

Formation Multi-threading et gestion de la mémoire en Java

■ Durée :	2 jours (14 heures)
■ Tarifs inter-entreprise :	1 275,00 € HT (standard) 1 020,00 € HT (remisé)
■ Public :	Développeurs et analystes programmeurs Java
■ Pré-requis :	Avoir suivi le stage "Java initiation" ou posséder les connaissances équivalentes
■ Objectifs :	Maîtriser l'implémentation du multi-threading en Java - Gérer efficacement la concurrence et la mémoire
■ Modalités pédagogiques, techniques et d'encadrement :	<ul style="list-style-type: none">• Formation synchrone en présentiel et distanciel.• Méthodologie basée sur l'Active Learning : 75 % de pratique minimum.• Un PC par participant en présentiel, possibilité de mettre à disposition en bureau à distance un PC et l'environnement adéquat.• Un formateur expert.
■ Modalités d'évaluation :	<ul style="list-style-type: none">• Définition des besoins et attentes des apprenants en amont de la formation.• Auto-positionnement à l'entrée et la sortie de la formation.• Suivi continu par les formateurs durant les ateliers pratiques.• Évaluation à chaud de l'adéquation au besoin professionnel des apprenants le dernier jour de formation.
■ Sanction :	Attestation de fin de formation mentionnant le résultat des acquis
■ Référence :	JAV1065-F
■ Note de satisfaction des participants:	Pas de données disponibles
■ Contacts :	commercial@dawan.fr - 09 72 37 73 73

■ Modalités d'accès :	Possibilité de faire un devis en ligne (www.dawan.fr , moncompteformation.gouv.fr , maformation.fr , etc.) ou en appelant au standard.
■ Délais d'accès :	Variable selon le type de financement.
■ Accessibilité :	Si vous êtes en situation de handicap, nous sommes en mesure de vous accueillir, n'hésitez pas à nous contacter à referenthandicap@dawan.fr , nous étudierons ensemble vos besoins

Implémenter le multi-threading en Java

Notions de threads et de processus

Le Contexte Switching : fonctionnement, conséquences

Parallélisme et accès concurrent

Synchronisation Java : verrous, synchronized, volatile

Bonnes pratiques de synchronisation et erreurs à éviter (Deadlock)

Implémentation de coordination des threads

Immuabilité : publications, encapsulation confinement

Collections synchronisées, collections concurrentes

Atomics

Lock, Sémaphore

CountDownLatch

Pooling de threads, executors

Atelier : implémentation de scénarios en multi-threads, synchronisation - utilisation des executor.

Gérer efficacement la mémoire au sein de la JVM

Organisation de la mémoire de la JVM : stack, heap

Présentation du mécanisme de ramasse-miettes

Fonctionnement du Garbage Collector

Garbage Collectors générationnels

Garbage Collector et fuites mémoires

Paramétrage de la JVM pour optimiser l'emprise mémoire

Présentation des outils de la JVM pour monitorer une application : jConsole, jVisualVM

Atelier : optimisation de la JVM pour une application - monitoring : dump et analyse de code.