Formation : Mise à jour vers Java 21 – Nouvelles fonctionnalités et migration

Module 1 – Introduction à la mise à jour de Java

- Pourquoi Java évolue désormais tous les 6 mois (modèle de release semestrielle)
- Versions LTS (11, 17, 21) et enjeux pour les entreprises
- Différences clés entre Java 11 et Java 21 : syntaxe, performances, sécurité, outils
- Choisir le bon JDK : Oracle, OpenJDK, Temurin, Amazon Corretto
- Mise à jour des IDE (IntelliJ, Eclipse, VS Code)
- Configuration de Maven et Gradle pour Java 21
- Bonnes pratiques pour planifier une migration
- **Démonstration :** migration rapide d'un projet Java 11 → 21

Module 2 – Évolutions du langage Java

- Utilisation de var dans les lambdas et boucles
- Nouveau switch et pattern matching
- Text Blocks (""") pour la gestion des chaînes
- Records et Sealed Classes
- Pattern Matching étendu et usages concrets
- **Atelier :** refactorisation d'un code Java 11 vers un style Java 21

Module 3 – Améliorations des API standard et productivité

- Nouvelles méthodes Stream : toList(), mapMulti(), teeing()
- Améliorations des classes Optional, List.of, Map.of, Set.of
- Nouveautés dans String, Files, Math, Random
- Découverte des outils modernes : jshell, jdeps, jpackage, jlink
- Introduction à Project Amber
- **Atelier :** modernisation de flux et opérations sur fichiers

Module 4 – Performances et fonctionnement de la JVM

- Nouveaux Garbage Collectors : G1, ZGC, Shenandoah
- Optimisations de la gestion mémoire et du runtime
- Outils d'analyse : Java Flight Recorder, Mission Control
- Introduction à GraalVM et au compilateur JIT
- **Atelier :** profilage et analyse d'une application Java

Module 5 – Concurrence moderne et Project Loom

- Threads classiques vs Virtual Threads
- Project Loom : simplification de la gestion concurrente
- Structured Concurrency (preview Java 21)
- Intégration avec ExecutorService, HTTP, JDBC
- Comparaison avec CompletableFuture et la programmation réactive
- **Atelier :** conversion d'un code multi-thread en version Loom

Module 6 – Migration, sécurité et fonctionnalités avancées

- Migration pas à pas avec jdeps, jlink, Maven/Gradle
- API supprimées ou dépréciées
- JPMS (Java Platform Module System) : module-info.java, requires, exports
- Sécurité : TLS 1.3, certificats, KeyStore/TrustStore
- Foreign Function & Memory (FFM) API
- **Atelier final :** migration complète d'un mini-projet Java 11 → 21