Test Driven Development with Java

Abstract

By now Test Driven Development (TDD) is a well known and appreciated practice exercised by many development teams around the globe. Unlike the name suggests, TDD is not a testing technique but a development technique that results in cleaner high quality code. Implementing effective and useful TDD is a complicated task that requires developers to have both discipline and familiarity with a set of test automation tools. This course explains the TDD methodology, introduces a recommended set of open-source testing tools and includes hands-on exercises (about 50% of the course's duration) to practice the tools.

Target Audience

Java developers, team leaders and project managers.

Prerequisites

Familiarity with the Java language.

Content:

Introduction to Test Driven Development (4 hours):

Traditional software testing.
Functional/Regression/Integration/Unit Testing.
Introduction to Agile software development.
The Test First approach.
Test First challenges.
Automated Testing.
Demo.

JUnit (4 hours):

Introduction.
TestCase.
TestSuite.
Test's life cycle.
Running JUnit from the IDE.

Mock Objects & EasyMock (3 hours):

Mediator objects and Testing.
Introduction to Mock objects.
Introduction EasyMock.
Setting Mock Expectations.
Mock Verification.

Writing efficient tests and testable classes (3 hours):

What should be tested?
How to write a test - Best Practices.

Design for Testability (2 hours):

Should testing change my design?
What is design for testability.
Tips.
Dependency Injection frameworks.

Integration Testing (3 hours):

Challenges.
In Container Testing.
HttpUnit.
DBUnit.
Cactus.

Integrating Test Driven Development in the build process (1 hour):

Introduction to Ant.
Using JUnit from Ant.

Duration: **2 days.**