



---

## Développeur Java certifié Oracle OCA, Java SE 7 Programmer

Développeur Java certifié Oracle Certified Associate, Java SE 7 Programmer

durée : 5 jour(s)

code formation : JA1

---

### Description :

En 5 jours, cette formation vous prépare à la certification 1Z0-803 Java SE 7 Programmer qui donne le titre de Oracle Certified Associate, Java SE 7 Programmer

### Pré-requis :

Savoir utiliser Windows XP

Connaître les principes fondamentaux de la programmation orientée objet

Connaître les principes de l'algorithmie

Avoir déjà réalisé un programme ou une application (logiciel, outil, site internet...), quel que soit le langage.

---

### Programme :

#### Culture générale Java

| L'introduction au langage Java

| L'origine du langage et de la plate-forme

| Les nouveaux concepts

| La notion de JVM

| Les différentes versions de Java

| Exercice : Comparaison avec les autres langages

| Les applications Java

| Les Applets et les applications Internet

| Les applications autonomes

| La comparaison et les éléments de choix

| Les bases du langage Java

| La structure d'un programme Java

| Le modèle mémoire et le garbage collector

| Les types primitifs et les tableaux

| Les structures de contrôle

| Les conventions de codage

| La structure syntaxique d'une application Java

| Exemple de syntaxe sur une application de démonstration simple

| La réalisation de votre première application simple

| Les types

| Les types de base et les types objet

| Les Expressions

| La déclaration de variables et de constantes

- | L'utilisation des opérateurs avec les objets
- | L'utilisation des tableaux
- | La conversion de types
- | Les instructions de contrôle : if, for, while , return et break.
- | La réalisation d'un programme minimum
- | Les conventions d'écriture

## **Les concepts objets obligatoires pour Java**

- | La présentation et la définition d'une classe
- | Les attributs et les méthodes
- | L'encapsulation
- | Le polymorphisme
- | Les méthodes :
- | L'invocation des méthodes
- | L'exécution des méthodes de classes et des méthodes d'instances
- | La surcharge des méthodes
- | L'abstraction :
- | L'utilisation d'un objet de la classe Date
- | L'instanciation des objets
- | L'utilisation de la classe String
- | L'utilisation de la classe StringBuffer
- | L'héritage
- | La présentation du concept d'héritage
- | L'utilisation de l'héritage
- | La classe Object et la généricité
- | L'utilisation du polymorphisme
- | La spécialisation d'une référence polymorphe
- | Le typage des référence et des objets
- | Le comportement des méthodes et du typage
- | La généricité des classes conteneurs : exemple de la classe Vector
- | Les interfaces
- | La terminologie et la syntaxe
- | L'extension et l'implémentation
- | L'utilisation des interfaces pour les méthodes
- | L'intérêt du mécanisme d'interface pour les méthodes
- | Les interfaces et constantes

## **La gestion des erreurs avec les exceptions**

- | Le principe des exceptions
- | Le mode de détection
- | La notification d'une situation d'exception
- | Le report d'une exception : clause throws

- | Le report d'une exception : bloc try/catch
- | L'utilisation des exceptions pré-définies
- | Le flux d'entrée/sortie

## **Le développement de classes**

- | Le squelette d'une classe
- | L'organisation en packages
- | Les contraintes liées aux packages
- | L'écriture des constructeurs
- | Le constructeur par défaut
- | L'auto-référence this
- | Les champs et les méthodes statiques
- | La méthode main
- | Le développement d'interfaces
- | Les rappels et compléments sur les principes
- | La syntaxe associée aux interfaces
- | La définition d'interfaces pour les méthodes
- | L'implémentation et les extensions multiples d'interfaces
- | L'implémentation partielle d'interface
- | Les développement de classes dérivées
- | Les rappels sur les classes dérivées
- | Les méthodes abstraites
- | Les classes abstraites et les interfaces
- | Le droit d'accès aux champs et l'héritage
- | Les constructeurs et l'héritage
- | La redéfinition et la surcharge

## **Les concepts objets avancés**

- | Les classes et méthodes abstraites en Java
- | Les classes imbriquées et classes anonymes
- | La présentation des paquetages (packages)