

Formation Certification Java 11

niveau 2 - Certified Professional Java

SE 11 Developer

Description de la formation Java SE 11 Developer

Cette **formation Java SE 11 Developer** constitue la deuxième étape afin de vous préparer à la certification officielle Oracle Certified Professional - Java SE 11 Developer (1Z0-819). Le cours convient également pour préparer l'examen 1Z0-817, destiné uniquement aux candidats qui possèdent déjà une certification OCP Java SE 6, 7 ou 8.

Cette certification étant réputée difficile, sa préparation a été décomposée en deux formations distinctes :

- Le niveau 1 concerne les développeurs qui ont de bonnes bases en Java
- Le niveau 2 concerne les développeurs qui auront suivi la préparation niveau 1 ou qui ont déjà obtenu une certification, soit OCA java 8, soit OCP en Java 6, Java 7 ou Java 8

Objectifs

Objectif opérationnel :

Préparer et réussir la certification Java 11 SE Developer.

Objectifs pédagogiques :

À l'issue de cette **formation Java SE 11 Developer**, vous saurez :

- Effectuer un choix stratégique vous évitant de perdre du temps inutilement lors de l'examen
- Déterminer les sujets qui nécessiteront un travail de préparation complémentaire
- Augmenter vos chances de réussir l'examen de certification 1Z0-817 ou 1Z0-819

À qui s'adresse cette formation ?

Public :

Cette formation Java SE 11 Developer s'adresse aux développeurs qui souhaitent se préparer au passage de l'un des deux examens officiels (1Z0-817 ou 1Z0-819) d'Oracle, permettant d'obtenir la certification Oracle Certified Professional, Java SE 11 Developer.

Prérequis :

Pour tirer pleinement profit de cette formation de préparation à l'un des deux examens de certification, réputés difficiles, il est nécessaire d'avoir au préalable d'excellentes bases en programmation Java ainsi qu'une bonne pratique.

PLB propose un QCM permettant au participant de déterminer s'il peut suivre avec profit cette formation.

Contenu du cours Java SE 11 Developer

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 1Z0-817

Thèmes de l'examen 1Z0-819

Révisions

Nouveautés dans les interfaces Java 8+

Méthodes par défaut et méthodes static

Méthodes privées

Atelier

Passage d'un QCM-bilan

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

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



ITgate

Training

Your Gateway to Excellence

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 Array List, TreeSet, Tree Map et Array Deque

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

Programmation fonctionnelle

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

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

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

Atelier

Passage d'un QCM-bilan

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`



ITgate

Training

Your Gateway to Excellence

Gérer les attributs d'un fichier ou répertoire et autres méta-données

Utiliser l'API Stream sur fichiers et répertoires

Atelier

Passage d'un QCM-bilan

Les accès concurrents en multithreading

Créer des threads avec Runnable, Callable

Utiliser un Executor Service pour exécuter les tâches

Identifier les problèmes potentiels du multithreading 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

Accès aux bases de données avec JDBC

Rôle d'un driver JDBC

Utiliser la classe Driver Manager pour se connecter à une base de données

Mettre en oeuvre les interfaces Connection, Statement, Prepared Statement, Callable Statement et Result Set

Émettre 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 SQL Exception

Atelier

Passage d'un QCM-bilan

Mise en œuvre des modules

Modules nommés, anonymes et automatiques

Directives

Mise en œuvre de modules

Gestion des dépendances entre modules

Rôle du module path

Mise en œuvre de services

Nouvelle organisation de la documentation des API Java

Utiliser la commande jdeps

Migration d'une application non modulaire vers une application modulaire

Atelier

Passage d'un QCM-bilan

Internationalisation

Lire et configurer la langue/le pays avec un objet Locale

Créer et lire un fichier Properties

Créer un fichier Properties par langue et le charger dans l'application via un ressource bundle

Formatage des nombres et des dates

Atelier

Passage d'un QCM-bilan

Sécurité

Conception sûre d'une classe

Validation des saisies

Protection des données sensibles

Sérialisation d'objets

Prévention des attaques par déni de service

Lecture attentive de « Secure Coding Guidelines for Java SE » fourni par Oracle

Atelier

Passage d'un QCM-bilan

Travaux Pratiques

Les travaux pratiques consistent essentiellement à passer des tests blancs en anglais. Les tests sont ensuite corrigés afin de mesurer la progression de chacun et d'insister encore sur les points où des erreurs ont été commises.