

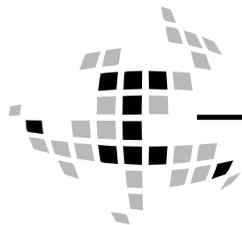
Universität Karlsruhe (TH)

Forschungsuniversität · gegründet 1825

Softwaretechnik 1 Tutorium

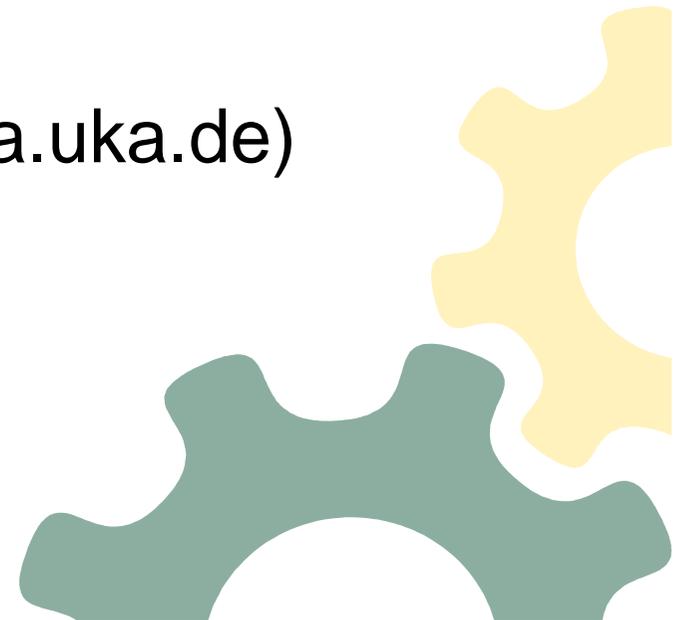
06. Juli 2009

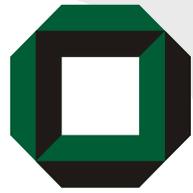
Matthias Thoma (s_thoma@ira.uka.de)



Fakultät für **Informatik**

Lehrstuhl für Programmiersysteme

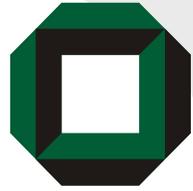




Heute

- Übungsblatt Nr. 4 + Übungsblatt Nr. 5
- Softwarequalität
- Zwischensprache
- Anweisungsüberdeckung
- Zweigüberdeckung
- Pfadüberdeckung
- Bedingungsüberdeckung

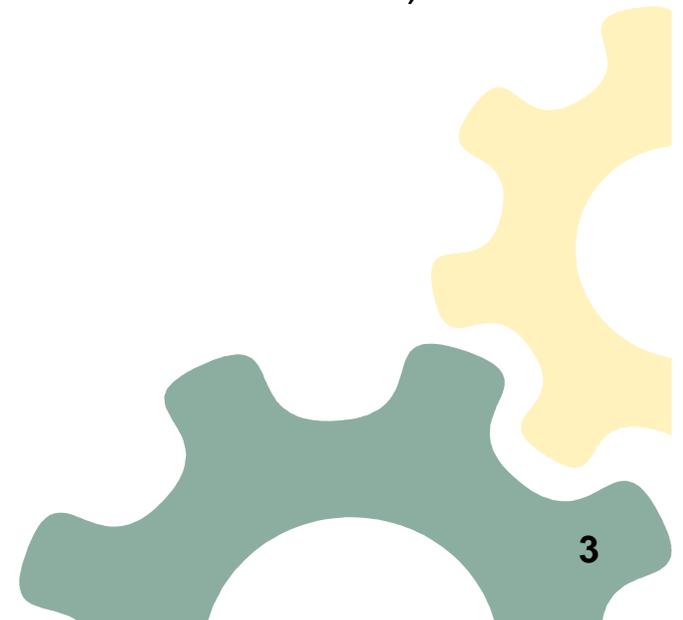
- Hinweise zum nächsten Übungsblatt

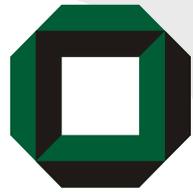


Softwarequalität

- Gesamtheit der Merkmale und Merkmalswerte eines Software-Produkts, die sich auf dessen Eignung beziehen, festgelegte oder vorausgesetzte Erfordernisse zu erfüllen

