



Chapitre 1 : Introduction à la programmation orientée objet

Dr Amina Debbah
Université Badji Mokhtar-ANNABA
Promotion: 2ème année LMD
Année : 2023/2024

- Paradigmes de programmation
- Programmation procédurale
- Problèmes
- Programmation orientée objet
- Avantages
- Notion d'objet et de classe
- Objet Vs classe
- Java
- Un premier programme

Paradigmes de programmation

- Le paradigme est une manière de résoudre un problème, une manière de penser et une concrétisation d'une philosophie de programmation.

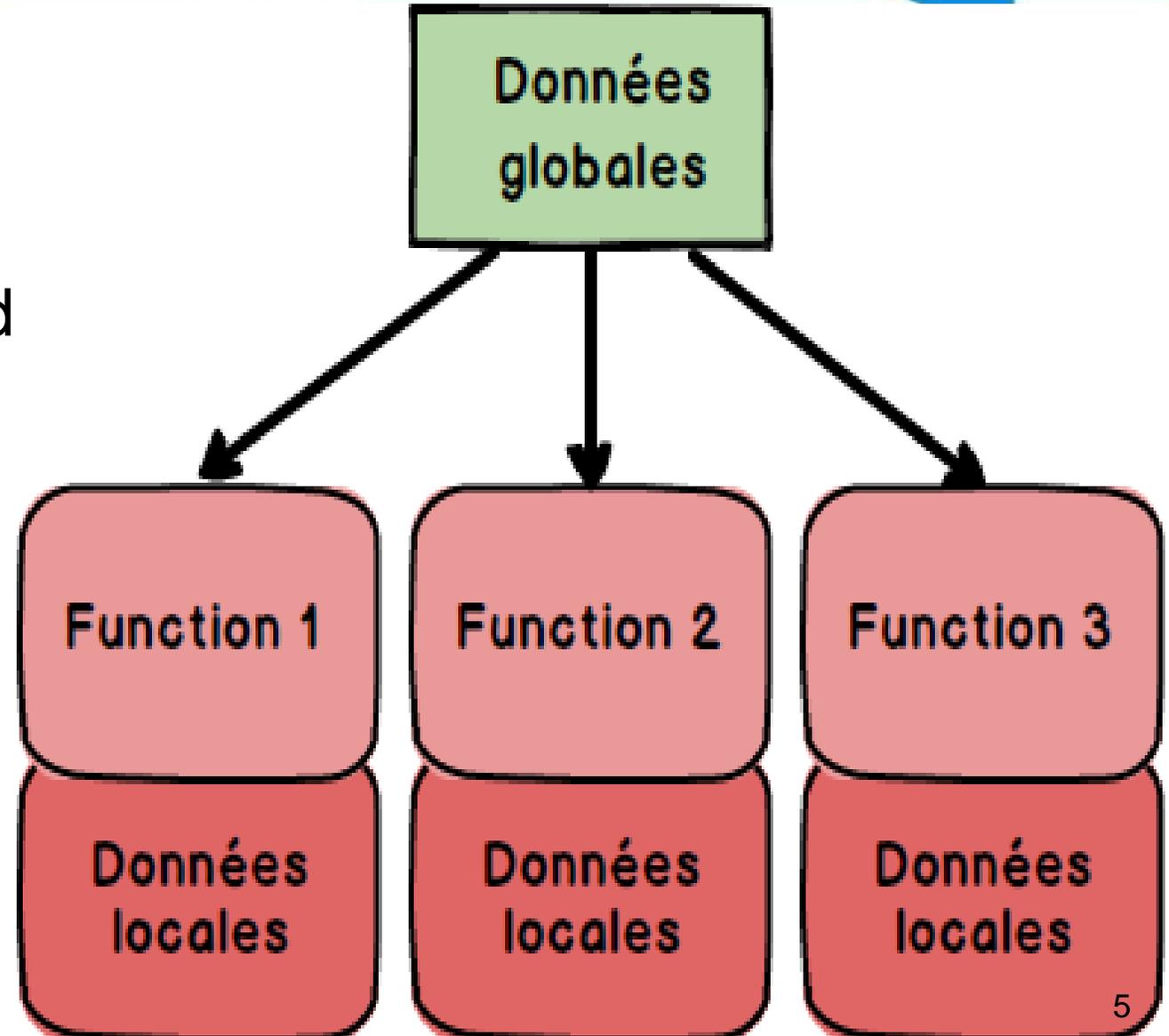
- Deux principaux paradigmes de programmations
 - programmation structurée(procédurale)
 - programmation par objet ou orientée objet(POO)

Que doit faire mon programme ?

