

# Introduction

Thibaud Martinez

[thibaud.martinez@dauphine.psl.eu](mailto:thibaud.martinez@dauphine.psl.eu)

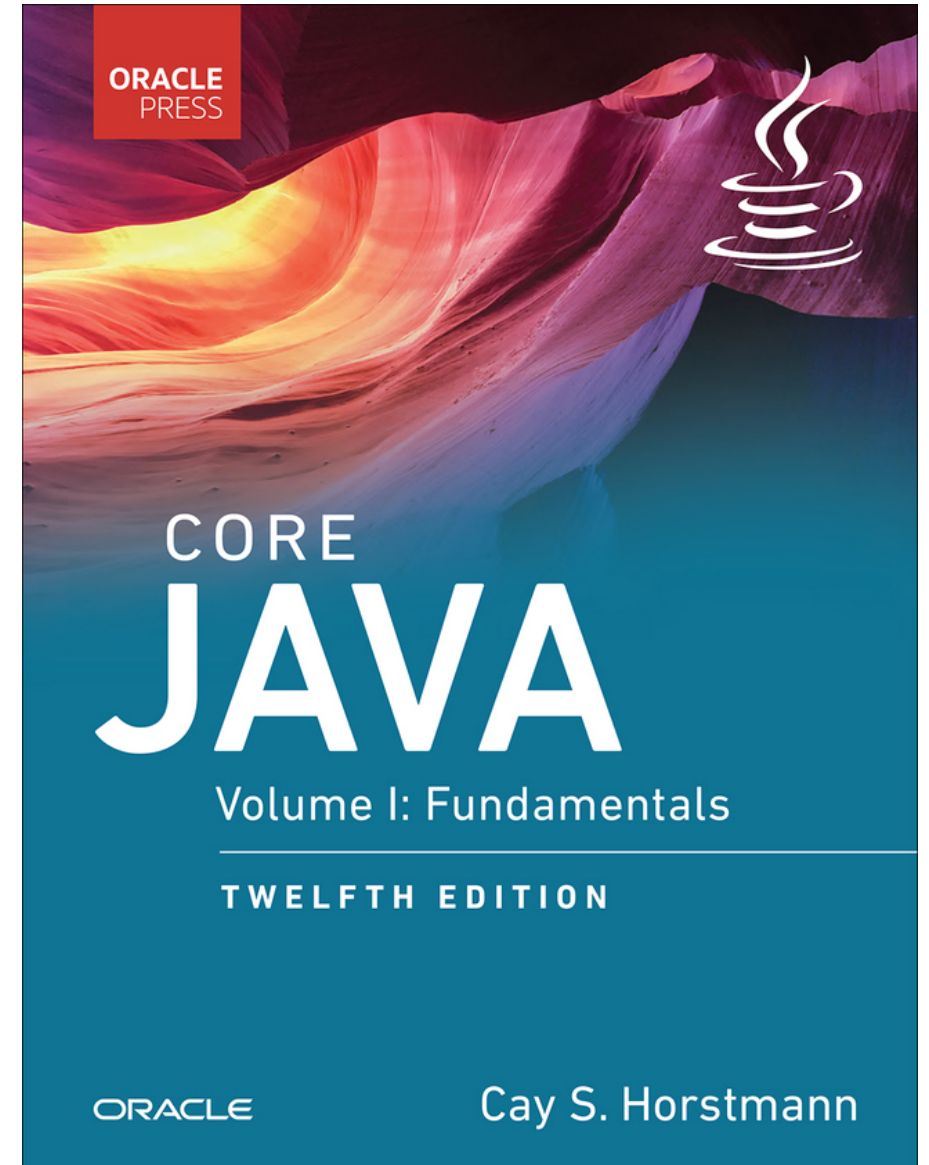
# Objectifs

- Se familiariser avec la syntaxe Java et être capable de réaliser des programmes simples;
- comprendre les concepts de base de la programmation orientée objet;
- apprivoiser les outils de la plateforme Java et découvrir la bibliothèque standard.

# Prérequis

- Une précédente expérience avec un langage de programmation (C, C++, Python, PHP, VBA, etc.);
- une précédente expérience avec un système d'exploitation (UNIX de préférence);
- de la motivation.





L'ouvrage de référence pour ce cours  
est **Core Java, Volume I:  
Fundamentals, 12th Edition** par  
*Cay Horstmann*.



