
Protokoll

Software Testing mit JUnit

**SEW
3AHIT 2015/16**

Markus Reichl

Version 0.1

Inhaltsverzeichnis

1Einführung.....	3
1.1Ziele.....	3
1.2Voraussetzungen.....	3
1.3Aufgabenstellung.....	3
2Ergebnisse.....	4

1 Einführung

Software Tests dienen zur Fehlerfindung innerhalb von Software Projekten. Es gibt verschiedene Arten von SW-Tests, wir haben besprochen:

- Blackbox
- White
- Graybox

Zusätzlich haben wir noch einen Schreibtischtest angesprochen welcher jedoch mehr als Konzept dient.

1.1 Ziele

Ziel unserer Aufgabe war es sich die Grundlagen von Unit-Testing mittels Junit anzueignen.

1.2 Voraussetzungen

Unit-Tests werden genutzt um einzelne Teile einer Software zu testen. Ziel ist es Erwartungen zu überprüfen und dadurch Fehler oder Unklarheiten zu finden.

1.3 Aufgabenstellung

a) Implementieren Sie den angegebenen Testfall mit JUnit, so dass kein weiterer manueller Vergleich mehr notwendig ist. Nennen Sie die Testklasse (Klasse die den zu implementierenden Testfall enthält) StringTokenizerTest, und legen Sie diese Klasse im Package example1 ab.

b) Implementieren Sie einen weiteren Testfall, der das Verhalten der Methode nextToken() zeigt, wenn keine weiteren Token vorhanden sind.

Hinweis: Das erwartete Verhalten ist – wie in der Javadoc-Dokumentation angegeben

– eine NoSuchElementException. D.h. der Testfall geht gut, wenn genau diese Exception geworfen wird.

OPTIONAL

c) Implementieren Sie drei weitere Testfälle für die Klasse StringTokenizer, die zusätzliche (Sonder-)Fälle (z.B. ein String der ausschließlich aus Trennzeichen besteht) und andere Methoden (z.B. countTokens()) als Ziel haben.

Wie Sie sehen sind weit mehr als die insgesamt fünf Testfälle notwendig, um sämtliche Kombinationen von Methoden und Sonderfällen vollständig zu testen. Treffen Sie daher eine möglichst sinnvolle Auswahl an Testfällen die Sie implementieren.

2 Ergebnisse

Das hierzu genutzt Eclipse Project findet sich nun in einem Github Repository unter: https://github.com/mreichl-tgm/SEW_15-16.git