

## **Programmation avancé Java SE 8**

**Durée:** 5 jours

2450 €

20 au 24 mars

15 au 19 mai

25 au 29 septembre

4 au 8 décembre

### **Public:**

Développeurs java, ingénieurs logiciels et architectes d'applications.

### **Objectifs:**

Approfondir la connaissance de Java notamment dans les domaines de la programmation multi-tâches, des I/O, des tests et du logging des technologies jdbc et de la persistance avec JPA.

### **Connaissances préalables nécessaires:**

Maîtriser les concepts objets et les bases du langage Java, ou avoir suivi le stage « Concepts Objets et bases Java »

### **Programme:**

- Java SE : Présentation des et rappels sur les classes Java.  
Les interfaces et les expressions lambda.
- Collections : Création de collections : ArrayList, TreeSet, HashMap, etc ...  
L'interface Stream.  
Filtrage de collections avec les expressions Lambda.
- Le multi-threading : Fonctionnement.  
Ordonnancement et priorité.  
Exclusion mutuelle. Synchronisation.  
Thread démon. Communication par flux "pipe".
- L'API de concurrence : Les exécuteurs de tâches. Les queues. Les maps atomiques. La représentation du temps et de ses unités. Les synchroniseurs.  
Les traitements asynchrones anticipés. Les variables atomiques.  
Les verrous "haute performance".
- Les annotations : Objectif. l'API Reflection. Annotations standards. Les méta-annotations.  
Fabriquer ses annotations. Annotation Processing Tool (APT)
- La gestion des I/O : La gestion des flux standards.  
Lecture/écriture depuis la console  
Utilisation des streams  
API d'accès aux fichiers (NIO.2)

## **Programmation avancé Java SE 8**

- L'API JDBC** : Objectif de JDBC. Les types de drivers. Les architectures applicatives.  
Les classes et interfaces en jeu. Connexion.  
La gestion des transactions et l'isolation transactionnelle.  
Interrogation et mise à jour.  
Appel d'une procédure stockée. Les types de données. Les pools de connexion.  
Les Rowset. La libération des ressources.  
La gestion des exceptions.  
Présentation de JPA (Java Persistence API).
- Les tests** : Objectif. Le framework JUnit.
- Traçabilité des applications** : Objectif. L'API Java Logging.  
Gestion des dates avec l'API Java Date/Time  
Création de timestamps.