

Test Driven Development with JUnit

Audience Test Driven Development with JUnit

The course Test Driven Development met JUnit is intended for experienced Java developers who want to apply JUnit for Test Driven Development.

Prerequisites Training TDD with JUnit

Knowledge of and experience with programming in Java is required to join the course Test Driven Development with JUnit.

Realization Test Driven Development with JUnit

The theory is covered on the basis of presentation slides and is interspersed practical exercises. Demos are used to clarify the discussed concepts. The course material is in English.

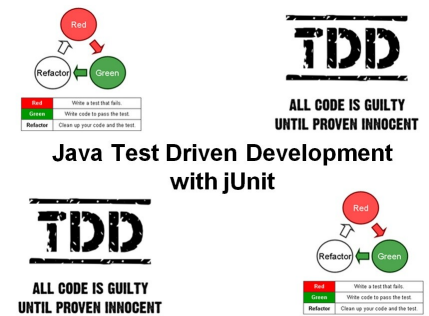
Certification Course TDD with JUnit

Participants receive an official certificate Test Driven Development with JUnit after successful completion of the course.

Duration: 2 days

Price: € 1299

[Open Schedule](#)



Content Course Test Driven Development with JUnit

The course Test Driven Development with JUnit will teach participants the principles of and reasoning behind Test Driven Development and the role of unit testing therein.

JUnit Library

The course Test Driven Development with JUnit starts with an overview of the different types of testing and their use, detailed attention is given to the workings of the JUnit library, the integration of this library in Eclipse and the use of assertions in Test Cases and Test Suites.

Test Fixtures

Fixtures used for the formulation of pre- and postconditions of Test Cases are also discussed as is the automation of tests and the concept of continuous integration.

TDD Methodology

Next the course Test Driven Development with JUnit discusses the methodology of Test Driven Development (TDD), the three rules of TDD and the steps in TDD are explained as are the benefits and limitations of TDD. The participants will exercise TDD by trying to solve so called code Kata's, small programming problems, using TDD.

Stubs en Mocks

After an overview of the importance of writing clean code, the course Test Driven Development with JUnit explains the use of stubs and mocks. These stubs and mocks are used as replacement for code that is not ready yet in a testing environment and can be replaced by real code in a production environment. In this respect the Mockito library is used as an example of a mocking framework.

Database Testing

Finally the course Test Driven Development with JUnit pays attention to database unit testing using DBUnit and the testing of Web Applications using HTMLUnit.

Modules Course Test Driven Development with JUnit

Module 1 : Unit Testing	Module 2 : JUnit	Module 3 : Test Driven Development
What is Unit Testing? Benefits of Unit Testing Manual Testing Automated Testing Time to Test Unit Test Example Unit Testing Best Practices Testing Frameworks Other Types of Testing Continuous Integration Regression Testing Usability Testing Exploratory Testing Acceptance Tests Concurrency Testing	What is JUnit? JUnit Features JUnit View in Eclipse JUnit Test Code JUnit Classes Test Cases TestCase Class TestResult Class JUnitCore Assert Statements Fixtures Test Suites Annotations Special Cases Testing for Exceptions	What is Test Driven Development? Traditional Testing versus TDD Three Rules of TDD Steps in TDD Test Cycles Benefits of TDD Limitations of TDD Testing versus Design TDD Adaptation Behavior Driven Development Designing for Testing Code Kata's Example Kata Domain Model Kata Test and Implementation
Module 4 : Clean Code	Module 5 : Stubs and Mocks	Module 6 : Database Unit Testing
What is Clean Code? Clean Code Principles Technical Debt Meaningful Naming Naming Guidelines What to Avoid Functions Abstraction Level Switch Statements Function Arguments Avoid Side Effects Command Query Separation Good Comments Bad Comments Code Smells	Using Test Doubles What are Stubs? Stub Usage Method under Test Stub HTTP Connection Stubbing Web Server Use Embedded Web Server Stubbing Server Resources Mock Object Simple Mock Example Collaborating Objects Mock Implementation Test using Mock Anti Patterns Using Mockito	Unit Testing Data Access Types of DB Unit Testing Database Integration Unit Testing DB Unit Advantages of DBUnit DB Unit Life Cycle Core Components IDataset Implementations Concrete Database Operations Presetting Database Data Extract Data From Database DBUnit Fixture Setup Tables and Dataset Data Access Test Cases Abstract Class Fixture
Module 7 : Web Application Testing		
Testing Web Applications What is HTMLUnit HTMLUnit Features Simple HTMLUnit Test Imitating Browsers HTML Form Test Finding Specific Elements Button Click Test		