


TDD, BDD et principes SOLID

Développements dirigés par les tests

 Présentiel ou en classe à distance

Durée : 2 jours (14 h)

Réf. : OB216

Prix inter : 1.640,00 € HT

Forfait intra : 4.135,00 € HT

La réussite d'un projet logiciel repose sur la qualité de sa base de code et l'adéquation entre ce qui est développé et les besoins des utilisateurs finaux. La mise en application des principes SOLID permet de garantir la qualité du code produit et sa maintenabilité à long terme. Cette formation s'appuie sur cette première présentation et présente ensuite la pratique du développement dirigé par les tests, déclinée en deux volets. Le premier, le TDD, dans lequel les tests sont écrits de façon classique et le second, le BDD qui repose sur des tests écrits en langage naturel.

Les objectifs de la formation

- Comprendre le rôle des tests dans le processus de développement des logiciels, les enjeux de l'écriture de tests automatisés et le coût de la mise en production de code non testé automatiquement
- Comprendre ce qu'est un test d'acceptance et comment découper une fonctionnalité en tests d'acceptance
- Pouvoir maîtriser l'écriture de tests automatisés, unitaires et tests d'intégration ainsi que les différentes étapes du développement TDD : découpage d'un problème en tâches élémentaires, écriture de tests ciblés et recouvrant, cycle TDD
- Être en mesure de maîtriser les principes et patterns du "clean code" en programmation objet, les outils de frameworks de test, la syntaxe Gherkin pour l'écriture de tests d'acceptance, le framework Cucumber pour exécuter des tests d'acceptance
- Être capable d'appliquer ces principes et patterns au développement de nouveau code (greenfield development) et au code existant (brownfield development)

A qui s'adresse cette formation ?

Pour qui

- Développeurs expérimentés

Prérequis

- Bonne maîtrise du langage Java et de ses API fondamentales
- Une première expérience dans l'écriture de tests sur du code legacy est un avantage

Programme

1 - Introduction à la pratique du développement dirigé par les tests, le cycle de développement TDD

- Exemple du Kata FizzBuzz

- Développement du Kata
- Bilan : les étapes de la pratique du TDD

2 - Introduction des principes du Clean Code et des principes SOLID

3 - Pratiques de l'eXtreme Programming

- Pratiques des Katas
- Pratiques des Coding Dojo
- Pratiques du Pair Programming

4 - Le principe Open / Closed : application au pattern Strategy

- Exemple du Kata RPN Calculator
- Développement du Kata
- Bilan : le pattern Strategy et son implémentation

5 - Utilisation de Gherkin / Cucumber pour l'écriture de tests

- Écriture de tests en Gherkin : méthodes, organisation, syntaxe
- Écriture de classes Cucumber pour l'exécution de ces tests
- Fonctionnalités avancées : tests paramétrés, tables de données, tags
- Intégration avec Maven, génération de rapports de tests

6 - Mise en oeuvre sur un kata complexe : Mars Rover

- Écriture des tests avec Cucumber
- Résolution du kata
- Conclusion

Evaluation

- Cette formation fait l'objet d'une évaluation formative.

Les points forts de la formation

- Le programme se compose de 30% de présentations théoriques et de 70% de travaux pratiques (codage).
- La partie codage porte principalement sur le codage de "Katas", un exercice essentiel pour s'exercer à la pratique du TDD / BDD.
- Une formation animée par des consultants-formateurs expérimentés et reconnus par leurs pairs.

Dates et villes 2024 - Référence OB216

A distance

du 10 juin au 11 juin

du 12 sept. au 13 sept.

du 9 déc. au 10 déc.

Paris

du 10 juin au 11 juin

du 12 sept. au 13 sept.

du 9 déc. au 10 déc.