



CODING AKADEMIE
MÜNCHEN

Java Unit Testing Kurs

JUnit 5, Mockito & PowerMock

Sie wollen das wichtige Java Unit Testing Framework erlernen und unmittelbar in Ihrem Projekt einsetzen? Dann brauchen Sie nicht weiter suchen! Das Java Unit Testing Praxis Seminar ist ideal für jeden Java Programmierer, der JUnit 5 und Mockito schnell und auf praxisorientierte Weise erlernen und seinen Java Code damit testen will. Erfahren Sie hier mehr..

Die Unit 5 mit Mockito Schulung der Coding Akademie München ist auf 2 Tage angesetzt und wie gemacht für alle Java Programmierer, die ihren eigenen oder Java Code von Dritten mit JUnit 5 und Mockito testen wollen. Mit Unit Tests können Sie überprüfen, ob Ihr Code genau das tut, was man von ihm erwartet. Unit Tests erhöhen erfahrungsgemäß die Qualität des Programmiercodes. Denn zum Schreiben von Tests muss man sich mit dem Production Code genau beschäftigen und nicht selten entdeckt man dabei Fehler (Bugs) und auch bessere Lösungsansätze. In diesem Unit Testing Kurs geht es darum, die Grundideen von JUnit 5 und Mockito sowie grundlegende Features mittels vielfältiger Beispiele und Programmieraufgaben kennen zu lernen, so dass Sie dieses Testing Framework für Ihre Projekte unmittelbar einsetzen können. Nach dem Kurs können Sie Unit Tests und Mockito Tests für Ihren Production Code schreiben.

Mit JUnit 5 können Sie Ihren Production Code schnell und einfach testen. Das Prinzip ist relativ einfach, es geht um die Frage: was erwarte ich von meiner Methode und was liefert sie genau, also um einen Soll/Ist Vergleich der Werte. JUnit 5 ist der Nachfolge von JUnit 4. Es ist ein Testing Framework, das Ihnen in Java auch dabei hilft neue Sprachkonzepte wie Lambda Expressions leicht zu testen – denn ab JUnit 5 dürfen Test Methoden sogar Parameter haben. Mit Mockito können Sie APIs Code, Datenbank (CRUD Operationen) sowie Client/Server Anwendungen testen, indem Sie das Verhalten von einem solchen Code lokal imitieren und simulieren.

Falls Sie sich mit JUnit 5 und Mockito gut auskennen, dann wäre der Kurs "Java Effektives Unit Testing" eher der richtige Kurs für Sie. Bei der Effektiven Unit Testing Schulung geht es darum, wie Sie gute und smarte Unit Tests schreiben, wie Sie Ihre Test Suite strukturieren und schlechte Tests (Test Smells) erkennen und in bessere Unit Tests umwandeln können. Dort lernen Sie auch wie Sie von Anfang an Ihren Code so strukturieren, dass er leicht zu testen ist (Testable Design). Dennoch setzt der Kurs "Java Effektives Unit Test" voraus, dass Sie die Inhalte aus diesem Kurs bereits beherrschen und in der Praxis umsetzen können.

Was lernen Sie am ersten Tag? (JUnit 5 und Test Double)

Am ersten Tag steigen wir langsam aber sicher ein in die Testing Welt. Hier lernen Sie, warum das Testen so wichtig und was die Grundidee hinter Test-Frameworks wie dem JUnit 5 Framework ist. Dabei gehen wir folgenden Fragen nach: Welche Struktur sollte mein Test haben (Test Lebenszyklus), wie kann ich vernünftige Testdaten auswählen (Äquivalenzklassen und Randtest), und wie sollen meine Test Methoden heißen? Anhand von vielen Beispielen und kleinen

Programmieraufgaben werden Sie alle wichtigen Annotationen und Assertions von JUnit 5 kennen und anzuwenden lernen. Mit diesen Annotationen und Assertions können Sie bereits jetzt Ihre ersten Unit Tests schreiben. Mit JUnit 5 ist es möglich, Testdaten von CSV einzulesen oder sogar eine Methode zu schreiben, die Ihnen Testdaten generiert. Solche fortgeschrittenen Techniken nennt man "Parameterized Tests", auch diese lernen Sie wieder mittels vieler anschaulicher Beispiele und Programmieraufgaben kennen.

