

Objectifs

Base idéale pour l'apprentissage du développement en langage Java : côté serveur (classes, types de variables, structure de contrôles, POO). Afin d'acquérir la maîtrise du langage, construire des applications simples, être productif, et entrevoir l'écosystème Java.

Participants

Programmeurs, chefs de projets.

Prérequis

Pratique de la programmation structurée.

Pédagogie

La pédagogie est basée sur le principe de la dynamique de groupe avec alternance d'apports théoriques, de phases de réflexion collectives et individuelles, d'exercices, d'études de cas et de mises en situations observées. Formation / Action participative et interactive : les participants sont acteurs de leur formation notamment lors des mises en situation car ils s'appuient sur leurs connaissances, les expériences et mettront en oeuvre les nouveaux outils présentés au cours de la session.

Profil de l'intervenant

Consultant-formateur expert sur cette thématique. Suivi des compétences techniques et pédagogiques assurée par nos services.

Moyens techniques

Encadrement complet des stagiaires durant la formation. Espace d'accueil, configuration technique des salles et matériel pédagogique dédié pour les formations en centre. Remise d'une documentation pédagogique papier ou numérique à échéance de la formation.

Méthodes d'évaluation des acquis

Exercices individuels et collectifs durant la formation. Evaluation des acquis et attestation de fin de stage adressés avec la facture.

Programme

Le langage Java

Naissance de Java

Pourquoi apprendre Java ?

Java et le Web

La JVM

Concepts de base et Concepts avancés

Concepts de base

Modèle de programme Java : main (String [])

Commentaire //, /**/

Référence et variables

Type de données simples et agrégés avec les tableaux

Casting et conversion de données

Opérateurs : arithmétique, relationnels et logiques

Structure de controle (for, if, switch...)

Blocs {}, portée et visibilité de références

Création et appels de méthode

Programmation objet avec Java

Approche objets et notation UML

Classes et Instances

Constructeur

Méthodes membres

Objet courant : this

Règles de Composition

Héritage

Utilisation des classes internes

Gestion des packages avec les classes

L'Autoboxing

Concepts avancés de la Programmation objet

Interfaces

Polymorphisme

Classes et méthodes abstraites

Classes templates (génériques)

Gestion des exceptions

Concepts et syntaxe : try, catch, finally, ...

Interceptions des exceptions

Création d'exceptions

Gestion des packages

Conceptions et syntaxe : import et déclaration

Constitution de packages

Archivage et outil : jar

Gestion des fichier et Streams : java.io

Hierarchies de classes

I/O standards

Fichiers et accès

Persistences de données

Les Collections

Manipulation des collections : Stack, Vector, Queue,...

Boucler sur les collections

Accès aux bases de données : java.sql

Concepts et API JDBC

Relation et SQL

Connections/déconnections

Manipulation de la définition des données : DDL

Manipulation des données : DML

Gestion de la Concurrency

Threads et Synchronisation

Mutex et verrous

Future et ExecutorService

Programmation graphique

Présentation de l'API AWT

Présentation de l'API Swing

Java et la programmation Web

Concepts et architecture

Intégration HTML/Java : JSP et servlets

Installation du serveur d'applications : Tomcat/CATALINA

Echanges HTTP, formulaires HTML

Bean et classes