

Initiation à la programmation Objet/Java

-Référence: **SII-298**

-Durée: **3 Jours (21 Heures)**

Les objectifs de la formation

- À l'issue de la formation, les participants seront en mesure de :
 - Comprendre les principes fondamentaux de l'objet
 - Maîtriser la syntaxe du langage de programmation Java
 - Collaborer efficacement avec des équipes de développement techniques
 - Maîtriser la construction de spécifications fonctionnelles de type Objet

A qui s'adresse cette formation ?

POUR QUI :

- 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.

Programme

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

- **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.

- **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.

- **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.
- **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.
- **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.
- **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.

- **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.

- **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.

- **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.



(+212) 5 22 27 99 01



(+212) 6 60 10 42 56



Contact@skills-group.com

Nous sommes à votre disposition :
De Lun - Ven 09h00-18h00 et Sam 09H00 – 13H00

Angle bd Abdelmoumen et rue Soumaya, Résidence Shehrazade 3, 7ème étage N° 30
Casablanca 20340, Maroc