

# Formation Java Avancé, perfectionnement

## Description de la formation Java Avancé

Cette formation Java Perfectionnement permet aux personnes ayant suivi une première initiation à Java, d'approfondir leurs connaissances du langage afin d'intégrer directement une équipe projet.

La formation commence par détailler le fonctionnement interne de la JVM (Java Virtual Machine) afin que le développeur puisse diagnostiquer des problèmes de performances liés à la conception de son programme et à l'utilisation des objets Java que ce soit en natif ou via l'utilisation de framework (Spring, Hibernate, etc.).

Une introduction aux Design Patterns vous permettra d'aborder les bonnes pratiques de conception tout en faisant le lien avec le chapitre précédent sur les performances.

Vous apprendrez ensuite à utiliser le langage Java dans le contexte des bases de données relationnelles (JDBC, JPA/Hibernate), ce qui apporte une dimension de projet professionnel à la formation.

La manipulation avancée de fichiers/répertoires est également visitée avec les API Nio.2 Les nouvelles API de gestion des dates et du temps sont également abordées, et vous comprendrez la nécessité des tests à travers l'exemple de JUnit.

Pour finir, vous apprendrez la manipulation des expressions lambdas et la mise en œuvre de l'API Stream.

## Objectifs

### Objectifs Opérationnels :

Maitriser les aspects le plus avancés du langage Java

### Objectifs Pédagogiques :

A l'issue de cette formation Java Avancé vous serez en mesure de :

- Avoir compris le fonctionnement de la JVM
- Connaître les bonnes pratiques de conception
- Programmer en Java dans le contexte des bases de données relationnelles
- Développer une application Java multitâches en utilisant les threads
- Utiliser les expressions Lambdas
- Mettre en œuvre l'API Stream
- Connaître les nouveautés Java 8 à 11

## À qui s'adresse cette formation ?

### **Public :**

Cette formation Java s'adresse aux développeurs et chefs de projet. Elle peut intéresser également ceux qui cherchent à rafraîchir des concepts vus lors d'une précédente formation Java suivie il y a plusieurs mois sans pratique ultérieure.

### **Prérequis :**

Cette formation s'adresse à tout développeur possédant une connaissance du langage Java.

## Contenu du cours Java Avancé

Fonctionnement de la JVM (Java Virtual Machine)

Chargement des classes, Classloader, classpath

Gestion de la mémoire, le garbage collector



ITgate

Training

Your Gateway to Excellence

Le modèle de sécurité de Java

La réflexion, les annotations

Les outils d'inspection, surveillance, traces

Travaux Pratiques :

Observation des cycles d'allocation/désallocation des objets et des performances associées.

L'apport des mécanismes de pool, de certains design pattern. Sollicitation de la JVM par les framework Java (Spring, Hibernate).

### Introduction aux Design Patterns

Programmation par interfaces

Introduction aux design patterns du GoF (Gang Of Four)

Les documentations de référence

Pattern Singleton

Pattern Factory

Pattern Decorator

Pattern Observer

Pattern Proxy

Architecture MVC

Travaux Pratiques :

Mise en œuvre des associations un-vers-un, un-vers-plusieurs, plusieurs-vers-plusieurs. Mise en œuvre de la composition. Implémentation du pattern d'architecture MVC. Description de problèmes invitant le stagiaire à identifier le "bon" design pattern à utiliser

### Programmation parallèle

Concepts et spécificités du multithreading par rapport à la programmation « traditionnelle »

États d'un thread

Synchronisation des threads

Interfaces Lock et Condition

Pools de threads



ITgate

Training

Your Gateway to Excellence

Les apports du framework Executor (invocation, ordonnancement, exécution, contrôle)

Le package java.util.concurrent

Travaux Pratiques :

Écriture d'un chat en Java (multi-fenêtres en lecture/écriture, passage du jeton entre les threads)

### L'accès aux bases de données

Historique de l'accès aux données en Java, présentation de JDBC

L'API JDBC, Connection, Statement, ResultSet

Gestion des transactions

Problématique de l'ORM et introduction à JPA

Configuration JPA, l'exemple d'Hibernate

Mapping basique, API de l'EntityManager, les transactions

Mapping de l'héritage, des associations, lazy-loading et cascading

Introduction au langage JPQL

Travaux Pratiques :

