

# JAVA, programmation avancée

## INFORMATIONS GÉNÉRALES

**Type de formation :** Formation continue

**Éligible au CPF :** Non

**Domaine :** Développement

**Action collective :** Non

**Filière :** Action collective ATLAS "Java, .Net, C++"

**Rubrique :** Java - Jee - Android

**Code de formation :** F28049

## € Tarifs

**Prix public :** 2295 €

### Tarif & financement :

Nous vous accompagnons pour trouver la meilleure solution de financement parmi les suivantes :

- **Le plan de développement des compétences** de votre entreprise : rapprochez-vous de votre service RH.
- **Le dispositif FNE-Formation.**
- **L'OPCO** (opérateurs de compétences) de votre entreprise.
- **Pôle Emploi** sous réserve de l'acceptation de votre dossier par votre conseiller Pôle Emploi.
- **CPF** -MonCompteFormation

[Contactez nous](#) pour plus d'information

## PRÉSENTATION

### Objectifs & compétences

Maîtriser les aspects avancés du langage Java Comprendre comment appliquer les principaux Frameworks et bibliothèques Java Comprendre le modèle de sécurité et le chargement des classes Java Savoir programmer en Java dans le contexte des bases de données relationnelles Savoir tester, debugger et optimiser ses applications Appréhender les nouveautés Java

### Public visé

Développeurs, chargés de développement d'applications informatiques, chefs de projets proches du développement...

### Pré-requis

Maîtriser le langage Java, connaître les concepts de bases de données relationnelles et du langage SQL, disposer d'une expérience en programmation Java.

## 📍 Lieux & Horaires

**Campus :** Ensemble des sites

**Durée :** 28 heures

### Délai d'accès :

Jusqu'à 8 jours avant le début de la formation

**Distanciel possible :** Oui

## PROGRAMME

### Chapitre 1 : Invariants et records

- Rappel sur les invariants (String, Class ...)
- Les records en Java 16
- Création d'un record

### Chapitre 2 : Evolution des interfaces

- Rappel
- Evolution de l'interface du Java 5 à Java 9

### Chapitre 3 : Les génériques

- Le rôle des wrappers
- Les méthodes generic
- Créer des classes et interface génériques
- Utiliser l'héritage avec les generics
- Faire usage de la syntaxe avancée \*, ?, extends, super Exemple d'exercice : En partant d'un code trop 'copié/collé' d'accès aux données, mise en place des generics pour gagner en maintenabilité

### Chapitre 4 : Les annotations

- Créer ses propres annotations
- @Retention
- @Target
- @Inherited

### Chapitre 5 : La gestion de la mémoire

- Les options du Garbage Collector

## 📅 Prochaines sessions

Cliquez sur la date choisie pour vous inscrire :

■ 23 / 09 / 2024

- 📍 : Ensemble des sites
- ✓ : Distanciel possible
- 🕒 : 28 heures
- 📅 : 4 jours

■ 04 / 11 / 2024

- 📍 : Ensemble des sites
- ✓ : Distanciel possible
- 🕒 : 28 heures
- 📅 : 4 jours

- Les options de la JVM
- Les classes Java dans `java.lang.ref`
- `WeakReference`
- `PhantomReference`
- `SoftReference`

Atelier pratique : Mise en pratique des `WeakReference` avec une attention particulière sur les effets de bords possibles.

### **Chapitre 6 : L'introspection / réflexion en JAVA**

- Les objectifs et les principes du réflexif.
- L'objet `java.lang.class`
- Les objets du package `java.lang.reflect`
- `Method`
- `Field`
- `Constructor`
- `Modifier`
- `Parameter`
- Instanciation, appel, récupération d'attribut en introspection

### **Chapitre 7 : Les modules en JAVA 9**

- Définition
- Exposition des packages
- Gestion des dépendances de module
- Utilisation et exposition de services
- L'archive via `jmode`
- Les mini VM avec `jlink` Atelier pratique : En repartant d'un exercice simple, nous allons créer des modules afin de bien découper le projet.

### **Chapitre 8 : Le multitâche**

- Rappels sur les `Thread` et `Runnable`
- Définition pour un objet `Thread Safe`
- Le parallélisme avancé (`Futur`, `Executor`, `Callable`, ..)
- Le `ForkJoinPool`
- Les objets dits atomiques
- Les Sémaphores (`verrous`) Atelier pratique : Calcul du nombre de décimal de PI en traitement parallèle.

### **Chapitre 9 : La gestion des flux en NIO**

- Les entrées/sorties
- Sérialisation/désérialisation d'objets
- L'API NIO : `Paths`, `Files`, `Buffer`, `Channel`, `WatchService`, `View`

### **Chapitre 10 : L'accès aux données via JPA**

- Rappel JDBC
- Notion de `DataSource` et de `Pool` de connexions
- Le Mapping Objet Relationnel (ORM)
- L'`EntityManager`, l'`EntityManagerFactory`
- Le transactionnel en JPA
- Le JPQL un remplaçant du SQL
- Focus sur la sécurisation
- Eviter les login/pwd dans son code
- Eviter l'injection SQL

Atelier pratique : Chargement en parallèle de plusieurs fichiers textes afin de remplir une base de données de jeux d'essais.

### **Chapitre 11 : Les tests unitaires en Junit**

- Objectif des tests unitaires
- Historique des 3 versions de Junit (v3, 4 5)
- Annotations principales (Junit 5)
  - `@Test`
  - `@BeforeEach` / `@BeforeAll`
  - `@AfterEach` / `@AfterAll`
- Les Assertions et Assumptions
- Les suites de tests
- Les familles / catégories de test (`@Tag`)
- Les tests paramétriques
- Les tests répétés, annotation `@RepeatedTest`
- Les annotations conditionnelles
- `@EnabledOnXxx`

- @DisabledIfxxx
  - Les objets injectables en Junit 5
  - TestInfo, TestReporter, RepetitionInfo
  - Le TDD avec Mockito
  - Définition du covrage et comment l'obtenir
  - Via son IDE
  - Via le plugin jacoco
  - Visualisation dans SonarQube
- Atelier pratique : Mise en place des tests unitaires sur l'application écrite précédemment.  
Utilisation des mock pour la partie d'accès aux données

## MODALITÉS

### Modalités

**Modalités** : en présentiel, distanciel ou mixte – Horaires de 9H à 12H30 et de 14H à 17H30 soit 7H – Intra et Inter entreprise

**Pédagogie** : essentiellement participative et ludique, centrée sur l'expérience, l'immersion et la mise en pratique. Alternance d'apports théoriques et d'outils pratiques.

**Ressources techniques et pédagogiques** : Support de formation au format PDF ou PPT Ordinateur, vidéoprojecteur, Tableau blanc, Visioconférence : Cisco Webex / Teams / Zoom

**Pendant la formation** : mises en situation, autodiagnostic, travail individuel ou en sous-groupe sur des cas réels

### Méthode

**Fin de formation** : entretien individuel

**Satisfaction des participants** : questionnaire de satisfaction réalisé en fin de formation

**Assiduité** : certificat de réalisation (validation des acquis)