

## Java niveau avancé

### Maîtriser les API avancées de Java

Ce cours aide les développeurs à améliorer leur pratique de Java. Il aborde concrètement les principaux concepts avancés, les design patterns importants ainsi que les bases d'une architecture à objets distribués.

#### Détails

- Code : JA-AV
- Durée : 2 jours ( 14 heures )

#### Public

- Architectes
- Consultants
- Développeurs
- Ingénieurs

#### Pré-requis

- Bonne pratique de Java

#### Objectifs

- Développer une application complexe en Java
- Comprendre le fonctionnement de la machine virtuelle (JVM), savoir en modifier le comportement
- Réaliser une application multi-tâches et temps réel
- Sécuriser correctement son application
- Utiliser une base de données avec Java

#### Programme

##### Concepts et syntaxes avancés du langage

- Les « import static »
- Autoboxing et auto-unboxing des types primitifs
- Les types génériques
- Amélioration des boucles avec les collections
- Introspection et réflexion
- Méthodes à nombre d'arguments variable
- Annotations : utilisation, création

##### Comprendre la JVM (Machine Virtuelle)

- Comprendre comment fonctionne le garbage collector et la gestion de la mémoire, forcer un garbage
- Optimiser l'utilisation de la mémoire, modifier les paramètres de lancement
- Comprendre le class loader
- Modifier la stratégie de chargement des classes : écriture d'un class loader
- Piloter la JVM via API

##### Gérer les accès concurrents

- Définition d'un thread
- Créer et manipuler les threads
- Synchroniser les threads sur un objet moniteur
- Exemple de temps réel
- Bonnes et mauvaises pratiques

##### La sécurité en Java

- Les contextes : standalone, applet, serveur Java EE
- Les API de sécurité (JAAS, JCE, JSSE)
- La configuration
- Cryptographie et PKI
- Signer une applet java

##### Principes des architectures n-tiers, pattern MVC et Design Patterns

- Le couple Observer/Observable
- Façade
- Fabrique abstraite
- Inversion de contrôle
- Itérateur et Visiteur

##### Accéder à une base de données relationnelle avec JDBC

- Architecture et types de pilotes JDBC
- Charger le pilote et récupérer une connexion à la base de données
- Exécuter des requêtes SQL, requêtes paramétrées
- Exploitation des résultats

##### Notion de persistance objet :

- Principes
- Data Transfer Object
- Exemple d'Hibernate

##### Gestion de flux d'entrées/sorties

- Flux binaires, encodages
- Sérialisation/désérialisation d'objets
- Les modifieurs Java et leurs impacts

##### Architecture à objets distribués : RMI (Remote Method Invocation)

- Présentation
- Appel de méthodes d'un objet distant (TCP-IP)
- Appel de méthodes d'un objet distant (RMI)

- Annuaire d'objets : RMIRegistry
- Interface Remote
- Le compilateur rmic
- Stub et skeleton
- Construction du client et du serveur
- Introduction à CORBA et IIOP

## Modalités

---

- **Type d'action** : Acquisition des connaissances
- **Moyens de la formation** : Formation présentielle - 1 poste par stagiaire - 1 vidéo projecteur - Support de cours fourni à chaque stagiaire
- **Modalités pédagogiques** : Exposés - Cas pratiques - Synthèse
- **Validation** : Exercices de validation - Attestation de stages
- **Profils des intervenants** : Toutes nos formations sont animées par des consultants-formateurs expérimentés et reconnus par leurs pairs.