



Développeur Java

Java - Nouveautés des versions 8 à 17

3 jours (21h00) | ★★★★★ 4,6/5 | JAV-NEW | 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 :

- Exploiter les principaux apports des versions de la Java Development Kit (JDK) 8 à 16
- 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.

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 suivi les cours JAV-SE "Java - Les fondamentaux de la programmation" et JAV-DVO "Java - Pour les développeurs objet" ou avoir une connaissance pratique du langage Java.

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 émargée par demi-journée par les stagiaires et le formateur.

Programme

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

Exemple 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

Exemple 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

Exemple de travaux pratiques (à titre indicatif)

- Utilisation de l'outil JShell

L'API Optional (JDK 9 et plus)

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

Exemple 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

Exemple de travaux pratiques (à titre indicatif)

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

Jour 3

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

Exemple de travaux pratiques (à titre indicatif)

- *Utilisation des Reactive Streams*

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, JavaJX et CORBA

Exemple 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

Exemple 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

- Apports de la LTS 17

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.