



## Programmation avancée Java

LJ002

Durée: 5 jours

### Public :

Développeurs java, ingénieurs logiciels et architectes d'applications.

### Objectifs :

Approfondir la connaissance de Java à un niveau avancé notamment dans les domaines de la programmation multi-tâches, des I/O, de la programmation réflexive et du logging des technologies jdbc pour l'accès aux bases de données, 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.  
Technique de parcours d'une collection.  
Bonnes pratiques mises en oeuvre sur des exercices d'application.

#### Le multi-threading

Fonctionnement.  
Ordonnement et priorité.  
Exclusion mutuelle. Synchronisation.  
Thread démon. Communication par flux "pipe".  
Mise en oeuvre sur des exemples , lien collection et thread.

#### 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".



# Phirio

---

## Les annotations et les génériques.

Objectif. L'API Reflection. Annotations standards. Les méta-annotations.  
Annotation Processing Tool (APT)  
Principe de l'introspection : explications et démonstrations sur des exemples.

Atelier : mise en oeuvre, fabriquer ses propres annotations.

---

## La gestion des I/O et des fichiers.

La gestion des flux standards.  
Lecture/écriture depuis la console  
Utilisation des streams  
API d'accès aux fichiers (NIO.2)

Atelier : exercices d'accès I/O fichiers, optimisation

---

## 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).

---

## Architectures 3 tiers

Le standard JEE  
Le modèle MVC : Model View Controller.  
Technologies Java, XML  
Rappel sur la méthode de construction d'une application avec les pages jsp et les servlets.  
Architectures 3 tiers : explication des couches présentation, modèle , persistance

Atelier : démonstrations applications Jee

---

## Bonnes pratiques : les tests et la traçabilité des applications.

Objectif. Le framework JUnit.  
Objectif. L'API Java Logging.  
Gestion des dates avec l'API Java Date/Time  
Création de timestamps.