

## Programmation Java : Introduction complète : Travaux Pratiques - 4 jours

*formation 471*

- Vous apprendrez à**
- Écrire, compiler et exécuter des programmes Java
  - Créer des applications robustes en utilisant les fonctionnalités objet de Java
  - Créer des applications fiables en utilisant les bibliothèques de classes Java
  - Développer des interfaces graphiques (GUI) indépendantes de toute plate-forme
  - Lire et écrire des données en utilisant Java Streams
  - Extraire des données de bases relationnelles avec JDBC

**Objectif** L'architecture unique de Java permet aux programmeurs de développer une seule et même application capable de fonctionner sur différentes plates-formes de manière constante et fiable. Lors de cette formation pratique, vous acquerez une expérience approfondie de Java et de ses fonctionnalités orientées objet. Vous apprendrez à créer des applications console et GUI ainsi qu'à stocker et récupérer des données à partir de bases de données relationnelles.

**À qui s'adresse cette formation** Toute personne devant développer des applications Java. L'expérience pratique préalable d'un langage de programmation (C, Pascal ou COBOL) est indispensable. Une connaissance des technologies Web et des concepts objet est utile.

**Travaux pratiques** Les exercices vous permettront d'acquérir une bonne expérience de la création d'applications Java. Lors d'une étude de cas, vous concevrez et construirez une application bureautique complexe. Les exercices incluent :

- Développement d'un modèle objet avec la notation UML
- Création d'objets Java et appel à leurs méthodes
- Structuration des données avec l'API des collections de Java
- Création d'interfaces utilisateur graphiques portables avec les composants Swing
- Ajout du traitement des événements aux GUI
- Extraction de données d'une base relationnelle JDBC

## Programmation Java : Introduction complète : Travaux Pratiques - 4 jours

*formation 471*

### Introduction à la programmation Java

#### Les avantages de Java

- Indépendance vis-à-vis de la plate-forme
- Applications et servlets autonomes

#### La structure d'un programme Java

- Compilation du code source dans du bytecode
- Vue d'ensemble des bibliothèques des classes

### Programmation objet avec Java

#### L'approche à objet

- La programmation objet
- Encapsulation, héritage et polymorphisme
- Analyse et conception objet : associations "Est un" et "A un"
- Conception pas à pas d'une application objet
- Modélisation de la structure objet avec UML (Unified Modeling Language)

#### Les fonctionnalités objet de Java

- Instanciation d'objets à partir des classes
- Agrégation et composition
- Extension des classes existantes
- Surcharge des méthodes

### Structure du langage Java

#### Syntaxe du langage

- Déclaration et initialisation des variables
- Instructions et expressions
- Déclaration et utilisation des tableaux
- Autoconversion du type de variable

#### Contrôle de flux

- Invocation des méthodes et passage de paramètres
- Conditionnels et boucles
- Traitement des exceptions avec **try** et **catch**

#### Définition des classes

- Champs (données d'instance)
- Méthodes (fonctions)
- Classes abstraites et interfaces
- Organisation des classes avec des packages et des modificateurs de la visibilité
- Composition ou héritage

### Construction des composants d'un programme Java

- Utilisation des classes existantes
- Optimisation des API collections avec l'utilisation des génériques
- Extension des classes de base
- Développement de nouvelles classes
- Compilation et débogage

### Développement de GUI

#### Les bases d'une interface utilisateur

- Objets graphiques GUI de base
- Programmation pilotée par les événements
- Atouts d'une bibliothèque de fenêtrage portable

#### JFC : Java Foundation Classes

- Avantages de composants Swing légers
- Étude de la bibliothèque de composants Swing
- Création de composants Swing : boutons, champs de saisie, listes défilantes
- Ajout de composants Swing aux conteneurs
- Mise en forme de composants Swing en utilisant les gestionnaires d'agencement
- Boîtes de dialogues et de messages

#### Traitement des événements

- Enregistrement des gestionnaires d'événements
- Classes internes et classes racines

### Stockage et récupération de données avec les entrées/sorties

#### Java Streams

- Streams, Readers et Writers
- Accès aux fichiers
- Attraper et lancer des exceptions
- Formatage aux sorties de texte

#### Fichiers et répertoires

- Lecture et écriture de fichiers
- Création, suppression et renommage de fichiers
- Obtenir des informations de fichiers et répertoires

#### Bases de données relationnelles

##### Accès aux bases de données JDBC

- Utilisation de l'API JDBC
- Sélection des pilotes de base de données

- Connexion à une base de données

### Amélioration des performances avec les instructions préparées et les procédures stockées

- Exécution d'instructions SQL
- Extraction et traitement des résultats

### Outils de développement Java

- Kit de développement Java (JDK)
- Compilateur (javac)
- Utilitaire Javadoc
- Utilitaire JAR (Java Archive)
- JDEs (Java Integrated Development Environments)