

# Java – Programmation avancée

4 jours (28 heures)

Perfectionnement

Présentiel ou classe à distance

à partir de

**2 540 € HT** par personne (TVA en vigueur) Repas inclus en présentiel

Contenu mis à jour le 25/09/2026. Document téléchargé le 02/10/2026.

## Objectifs

A l'issue de cette formation, vous serez capable de :

- Distinguer les aspects avancés du langage Java (Java 21 LTS)
- Appliquer les principaux frameworks et librairies Java
- Reconnaître le modèle de sécurité et le chargement des classes Java
- Réaliser des traitements parallèles (Fork / Join, Virtual Threads)
- Gérer les accès concurrentiels (avec les sémaphores)
- Mettre en place du code Java implémentant la notion de "promesse"
- Tester, debugger et optimiser vos applications (JUnit 5)
- Identifier les nouveautés Java.

Les compétences visées par cette formation font l'objet d'une évaluation.

## À qui s'adresse cette formation ?

Développeurs, architectes et chefs de projets techniques.

### Prérequis

Avoir une connaissance pratique du langage Java ou avoir suivi le cours JAV-SE "Java – Les fondamentaux de la programmation".

# Programme

## 1 Pendant votre formation

### Java – Programmation avancée

Jour 1 – Matin

#### Aspects avancés de Java

- La programmation concurrente
- Les concepts de la programmation multi-thread :
  - Le modèle d'activités de Java (Runnable et Thread)
  - Virtual Threads (Java 21 – Project Loom)
- La création / destruction des threads
- Ordonnancement des threads
- La synchronisation des threads
- Le verrouillage des méthodes et des instructions (synchronized)
- Les moniteurs
- Quelques problèmes du multi-thread :
  - L'interblocage (caractérisation, évitement, prévention, détection), la famine
- Les extensions du modèle introduites dans Java 5 (Callable<T>, Future<T>, ExecutorService), les nouvelles collections
- Scoped Values (Java 21+) – alternative moderne à ThreadLocal

#### Exemple de travaux pratiques (à titre indicatif)

- Construction d'une application multi-thread avec contraintes de concurrence

Jour 1 – Après-midi

#### Aspects avancés : la programmation réflexive

- Le modèle Fork / Join
- Structured Concurrency (Java 21+)
- Les extensions avec CompletableFuture
- Les divers outils de gestion de la concurrence :
  - Les verrous partagés / exclusifs
  - Les sémaphores
  - Les barrières cycliques
- Les objectifs et les principes
- La découverte dynamique des informations relatives à une classe ou à un objet
- L'instanciation et l'invocation dynamique

*Exemples de travaux pratiques (à titre indicatif)*

- Mise en pratique sur :
  - La programmation parallèle
  - Le `CompletableFuture`
  - Les sémaphores

## Jour 2 – Matin

**Implémentation d'un `ClassLoader` chargeant des classes**

- Rôle du compilateur Java
- Rôle des `ClassLoaders`
- Les différentes zones mémoires de la JVM (Java Virtual Machine) et leur gestion par le GC (Garbage Collector)
- Garbage Collectors modernes : ZGC, Shenandoah
- Hiérarchie des différents `ClassLoaders`
- Vérification du Bytecode
- Chargement dynamique de classe
- Implémenter un `ClassLoader`
- Manipulation des énumérations

*Exemple de travaux pratiques (à titre indicatif)*

- Quelques exemples de chargement de classes et traitements des annotations personnalisées

## Jour 2 – Après-midi

**Les nouveautés du `Switch` et `Record`**

- `Switch` expressions
- Pattern Matching `instanceof`
- `Record` (classe de données simplifiée pour DTO)

*Exemple de travaux pratiques (à titre indicatif)*

- Mise en oeuvre des instructions `Switch`

## Jour 3 – Matin

**La console `JShell` Java**

- Principe REPL (Read-Eval-Print Loop)
- Présentation de `JShell` Java
- Fonctionnement interne de `JShell`
- Utilisation de `JShell` (exemples)

**Process API, HTTP2 API**

- L'interface `ProcessHandle`
- Gestion des processus depuis Java
- Présentation de HTTP Client (API stable)