Ebenfalls schauen wir uns Hamcrest Matcher an, mit denen wir unsere Tests lesbarer machen und mit anspruchsvollen Bedingungen (Matchers) versehen können.

Die Grundlagen von JUnit Test sind einfach zu erlernen. Was aber, wenn Sie ein großes Projekt mit Datenbank-Operationen (CRUD-Operationen), Zugriff auf Dateien über das Netzwerk (Client/Server-Anwendungen) und noch dazu fremde APIs zu bewältigen haben? Soll dies alles auch getestet werden? Die Antwort lautet nein, denn für das Schreiben von Unit Tests ist das Betreiben von Datenbank-Programmierung oder Client/Server-Programmierung oder das Anbinden fremder APIs gar nicht nötig. Stattdessen widmen wir uns den sogenannten Test Double Techniken. Test Double ist ein Begriff, ein Schirm, unter dem sich fünf verschiedene Techniken verbergen: Dummy, Fake, Stub, Spy und Mock. Die Idee hinter Test Doubles ist sehr einfach und gleichzeitig sehr mächtig und faszinierend. Mit Test Doubles können Sie das Verhalten von Production Code, der Ihnen nicht zur Verfügung steht – wie beispielsweise API-Methoden oder Methoden, die wegen des Aufbaus von Datenbank- und Netzwerkverbindungen langsam sind und die ggf. auch von anderen Faktoren abhängen – lokal imitieren und simulieren. Dadurch können Sie aussagekräftige Unit Tests gegen die Spezifikation und gegen das zu erwartende Verhalten von Production Code schreiben. Sinn von Test Doubles ist es, sich unabhängig von den genannten Faktoren zu machen und Unit Tests zu schreiben, die schnell ausführbar sind. Denn ein Test, der mehrere Stunden oder sogar mehrere Minuten dauert ist beim Entwicklungsprozess wenig hilfreich. Stattdessen sollten Tests in wenigen Sekunden ausführbar sein, um die Entwickler nicht bei jeder neuen kleinen Änderung zu bremsen. Dauern sie zu lange, besteht die Gefahr, dass sie nie mehr ausgeführt und letztendlich ignoriert werden. Natürlich ersetzen Test Doubles keinen umfassenden Test mit dem späteren Production Code, aber dennoch können Sie damit auch gute und aussagekräftige Tests schreiben.

Was lernen Sie am zweiten Tag (Mockito und PowerMock)?

Das Schreiben von Test Doubles erfordert viel Übung und kann schnell zu einer Herausforderung werden. Daher existieren Frameworks, welche das Einsetzen von Test Doubles einfacher machen. Bei unserem Unit Testing verwenden wir das Mockito Framework. Sie lernen alle wichtigen Grundlagen von Mockito kennen und wie Sie es effektiv einsetzen können. Wir ersetzen Test Double Code durch Mockito Code und schauen uns dabei genau an, was Mockito uns an Arbeit abnimmt und auf welche Weise.

In dieser Lerneinheit starten wir mit einem einfachen Bestellsystem mit Spring 5. Keine Sorge, Spring 5 wird dabei nicht vorausgesetzt, wir wollen nur Mockito realitätsnah einführen. Dann schauen wir uns an wie wir ein solches System testen: Wir beginnen mit der Setup Phase und der Mockito Dependency. Anschließend setzen wir die sogenannten "Expectations", welche definieren, was wir von unseren Methoden genau erwarten. Dann rufen wir unsere Methoden auf und verifizieren die Ergebnisse (Verify the result). Danach sind Sie an der Reihe und wiederholen diese 3 Schritte an mehreren Methoden. Wir schauen uns dabei auch sogenannte Test Coverage Tools an, um zu sehen, ob wir noch ungetesteten Code haben. Ungetesteter Code ist z.B. ein else-Zweig in einer Bedingung, falls uns keine Testdaten zur Verfügung stehen, welche die Bedingung "false" machen. In diesem Fall würde der else-Zweig nie ausgeführt. Solche Fälle werden von den Test Coverage Tools sofort entdeckt.

