JAV107

Tests unitaires en Java avec JUnit

Durée

1 jour.

Pré-requis

Connaissance du langage Java.

Orientation

Développeurs ou chargés de tests impliqués dans la réalisation d'applications Java autonomes, Web ou réparties.

Dates

22 fév 2013 . 21 juin 2013 . 25 oct 2013

Objectifs

Savoir mettre en oeuvre le framework JUnit, dans ses versions 3 et 4 pour concevoir des tests unitaires et les organiser en suites de tests. Les tests unitaires automatisés proposés par JUnit augmentent l'efficacité des équipes de développement en diminuant les temps de développement initial et de maintenance. La méthode de développement TDD est présentée comme guide de développement. Les aspects avancés tels que les tests paramétrés, les objet stub ou mock sont présentés en détail.

Organisation du cours

Théorie : 50%
Pratique : 50%

Configuration matérielle

Une machine par stagiaire. Environnement de développement Eclipse.

Documentation fournie

Support de cours et exercices corrigés.

Tarif H.T. *

1 personne : 520 €
2 personnes : 936 €
3 personnes : 1092 €
* personnes d'une même entreprise inscrites à la même session.

Programme

Présentation générale
- Les différents types de tests, avantages des test unitaires
- Fonctionnalités du framework JUnit
- Architecture des classes JUnit dans les versions 3 et 4

Organisation des tests
- Le processus TDD (Test Driven Development)
- Appliquer les stratégies de test
- Organisation du projet : où placer les classes de tests ?

Tests simples
- La classe TestCase dans JUnit 3, les annotations dans JUnit 4
- Exécution d'un test simple
- Contexte d'exécution des tests : principe d'isolation dans JUnit

Tester les valeurs : assertions et prédicats
- Définition des prédicats avec la classe Assert
- Tester la levée d'une exception avec JUnit 3 et JUnit 4

Définition de test paramétrés avec JUnit 4

Suites de tests
- Construction des suites de tests avec JUnit 3 et JUnit 4
- Exécution des suites de tests

Les interfaces d'exécution disponibles dans JUnit
- Exécution en mode texte
- Exécution en mode graphique
- Intégration à un environnement de développement comme Eclipse
- Automatiser l'exécution des tests avec Ant

Utiliser des objets de substitution pour simuler du code complexe ou non encore développé
- Objets stubs
- Objets mocks, présentation de l'extension JUnit jMock

Extensions à JUnit
- Tester les accès à la base de données avec DBUnit
- Tester les méthodes privées avec JUnitX
- Créer sa propre extension à JUnit