



Programme détaillé formation 'JAVA'- 3j

Objectifs : fournir connaissances théoriques et pratiques permettant d'intégrer un projet de développement JAVA

Audience : Développeurs, chefs de projets .

Prérequis : pratique d'un langage de programmation type Visual Basic, C/C++ .

Moyens pédagogiques : 1 ordinateur/stagiaire. Supports cours. Travaux pratiques. Vidéoprojecteur. Tests

Durée : 3 jours (total 21 h)

Vous allez apprendre à :

- ▶ Installer le kit de développement Java 7
- ▶ comprendre les atouts clés de la platte forme Java
- ▶ Lire, écrire, compiler, packager des classes Java
- ▶ Augmenter votre productivité avec Eclipse
- ▶ Mettre en œuvre l'héritage, le polymorphisme, l'interface
- ▶ Mettre en œuvre la généricité, l'autoboxing, les annotations
- ▶ Traduire des modèles de conception UML
- ▶ Récupérer des données d'une base de données.
- ▶ Travailler avec des Collections d'objets
- ▶ Découvrir les nouveautés Java 7 et Java 8

Programme détaillé cours :

Introduction

- ▶ Atouts java : portabilité, sécurité, réseau, gestion mémoire
- ▶ Comparaison Java et C/C++, PHP, .NET, Visual basic
- ▶ Lien entre java standard (jse), java entreprise (jee) et java micro (jme)
- ▶ Notion de JSR : Java Specification Request
- ▶ Notion de JCP : Java Community Process
- ▶ Historique : java a 20 ans !

Développement Java

- ▶ cycle : coder, compiler, exécuter, déployer
- ▶ outils clés de java : javac, java, jar, javadoc
- ▶ Byte code, machine virtuelle Java (JVM)
- ▶ Ramasse-miettes Java : Garbage Collector
- ▶ Gestion des exceptions dans java
- ▶ Différence entre JRE et JDK

Environnement Eclipse

- ▶ Valeur ajoutée d'un environnement de développement
- ▶ Comparaison Eclipse / Netbeans

- ▶ Notions de vues, perspectives, plugins
- ▶ Comprendre le Marketplace Eclipse
- ▶ Génération de la Javadoc,
- ▶ Génération d'un livrable Jar

API Java

- ▶ java.lang, java.util, java.io
- ▶ Librairies de base
- ▶ Java Virtual Machine (JVM)
- ▶ Technologies d'intégration (JDBC, RMI, JNDI)
- ▶ Technologies de déploiement java
- ▶ Généricité, bouche for étendue
- ▶ Scripting, annotations java
- ▶ Surveillance : technologie JMX, outils jconsole, jvisualjvm
- ▶ Sécurité java, Multithreading java, Logging java, XML
- ▶ Javadoc API JSE 7

Bases du langage

- ▶ Types primitifs, types Objets
- ▶ Tableaux dans java
- ▶ Chaînes de caractères dans java
- ▶ Opérateurs et Instructions
- ▶ Notion de package
- ▶ Gestion des exceptions java

Définition de classes Java

- ▶ Définitions de classes et fichiers source java
- ▶ méthodes, arguments,
- ▶ Champs, portée public, private, protected, default
- ▶ Organisation des fichiers et packages
- ▶ Importation de classes java
- ▶ Contrôle des accès dans java
- ▶ Constructeurs , this, super
- ▶ Conteneurs du JDK et classes string
- ▶ Méthode toString()
- ▶ Méthodes main() et points d'entrée
- ▶ Annotations java
- ▶ Javabeen, POJO

Classes utilitaires

- ▶ Classe de base : Object, Integer, String,
- ▶ StringBuffer, Math
- ▶ Les Collections : Vector, List, HashMap, ...

Entrées / Sorties

- ▶ Lectures / écritures dans un fichier
- ▶ Lectures / écritures à l'écran
- ▶ Gestion des répertoires et des attributs de fichiers
- ▶ Sérialisation et Introspection
- ▶ NIO.2

Concepts objets

- ▶ Classe et Objet
- ▶ Encapsulation
- ▶ Héritage (super)
- ▶ Interface
- ▶ Polymorphisme
- ▶ Surcharge, redéfinition

Accès aux données

- ▶ Introduction java JDBC

Objis, spécialiste formation Java

- ▶ Différences api java.sql et javax.sql
- ▶ Approche classique : Connection, Statement, ResultSet
- ▶ Interface Rowset, et comparaison rowset/resultset
- ▶ Notion de transaction. Autocommit JDBC
- ▶ Gestion Exceptions

Nouveautés Java 7

- ▶ Amélioration du langage (JSR 334 - projet Coin)
- ▶ Support des langages dynamiques (JSR 292 - invokedynamic)
- ▶ Accès complet au système de fichier (JSR 203 - NIO.2)

A venir dans Java 8

- ▶ Expressions Lambda et closures (JSR 335)
- ▶ Aliasing de méthodes et de constructeurs par référence,
- ▶ Ajout de méthodes par défaut dans les interfaces.