

# Java pour les développeurs Objet expérimentés



Les concepts de modélisation objet sont à la base d'une grande majorité des développements d'entreprise. Pour construire des applications performantes, le langage JAVA intègre, de façon native, l'ensemble des briques nécessaires à un développement orienté objet. Les participants à cette formation découvriront les notions fondamentales de l'objet mais également leur mise en oeuvre avec Java

## Objectifs

- Connaître les composants et tiers techniques de la plate-forme Java
- Apprendre la syntaxe et les principales APIs de base du langage Java
- Se connecter et exécuter des requêtes sur un SGBDR
- Passer d'un langage de POO à Java
- Découvrir les nouveautés de Java-Tiger

## Public concerné

- Concepteurs
- Développeurs C
- , C++ expérimentés
- Programmeurs Java souhaitant passer à l'objet

## Pré requis

- Maîtriser un langage de programmation objet (C
- , C++)

## Une formation de 4 jours

Caractéristiques
<b>Tarif : 1740 € HT par personne</b>
<b>Numéro de formateur : 11753687675</b>
<b>Nombre d'heures : 28</b>
<b>Référence : OB280</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>Présentation de Java (Architectures N-Tiers, J2SE, J2EE, JVM, JAR et JavaDoc)</b>
<b>Détails</b>	<ul style="list-style-type: none"> <li>- Présentation de Java et de ses tiers techniques J2SE/J2EE/J2ME et la JVM</li> <li>- Présentation des composants techniques de Java (JDK), principales APIs</li> <li>- Présentation des outils Java : JavaDoc, Jar et co</li> <li>- Présentation des principaux IDE : Eclipse, NetBeans</li> </ul>
2	<b>Les bases du langage : La Syntaxe</b>
<b>Détails</b>	<ul style="list-style-type: none"> <li>- Les Variables : déclaration, initialisation, instruction de contrôle, boucles et itérations</li> <li>- La syntaxe objet en Java : classe, héritage, classes abstraites et interfaces</li> </ul>
3	<b>Gestion des erreurs avec les exceptions (Try/Catch, Exceptions métiers)</b>
<b>Détails</b>	<ul style="list-style-type: none"> <li>- Les différentes exceptions en Java</li> <li>- Les blocs try/catch/finally</li> <li>- Gestion Locale ou centralisée</li> <li>- Créer ses propres exceptions métier</li> </ul>
4	<b>Bibliothèques standards (Collection, Date, Entrées/Sorties)</b>
<b>Détails</b>	<ul style="list-style-type: none"> <li>- Gestion des tableaux et collections</li> <li>- Les dates et calendrier</li> <li>- Les Entrées / Sorties de fichiers</li> </ul>
5	<b>Nouveautés de Tiger</b>
<b>Détails</b>	<ul style="list-style-type: none"> <li>- Améliorations et nouveautés du langage</li> <li>- Les nouvelles APIs majeures</li> <li>- Les annotations</li> <li>- Les Generics</li> </ul>
6	<b>Accès aux SGBDR avec JDBC (Connection, Query, ResultSet, ResourceBundle, ORM)</b>
<b>Détails</b>	<ul style="list-style-type: none"> <li>- Présentation de Java et JDBC</li> <li>- Connexion à une BDD, exécution d'une requête et itération sur les résultats (connexion, Statement et Resultset)</li> <li>- Externaliser les paramètres avec les fichiers Properties et ResourceBundle</li> <li>- Introduction au Mapping Objet-Relationnel</li> </ul>
7	<b>Introduction au modèle MVC (Modèle/Vue/Contrôleur)</b>
<b>Détails</b>	<ul style="list-style-type: none"> <li>- Architecture MVC</li> <li>- Le Modèle</li> <li>- La Vue</li> <li>- Le Contrôleur</li> </ul>
8	<b>Aperçu de AWT et SWING (Composants, Conteneurs, Layouts)</b>
<b>Détails</b>	<ul style="list-style-type: none"> <li>- Les composants et conteneurs (JButton, JTextField, JPanel, JFrame)</li> <li>- Gestionnaire de mise en forme : les Layouts</li> <li>- Gestionnaire des événements : les Listeners</li> </ul>
9	<b>API pour les fichiers de logs</b>
<b>Détails</b>	<ul style="list-style-type: none"> <li>- Générer des logs avec Log4j</li> <li>- Filtrer les Logs par niveau d'importance</li> <li>- Configurer la / les destinations où sont générés les Logs</li> </ul>
10	<b>Améliorer la qualité d'une application grâce aux tests unitaires</b>
<b>Détails</b>	<ul style="list-style-type: none"> <li>- Créer des tests unitaires JUnit</li> </ul>

- Créer des suites de tests avec JUnit
- Générer des rapports

## 11 Préparer une application à l'internationalisation

**Détails** - Gérer les langues avec l'API ResourceBundle et les fichiers properties