Introduction à Java

Par Aïcha El Golli

aicha.elgolli@essai.ucar.tn

Organisation du cours

✓Module de 42h : **14 semaines** x3h :

- 1h30 cours +1h30 TD (Mme El Golli) P3 et P4
- Travail et cours sur classroom

✓Modalités d'évaluation :

Note de CC: des rendus de travaux dirigés (évaluations au niveau des TD) - Bonus/malus sur l'implication (présence / pertinence des interactions) en cours/TD et DS en fin de P3 (35%)

✓Examen final sur les concepts vus en cours/TDs (65%) ==>fin P4

Objectifs du cours

- Apprendre à programmer avec des objets
  - adopter le “penser objet”
  - connaître et savoir mettre en œuvre les concepts fondamentaux de la programmation objet
- Apprendre à coder proprement
- Connaître le langage Java

Contenu de la matière

- **Chapitre 1 :** Introduction à java
- **Chapitre 2:** Instructions de base
  - Les commentaires, Variables, types primitifs, portée, Opérateurs Instructions de contrôle, Entrées-sorties standards, Fonctions Programme, compilation, exécution
- **Chapitre 3:** Tableaux de types primitifs et chaines de caractères, Méthodes statiques
- **Chapitre 4:** Notions de classes, objets et tableaux d'Objets
- **Chapitre 5:** Héritage
- **Chapitre 6:** Polymorphisme downcasting et upcasting
- **Chapitre 7:** Héritage et abstraction: Classes abstraites, Interfaces et initiation au graphisme
- **Chapitre 8** Les exceptions? / Types paramétrés et généricité
- **Pré-requis :** Des notions basiques ou avancées du langage C ou C++ /connaissance de la conception OO

Apprendre à programmer avec des objets

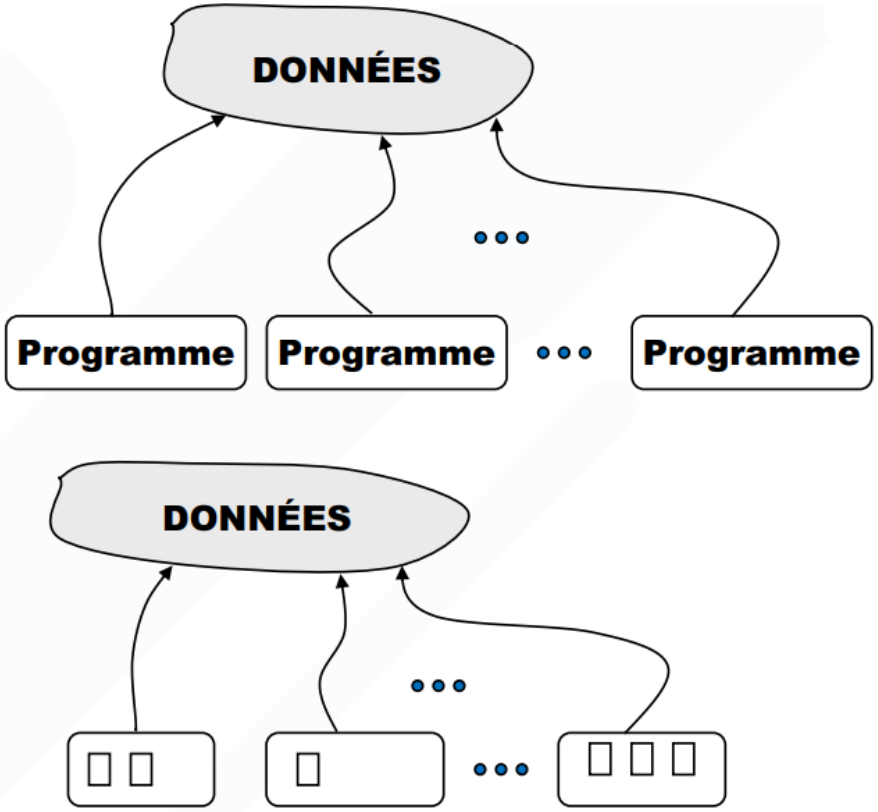
Notions de programmation objet

- Classes et instances
- Visibilité, composition et délégation
- Interface et polymorphisme
- Extension et redéfinition de méthode
- Types paramétrés et généricité
- …

Évolution de la programmation

- **Programmation séquentielle :**
  Les instructions sont exécutées de façon linéaire. Les données sont globales et aucune réutilisation du code n'est réalisée.
- **Programmation procédurale :**

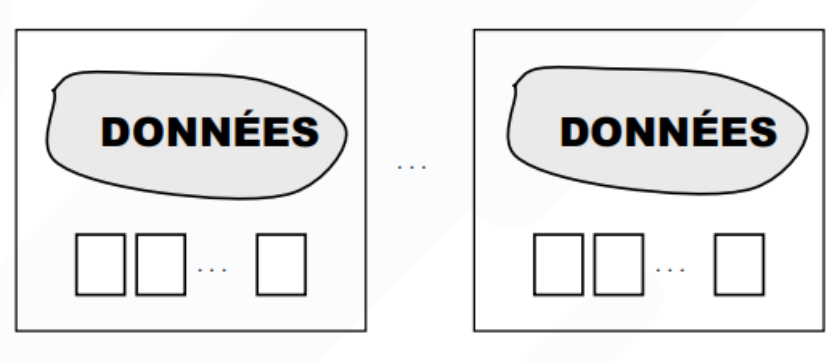
Il s'agit de découper un programme en une série de fonctions. chaque fonction a pour but de réaliser un traitement particulier. On procède à un découpage du traitement mais les données restent globales.



Évolution de la programmation

- **Programmation orientée objet :** Il s'agit de regrouper l'ensemble des données et les traitements qui s'appliquent à ces données.

On procède à un découpage naturel des données et du traitement des données.



Exemple : Application chargée de la paye des employés d'une entreprise. Créer une class « Employe » : toutes les données qui caractérisent un employé et les méthodes qui manipulent ces données,

Qu’est-ce que programmer proprement ?

- Un programme propre :
- respecte les attentes des utilisateurs
- est fiable
- peut **évoluer facilement**/rapidement
- est **compréhensible** par tous
- Et est **bien commenté**

Comment programmer proprement ?

- Pour programmer proprement dans un langage objet, il faut :

- nommer correctement les éléments du code
- écrire du code lisible (par un autre humain)
- relire et améliorer le code
- Commenté son code correctement

➔ Du code sale est souvent sources d'erreurs et n'est pas maintenable.

JAVA c'est quoi ?

- Le langage Java a été conçu au sein de l'entreprise Sun Microsystems (par une équipe dirigée par James Gosling et Patrick Naughton).
  - Initialement baptisé Oak (chêne), il a été officiellement lancé en 1995 sous le nom Java
    - Java est synonyme de "café" en argot américain

- **Les langages de programmation sont en général interprétés (Basic ou JavaScript) ou compilés (Pascal ou C).**

- Pour pouvoir être un langage multi-plateforme, Java est un mélange de ces deux possibilités : les fichiers sources **.java** doivent être compilés pour fournir un fichier **.class** qui pourra lui, être interprété.

