



Test Driven Development

Duration 2 day(s) (TESTS-02)

Test Driven Development with Java

Description

Master the test-driven development method in a practical and practical context.

Goals

- Understand TDD as design tools
- Use the most suitable tools
- TDD as part of a refactoring and legacy code
- Understand the types of tests and their uses

Public

- Architect
- Developer
- Project Manager

Prerequisites

- Basic knowledge of Java

Structure

30% Theory, 70% Practice

Program

Different types of tests

- Unit tests
- Integration tests
- Functional tests
- Performance tests

Test doubles

- Dummy
- Stub
- Spy
- Mock
- Fake
- Summary

Test Coverage

- Metric
- Mutation Testing

TDD principles

- Origin and Extreme Programming
- TDD cycle
- Emerging design
 - YAGNI Principle
 - Design Patterns

Building Maintainable tests

- Configuration
- Test fixtures
- Naming classes & methods

SOLID principles in the context of TDD

- Single Responsibility Principle (SRP)
- Open/Close Principle (OCP)
- Liskov Substitution Principle
- Interface Segregation Principle
- Dependency Inversion Principle

Legacy Code & Refactoring

- What is a legacy code ?
- Why do I need to refactor ?
- When is refactoring needed ?
- How to apply TDD in legacy code ?

What's Next?

- Behaviour Driven Development

- Acceptance Test Driven Development
- Kata & Coding Dojo
- Continuous integration