

Java Objet

Apprendre les concepts objets avec Java

Les langages de dernière génération dont Java et tant d'autres ont rendu la Programmation Orientée Objet incontournable. Développer une application Internet/Intranet, un service ou un composant dans un environnement distribué implique de réfléchir, concevoir et modéliser des « Objets ».

Pour l'informaticien habitué à aborder une problématique sous l'angle procédural, « passer à l'objet » est un changement radical qui se fera dans la durée.

Ce cours est pour nous le préalable nécessaire à toute formation aux nouvelles technologies des Systèmes d'Information utilisant le langage Java.

Détails

- Code : JA-OBJ
- Durée : 1 jour (7 heures)

Public

- Architectes
- Chefs de projets
- Consultants
- Développeurs
- Ingénieurs

Pré-requis

- Connaissance d'un langage de programmation

Objectifs

- Assimiler et mettre en oeuvre les concepts de la Programmation Orientée Objet
- Connaître les normes et standards, les langages de développement et de modélisation, les méthodes
- Savoir développer avec Java

Programme

Etat de l'art des technologies objet

- L'évolution des langages de programmation
- Comparaison entre l'approche fonctionnelle et l'approche objet
- Etat de l'art concernant les langages et technologies objet actuelles: C++, Java, C#, PHP

Introduction à Java

- La machine virtuelle
- Le classpath
- Les types de base et opérateurs
- Les structures de contrôle
- Les tableaux
- Les chaînes

Les concepts Objets de base

- Concepts de classe, d'instances, d'état et de comportement
- Messages, communication et collaboration entre les objets
- Encapsulation, attributs et méthodes
- Mise en oeuvre avec Java

Les concepts Objets avancés

- Abstraction, objets et classes
- Généralisation et spécialisation, classification, principe d'héritage

- Comprendre le polymorphisme
- Relations entre les classes, association, agrégation, composition et dépendance
- Réutilisation
- Mise en oeuvre avec Java

Concepts Java avancés

- Introspection et notion de méta modèle :
 - Les classes Class, Method
- Réflexion : invocation dynamique de code
- Les exceptions : principes, utilisation

Concepts architecturaux

- Programmation par contrat de service, notion d'interface
- Notions d'architecture et de composants
- Couplage des composants, organisation en couches applicatives d'un système

Modélisation Objet

- Représentation naturelle des entités métiers
- Approche systémique : organisation, responsabilités
- Les Design Patterns de conception, des solutions optimisées pour les problèmes récurrents
- Bonnes pratiques de la programmation orientée objet
- Frameworks : introduction, utilisation

Modalités

- **Type d'action** :Acquisition des connaissances
- **Moyens de la formation** :Formation présentielle – 1 poste par stagiaire – 1 vidéo projecteur – Support de cours fourni à chaque stagiaire
- **Modalités pédagogiques** :Exposés – Cas pratiques – Synthèse
- **Validation** :Exercices de validation – Attestation de stages