JAVA c'est quoi ?

- **Programmer en java suppose donc que l'on dispose d'un compilateur et d'un interpréteur java. Ceux-ci sont fournis par le JDK (Kit de Développement en Java) disponible sur le site d'Oracle (qui a racheté SUN)**

Java is a popular programming language, **created** in **1995**.

It is owned by Oracle, and more than **3 billion devices run Java**.

It is used for:

Mobile applications (specially Android apps)

Desktop applications

Web applications

Web servers and **application servers**

Games

Database connection

And much, much more!

Que recouvre le mot Java ?

- Un projet
  - 1993 : projet Oak (langage pour l'électronique grand public)
  - 1995 : Java 1.0 est rendu public, marquant la naissance officielle de Java en tant que langage de programmation.
  - 1996 : Java 1.1 introduit les classes internes, JDBC pour la connectivité des bases de données, etc.
  - …
  - 2004 : Java 5 (également connu sous le nom de Java 1.5 ou « Tiger » ou J2SE 5.0 !!) introduit des améliorations majeures du langage, notamment des génériques (proche des templates de C++), des types énumérés, autoboxing/unboxing des types primitifs dans les collections, et des annotations.
  - 2006 : v.1.6 (ou « Mustang » ou Java SE 6) améliorations des performances (Swing notamment), interactions avec scripts (PHP, Python, Ruby, JavaScript), Java DB…
  - Mars 2014: v1.8 (Java SE 8) a changé la donne avec les expressions lambda, l'API Stream et le nouveau package java.time pour la manipulation de la date et de l'heure.

Que recouvre le mot Java ?

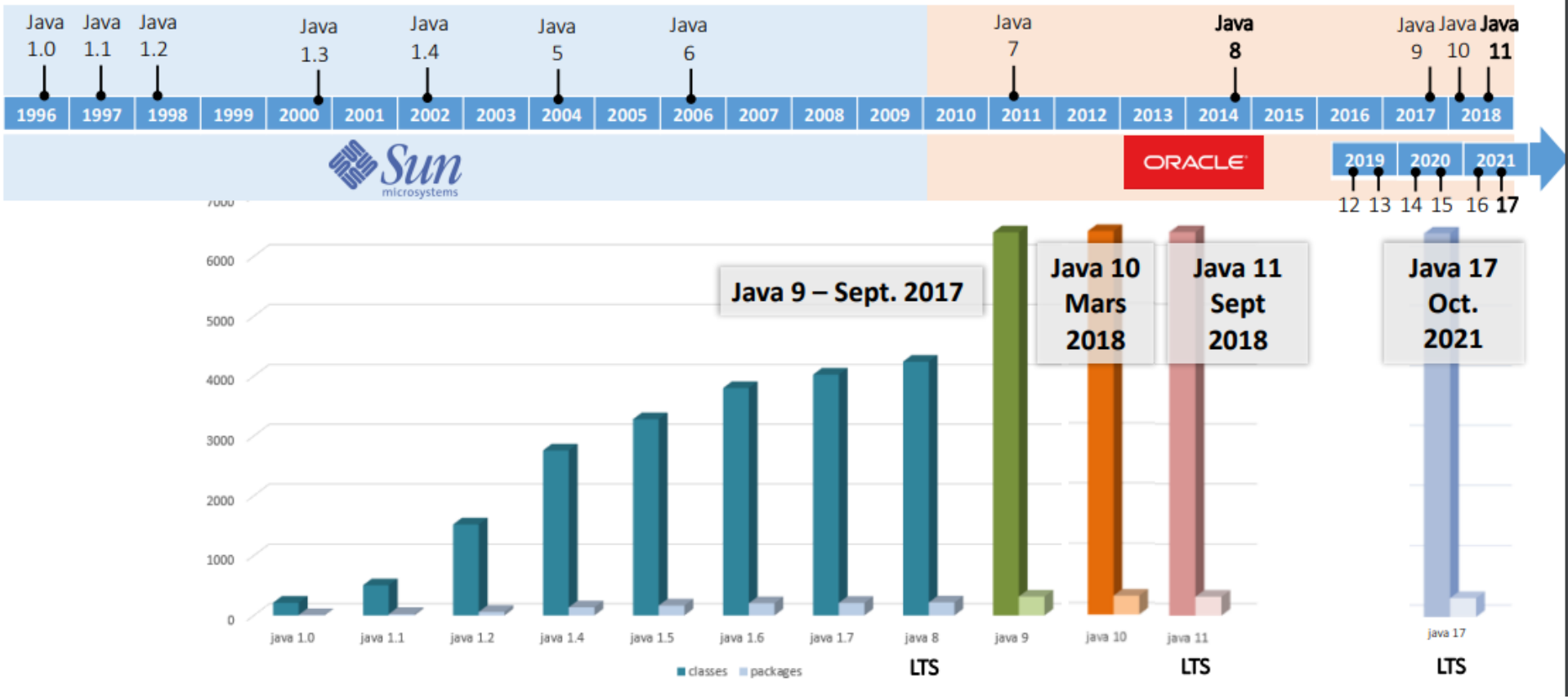
- Mars 2019 : Java SE 12
- Septembre 2019: java SE 13
- Java SE 14. Cette version est sortie le 17 mars 2020 a introduit des fonctionnalités d'aperçu telles que les enregistrements et la correspondance de modèles.
- Java SE 15 (novembre 2020)
- Java SE 16 (mars 2021) a continué à faire évoluer le langage avec de nouvelles fonctionnalités et améliorations.
- L'évolution de Java est continue, avec des mises à jour et des améliorations régulières pour le maintenir pertinent dans le développement de logiciels modernes.
- Et maintenant **Java SE 17** (septembre 2021), SE 18 (mars 2022), Java SE 19 (septembre 2022) et **Java SE 21 (septembre 2023)**

PS: versions 21, 17, 11 and 8 are the currently supported long-term support (LTS) versions

La plateforme Java

Evolution de l'API JSE

Depuis Java 9, Oracle JCP adopte une politique de release tous les 6 mois. Seules les releases LTS (Long Term Support) sont maintenues sur plusieurs années



Pourquoi utiliser Java ?

- Java fonctionne sur différentes plates-formes (Windows, Mac, Linux, Raspberry Pi, etc.)
- C'est l'un des langages de programmation les plus populaires au monde
- Il a une forte demande sur le marché du travail actuel
- Il est facile à apprendre et simple à utiliser
- Il est open-source et gratuit
- Il est sécurisé, rapide et puissant
- Il a un énorme support communautaire (des dizaines de millions de développeurs)
- Java est un langage orienté objet qui donne une structure claire aux programmes et permet de réutiliser le code, réduisant ainsi les coûts de développement
- Comme Java est proche de C++ et C#, il est facile pour les programmeurs de passer à Java ou vice versa

Que recouvre le mot Java ?

