



Développeur Java

Java - Programmation avancée

4 jours (28h00) | ★★★★★ 4/5 | JAV-AV | Évaluation qualitative de fin de stage |
Formation délivrée en présentiel ou distanciel

Formations Informatique > Langages et développement > Développeur Java

Contenu mis à jour le 13/10/2023. Document téléchargé le 23/05/2024.

Objectifs de formation

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

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

Modalités, méthodes et moyens 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

Prérequis

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

Public concerné

Développeurs, architectes et chefs de projets techniques.

Cette formation :

- Est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- Bénéficie d'un suivi de son exécution par une feuille de présence émarginée par demi-journée par les stagiaires et le formateur.

Programme

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)
- La création / destruction des threads
- Ordonnement 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, Future, ExecutorService), les nouvelles collections

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
- 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)
- Hiérarchie des différents `ClassLoaders`
- Vérification du Bytecode
- Chargement dynamique de classe
- Implémenter un `ClassLoader`
- Manipulation des énumérations

Exemples 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 9
- Présentation de `HTTP2 Client`

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

Exemple de travaux pratiques (à titre indicatif)

- Utilisation de JUnit

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
- JVisualVM
- 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
- 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

Le contenu de ce programme peut faire l'objet d'adaptation selon les niveaux, prérequis et besoins des apprenants.

Modalités d'évaluation des acquis

- En cours de formation, par des études de cas ou des travaux pratiques

- Et, en fin de formation, par un questionnaire d'auto-évaluation

Accessibilité de la formation

Le groupe M2i s'engage pour faciliter l'accessibilité de ses formations. Les détails de l'accueil des personnes en situation de handicap sont consultables sur la page Accueil et Handicap.

Modalités et délais d'accès à la formation

Les formations M2i sont disponibles selon les modalités proposées sur la page programme. Les inscriptions sont possibles jusqu'à 48 heures ouvrées avant le début de la formation. Dans le cas d'une formation financée par le CPF, ce délai est porté à 11 jours ouvrés.