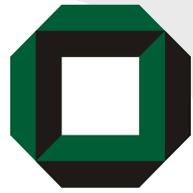
(nach DIN ISO 9126)



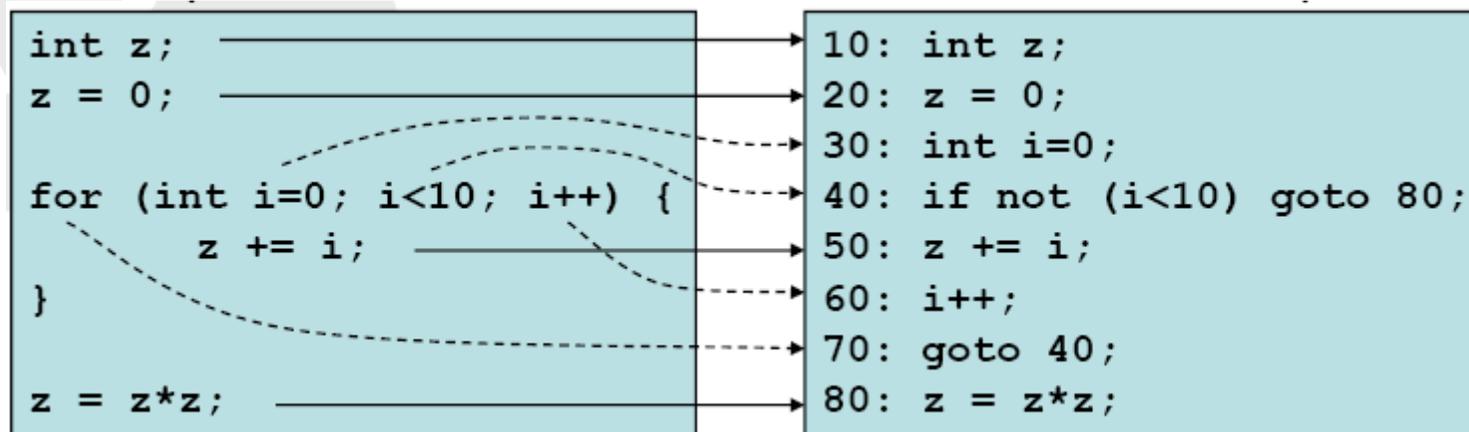


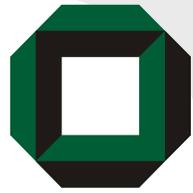
Zwischensprache - kurzgefasst

- Man nehme JAVA oder C++ und ersetze alle Anweisungen, die die Befehlsreihenfolge ändern durch **goto**
- If und andere bedingte Anweisungen werden in `if (bla) goto <Zeilennummer>` ersetzt.
- Die Ausführungsreihenfolge der anderen Befehle bei gleicher Parametrisierung bleibt gleich mit der in der Quellsprache!



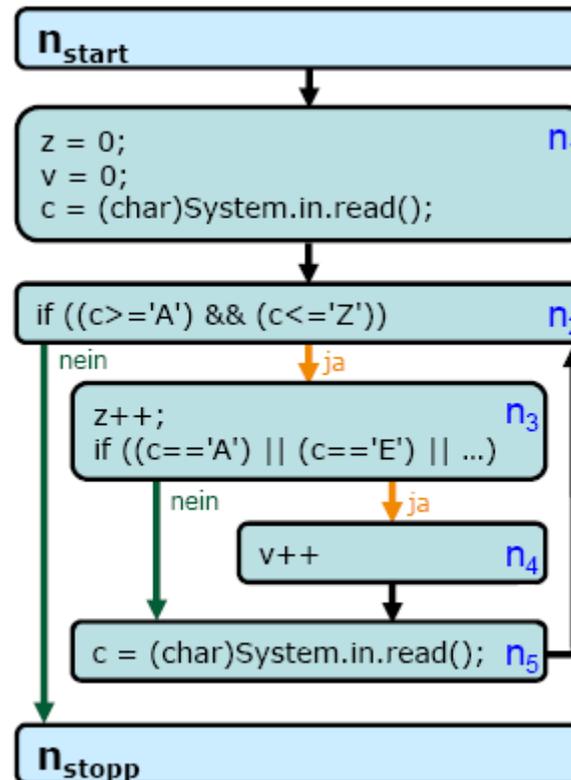
Zwischensprache - kurzgefasst





Grundblöcke

- Ein Grundblock bezeichnet eine maximal lange Folge fortlaufender Anweisungen der Zwischensprache, in die der Kontrollfluss nur am Anfang eintritt und die außer am Ende keine Sprungbefehle enthält.



