

# Formation Améliorer la qualité des applications Java avec les tests unitaires



Dans tout projet de développement, les tests représentent une étape indispensable pour obtenir des applications de qualité. Les dernières méthodologies de production comme l'eXtreme Programming place le test comme activité centrale dans le processus de production d'applications. Cette formation essentiellement axée sur la pratique, permettra aux participants de mettre en place les différents types de tests usuels sur une application typique

## Objectifs

---

- Découvrir les outils permettant d'automatiser les tests d'une application
- Savoir mettre en place une série de tests
- Maîtriser les bonnes pratiques de tests unitaires
- Découvrir différentes méthodes de tests

## Public concerné

---

- Développeurs
- Chefs de projet, architectes impliqués dans la réalisation d'applications Java J2EE

## Pré requis

---

- Connaissances de Java

## Une formation de 3 jours

---

Caractéristiques
<b>Tarif : 1605 € HT par personne</b>
<b>Numéro de formateur : 11753687675</b>
<b>Nombre d'heures : 21</b>
<b>Référence : OB381</b>
<b>Contact : Patrick LE GOFF</b>
<b>Telephone : 01.76.60.66.10</b>
<b>Email : <a href="mailto:contact@kaptive.com">contact@kaptive.com</a></b>

## Description des modules

num	Module
<b>1</b>	<b>Introduction aux tests des applications J2EE</b>
<b>Détails</b>	<ul style="list-style-type: none"><li>- Qu'est-ce que le test ? Sa problématique et son importance</li><li>- Les types de tests : tests unitaires, tests d'intégration, tests fonctionnels, tests de charge...</li><li>- Les moyens : les plates-formes de test, l'instrumentation, l'automatisation, l'organisation des activités de test</li><li>- Tests et conduite de projet : problématique de l'organisation des tests, comment gérer l'incertitude, la documentation liée aux tests</li><li>- Tests et méthodologies : tests et USDP, tests et UML, l'approche eXtreme Programming</li><li>- Particularités des applications J2EE : différence entre les tiers, prise en compte de la concurrence, notion d'anomalies</li></ul>
<b>2</b>	<b>Tests unitaires et JUnit</b>
<b>Détails</b>	<ul style="list-style-type: none"><li>- La finalité des tests unitaires</li><li>- Le test, pourquoi s'en priver ? la simplicité de JUnit</li><li>- Séparation de la classe test et de la classe à tester</li><li>- Définition de la classe de tests et des cas de tests</li><li>- Mise en place des suites de tests</li><li>- L'automatisation des tests avec Ant, JUnit et Eclipse, plug-ins disponibles, paramétrage de l'environnement, écriture des cas de tests, des suites des tests, exécution du test</li></ul>
<b>3</b>	<b>Tests fonctionnels HttpUnit</b>
<b>Détails</b>	<ul style="list-style-type: none"><li>- Différence entre scénarii enregistrés et utilisation d'API</li><li>- L'approche d'HttpUnit : test-first development, l'importance des cas d'utilisation</li><li>- Différentes versions, installation et dépendances avec les autres librairies Open Source</li><li>- Parcours de l'API disponible</li><li>- La classe WebConversation, les utilitaires pour parser la réponse</li><li>- Combiner JUnit et HttpUnit</li><li>- Extensions d'HttpUnit</li></ul>