



# Fondamentaux du langage Java

## Objectifs

5 jours

Ce cours vous permettra de maîtriser les principes de l'approche objet et les mécanismes du langage. Les constructions du langage seront progressivement introduites à partir des concepts fondamentaux. Le cours abordera aussi les problèmes de conception (via la notation UML) et présentera les principales API et bibliothèques standard : les entrées/sorties, les utilitaires, les classes graphiques (AWT et Swing), les applets.

## Contenu de la formation

- Les techniques "objet"
- Les constructions de base du langage
- La définition et l'instanciation des classes
- L'héritage
- Les exceptions
- Présentation de quelques classes et bibliothèques standard

## Participants

Développeurs non confirmés, ingénieurs, chefs de projets proches du développement.

## Pré-requis

Connaissances de base en programmation. Expérience souhaitable en développement d'applications.

## Travaux pratiques

Les exercices pratiques ont été conçus pour illustrer tous les éléments du langage et pour mettre en oeuvre les concepts de la conception orientée objet : tous les exercices comportent une phase d'analyse/conception suivie d'une phase de programmation.



# Fondamentaux du langage Java

Programme

## » Les techniques " objet "

Les principes généraux de la modélisation et de la programmation " objet ".

L'abstraction et l'encapsulation.

Les différentes formes d'héritage.

Le polymorphisme.

Quelques modèles UML : diagrammes de classes, de séquences, de cas d'utilisation.

### Travaux pratiques

*La spécification UML d'une étude de cas qui sera l'un des fils directeurs des exercices suivants.*

---

## » Les constructions de base du langage

Déclaration et typage des variables et constantes.

Définition de méthodes.

Les expressions.

Les instructions de contrôle : les instructions conditionnelles, de boucle, de branchement.

Utilisation des tableaux.

Les types énumérés.

Les Unités de compilation et packages : le contrôle de la visibilité des classes, le mécanisme d'import.

Les imports statiques.

### Travaux pratiques

*Suite d'exercices simples permettant la prise en main de l'environnement de développement et la réalisation d'un programme simple. Utilisation des packages.*

---

## » La définition et l'instanciation des classes

Méthodologie de conception des classes.

Différence entre classes et objets.

Identification et création de propriétés et de méthodes.

Les constructeurs.

L'autoréférence.

Les propriétés et méthodes statiques.

La surcharge de méthodes.

### Travaux pratiques

*Programmation de l'étude de cas.*

---

## » L'héritage

Héritage par extension d'une classe existante.  
Simulation d'héritage multiple au travers de l'implémentation d'interface.  
Le polymorphisme et sa mise en œuvre.  
Définition de classes dérivées: impact sur les constructeurs.  
Utilisation de classes abstraites pour représenter des concepts génériques.  
L'utilisation simultanée de l'implémentation et de l'extension.  
La généricité.

### **Travaux pratiques**

*Conception et construction d'une hiérarchie de classes et d'interfaces. Mise en place du polymorphisme et de la généricité dans l'étude de cas.*

---

## » Les exceptions

Principe de la capture des erreurs par le mécanisme des exceptions.  
Capture d'exceptions  
Génération d'exceptions  
Délégation du traitement d'une exception  
Construction d'exceptions personnalisées

### **Travaux pratiques**

*Introduction des exceptions dans l'étude de cas.*

---

## » Présentation de quelques classes et bibliothèques standard

### ■ *La programmation des entrées/sorties*

La hiérarchie des classes d'entrée/sortie.  
Quelques classes de manipulation des systèmes de fichiers.  
Les entrées/sorties clavier.

### ■ *La programmation graphique*

Les concepts de base depuis jdk1.1.  
Le positionnement des composants graphiques : les conteneurs et les Layouts.  
Quelques composants graphiques : labels, boutons, zones de texte.  
La gestion des événements : les Listeners et Adapters.  
L'association de gestionnaires aux composants graphiques.

### ■ *Quelques classes utilitaires*

Les classes système.

Les classes de conteneur.

Les classes Date et Calendar

La classe Vector

### **Travaux pratiques**

*Construction d'une IHM avec lecture/écriture dans des fichiers. Mise en oeuvre de classes utilitaires.*