

## Formation Préparation de la certification Java SE 8 Programmer II (1Z0-809)

|  |  |
|--|--|
| ■ <b>Durée :</b>   | 2 jours (14 heures)  |
| ■ <b>Tarifs inter-entreprise :</b>                             | 1 980,00 CHF HT (standard)<br>1 584,00 CHF HT (remisé)   |
| ■ <b>Public :</b>  | Développeurs et analystes programmeurs Java  |
| ■ <b>Pré-requis :</b>  | Connaissances en programmation orientée objet Java   |
| ■ <b>Objectifs :</b>   | Découvrir le cursus de certification Oracle Java ' Préparer la certification Java SE 8 Programmer I : comprendre le type et le format des questions, s'entraîner à répondre à des questions, révision des thèmes abordés dans l'examen.  |
| ■ <b>Modalités pédagogiques, techniques et d'encadrement :</b> | <ul style="list-style-type: none"><li>• Formation synchrone en présentiel et distanciel.</li><li>• Méthodologie basée sur l'Active Learning : 75 % de pratique minimum.</li><li>• Un PC par participant en présentiel, possibilité de mettre à disposition en bureau à distance un PC et l'environnement adéquat.</li><li>• Un formateur expert.</li></ul>                                       |
| ■ <b>Modalités d'évaluation :</b>                              | <ul style="list-style-type: none"><li>• Définition des besoins et attentes des apprenants en amont de la formation.</li><li>• Auto-positionnement à l'entrée et la sortie de la formation.</li><li>• Suivi continu par les formateurs durant les ateliers pratiques.</li><li>• Évaluation à chaud de l'adéquation au besoin professionnel des apprenants le dernier jour de formation.</li></ul> |
| ■ <b>Sanction :</b>  | Attestation de fin de formation mentionnant le résultat des acquis   |
| ■ <b>Référence :</b>   | JAV1107-F  |
| ■ <b>Note de satisfaction des participants:</b>                | Pas de données disponibles   |
| ■ <b>Contacts :</b>  | commercial@dawan.fr - 09 72 37 73 73   |

|                              |   |
|------------------------------|---|
| ■ <b>Modalités d'accès :</b> | Possibilité de faire un devis en ligne ( <a href="http://www.dawan.fr">www.dawan.fr</a> , <a href="http://moncompteformation.gouv.fr">moncompteformation.gouv.fr</a> , <a href="http://maformation.fr">maformation.fr</a> , etc.) ou en appelant au standard. |
| ■ <b>Délais d'accès :</b>    | Variable selon le type de financement.  |
| ■ <b>Accessibilité :</b>     | Si vous êtes en situation de handicap, nous sommes en mesure de vous accueillir, n'hésitez pas à nous contacter à <a href="mailto:referenthandicap@dawan.fr">referenthandicap@dawan.fr</a> , nous étudierons ensemble vos besoins                             |

## Présentation de la certification

Cursus de certification Oracle

Java SE 8 Programmer : type d'examen, nombre de questions/durée, % minimal de réussite

Plateforme de certification

Considérations lors de l'examen : missing packages, import, fichiers, chemins, fragments de code, commentaires

**Atelier :** présentation de l'examen et des sujets couverts par le questionnaire

## Révision du contenu de la certification

- **Implémentation de classes :** encapsulation, héritage, polymorphisme, redéfinition des méthodes `hashCode/equals/toString`, codage d'un singleton, blocs static.

- **Notions avancées de l'objet :** abstraction, mot-clé `final`, classes internes, `static`, anonymes, types `enum`, implémentation/héritage d'interfaces, création et utilisation d'expressions `lambda`.

- **Généricité et collections :** création et utilisation de classes génériques, collections génériques (`ArrayList`, `TreeSet`, `TreeMap`, `ArrayDeque`), Utilisation de comparateurs `Comparable/Comparator`, `Streams` et filtres de collections, itération de `streams` et `lists`, interface `Stream`, filtres de collections avec des expressions `lambda`, références de méthodes avec `Streams`.

- **Interfaces fonctionnelles :** package `java.util.function`, implémentation d'interfaces fonctionnelles (`primitive`, `binary`, `unary`).

- **Java Stream API** : extraction de données (peek, map), recherches (findFirst, findAny, anyMatch, allMatch, noneMatch), classes optionnelles, tri de collections avec Stream API, méthodes de collecte de résultats, usage de flatMap().

- **Exceptions et assertions** : utilisation du try/catch et du throw, multi-catch/finally, AutoClose (try-with-resources), création d'exceptions et d'auto-closeable resources, utilisation d'assertions.

- **API Time** : Gestion des dates/heures : LocalDate, LocalTime, Instant, Period et Duration, usage des timezones et formatage de dates, TemporalUnit.

- **Java IO et NIO2** : Lecture et écriture depuis la console, utilisation du package java.io, nouveautés du package nio : Path, Files, Stream API avec NIO.2

- **Concurrence en Java** : Implémentation de threads (Runnable, Callable, ExecutorService), identification de problèmes (deadlock, starvation, livelock, race conditions), contrôle et synchronisation (synchronized, package java.util.concurrent.atomic), collections concurrentes (java.util.concurrent), parallel Fork/join, parallel Streams (reduction, decomposition, merging processes, pipelines).

- **Accès aux bases de données avec JDBC** : Interfaces de JDBC (Driver, Connection, Statement, ResultSet) et implémentations, composants nécessaires de connexions, écriture de requêtes et traitement de résultats.

- **Localisation** : object Local, manipulation de fichiers properties, création de ressources bundles et chargement.

**Passage et correction d'un test blanc**

**Ateliers corrigés**

**Questions/réponses, gestion du temps lors du passage de l'examen**