

Mis à jour le 21/11/2022

S'inscrire

## Formation Java 19 mise à niveau

### Présentation

**Java** 19 est la dernière évolution technologique du langage et de la plateforme d'Oracle Java SE (Standard Edition). Sortie en septembre 2022, cette version apporte de nombreux ajouts comme les threads virtuels et le filtrage par motifs.

Dans cette formation, vous verrez toutes les nouveautés apportées par les versions majeures de Java 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 & 19 sortie en 2022, ainsi que les bonnes pratiques de développement pour démarrer un projet sereinement !

Concernant Java 8 il sera question de passer en revue l'ajout des Lambdas, la refonte de l'API, les Dates, les Collections, les Streams, ainsi que la programmation fonctionnelle.

Coté Java 9, nous verrons entre autres comment modulariser vos projets avec Jigsaw, Kulla / JShell un puissant Shell dédié au test read-eval-print-loop (REPL), l'amélioration des types avec le projet Valhalla, ainsi que le support natif du format JSON et de HTTP/2.

Coté Java 12, nous aborderons les dernières fonctionnalités et améliorations de l'API, l'utilisation du formatage compact des nombres et l'exploration de nouvelles méthodes de comparaison de fichiers, les fonctions de prévisualisation, évaluer la fonction d'expressions de commutation proposée. Nous passerons en revue les suppressions des API et leurs impacts sur les fonctionnalités. Enfin, niveau Performance et sécurité : Découvrez pourquoi les améliorations apportées au Garbage Collector G1 comptent parmi les arguments de vente les plus importants de Java 8.

Pour Java 13, nous aborderons les blocs de texte et les nouveautés concernant l'expression "switch". En ce qui concerne Java 14, l'accent sera mis sur les records, le pattern matching et les différentes améliorations apportées à la version 13.

Enfin, vous apprendrez les nouveautés de Java 15 et 16 : les sealed class, l'EdSDA, le Legacy DatagramSocket API CharSequence.isEmpty() ou encore migrate to Git.

Comme dans toutes nos formations, celle-ci vous présentera la toute dernière version de Java (à la date de rédaction de l'article : [JDK 19](#)).

# Objectifs

- Maîtriser les nouveautés de Java 8 à 19 : Lambdas, Stream, Date & Time, Teeing Collectors
- Maîtriser la syntaxe de Java
- Savoir tester l'environnement de Java
- Programmer en Java via les API
- mettre en œuvre les différentes librairies de Java
- Migrer une application de Java 8 à Java 19
- Utiliser JShell / REPL pour tester votre code

# Public visé

- Développeurs
- Architectes techniques
- Chargés de développement d'application informatique
- Chefs de projets

# Pré-requis

- Connaissances en programmation objet
- Posséder une expérience en langage java

# Pour aller plus loin

## Programme de Notre Formation Java 19

### Introduction

- Convergence : Oracle JDK & OpenJDK
- Licensing : Oracle JDK & OpenJDK
- Release / Nouvelle cadence de 6 mois : Qu'est ce que cela signifie ?
- Long-Term Support

### Les fondamentaux

- Plus de 200 nouveautés (nouvelles entités, classes, interfaces...)
- Les expressions lambda
- Les implémentations par défaut des interfaces et @FunctionalInterface
- Les nouvelles annotations
- Les évolutions des streams
- Nouvelle écriture SQL dans Java 19
- La nouvelle API de gestion des dates et des heures
- Les évolutions d'API de java.lang, java.util, java.net...
- Exécuter du code JavaScript avec Java 18 Nashorn

### Nouveautés des versions 11 à 19 de Java

- Communication HTTP intégrée et améliorée avec HttpClientLaunching

- Programme Single-File sans compilation (Shebang)
- Nouvelles méthodes de bibliothèque pour les chaînes de caractères, les collections et les fichiers
- Améliorations apportées Optional et au prédicat (Predicate)
- Fonctions et API supprimées
  - Suppression des modules Java EE
  - Applets et Java Web Start
  - JavaFX devient OpenJFX
  - Deprecated : Nashorn JavaScript Engine
  - VM Options and Other Tools

