

Formation & Certification • ITS Java

IT Specialist – Java

Formation alignée sur les objectifs officiels de la certification IT Specialist (ITS) – Java. Les participants apprennent à écrire, compiler, exécuter, documenter et déboguer des applications Java bien structurées.

Distributeur officiel Certiport

Centre d'examen Certiport

Learn • Practice • Certify

Durée 17 h	Examen ITS – Java
Modalité Distanciel	Niveau Fondamental / Intermédiaire

INSCRIPTION / RÉSERVATION



Je m'inscris
maintenant

- **Learn** : Parcours complet couvrant tous les objectifs de l'examen ITS Java, de la POO au débogage.
- **Practice** : accès à un simulateur d'examen interactif pour s'entraîner dans des conditions proches de l'examen.
- **Certify** voucher de certification ITS – Java pour passer l'examen dans un centre Certiport.

OBJECTIFS PÉDAGOGIQUES

- Comprendre les bases du langage Java et la structure d'une application.
- Maîtriser les types, variables, expressions et contrôles.
- Appliquer la programmation orientée objet avec classes et objets.
- Gérer tableaux, listes, chaînes, exceptions, compilation et débogage.

PUBLIC CIBLE

- Développeurs débutants travaillant avec Java SE 6 ou version ultérieure.
- Étudiants du secondaire ou post-secondaire en développement logiciel.
- Professionnels en reconversion souhaitant valider une première certification Java.

PRÉREQUIS

- Niveau de lecture au moins équivalent à la 8e année.
- Bonnes bases en mathématiques (niveau Algèbre II).
- Compétences de base sur les systèmes d'exploitation et expérience avec la ligne de commande.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Fondamentaux de Java (Java Fundamentals)

Décrire l'utilisation de main dans une application Java

- Signature de la méthode main et rôle du point d'entrée d'une application Java.
- Consommer une instance de sa propre classe à partir de main.
- Utiliser les arguments de la ligne de commande.

Effectuer des entrées/sorties de base avec les packages standards

- Utiliser les instructions d'affichage (print/println).
- Importer et utiliser la classe Scanner pour lire les entrées utilisateur.

Évaluer la portée d'une variable

- Déclarer des variables au niveau d'un bloc, d'une classe ou d'une méthode.
- Comprendre la durée de vie et la visibilité des variables.

Commenter et documenter les programmes

- Écrire des commentaires de ligne et de bloc en Java.
- Comprendre la syntaxe des Javadocs et produire une documentation correcte.

Types de données, variables et expressions

Déclarer et utiliser des variables de types primitifs

- Types primitifs : byte, char, int, double, short, long, float, boolean.
- Identifier les cas de perte de précision lors des conversions.
- Initialisation des variables et différences entre types primitifs et classes enveloppes (Integer, Boolean, etc.).

Construire et évaluer du code manipulant les chaînes de caractères

- Utiliser la classe String et les littéraux de chaîne.
- Comparaison, concaténation, changement de casse, longueur, méthodes String.format.
- Comprendre la nature immuable des chaînes, l'initialisation et la valeur null.

Construire et évaluer du code utilisant des tableaux et des listes

- Tableaux à une ou deux dimensions : initialisation, taille, itération et accès aux éléments.
- Listes de type ArrayList : ajout, suppression et parcours des éléments.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Effectuer des opérations de parsing, casting et conversion

- Convertir entre types primitifs (cast).
- Convertir les types primitifs vers les types objets équivalents (autoboxing).
- Analyser des chaînes de caractères pour obtenir des nombres et convertir des valeurs en chaînes.

Construire et évaluer des expressions arithmétiques

- Utiliser les opérateurs arithmétiques de base.
- Utiliser l'affectation simple et les opérateurs d'affectation composée.
- Comprendre la priorité des opérateurs.

Implémentation du contrôle de flux

Construire et évaluer du code utilisant des instructions de branchement

- Utiliser les instructions if, else, else if et switch.
- Faire la différence entre instructions sur une ligne et blocs de code.
- Imbriquer des structures de décision.
- Utiliser les opérateurs logiques et relationnels.

Construire et évaluer du code utilisant des boucles

- Utiliser les boucles while, for, for-each et do-while.
- Utiliser les instructions break et continue.
- Imbriquer des boucles et utiliser les opérateurs logiques, relationnels et unaires.

Programmation orientée objet (OOP)

Construire et évaluer des définitions de classes

- Définir des constructeurs et pratiquer la surcharge de constructeurs.
- Respecter la convention « une classe par fichier .java ».
- Utiliser le mot-clé this.
- Comprendre les notions d'héritage et de redéfinition (override) de méthodes.

Déclarer, implémenter et accéder aux membres de données

- Utiliser les modificateurs d'accès private, public, protected.
- Déclarer des membres d'instance et des membres static.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

- Utiliser static final pour créer des constantes.
- Expliquer le principe d'encapsulation.

Déclarer, implémenter et accéder aux méthodes

- Utiliser les modificateurs d'accès pour les méthodes.
- Définir les paramètres, le type de retour et les méthodes void.
- Mettre en œuvre des méthodes d'instance et des méthodes static.
- Pratiquer la surcharge de méthodes.

Instancier et utiliser des objets dans les programmes

- Instancier des objets et gérer leur cycle de vie (initialisation, référence null).
- Accéder et modifier les membres de données.
- Appeler les méthodes d'instance et static.
- Importer des packages et classes pour organiser son code.

Compilation et débogage du code

Résoudre les erreurs de syntaxe, de logique et d'exécution

- Utiliser des instructions d'affichage pour le débogage.
- Interpréter les messages du compilateur javac.
- Identifier les erreurs de logique.
- Lire les exceptions et la pile d'appels (stack trace) dans la console.

Mettre en œuvre la gestion des exceptions

- Utiliser les blocs try, catch et finally.
- Comprendre la classe Exception et les types d'exceptions.
- Afficher et journaliser les informations relatives aux exceptions.