# Plan

- Chapitre 1 - **Structures de programmation de base (1)**
- Chapitre 2 - **Structures de programmation de base (2)**
- Chapitre 3 - **Objets et classes**
- Chapitre 4 - **Héritage**
- Chapitre 5 - **Exceptions**
- Chapitre 6 - **Généricité**
- Chapitre 8 - **Collections**

# Conventions de présentation

-  matérialise une remarque
-  signale un point d'attention
-  indique la survenance d'une erreur
-  fait référence à une pratique de programmation recommandée

Les liens vers les sources sont situés en pied de page.

# Java, un historique

- Créé à *Sun Microsystems* par une équipe menée par **James Gosling**.
- Première version publiée en 1996.
- *Sun Microsystems* est rachetée en 2009 par *Oracle* qui détient et maintient désormais Java.
- [OpenJDK](#) est une implémentation *open source* de Java. Depuis Java SE 7 en 2011, c'est l'[implémentation de référence](#).
- [Java SE 18](#) est actuellement la version la plus récente.

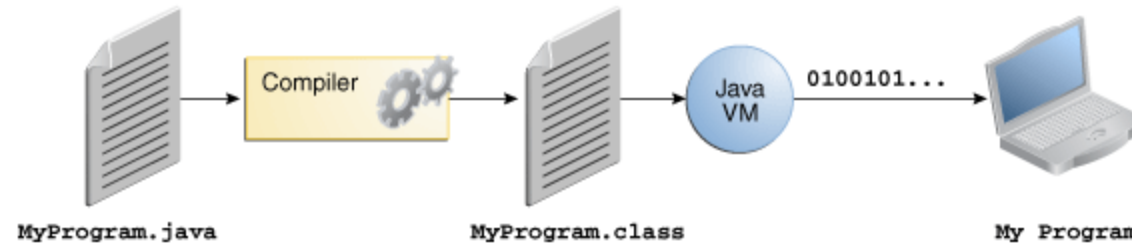
# Principes

Plus qu'un langage, est une plateforme de programmation comprenant des bibliothèques logicielles et un environnement d'exécution, construite autour de 11 principes.

1. Simple
2. Orienté-objet
3. Distribué
4. Robuste
5. Sécurisé
6. *Architecture-Neutral*
7. Portable
8. Interprété
9. Haute-performance
10. *Multithreaded*
11. Dynamique



# Fonctionnement



- Le code source est d'abord écrit dans des fichiers textes se terminant par l'extension `.java`.
- Ces fichiers sources sont ensuite compilés en fichiers `.class` par le compilateur `javac`. Ceux-ci contiennent ne contient pas le code natif du processeur, mais des *bytecodes*, une représentation intermédiaire pour la machine virtuelle Java (JVM).
- Le lanceur `java` exécute ensuite l'application avec une instance de la machine virtuelle Java.

# Pourquoi apprendre Java en 2022 ?

- 3e langage le plus populaire au monde selon l'[index TIOBE](#).
- [Un nombre important de grandes entreprises utilise Java](#) (Google, AirBnb, BNP Paribas...)
- L'écosystème Java est mature et très fourni.
- Le langage [continue d'évoluer et se modernise](#).

*"There are only two kinds of languages: the ones people complain about and the ones nobody uses."*

**Bjarne Stroustrup**, créateur de C++

# Le jargon Java

Nom	Description
Java Development Kit (JDK)	Pour développer des programmes Java.
Java Runtime Environment (JRE)	Pour exécuter des programmes Java, ne contient pas les outils de développement.
Standard Edition (SE)	La plateforme Java pour les ordinateurs classiques et les serveurs.
HotSpot	Un compilateur <i>just-in-time</i> développé par Oracle.
Long Term Support (LTS)	Une version de Java maintenue pendant plusieurs années, contrairement aux versions publiées tous les 6 mois qui servent à mettre en avant les nouvelles fonctionnalités.

# Installation

- On utilisera **OpenJDK 17** qui implémente Java SE 17 (une version LTS), avec le compilateur HotSpot.
- Il existe plusieurs distributions (*builds*) d'OpenJDK.
- On recommande l'installation de la distribution [Adoptium Eclipse Temurin 17](#) disponible sur [adoptium.net](#).

⚠ Pour pouvoir exécuter les binaires Java sans avoir à utiliser le chemin complet vers ceux-ci, il faut [ajouter le répertoire des binaires à la variable d'environnement](#) `PATH`.

# JShell

- Java Shell (JShell) est un outil interactif pour la programmation Java .
- JShell est une boucle ***read-eval-print*** (REPL), qui évalue les déclarations, les instructions et les expressions au fur et à mesure de leur saisie et affiche immédiatement les résultats.
- Bien que son fonctionnement diffère légèrement du mode d'exécution classique, il s'agit d'un outil pratique pour tester des fonctionnalités du langage et prototyper.