## Nouvelles fonctionnalités Java 18

- Collecteurs : Teeing Collector
- Améliorations de l'API de chaînes de caractères
- Comparaison des fichiers
- Formatage compact des nombres
- Adoption de l'Unicode 11

## Migrer son code vers Java 19

- Migration sans le système de Modules
- Migration vers les Modules
- Migrer vos Librairies - tooling avec jdeps
- Compiler et lancer une application modularisée

## Sécurité et Performance

- Garbage Collection
- Security / TLS 1.3 : Support Cryptographic Algorithms

## Lambda expression

- Classe anonyme, concept d'expression lambda
- Nouvelle syntaxe (déclaration, implémentation, passage de paramètres, portée des variables...)
- Le concept de « foncteur » à travers les interfaces « fonctionnelles » et le package java.util.function
- Manipuler les collections grâce aux lambda expressions
- Code SQL: Les Listeners

## Système de modules introduit en Java 9

- Notion de module
- Utilisation des modules
- Modularité dans le JDK 9
- Déploiement d'applications Java
- Rôle du linker Java : jlink
- Le fichier module-info.java

- Le format des fichiers jar avec JMOD

## Amélioration de l'API dédiée aux collections

- Les méthodes Factory
- Les collections non mutables
- Optimisation d'accès

## Améliorations des API Stream

- Rappels rapides de l'API Stream
- TakeWhile et dropWhile
- Les méthodes iterate et ofNullable
- Nouveau : java.util.Stream, java.util.ParallelStream
- Différence entre collection et Stream
- Source, fonction intermédiaire, fonction terminale
- Fonctions intermédiaires : filter(), sort(), distinct()
- Fonctions terminales : collect(), count, foreach()

## Les Streams Parallel

- Exploiter tous les processeurs & cores de votre machine
- Le framework Fork / Join
- ParallelStream
- Personnalisation : découpage avec Spliterator
- Comparaison de performance & Benchmark systématique

## L'API Optional

- Rappels rapides de l'API Optional
- Méthodes :
  - or()
  - ifPresentOrElse()
  - stream()

## L'API Process

- Les interfaces ProcessHandle et ProcessHandle.info

## L'API HTTP 2

- Le protocole HTTP 2
- Le mode asynchrone
- Intégration des WebSockets

## La programmation réactive

- Présentation de la programmation réactive
- Utilisation de Computable Future
- Les Reactive Streams
- Les Flow API
- Présentation de RxJava

## Fonctionnalités de prévisualisation

- La façon traditionnelle de définir les instructions de commutation
- Expressions de commutation
- Activation des fonctions d'aperçu à partir des outils de création
- Activation des fonctions d'aperçu dans l'EDI

## Fonctions et API supprimées

- Support du compilateur pour Java 18
- Suppression des méthodes de finalisation
- Fonctions et API obsolètes

## Benchmark

- Microbenchmark Suite pour le code source du JDK
- Shenandoah : Un collecteur d'ordures à faible temps de pause
- Améliorations du Garbage Collector G1
- Améliorations de la sécurité
- Sommaire et lectures complémentaires

## Concurency : Multithread

- L'API Concurrent : les Executor et ses différents types
- Interfaces Runnable, Future, Callable
- Expressions Lambda et Multithread
- Nouveautés Java 18 : ComposableFuture
- Async, Timeout, Utilities
- TimeUnit a ChronoUnit, Queue, Atomic Classes
- Orchestration de vos tâches en asynchrones
- Pools de threads
- Mise en œuvre dans une couche de Service

## Java 19 Nashorn (module optionnel sur demande)

- Le moteur ECMAScript de Java 19
- Ligne de commande jjs

- Appeler du code JavaScript
- Appeler du code Java depuis du JavaScript

## De Java 13 à Java 19 et Exécution de scripts avec JShell (module optionnel sur demande)

- Introduction à JShell
- Exécuter et tester du code avec JShell
- Les blocs de texte
- L'expression "switch"
- Les records
- Le pattern matching
- Les sealed class
- L'EdSDA
- Le Legacy DatagramSocket API
- La fonction CharSequence.isEmpty()
- Migrate to Git

## Sociétés concernées

Cette formation s'adresse aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée.

## Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

## Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

## Validation

À la fin de la session, un questionnaire à choix multiple permet de vérifier l'acquisition correcte des compétences.

## Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.