


# Mise à niveau Java

Yohan Boichut & Frédéric Moal



# Présentations

Présentation du module

Présentation du langage

# Le module Mise à niveau

But :

maîtrise des bases de Java avant le module PASD

Format :

- 5 x 4h CTD, cours avec exercices sur machine [du fun, des surprises, du suspense, ... ]
- 3 x 4h TP, exercices en semi-autonomie

Intervenants :

Frédéric Moal, Yohan Boichut, Vivien Pelletier

[frederic.moal@univ-orleans.fr](mailto:frederic.moal@univ-orleans.fr), [yohan.boichut@univ-orleans.fr](mailto:yohan.boichut@univ-orleans.fr), [vivien.pelletier@univ-orleans.fr](mailto:vivien.pelletier@univ-orleans.fr)



# Kahoot !

<http://kahoot.it>



# Le langage Java

- Langage orienté objet, à classes (les objets sont décrits/regroupés dans des classes)
- de syntaxe proche du langage C
- fourni avec le JDK (Java Development Kit) :
  - outils de développement
  - ensemble de paquetages très riches et très variés
- portable grâce à l'exécution par une machine virtuelle : « Write once, run everywhere »

# Quelques propriétés

- multi-tâches (thread)
- sûr
  - fortement typé
  - nombreuses vérifications au chargement des classes et durant leur exécution
- adapté à Internet
  - chargement de classes en cours d'exécution (le plus souvent par le réseau : RMI)
  - facilités pour distribuer les traitements entre plusieurs machines (sockets, RMI, EJB, ...)

# Un premier programme

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
  
        System.out.println("Hello world");  
  
