

JAV950 : Test Driven Development met JUnit

Code :

JAV950

Duur :

2 dagen

Categorie :

Tooling

Doelgroep :

Deze cursus is bedoeld voor ervaren Java developers die JUnit willen gebruiken voor Test Driven Development.

Voorkennis :

Om aan deze cursus te kunnen deelnemen is kennis van en ervaring met programmeren in Java vereist.

Uitvoering :

De theorie wordt behandeld aan de hand van presentatie slides en wordt afgewisseld met praktische oefeningen. Demos dienen ter verduidelijking van de behandelde concepten.

Inhoud :

De cursus Test Driven Development met JUnit leert de deelnemers de beginselen van en de redenering achter Test Driven Development en de rol van unit testing daarin. Na een overzicht van de verschillende types van testen en het gebruik ervan, wordt uitgebreid aandacht besteed aan de werking van de JUnit library, de integratie van deze library in Eclipse en het gebruik van assertions in Test Cases en Test Suites. Fixtures voor de formulering van de pre-en postcondities van Test Cases worden besproken evenals de automatisering van tests en het concept van continuous integration. Vervolgens komt de methodologie van Test Driven Development (TDD) aan de orde, worden de drie regels van TDD en de stappen in TDD uitgelegd en komen de voordelen en de beperkingen van TDD ter sprake. De deelnemers zullen TDD oefenen op zogenaamde code Kata's, kleine programmeer problemen die zij oplossen met behulp van TDD. Na een overzicht van het belang van het schrijven van clean code, wordt het gebruik van stubs en mocks behandeld. Deze stubs en mocks worden in een test omgeving gebruikt als vervanging voor code die nog niet klaar is. In een productie omgeving komt echte code dan voor de stubs of mocks in de plaats. Hierbij wordt de Mockito library gebruikt als voorbeeld van een mocking framework. Tot slot wordt aandacht besteed aan de database unit testen met behulp van DbUnit en het testen van web applicaties met behulp van HtmlUnit.

Module 1 : Unit Testing

- What is Unit Testing?
- Benefits of Unit Testing
- Manual Testing
- Automated Testing
- Time to Test
- Unit Test Example
- Unit Testing Best Practises
- Using Seams
- Testing Frameworks
- Other Types of Testing
- Continuous Integration
- Regression Testing
- Usability Testing
- Exploratory Testing
- Acceptance Tests
- Concurrency Bug
- Concurrency Testing

Module 4 : Clean Code

- 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
- Comments
- Expressing in Code
- Good Comments
- Bad Comments
- Code Smells

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

Module 2 : JUnit

- What is JUnit?
- JUnit Features
- JUnit Integration
- JUnit View in Eclipse
- JUnit Test Code
- JUnit Classes
- JUnit Class Diagram
- Test Cases
- TestCase Class
- TestResult Class
- JUnitCore
- Assert Statements
- Assert Class
- Fixtures
- Test Suites
- Annotations
- Special Cases
- Testing for Exceptions

Module 5 : Stubs and Mocks

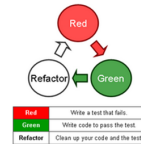
- Using Test Doubles
- What are Stubs?
- Stub Usage
- Method under Test
- Stub HTTP Connection
- Stubbing Web Server
- Use Embedded Web Server
- Stubbing Server Resources
- Using Jetty Handlers
- Mock Object
- Simple Mock Example
- Collaborating Objects
- Mock Implementation
- Test using Mock
- Anti Patterns
- Using Mockito

Module 3 : Test Driven Development

- 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 6 : Database Unit Testing

- 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



Java Test Driven Development with jUnit

