

## Formation Multi-threading et gestion de la mémoire en Java

<b>Durée :</b>	2 jours
<b>Public :</b>	Développeurs et analystes programmeurs Java
<b>Pré-requis :</b>	Avoir suivi le stage "Java initiation" ou posséder les connaissances équivalentes
<b>Objectifs :</b>	Maîtriser l'implémentation du multi-threading en Java - Gérer efficacement la concurrence et la mémoire
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	JAV1065-F
<b>Note de satisfaction des participants:</b>	Pas de données disponibles

### 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.**