

Formation JUnit, Maîtriser les tests unitaires en Java

Présentation

Cette formation de 2 jours permet aux développeurs Java de maîtriser l'écriture, l'exécution et la maintenance de tests unitaires avec JUnit. Grâce à une approche centrée sur la pratique, les participants découvriront les bonnes pratiques du test unitaire, l'organisation des tests, l'utilisation des assertions, des annotations JUnit 5, et l'intégration des tests dans un cycle CI/CD. La formation aborde également Mockito pour le test unitaire avec objets simulés.

Durée : 14,00 heures (2 jours)

Tarif INTRA : Nous consulter

Objectifs de la formation

À l'issue de la formation, les participants seront capables de :

- Comprendre le rôle des tests unitaires dans le cycle de vie logiciel
- Utiliser JUnit 5 pour écrire et exécuter des tests unitaires efficaces
- Maîtriser les annotations, assertions, et structures de tests
- Simuler des dépendances avec Mockito
- Intégrer les tests dans une démarche d'intégration continue
- Appliquer les bonnes pratiques de conception orientée test (TDD)

Prérequis

- Maîtrise des bases du langage Java
- Connaissances en POO et manipulation de classes Java
- Aucune connaissance préalable en tests unitaires n'est requise

Public

- Développeurs Java
- Testeurs techniques ou QA souhaitant écrire des tests automatisés
- Architectes ou chefs de projet technique impliqués dans la qualité logicielle

Programme de la formation



Jour 1 : Fondamentaux de JUnit et écriture de tests unitaires

Module 1 : Introduction aux tests logiciels

- Pourquoi tester ? Types de tests (unitaire, intégration, fonctionnel)
- Rôle des tests dans le cycle DevOps / CI/CD

Module 2 : Présentation de JUnit 5

- Architecture (platform, Jupiter, vintage)
- Installation (Maven, Gradle, IDE)

Module 3 : Premier test avec JUnit

- Structure d'un test
- Annotations de base : @Test, @BeforeEach, @AfterEach, @DisplayName
- Assertions : assertEquals, assertTrue, assertThrows, etc.
- TP : Installation de JUnit dans un projet Maven ; création de tests simples

Module 4 : Organisation des tests

- Conventions de nommage, packages, structure de projet
- Regroupement des tests : @Nested, @Tag, @Disabled
- Test de classes métier (calcul, validation, règles de gestion)

Introduction au TDD

- TP : Développement d'une fonctionnalité en TDD (ex : moteur de règles simples)

Jour 2 : Tests avancés, mocks et intégration continue

Module 5 : Cas avancés de tests

- Tests paramétrés (@ParameterizedTest, @CsvSource, @MethodSource)
- Tests d'exceptions, couverture de code

Module 6 : Introduction à Mockito

- Principe des mocks, stubs, spies

- Création de dépendances fictives : mock(), when(), verify()
- Injection avec @InjectMocks, @Mock, MockitoExtension
- TP : Tester un service Java avec dépendances (DAO, API) via mocks

Module 7 : Bonnes pratiques des tests unitaires

- Test isolé vs test d'intégration
- Éviter les pièges : dépendances cachées, tests non déterministes
- Mesure de la couverture (JaCoCo, Sonar)

Module 8 : Intégration dans une chaîne CI/CD

- Lancer les tests dans Maven/Gradle
- Exécution automatique dans GitLab CI / GitHub Actions / Jenkins
- TP final : Création d'un mini-projet testé de bout en bout + pipeline CI avec tests automatisés

Organisation

Formateur

Les formateurs de Docaposte Institute sont des experts de leur domaine, disposant d'une expérience terrain qu'ils enrichissent continuellement. Leurs connaissances techniques et pédagogiques sont rigoureusement validées en amont par nos référents internes.

Riches de leur expérience sur le sujet, ils sauront accompagner vos collaborateurs dans leur montée en compétence.

Moyens pédagogiques et techniques

- Alternance d'apports théoriques et de mises en pratique contextualisées.
- Progression pédagogique structurée, de la compréhension à la mise en œuvre opérationnelle.
- Pédagogie immersive favorisant une mise en application immédiate des compétences.
- Formation interactive et participative s'appuyant sur des ateliers pratiques, études de cas et échanges collectifs (+ 60 % du temps de formation.)

- Séquences pédagogiques rythmées et évaluations formatives régulières pour renforcer l'ancrage des apprentissages.
- Supports de cours, ressources documentaires et références mis à disposition par le formateur

Dispositif de suivi de l'exécution et de l'évaluation des résultats de la formation

En amont de la formation

- Recueil des besoins des apprenants afin de disposer des informations essentielles au bon déroulement de la formation (profil, niveau, attentes particulières...).
- Auto-positionnement des apprenants afin de mesurer le niveau de départ.

Tout au long de la formation

- Évaluation continue des acquis avec des questions orales, des exercices, des QCM, des cas pratiques ou mises en situation...

A la fin de la formation

- Auto-positionnement des apprenants afin de mesurer l'acquisition des compétences.
- Évaluation par le formateur des compétences acquises par les apprenants.
- Questionnaire de satisfaction à chaud afin de recueillir la satisfaction des apprenants à l'issue de la formation.
- Questionnaire de satisfaction à froid afin d'évaluer les apports ancrés de la formation et leurs mises en application au quotidien.

Accessibilité

Nos formations peuvent être adaptées à certaines conditions de handicap. Nous contacter pour toute information et demande spécifique.