

## Caractéristiques

- Durée : 3 jours
- Modalité : présentiel
- Niveau : intermédiaire
- Type : professionnalisation
- CPF : non
- Formacode : non
- Certification possible : non

## Méthodes pédagogiques

- Pédagogie par objectif
- Alternance de pédagogie par exposé et de pédagogie participative sur cas pratiques professionnels
- Productions individuelles
- Evaluation et régulations individualisées
- Possibilité de certification suivant cursus

## Moyens et supports

- 5 salles pouvant accueillir entre 6 à 12 stagiaires
- Postes de travail en réseau sous Windows 10 Pro ou ultérieur
- Kit de vidéo projection ou grand écran connecté
- Supports de cours et d'exercices papiers et numériques
- Supports complémentaires possibles par e-learning
- Questionnaire d'évaluation et certificat de stage
- Formateur professionnel de plus de dix ans d'expérience et/ou expert de Bac+5 à Bac +7

## Objectifs

A l'issue de ce stage, les participants auront acquis :

- De mettre en œuvre des tests unitaires automatisés
- D'améliorer la stabilité d'une application grâce aux techniques de tests

## Prérequis et public concerné

- Tout développeur Java ou tout autre développeur familier avec un langage objet
- Tout testeur désireux d'automatiser ses suites de tests
- Bonne connaissance d'un langage objet : Java, C# ou PHP

## Programme

### Résumé des compétences à acquérir

- Connaître les différents types de test en informatique
- Comprendre les enjeux des tests automatisés
- Concevoir un cas de test
- Utiliser un framework de test (JUnit)
- Utiliser un framework de build automatisé (Maven)
- Mesurer la couverture de code
- Utiliser les doubles de test

### Méthodologie

- Familles de tests
- Responsabilités
- Test unitaire
- Historique
- Enjeux

### Technique du test

- Vocabulaire
- Définir l'unité
- Frameworks xUnit
- Limites

### Java / Junit

- Versions
- Conventions
- Les annotations
- Les assertions
- Les suppositions
- Les règles
- Les ressources
- Les paramètres
- Les matchers
- Suites de test
- Intégration avec Maven

### Pratiques additionnelles

- Couverture de code
- TDD ou « Test Driven Development »
- BDD ou « Behavior Driven Development »
- Les doubles de tests

### Ouverture

- Bonnes pratiques
- Problèmes communs
- Solutions
- Vers d'autres types de test