

# Formation Concevoir et développer des applications Java en équipe projet



Une programmation Java de qualité nécessite aujourd'hui de maîtriser en simultan  les aspects syntaxe, mais aussi conception et communication avec UML et les outils de base n cessaires sur un projet pour automatiser et documenter les d veloppements. Ce stage permet aux programmeurs Java exp riment s d' voluer vers le m tier de concepteur/d veloppeur

## Objectifs

- Passage en revue des concepts de base d'une architecture applicative : syst me, couches logicielles, composants, packages
-  liminer les d pendances entre les diff rents composants de vos applications
- Description du r le des interfaces dans la communication des composants
- Passage en revue de la syntaxe et des principales API du langage Java
- Acquisition des bonnes pratiques relatives aux concepts Objets avec UML, Design Patterns et Refactoring d'applications
- Une exp rience de d veloppement d'application N-tiers avec deux interfaces : client riche Swing et client l ger Web
- La bo te   outils Java des  quipes de RetD : CVS/Subversion, Ant, Maven, Log4j, JUnit, i18n, g n ration de code   partir d'UML

## Public concern 

- Concepteurs/d veloppeurs
- D veloppeurs C, C++ exp riment s
- Programmeurs Java exp riment s

## Pr  requis

- Ma triser Java et UML

## Une formation de 5 jours

Caract�ristiques
<b>Tarif : Nous consulter</b>
<b>Num�ro de formateur : 11753687675</b>
<b>Nombre d'heures : 35</b>
<b>R�f�rence : OB340</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
1	<b>Passage en revue de la syntaxe Java et des nouveautés Java 1.5</b>
<b>Détails</b>	- Nouveautés de Tiger 1.5 - Syntaxe
2	<b>Concepts de base d'une architecture applicative</b>
<b>Détails</b>	- Découpage d'une application Java - Couches logicielles - Architectures N-Tiers - Dépendances entre couches / composants - Réduction des dépendances avec IOC
3	<b>Séparer les couches domaine métier et les interfaces avec designs patterns</b>
<b>Détails</b>	- Construire la couche métier de votre application - Construire une interface Swing - Construire une interface Web - Mise en oeuvre des patterns les plus utilisés (DAO, DTO, MVC...)
4	<b>Packaging d'objets et composants en archives avec l'utilitaire JAR</b>
<b>Détails</b>	- Produire un composant logiciel - Utilité des Diagrammes de composants UML - Comment découper son application en composants ?
5	<b>Améliorer la génération et configurations de vos Logs avec Log4j</b>
<b>Détails</b>	- Filtrer dynamiquement vos Logs - Modifier le layout de vos logs - Ajouter une nouvelle destination
6	<b>Gérer facilement l'internationalisation de l'API i18n</b>
<b>Détails</b>	- Supporter plusieurs langues dans vos applications - Ajouter "à chaud" de nouvelles langues à vos applications
7	<b>Gestion des exceptions</b>
<b>Détails</b>	- Gestion locale ou centralisée des exceptions avec Try/Catch et Throw - Concevoir vos propres exceptions métier
8	<b>Génération de code à partir d'UML et Re-factoring d'applications</b>
<b>Détails</b>	- Générer le code Java à partir de diagrammes de classes et de séquences - Refactoring/Reverse Engineering d'applications
9	<b>Travailler en équipe avec CVS/Subversion</b>
<b>Détails</b>	- Gestion de configuration avec CVS/Subersion - Les versions - Les branches et les conflits - Systèmes centralisés ou locaux
10	<b>Automatisation des builds avec ANT et Gestion de projet avec MAVEN</b>
<b>Détails</b>	- Créer des scripts ANT - Automatisation des tâches - Gérer son projet avec MAVEN - Générer des rapports sur les dépendances du projet, les résultats et statistiques des tests, le pourcentage du code testé, la complexité du code, le respect d'une norme de codage