



Java SE 17 Developer – Partie 1

Code : NDJ21

Durée : 3 jours

Classe : Présentiel / à distance

Public

- Cette formation s'adresse à toute personne souhaitant devenir Java SE 17 Developer certifié et se préparer efficacement à l'examen Oracle 1Z0-829.

Prérequis

- Connaissances de base en programmation et notions fondamentales de Java SE 17, avec une pratique exigée ou une expérience équivalente.

Objectifs

- Faire des choix stratégiques pour gagner du temps lors de l'examen
- Identifier les sujets nécessitant un travail de préparation complémentaire
- Augmenter vos chances de réussite à l'examen de certification
- Appliquer les concepts avancés de Java SE 17 dans des exercices pratiques et des projets réels

Programme détaillé

1- Construction de blocs

- Différentes catégories de variables, portée et cycle de vie
- Type primitifs, conversions implicites et explicites (cast)
- Structure d'une classe : attributs, constructeurs, méthodes
- Accès aux attributs d'un objet
- Portée et cycle de vie d'un objet
- Création d'objets
- Compiler/exécuter un programme Java en ligne de commande Importer des classes d'autres packages
- Comparer et distinguer les caractéristiques de Java comme: indépendance de la plateforme, orienté objet, l'encapsulation, etc

2- Opérateurs

- Utiliser les opérateurs Java, priorité des opérateurs
- Opérateurs unaires, binaires, ternaire
- Opérateurs arithmétiques, sur bits, de comparaison, parenthèses, d'affectation
- Comparaison des objets en utilisant == et la méthode equals
- Affectations simples et calculée

3- Décisions

- Utiliser les branchements avec if, if/else et l'opérateur ternaire
- Utiliser l'instruction switch dans ses différentes syntaxes, notamment yield
- Utiliser l'opérateur instanceof et le Pattern Matching
- Créer et utiliser les boucles while
- Créer et utiliser les boucles for y compris la boucle for-each
- Créer et utiliser les boucles do-while
- Utiliser les instructions break et continue





Java SE 17 Developer – Partie 1

Code : NDJ21

Durée : 3 jours

Classe : Présentiel / à distance

4- API essentielles

- Manipuler les chaînes de caractères avec les classes String et StringBuilder
- Manipuler les blocs de texte
- Déclarer, instancier et manipuler des tableaux à une dimension
- Déclarer, instancier et manipuler des tableaux à plusieurs dimensions
- Les méthodes de la classe Math
- Utiliser les méthodes des classes LocalDate, LocalTime, LocalDateTime, Instant, TimeZone, 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

5- Méthodes

- Conception de méthodes
- Déclaration de variables locales et d'instance
- Utilisation de paramètres varargs
- Modificateurs d'accès
- Accès aux membres static
- Passage d'arguments aux méthodes
- Surcharge de méthodes

6- Conception de classes

- Appliquer aux classes les principes de l'encapsulation
- Distinguer le constructeur par défaut des constructeurs définis par l'utilisateur
- Appliquer les modificateurs d'accès
- Créer des méthodes avec arguments et valeur de retour
- Cycle de vie des variables locales
- Surcharger des méthodes, notamment les constructeurs
- Maîtriser le passage d'arguments aux méthodes par valeur et par référence
- Appliquer le mot-clé static aux attributs et aux méthodes
- Les blocs d'initialisation, les blocs static
- Méthodes par défaut, privées et méthodes static dans les interfaces
- Utiliser des classes et des objets immutable
- Utiliser les classes et les méthodes abstraites
- Développer du code qui utilise le mot-clé final
- Décrire le principe de l'héritage et ses bénéfices
- Utilisation de l'opérateur instanceof
- Utiliser les mot-clés this et super pour l'accès aux objets et aux constructeurs
- Hériter de classes abstraites





Java SE 17 Developer – Partie 1

Code : NDJ21

Durée : 3 jours

Classe : Présentiel / à distance

7- Au-delà des classes

- Créer des classes emboîtées static, les classes locales, classes internes et anonymes
- Implémenter des interfaces
- Utiliser les types énumérés, y compris ceux comportant des méthodes et des constructeurs
- Créer/utiliser des records
- Créer/utiliser des classes ou interfaces scellées
- Décrire le principe de l'héritage et ses bénéfices
- Mettre en œuvre le polymorphisme, redéfinir des méthodes
- Savoir déterminer quand un cast est nécessaire
- Utilisation de l'opérateur instanceof
- Hériter de classes abstraites et implémenter des interfaces

