Développez vos talents
Certifiez vos compétences

Développeur Java

Java - Développement avancé

4 jours (28h00) | ★★★★☆ 4/5 | JAV-AV | Certification M2i Langages de Programmation (non incluse) | Évaluation qualitative de fin de stage | Formation délivrée en présentiel ou distanciel [1]

Formations Informatique › Langages et développement › Développeur Java

À l'issue de ce stage vous serez capable de :

- Reconnaître les aspects avancés du Java
- Comparer le multi-threading et la programmation concurrente
- Gérer l'asynchronisme
- Créer vos propres annotations
- Utiliser Java Reflection API
- Exploiter les Web Services et les Web Sockets
- Reconnaître JMS et JMX
- Utiliser les lambda expressions et les streams.

Niveau requis

Avoir une connaissance pratique du langage Java ou avoir suivi le cours JAV-SE "Java - Les fondamentaux et le développement Java SE".

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.

(1) Modalité et moyens pédagogique :

Formation délivrée en présentiel ou distanciel * (e-learning, classe virtuelle, présentiel à distance). Le formateur alterne entre méthodes ** démonstrative, interrogative et active (via des travaux pratiques et/ou des mises en situation). La validation des acquis peut se faire via des études de cas, des quiz et/ou une certification.

Les moyens pédagogiques mis en oeuvre (variables suivant les formations) sont : ordinateurs Mac ou PC (sauf pour les 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.

* Nous consulter pour la faisabilité en distanciel. ** Ratio variable selon le cours suivi.

Programme

Jour 1

Quelques aspects évolués de Java

- Classes anonymes
- Nouvelles boucles for
- Les annotations intéressantes
- Les varargs
- Les tableaux
- Le try-with-resource
- NIO 2 et Non-blocking I/O

Exemple de travaux pratiques (à titre indicatif)

- *Mise en oeuvre des divers apports de Java avec try-with-resource, NIO, boucles for*

Threading et concurrence

- Thread et Runnable
- Problèmes du multi-threading
- Attente et synchronisation
- Thread pools et environnements "thread safe"
- Package java.util.concurrent
- Le Framework Executor
- Asynchronisme en Java
- Future et Callable
- Le Fork and Join

Exemples de travaux pratiques (à titre indicatif)

- *Création d'un projet mettant en oeuvre le multi-threading*
- *La synchronisation entre threads*
- *La protection de données entre threads*
- *La programmation concurrente*
- *Le Fork and Join*

Jour 2

Echange de données avec les Web Services en Java

- Clients et serveurs
  - SOAP avec JAX-WS et CXF
  - REST avec JAX-RS et Jersey

Exemples de travaux pratiques (à titre indicatif)

- *Reprise du projet*
- *Ajout de Web Services REST entre composants serveur et client*

Echange de données avec les Web Sockets en Java

- Mise en oeuvre de Web Sockets en Java
  - Côté serveur
  - Côté client

Exemples de travaux pratiques (à titre indicatif)

- *Reprise du projet*
- *Ajout de Web Sockets entre composants*

- *Affichage de messages non sollicités par le client*

Echange de données avec RMI

- Echange de données entre composants Java

Exemples de travaux pratiques (à titre indicatif)

- *Reprise du projet*
- *Ajout d'échange de composants Java via RMI*

Jour 3

Programmation fonctionnelle en Java

- La programmation fonctionnelle
- Les implications de la programmation fonctionnelle
- Avantages et inconvénients de la programmation fonctionnelle

Les lambda en Java

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

Exemples de travaux pratiques (à titre indicatif)

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

Jour 4

Les streams

- 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*

Annotations et "reflection"

- Annotations prédéfinies
- Le processeur d'annotations
- Définition d'interface (@Interface)
- Traitement à la compilation ou à l'exécution (@Retention)
- Cible des annotations (@Target)
- Cas d'utilisation des annotations
- L'API Reflection
- Chargement et appel dynamique

Exemples de travaux pratiques (à titre indicatif)

- *Création d'un nouveau projet*
- *Création d'une nouvelle annotation et exploitation de celle-ci au runtime*

JMS (Java Message Service)

- La communication asynchrone par messages
- Les interfaces et les classes de JMS
- Les providers de message
- La communication par Queue et par Topic

JMX (Java Management Extensions)

- Avantages d'administration des composants
- Les concepts de JMX
  - MBean
  - MBeanServer
  - Agent
  - Connector
- La console d'administration

Exemples de travaux pratiques (à titre indicatif)

- *Création d'un projet JMS permettant l'échange de données entre composants, avec Queue et Topic*
- *Démonstrations pratiques de l'administration de composants avec JMX*

JNI (Java Native Interface) et JNA (Java Native Access)

- Appel natifs depuis Java

Le scripting en Java

- Exécution d'un script depuis Java

Exemples de travaux pratiques (à titre indicatif)

- *Appel d'un point d'entrée de DLL depuis Java avec JNA*
- *Exécution de plusieurs scripts Javascript depuis Java*

Certification (en option)

- Prévoir l'achat de la certification en supplément
- L'examen (en français) sera passé le dernier jour, à l'issue de la formation et s'effectuera en ligne
- Il s'agit d'un QCM dont la durée moyenne est d'1h30 et dont le score obtenu attestera d'un niveau de compétence
- La certification n'est plus éligible au CPF depuis le 31/12/2021, mais permettra néanmoins de valider vos acquis

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 ou une certification (M2i ou éditeur)