Formation Java Avancé

Durée :	5 jours
Public :	Développeurs Java
Pré-requis :	Avoir suivi le stage "Java initiation+approfondissement" ou posséder les connaissances équivalentes
Objectifs :	Connaître et maîtriser les concepts avancés du langage
Sanction :	Attestation de fin de stage mentionnant le résultat des acquis
Taux de retour à l'emploi:	Aucune donnée disponible
Référence:	JAV100573-F
Note de satisfaction des participants:	5,00 / 5

Découvrir les nouveautés du langage depuis Java 8

Disparition du permgen space
Méthodes par défaut
Annotations répétées, annotations de type Références de méthodes, Interface avec méthodes default
Méthodes privées d'interface
Libération de ressources
L'API CompletableFuture
Améliorations du garbage collector
Modularité du jdk

Atelier : Installation de l'environnement - démonstration des fonctionnalités - profiling de code et amélioration des performances.

Manipuler des dates

Manipuler des dates et durées avec la nouvelle API time
Faiblesses de la bibliothèque actuelle
Nouvelles classes et interfaces
Choix technique entre durées, instants et dates locales
Gestion des fuseaux horaires
Formatage

Atelier : Utilisation et formatage de dates, opérations sur des dates et des durées

Maîtriser les expressions Lambda et les interfaces fonctionnelles

Expressions lambda : définition, intérêt
Règles d'écritures, déclarations
Interface fonctionnelle : principe, compatibilité
Accès à l’état englobant d’une expression lambda
Implémentation et performances.
Evolutions de l'API Collection

Atelier : utilisation des nouveaux patterns et interfaces fonctionnelles introduites depuis Java 8

Manipuler des collections avec les streams

Nouvelles fonctionnalités
Utilisation de Patterns for-each, replace-if, do-if-absent, do-if-present
Fusions de collections
API Stream : principe, intérêt
Patterns de création de streams
Opérations intermédiaires et terminales sur les streams
Gestion des cas spécifiques : streams vides, optional
Utilisation de parallel streams

Atelier : utilisation des streams avec différents collectors - exploitation de ressources avec les parallel streams - utilisation de patterns

Factoriser du code en utilisant la généricité et la réflexion

Généricité : principe de typage, usages
Classes génériques et contraintes sur les types
Implémentation de méthodes génériques
Interfaces génériques et polymorphisme
Réflexion et classes disponibles
Introspection dynamique d'objets
Invocation de membres d'un objet
Instanciation dynamique d'objets

Atelier : écriture de méthodes génériques pour gérer des imports/exports et des sérialisations de données - Patterns et généricité

Traiter du Javascript

Nashorn : présentation, cas d'usage
Ligne de commande jjs
Interprétation de code JavaScript
Appel de code Java depuis du JavaScript

Atelier : Instanciation et utilisation du moteur Nashorn, utilisation d'objets Java en JavaScript - appel de fonctions Javascript et gestion des paramètres

Manipuler des processus et des threads concurrents

Insanciation de processus
ProcessHandle et ProcessHandle.inf

L'API de concurrence : différents types d'Executors
Choix des interfaces Runnable, Future, Callable
Application de multithreading et utilisation d'expressions Lambda
ComposableFuture
Gestion du mode asynchrone et du timeout
Files d'attente et classes atomic
Gestion des pools de threads

Atelier : Codage de tâches planifiées à l'aide d'executors - implémentation multi-tâches avec l'utilisation de threads et synchronisation