

Praktikum WP Certified Tester SoSe 22

Aufgabenblatt 6 (für Praktikumstermine 31.5.-13.6.)

Prof. Dr. Bettina Buth <buth@informatik.haw-hamburg.de>

Raum 11.80a, Tel. 040/42875-8150

Torge Hinrichs <torge.hinrichs@haw-hamburg.de>

Eike Meyer <eike.meyer@haw-hamburg.de>

Bearbeitungshinweise:

- Die Bearbeitung der Aufgaben findet in **festen Vierergruppen** statt.
- Die Abgabeform der einzelnen Aufgaben ist einzeln ausgewiesen; schriftliche Abgabe in die Dropbox in der owncloud.
- Es gibt **100% Anwesenheitspflicht** beim Praktikum. Beim Fehlen wegen z.B. Krankheit müssen Atteste eingereicht werden und ein Nachholtermin wird vereinbart.

Ziel des Praktikums:

- Varianten der Bedingungsüberdeckung besser verstehen durch Umsetzung der Regeln im Programm
- Code-Coverage Analyse bei Tests kennenlernen und am Beispiel lernen, Test-Coverage zu erhöhen durch geeignete Testfälle

Vorbereitung vor dem Praktikum

- Coverage-Tool in der IDE auswählen (zB IntelliJ in der RUN Konfiguration)
- Die Varianten der Bedingungsüberdeckung sind auf den GTB Folien in Kap 4.2 ausführlich beschrieben; bitte machen Sie sich damit ausreichend vertraut.

Zusatzinformationen:

- Markdown Format: <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>
- Eine MC/DC Definition findet sich zB unter https://en.wikipedia.org/wiki/Modified_condition/decision_coverage und mit einem Beispiel unter http://www.verifysoft.com/en_example_mcdc.html und <http://www.utdallas.edu/~ewong/SE6367/03-Lecture/Special-Lecture-10-24-MC-DC-Coverage-two-slides.pdf>

Abgabetermin für der Implementierung:

bis Mo, 13.6.2022

als zip-Datei WP-CT-WB-CCov-xxx.zip im Dropfolder,

(wobei xxx die Liste Eurer Namen sein soll!)

WP Certified Tester SoSe 22 – Aufgabenblatt 6

White-Box Tests – Bedingungsüberdeckung

Bedingungsüberdeckungen sind ein spezieller Teil des Whitebox-Testens; die Analyse der Auswertungsmöglichkeiten der einzelnen Bestandteile einer Bedingung unterstützen nicht nur das Erreichen aller Code-Anteile sondern berücksichtigen in gewissem Umfang auch Abhängigkeiten von vorherigen Berechnungen der verwendeten Variablen. Eine mögliche Reduktion des Testaufwands für die Bedingungen erlauben die minimal bestimmende Mehrfachbedingungsüberdeckung (MMBÜ) und der MC/DC Ansatz, die im Rahmen dieser Praktikumsaufgabe implementiert werden sollen.

Konkret soll die Implementierung für eine gegebene Wahrheitswerttabelle, die eine Bedingung repräsentiert, die signifikanten Testfälle im Sinne der MMBÜ wie auch im Sinne von MC/DC als Menge von Eingaben und Sollwerte bestimmen und zwar unabhängig von der Anzahl der atomaren Teilbedingungen.

Aufgabe 1.1: Implementierung der Auswahl für Bedingungsüberdeckungen

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

Implementieren Sie die Auswahl der Bedingungsüberdeckung entsprechen der folgenden Usage Szenario:

Der Technical Test Analyst (User) liefert die Bedingung für die die Bedingungsüberdeckung bestimmt werden soll als Wahrheitswerttabelle in einer Datei. Für die Gesamtbedingung sollen beliebig viele atomare Teilbedingungen möglich sein. Das System zur Testfallselektion liest diese Wahrheitswerttabelle und eine oder mehrere Bedingungsüberdeckungskriterien ein und konvertiert sie in die interne Darstellung. Danach wird die Tabelle bezüglich des oder der gewählten Überdeckungskriterien analysiert und die Tabelle mit den ausgewählten Fällen wird im gleichen Format wie dem Eingabeformat an den Nutzer in einer Datei zurückgegeben.

Für die Eingabe der Wahrheitswerttabelle sollen verschiedene Formate möglich sein, für die Pilotimplementierung soll Markdown als Eingabeformat realisiert werden. Optional ist CSV als mögliches weiteres Format.

