

Développer en langage Java

Référentiel mars 2024



Editions ENI

Référentiel de compétences

Description du métier, de l'activité ou de la situation professionnelle à partir duquel le dispositif visant la certification est initié :

Le langage Java est un langage de programmation informatique de référence créé en 1995 qui est un des plus utilisés dans le monde.

Il est très plébiscité par les entreprises qui ont développé (et développent encore) de nombreuses applications en Java ces 20 dernières années et il existe un fort besoin de développeurs Java sur le marché pour maintenir des applications existantes et développer de nouveaux projets. Les entreprises ne cessent de recruter des développeurs spécialistes de ce langage de programmation.

Notre certification vise donc aussi bien de jeunes diplômés d'une formation initiale en développement informatique que des développeurs expérimentés disposant d'une expérience passée avec le langage Java et qui souhaitent ajouter le développement Java comme compétence complémentaire à leur métier et ainsi faire reconnaître leurs compétences par l'obtention d'une certification.

En fonction du score obtenu à l'épreuve, les candidats valident la certification :

- **De 0 à 699/1000** : le candidat n'est pas certifié
- **De 700 à 1000/1000** : le candidat est certifié

En dessous de 700 points, la certification n'est pas validée.

Développer en langage Java

Référentiel mars 2024

Référentiels		
Référentiel de compétences	Référentiel de certification	
	Modalité(s)	Critères
C1 - Ecrire du code Java en utilisant les éléments fondamentaux d'algorithmie du langage pour créer une application qui s'exécute en ligne de commande.	<p>1 – 5 études de cas (écriture de code en langage Java) pour un total de 50 points</p> <p>2 – 7 questions de type QCM/QCU à visée professionnelle pour un total de 7 points</p>	<p>Cr1 - Les outils du JDK et/ou du JRE sont utilisés : le compilateur Java et la JVM sont invoqués en ligne de commandes</p> <p>Cr2 - Un programme Java simple est écrit (création d'une classe avec ajout de la méthode main()) pour lequel les conventions de nommage sont respectées et des commentaires ajoutés au code</p> <p>Cr3 - La documentation technique des projets Java est générée</p> <p>Cr4 - Les variables et constantes sont correctement déclarées et initialisées et le transtypage d'une donnée effectué</p> <p>Cr5 - Les opérateurs du langage (arithmétiques, d'incrémentement et de décrémentement, d'affectation, de comparaison et logiques) sont utilisés</p> <p>Cr6 - Les chaînes de caractères sont déclarées et initialisées (avec une variable de type String), exploitées (méthodes substring(), length(), getBytes() et split()), comparées (méthodes equals() et compareTo()), et formatées (méthode format()).</p> <p>Cr7 - Des instructions selon une ou plusieurs conditions (if, else if, else...) sont exécutées</p> <p>Cr8 - Des ensembles de données sont itérés (structure for, boucles while et do while, mots-clé break et continue)</p>

Développer en langage Java

Référentiel mars 2024

		Cr9 - Un tableau multidimensionnel est déclaré et initialisé (opérateur new) avec des valeurs initiales et manipulé
C2 - Développer une application en utilisant les mécanismes de programmation orientée objet de Java pour maîtriser la complexité d'un programme.	<p>1 – 6 études de cas (écriture de code en langage Java) pour un total de 60 points</p> <p>2 – 6 questions de type QCM/QCU à visée professionnelle pour un total de 6 points</p>	<p>Cr1 - Une classe est créée avec le mot-clé class et déclarée dans un fichier d'extension .java portant le même nom que la classe</p> <p>Cr2 - Les attributs sont correctement déclarés dans le corps de la classe et typés</p> <p>Cr3 - Les variables d'instances, les variables de classes et les variables locales sont différenciées</p> <p>Cr4 - Des constructeurs sont créés et implémentés</p> <p>Cr5 - Des méthodes sont créées et implémentées</p> <p>Cr6 - Un objet est instancié (opérateur new) et manipulé</p> <p>Cr7 - L'encapsulation est mise en oeuvre</p> <p>Cr8 - L'héritage est mis en oeuvre</p> <p>Cr9 - Le polymorphisme est mis en oeuvre</p>
C3 - Mettre en oeuvre les concepts avancés de programmation orientée objet en exploitant les classes abstraites, les interfaces, les exceptions et la programmation fonctionnelle pour améliorer la robustesse d'une application.	<p>1 - 6 études de cas (écriture de code en langage Java) pour un total de 60 points</p> <p>2 – 6 questions de type QCM/QCU à visée professionnelle pour un total de 6 points</p>	<p>Cr1 - Les classes sont organisées dans les packages</p> <p>Cr2 - Les modules sont utilisés pour organiser l'application</p> <p>Cr3 - Des énumérations sont déclarées avec le mot-clé enum et enrichies</p> <p>Cr4 - Des classes abstraites sont déclarées avec le mot-clé abstract et utilisées</p> <p>Cr5 - Des interfaces sont déclarées avec le mot-clé interface et utilisées</p> <p>Cr6 - La programmation fonctionnelle avec les expressions lambda est mise en oeuvre</p> <p>Cr7 - La méthode equals() est redéfinie dans les classes d'objets qui doivent être comparés et la méthode hashCode() est redéfinie correctement sur la base de la méthode equals()</p>

Développer en langage Java

Référentiel mars 2024

		<p>Cr8 - Des exceptions dans des méthodes sont déclenchées</p> <p>Cr9 - Les exceptions sont gérées localement (try-catch-finally, try-multicatch, try-with-resource)</p> <p>Cr10 - Des classes d'exceptions utilisateurs sont créées</p>
<p>C4 - Exploiter les fonctionnalités de la bibliothèque de classes Java en utilisant les collections, les streams et l'API de gestion de dates de Java pour améliorer la gestion et la manipulation des données.</p>	<p>1 – 6 études de cas (écriture de code en langage Java) pour un total de 60 points</p> <p>2 - 6 questions de type QCM/QCU à visée professionnelle pour un total de 6 points</p>	<p>Cr1 - Les collections sont utilisées</p> <p>Cr2 - Un Stream est créé à partir d'une collection puis transformé (méthode map()), filtré (méthode filter()) et des calculs sont réalisés sur les données filtrées et transformées</p> <p>Cr3 - Les données temporelles sont gérées avec l'API java.time</p>
<p>C5 - Connecter une application Java à une base de données relationnelle en utilisant les classes et interfaces de l'API JDBC pour implémenter des fonctionnalités de persistance de données.</p>	<p>1 – 3 études de cas (écriture de code en langage Java) pour un total de 30 points</p> <p>2 – 9 questions de type QCM/QCU à visée professionnelle pour un total de 9 points</p>	<p>Cr1 - La connexion d'une application Java à une base de données est réalisée</p> <p>Cr2 - Une requête SQL est définie dans le code de l'application puis exécutée</p> <p>Cr3 - Une requête SQL contenant des paramètres est définie dans le code de l'application puis exécutée</p> <p>Cr4 - Les résultats des requêtes SQL SELECT sont traités avec l'objet ResultSet et les méthodes next(), first(), last(), relative() et absolute().</p> <p>Cr5 - Les ressources JDBC allouées sont correctement libérées</p> <p>Cr6 - Les exceptions de type SQLException sont gérées localement et des exceptions applicatives sont déclenchées</p> <p>Cr7 - Les transactions sont mises en œuvre (méthodes commit(), rollback())</p>