Zum Schluss schauen wir uns noch die Grundlagen von PowerMock an. Mockito kann static Methoden, final Methoden, Konstruktoren, private Methoden und Enums nicht testen. Wir lernen warum, und wie PowerMock diese Einschränkungen von Mockito dann umgeht und weitere Methoden testen kann.

Dazu erweitern wir unser Bestellsystem um einige weitere Methoden, die wir mit Mockito nicht testen können. Wir setzen hier nun PowerMock schrittweise ein und lernen dabei die wichtigsten Features und Ideen dieses Java-Frameworks kennen.

Nach Absolvierung dieses Kurses sind Sie mit JUnit 5, Mockito und PowerMock so vertraut, dass Sie alle diese Testing Frameworks in Ihrem Projekt einsetzen und umfangreiche Tests damit schreiben können.

Inhalte

JUnit 5 mit Moskito Seminar

- Why Unit Test?
- Test Structure
- Name Conversion
- Test Structure
- Test Lifecycle
- Test Data
- Annotations
- Assertions
- Parameterized Tests
- Exkurs Test Double
- Mockito Framework
- PowerMock Framework

Nutzen

- **Einen schnellen und effektiven Einstieg** in Unit Testing mit Java
- **Kompetente Trainer:** Buchautoren und Universitätsdozenten
- **Kompakt:** auf das Wesentliche optimiert: 2 statt 3 Tage
- **Praxisorientiert:** 30% Theorie und 70% Praxis nach Raed Lehrmethode
- **Intensive Betreuung:** intensiv in kleinen Gruppen. Max. 8 Teilnehmer
- **Individuelle Betreuung:** Schwerpunkte, Themen und Tempo werden während des Kurses - soweit wie möglich - an Ihre Vorkenntnisse angepasst
- **Nachhaltige Betreuung,** falls Sie später Fragen zu den Kursfolien haben
- **Hochwertiges Lernmaterial** und vielfältige Programmieraufgaben
- **Schönes Ambiente mit Verpflegung:** Vormittagssnack, Getränke im Raum (Wasser, Saft, Kaffee, Cola), warmes Mittagessen, Kaffee und Kuchen Pause

Ziele

- JUnit 5 Grundlagen schnell und praxisorientiert erlernen
- Test Double zum Testen von Datenbank/APIs lernen
- Mockito und PowerMock schnell und praxisorientiert erlernen
- Die Qualität von Code und Architektur durch Unit Testing erhöhen
- Ihren Kunden ein Testdokument als Nachweis der Code-Qualität liefern
- Motivation und Begeisterung für das Schreiben von Unit Tests gewinnen

Zielgruppe

Das JUnit 5 und Mockito Seminar an der Coding Akademie München richtet sich an Programmierer/innen mit grundlegenden Java und JUnit Kenntnissen. Im Optimalfall haben Sie zuvor unser JUnit 5 Seminar besucht oder kennen die theoretische Basis. Dann sind Sie in diesem Seminar garantiert richtig.

Programmierer mit Java Grundkenntnissen

Zertifikat

Sie erhalten als Teilnehmer am JUnit 5 und Mockito Kurs ein anerkanntes Zertifikat. Voraussetzung dafür ist die komplette Teilnahme an allen Kurseinheiten und Programmier-Aufgaben und die erfolgreiche Programmierung eines kleinen Abschlussprojektes. Das allerdings wird Ihnen nach diesem intensiven Kurs mehr Freude als Stress bereiten.

Inklusive

Seminarunterlagen und Verpflegung

Kursdauer

2 Tage, je 8 Stunden pro Tag

Maximale Teilnehmeranzahl:

Max. Teilnehmer

8

Die nächsten Termine

Mo., 18.11. - Mi., 20.11.2019

Mo., 10.02. - Mi., 12.02.2020

Don, 09.04. - Fr., 10.04.2020

[Seminar buchen](#)