

JAVA - Programmation avancée

Cette formation Java Avancé de 5 jours s'adresse aux développeurs et ingénieurs souhaitant approfondir leur expertise du langage et adopter les meilleures pratiques professionnelles. Elle couvre les fonctionnalités avancées de Java, la programmation fonctionnelle, la concurrence, la performance et le fonctionnement interne de la JVM. À travers de nombreux ateliers pratiques et un projet fil rouge, les participants renforcent leur capacité à concevoir des applications Java robustes, performantes et maintenables, adaptées aux exigences des environnements professionnels modernes.

Présentiel - Synchrone



Public Visé

Développeurs Java, Ingénieurs logiciels, Développeurs back-end, Chefs de projet techniques avec background Java, Architectes logiciels juniors.



Pré Requis

Maîtrise de Java SE, POO, expérience pratique en développement Java et notions sur exceptions, collections et threads.

Objectifs pédagogiques et d'évaluation

Maîtriser les fonctionnalités avancées du langage Java
Concevoir des applications robustes, maintenables et performantes
Exploiter efficacement le multithreading et la programmation concurrente
Appliquer les bonnes pratiques de conception et de performance
Comprendre le fonctionnement interne de la JVM

Méthodes pédagogiques

Beaucoup de tests et d'exemples concrets en lien direct avec les besoins des participants
Des techniques directement mobilisables dans le cadre de leurs fonctions
Formation axée sur la mise en pratique
Méthode participative
Mise en situation professionnelle
Alternance de cours et d'exercices dirigés ou en autonomie
Travaux pratiques
Evaluation croisée et partage de bonnes pratiques

Parcours pédagogique

Approfondissement du langage Java
Rappels et bonnes pratiques Java
Conventions de codage et clean code
Immutabilité et encapsulation
Gestion avancée des exceptions
Généricité avancée
Wildcards (extends, super)
Bornes multiples
Type erasure
Cas d'usage avancés
Java Reflection et annotations
API Reflection
Création et utilisation d'annotations personnalisées
Cas pratiques (frameworks, injection, validation)
Atelier pratique

Collections, Streams et programmation fonctionnelle
Collections avancées
Choisir la bonne collection
Performances et complexité
Collections immuables
ConcurrentHashMap et collections thread-safe
API Stream
Streams séquentiels vs parallèles
Opérations intermédiaires et terminales
Collectors avancés
Pièges courants et bonnes pratiques
Programmation fonctionnelle
Lambdas avancées
Interfaces fonctionnelles personnalisées
Optional : bonnes pratiques d'utilisation
Atelier pratique

Concurrence, multithreading et parallélisme**Modèle de concurrence Java**

Thread lifecycle

Problèmes classiques (race conditions, deadlocks)

synchronized et verrous explicites

API Concurrency

ExecutorService, Callable, Future

Thread pools

Fork/Join framework

Outils avancés de synchronisation

Lock, ReadWriteLock

Atomic classes

CompletableFuture

Atelier pratique

JVM, performance et gestion mémoire**Fonctionnement interne de la JVM**

ClassLoader

Stack vs Heap

Garbage Collectors (G1, ZGC, Shenandoah – concepts)

Optimisation des performances

Profiling (JVisualVM, JProfiler – concepts)

Analyse de la consommation mémoire

Optimisation du temps de réponse

Gestion des erreurs et robustesse

OutOfMemoryError

Gestion avancée des exceptions runtime

Logs et monitoring

Atelier pratique

Architecture, design patterns et bonnes pratiques professionnelles**Design Patterns en Java**

Patterns de création, structurels et comportementaux

Anti-patterns courants

Bonne pratiques d'implémentation

Architecture et modularité

SOLID et Clean Architecture

Java Modules (JPMS)

Packaging et structuration de projets

Tests et qualité logicielle

Tests unitaires avec JUnit

Tests de performance

Introduction aux tests d'intégration

Synthèse et mise en pratique

Revue de code collective

Bonnes pratiques industrielles

Évolutions de Java (LTS, nouveautés majeures)

Projet fil rouge : conception et amélioration progressive d'une application Java avancée

Moyens pédagogiques

1 ordinateur par stagiaire - Salle de formation claire, climatisée et spacieuse - Tableau blanc - Vidéo projecteur - Support de cours - Logiciel d'assistance des stagiaires à distance

Qualification Intervenant-e-s

Formateur.trice - Consultant.e spécialiste Java

Méthodes et modalités d'évaluation

Evaluation diagnostique en amont : questionnaire d'auto-positionnement et recueil des attentes & besoins - Evaluation formative en cours de formation : exercices - Evaluation sommative en fin de formation : exercice/QCM - Bilan individuel des compétences acquises – Attestation de fin de formation - Questionnaire de satisfaction à chaud - Questionnaire de satisfaction à froid envoyé 15 jours après la formation pour assurer le suivi post formation

Modalités d'accessibilité handicap

Nos formations sont accessibles aux personnes en situation de handicap et aux besoins spécifiques. Une étude personnalisée avec notre référente handicap, permettra d'adapter les moyens pédagogiques, techniques ou organisationnels.

 Durée	 Effectif
35.00 Heures	5 Jours

De 3 à 8 Personnes