# Bienvenue à ProSkills IT – Formations professionnelles au Togo

# Fiche du cours

65 h

#### Titre:

JAV200 - Prgrammation Java 2

### **Description:**

Niveau intermédiaire : on renforce la POO avancée, les structures de données & algorithmes, les collections (performances), l'I/O (fichiers/flux), la concurrence (threads, executors, CompletableFuture) et le réseau (sockets, HTTP client). On introduit JavaFX (GUI), logging et des tests plus complets (JUnit 5/Mockito).

## Objectifs:

- Maîtriser interfaces, classes abstraites, génériques (avancés), records et sealed classes (aperçu).\*
- Implémenter et comparer listes chaînées, piles, files ; pratiquer tris/recherches (complexité Big-O).\*
- Utiliser Collections de manière performante (iterator, comparator, Map vs Set, vues immuables).\*
- Travailler les flux & fichiers (I/O, NIO.2), sérialisation (aperçu).\*
- Programmer en concurrent : threads, Executors, synchronisation, CompletableFuture.\*
- Écrire des applications réseau : sockets TCP/IP, client HTTP moderne.\*
- Introduire JavaFX (GUI); améliorer la qualité: JUnit 5, Mockito, logging (SLF4J/Logback).

### Chapitres:

- 1. POO avancée : interfaces, classes abstraites, génériques avancés (bornes, wildcards)\*
- 2. Langage moderne: lambdas, Stream API, Optional, switch pattern (selon version)\*
- 3. Structures de données : listes chaînées, piles, files ; implémentations & usages\*
- 4. Algorithmes: tri (insertion/merge/quick principes), recherche (linéaire/binaire), complexité\*
- 5. Collections avancées : comparateurs, vues, maps spécialisées, immutabilité\*
- 6. Fichiers & flux: NIO.2, buffers, parcours de répertoires, sérialisation (aperçu)\*
- 7. Concurrence: threads, java.util.concurrent, pools d'exécution, synchronisation, CompletableFuture\*
- 8. Réseau : sockets (serveur/client), protocole simple ; HttpClient (GET/POST, JSON)\*
- 9. Qualité: JUnit 5, Mockito (mocks/stubs), logging (SLF4J/Logback)\*
- 10. JavaFX (intro): scènes, contrôles, events; patterns MVC (aperçu)\*

- 11. Organisation du code : architecture modulaire (aperçu), packaging, conventions\*
- 12. Mini-projet : appli modulaire avec I/O, concurrence et réseau + tests

## À la fin:

À l'issue du module, vous serez capables de concevoir des applications Java modulaires, fiables et performantes, de choisir/implémenter les structures de données adaptées et d'écrire du code concurrent (Executors, CompletableFuture) et réseau (sockets/HTTP client).

Vous livrerez un projet testé (JUnit 5/Mockito), journalisé proprement (SLF4J/Logback) et configuré de façon claire (Maven/Gradle, profils).

Vous saurez aussi diagnostiquer des problèmes fréquents de performance (collections mal choisies, N+1, contention simple) et documenter votre travail (README, scripts) pour préparer sereinement la montée vers JAV300.