

Formation Java : Approfondissement (Nouvelle version)

■ Durée :	2 jours (14 heures)
■ Tarifs inter-entreprise :	1 175,00 CHF HT (standard) 940,00 CHF HT (remisé)
■ Public :	Développeurs Java débutants ou intermédiaires
■ Pré-requis :	Avoir suivi la formation « Java SE - Initiation » ou posséder un niveau équivalent - Savoir écrire une application console simple en Java en utilisant classes, héritage, exceptions et collections - Connaître les bases du SQL (SELECT, INSERT, UPDATE, DELETE).
■ Objectifs :	Approfondir l'utilisation des collections, des comparateurs, des lambdas et des premiers streams pour manipuler des ensembles de données - Lire et écrire des fichiers texte avec les API modernes de Java et externaliser la configuration de l'application - Configurer et utiliser JDBC pour se connecter à une base de données relationnelle et réaliser des opérations CRUD - Gérer les erreurs et transactions liées aux accès aux données afin de garantir l'intégrité des informations - Organiser le code en couches (présentation console, services métier, DAO) pour améliorer la maintenabilité - Finaliser une application console complète en Java intégrant POO, collections, fichiers et persistance en base.
■ Modalités pédagogiques, techniques et d'encadrement :	• Formation synchrone en présentiel et distanciel. • Méthodologie basée sur l'Active Learning : 75 % de pratique minimum. • Un PC par participant en présentiel, possibilité de mettre à disposition en bureau à distance un PC et l'environnement adéquat. • Un formateur expert.

■ Modalités d'évaluation :	• Définition des besoins et attentes des apprenants en amont de la formation. • Auto-positionnement à l'entrée et la sortie de la formation. • Suivi continu par les formateurs durant les ateliers pratiques. • Évaluation à chaud de l'adéquation au besoin professionnel des apprenants le dernier jour de formation.
■ Sanction :	Attestation de fin de formation mentionnant le résultat des acquis
■ Référence :	JAV102783-F
■ Demandeurs d'emploi:	Des entreprises recrutent des demandeurs d'emploi qui ont suivi ce cours dans le cadre d'une POEI, contactez-nous au 09.72.37.73.73 pour plus d'informations.
■ Note de satisfaction des participants:	4,59 / 5
■ Contacts :	commercial@dawan.fr - 09 72 37 73 73
■ Modalités d'accès :	Possibilité de faire un devis en ligne (www.dawan.fr , moncompteformation.gouv.fr , maformation.fr , etc.) ou en appelant au standard.
■ Délais d'accès :	Variable selon le type de financement.
■ Accessibilité :	Si vous êtes en situation de handicap, nous sommes en mesure de vous accueillir, n'hésitez pas à nous contacter à referenthandicap@dawan.fr , nous étudierons ensemble vos besoins

Approfondir les collections, lambdas et premiers streams

Revenir sur le choix des collections selon le besoin fonctionnel

Trier des collections à l'aide de Comparable et Comparator

Introduire les expressions lambda pour écrire des traitements compacts sur les objets

Découvrir les interfaces fonctionnelles courantes (Runnable, Comparator, Predicate, etc.)

Utiliser l'API Stream pour filtrer, transformer et agréger des données de manière lisible

Mettre en pratique quelques opérations courantes : filter, map, sorted, collect

Atelier fil rouge : Implémenter des recherches, tris et statistiques sur les collections de l'application

Lire et écrire des fichiers, externaliser la configuration

Utiliser l'API NIO.2 (java.nio.file) pour lire et écrire des fichiers texte
Gérer les chemins, répertoires, création et suppression de fichiers
Mettre en place une logique d'import/export simple de données au format texte (CSV ou équivalent)

Externaliser des paramètres dans des fichiers de propriétés (config applicative)
Appliquer une gestion d'erreurs cohérente sur les opérations d'entrée/sortie

Atelier fil rouge : Ajouter à l'application des fonctions d'import/export de données et un fichier de configuration

Accéder à une base de données relationnelle avec JDBC

Comprendre l'intérêt de la persistance en base de données pour une application Java
Configurer un pilote JDBC et établir une connexion à une base de données (type PostgreSQL, MySQL, etc.)

Écrire et exécuter des requêtes SQL simples : SELECT, INSERT, UPDATE, DELETE

Traiter les résultats avec ResultSet et mapper les lignes vers des objets métier

Utiliser les PreparedStatement pour paramétrer les requêtes et sécuriser les entrées

Gérer les transactions : commit, rollback, gestion des erreurs liées à la base

Atelier fil rouge : Persister les principales entités de l'application en base (création, consultation, mise à jour, suppression)

Structurer la couche d'accès aux données et l'architecture de l'application

Organiser le code en couches simples : présentation console, services métier, accès aux données

Mettre en place le pattern DAO pour isoler l'accès à la base

Séparer les responsabilités pour faciliter les tests et l'évolution de l'application

Introduire la notion de tests simples (exécution de scénarios via main, jeux de données) pour valider les traitements

Préparer la transition vers des frameworks plus avancés (JPA, Spring, etc.) sans les utiliser encore

Atelier fil rouge : Réorganiser le code existant en couches et factoriser l'accès aux données dans des DAO dédiés

Consolider les acquis par un projet complet en ligne de commande

Reprendre le cahier des charges global de l'application console (gestion d'un mini SI : stock, clients, commandes, etc.)

Valider l'utilisation cohérente de la POO, des collections, des exceptions et de l'accès aux données

Compléter les fonctionnalités manquantes, améliorer les messages d'erreur et les contrôles de cohérence

Mettre en place quelques scénarios de test de bout en bout

Faire le lien avec les compétences évaluées dans le cadre de la certification

“Développer en langage Java” (sans imposer l’examen)

Atelier fil rouge : Finaliser l’application complète, exécuter des scénarios réalistes et présenter le travail réalisé