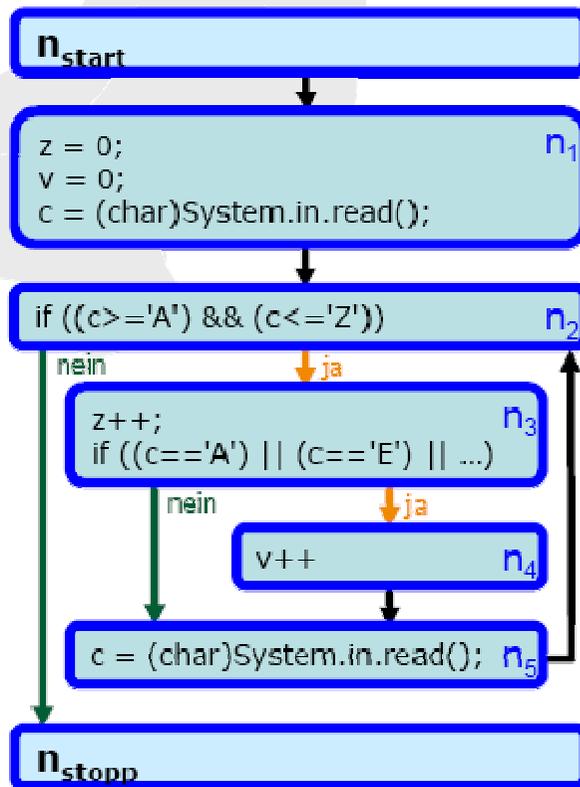


Anweisungs-, Zweig- und Pfadüberdeckung

- Die Teststrategie *Anweisungsüberdeckung* C0 verlangt die Ausführung aller Grundblöcke des Programms P.
- Die *Zweigüberdeckung* C1 verlangt das Traversieren aller Zweige im KFG.
- Die *Pfadüberdeckung* fordert die Ausführung aller unterschiedlichen Pfade im Programm.



Beispiel 1



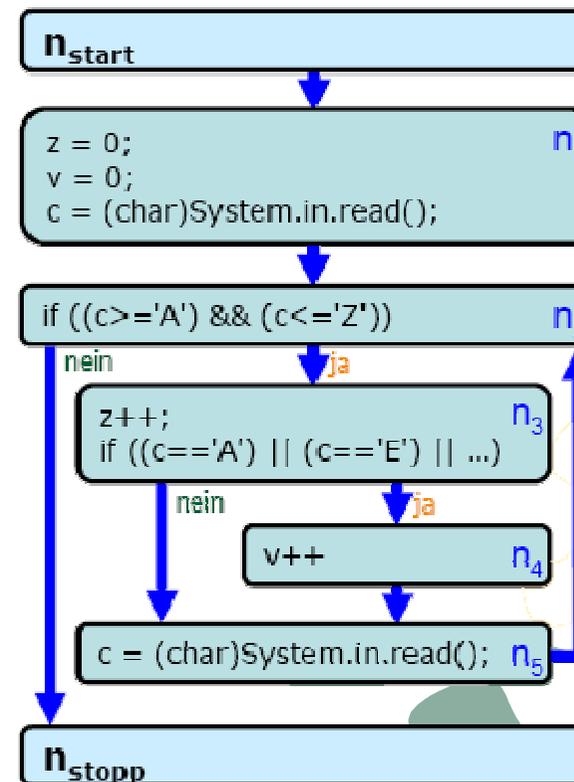
Zweigüberdeckung, alle Kanten

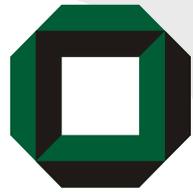
$(n_{\text{start}}, n_1, n_2, n_3, n_4, n_5, n_2, n_3, n_5, n_2, n_{\text{stopp}})$