- Un langage orienté objet, simple
- Sécurité intégrée
- Portabilité
- Robustesse (typage fort, gestion de la mémoire, ...) et sûr
- Richesse des librairies de classes *(plate-forme Java)*
- Multitâche intégré au langage (*Multithreading*) et performant
- Bonne intégration des communications réseau *(Sockets, RMI, …)*
- Java ≠JavaScript
- indépendant de la machine (machine virtuelle Java) son architecture basée sur une machine virtuelle (un programme java s'exécute dans une machine virtuelle, dite machine virtuelle Java)
- Des API (Une **interface de programmation** :*Application Programming Interface)*
- Des outils (JDK)

Que recouvre le mot Java ?

Dans un des premiers papiers* sur le langage JAVA, SUN le décrit comme suit :

« *Java : a simple, **object-oriented**, distributed, robust, secure, **architecture neutral, portable**, high-performance, multithreaded, and dynamic language* »

* *White Paper :The Java Language Environment* - James Gosling, Henry McGilton - May 1996. http://java.sun.com/docs/white/langenv/

Java en 2021

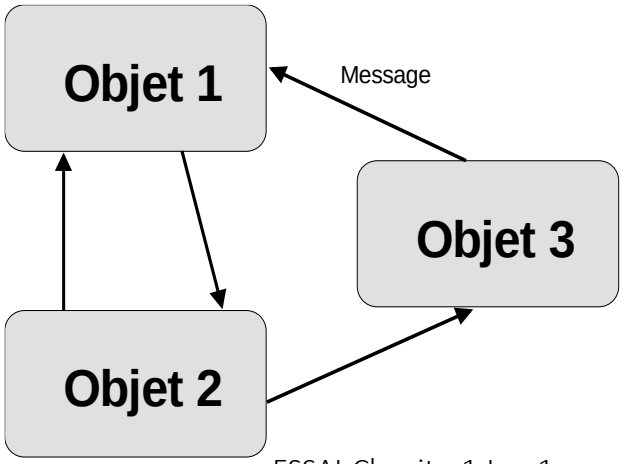
- Au top des classements :
  - http://pypl.github.io/PYPL.html
  - https://www.tiobe.com/tiobe-index/
  - http://githut.info/
- Partout : Desktop, Servers, mobile ...
- A l'INSEE (**L'Institut national de la statistique et des études économiques)**: 92.31% des applications
- Le langage le plus classe du monde

Java / Python : le jeu des 7 différences

Java	Python
Pensé POO	POO optionnelle
Typage fort	Typage faible
Compilé	Exécuté
Application	Script
Accolades {}	Indentation
Verbeux	Concis
Packaging (JAR)	Fichiers .py

Programmation Orientée Objet (POO)

- Logiciel orienté objet: Nouvelle façon de concevoir le logiciel orientée vers les données plutôt que sur les actions, les données ayant une plus longue pérennité conceptuelle. Un logiciel est alors vu comme une collection d'objets communiquant par messages.
- Qu'est ce qu'un Programme Orientée Objet?
  - Ensemble d'objets autonomes et responsables qui s'entraident pour résoudre un problème final en s'envoyant des messages.

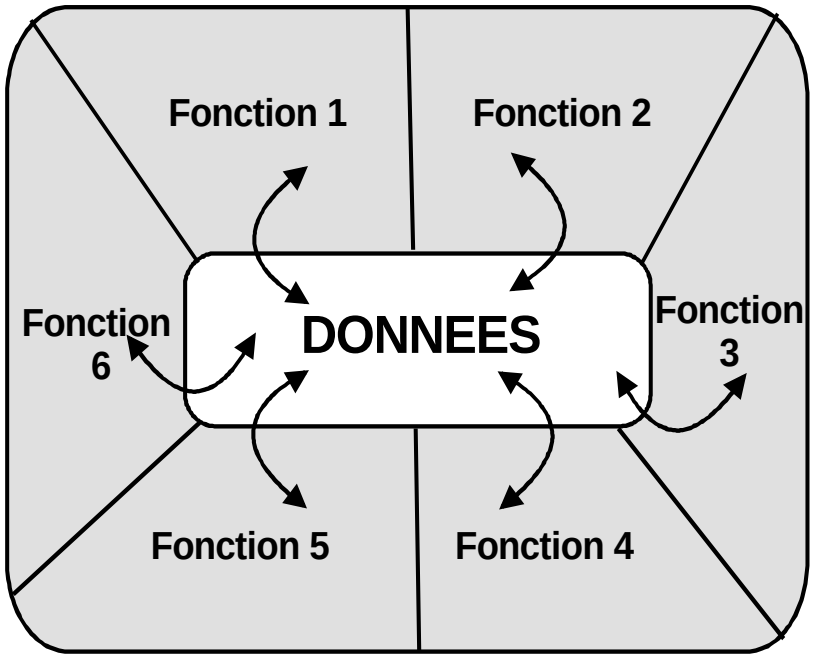


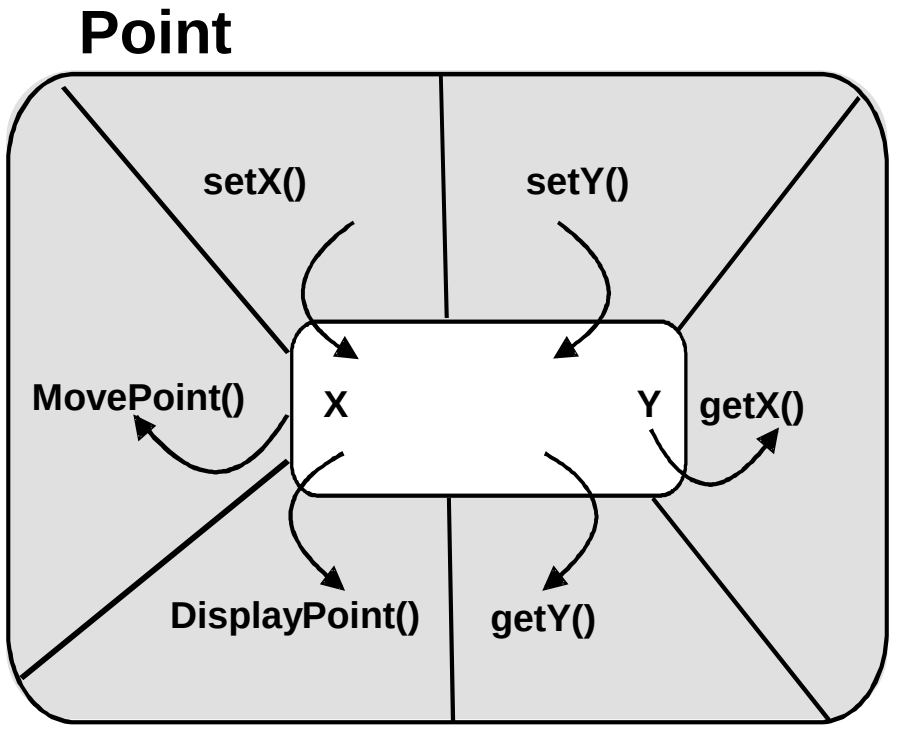
Qu’est ce qu’un objet?

