



Java SE 17 Developer - Partie 2

Code : NDJ22

Durée : 4 jours

Classe : Présentiel / à distance

Public

- Cette formation Java s'adresse aux développeurs qui souhaitent se préparer au passage de l'examen officiel 1Z0-829 d'Oracle, permettant d'obtenir la certification Oracle Certified Professional, Java SE 17 Developer.

Prérequis

- Pour tirer pleinement profit de cette formation de préparation à l'examen de certification, il est nécessaire d'avoir au préalable de bonnes bases en programmation Java 17 ainsi qu'une bonne pratique.

Objectifs

Objectif opérationnel :

- Réussir la certification Java SE 17 Developer

Objectifs pédagogiques :

- A l'issue de cette formation de préparation à la certification Java 17 vous aurez acquis les connaissances et les compétences nécessaires pour :
 - 1.Effectuer un choix stratégique vous évitant de perdre du temps inutilement lors de l'examen
 - 2.Déterminer les sujets qui nécessiteront un travail de préparation complémentaire
 - 3.Augmenter vos chances de réussir l'examen de la certification

Programme détaillé

(les chapitres 1 à 7 sont étudiés dans la première partie de la préparation)

8- Expressions Lambdas et interfaces fonctionnelles

- Comprendre les principes de la programmation fonctionnelle
- Syntaxe des classes internes anonymes
- Notion d'interface fonctionnelle
- Anatomie d'une Expression Lambda
- Mise en œuvre des interfaces fonctionnelles prédéfinies : Consumer, Supplier, Predicate, Function, UnaryOperator, BinaryOperator
- Variantes spécialisées pour types primitifs des interfaces fonctionnelles prédéfinies

9- Collections et généricité

- Principe de l'autoboxing
- Créer et utiliser une classe générique
- Créer et utiliser une méthode générique
- Utilisation du caractère ? dans un paramètre générique
- Présentation des interfaces List, Set, DeQue, Map
- Savoir choisir un type de collection
- Créer et utiliser les objets ArrayList, TreeSet, TreeMap et ArrayDeque
- Parcours de collections avec itérateur ou boucle for-each
- Rôle des méthodes equals et hashCode
- Utilisation des classes enveloppes
- Méthodes de la classe Collections

10- Streams

- Obtention de streams
- Méthodes intermédiaires et terminales
- Exploiter les méthodes peek et map , y compris les versions primitives de map ainsi que flatMap
- Rechercher des données au moyen de méthodes des classes de l'API Stream comme findFirst, findAny, anyMatch, allMatch, noneMatch
- Savoir utiliser les méthodes groupingBy et partitioningBy





Java SE 17 Developer - Partie 2

Code : NDJ22

Durée : 4 jours

Classe : Présentiel / à distance

- Utiliser les méthodes de la classe Optional
- Trier une collection au moyen de l'API Stream
- Utiliser les méthodes reduce et collect pour effectuer des réductions
- Principe de l'évaluation paresseuse
- Streams sur types primitifs
- Utiliser les streams parallèles pour améliorer les performances

11- Gestion des exceptions et Internationalisation

- Lire et configurer la langue/le pays avec un objet Locale
- Formater les dates et le temps
- Gérer les zones géographiques
- Créer et lire un fichier Properties
- Créer un fichier Properties par langue et le charger dans l'application via un resource bundle
- Formatage des nombres et des dates

12- Mise en œuvre des modules

- Notion de module, format du descripteur de module
- Directives export, requires, uses, provides, opens
- Mise en œuvre de modules
- Gestion des dépendances entre modules
- Compilation/exécution d'un projet modulaire
- Rôle du modulepath
- Nouvelle organisation de la documentation des API Java
- Modules nommés, anonymes et automatiques
- Mise en œuvre de services

13- Les accès concurrents en multi-threading

- Créer des threads avec Runnable, Callable
- Utiliser un ExecutorService pour exécuter les tâches
- Identifier les problèmes potentiels du multi-threading parmi les deadlocks, starvation, livelock, race conditions
- Utiliser le mot-clé synchronized, les verrous Lock
- Utiliser les classes Atomic
- Utiliser les collections java.util.concurrent et les classes CyclicBarrier et CopyOnWriteArrayList
- Utiliser le framework Fork/Join

14- Entrées/sorties

- Savoir choisir entre les hiérarchies InputStream/OutputStream et Reader/Writer
- La classe File
- Lire des données au clavier et écrire sur la console
- Utiliser les classes du package java.io, notamment BufferedReader, BufferedWriter, File, FileReader, FileWriter, FileInputStream, FileOutputStream, PrintWriter
- ObjectOutputStream, ObjectInputStream
- Streams sur fichiers
- Utilisation de l'interface Path pour manipuler les chemins de répertoires ou de fichiers
- Manipulation de fichiers/répertoires avec la classe Files
- Gérer les attributs d'un fichier ou répertoire et autres méta-données
- Utiliser l'API Stream sur fichiers et répertoires





Java SE 17 Developer - Partie 2

Code : NDJ22

Durée : 4 jours

Classe : Présentiel / à distance

15- Accès aux bases de données avec JDBC

- Rôle d'un driver JDBC
- Utiliser la classe DriverManager pour se connecter à une base de données
- Mettre en oeuvre les interfaces Connection , Statement , PreparedStatement, CallableStatement et ResultSet
- Emettre des requêtes SQL et lire les résultats de la base de données : parcours des résultats, fermeture des résultats, des états et des connexions
- Gestion des exceptions SQLException
- Gestion des transactions avec l'API JDBC

