



Software Engineering Übung 6

Softwaretests

1 Informationen

1.1 Daten

- Ausgabe Di. 25.11.2008
- Abgabe So. 07.12.2008 bis 23:59 Uhr
- Besprechung am Di 16.12.2008

1.2 Formales

Die Lösungen sollen als PDF Datei mit dem Namen **Ex[n].[NameA_NameB_NameC].pdf** abgegeben werden, wobei [n] die Nummer der Übung ist und [NameA_NameB_NameC] die Nachnamen der Gruppenmitglieder sind. Die PDF Datei sollte ausserdem ebenfalls Ihre Namen und Matrikelnummern beinhalten.

Mailen sie Ihre Lösungen vor dem Abgabetermin an jeanneret@ifi.uzh.ch. Der Betreff der E-mail sollte mit **[SE EX HS08]** beginnen. Falls Sie zusätzliche Abgabematerialien (z.B. Source Code) haben, mailen Sie bitte ein Archiv (.zip-File), welches alle Dateien, einschliesslich dem PDF, enthält. Benennen sie das Archiv anhand der oben erwähnten Konventionen.

Die Übungen sollen in 3er Gruppen gelöst werden. Jedes Gruppenmitglied muss über alle Teile der Lösungen Auskunft geben können. Verspätete Abgaben werden korrigiert, aber nicht bewertet.

2 Aufgabenstellung

Die gestellten Teilaufgaben beziehen sich auf das Projekt, welches Sie bereits in den Übungen 2 bis 5 bearbeiteten, und auf JClusim.

2.1 Funktionsorientiertes Testen (13 Punkte)

In einer erweiterten Version von JClusim können die Agenten benannt werden. Die Namen werden in der Simulation über den jeweiligen Agenten angezeigt. Für die Namen der Agenten kann man entweder simple Zeichenketten eingeben (z.B. "AgentTyp1"), oder Uniform Resource Identifiers (URIs) benutzen (z.B. <http://www.meineAgenten.ch/AgentTyp2>). Im letzteren Fall

möchte man nicht den ganzen String während der Simulation anzeigen, da lange URLs/URIs auf dem Bildschirm zuviel Platz einnehmen. Deshalb wurde die folgende Funktion erstellt. Die Methode bekommt als Parameter den vollständigen Namen eines Agenten. Falls der Name Schrägstriche (Slashes) beinhaltet, wird nur der Teil des Namens hinter dem letzten Schrägstrich zurückgegeben (zum Beispiel "AgentTyp2"). Da Namen auch mit einem Schrägstrich enden können, werden Schrägstriche am Schluss des Strings zuerst noch abgeschnitten.

Erklärungen zu den verwendeten Funktionen:

- Die Funktionen `lastIndexOf()` und `substring()` verwenden Strings wie Arrays welche mit einzelnen Zeichen gefüllt sind. Das erste Zeichen des Strings hat den Index 0, das letzte Zeichen des Strings hat den Index `String.length() - 1`.
- `int lastIndexOf(char a)` liefert den Index des letzten Vorkommens des eingegebenen Zeichens a, oder -1 falls das Zeichen im String nicht vorkommt.
- `String substring(int a)` liefert den Schluss eines Strings (angefangen mit dem Zeichen bei Index a (z.B. `'Hallo'`.`substring(1)` ergibt "allo").
- `String substring(int a, int b)` liefert den Teilstring zwischen den Indices a und b (inklusive dem Zeichen bei Index a und ohne das Zeichen bei Index b, z.B. `'Hallo'`.`substring(1, 3)` ergibt "al").

```
/**
 *Die Methode liefert eine gekuerzte Version des eingegebenen Namens.
 *Falls der Name einen oder mehrere Schraegstriche enthaelt, wird nur
 *die Zeichenkette nach dem letzten Schraegstrich zurueckgegeben.
 *Schraegstriche am Ende des Strings werden abgeschnitten.
 *
 *@param name vollstaendiger Name
 *@return shortName die gekuerzte Version des Namens
 */
String getShortName( String name ) {
    int slashPos;
    String shortName = name;

    do {
        slashPos = shortName.lastIndexOf("/");
        shortName = shortName.substring( 0, shortName.length() - 1 );
    }
    while( slashPos == shortName.length() - 1 );

    slashPos = shortName.lastIndexOf("/");
    if( slashPos >= 0 ) {
        shortName = shortName.substring( slashPos + 1 );
    }

    return shortName;
}
```

- a) Entwerfen Sie einen Black-Box-Test für diese Funktion und stellen Sie die von Ihnen ausgewählten Testfälle tabellarisch dar.
- Welche Äquivalenzklassen verwenden Sie?
 - Welche Grenzfälle entwerfen Sie?

- iii. Schreiben Sie eine Testvorschrift.
- b) Testen Sie diese Implementierung mit der Testvorschrift aus Aufgabe iii. und dokumentieren Sie das Ergebnis (Schreibtischtest ohne Ausführung auf dem Rechner).
- c) Berechnen Sie die Zweigüberdeckung Ihres Tests.
- d) Zeigen Sie, dass es Testfälle gibt, die 100% Zweigüberdeckung erreichen, ohne dass ein Fehler in `getShortName` auftritt.
- e) Begründen Sie abschliessend generell anhand eines Beispiels, welche Fehlerarten beim White-Box-Test leicht und welche im Gegensatz zum Black-Box-Test schwer erkannt werden können.

2.2 Zielorientiertes Messen (7 Punkte)

Als Fallstudie für diese Aufgabe dient das BlascoArt Projekt. Die in Übung 4 vorgeschlagenen Erweiterungen wurden implementiert und die neue Version der Software wird an der Kunstakademie BlascoArt bereits verwendet. Dieses Semester wurden nach dem Einschreibeschluss alle Benutzer der Software gebeten, an einer kurze Umfrage über die Benutzerfreundlichkeit des Systems teilzunehmen. Offenbar gibt es Studierende, welche mit dem vorhandenen System nicht zufrieden sind. Vielen ist das Einschreibeverfahren zu kompliziert, und das Ergebnis der Umfrage ist nach der Meinung der Akademieleitung zu negativ ausgefallen. Ihr Vorgesetzter beauftragt Sie daher, einen Vorschlag für die Messung des Qualitätsziels "Einfaches und schnelles Einschreiben zu Workshops sowie ändern der Einschreibung" zu erarbeiten. Verwenden Sie zur Erfüllung dieses Auftrags den GQM-Ansatz aus Kapitel 10, Folie 46 der Vorlesung.

- a) Bestimmen Sie Faktoren für dieses Ziel und formulieren Sie Fragen, mit denen für jeden Faktor die Erreichung der geforderten Qualität geprüft werden kann. (3 Punkte)
- b) Leiten Sie aus den Fragen messbare Merkmale ab. Gehen Sie dabei auch auf die jeweils verwendeten Skalentypen ein. (4 Punkte)