- Programmation structurée basée sur l'utilisation des fonctions et des procédures
- Approche logique classique mise en évidence de la fonction à réaliser
- Décomposition hiérarchique descendante en un ensemble de sous-fonctions
- Les données peuvent être accessible et modifiable
- l'accent est mis sur la manière dont la tâche sera effectuée dans un ordre séquentiel

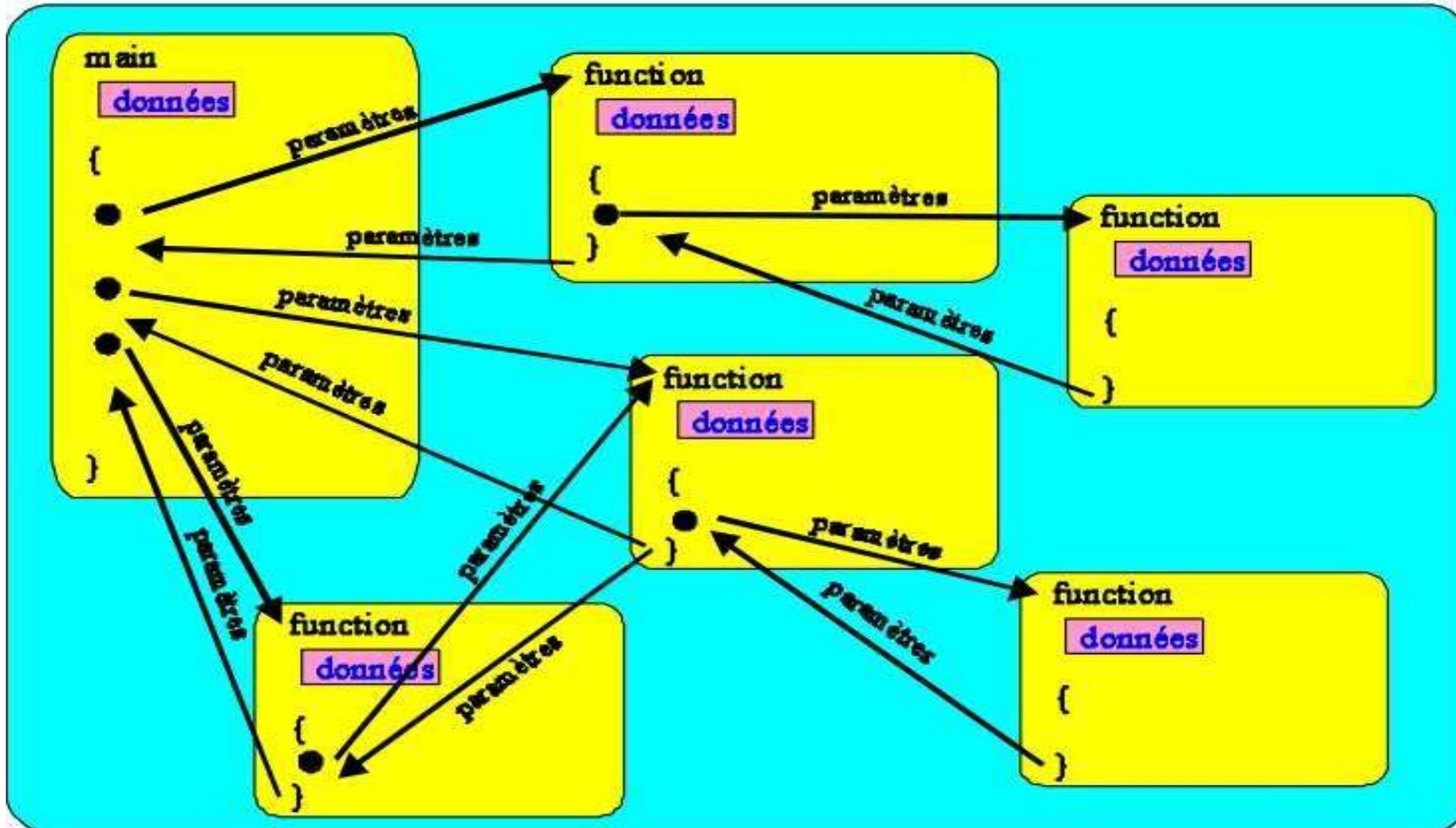
Programmation procédurale (2/4)

