + (i - 1) + "]: " + strs[i - 1]);  
}
```

```
[2]: Maus  
[1]: Katze  
[0]: Hund  
[-1]: Maus
```

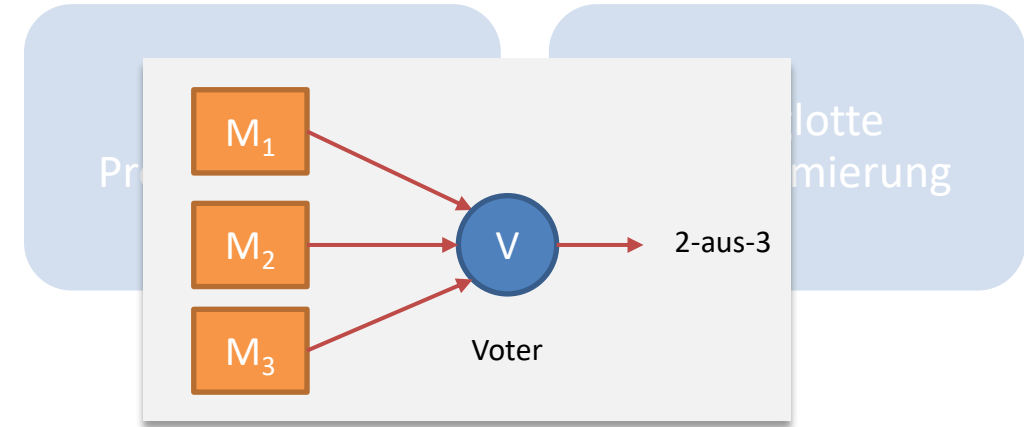
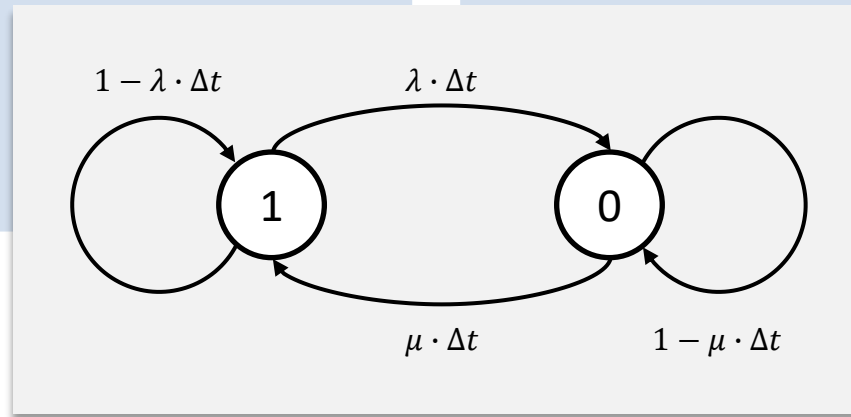
Funktionale  
Programmierung

Polyglotte  
Programmierung

Verlässlicher  
Programmierung

Bewertung von  
Zuverlässigkeit





$$A(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)t}$$

Aspe  
Programmierung

Kritische Systeme

Ansätze verlässlicher  
Programmierung

Bewertung von  
Zuverlässigkeit

# Hinweise zur Veranstaltung

- Vorlesung und Übung: 3 SWS
  
- Prüfungsleistung umfasst (unter Vorbehalt)
  1. Vortrag zu einer technischen Themenstellung
  2. Schriftliche Ausarbeitung zu einer Themenstellung (Fallstudie)

# Abschluss

- Allgemeine Fragen