Anweisungsüberdeckung, alle Knoten

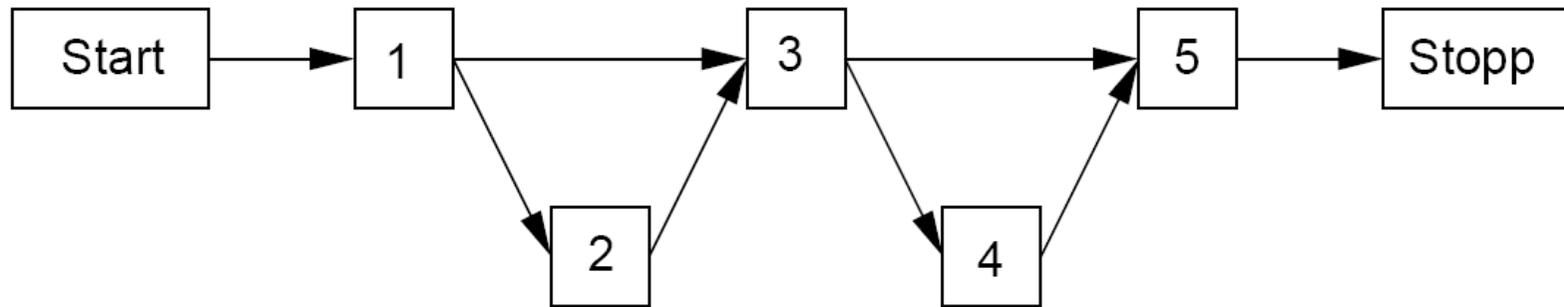
Beispielfolge: $(n_{\text{start}}, n_1, n_2, n_3, n_4, n_5, n_2, n_{\text{stopp}})$

→ Zweig (n_3, n_5) wird nicht ausgeführt





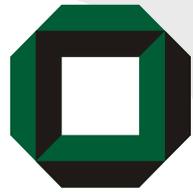
Beispiel 2



$A = \{\text{Start}; n1; n2; n3; n4; n5; \text{Stopp}\}$

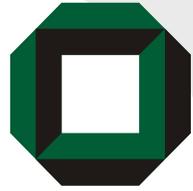
$Z = \{\text{Start}; n1; n3; n5; \text{Stopp}\},$
 $\{\text{Start}; n1; n2; n3; n4; n5; \text{Stopp}\}$

$P = \{\text{Start}; n1; n3; n5; \text{Stopp}\},$
 $\{\text{Start}; n1; n2; n3; n5; \text{Stopp}\},$
 $\{\text{Start}; n1; n3; n4; n5; \text{Stopp}\},$
 $\{\text{Start}; n1; n2; n3; n4; n5; \text{Stopp}\}$



Bedingungsüberdeckung I

- Die *einfache Bedingungsüberdeckung* fordert, dass jede atomare Bedingung einmal mit W und einmal mit F belegt wird.
- Die *mehrfache Bedingungsüberdeckung* fordert, dass die atomaren Bedingungen mit allen möglichen Kombination der Wahrheitswerte W und F belegt werden.
- Die *minimale-Mehrfache Bedingungsüberdeckung* fordert, dass jede Atomare und jede zusammengesetzte Bedingung jeweils zu W und F evaluieren muss.



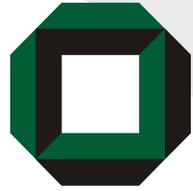
Bedingungsüberdeckung

```
if ( A && B ) {  
    S1;  
} else {  
    S2;  
}
```

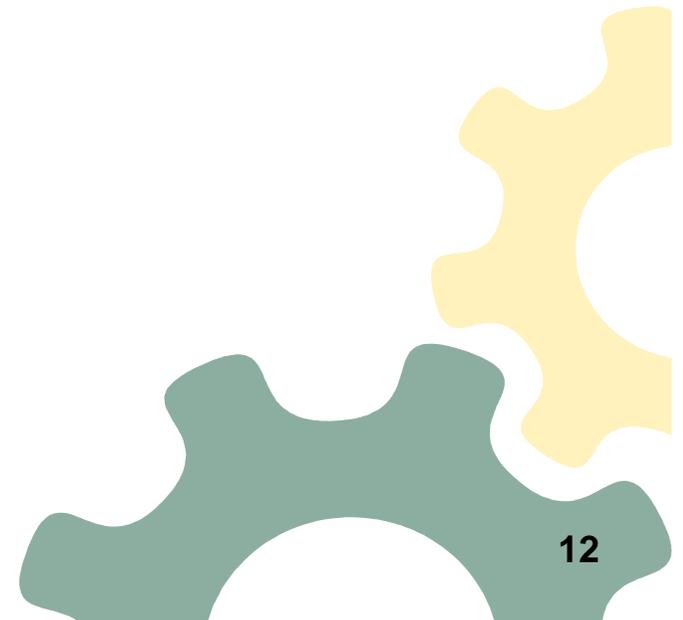
Einfache Überdeckung: {(false, false), (true, true)}

