

Java 8, découvrir les nouveautés par la pratique

Cours Pratique de 3 jours - 21h

Réf : JVA - Prix 2024 : 1 870€ HT

Cette formation vous permettra d'appréhender et de pratiquer les nouveautés de la version 8 de Java. Vous découvrirez les expressions lambda et exploiterez les possibilités de la programmation fonctionnelle. Vous apprendrez à faire interagir du code JavaScript avec du code Java et découvrirez de nouvelles API.

OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Utiliser les expressions lambda

Exploiter les possibilités de la programmation fonctionnelle

Utiliser des codes JavaScript depuis Java 8

Connaître la nouvelle API de manipulation des dates

TRAVAUX PRATIQUES

Des exercices et travaux pratiques permettront de mettre en œuvre les concepts abordés.

LE PROGRAMME

dernière mise à jour : 04/2022

1) Synthèse des nouveautés de Java 8

- Vue d'ensemble des ajouts : 195 nouvelles entités (classes, interfaces...).
- Les expressions lambda.
- Ecrire du code SQL dans un code Java 8.
- Les implémentations par défaut des interfaces et @FunctionalInterface, les nouvelles annotations.
- Les évolutions des streams.
- La nouvelle API de gestion des dates et des heures.
- Les évolutions d'API de java.lang, java.util, java.net...
- La possibilité d'exécuter du code JavaScript avec Java 8 Nashorn.

Travaux pratiques : Vérification de l'installation du JDK 8 (outils, javadoc...).

2) Les expressions lambda

- Notion de classe anonyme imbriquée, améliorations apportées par le concept d'expression lambda.
- Les aspects syntaxiques (déclaration, implémentation, passage de paramètres, portée des variables...).
- Le concept de "foncteur" à travers les interfaces "fonctionnelles" et le package java.util.function.
- Utiliser les lambda-expressions pour manipuler les collections.
- Apports dans l'écriture de code SQL Les Listeners...

Travaux pratiques : Découverte des lambda-expressions.

3) La programmation fonctionnelle avancée

- L'usage de @FunctionalInterface.
- Les interfaces fonctionnelles fournies par le JDK 8.
- Predicate, Function, Supplier, Consumer...
- Références aux méthodes et aux constructeurs.

Travaux pratiques : Mise en œuvre d'aspects avancés.

PARTICIPANTS

Concepteurs, développeurs, responsables de projets Java.

PRÉREQUIS

Pratique du développement Java, connaissance de Java 5, 6 ou 7 et des bases JavaScript.

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

4) Java 8 Nashorn

- Du JavaScript au sein de Java ? De Rhino présent en Java 7 à Oracle Nashorn, moteur ECMAScript-262 de Java 8.
- Utilisation de la ligne de commande jjs.
- Appeler du code JavaScript depuis une application Java, soit directement soit à l'aide d'un fichier séparé .js.
- Appeler du code Java depuis un script JavaScript.

Travaux pratiques : Appels de JavaScript depuis Java.

5) Nouvelle API Date et Time

- Les difficultés engendrées par les API de Date et GregorianCalendar, l'alternative JodaTime et la JSR 310.
- Les concepts de temps machine et temps humain, les principes de base de la nouvelle API.
- La gestion du temps machine avec java.time.Instant et java.time.Duration.
- Le temps humain avec java.time.LocalDate, java.time.LocalDateTime, java.time.LocalTime.

Travaux pratiques : Utilisation de la nouvelle API de gestion des dates et temps.

6) Autres nouveautés

- Les nouveaux streams.
- Map-reduce.

Travaux pratiques : Mise en œuvre des nouveautés.

LES DATES

Nous contacter