- ✓Entité fondamentale rassemblant des données ainsi que le code opérant sur ces données.
- ✓Un objet communique avec son environnement par messages et répond aux principes d'abstraction de données et d'abstraction procédurale.
- ✓Un ensemble d’objets présentant les mêmes caractéristiques forme une classe.

Programmation Orientée Objet

- Objet/classe = Données + Méthodes (Fonctions Membres)





Programmation Orientée Objet

1- Concept de Classe

- Type réunissant une description
  - D'une collection de données membres hétérogènes ayant un lien entres elles
  - D'une collection de méthodes (fonctions membres) servant à manipuler les données membres
- Généralisation du type *structure* ou *record* des langages non **OO** (classe = définition de données + définition de méthodes)
- Possibilité de fixer la visibilité externe des différents constituants de la classe

Programmation Orientée Objet

2- Encapsulation des données

- Prône, d'une part le rassemblement des données et du code les utilisant dans une entité unique nommée objet, d'autre part, la séparation nette entre la partie publique d'un objet (ou interface) seule connue de l'utilisateur de la partie privée ou implémentation qui doit rester masquée.
- Exemple du Point: Les données Abscisse (X) et Ordonnée (Y) ne peuvent être utilisées que par les fonctions membres

3- Polymorphisme (du Grecque→plusieurs formes)

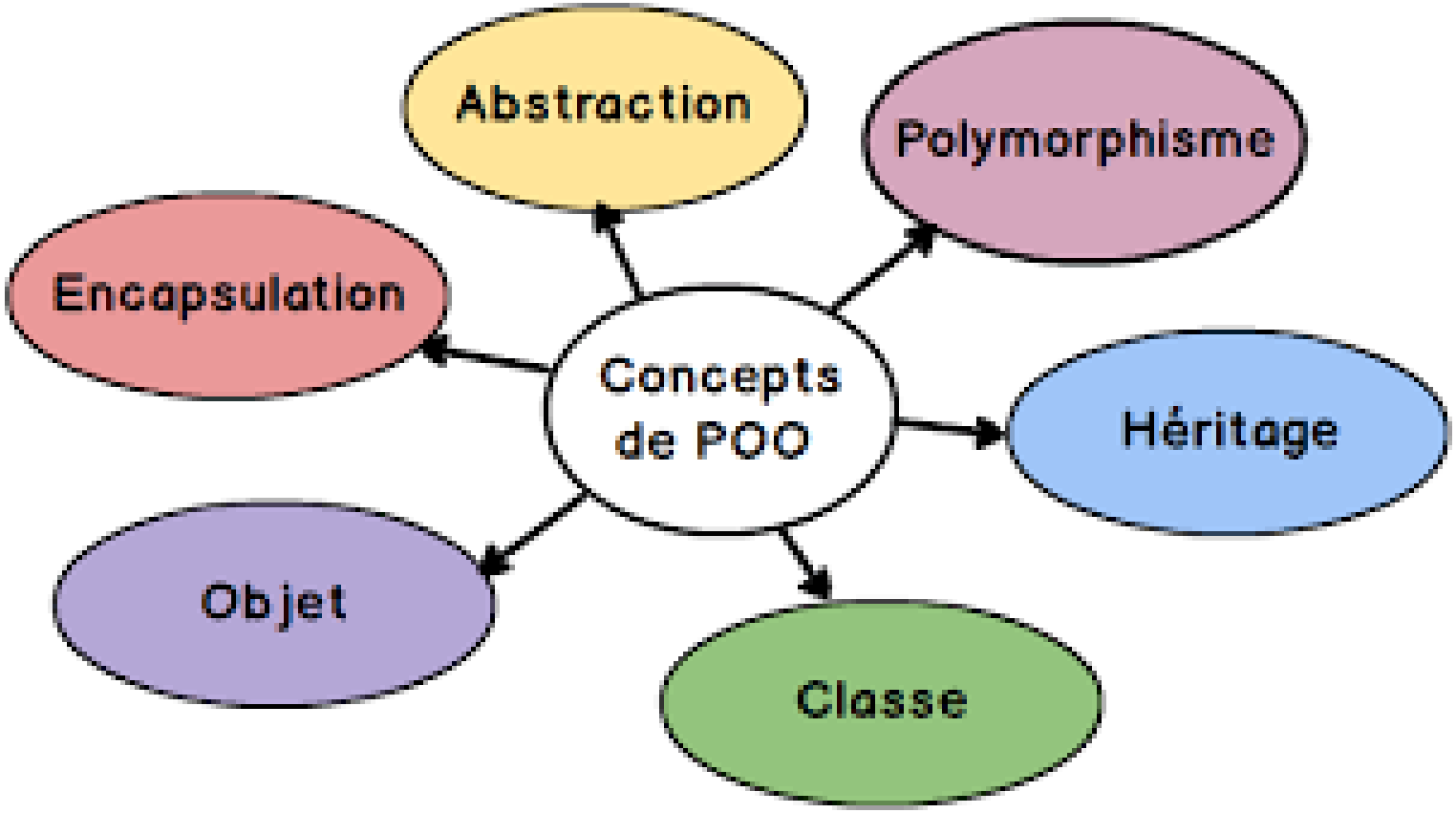
- Un nom (de fonction, d'opérateur) peut être associé à plusieurs mais différentes utilisations (surcharge/redéfinition).
- Exemple
  - Si **a** est un nombre complexe, sqrt(**a**) appellera si elle existe la fonction adaptée au type de **a**.
  - Dans un langage non OO, il aurait fallu connaître le nom de deux fonctions distinctes (selon que **a** soit complexe ou réel)
  - C'est le système qui choisit selon le type de l'argument ou de l'opérande

Programmation Orientée Objet

4- L'Héritage: Possibilité offerte par les langages orientés O de définir des arborescences de classes traduisant le principe de généralisation/spécialisation. Une relation d'héritage peut se traduire par la phrase : *« La classe dérivée est une version spécialisée de sa classe de base »*

- ➔ Permet de définir les bases d'un nouvel objet à partir d'un objet existant
- ➔ Le nouvel objet hérite des propriétés de l'ancêtre et peut recevoir de nouvelles fonctionnalités

PROGRAMMATION ORIENTÉE OBJET



Caractéristiques du langage Java (1)

- Simple
  - Apprentissage facile
    - faible nombre de mots-clés
    - simplifications des fonctionnalités essentielles
  - Développeurs opérationnels rapidement
- Familier
  - Syntaxe proche de celle de C/C++
- Orienté objet
  - Java ne permet d'utiliser que des objets *(hors les types de base)*
  - Java est un *langage objet* de la famille des langages de *classe* comme C++ ou SmallTalk
  - Les grandes idées reprises sont : encapsulation, dualité classe/instance, attribut, méthode / message, visibilité, dualité interface/implémentation, héritage simple, redéfinition de méthodes, polymorphisme

Caractéristiques du langage Java (2)

- Sûr
  - Seul le bytecode est transmis, et «vérifié» par l'interpréteur
  - Impossibilité d'accéder à des fonctions globales ou des ressources arbitraires du système
- Fiable
  - Gestion automatique de la mémoire *(ramasse-miette ou "garbage collector")*
  - Gestion des exceptions
  - Sources d'erreurs limitées
    - typage fort,
    - pas d'héritage multiple,
    - pas de manipulations de pointeurs, etc.
  - Vérifications faites par le compilateur facilitant une plus grande rigueur du code

Caractéristiques du langage Java (3)

- ✓Java est indépendant de l'architecture : Le bytecode généré par le compilateur est indépendant de toute architecture. Toute application peut donc tourner sur une plate-forme implémentant une machine virtuelle Java

« Ecrire une fois, exécuter partout »

- ✓Java est multi-tâches : Exécution de plusieurs processus effectuant chacun une tâche différente
- ✓Mécanismes de synchronisation: Fonctionnement sur des machines multiprocesseurs

Type de programme JAVA

- **Applications** Java : programmes autonomes, "stand-alone"
- **Applets** (mini-programmes) : Programmes exécutables uniquement par l'intermédiaire d'une autre application
  - navigateur web : Internet explorer, safari
  - application spécifique : Appletviewer
  - Deviennent **obsolètes (deprecated)**The Java applet API is now deprecated since Java 9 in 2017, Java 11 have disappeared

Java est souvent confondu avec le langage de script Javascript auquel il n'est en aucune manière apparenté

TYPES DE PROGRAMMES JAVA
APPLICATION INDÉPENDANTE

Application est définie par un ensemble de classes dont une jouera le rôle de classe principale

La compilation de la classe principale entraîne la compilation de toutes les classes utilisées

Pour exécuter l'application on indique à l'interpréteur java le nom de la classe principale

java charge les classes nécessaires au fur et à mesure de l'exécution

- **Application doit posséder une classe principale**
  - classe possédant une méthode de signature

public static void main(String[ ] args)

- **Cette méthode sert de point d'entrée pour l'exécution**
  - l'exécution de l 'application démarre par l'interprétation de cette méthode

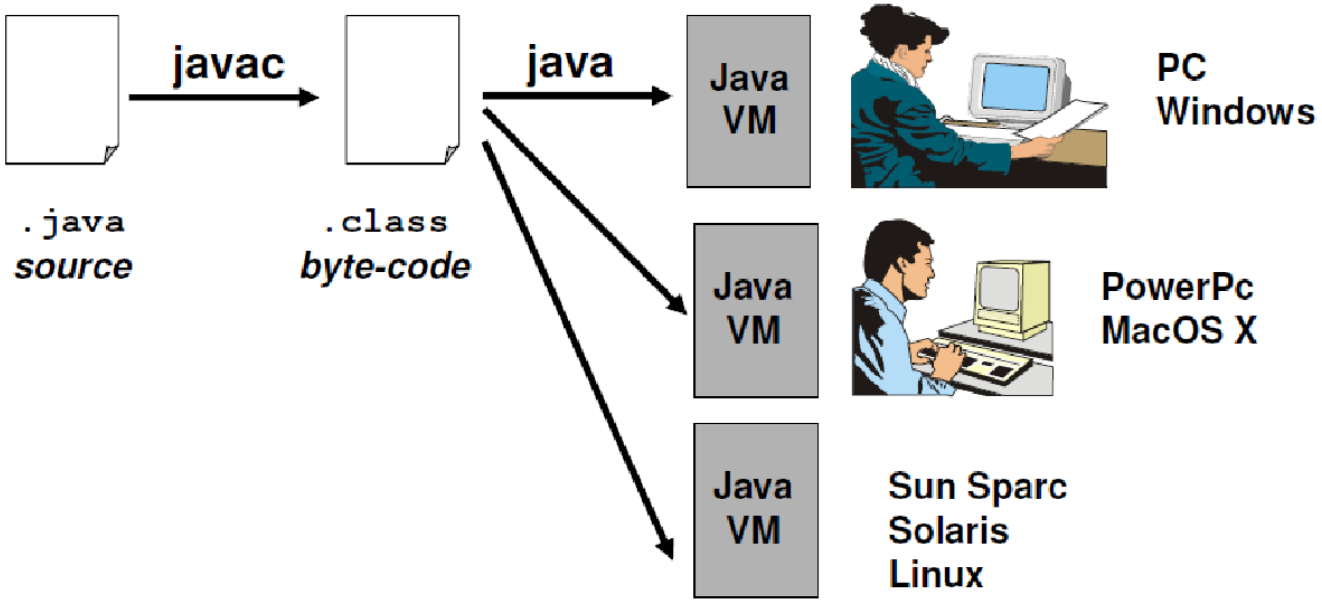
ex : java HelloWorld

Java, un langage indépendant?

- Java est un langage interprété
  - La compilation d'un programme Java crée du pseudo-code portable : le "bytecode"
  - Sur n'importe quelle plateforme, une machine virtuelle Java peut interpréter le pseudo-code afin qu'il soit exécuté
- Les machines virtuelles Java peuvent être
  - des interpréteurs de bytecode indépendants (pour exécuter les programmes Java)
  - contenues au sein d'un navigateur (pour exécuter des applets Java)

Java, un langage indépendant?

- **byte-code assure la portabilité des programmes Java**
  - langage d'une Machine Virtuelle
  - à l'exécution un interpréteur simule cette machine virtuelle



Java, un langage indépendant?

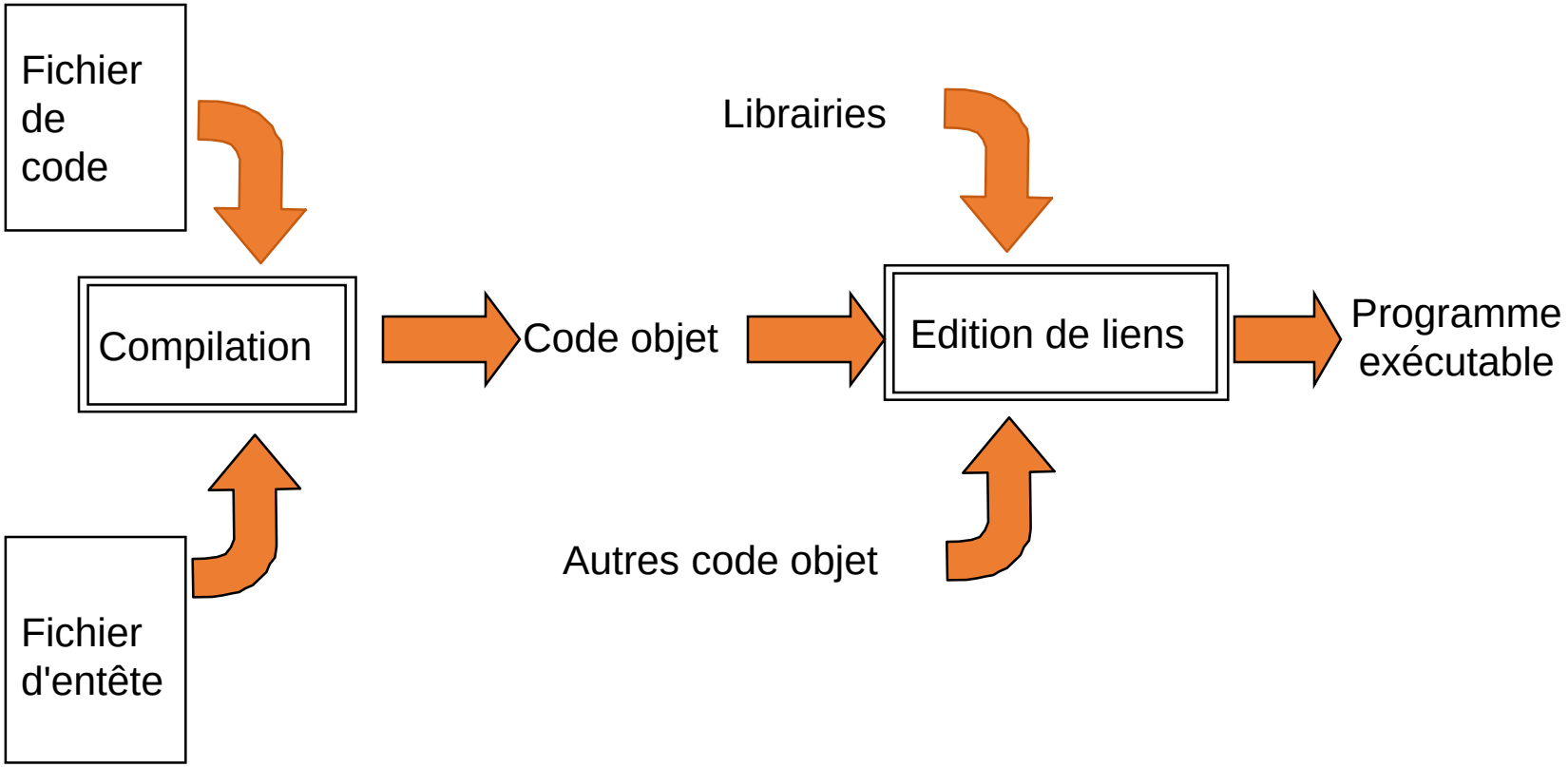
- **Avantages :**
  - **Portabilité**
    - Des machines virtuelles Java existent pour de nombreuses plates-formes dont : Linux, Windows, MacOS
  - **Développement plus rapide**
    - courte étape de compilation pour obtenir le bytecode,
    - pas d'édition de liens,
    - débogage plus aisé,
  - **Le bytecode est plus compact que les exécutables**
    - pour voyager sur les réseaux.

Java, un langage indépendant?

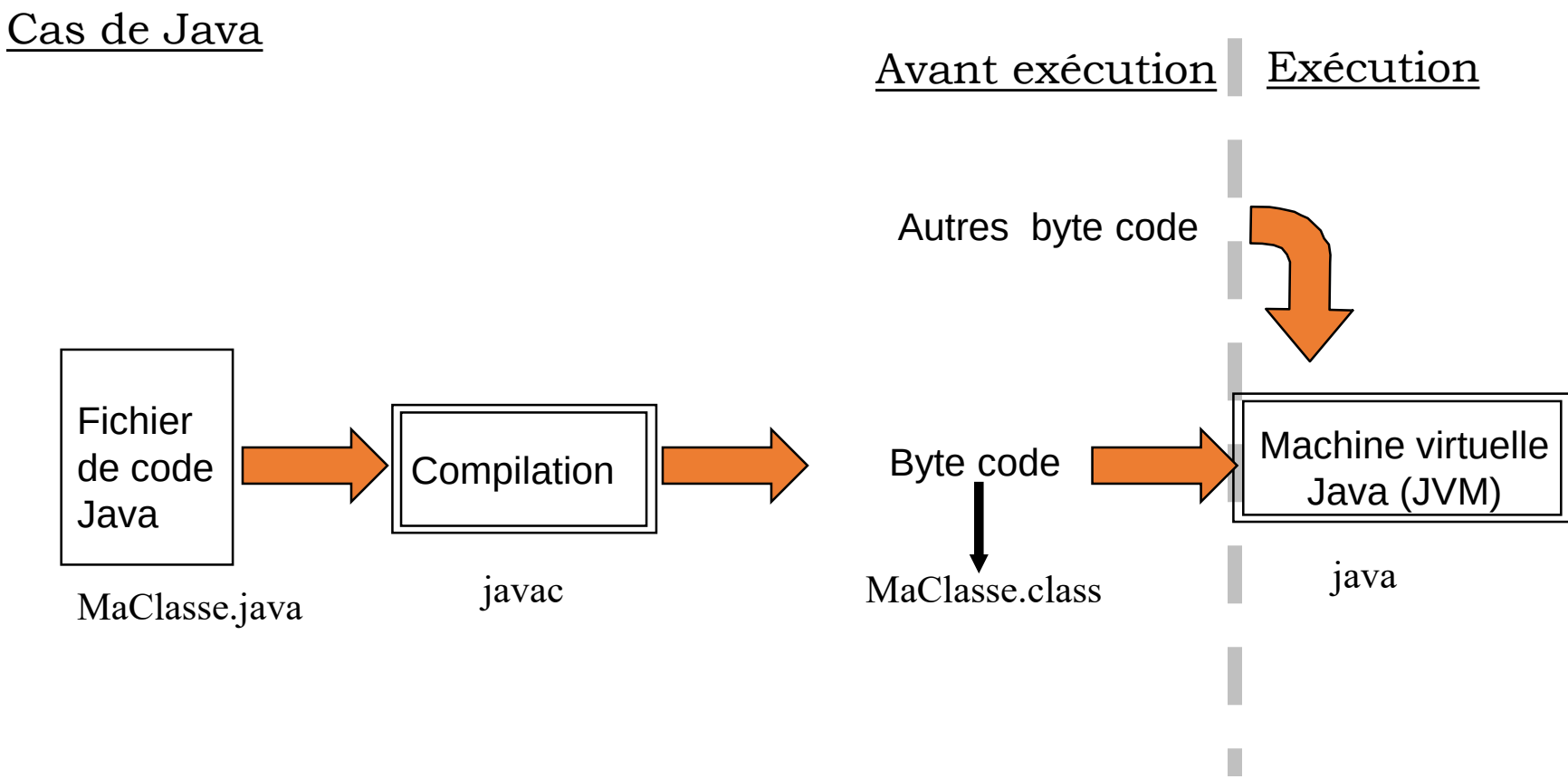
- **Inconvénients :**
  - Nécessite l'installation d'un interpréteur pour pouvoir exécuter un programme Java
  - L'interprétation du code ralentit l'exécution
  - Les applications ne bénéficient que du dénominateur commun des différentes plateformes
    - limitation, par exemple, des interfaces graphiques
  - Gestion gourmande de la mémoire
  - Impossibilité d'opérations de « bas niveau » liées au matériel

