# Java


## Introduction

Java est un langage de programmation orienté objet, développé par Sun Microsystems (aujourd'hui propriété d'Oracle Corporation), qui a révolutionné le monde du développement logiciel depuis sa création en 1995.

### Caractéristiques principales de Java

Conçu pour être simple, sécurisé et portable, Java permet aux développeurs de créer des applications robustes et évolutives qui peuvent fonctionner sur une multitude de plateformes sans nécessiter de modifications spécifiques. Grâce à sa machine virtuelle (JVM), Java offre une indépendance vis-à-vis du matériel et du système d'exploitation, ce qui en fait un choix privilégié pour le développement d'applications web, mobiles, de bureau et embarquées. Sa vaste bibliothèque standard et son écosystème riche en frameworks et outils contribuent à sa popularité et à son adoption massive dans l'industrie. Java continue d'évoluer, intégrant de nouvelles fonctionnalités et améliorations pour répondre aux besoins changeants des développeurs et des entreprises.

### Historique et évolution de Java

## Installation et configuration de l'environnement de développement (JDK, IDE)

## Syntaxe de base et structures de contrôle

### Variables et types de données

### Opérateurs et expressions

### Structures de contrôle (boucles, conditions)

## Classes et Objets

### Définition de classes et création d'objets

### Constructeurs et méthodes

### Encapsulation et modificateurs d'accès

## Héritage et polymorphisme

### Concepts d'héritage et de polymorphisme

### Classes abstraites et interfaces

### Surcharge et redéfinition de méthodes

## Gestion des exceptions

### Introduction aux exceptions

### Blocs try-catch-finally

### Exceptions personnalisées

## Collections et génériques

### Introduction aux collections (List, Set, Map)

### Utilisation des génériques

### Classes utilitaires (Collections, Arrays)

## Flux d'Entrée/Sortie (I/O)

### Lecture et écriture de fichiers

### Flux de données (InputStream, OutputStream)

### Sérialisation et désérialisation

# Programmation concurrente

## Processus

Un processus est une instance d'un programme en cours d'exécution sur un système d'exploitation. Il représente une unité de travail indépendante qui peut être gérée par le système d'exploitation. Un processus possède son propre espace mémoire, ses propres ressources (comme des fichiers ouverts, des connexions réseau, etc.), et son propre contexte d'exécution.

## Concurrence

La concurrence (ou "concurrency" en anglais) est un concept général qui fait référence à la capacité d'un système à gérer plusieurs tâches en même temps. Cela inclut non seulement le multithreading, mais aussi d'autres formes de parallélisme comme le multiprocessus, les coroutines, et les tâches asynchrones. L'objectif principal de la concurrence est de permettre une meilleure utilisation des ressources du système et d' améliorer la réactivité des applications.

## Thread

Un thread, ou fil d'exécution, est une unité d'exécution indépendante au sein d'un processus. Il représente une séquence d'instructions qui peut être exécutée simultanément avec d'autres threads du même processus. Bien qu'ils partagent les ressources globales de ce processus, comme la mémoire et les descripteurs de fichiers, chaque thread possède son propre compteur de programme, sa propre pile et ses propres variables locales.

## Multithreading

Le multithreading est une technique de programmation qui permet de créer et de gérer plusieurs threads au sein d'un même processus. Chaque thread est une unité d'exécution indépendante qui peut effectuer des tâches en parallèle avec d'autres threads.

## Création et gestion de threads

```
Thread thread = Thread(() -> {
    System.out.println("hello " + Thread.currentThread().getName());
});
thread.start();
```

## Synchronisation et classes concurrentes (CopyOnWriteArrayList, ConcurrentHashMap)

# Bibliothèques Standard et API Java

## Introduction aux bibliothèques standard de Java

## Utilisation des API courantes (java.util, java.time, java.nio)

### Gestion des dépendances avec Maven ou Gradle

## Développement d'Applications de Bureau

### Introduction à Swing et JavaFX

### Création d'interfaces utilisateur graphiques (GUI)

### Gestion des événements et des composants graphiques

## Développement Web avec Java

### Introduction aux servlets et JSP

### Frameworks web (Spring MVC, JSF)

### Déploiement d'applications web sur des serveurs (Tomcat, Jetty)

## Accès aux Bases de Données avec JDBC

### Introduction à JDBC

### Connexion et interaction avec une base de données

### Gestion des transactions et des pools de connexions

## Frameworks et ORM

### Introduction aux ORM (Hibernate, JPA)

### Mapping objet-relationnel

### Gestion des entités et des relations

## Développement d'Applications RESTful

### Introduction aux services web REST

### Création d'API REST avec Spring Boot

**Consommation de services web REST**

## Sécurité des Applications Java

**Introduction à la sécurité des applications Java**

**Authentification et autorisation**

**Protection contre les vulnérabilités courantes (injection SQL, XSS)**

## Projet de Développement Logiciel en Java

**Réalisation d'un projet de développement logiciel en groupe**

**Application des concepts appris tout au long du cours**

**Présentation et évaluation du projet**

## Microservices et Architecture Modulaire

**Introduction aux microservices**

**Conception et développement de microservices avec Spring Boot**

**Communication entre microservices (REST, gRPC)**

## Développement Mobile avec Java

**Introduction à Android et Kotlin**

**Développement d'applications mobiles avec Android Studio**

**Intégration de services backend avec des applications mobiles**

## Tests et Qualité du Code

**Introduction aux tests unitaires et d'intégration (JUnit, TestNG)**

**Tests de performance et de charge**

### Outils de qualité du code (SonarQube, Checkstyle)

## Déploiement et DevOps

### Introduction aux pratiques DevOps

### Déploiement continu et intégration continue (CI/CD)

### Outils de déploiement (Docker, Kubernetes, Jenkins)

## Liens Github

Programmation concurrente