

## Test Driven Development met JUnit

### Doelgroep Cursus Test Driven Development

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

### Voorkennis Cursus TDD met JUnit

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

### Uitvoering Training Test Driven Development

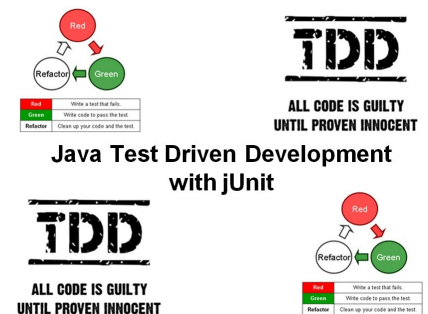
De theorie wordt behandeld aan de hand van presentatie slides en wordt afgewisseld met praktische oefeningen. Demos dienen ter verduidelijking van de behandelde concepten. De cursustijden zijn van 9.30 tot 16.30.

### Certificering Test Driven Development met JUnit

De deelnemers krijgen na het goed doorlopen van de cursus een officieel certificaat Test Driven Development met JUnit.

Duur: 2 dagen

Prijs: € 1499

[Open Rooster](#)


## Inhoud Cursus Test Driven Development met JUnit

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.

### JUnit Library

De cursus Test Driven Development met JUnit gaat van start met 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.

### Test Fixtures

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.

### TDD Methodology

Vervolgens gaat de cursus Test Driven Development met JUnit in op de methodologie van Test Driven Development (TDD), 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 met behulp van TDD.

### Stubs en Mocks

Na een overzicht van het belang van het schrijven van clean code, wordt in de cursus Test Driven Development met JUnit 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.

### Database Testing

Tot slot wordt in de cursus Test Driven Development met JUnit aandacht besteed aan de database unit testen met behulp van DbUnit en het testen van web applicaties met behulp van HtmlUnit.

## Modules Cursus Test Driven Development met 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		