

# Praktikum WP Certified Tester SoSe 14

## Aufgabenblatt 3

---

Prof. Dr. Bettina Buth <[buth@informatik.haw-hamburg.de](mailto:buth@informatik.haw-hamburg.de)>

Raum 7.86b, Tel. 040/42875-8150

### Bearbeitungshinweise:

- Die Bearbeitung der Aufgaben findet in **festen Vierergruppen** statt.
- Ein Teil der bearbeiteten Lösungen wird während der Praktikumsstunde abgenommen. Dazu werden Sie abwechselnd Ihre Lösungen vorstellen.
- Ein anderer Teil erfordert eine schriftliche Abgabe, diese geschieht über EMIL – der Abgabetermin wird im Aufgabenblatt explizit genannt.
- Es gibt **100% Anwesenheitspflicht** beim Praktikum. Beim Fehlen wegen z.B. Krankheit müssen Atteste eingereicht werden und ein Nachholtermin wird vereinbart.

### Ziel des Praktikums:

- Die Varianten der Bedingungsüberdeckung besser verstehen durch Umsetzung der Regeln im Programm
- Vorbereitung für das Praktikum 4 mit Reviews.
- Den Zweck des Mutationstests verstehen und am Beispiel von Jumble lernen, wie man Tests ergänzt um mehr Mutationen abzudecken.

### Vorbereitung vor dem Praktikum

Die Varianten der Bedingungsüberdeckung sind auf den Folien zur Vorlesung ausführlich beschrieben; bitte machen Sie sich damit ausreichend vertraut.

Für die Aufgabe zum Mutationstesten soll das CoffeMaker\_Coverage-Projekt wieder aufgesetzt werden; Zusätzlich ist das Jumble-Plugin notwendig, dass sich entsprechend dem Tutorial der ncsu einfach mittels update-site installieren lässt.

### Zusatzinformationen:

- Link zum Tutorial mit Muclipse: <http://realsearchgroup.org/SEMaterials/tutorials/muclipse/>
- Link zum Tutorial mit Jumble: <http://realsearchgroup.org/SEMaterials/tutorials/jumble/>
- Link zu Jumble: <http://sourceforge.net/projects/jumble/files/jumble/1.2.0/plugins/>
- Hilfe zur Interpretation des Jumble-outputs: <http://jumble.sourceforge.net/FAQ.html>
- 

**Abgabetermin für die schriftlichen Aufgabenteile: Do, 22.5.2014**

## White-Box Tests - Bedingungsüberdeckung

### Aufgabe 1.1: Darstellung von Wahrheitwerttabellen

Abgabeform: Abnahme im Praktikum

Entwickeln Sie in Java eine Klasse für die Darstellung von Wahrheitwerttabellen wie sie bei der Betrachtung zur Bedingungsüberdeckung zugrunde liegen – dh mit einem Array für Eingabevektoren und einen Resultatwert sowie den relevanten Zugriffsmethoden auf die Bestandteile – ev inkl Methoden zur Prüfung ob zwei Eingabevektoren Nachbarn sind sowie der Ermittlung des Ergebniswerts zu einem Eingabevektor. Eventuell ist auch eine Eingabemethode für Wertetabellen und eine Print-Methode für Wertetabellen hilfreich.

Demonstrieren Sie die Funktion Ihrer Klasse durch die Wertetabelle aus dem Vorlesungsbeispiel:

A1	0	1	0	1	0	1	0	1
A2	0	0	1	1	0	0	1	1
A3	0	0	0	0	1	1	1	1
B	1	0	0	0	1	0	1	0

### Aufgabe 1.2: Ermittlung der Bedingungsüberdeckungen

Abgabeform: Abnahme im Praktikum und schriftliche Abgabe des endgültigen Codes (nach Tests)

Ergänzen Sie Ihr Programm um eine Methoden zur Ermittlung der benötigten Testfälle (als Vektoren der Eingabeparameter mit erwartetem Ergebniswert) für die einfache Bedingungsüberdeckung inklusive der Prüfung ob eine Entscheidungsüberdeckung erreicht wird sowie für eine minimal bestimmende Mehrfachbedingungsüberdeckung.

Demonstrieren Sie die Funktion der Methode durch

- Die Tabelle aus der Folien (s.o.)
- Das Beispiel aus der Übung in der Vorlesung (Aufgabe 5, Test2 WP-CT SoSe 11)

### Aufgabe 1.3: Test der Methoden

Abgabeform: Abnahme im Praktikum und schriftliche Abgabe

Geben Sie einige Testfälle an, mit denen Sie Ihre Methoden zur Bestimmung der Bedingungsüberdeckung prüfen würden und begründen Sie mit geeigneten Mitteln, warum dies sinnvolle Testfälle sind.

### Mutationstests

Nach: Open Seminar Software Engineering der North Carolina State University – Tutorials mit Jumble und Muclipse

#### Aufgabe 2.1: Analyse der Ausgangssituation

Abgabeform: Abnahme im Praktikum

Der Source Code des CoffeeMaker\_Coverage ist vollständig, d.h. lauffähig; die mitgelieferten Tests sind in JUnit 3 geschrieben.

Gehen Sie wie im Aufgabenblatt 2 wiederum zunächst von den mitgelieferten Tests aus und generieren Sie die Information, die Jumble zu diesem Test anbietet.

Prüfen Sie durch Auskommentierung von Tests, welche der ursprünglichen Testfälle wie zum Jumble-Score beitragen.

Erläutern Sie die Ausgaben zur Abnahme im Praktikum.

#### Aufgabe 2.2: Mutationen

Abgabeform: Abnahme im Praktikum – erst nach Abnahme von Aufgabe 2.1 beginnen!

Ergänzen Sie die Tests für die CoffeeMaker Klasse so, dass alle Mutationen erledigt werden (Score = 100%) sofern dies möglich ist. Falls es nicht möglich ist, begründen Sie warum nicht.

Hinweis: während der Entwicklung und Durchführung der Tests darf der Code nicht verändert werden!

Hinweis: Jumble arbeitet erst, wenn kein Test fehlschlägt, d.h. dass die Fehler im Code durch „falsche asserts“ akzeptiert werden müssen.

## Viel Spaß!