□ Chaque fonction résoud une partie du problème



Programmation procédurale (3/4)

- Exemple d'un programme en C



□ Problèmes :

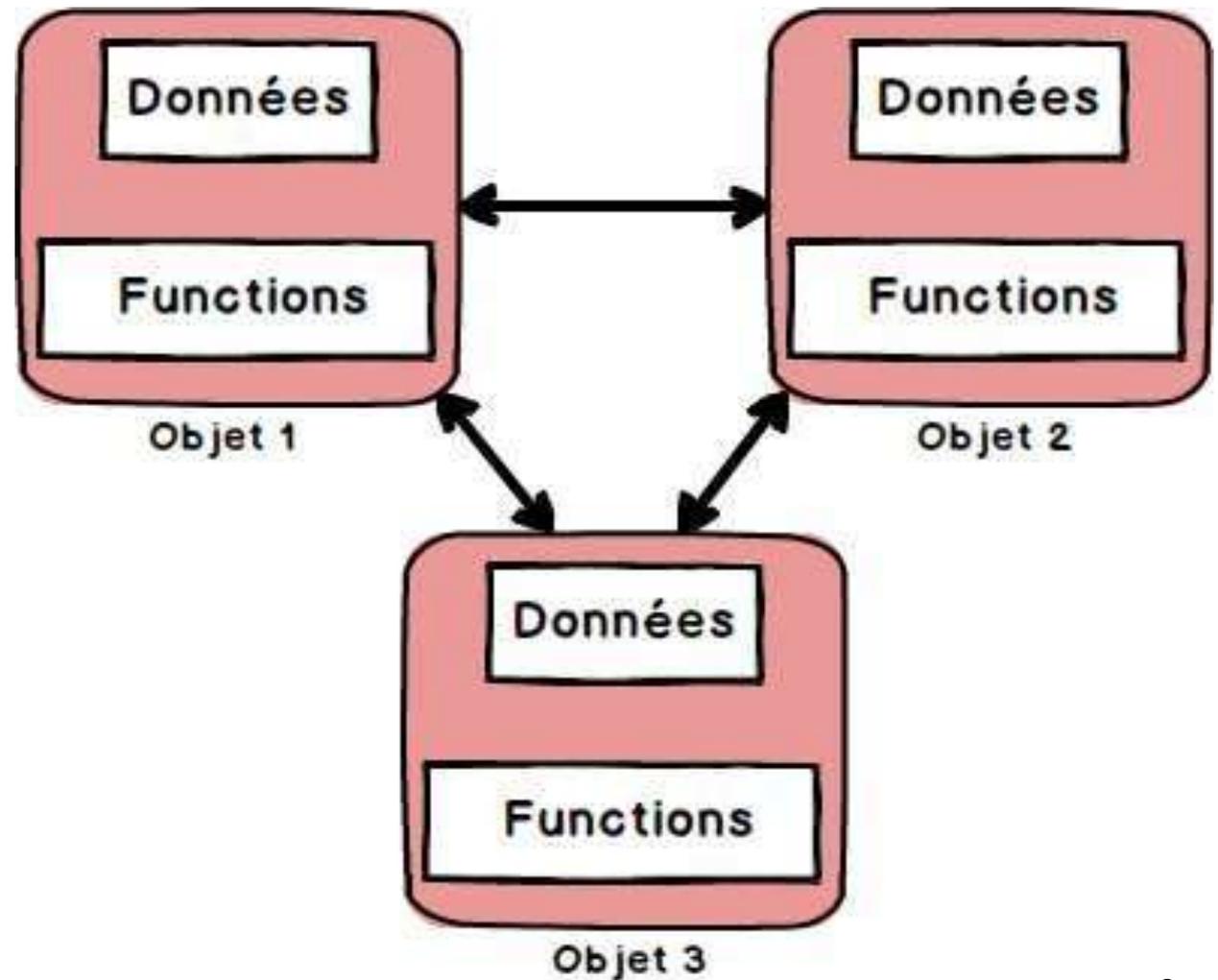
- Difficulté de faire évoluer le système et le maintenir.
- Difficulté de réutiliser le codes tout au long du programme.
- Accès libre aux données par n'importe quelle fonction.
- Les modifications d'une fonction entraînent d'autres modifications dans autre fonctions, etc.
- Redondance dans le code , la même chose est codée plusieurs fois)

De quoi doit être composé mon programme ?