Mehrfache Überdeckung: {(false, false), (false, true), (true, false), (true, true)}

Minimal-mehrfache Überdeckung: {(false, false), (true, true)}

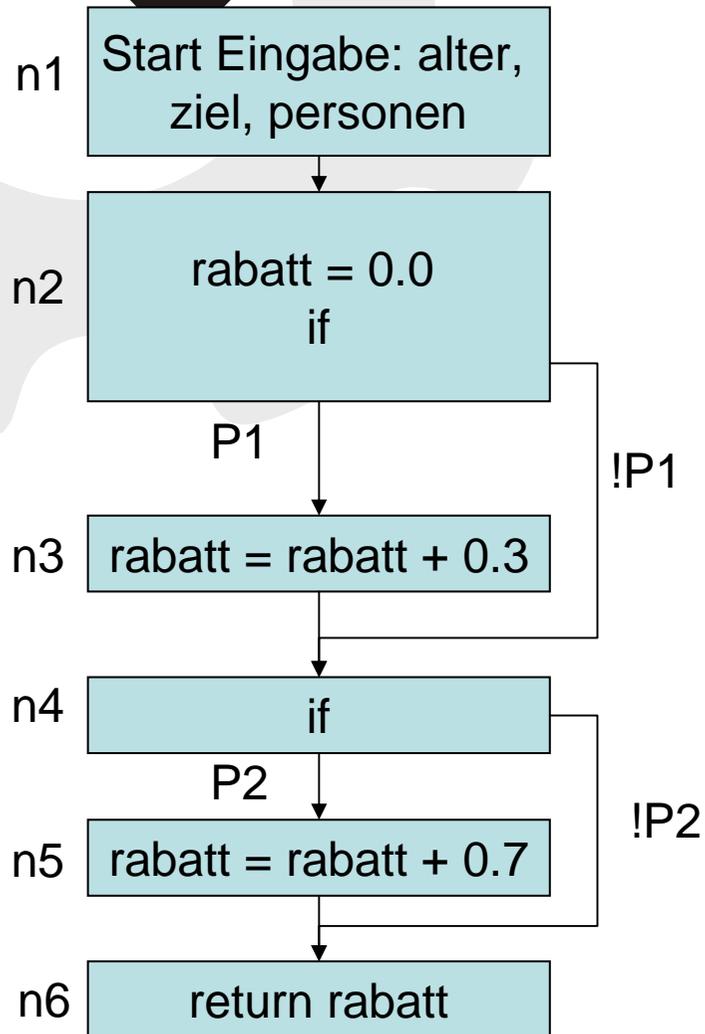


Aufgaben





Lösungen A1



Anweisungsüberdeckung:

$\{(n1, n2, n3, n4, n5, n6), (n1, n2, n4, n6)\}$

Zweigüberdeckung:

