

Programme de formation Java 9, 10 et 11 : les nouveautés du JDK

• Objectifs

Cette formation Java permettra aux apprenants de manipuler les nouvelles fonctionnalités de Java 9, 10 et 11. Plusieurs améliorations significatives sont venues modifier le développement et déploiement d'applications Java : les modules, le JShell, les collections immutables ou encore quelques nouveautés sur les APIs. Ces évolutions entraînent une baisse du poids des applications, des gains de performance et une meilleure sécurité. Un nouveau modèle de release a également été annoncé par Oracle, qui bouleverse ses habitudes avec une version bénéficiant du LTS (Long-Term Support) tous les 3 ans désormais. Notre formation Java 9, 10 et 11 a pour objectif de familiariser les stagiaires avec l'ensemble de ces nouveautés, tout en leur enseignant les meilleures pratiques pour migrer les applications existantes sur la dernière version de Java.

• Pré requis

Être à l'aise avec le développement Java (versions antérieures).

• Durée

2 jours

• Public

Chefs-de-projet,
Concepteurs, Consultants,
Développeurs

• Plan de formation

Introduction à la formation Nouveautés Java 9, 10 et 11

Le nouveau cycle de release Java
La nouvelle approche commerciale d'Oracle
Vue d'ensemble de Java 9
Vue d'ensemble de Java 10
Vue d'ensemble de Java 11
Nouveautés majeures
Exemples de cas pratiques : présentation des participants et du formateur, tour de table et discussions autour des principales évolutions du langage Java et les perspectives d'application post-formation.

Jigsaw : les modules Java 9

Les lacunes de classpath et l'intérêt du module-path
Présentation du système de modularité Java 9 (Jigsaw)
Descripteur de module, exports, requires, provides...
Les différents types de modules
Créer et utiliser des modules
Packaging de modules
Graph de dépendances entre les modules

Avancé : services et runtimes personnalisés
Exemples de cas pratiques : utilisation du nouveau système de modularité, découpage d'une application en plusieurs modules, lancement d'une application à partir des modules, export d'un package, mise en oeuvre de services.

Migration vers Java 9+

Quelles problématiques ? Travailler avec les jars et le classpath
Préparer une migration : mises-à-jour, dépendances...
Avec ou sans modules ? Faut-il tout rendre modulaire ?
Exemple de cas pratique : évolution d'un projet du JDK 8 au JDK 11.

JShell, l'implémentation de REPL (Read Evaluate Print Loop)

Introduction au JShell
Evaluer le code (snippets)
Utiliser les bibliothèques (Modules, Jar, etc)
Les autres possibilités avec JShell : commandes, scripts, etc

Exemples de cas pratiques : exécution de scripts et test de code à la volée avec JShell.

Inference de type

L'inference de type pour les variables locales

Paramètres de lambdas

Syntaxe des variables locales pour les lambdas

Exemples de cas pratiques : utilisation de l'inference de type pour variable locale (Java 10) et de la syntaxe pour les paramètres des expressions lambdas (Java 11).

Modifications et ajouts sur les APIs

Vue d'ensemble des nouveautés dans les APIs

Fonctionnalités dépréciées ou supprimées

Nouvelles APIs

HttpClient : protocole HTTP 2, cas d'usage, fonctionnalités et compatibilité avec les WebSockets

API Optional et les nouvelles méthodes apportées par Java 9

Les améliorations de l'API Stream

Les améliorations de l'API Process (interfaces ProcessHandle et ProcessHandle.Info)

Programmation réactive avec l'API Flow

Exemples de cas pratiques : utilisation des nouvelles méthodes apportées par Java 9, 10 et 11, mise en oeuvre d'un client http,

Autres nouveautés

Les collections immutables : méthodes

Factory, optimisation d'accès

Méthodes privées d'interface

Le garbage collector Epsilon

L'instruction try-with-resource