- Programmation Basée sur la notion d'objet qui est l'abstraction du monde réel.
- Un programme est composé d'un ensemble d'objets qui interagissent entre eux.
- Architecture du système fondées sur les données, qui a plus de chance de résister au temps et aux changements exigés par les utilisateurs
- Les données et les traitements sont regroupés ensemble.

Programmation orientée objet (2/4)

- ❑ Objet = entité qui regroupe des variables et des fonctions permettant de les manipuler
- ❑ Chaque objet contrôle ces propres données
- ❑ Les objets communiquent à travers des messages



□ Avantages:

- Offrir une meilleure organisation du code
- Faciliter la réutilisation du code : réutiliser le fragment du code développé dans un cadre différent
- Faciliter l'évolution du code, le développement, le maintien.
- Améliorer la conception et la maintenance des grands systèmes
- Assurer la Modularité : les objets forment des modules compacts regroupant des données et un ensemble d'opérations.

Programmation orientée objet (4/4)

- Améliorer la qualité, la lisibilité du code.
- Fournir un cadre plus homogène, pour étudier simultanément les données et leur comportement.

□ Un objet

Représente un concept, une idée ou toute entité du monde physique ayant une identité , Capable de sauvegarder un état c'est-à-dire un ensemble d'information dans des variables internes.

Un objet est défini par :

- **Un état :**

Les valeurs des attributs (variables qui stockent les valeurs)

- **Une identité :**

Permet de distinguer un objet d'un autre objet

- **Un comportement :**

Définit les méthodes qui modifient les états et /ou envoient des messages à d'autres objets

un objet = un état + une identité + un comportement

- **Exemple 1:**

- Objet = être humain.

- Son identité peut être son nom ou bien un numéro.
 - Ses états seront sa taille, son poids, la couleur de ses yeux, etc..
 - Ses comportements seront respirer, marcher, parler, etc..



- **Exemple 2:**

- Objet = voiture

- Etat =

- Puissance

- Suspension

- Matricule

- Fabricant

- Etc.

- comportements = marche arrière, marche avant, freiner



- **Les attributs** d'un objet sont l'ensemble des informations qui décrivent la structure interne présentés sous forme de variable
 - l'état de l'objet est représenté par les valeurs des attributs à un instant précis.

- **Les méthodes** Ce sont des fonctions liées à des objets et qui précisent le comportement de ces objets

□ Les Message

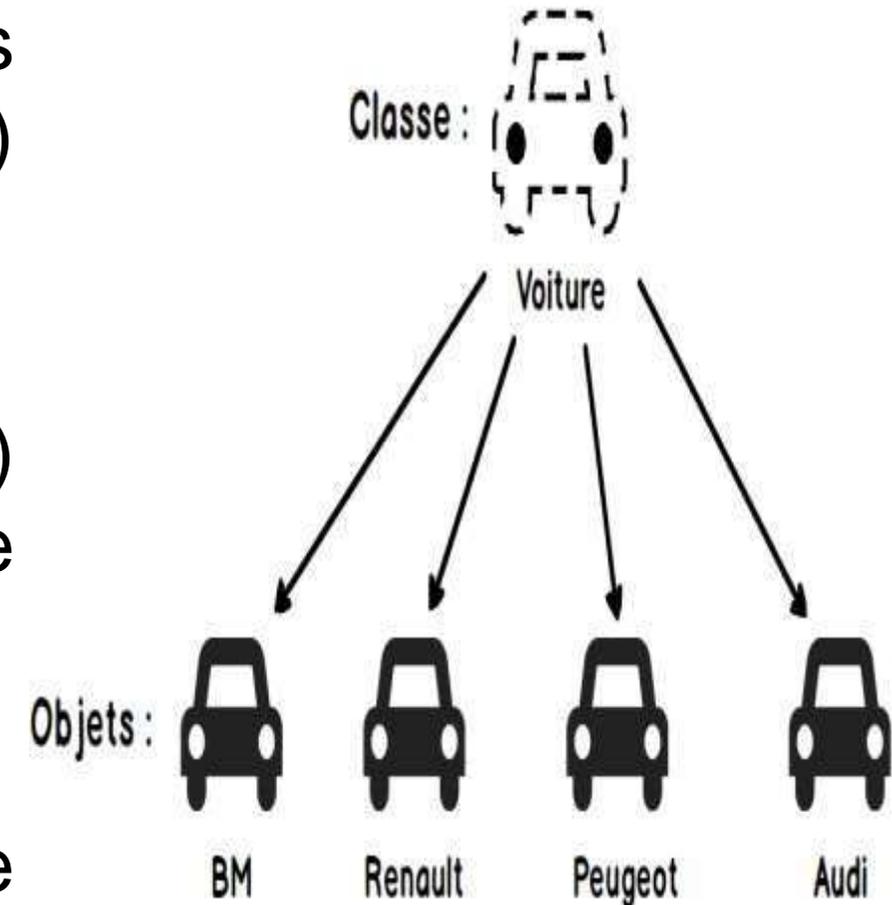
- Un message est une demande d'activation d'une méthode envoyé à un objet.
- un objet peut recevoir un message qui déclenche : une fonction qui modifie son état et/ou une fonction qui envoie un message à un autre objet
- Une méthode est déclenchée à la réception d'un message particulier

