



# LES FONDAMENTAUX DE LA PROGRAMMATION JAVA

## INFORMATIONS

### CONTACT

03 88 47 10 96

mfo@metaformose.org

### A QUI S'ADRESSE LA FORMATION ?

- Développeurs
- Chargés de développement d'applications informatiques

### MÉTHODES ET OUTILS PÉDAGOGIQUES

- Formation en présentiel, interactive axée sur la pratique pour une meilleure compréhension et application
- Supports vidéo et exercices
- Documents informatiques et papiers

### PREREQUIS

- Disposer d'une expérience d'un langage de programmation
- Connaître les principes de la programmation orientée objet
- Connaissance des concepts de bases de données relationnelles et du langage SQL
- Avoir déjà développé et livré une application

### NOMBRE DE PARTICIPANTS

2 à 8 personnes

### DURÉE DE L'INTERVENTION

5 journées soit 35 heures

9h-12h30 et 13h30h-17h

### EVALUATION

- Contrôle des connaissances en cours de formation, tests, questionnaires
- Fiche d'évaluation et de satisfaction stagiaire
- Attestation individuelle de formation

### INTERVENANTS

- Formateurs seniors experts en développement

## LES OBJECTIFS DE LA FORMATION

- » Maîtriser la syntaxe et les principales APIs du langage Java
- » Savoir programmer en Java en toute autonomie
- » Pouvoir manipuler les Frameworks les plus utilisés du langage tels que JUnit, log4j ou i18n
- » Être en mesure de manipuler des données à l'aide de JDBC
- » Savoir appliquer le paradigme de la programmation fonctionnelle
- » Découvrir les nouveautés de Java 9

## LE PROGRAMME DE LA FORMATION

### 1. Présentation de l'écosystème Java

- Introduction à Java et à ses tiers techniques J2SE/J2E/J2ME et la JVM
- Présentation des composants techniques de Java (JDK) ainsi que des principales APIs
- Les outils Java : JavaDoc, Jar et co
- Panorama des principaux IDE : Eclipse, NetBeans et IntelliJ

### 2. L'ide Eclipse

- Notions de workspace, vues et perspectives
- Présentation des différents wizard
- Extensions avec Eclipse Marketplace
- Les accès aux bases de données et outils serveurs dans Eclipse

### 3. Les bases du langage : la syntaxe

- Déclaration de variables
- Les opérateurs
- Initialisation
- Instructions de contrôle
- Boucles et itérations
- Notions de visibilité et de variable de classe Vs. variable d'instance

### 4. Les concepts orientés objet en Java

- La classe et ses attributs / méthodes (nombre d'arguments fixe et variable)
- Les constructeurs
- Les références
- L'encapsulation, les imports et packages
- Les interfaces
- Les classes abstraites
- L'héritage
- Unboxing et l'autoboxing

### 5. Gestion des erreurs avec les exceptions

- Qu'est-ce que qu'une exception ?
- Les différentes exceptions en Java
- Les blocs try/catch/finally
- Gestion locale ou centralisée
- Créer ses propres exceptions métier

### 6. Bibliothèques standards (collection, date, entrées/sorties)

- Gestion des tableaux et collections
- Les dates et calendriers

- Les entrées / sorties de fichiers
- La sérialisation

### 7. Accès aux Sgbd avec Jdbc

- Présentation de la librairie JDBC
- Connexion à une base de données, exécution d'une requête et itération sur les résultats avec les interfaces Connection, Statement et ResultSet
- Requêtes paramétrées et requêtes stockées

### 8. Fichiers de propriétés et l'api d'internationalisation (i18n)

- Accès et manipulation des fichiers .properties
- Comment faire des applications en plusieurs langues
- Resource Bundle

### 9. Api pour les fichiers de logs

- Générer des Logs avec Log4j
- Filtrer les Logs par niveau d'importance
- Configurer la / les destinations où sont générés les logs

### 10. Améliorer la qualité d'une application grâce aux tests unitaires

- Créer des tests unitaires avec JUnit
- Créer des suites de tests avec JUnit
- Générer des rapports
- Introduction à la notion de couverture de code et exemples de frameworks
- Options avancées de la perspective Debug

### 11. Programmation fonctionnelle et lambda expressions

- Introduction à la programmation fonctionnelle
- Exemples de Lambda Expressions
- Référentiels et méthodes

### 12. Introduction aux nouveautés Java 9

- La modularisation via le projet Jigsaw
- Fabriques pour les collections
- Améliorations de la classe Optional et de l'API Stream
- Un Shell Java : REPL jShell