Mise en œuvre de JPA avec Hibernate pour rendre persistants des beans entités

Utilisation du langage de requêtes JPQL pour rechercher des beans entités

Mise en œuvre de l'héritage et d'associations unidirectionnelles et bidirectionnelles

### Accès complet au système de gestion de fichiers avec Nio.2

Les nouvelles classes java.nio.file.Files et java.nio.file.Path

Manipulation de chemins, parcours de répertoires et d'arborescences

Accès aux attributs d'un fichier

Notification des changements dans un répertoire

Gestion des archives Jar/zip

Entrées/sorties asynchrones

Travaux pratiques



ITgate

Training

Your Gateway to Excellence

Synchronisation de répertoires entre sites distants (détection d'ajout ou de suppression de fichiers dans un répertoire, comparaison des données temporelles sur les fichiers/répertoires des deux sites).

## Java et les tests unitaires

Présentation, intérêt

Automatisation et intégration des tests dans le processus de builds, l'exemple de Maven

Mise en œuvre avec JUnit5, assertions, annotations

Cycle de vie d'exécution d'un test

Données de tests

Suite de tests

Travaux Pratiques :

Création de tests JUnit pour tester des classes

Mise en œuvre d'une suite de tests

## Nouvelle API Date/Time

La gestion du temps « classique » et sa lourdeur de manipulation en Java (jour, mois, année, heure, minutes, sec.)

Les concepts de temps machine et temps humain avec la nouvelle API

La gestion du temps machine avec `java.time.Instant` et `java.time.Duration`

Le temps humain avec `java.time.LocalDate`, `java.time.LocalTime`, `java.time.LocalDateTime`

Travaux pratiques

Calcul de la date de règlement d'une facture

Analyse de chaînes de caractères pour conversion en dates

Calcul différentiel sur le temps (heure/minutes/secondes)

Gestion des décalages de fuseau horaire



ITgate

Training

Your Gateway to Excellence

## Expressions Lambda, références de méthodes

Comprendre les principes de la programmation fonctionnelle

Syntaxe

des classes internes anonymes

Notion d'interface fonctionnelle

Anatomie d'une expression Lambda

Simplifier les expressions lambdas avec les références de méthodes

Mise en œuvre des interfaces fonctionnelles : Consumer, Supplier, Predicate, Function

Travaux Pratiques :

Remplacement de classes internes anonymes par des expressions lambda

Mise en œuvre d'interfaces fonctionnelles de la bibliothèque

Simplification d'expressions lambdas par des références de méthodes

## Mise en œuvre de l'API Stream pour la manipulation des flux de données

Objectif de l'API Stream

Obtention et sortie d'un stream

Opérations intermédiaires et terminales

Mise en œuvre des méthodes forEach, map, filter, findFirst

Principe de l'évaluation paresseuse

Intérêt de la la classe Optional

Réduction avec reduce et collect

Interface Collector et classe Collectors

Autres méthodes : limit, skip, sorted, min, max, distinct, allMatch, anyMatch, noneMatch, count

Streams parallèles, précautions d'usage

Travaux Pratiques :

Utilisation de nombreux streams séquentiels sur tableaux et collections

Mise en œuvre de streams parallèles

Comparaison de performances entre streams séquentiels et parallèles

---

**Capital Social:** 50000 DT **MF:** 1425253/M/A/M/000 **RC:** B91211472015

**Tél. / Fax.:** +216 73362 100 **Email:** contact@itgate-training.com **Web:** www.itgate-training.com

**Adresse :** 12 Rue Abdelkader Daghrrir - Hammam Sousse 4011 – Tunisie

## Nouveautés de Java 8 à Java 11

Exécution de scripts Java avec JShell

Nouvelle organisation de la documentation des API Java

API Parser pour Nashorn

Notion de module

Utilisation des modules

Modularité du JDK

Déploiement d'applications Java

Rôle du linker Java jlink

Fichiers jar avec module.info

Nouveau format des fichiers jar avec JMOD

Nouvelle gestion des versions de Java

Gestion de licences pour les JDK11+

Mot-clé var

Nouveau client http

Exécution directe d'un fichier source java

## Travaux Pratiques :

Migration d'une application Java 8 vers Java 11 (mise en œuvre des modules)