

# 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:

- Begriffe im Umfeld des Testens durch Literaturrecherche erarbeiten und dokumentieren
- JUnit 4 und JUnit 3 kennenlernen, den Unterschied verstehen
- Unit-Tests selber schreiben zu vorgegebenen Anforderungen
- Testspezifikationen dokumentieren und Fehlerreports schreiben

### Vorbereitung vor dem Praktikum

Das Anwendungsbeispiel für diese Aufgabe stammt aus einem Softwareengineering-Kurs der North Carolina State University. Es handelt sich um einen *CoffeeMaker*, der als Eclipse-Projekt mit Code und ersten JUnit-Tests daherkommt.

Im pub liegen die zip-Archive *CoffeeMaker\_JUnit* und *CoffeeMaker\_Coverage*.

Im Rahmen der Aufgaben muss zunächst der CoffeeMaker als Eclipse-Projekt mit JUnit aufgesetzt werden. Dazu wie folgt vorgehen:

- Das zip-Archiv *CoffeeMaker-JUnit* ins lokale Verzeichnis kopieren und entpacken.
- In Eclipse entweder das Projekt CoffeeMaker\_JUnit importieren oder ein neues Projekt CoffeeMaker\_JUnit anlegen und die Quellen des CoffeeMaker dorthin importieren (funktioniert beides).
- Sicherstellen, dass JUnit im Buildpath liegt - ansonsten als external lib angeben.

Danach sollte das Projekt fehlerfrei aufgesetzt sein. Das kann dadurch geprüft werden, dass die vorhandenen JUnit-Tests ausgeführt werden; die JUnit Tests befinden sich im Verzeichnis unittests.

### Zusatzinformationen:

- Zum CoffeeMaker als Anwendung: [http://realsearchgroup.org/SEMaterials/tutorials/coffee\\_maker/](http://realsearchgroup.org/SEMaterials/tutorials/coffee_maker/)
- JUnit-Tutorial mit dem CoffeeMaker:  
[http://realsearchgroup.org/SEMaterials/tutorials/junit/junit\\_tutorial\\_3.5.html](http://realsearchgroup.org/SEMaterials/tutorials/junit/junit_tutorial_3.5.html)

**Abgabetermin für die schriftlichen Aufgabenteile: Do, 10.4.2014**

## Testbegriffe und Glossar

### Aufgabe 1: Testbegriffe recherchieren und dokumentieren

Abgabeform: schriftliche Abgabe

Erstellen Sie ein Glossar für die folgenden Begriffe, die im Zusammenhang mit Software-Engineering relevant sind.

- Software Qualität
- Unit-Test
- Stress-Test
- Smoke-Test
- Branch Coverage
- Condition Coverage

Geben Sie für jeden Begriff eine knappe Erklärung an sowie mindestens drei Quellen, in denen Sie Informationen zum Begriff gefunden haben.

Hinweis: mögliche Quellen sind Web-Seiten, Bücher, Zeitschriften

## JUnit Tests für den CoffeeMaker

Nach: Open Seminar Software Engineering der North Carolina State University

### Aufgabe 2.1: JUnit 3 und JUnit 4

Abgabeform: Abnahme im Praktikum

Der Source Code des CoffeeMaker ist vollständig, d.h. lauffähig.

Die mitgelieferten Tests sind in JUnit 3 geschrieben. Schreiben Sie die Tests in JUnit 4 um und ändern Sie die Projektkonfiguration entsprechend.

Ergebnis: alle bisherigen Tests sind nach wie vor lauffähig und liefern einen grünen Balken.

### Aufgabe 2.2: Tests ergänzen

Abgabeform: Abnahme im Praktikum, schriftliche Abgabe der Test Cases und der Problem Reports

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

Die vorhandenen Tests sind unvollständig und der Code fehlerhaft.

Die Anforderungen für den CoffeeMaker sind als User-Stories zu finden (vgl Quellen oben).

Die Tests sollen vervollständigt werden für die folgenden User-Stories:

- a) Add a Recipe
- b) Delete a Recipe
- c) Edit a Recipe
- d) Add Inventory
- e) Check Inventory
- f) Purchase Coffee

Finden Sie so viele Fehler wie möglich.

Zur Dokumentation der neuen Testfälle verwenden Sie bitte das Muster [Test Case Dokumentation Template](#);

zur Dokumentation der gefundenen Fehler das [Problem Report Template](#), beide finden sich im pub zur Vorlesung und in EMIL.

### Aufgabe 2.3: Fehler beheben

Abgabeform: Abnahme im Praktikum

Hinweis: mit der Bearbeitung sollte nicht begonnen werden, bevor Aufgabe 2.2 abgenommen wurde, gearbeitet werden soll auf einer neuen Kopie der CoffeeMaker Quellen!

Beheben Sie die gefundenen Fehler und demonstrieren Sie, dass alle Tests erfolgreich laufen.

Hinweis:

- Melden Sie sich für die Abnahme der Aufgabe 2.1 bis 2.3 sobald die jeweilige Aufgabe fertig bearbeitet ist und bevor Sie mit den folgenden Aufgaben beginnen
- Aufgabe 1 kann unabhängig von den übrigen Aufgaben bearbeitet werden.

Viel Spaß!