

# Java – Nouveautés des versions 8 à 25

3 jours (21 heures)

Initiation

Présentiel ou classe à distance

à partir de

**1905 € 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 :

- Exploiter les principaux apports des versions de la Java Development Kit (JDK) 8 à 22
- Exploiter des lambda expressions en passant par les streams et l'API DateTime
- Gérer des modules en passant par les étapes d'une migration réussie
- Reconnaître les améliorations apportées au langage au fil des versions
- Utiliser les bases du langage Java avec une meilleure productivité.

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

# Programme

## 1 Pendant votre formation

### Java – Nouveautés des versions 8 à 25

Jour 1

#### Rappels des apports en programmation concurrente (JDK 7 et plus)

- Le package `java.util.concurrent`
- Les facilités apportées pour le multi-threading et la programmation parallèle
- Le Fork and Join

*Exemples de travaux pratiques (à titre indicatif)*

- Mise en oeuvre d'un programme utilisant le pattern Fork and Join

#### Les lambda en Java (JDK 8 et plus)

- S'approcher de lambda sans Java 8
- Les lambda expressions
- Les interfaces fonctionnelles
- Liste des interfaces fonctionnelles
- Paramètres d'une lambda expression
- Cas d'utilisation des lambda expressions
- Les références de méthode dans les lambda expressions

*Exemples de travaux pratiques (à titre indicatif)*

- Création d'un nouveau projet
- Utilisation des interfaces fonctionnelles fournies et créées
- Utilisation des lambda expressions sous différentes formes
- Simplification de la programmation avec les lambda

#### Les streams (JDK 8 et plus)

- Présentation générale des streams
- Créer des streams
- Opérations intermédiaires sur les streams
- Opérations terminales

*Exemples de travaux pratiques (à titre indicatif)*

- Création d'un nouveau projet
- Application de la gestion des streams à des flux de données fournis
- Opérations de création, intermédiaires et finales

#### L'API `DateTime` (JDK 8 et plus)

- Classes importantes de DateTime
- Utilisation de DateTime

*Exemples de travaux pratiques (à titre indicatif)*

- *Exploitation de l'API DateTime*

Jour 2

### Le système de modules (JDK 9 et plus)

- Architecture et modularité dans le JDK 9
- La déclaration des modules :
  - Requires
  - Exports
  - Uses
  - Provides
  - With
  - Opens
- Les mots réservés
- Exemples des classes d'applications modularisées
- Le fichier module-info.java
- Le graphe de dépendances
- Lancer une application à partir des modules
- Le packaging des modules et les JAR
- Exporter un package pour l'utiliser ailleurs
- Encapsulation forte et accessibilité

*Exemples de travaux pratiques (à titre indicatif)*

- *Création d'une application utilisant les modules*
- *Migration d'une application non modulaire vers le système de module*
- *Compatibilité du système modulaire avec l'existant*

### JShell (JDK 9 et plus)

- Introduction à JShell
- Exécuter et tester du code à la volée

*Exemples de travaux pratiques (à titre indicatif)*

- *Utilisation de l'outil JShell*

### L'API Optional (JDK 9 et plus)

- Les méthodes :
  - or()
  - ifPresent()
  - ifPresentOrElse()

- `stream()`

*Exemples de travaux pratiques (à titre indicatif)*

- Exploitation de l'API Optional afin de gérer les références nulles

### L'API Process (JDK 9 et plus)

- Gestion des process lancés
- L'interface `ProcessHandle`

*Exemples de travaux pratiques (à titre indicatif)*

- Création d'un outil d'administration des process lancés

### Autres apports de la JDK 9

- Améliorations des streams Java 8
- L'API HTTP 2
- Nouvelles collections
- Les Reactive Streams, les streams asynchrones
- Méthodes privées d'interface
- L'API `CompletableFuture`

*Exemples de travaux pratiques (à titre indicatif)*

- Utilisation des Reactive Streams

Jour 3

### Apports de la JDK 10

- Inférence de type de variables locales
- Améliorations dans les API existantes

### Apports de la JDK 11

- Inférences de type pour les lambda expressions
- Simplification des "run" de programmes
- Améliorations dans les classes `String`, `StringBuilder` et `StringBuffer`
- Améliorations des Predicate Java 8
- Classes imbriquées et visibilité des attributs
- Suppression des modules JEE, JavaFX et CORBA

*Exemples de travaux pratiques (à titre indicatif)*

- Test des fonctionnalités apportées par la JDK 11

### Apports de la JDK 12 et 13

- Le switch comme instruction
- Les blocs de texte
- Le mot-clé "yield"

- Les outils apportés par la JDK 12

*Exemples de travaux pratiques (à titre indicatif)*

- Test des fonctionnalités apportées par les JDK 12 et 13

### Apports de la JDK 14

- Changements dans les switch
- Clarifications du NullPointerException
- Le Live Monitoring
- Changements dans le "instanceof"
- Nouveaux outils associés à la JDK 14

### Apports de la JDK 15

- Les classes "Sealed"
- Fonctionnalités dépréciées

### Apports de la JDK 16

- L'outil "jpackage"

### Apports de la JDK 17

- Interfaces "Sealed"
- Les Records

*Exemples de travaux pratiques (à titre indicatif)*

- Test des fonctionnalités apportées par les JDK 14 à 17

### Apports de la JDK 18

- Le Simple Web Server

### Apports de la JDK 19, 20 et 21

- Les projets Amber, Panama, Loom, Valhalla, Leyden
- Les threads virtuels pour l'optimisation des ressources
- L'interface Thread.Builder et la méthode VirtualThreadStart
- La Structured Concurrency
- Amélioration du switch
- Les Record Patterns et la déconstruction
- Les Foreign Functions et Memory API
- Accès aux données Off-Heap
- L'API Vector
- Modifications dans HashMap et HashSet

### Apports de la JDK 22

- Les unnamed variables et patterns

- Les classes déclarées implicitement
- Modifications dans l'appel de l'instruction "super"
- Le lancement de programmes multi source sans compilation

### Apports de la JDK 23

- Pattern Matching for switch
- Markdown Javadoc

### Apports de la JDK 24

- Stream Gatherers, Class-File API (final)
- Cryptographie post-quantique – ML-KEM, ML-DSA

### Apports de la JDK 25

- Compact Source Files et Instance Main Methods (JEP 512)
- Module Import Declarations
- Flexible Constructor Bodies
- Scoped Values
- PEM Encodings of Cryptographic Objects
- JFR Cooperative Sampling

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

- Test des fonctionnalités apportées depuis les JDK 18 à 25

## 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 25 novembre 2026 au 27 novembre 2026 (3 jours / 21 heures)

Chartres

Du 25 novembre 2026 au 27 novembre 2026 Chartres

- 25 novembre 2026 ..... 9h00 – 16h00
- 26 novembre 2026 ..... 9h00 – 16h00
- 27 novembre 2026 ..... 9h00 – 16h00