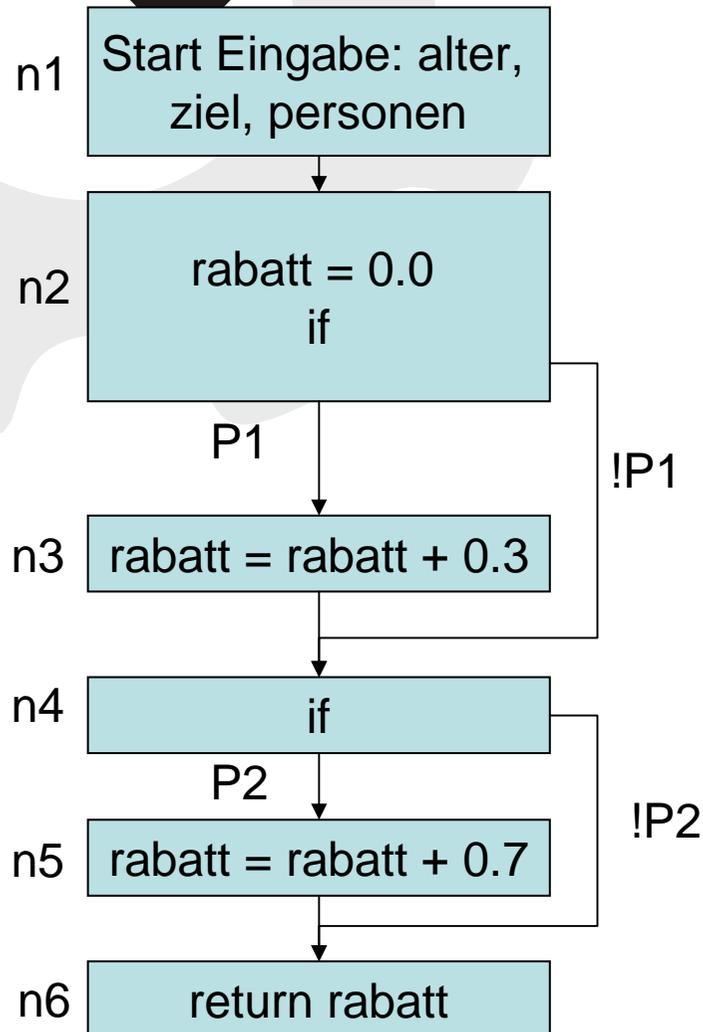
$\{(n1, n2, n3, n4, n5, n6), (n1, n2, n4, n6)\}$

Pfadüberdeckung:

$\{(n1, n2, n3, n4, n5, n6), (n1, n2, n4, n6), (n1, n2, n4, n5, n6), (n1, n2, n3, n4, n6)\}$



Lösungen A1



(2, „Adelaine“, 1):
(n1, n2, n4, n5, n6)
⇒ Keine Anweisungsüberdeckung
⇒ Keine Zweigüberdeckung

P1: keine einfache Bedingungsüberdeckung
P2: keine einfache Bedingungsüberdeckung

(3, „New York“, 5), (30, „New York“, 5):
(n1, n2, n4, n5, n6),
(n1, n2, n3, n4, n6)
⇒ Anweisungsüberdeckung
⇒ Zweigüberdeckung

P1: keine einfache Bedingungsüberdeckung
P2: keine einfache Bedingungsüberdeckung



Lösungen A1

P1:

$(\text{alter} \geq 18) \text{ and } (\text{ziel} = \text{„New York“}) \text{ and } (\text{personen} \geq 3)$

Einfache Bedingungsüberdeckung:

(30, „New York“, 2)

(10, „Karlsruhe“, 20)

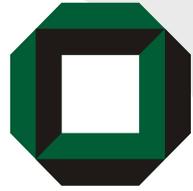
Alle atomaren Bedingungen einmal wahr und einmal falsch!

Minimal mehrfache Bedingungsüberdeckung:

(30, „New York“, 5)

(10, „Karlsruhe“, 1)

Alle atomaren und zusammengesetzten Bedingungen einmal wahr und einmal falsch (inkl. P1 selbst)!



Lösungen A2

```
10: int i = 0
```

```
20: if not (a != b) goto 100;
```

```
30: if not (a > b) goto 60;
```

```
40: a = a - b;
```

```
50: goto 70;
```

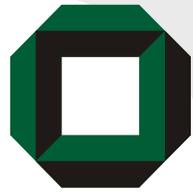
```
60: b = b - a;
```

```
70: System.out.println(i+": a="+a+" b="+b);
```

```
80: i++;
```

```
90: goto 20;
```

```
100: return a
```



Lösungen A3

- 2. Anweisungsüberdeckung
- 4. Zweigüberdeckung
- 3. minimal-mehrfache Bedingungsüberdeckung
- 1. Mehrfache Bedingungsüberdeckung

Whitebox Testen: Die Implementierung ist bekannt und wird als Grundlage für das Testen verwendet..

Blackbox Testen: Das Testen wird nur aufgrund der Spezifikation und des Verhaltens der zu untersuchenden Software vorgenommen.