

Java, programmation avancée

4 j (28 heures)

Ref : JAVX

Public

Développeurs, chargés de développement d'applications informatiques, chefs de projets proches du développement...

Pré-requis

Niveau : Maitriser le langage Java, connaître les concepts de bases de données relationnelles et du langage SQL, disposer d'une expérience en programmation Java.

Techniques (formations en classe virtuelle) : Vous devez disposer d'un ordinateur connecté à internet, d'un micro et d'une caméra

Moyens pédagogiques

Modalité : Formation présentielle ou Formation distancielle (classe virtuelle) - Inter / Intra - Groupes de 4 à 12 stagiaires

Méthodes : Présentation des concepts, discussion technique, démonstrations, exercices et TP

Matériel :

Présentiel : Un poste informatique par stagiaire connecté à internet, à une imprimante en réseau et au réseau informatique,

Les salles sont équipées d'un tableau interactif ou d'un vidéoprojecteur et d'un paperboard

Distanciel : Aelion met à disposition de chaque stagiaire

- Un PC équipé des outils et logiciels nécessaires à la formation auquel vous accédez via un outil de prise en main à distance

- Un accès à un outil de classe virtuelle (Meet)

Support de formation : Un support de formation sera remis à chaque stagiaire en fin de formation : plateforme collaborative intégrant le code source des exercices réalisés en formation, webographie, mémos

Modalités de suivi et d'évaluation

Questionnaire d'évaluation des pré-requis, suivi des connaissances tout au long de la formation, Evaluation des acquis en fin de formation

Questionnaire d'évaluation de la satisfaction en fin de stage, feuille de présence émargée par demi-journée par les stagiaires et le formateur, Attestation de fin de formation

Avec un taux d'intégration très important dans de nombreux secteurs industriels, Java est devenu un langage majeur dans le développement d'application industrielle, embarquée ou de gestion. La plateforme J2SE met à disposition un socle technique riche et complet permettant aux développeurs de concevoir des applications complexes dans un cadre structuré et industrialisé. Au-delà du langage, acquérir une bonne compréhension de la philosophie de Java permet d'améliorer la productivité et la qualité du livrable logiciel. A l'issue de la formation, vous serez capable d'utiliser les fonctions avancées et les principales bibliothèques du langage Java dans les projets de développement d'applications.

Formation finançable par votre OPCO

Spécificité OPCO ATLAS : cette formation est 100% financée par ATLAS dans le cadre du campusAtlas pour la branche Bureau d'Etude. Sous réserve de validation de votre dossier par ATLAS.

Objectifs

- Savoir tester, debugger et optimiser ses applications
- Maîtriser les aspects avancés du langage Java
- Comprendre comment appliquer les principaux Frameworks et librairies Java Comprendre le modèle de sécurité et le chargement des classes Java
- Savoir programmer en Java dans le contexte des bases de données relationnelles
- Appréhender les nouveautés Java

Programme détaillé

SAVOIR TESTER, DEBOGUER ET OPTIMISER SES APPLICATIONS (1/2)

- Historique de JAVE SE
- JVM, JRE et JDK
- Édition Oracle ou OpenJDK et licences respectives
- Code source et bytecode
- Packages et modules
- Shell Java
- IDE, tests et débogueur

SAVOIR TESTER, DEBOGUER ET OPTIMISER SES APPLICATIONS (2/2)

- Gestion de la mémoire
- Pile et tas
- Le ramasse miette
- Référence Forte
- Référence Faible
- Référence fantôme
- Monitoring
- Console Java
- Java Mission Control (version Oracle uniquement)

MAITRISER LES ASPECTS AVANCES DU LANGAGE JAVA (1/6)

- Autoboxing des types primitifs
- Les varargs
- Les énumérations complexes
- Les données temporelles depuis Java 8
- Création, formatage, parsing des données temporelles
- Fuseaux horaires et paramètres locaux

MAITRISER LES ASPECTS AVANCES DU LANGAGE JAVA (2/6)

- Interface fonctionnelle

- Méthodes « default » et « static » des interfaces
- Les fonctions anonymes
- Référence de méthode objet
- Référence de méthode sur un objet particulier
- Référence de constructeur
- Référence, héritage et liaison tardive
- Référence et surcharges
- Référence et autoboxing
- Référence et conversions implicites sur les nombres

MAITRISER LES ASPECTS AVANCES DU LANGAGE JAVA (3/6)

- Inférence de type
- Déclaration de variable avec mot clé var
- Effacement de type
- Types génériques avec lower ou upper bound
- Covariance et contravariance
- Méthode générique

MAITRISER LES ASPECTS AVANCES DU LANGAGE JAVA (4/6)

- Collections de Java
- Collections simplifiées pour l'écriture des tests
- Itérateurs
- Instruction for "each" vs méthode forEach
- Boîte à outils java.util.Collections?

MAITRISER LES ASPECTS AVANCES DU LANGAGE JAVA (5/6)

- Les streams d'objets
- Les principes du map/reduce
- Interfaces fonctionnelles dédiées au map/reduce
- Filtrer les données
- Transformation des données
- Collecte des données
- Gestion particulière des types primitifs
- Utilisation des collecteurs prédéfinis
- Écrire un collecteur custom
- Génération de streams
- Combiner des streams

MAITRISER LES ASPECTS AVANCES DU LANGAGE JAVA (6/6)

- Les annotations
- Annotations standards et Méta-Annotations
- Rétention des annotations

Java, programmation avancée

- Cible des annotations
- Les arguments d'une annotation
- Lecture par réflexion d'une annotation
- Traitement avec javac ou javax.annotation.processing
- Manipulation de bytecode avec Javassist

COMPRENDRE LE MODELE DE SECURITE ET LE CHARGEMENT DES CLASSES JAVA

- API d'introspection
- La classe Class
- Les autres classes du méta modèle
- Instanciations et appels dynamique
- Restriction de sécurité
- Exploitation dynamique des annotations

COMPRENDRE COMMENT APPLIQUER LES PRINCIPAUX FRAMEWORKS ET LIBRAIRIES JAVA

- Données optionnelles
- Interface Optional
- Types primitifs optionnels
- Traitement impératif optionnel
- Transformation d'une donnée optionnelle
- Objets optionnels imbriqués

SAVOIR PROGRAMMER EN JAVA DANS LE CONTEXTE DES BASES DE DONNEES RELATIONNELLES

- Spécification JDBC ET DRIVERS JDBC
- ORM, JPA et Hibernate
- Configuration ORM et gestion des sessions
- Mapping d'une classe simple avec une table
- Types temporels, énumérations
- Collections simples
- Mapping d'associations
- CRUD sur les objets
- Requêtes JPQL, HQL ou SQL natif
- Requêtes avec l'API Criteria
