Smartfuture
Certifications | Training | Technology solutions

Formation

# Java SE Professionnel (Java 8 – Java 11)

Formation présentielle et en ligne à la demande. Maîtrisez le langage Java moderne, consolidez vos bases objet et découvrez les nouveautés clés de Java 8 à Java 11 pour développer des applications robustes, maintenables et performantes.

**5 jours**
35 h

**Format**
Atelier pratique

**Modalité**
Présentiel ou en ligne

**Niveau**
Intermédiaire

**INSCRIPTION / RÉSERVATION**

Je m'inscris maintenant

Formation 100% pratique

Exercices & cas d'usage

Formateurs expérimenté

## OBJECTIFS PÉDAGOGIQUES

- Maîtriser les fondamentaux de Java (syntaxe, types, API de base).
- Appliquer la programmation orientée objet dans des applications robustes.
- Exploiter collections, exceptions, I/O, gestion du temps et flux.
- Découvrir et utiliser les nouveautés de Java 8 à 11 (lambdas, Streams, Date/Time, modules).

## PUBLIC CIBLE

- Développeurs débutants en Java avec des bases en programmation.
- Développeurs Java souhaitant se mettre à jour vers Java 8–11.
- Ingénieurs, étudiants ou personnes en reconversion vers le développement Java.

## PRÉREQUIS

- Connaissances de base en algorithmique et notions de variables, boucles, conditions.
- Une première expérience de programmation (peu importe le langage) est recommandée.



© 2026 Smartfuture —Java SE Professionnel (Java 8 - Java 11)

PROGRAMME DE LA FORMATION – DÉTAILLÉ

## Introduction à Java et environnement de développement

### Écosystème Java SE

- Présentation de Java, de la JVM, du JDK, du JRE et des différentes éditions (SE, EE, ME).
- Versions de Java et évolution jusqu'à Java 11 (cycle de releases, LTS).
- Organisation d'un projet Java : packages, conventions de nommage, structure des sources.

### Prise en main de l'environnement

- Installation du JDK et configuration de l'environnement (variables, PATH).
- Utilisation d'un IDE (IntelliJ IDEA / Eclipse / VS Code) pour créer un projet Java.
- Compilation et exécution d'un premier programme Java en ligne de commande et via l'IDE.

## Syntaxe de base, types et structures de contrôle

### Types, variables et opérateurs

- Types primitifs, types références, constantes, déclaration et initialisation de variables.
- Opérateurs arithmétiques, de comparaison, logiques et opérateurs d'affectation composés.
- Conversion de types, promotion numérique, cast explicite et implicite.

### Structures de contrôle

- Instructions conditionnelles if / else if / else, opérateur ternaire.
- Switch classique et bonnes pratiques de lecture du code.
- Boucles for, while, do-while, utilisation des mots-clés break et continue.

### Tableaux et premières manipulations

- Déclaration, création et initialisation de tableaux à une dimension.
- Boucles sur les tableaux, for "classique" et for-each.
- Tableaux multidimensionnels : principes et exemples simples.


© 2026 Smartfuture —Java SE Professionnel (Java 8 - Java 11)

PROGRAMME DE LA FORMATION – DÉTAILLÉ

## Programmation orientée objet en Java

### Classes et objets

- Déclarer une classe, champs, méthodes, constructeur.
- Instanciation d'objets, référence et cycle de vie.
- Encapsulation, modificateurs d'accès (public, private, protected, package).

### Héritage et polymorphisme

- Notions de super-classe et sous-classe, héritage simple en Java.
- Redéfinition (override) et surcharge (overload) de méthodes.
- Utilisation de super, du mot-clé final, classes et méthodes finales.

### Interfaces et classes abstraites

- Différences entre interface et classe abstraite.
- Interfaces fonctionnelles (Java 8), méthodes par défaut et statiques.
- Organisation en packages, visibilité et modularisation du code.

## API de base Java SE : Strings, Date/Time, Collections

### Chaînes de caractères

- Classe String : immutabilité, méthodes principales (substring, indexOf, replace…).
- StringBuilder et StringBuffer pour la manipulation performante de textes.

### Gestion des dates et du temps (API java.time)

- LocalDate, LocalTime, LocalDateTime, Instant.
- Formatage et parsing avec DateTimeFormatter.
- Durée, période et opérations sur les dates.

### Collections & génériques

- Interfaces Collection, List, Set, Map : concepts et choix de structures.
- Implémentations courantes : ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap.
- Introduction aux génériques : List<String>, Map<K,V>, contraintes et avantages.


© 2026 Smartfuture —Java SE Professionnel (Java 8 - Java 11)

PROGRAMME DE LA FORMATION – DÉTAILLÉ

**Entrées / sorties**

- Flux d'entrée et de sortie (InputStream, OutputStream, Reader, Writer).
- Lecture et écriture de fichiers texte simples.
- try-with-resources : gestion automatique des ressources (Java 7+).

**API NIO.2**

- Package java.nio.file : Path, Files, manipulation de répertoires et de fichiers.

## Programmation fonctionnelle, Streams et concurrence

**Lambdas et programmation fonctionnelle (Java 8)**

- Expressions lambda : syntaxe, contexte et type cible.
- Interfaces fonctionnelles standard : Predicate, Function, Consumer, Supplier.
- Method references et simplification du code.

**Streams API**

- Création de streams à partir de collections et de tableaux.
- Opérations intermédiaires (filter, map, sorted, distinct…) et terminales (forEach, collect, reduce).
- Streams parallèles : principes et précautions.

**Notions de concurrence**

- Threads, Runnable et exécuteurs (ExecutorService).
- Synchronisation simple et problématiques de concurrence (vue d'ensemble).

## Nouveautés de Java 9 à Java 11 et bonnes pratiques

**Modules Java (JPMS)**

- Concept de module, fichier module-info.java.
- Organisation d'un projet modulaire simple.


© 2026 Smartfuture —Java SE Professionnel (Java 8 - Java 11)

PROGRAMME DE LA FORMATION – DÉTAILLÉ

## Exceptions, entrées/sorties et fichiers

### Gestion des exceptions

- Exceptions checked / unchecked, hiérarchie des exceptions.
- Blocs try / catch / finally, multi-catch, bonnes pratiques.
- Définir ses propres exceptions métier.

### Évolutions du langage et de l'API

- var pour les variables locales (Java 10) : usage et limites.
- Nouveau client HTTP (HttpClient) en Java 11 : principes et exemples.
- Autres petites évolutions utiles pour le développeur au quotidien.

### Atelier de synthèse

- Réalisation d'un mini-projet de bout en bout (conception, code, tests de base).
- Revue de code, conseils d'industrialisation et pistes pour aller plus loin (Spring, JPA, tests unitaires).


© 2026 Smartfuture —Java SE Professionnel (Java 8 - Java 11)