Langage compilé

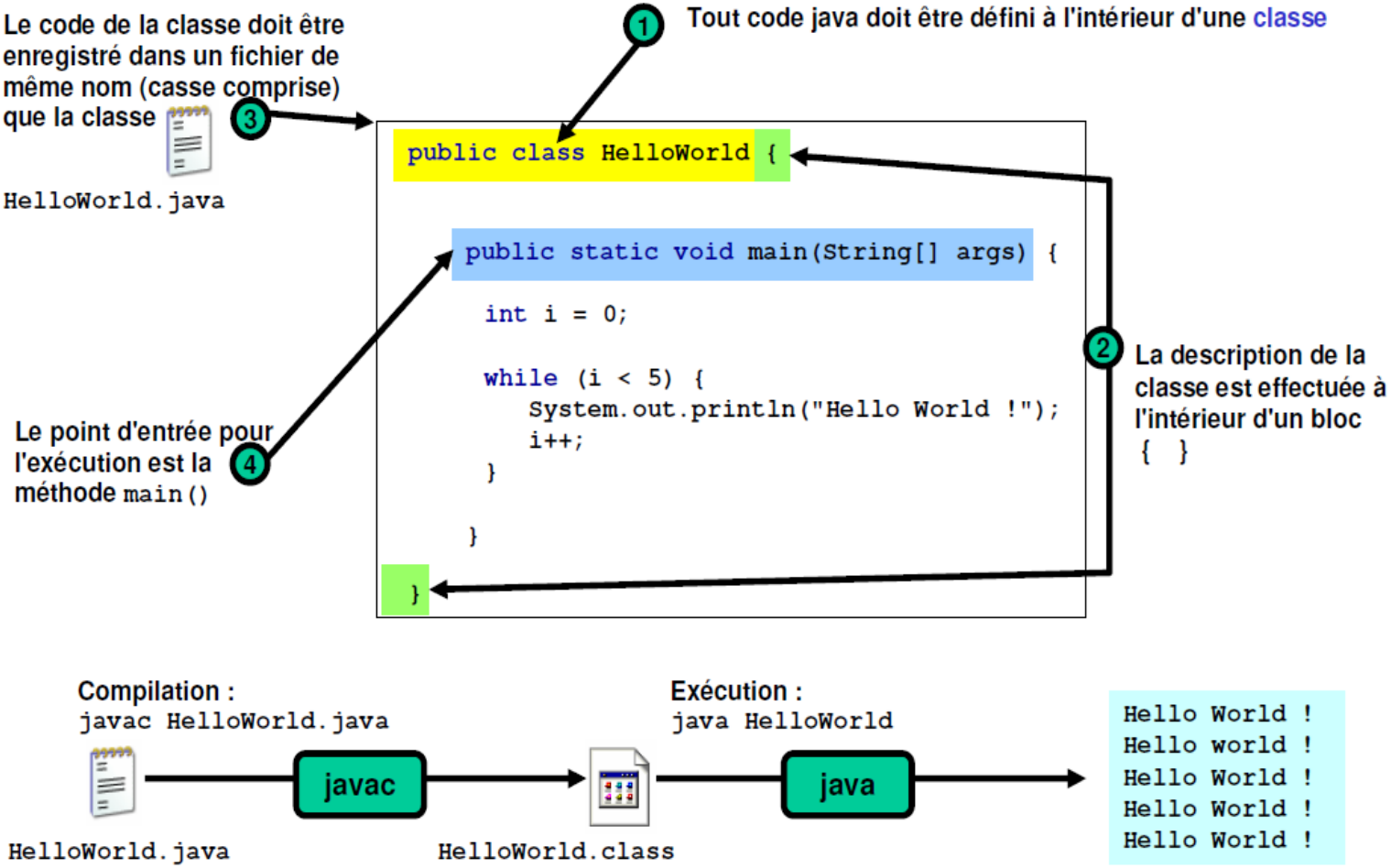
Etapes qui ont lieu avant l'exécution pour un langage compilé comme C++



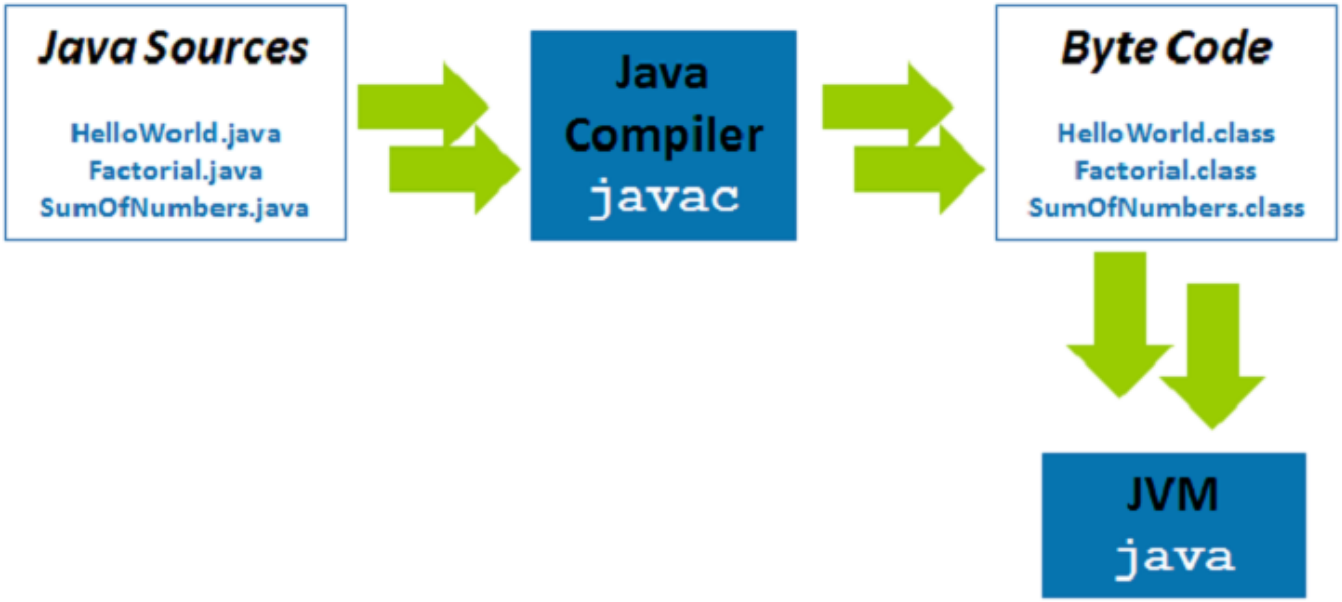
Langage interprété



LE LANGAGE JAVA: MON PREMIER PROGRAMME JAVA (PAS TRÈS OBJET…)

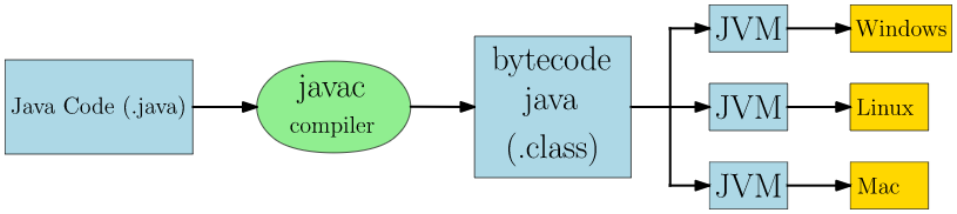


LE LANGAGE JAVA
UN LANGAGE COMPILÉ / INTERPRÉTÉ



Le concept de JVM

- "Write once, run everywhere "



- JVM = Java Virtual Machine
- JDK = Java Development Kit
- JRE = Java Runtime Environment est un progiciel qui fournit des bibliothèques de classes Java, ainsi que JVM (Java Virtual Machine), ainsi que d'autres composants permettant d'exécuter des applications écrites en programmation Java.