    }  
  
}
```

- La classe HelloWorld est public, donc le fichier qui la contient doit s'appeler (en tenant compte des majuscules et minuscules) HelloWorld.java

# Compilation

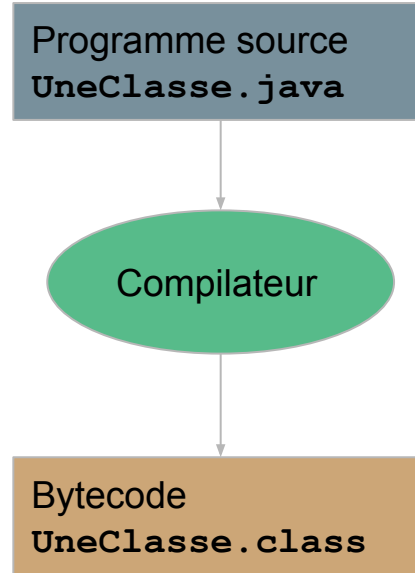
- En Java, le code source n'est pas traduit directement dans le langage de l'ordinateur
- Il est d'abord traduit dans un langage appelé « bytecode », langage d'une machine virtuelle (JVM ; Java Virtual Machine) définie par Oracle
- Ce langage est indépendant de l'ordinateur qui va exécuter le programme



# La compilation construit du bytecode

Programme écrit en Java

Programme en bytecode,  
indépendant de l'ordinateur



# Compilation avec javac

- Oracle fournit le compilateur javac avec le JDK
- `javac HelloWorld.java`

crée un fichier « `HelloWorld.class` » qui contient le bytecode, situé dans le même répertoire que le fichier « `.java` »

- Le fichier à compiler peut être désigné par un chemin absolu ou relatif :

`javac util/Liste.java`

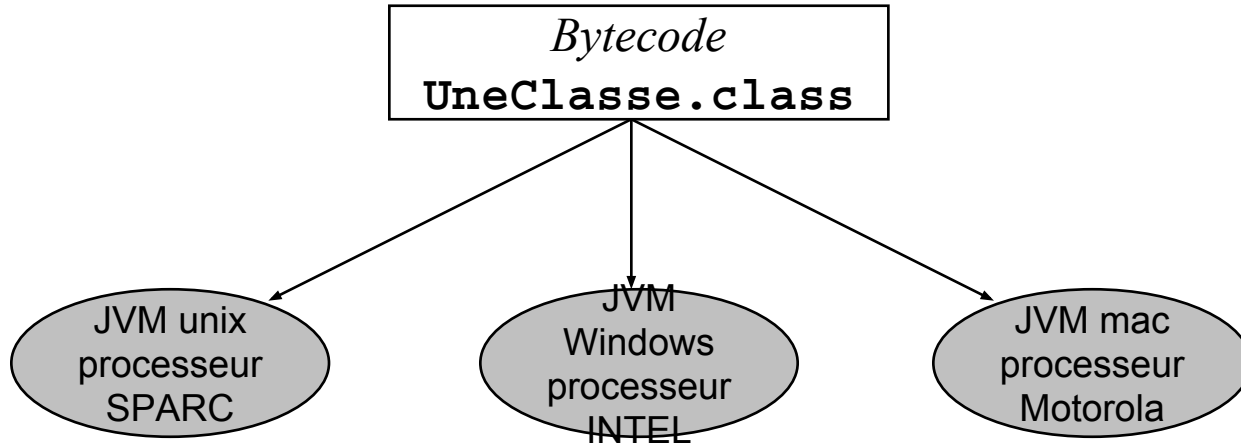
cf chapitre env. et packages

# Exécution du bytecode

- Le bytecode doit être exécuté par une JVM
- Cette JVM n'existe pas ; elle est simulée par un programme qui interprète le bytecode :
  - lit les instructions (en bytecode) du programme .class,
  - les traduit dans le langage natif du processeur de l'ordinateur
  - lance leur exécution

# bytecode portable : run everywhere

Si un système possède une JVM, il peut exécuter tous les fichiers .class compilés sur n'importe quel autre système



# Exécution du bytecode avec le JDK (JRE)

- Oracle fournit le programme java qui simule une JVM

- java HelloWorld



*Nom d'une classe :  
pas de suffixe .class !*

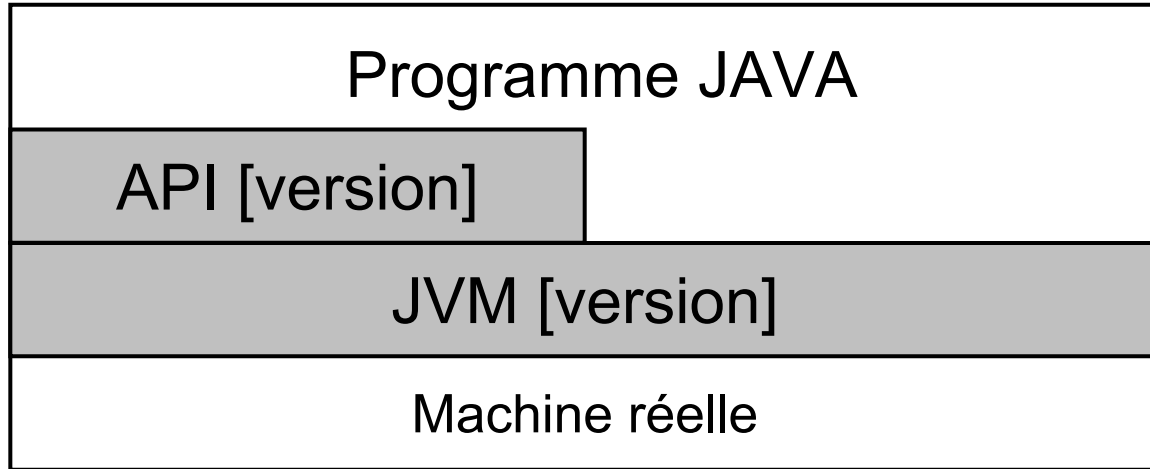
exécute le bytecode de la méthode main de la classe HelloWorld

- HelloWorld est un nom de classe et pas un nom de fichier :
  - on ne peut pas donner un chemin
  - pas de suffixe .class

# Spécifications

- Java, c'est en fait
  - le langage Java : <http://java.sun.com/docs/books/jls/>
  - une JVM : <http://java.sun.com/docs/books/vmspec/>
  - les API : selon la documentation javadoc fournie avec les différents paquetages
- Java n'est pas normalisé ; son évolution est gérée par le JCP (Java Community Process ; <http://www.jcp.org/>) dans lequel Oracle tient une place prépondérante

# Plateforme Java



# 3 éditions

- Java SE : Java Standard Edition ; JDK = Java SE Development Kit
- Java EE : Enterprise Edition qui ajoute les API pour écrire des applications installées sur les serveurs dans des applications distribuées : servlet, JSP, JSF, EJB,...
- Java ME : Micro Edition, version pour écrire des programmes embarqués (carte à puce/Java card, téléphone portable,...)



# Plusieurs versions

actuellement Java SE 8

Attention, Java est passé directement de la version 1.4 à la version 5.0, puis 6, 7, 8...

Cours essentiellement sur Java 7 (compatibilité ascendante)

# Notre environnement de travail

- Éditeur de texte (...)
- Compilateur (javac)
- Interpréteur de bytecode (java)
- Aide en ligne sur le JDK (sous navigateur Web)
- ...
- un IDE : IntelliJ <https://www.jetbrains.com/idea/>
  - version gratuite [Community] et payante [Ultimate], licences accessibles dans les salles de TP de la fac
  - pour chez vous : <https://www.jetbrains.com/shop/eform/students> inscription avec votre mail de l'Université

# Bibliographie

- Tutos d'Oracle : <https://docs.oracle.com/javase/tutorial/>
- Thinking in Java (<http://mindview.net/Books/TIJ/>) / Penser en Java (<http://penserenjava.free.fr/>)
- Livre : Core Java, Prentice-Hall (vol. 1) / Au coeur de Java, Campus Press
- Livre : Java in a Nutshell, O'Reilly
- Livre : Java tête la première / Head First Java, O'Reilly
- Le doudoux (<http://www.jmdoudoux.fr/>, 3413 pages)
- Internet :
  - developpez.com
  - dzone.com [refcardz]
  - slides de Richard Grin, Miage Nice
  - les cast codeurs
  - java code geek, java ranch, javaworld, InfoQ, ... Stack Overflow !
  - Javarevisited (<http://javarevisited.blogspot.fr>)
  - twitter / youtube (devox FR)

# A vous de jouer...

- lancer un terminal
- créer un répertoire manj
- créer un fichier texte  
Hello.java
- compiler Hello.java
- exécuter le main de Hello