- Les classes sont des moules, des patrons qui permettent de créer des objets en série sur le même modèle.
- On peut représenter une classe comme le schéma de construction ainsi que la liste des fonctionnalités d'un ensemble d'objets.
- Ensemble d'éléments qui partagent un même ensemble de comportements et une même structure

une classe = attributs+ méthodes

Objet vs classe

- Classe = définitions pour des données (variables) + fonctions (méthodes) agissant sur ces données
- Objet = élément d'une classe (instance) avec un état, une méthode ou une variable
- Deux objets issus de la même classe peuvent contenir des valeurs différentes



- Créé par James Gosling et Patrick Naughton, employés dans Sun Microsystems en 1995
- Open source : on peut récupérer le code source java pour le modifier et l'utiliser plus tard .
- Dernière version java SE 17
- Est multiplateforme
- Est fortement typé
 - Toute variable doit être déclarée avec un type

□ Utilisé pour :

- Application Web
- Application mobile (Pour Android)
- Application desktop

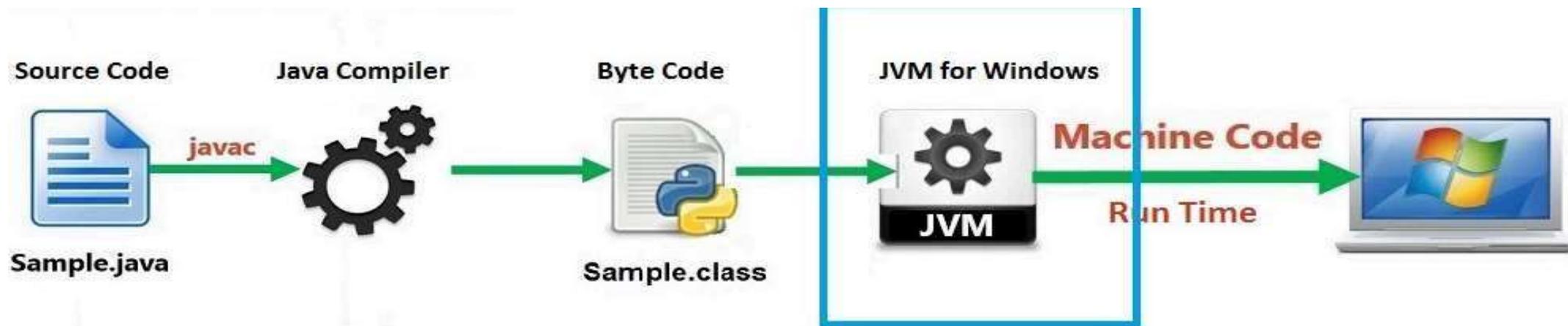
□ Il existe plusieurs plateformes Java

- Java SE (Standard Edition)
- Java ME (Micro Edition) et Java EE (Entreprise Edition).
- Un ensemble d'outils, le JDK (Java Development Kit) ou SDK (Software Development Kit)

□ Points forts

- Simple
- Syntaxe proche du langage (C/C++)
- Assure le développement
- Portable
- Sécurisé
- Garantit une haute performance
- Robuste
- Acroît la réutilisabilité et la productivité

