

Programmation avancée – Java

ISIMA 2020/2021



Olivier Goutet
o.goutet@openium.fr

29 septembre 2020

Objectifs

- Rappels sur le langage Java et la POO
- Les collections génériques
- Les interfaces graphiques avec Swing/JavaFx
- La sécurité
- Les threads (parallélisme et partage de ressources)
- Développement et déploiement

Notation

- Présence
 - Obligatoire
- Attention en cours
 - Obligatoire
- TP
 - Chaque TP doit être rendu
 - Noté au hasard
 - Si une personne du groupe ne rends pas son TP. Le TP sera noté pour tout le groupe et zéro pour ceux qui ne l'ont pas rendu
- Examen final

Plan

Le langage Java

Notions de base et syntaxe

Plan

Le langage Java

- Java

- Machine virtuelle

- Premier programme

Notions de base et syntaxe

Le langage Java

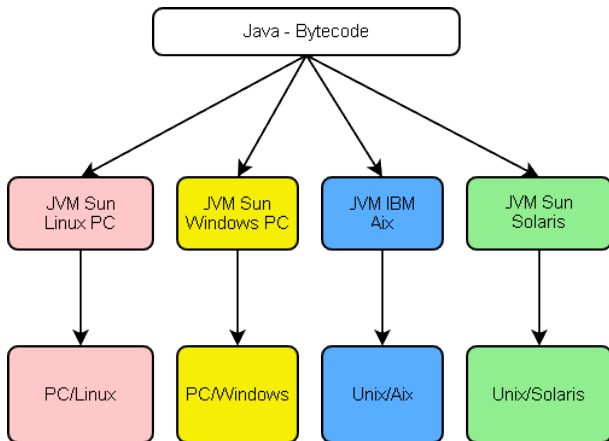
- Portable
- Sécurisé
- Simple
- Performant

Langage

- Proposé par Sun microsystems en 1995
- Syntaxe proche du C++
- Tout est objet (encapsulé dans des classes)
- Pas de pré-processeur (~~#define~~)
- