

# **PROGRAMMATION AVANCÉ JAVA SE 8**

## **CODE STAGE : J-AV2**

### **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.

### **DURÉE**

5 jours

### **PUBLIC**

Développeurs java, ingénieurs logiciels et architectes d'applications.

### **PRÉ-REQUIS**

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

### **PROGRAMME**

#### **1. JAVA SE**

Présentation des et rappels sur les classes Java.  
Les interfaces et les expressions lambda.

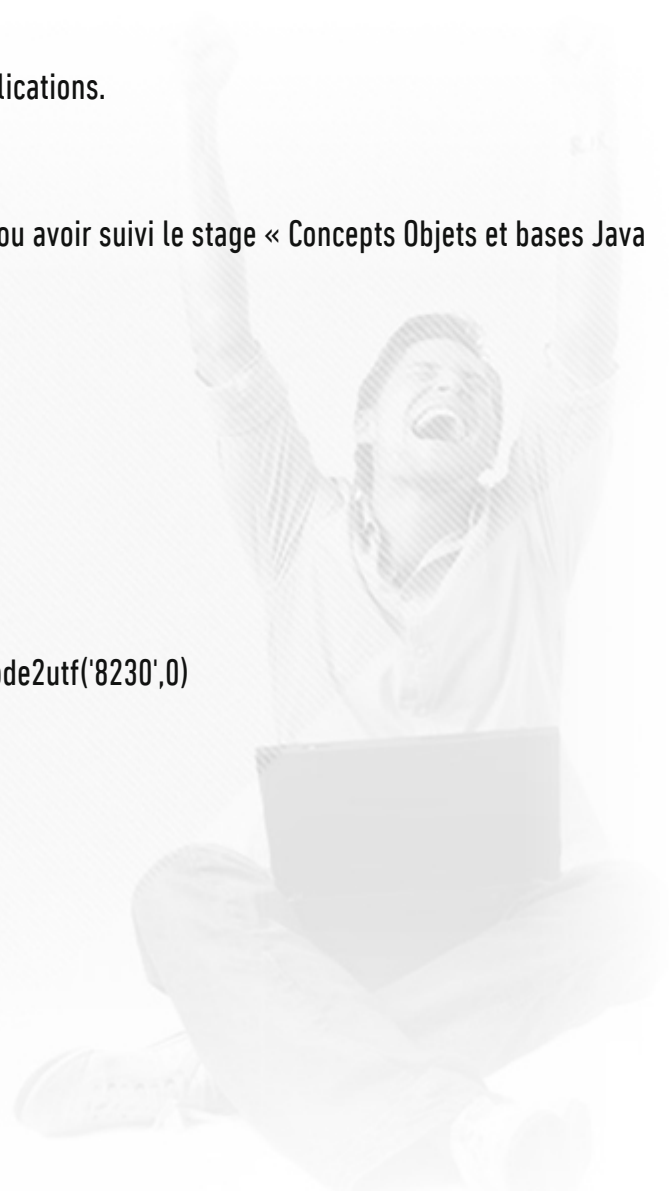
#### **2. COLLECTIONS**

Création de collections : ArrayList, TreeSet, HashMap, etc code2utf('8230',0)  
L'interface Stream.  
Filtrage de collections avec les expressions Lambda.

#### **3. LE MULTI-THREADING**

Fonctionnement.  
Ordonnancement et priorité.  
Exclusion mutuelle. Synchronisation.  
Thread démon. Communication par flux « pipe ».

#### **4. 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 ».

## 5. LES ANNOTATIONS

Objectif. L'API Reflection. Annotations standards. Les méta-annotations.

Fabriquer ses annotations. Annotation Processing Tool (APT)

## 6. 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)

## 7. 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).

## 8. LES TESTS

Objectif. Le framework JUnit.

## 9. TRACABILITE DES APPLICATIONS

Objectif. L'API Java Logging.

Gestion des dates avec l'API Java Date/Time

Création de timestamps.