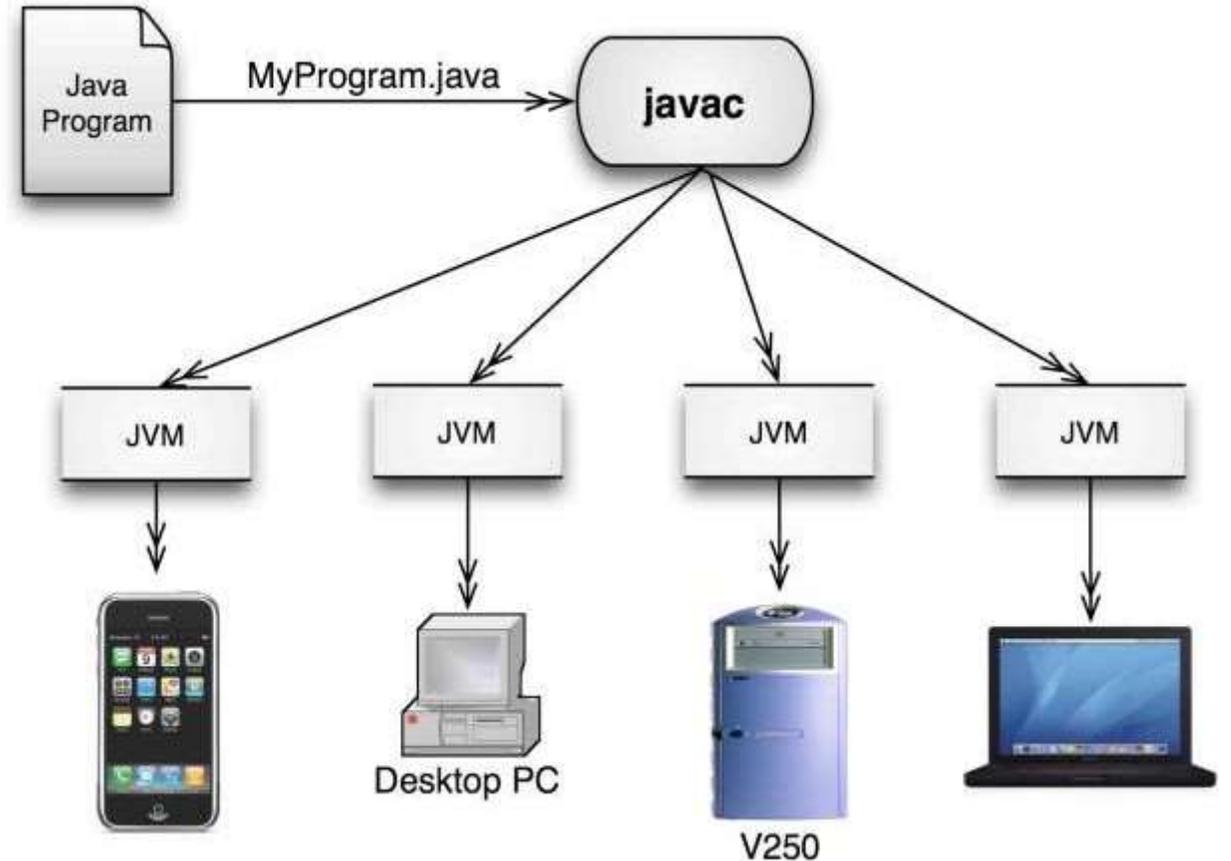
- Java est à la fois un langage compilé et interprété.
- Le compilateur JAVA (javac) génère du byte code (Création d'un fichier .class)
- Le byte-code est intermédiaire entre le code source et le langage machine indépendant de la plateforme



- La JVM Un “ processeur logiciel” qui interprète des instructions du “bytecode” en les traduisant dans le langage machine du processeur sur lequel s’exécute notre programme.”
- La JVM interprète le byte-code généré par le compilateur
- Permet l'interaction avec le système d'exploitation
- Assure la gestion de la mémoire grâce au système garbage-collector (ramasse-miettes)

