

Initiation à la programmation Objet/Java

Cours Pratique de 3 jours - 21h

Réf : IPJ - Prix 2025 : 1 870 HT

Grace aux notions de classe et d'exception, la programmation objet facilite la réutilisation du code et sa maintenance. Par ailleurs, un bon usage des interfaces permet d'accorder la MOA et la MOE. Par la réalisation de nombreux programmes Java, ce cours détaille tous les principes fondamentaux du développement objet.

OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Comprendre les principes fondateurs de l'Objet

Appréhender la syntaxe du langage Java

Maîtriser les échanges techniques avec des équipes de développement

Maîtriser la construction de spécifications fonctionnelles de type Objet

LE PROGRAMME

dernière mise à jour : 07/2021

1) Présentation générale

- Principes fondateurs de l'Objet : abstraction/encapsulation. Héritage, mise en œuvre.
- Présentation générale : le langage, les outils, la bibliothèque.
- Distributions de Java.

2) Aspects syntaxiques, types et expressions

- Structuration syntaxique d'une application Java.
- Exemple de syntaxe sur une application simplifiée.
- Vue externe d'une classe : syntaxe d'utilisation.
- Vue interne d'une classe : syntaxe d'implémentation.
- Notion de type. Utilisation comparée des types de base et des types Objet.
- Utilisation simple des types de base : les nombres entiers, les flottants, les types Char et Boolean.
- Notion d'expression.
- Exemples de déclarations : variables et constantes.
- Désignation comparée des types de base et des types Objet.
- Utilisation des opérateurs avec les objets.
- Cas des champs static ou variables de classes.
- Complément sur les types : utilisation de base des tableaux.
- Conversion types de base/type Objet.
- Conventions d'écriture.

3) Méthodes et instructions

- Syntaxe d'invocation des méthodes.
- Méthodes de classes et méthodes d'instances.
- Définition et utilisation des méthodes.
- La surcharge des méthodes.
- Notion de sous-bloc.
- Catégories d'instructions.
- Principales instructions de contrôle : if, while, for, return, break.

PARTICIPANTS

Chefs de projets voulant acquérir une vision précise des techniques employées par leurs collaborateurs ou sous-traitants. Développeurs traditionnels voulant évoluer vers l'objet.

PRÉREQUIS

Connaissances de base en programmation.

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...
Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

4) Utilisation de l'abstraction

- Exemple simple d'utilisation d'un objet : déclaration, instanciation ou fabrication, délégation.
- Utilisation des constructeurs d'objets : découverte de la documentation en ligne.
- Utilisation de l'interface programmatique des objets : exemple de la classe Date.
- Une classe très utilisée : la classe String.
- Particularités liées aux chaînes de caractères.
- Utilisation de la classe StringBuffer : exemple d'utilisation de la surcharge de méthodes.

5) Utilisation de l'héritage

- Rappel du principe d'héritage et terminologie.
- Utilisation de l'héritage.
- Exemple de graphe d'héritage.
- La classe Object et la généricité.
- Utilisation du polymorphisme.
- Spécialisation d'une référence polymorphe.
- Typage des références/typage des objets.
- Comportement des méthodes et typage.
- Généricité des classes conteneurs : exemple de la classe Vector.
- Les ajouts de JAVA 5 (TIGER) : les generics.

6) Utilisation du mécanisme d'interface

- Interface implicite et explicite d'une classe.
- Syntaxe associée aux interfaces explicites.
- Cas d'utilisation des références d'interfaces : flexibilité, limitation de la portée, polymorphisme.
- Exemple d'implémentation multiple d'interfaces.
- Synthèse sur l'intérêt des interfaces pour les méthodes.
- Utilisation des interfaces pour les constantes.
- Exemples avancés d'utilisation d'interfaces.

7) Développement de classes

- Approche méthodologique, analyse statique, dynamique, métier.
- Notation UML : diagramme de classe, d'état/transition, de séquence.
- Squelette d'une classe : constituants de base, outils de génération automatique.
- Compléments sur les droits d'accès.
- Organisation en packages.
- Contraintes liées aux packages.
- Ecriture des constructeurs.
- Constructeur par défaut.
- Compléments sur l'écriture des constructeurs.
- L'auto-référence "this".
- Champs et méthodes statiques.
- La méthode Main.

8) Développement d'interfaces

- Rappels et compléments sur les principes.
- Syntaxe associée aux interfaces, cas des constantes.
- Définition d'interfaces pour les méthodes.
- Implémentation et extensions multiples d'interfaces.
- Implémentation partielle d'interface.
- Exemples sur l'utilisation d'interfaces.

9) Développement de classes dérivées

- Rappels des principes.

- Approche méthodologique pour le découpage en classes.
- Méthodes et classes abstraites.
- Classes abstraites et interfaces.
- Droit d'accès aux champs et héritage.
- Enchaînement des constructeurs et héritage.
- Redéfinition et surcharge.

10) Les exceptions

- Principes et cinématique générale.
- Détection, constat et notification d'une situation exceptionnelle.
- Report d'une exception : Clause Throws de la signature, bloc Try/Catch.
- Exceptions non vérifiées.
- Exemple avec gestion d'exception.

LES DATES

CLASSE À DISTANCE

2025 : 09 avr., 12 mai, 18 juin, 09 juil., 18 août, 22 sept., 27 oct., 19 nov., 10 déc.

PARIS

2025 : 02 avr., 05 mai, 11 juin, 02 juil., 11 août, 15 sept., 20 oct., 12 nov., 03 déc.

LYON

2025 : 18 juin, 09 juil., 27 oct., 19 nov.

AIX-EN-PROVENCE

2025 : 12 mai, 22 sept., 10 déc.

BORDEAUX

2025 : 12 mai, 22 sept., 10 déc.

DIJON

2025 : 09 juil., 27 oct.

GRENOBLE

2025 : 09 juil., 27 oct.

LILLE

2025 : 12 mai, 22 sept., 10 déc.

MONTPELLIER

2025 : 09 juil., 27 oct.

ORLÉANS

2025 : 05 mai, 15 sept., 20 oct.

NANTES

2025 : 12 mai, 22 sept., 10 déc.

RENNES

2025 : 09 juil., 27 oct.

SOPHIA-ANTIPOLIS

2025 : 12 mai, 22 sept., 10 déc.

STRASBOURG

2025 : 12 mai, 22 sept., 10 déc.

TOULOUSE

2025 : 12 mai, 22 sept., 10 déc.

TOURS

2025 : 15 sept., 20 oct.

BRUXELLES

2025 : 12 mai, 18 août, 10 déc.

LUXEMBOURG

2025 : 12 mai, 18 août, 10 déc.