Praktikum Software-Engineering 1 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.
- Die bearbeiteten Lösungen werden in der Regel während der Praktikumsstunde abgenommen. Dazu werden Sie abwechselnd Ihre Lösungen im Team vorstellen.
- Es gibt **100% Anwesenheitspflicht** beim Praktikum. Beim Fehlen wegen z.B. Krankheit müssen Atteste eingereicht werden und ein Nachholtermin wird vereinbart.

Ziel des Praktikums:

- Software Design – Modelle und Implementierung
  - Implementierung von Modellen mit dem State Pattern
- Testen – systematischer Ansatz
  - Blackbox Tests: Ableiten von Testfällen auf Basis von Zustandsmodellen
  - Whitebox Tests: Coverage mit EclEmma

Vorbereitung vor dem Praktikum

1) Machen Sie sich mit den State Pattern vertraut, speziell auch mit seiner Implementierung in Java
2) Machen Sie sich mit dem Werkzeug EclEmma zur CodeCoverage vertraut; ev. müssen Sie es über den Eclipse Marketplace einbinden.
3) Machen Sie sich mit dem Testfallerzeugung auf Basis von Zustandsautomaten vertraut – Stichworte Konformanztest und Robustheitstest
4) Lesen Sie die Aufgaben sorgfältig durch und stellen Sie Ihre Fragen zu den Aufgaben am Anfang des Praktikumstermins

Zusatzinformationen:

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

Aufgabe 1: Implementierung nach Pattern

Gehen Sie aus von der bisherigen Implementierung des Systems aus Praktikum 2, Aufgabe 2 aus.

1.1: State-Modell und Implementierung

Abgabe: Diskussion im Praktikum, schriftliche Abgabe von Modell und Code (vgl Aufgabenblatt 2)

Übergang von Praktikumstermin 2:

- Vorstellung der Zustandsmodelle
  - Abklärung der Grenzen zwischen Steuersoftware und Umgebung
  - Events oder Bedingungen als Trigger
  - Implementierung paralleler Zustände
  -

1.2: Alternative Implementierung nach State-Pattern

Abgabe: Diskussion im Praktikum, schriftliche Abgabe des Codes

Entwickeln Sie eine alternative Implementierung Ihres Systems auf Basis des State-Pattern.

a) Klassenstruktur neu

Geben Sie zunächst das Design der neuen Implementierung als Klassendiagramm an.

b) Implementierung

Implementieren Sie Ihr Design auf Basis der ermittelten Klassenstruktur.

Diskussion:

- Welche Klassen des bisherigen Systems bleiben erhalten, welche ändern sich? Welche neuen Klassen kommen hinzu?
- Wie wirken sich die parallelen Zustände auf die Struktur aus?
- Müssen die bisherigen Tests angepasst werden? Falls ja: was muss verändert werden?

Aufgabe 2: Testen am Beispiel

2.1: Testen – Coverage messen

Abgabe: Vorstellung im Praktikum, Diskussion

Analysieren Sie die bisher implementierten Testfälle bezüglich der Überdeckung des Source Codes der Klassen in den packages *fsm, implementation, boundaryclasses* mit EclEmma.

Diskussion:

- Welche Information liefert EclEmma?
- Wie hoch ist die Coverage durch Ihre bisherigen Tests?
- Was ist der Unterschied zwischen Instruction, Branch, Line, Method Coverage

2.2: Testen – Whitebox Tests: Coverage erhöhen

Abgabe: Vorstellung im Praktikum, Diskussion

Erstellen Sie weitere Tests, die die Coverage für die obigen Klassen auf mindestens 90% erhöhen für die Coverage Maße

- Instructions
- Branches

Diskussion:

- Welche Bereiche des Codes sind leicht, welche schwer zu erreichen?

2.3: Testen –Blackbox Tests: Zustandsbasierte Testfälle

Abgabe: Vorstellung im Praktikum, Diskussion

Bestimmen Sie auf Basis Ihres Automatenmodells aus Aufgabe 2 im Praktikum 2 und mit Hilfe des Zustandsübergangsbaums Konformanz- und Robustheitstests für das Gesamtsystem.

Diskussion:

- An welchen Stellen gibt es Probleme mit der Generierung des Zustandsübergangbaums?
- Welche Überdeckung für den Automaten erreicht man mit den Konformanztests und den Robustheitstests?

Optional: Implementierung der Tests.

Hinweis: Abgabe der Aufgabe schriftlich, per email an buth@informatik.haw-hamburg.de mit Betreff „[SEP1] Aufgabenblatt 3, Aufgabe 1“ – Kopie an alle Teammitglieder

---

Abgabe der schriftlichen Teile bis

Do, 11.6.2016, 23:00

Viel Spaß!