Testing Framework für JAVA

Version: 1.10.19

Agenda:

- Theorie
  - Mockito
  - Mock-Objekte
  - Funktionalität
  - Vorteile
  - Besonderheiten
- Praxisbeispiel

Was ist Mockito?

- Erweiterung/Verbesserung von EasyMock
- Programmbibliothek zum Erstellen von Mock-Objekten
- dynamische Erzeugung von Mock-Objekten
- wird vor Unit-Test geschaltet

Was sind Mock-Objekte?

- Platzhalter für echte Objekte innerhalb von Modultests
- Implementieren Schnittstellen für Zugriff auf Umgebung des zu testenden Objekts
- liefern für den Testfall passende Werte
- Werden verwendet, um bestimmtes Verhalten nachzustellen

Wann werden Mock-Objekte verwendet?

Wenn das „echte“ Objekt...

- nicht beschädigt werden soll
- nicht-deterministische Ergebnisse liefert
- Verhalten zeigen soll, das schwer auszulösen ist
- langsam ist
- noch nicht existiert
- Schwer rückgängig zu machende Prozesse anstößt

Wie funktioniert Mockito?

- Ersetzt Schnittstellen von Objekt und Umgebung durch Mock-Objekte
- Generiert Objekte und Verhalten
- Generierung zur Laufzeit
- Enge Kopplung von Unit-Tests an den zu testenden Code reduziert

Vorteile von Mockito

- keine Klassen von Hand schreiben
- keine Synchonhaltung des Quellcodes der Mock-Klassen mit denen der echten Klassen nötig
- dynamische Mock-Objekte sind sicherer gegenüber Refactoring
- Verhalten des Systems kann verifiziert werden, ohne vorher Annahmen zu treffen

Besonderheiten – Mockito

- Stub: when(....).thenReturn(....)
  - Spezifiziert eine Bedingung
  - Gibt Rückgabewert für diese Bedingung an
- Mock: verify-Methode
  - Überprüft, dass die erwarteten Aufrufe stattfinden
- Spy: spy-Methode
  - Benutzt ein echtes Objekt
  - Überprüft die erwarteten Aufrufe
  - Merkt sich alle Aufrufe

Vielen Dank für eure Aufmerksamkeit!

Quellen – Mockito

- http://mockito.org/
- http://de.wikipedia.org/wiki/Mockito
- http://de.wikipedia.org/wiki/Mock-Objekt
- http://de.wikipedia.org/wiki/Easymock