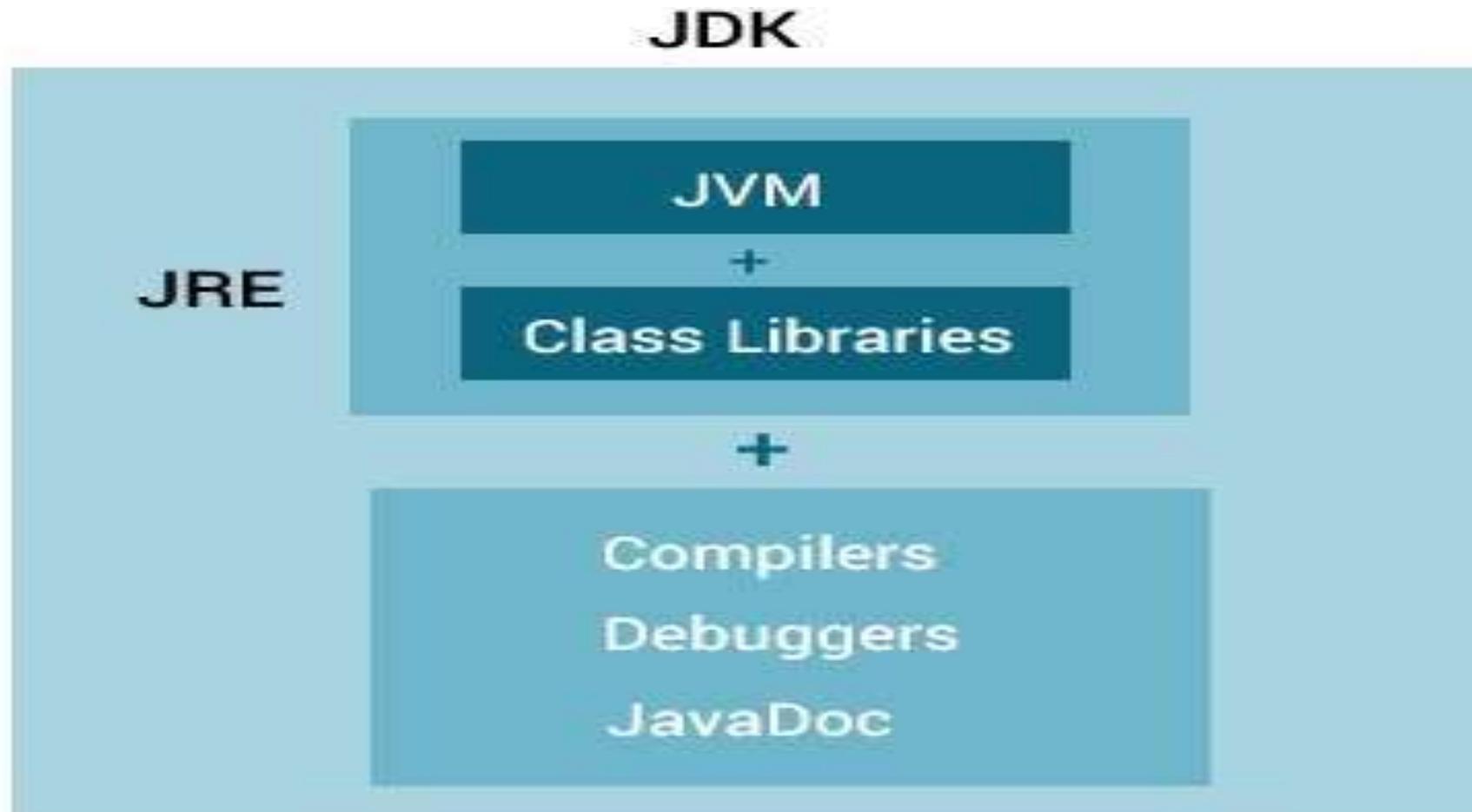
Interprétation JVM (2/2)

- Java est portable uniquement sur les plateformes où une JVM a été portée.



- JRE (Java Runtime Environment) : un environnement qui contient tout ce qui est nécessaire pour exécuter des programmes Java
- JDK (Java Development Kit) : contient tous les outils nécessaires pour développer, compiler, et exécuter des programmes Java

JVM vs JRE vs JDK



Un premier programme ! (1/2)

□ Tous les programmes Java sont composés d'au moins une classe.

▪ **Exemple :**

```
public class HelloWorld {
```

▪ Le point de départ de tout programme Java est la méthode suivante :

```
    public static void main (String[] args) {
```

```
System.out.println("Hello World");
```

```
}
```

```
}
```

▪ Dans un fichier de nom HelloWorld.java

Permet d'afficher un message

Un premier programme ! (2/2)

Règle 1: Toute classe publique doit être dans un fichier qui a le même nom que la classe

Règle 2: Pour avoir un programme exécutable il faut toujours une classe qui contient la méthode main

Fin du chapitre
Questions ?