

## Formation Java SE Initiation + Approfondissement

<b>Durée :</b>	5 jours
<b>Public :</b>	Tous
<b>Pré-requis :</b>	Notions de programmation
<b>Objectifs :</b>	Réaliser et déployer des applications complètes, performantes et maintenables en Java - Savoir choisir les technologies adaptées et mettre en place des interfaces efficaces - Connaître et maîtriser les concepts avancés de la programmation Java
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	JAV28-F
<b>Demandeurs d'emploi:</b>	Des entreprises recrutent des demandeurs d'emploi qui ont suivi ce cours dans le cadre d'une POEI, contactez-nous au 09.72.37.73.73 pour plus d'informations.
<b>Note de satisfaction des participants:</b>	4,59 / 5

### Découvrir la plateforme Java

Historique, versions  
Editions Java : Java SE, Java EE, Java ME  
Compilation et interprétation par la JVM (Java Virtual Machine)  
Technologies/frameworks Java et positionnement  
Environnement de développement  
Empaquetage et déploiement d'une application Java

**Atelier : Installation du JDK (Java Development Kit) et d'un IDE (Eclipse/NetBeans) - Structure d'un projet, buildPath,...**

### Maîtriser les bases

Utilisation de variables, constantes, opérateurs  
Types simples et types références  
Transtypage, Wrappers  
Expression de conditions : if/else, switch, opérateur ternaire  
Utilisation de boucles : for, while, do while  
Manipulation de tableaux  
Factorisation de codes avec méthodes  
Surcharge, arguments variables, récursivité  
Commenter et documenter du code

## **Atelier : Multiples exemples de manipulation de structures de contrôles et de fonctions**

### **Apprendre l'objet**

Définition de classes, POJO vs JavaBean  
Déclaration des membres d'instance / de classe (static)  
Constructeurs et instanciation  
Cycle de vie d'un objet en mémoire  
Diagramme de classes (UML)  
Agrégation d'objets (association)  
Encapsulation : getters et setters  
Extension de classes (Héritage)  
Comparaison d'objets  
Classes abstraites  
Interfaces et implémentation  
Polymorphisme

### **Atelier : Modélisation de problèmes en objet**

### **Gérer les exceptions**

Définition, types d'exceptions  
Capturer et traiter une exception (try/catch/finally)  
Lever/Remonter une exception (throw/throws)  
Création d'exceptions

### **Atelier : Gestion des exceptions susceptibles d'être déclenchées dans une application**

### **Utiliser des collections**

Présentation de l'API disponible, generics  
Comparatif, choix d'un type de collection  
Classes essentielles : ArrayList, HashMap, ...  
Parcours, opérations sur des collections et tris

### **Atelier : Manipulation de collections d'objets**

### **Manipuler des fichiers**

Flux binaires / caractères  
Lecture et écriture de fichiers  
Utilisation de buffers  
Manipulation de chemins, répertoires, surveillance  
Sérialisation d'objets : binaire, XML  
Externalisation de configuration dans des .properties  
Gestion des logs : java.util.logging, Log4j

### **Atelier : Implémentation d'exports et imports depuis des fichiers**

### **Construire des interfaces graphiques**

Présentation de Swing : containers, widgets  
Fenêtres modales/non modales, boîtes de messages  
Positionnement des contrôles

Gestion des évènements : claviers, souris  
WYSIWYG disponible  
Gestion du redimensionnement

**Atelier : Application complète de gestion**

**Accéder à des bases de données**

Présentation de l'API JDBC  
Ecriture de requêtes et traitement des résultats  
Gestion des transactions  
Mapping relationnel objet (Pattern DAO)  
Présentation de frameworks ORM

**Atelier : Organisation et implémentation d'une couche d'accès aux données**

**Notions avancées**

Communications réseau  
Gestion des processus  
Options JVM  
Optimiser du code, généricité  
Introspection (Reflection API)

**Atelier : optimisation de code et déploiements**

**Passage de la certification (si prévue dans le financement)**