

Testkonzept

Während der Implementierung des Softwareproduktes sollen gleichzeitig Tests durchgeführt werden, die die korrekte Funktion des Programms testen. Diese Tests gliedern sich in Komponententests und Systemtests.

1. Komponententests

Für die Komponententests verwenden wir das JUnit-Framework, welches Konzepte und Werkzeuge zur Verfügung stellt, mit denen diese Tests weitgehend automatisiert stattfinden können.

Bevor man mit dem Programmieren beginnt, sollte von den Programmierern der aktuellen Story festgelegt werden, welche Klassen/Funktionen verwendet werden, und was diese als Ergebnis liefern sollen. Mit dieser Grundlage lassen sich nun die verschiedenen Tests anlegen.

Sind die Tests erstellt können die Programmierer Schritt für Schritt ihre Klassen entwickeln und über die zugehörigen Testklassen deren Korrektheit prüfen.

Nach dem Fertigstellen einer Story, sollten die Programmierer die Testklassen in einer Testsuite zusammenfassen. Dies erleichtert das Testen, wenn der Quellcode später noch mal modifiziert wird. So müssen nicht alle Testklassen einzeln überprüft werden, sondern ein größerer Teil des Programms wird auf Korrektheit überprüft.

Diese Überprüfung muss jedes Mal nach einer Änderung des Quellcodes durchgeführt werden.

Was testen wir mit unseren Testklassen?

Das JUnit-Framework erlaubt es, Werte und Bedingungen zu testen. Diese müssen erfüllt sein, damit der Test erfolgreich ist. Dazu stehen uns durch die Klasse *Assert* verschiedene Methoden zur Verfügung, die uns den Quellcode auf bestimmte Bedingungen hin untersuchen lassen:

- `assertTrue(boolean condition)`: testet, ob eine logische Bedingung wahr ist;
- `assertFalse(boolean condition)`: testet, ob eine logische Bedingung falsch ist;
- `assertEquals(Object expected, Object actual)`: testet, ob zwei Objekte gleich sind;
- `assertNull(Object object)`: testet, ob eine Referenz null ist;
- `assertNotNull(Object object)`: testet, ob eine Referenz nicht Null ist;
- `assertSame(Object object)`: testet, ob zwei Referenzen auf das selbe Objekt verweisen;
- `assertNotSame(Object object)`: testet, ob zwei Referenzen unterschiedlich sind;

Bsp.: `assertTrue(MartinVoit.istStudent())` liefert als Ergebnis wahr, besteht also den Test.

Sollte ein Test scheitern, kann man eine Fehlermeldung ausgeben lassen:

`assertTrue(„Bedingung ist falsch“, ThomasRiechert.istStudent())`: liefert als Ergebnis „Bedingung ist falsch“ und bricht ab.

2. Systemtests

Die Systemtests betreffen die Funktionalität des erstellten Programms. Sie müssen quasi „von Hand“ durchgeführt werden, da das JUnit-Framework hierfür nicht geeignet ist.

Gruppe: ueb19

Verantwortlicher: Martin Voit

Testkonzept

Es muss jede Funktion getestet werden. So dürfen sich zum Beispiel nur angemeldete User über den Login anmelden. Sollten diese ihr Passwort falsch eingeben, muss ihnen der Login verweigert werden.

Ist ein User angemeldet muss sichergestellt sein, dass er das Programm nur entsprechend seinen Rechten nutzen kann. So darf ein Student zum Beispiel keine SHK einrichten, oder ein Dozent keine Passwörter von anderen Usern ändern.

Ebenfalls muss sichergestellt werden, dass die gespeicherten Passwörter nicht von jedem User und erst recht nicht von Außenstehenden eingesehen werden können.

Am besten lassen sich die Funktionen des Programms dadurch testen, dass man entsprechende Arbeitssitzungen der Usergruppen (Administrator, Dozent, Übungsleiter, SHK und Student) simuliert, und alle Funktionen einmal „durchspielt“.

Sollten an einer Stelle falsche Angaben (z.B.: Eingabe eines Buchstaben bei erwarteter Ziffer) gemacht werden, so muss dies mit einem entsprechenden Hinweis abgefangen werden.

Ebenfalls überprüft werden muss der HTML-Quelltext der Webseiten des Programms. Hierbei muss überprüft werden, ob die Seiten auf allen gängigen Browsern korrekt dargestellt werden, und ob alle Funktionen (Upload bzw. Download der Übungsaufgaben etc.) des Programms nutzbar sind.

Die Ergebnisse der Komponenten- und Systemtests sind vom Verantwortlichen für Tests festzuhalten und in einer entsprechenden Textdatei zu protokollieren.