



Formation Java 11 -Certified Professional Java SE 11 Developer – Niveau 2

Code : DJ02

Durée : 4 jours

Classe : Présentiel / à distance

Public

- Développeurs Java souhaitant passer la certification Java 11 (1Z0-819)
- Professionnels ayant déjà un niveau intermédiaire (niveau 1 requis)
- Ingénieurs logiciels souhaitant valider leurs compétences avancées

Prérequis

- Bonnes bases en programmation Java
- Maîtrise des concepts fondamentaux Java SE
- Avoir suivi le niveau 1 ou équivalent

Objectifs

- Se préparer efficacement à l'examen 1Z0-819
- Maîtriser les concepts avancés de Java 11
- Optimiser ses stratégies de passage de certification
- Renforcer ses compétences en développement professionnel

Programme détaillé

1- Présentation de la certification Oracle Certified Professional – Java SE 11 Developer

- Présentation des différentes certifications Java d'Oracle
- Déroulement du processus d'inscription
- Déroulement de l'épreuve
- Recommandations pour le passage de l'examen
- Stratégies gagnantes
- Thèmes de l'examen 1Z0-819

2- Nouveautés dans les interfaces

- Méthodes par défaut et méthodes static
- Méthodes privées

Atelier

Passage d'un QCM-bilan

3- Annotations

- Format d'une annotation
- Annotations prédéfinies de Java 11
- Création d'annotations
- Meta-annotations
- Annotations de type
- Annotations répétées

Atelier

Passage d'un QCM-bilan

4- Conception avancée de classes

- Méthodes par défaut, privées et méthodes static dans les interfaces
- Utiliser les classes et les méthodes abstraites
- Développer du code qui utilise le mot-clé final
- Créer des classes emboîtées static, les classes locales, classes internes et anonymes
- Utiliser les types énumérés, y compris ceux comportant des méthodes et des constructeurs

Atelier

Passage d'un QCM-bilan





Formation Java 11 -Certified Professional Java SE 11 Developer – Niveau 2

Code : DJ02

Durée : 4 jours

Classe : Présentiel / à distance

5- Collections et généricité

- Créer et utiliser une classe générique
- Créer et utiliser une méthode générique
- Utilisation du caractère ? dans un paramètre générique
- Présentation des interfaces List, Set, Queue, 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

Atelier

Passage d'un QCM-bilan

6- Programmation fonctionnelle

- Mise en œuvre des interfaces fonctionnelles prédéfinies : Consumer, Supplier, Predicate, Function, UnaryOperator, BinaryOperator
- Variantes spécialisées pour types primitifs des interfaces fonctionnelles prédéfinies
- 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
- 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
- Utiliser les streams parallèles pour améliorer les performances

Atelier

Passage d'un QCM-bilan

7- Date/temps-Internationalisation

- Utiliser les méthodes des classes LocalDate, LocalTime, LocalDateTime, Instant, Period, Duration
- Formater les dates et le temps
- Gérer les zones géographiques
- Calculer les durées avec les classes Period et Duration
- Compatibilité des nouvelles classes avec les classes java.util.Date et java.util.Calendar
- Obtention d'un objet Locale
- Localisation de dates, nombres
- Gestion des Resource Bundles

Atelier

Passage d'un QCM-bilan





Formation Java 11 -Certified Professional Java SE 11 Developer – Niveau 2

Code : DJ02

Durée : 4 jours

Classe : Présentiel / à distance

8- Entrées/sorties de base

- 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
- Mettre en œuvre la sérialisation d'objet avec ObjectOutputStream,
- ObjectInputStream
- Streams sur fichiers

9- Entrées/sorties sur fichiers (NIO.2)

- 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

10- 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

Atelier

Passage d'un QCM-bilan

11- Accès aux bases de données avec JDBC

- Rôle d'un driver JDBC
- Utiliser la classe DriverManager pour se connecter à une base de données