*Exemples de travaux pratiques (à titre indicatif)*

- Programmation d'un cas d'école avec la console JShell
- Mise en pratique d'un client HttpClient avec Java 9

Jour 3 – Après-midi

**Tester ses applications en Java**

- Test Driven Development – Découvrir les grands principes du TDD
- Cycle de développement du TDD
- Les 3A : Arrange, Act, Assert
- Remaniement de code : techniques de refactoring en TDD
- Conception émergente
- Gestion des exceptions
- Gestion des scénarios

**JUnit 5 (Jupiter 5.10+)**

- Savoir utiliser le framework JUnit 5 pour implémenter des tests
- Tests unitaires

*Exemple de travaux pratiques (à titre indicatif)*

- Utilisation de JUnit 5

Jour 4 – Matin

**D'autres nouveautés Java**

- Lambda fonctions
- Streams et Streams parallèles
- La gestion des modules

*Exemple de travaux pratiques (à titre indicatif)*

- Développement d'une application utilisant les Lambdas, Streams et modules

Jour 4 – Après-midi

**Le processus d'optimisation**

- Analyser
- Coder
- Mesurer pour vérifier

**Les outils**

- Profiling
- JDK Flight Recorder (JFR)
- JDK Mission Control
- JVisualVM (versions récentes)

- Eclipse MAT
- SoapUI
- JMeter

*Exemple de travaux pratiques (à titre indicatif)*

- Utilisation d'outils de profiling

### Gestion mémoire et optimisation

- L'organisation de la mémoire
- Le ramasse-miettes
- Garbage Collectors modernes : ZGC, Shenandoah
- Le cycle de vie des objets
- Configurer la mémoire et le GC
- Mesure de la mémoire
- Off Heap
- La gestion de la mémoire et les performances
- Les options de la JVM HotSpot pour le suivi de l'activité du GC
- Fuite de mémoire
- Fuite de ressources
- Optimiser l'utilisation de la mémoire
- Optimiser l'occupation mémoire

*Exemple de travaux pratiques (à titre indicatif)*

- Utilisation des outils d'optimisation mémoire

## Financement

Plusieurs dispositifs de financement sont accessibles via les OPCO (Opérateurs de Compétences), organismes agréés par le ministère du Travail dont le rôle est d'accompagner, collecter et gérer les contributions des entreprises au titre du financement de la formation professionnelle.

Pour plus d'information, une équipe de gestionnaires SKOLAE Formation spécialisée vous accompagne dans le choix de vos formations et la gestion administrative.

[Nous contacter pour en savoir plus →](#)

## Modalités pédagogiques

Formation délivrée en présentiel ou distanciel\* (blended-learning, e-learning, classe virtuelle, présentiel à distance).

Le formateur alterne entre méthode\*\* démonstrative, interrogative et active (via des travaux pratiques et/ou des mises en situation).

Variables suivant les formations, les moyens pédagogiques mis en oeuvre sont :

- Ordinateurs Mac ou PC (sauf pour certains cours de l'offre Management), connexion internet fibre, tableau blanc ou paperboard, vidéoprojecteur ou écran tactile interactif (pour le distanciel)
- Environnements de formation installés sur les postes de travail ou en ligne
- Supports de cours et exercices

En cas de formation intra sur site externe à M2I, le client s'assure et s'engage également à avoir toutes les ressources matérielles pédagogiques nécessaires (équipements informatiques...) au bon déroulement de l'action de formation visée conformément aux prérequis indiqués dans le programme de formation communiqué.

\* nous consulter pour la faisabilité en distanciel

\*\* ratio variable selon le cours suivi

### Accessibilité aux personnes en situation de handicap

## Détails de la session

Du 27 octobre 2026 au 30 octobre 2026 (4 jours / 28 heures)

**Bourges**

Du 27 octobre 2026 au 30 octobre 2026 **Bourges**

- 27 octobre 2026 ..... 9h00 – 15h30
- 28 octobre 2026 ..... 9h00 – 15h30
- 29 octobre 2026 ..... 9h00 – 15h30
- 30 octobre 2026 ..... 9h00 – 15h30