

Praktikum WP Certified Tester SoSe 16

Aufgabenblatt 3

Prof. Dr. Bettina Buth <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:

- Code-Coverage Analyse bei Tests kennenlernen und am Beispiel lernen, Test-Coverage zu erhöhen durch geeignete Testfälle
- Die Varianten der Bedingungsüberdeckung besser verstehen durch Umsetzung der Regeln im Programm
- Vorbereitung für das Praktikum 4 mit Reviews.

Vorbereitung vor dem Praktikum

- CT-Calculator in der Version *termin_3_v1.zip* in Eclipse importieren
- Plugin EcEmma in Eclipse einbinden (über Eclipse Marketplace)
- Die Varianten der Bedingungsüberdeckung sind auf den Folien zur Vorlesung ausführlich beschrieben; bitte machen Sie sich damit ausreichend vertraut.

Zusatzinformationen:

- Tutorials zur Coverage mit EcEmma: <http://realsearchgroup.org/SEMaterials/tutorials/eclemma/>
- EcEmma allgemein: <http://www.eclemma.org/> (dort auch Info für die Einbindung in Eclipse)
- EcEmma alternativer Installationsweg: Source von <http://sourceforge.net/projects/eclemma/> herunterladen. Inhalte (aus den Sources) des feature- und plugin-Ordners in den entsprechenden Ordner in Eclipse hineinkopieren. Mit einem Neustart von Eclipse ist EcEmma dann verfügbar.

Abgabetermin für die schriftlichen Aufgabenteile: Do, 19.5.2016

White-Box Tests – Coverage Messen

Aufgabe 1.1: Analyse der Ausgangssituation

Abgabeform: Abnahme im Praktikum

Der Source Code des CT-Calculator ist vollständig und lauffähig; eine kleine Anzahl von Basistests ist vorhanden

Analysieren Sie die bisher implementierten Testfälle bezüglich der Überdeckung des Source Codes der Klassen *processing* und *storage*. Erstellen Sie einen Coverage Report mit EcEmma und erläutern Sie diesen.

Aufgabe 1.2: Test-Coverage erhöhen

Abgabeform: Abnahme im Praktikum, schriftliche Abgabe der Coverage-Reports (vor und nach Entwicklung der neuen Tests) und des Codes (*processing/storage/tests*)

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

Ergänzen Sie die JUnit-Tests so, dass die Coverage bei EcEmma mindestens 90% beträgt für die Coverage-Maße

- instructions,
- branches,
- lines,
- methods.

Erstellen Sie bitte vor und nach der Ergänzung der Tests die Coverage-Reports mit EcEmma zum Vergleich und erläutern diese.

Aufgabe 1.3: Test-Coverage analysieren und Fehler beheben

Abgabeform: Abnahme im Praktikum, schriftliche Abgabe des endgültigen Coverage-Reports und des Codes (*processing/storage/tests*)

Beheben Sie die in Aufgabe 1.2 gefundenen Fehler und lassen sie Ihre Tests erneut laufen. Untersuchen Sie ob für die obigen Maße die Coverage auf 100% erhöht werden kann.

Hinweis:

- Melden Sie sich für die Abnahme der Aufgabe 1.1 und 1.2 sobald die jeweilige Aufgabe fertig bearbeitet ist und bevor Sie mit den folgenden Aufgaben beginnen

White-Box Tests - Bedingungsüberdeckung

Aufgabe 2.1: Darstellung von Wahrheitstabellen

Abgabeform: Abnahme im Praktikum

Entwickeln Sie in Java eine Klasse für die Darstellung von Wahrheitstabellen 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 2.2: Ermittlung der Bedingungsüberdeckungen

Abgabeform: Abnahme im Praktikum und schriftliche Abgabe des endgültigen Codes (nach Tests vgl 2.3)

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

- a) Die Tabelle aus der Folien (s.o.)
- b) Das Beispiel aus dem Test1 der Vorlesung

Aufgabe 2.3: Test der Methoden

Abgabeform: Abnahme im Praktikum und schriftliche Abgabe der Tests

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.

Viel Spaß!