

Java ist eine plattformunabhängige Programmiersprache, die ursprünglich von der Firma Sun Microsystems, entwickelt wurde. Java wird in allen IT-Bereichen eingesetzt und ist für unzählige Betriebssysteme und Plattformen, vom mobilen Telefon bis hin zur Echtzeit-Großrechneranlage geeignet.

Ihr Nutzen

In diesem Seminar lernen Sie warum und wie Softwaretesting zum Erfolg führt und einen wesentlichen Beitrag zur Qualitätssicherung leistet. Sie lernen die Durchführung automatisierter Tests von Java-Programmen sowie die wichtigsten Features und Konzepte von Junit kennen.

Preis pro Teilnehmer

EUR 1450,- exklusive der gesetzlichen MwSt.

Seminardauer

2 Tag(e)/Day(s)

Seminarinhalte

Tag 1:

- * Einstieg und Grundlagen für Unit Testing
- Motivation für Unit Tests
- Grundkonzept Unit Testing
- Unittest vs. Integrationstest
- White-Box-Test vs. Black-Box-Test
- Unit Testing versus Test Driven Development (TDD)
- Testmethodik und Testerstellung
- Der Zusammenhang von Refactoring und Testen
- Ermitteln der Testqualität durch Code-Coverage und Fehlerinjektion
- Anwendung von Stellvertreterobjekten (Dummy & Mock)
- Testdatenerstellung und -pflege
- Testfallfindung
- Grenz- und Extremwerte
- Erstellen einer Testumgebung
- Abgrenzung und Zusammenspiel von Unit Tests mit Integrations- und Systemtests

* Werkzeuge für den praktischen Einsatz

- Unit-Test-Frameworks
- Unit-Test-Runner
- Werkzeuge für Code Coverage
- Mock-Objekte
- User Interface-Testing (UI-Testing)

Tag 2:

- * Junit als Testwerkzeug
- Funktionsweise von Junit
- Erstellen einer Testumgebung
- Entwicklung von testbarem Code (Verwendung von Abstraktion, Schichten, Pattern)
- Der Zusammenhang von Refactoring und Testen
- Ermitteln der Testqualität durch Code-Coverage und Fehlerinjektion
- Ermitteln der Testqualität durch Code-Coverage und Fehlerinjektion
- Testdatenerstellung und -pflege
- Testfallfindung
- Grenz- und Extremwerte
- Äquivalenzklassen
- Abgrenzung und Zusammenspiel mit Integrations- und Systemtests
- Testen von persistenten Daten
- Integration von in Continuous Integration (CI) und Continuous Delivery/Deployment

Voraussetzungen

Fundamentals of the Java Programming Language~4272 oder entsprechende Kenntnisse.

Empfohlen wird auch:

Java Advanced Programming~4273

Hinweise

Version: 14

- Anwendung von Stellvertreter-Objekten (Dummy vs. Stub vs. Spy vs. Fake vs. Mock)
- Ausgewählte Mock-Werkzeuge (Easy-Mock, Jmock, Mockito)
- Persistenz / Datenbanken
- Testen von persistenten Daten
- Mock vs. InMemoryDB

* Übungen und Best Practices