Le concept de JVM

- JRE:

- Lorsque vous téléchargez JDK, JRE est également téléchargé et vous n'avez pas besoin de le télécharger séparément.
- En plus de JRE, JDK contient également plusieurs outils de développement (compilateurs, JavaDoc, Java Debugger, etc.)

Outil de développement : le JDK

- Environnement de développement fourni par Oracle
- JDK signifie Java Development Kit (Kit de développement Java).
- Il contient :
  - les classes de base de l'API java (plusieurs centaines),
  - la documentation au format HTML
  - le compilateur : javac
  - la JVM (machine virtuelle) : java
  - le visualiseur d'applets : appletviewer
  - le générateur de documentation : javadoc
  - etc.

L'API de Java

- Java fournit de nombreuses librairies de classes remplissant des fonctionnalités très diverses : c'est l'API Java
  - API (Application and Programming Interface /Interface pour la programmation d'applications) : Ensemble de bibliothèques permettant une programmation plus aisée car les fonctions deviennent indépendantes du matériel.
- Ces classes sont regroupées, par catégories, en paquetages (ou "packages").

L'API de Java (2)

- Les principaux paquetages
  - java.util : structures de données classiques
  - java.io : entrées / sorties
  - java.lang : chaînes de caractères, interaction avec l'OS, threads
  - java.applet : les applets sur le web
  - java.awt : interfaces graphiques, images et dessins
  - javax.swing : package récent proposant des composants « légers » pour la création d'interfaces graphiques
  - java.net : sockets, URL
  - java.sql : fournit le package JDBC (pas abordé dans ce cours mais dans le cours java niveau 2)

L’API de Java (3)

- La documentation de Java est standard, que ce soit pour les classes de l'API ou pour les classes utilisateur
  - possibilité de génération automatique avec l’outil Javadoc.
- Elle est au format HTML
  - intérêt de l'hypertexte pour naviguer dans la documentation

L'API de Java (4)

- Pour chaque classe, il y a une page HTML contenant :
  - la hiérarchie d'héritage de la classe,
  - une description de la classe et son but général,
  - la liste des attributs de la classe (locaux et hérités),
  - la liste des constructeurs de la classe (locaux et hérités),
  - la liste des méthodes de la classe (locaux et hérités),
  - puis, chacune de ces trois dernières listes, avec la description détaillée de chaque élément.

Java un langage Objet

- Tout est classe sauf : les types primitifs (int, float, double, ...) et les tableaux.
- Toutes les classes dérivent de java.lang.**Object**
- Les bibliothèques de programmation sont réalisées par un ensemble de classes
- L'héritage est simple entre les classes
- Il est possible de réaliser des héritages multiples via les interfaces
- Tout objet est manipulé à travers une référence

La plateforme Java

Les différentes éditions de Java

3 éditions de Java

Standard Edition JSE

- **Fourni les compilateurs, outils, runtimes, et APIs pour écrire, déployer, et exécuter des applets et applications dans le langage de programmation Java**

Entreprise Edition JEE

- **Destinée au développement d'applications «d'entreprise» *(«business applications»)* robustes et interopérables. Simplifier le développement et le déploiement d'applications distribuées et articulées autour du web.**

Mobile Edition JME

- **Java ME (micro edition) Environnement d'exécution optimisé pour les dispositifs « légers » :**
  - **Carte à puce (smart cards)**
  - **Téléphones mobiles**
  - **Assistants personnels (PDA)**

Environnements de développement intégrés

Nombreux IDE (Integrated Development Environment) pour java

- Editeur syntaxique, débogueur, compilateur, exécution

- **Commerciaux**

JCreator
Xinox

WebSphere Studio
Site Developer for Java
IBM

JBuilder
Codegear

JDeveloper
Oracle

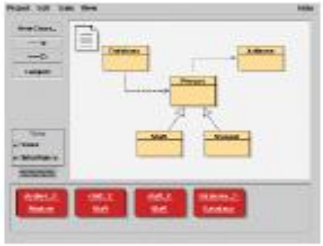
IntelliJIDEA
JetBrains

...

- **Open-source et/ou freeware**

NetBeans
www.netbeans.org

Eclipse
www.eclipse.org



BlueJ
www.bluej.org

Emacs + JDE
http://sunsite.auc.dk/jde

Références

- ✓ *« Introduction à Java », 2e édition*
- ✓ **Pat Niemeyer et Jonathan Knudsen, O'Reilly, déc. 2002**
- ✓ *«Cahiers du Programmeur - Java - 1 .4 et 5.0 »*
- ✓ **Emmanuel Puybaret, Eyrolles, mars 2006**
- ✓ « Thinking in Java »,
- ✓ **Bruce Eckel - Prentice-Hall (www.BruceEckel.com ,** www.penserenjava.free une traduction du livre de BruceEckel « Thinking in Java »**)**
- ✓ « JAVA in a nutshell, 5th Edition »,**David Flanagan - O'Reilly 2005**
- ✓ *« Java - tête la première »* **Kathy Sierra, Bert Bates,1re édition, O'Reilly, septembre 2004**
- ✓ Au cœur de Java 2 **: Volume I - Notions fondamentales. C. Hortsmann et G. Cornell. The Sun Microsystems Press. Java Series. CampusPress.**
- ✓ Au cœur de Java 2 **: Volume II - Fonctions avancées. C. Hortsmann et G. Cornell. The Sun Microsystems Press. Java Series. CampusPress.**
- ✓ Passeport pour l'algorithmique objet**. Jean-Pierre Fournier. Thomsom Publishing International.**
- ✓ **Programmer avec Java - Concepts fondamentaux et mise en oeuvre par l'exemple - collection O'Reilly** – 28 mars 2019

Références

URLs

- **https://www.oracle.com/java/technologies/ -**

Site officiel Java d'oracle, Pour récupérer le kit de développement

  - JDK, Tutoriels, Documentations, spécifications, …

- **Outils de développement**
  - **Eclipse : http://www.eclipse.org**
  - **JBuilder 5 : http://www.borland.fr/download/jb5pers/**
- **Des exemples de programmes commentés**
  - **www.technobuff.com/javatips/ (en anglais)**
- **http://www.javaworld.com**
  - Magazine électronique
- **www.developpez.com**
  - des tutoriels, des FAQ, des ressources….
- **http://penserenjava.free.fr**

  une traduction du livre de BruceEckel « Thinking in Java »