Als mögliche Überdeckungskriterien **sollen mindestens MMBÜ und MC/DC implementiert** werden, optional ist die Einfache Bedingungsüberdeckung.

Daraus ergeben sich folgende User Stories, die mindestens implementiert werden müssen:

ID	Story
IO-01	Als User möchte ich aus Dateien Wahrheitswerttabellen in verschiedenen Formaten einlesen können. (Markdown, CSV optional)
IO-02	Die Anwendung muss mir als User die Auswahl der Testfälle als Wahrheitswerttabelle im gleichen Format wie das Input-Format liefern können. (File Reader / Writer)
IO-03	Als User möchte ich mehrere beliebige Dateien automatisch hintereinander verarbeiten lassen können. (Bulkprocessing)
IO-04	Als User möchte ich darauf hingewiesen werden, dass eine Input Datei fehlerhaft ist.
PR-01	Als User möchte ich zwischen verschiedenen Coverage-Maßen wählen können. (mindestens MMBÜ und MCDC, EBÜ optional)
PR-02	Als User möchte ich mehrere Coverage-Maße in einem Durchlauf ausführen. (Mehrfachauswahl)
PR-03	Als User soll die interne Struktur der Anwendung nicht sichtbar sein. (sauberes Interface)

Als nicht-funktionale Anforderungen sollen bei der Implementierung berücksichtigt werden:

WP Certified Tester SoSe 22 – Aufgabenblatt 6

NF1	Die Implementierung soll Wahrheitswerttabellen mit beliebig vielen atomaren Bedingungen verarbeiten können (Scalability)
NF2	Die Implementierung soll einfach erweiterbar sein für andere Eingabeformate (Modifiability)
NF3	Die Implementierung soll einfach erweiterbar sein für weitere Bedingungsüberdeckungskriterien (Modifiability)
NF4	Der Code soll gut wartbar sein (Maintainability)

Aufgabe 1.2: Demonstration der Funktion

Abgabeform: Abnahme im Praktikum und schriftliche Abgabe des endgültigen Codes (inkl Tests vgl Aufgabe 2)

Demonstrieren Sie die Funktion des Auswahlsystems zunächst für die folgenden Fälle:

- a) Die Tabelle aus der Folien (s.o.) für Minimal Bestimmte Mehrfachbedingungsüberdeckung

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

- b) Erstellen Sie für das folgende Beispiel zunächst von Hand die ausgewählten Testfälle für MMBÜ und MC/DC und prüfen Sie ob diese mit ihrer Implementierung übereinstimmt.

B1	B2	B3	Cond
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

- c) die Beispiele exx.md, die schon im Markdown-Format vorliegt (vgl Ordner zu Aufgabe 5)

WP Certified Tester SoSe 22 – Aufgabenblatt 6

White-Box Tests – Coverage Messen

Hinweis:

- Dokumentieren Sie den Stand der Test-Coverage jeweils nach Aufgabe 2.2, 2.3 und 2.4. durch Ablage der Coverage Reports mit dem gewählten Coverage Werkzeug.

Aufgabe 2.1: Test der Methoden

Abgabeform: Abnahme im Praktikum

Implementieren Sie einige geeignete Testfälle für Ihren Code aus Aufgabe 1.

Aufgabe 2.2: Analyse der Ausgangssituation

Abgabeform: Abnahme im Praktikum, schriftliche Abgabe des Coverage-Reports und initialen Test

Erstellen Sie für Ihren Code aus Aufgabe 2.1 einen Coverage Report und erläutern Sie diesen.

Aufgabe 2.3: Test-Coverage erhöhen

Abgabeform: Abnahme im Praktikum, schriftliche Abgabe des Coverage-Reports und Test-Dokumentation

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

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

- instructions,
- branches,
- lines,
- methods

Aufgabe 2.4: Test-Coverage analysieren und Fehler beheben

Abgabeform: Abnahme im Praktikum, schriftliche Abgabe der Begründung

Beheben Sie die in Aufgabe 2.3 gefundenen Fehler und lassen Sie Ihre Tests erneut laufen. Untersuchen Sie ob für die obigen Maße die Coverage auf 100% erhöht werden kann. Begründen Sie Ihre Aussage schriftlich.

Viel Spaß!