



# Formation JAVA 6

## Plan de cours détaillé

### Objectifs:

Développer les compétences clés et les réflexes afin d'intégrer de façon productive une équipe de développement JAVA

### Connaissances préalables :

Une expérience dans un langage structuré est nécessaire (c/c++, Visual Basic, PHP, Delphi)

Ce stage constitue le point d'entrée pour toutes les formations Java et frameworks

### Profils participants :

développeurs, chefs de projet

### Durée : 5 jours

<p><b>Introduction</b> Historique : java a 15 ans !</p> <ul style="list-style-type: none"> <li>✦ Atouts : Portabilité, Sécurité, Réseau</li> <li>✦ Comparaison avec C/C++, PHP, .NET, Visual basic</li> <li>✦ Lien entre java et java entreprise (jee)</li> <li>✦ Liens utiles : site de sun et javadoc API</li> </ul>	<p><b>Technologie Java</b></p> <ul style="list-style-type: none"> <li>✦ outils clés : javac, java, jar, javadoc</li> <li>✦ Byte code, machine virtuelle</li> <li>✦ Ramasse-miettes</li> <li>✦ Gestion des exceptions</li> </ul>
<p><b>Environnement Eclipse</b></p> <ul style="list-style-type: none"> <li>✦ Historique, versions</li> <li>✦ Comparaison Eclipse / Netbeans / Jbuilder</li> <li>✦ Notions de vues, perspectives, plugins,</li> <li>✦ Intégration Javadoc,</li> <li>✦ Export Jar</li> </ul>	<p><b>API Java 6</b></p> <ul style="list-style-type: none"> <li>✦ java.lang, java.util, java.io</li> <li>✦ JVM client / serveur</li> <li>✦ Technologies d'intégration (JDBC, RMI, JNDI)</li> <li>✦ Technologies de déploiement</li> <li>✦ Généricité, bouche for étendue</li> <li>✦ Scripting, annotations</li> <li>✦ Surveillance avec JMX, outil jconsole</li> <li>✦ Sécurité, Multithreading, Logging , XML</li> </ul>
<p><b>Bases du langage</b></p> <ul style="list-style-type: none"> <li>✦ Types de données élémentaires</li> <li>✦ Tableaux</li> <li>✦ Chaînes de caractères</li> <li>✦ Opérateurs et Instructions</li> <li>✦ Notion de package</li> <li>✦ Gestion des exceptions</li> </ul>	<p><b>Définition de classes Java</b></p> <ul style="list-style-type: none"> <li>✦ Définitions de classes et fichiers source</li> <li>✦ méthodes, arguments,</li> <li>✦ Champs, portée public, private, protected</li> <li>✦ Organisation des fichiers et packages</li> <li>✦ Importation de classes</li> <li>✦ Contrôle des accès</li> <li>✦ Constructeurs , this, super</li> <li>✦ Conteneurs du JDK et classes string</li> <li>✦ Méthode toString()</li> <li>✦ Méthodes main() et points d'entrée</li> <li>✦ Annotations</li> <li>✦ Javabeen, POJO</li> </ul>
<p><b>Classes utilitaires</b></p> <ul style="list-style-type: none"> <li>✦ Classe de base : Object, Integer, String,</li> <li>✦ StringBuffer, Math</li> <li>✦ Les Collections : Vector, List, HashMap, ...</li> </ul>	<p><b>Entrées / Sorties</b></p> <ul style="list-style-type: none"> <li>✦ Lectures / écritures dans un fichier</li> <li>✦ Lectures / écritures à l'écran</li> <li>✦ Gestion des répertoires attributs de fichiers</li> </ul>

	<ul style="list-style-type: none"> <li>✦ Sérialisation et Introspection</li> </ul>
<b>Concepts objets</b> <ul style="list-style-type: none"> <li>✦ Classe et instance de Classe</li> <li>✦ Polymorphisme</li> <li>✦ Encapsulation</li> <li>✦ Héritage</li> <li>✦ Interface</li> </ul>	<b>Analyse et conception objet</b> <ul style="list-style-type: none"> <li>✦ Roles : Analyste, concepteur, réalisateur</li> <li>✦ Notation UML</li> <li>✦ Diagrammes de classes</li> <li>✦ Visibilité</li> <li>✦ Cycle de vie d'un projet</li> </ul>
<b>Polymorphisme et héritage</b> <ul style="list-style-type: none"> <li>✦ Super-classes et sous-classes</li> <li>✦ Redéfinition de méthodes</li> <li>✦ Liaisons statique et dynamique</li> <li>✦ Appel de constructeurs de super-classes</li> </ul>	<b>Accès aux données</b> <ul style="list-style-type: none"> <li>✦ Introduction JDBC</li> <li>✦ Différences api java.sql et javax.sql</li> <li>✦ Approche classique : Connection, Statement, ResultSet</li> <li>✦ Interface Rowset, et comparaison rowset/resultset</li> <li>✦ Notion de transaction. Autocommit JDBC</li> </ul>
<b>Interface utilisateur (IHM)</b> <ul style="list-style-type: none"> <li>✦ IHM utilisant Swing / AWT</li> <li>✦ JFrame, Layout, Panels</li> <li>✦ Listener d'événements</li> </ul>	<b>Tests unitaires avec JUnit</b> <ul style="list-style-type: none"> <li>✦ Présentation et utilité des tests</li> <li>✦ Différent types de test (intégration, fonctionnel, performance)</li> <li>✦ Définition et rôle d'un test unitaire</li> <li>✦ Présentation Junit : TestCase, TestSuite</li> <li>✦ Intégration JUnit dans Eclipse</li> </ul>
<b>Gestion des logs avec Log4j</b> <ul style="list-style-type: none"> <li>✦ Les limites de System.out.println</li> <li>✦ Présentation Log4j</li> <li>✦ Logger, Appenders, Layout</li> <li>✦ log4j.properties, log4j.xml</li> </ul>	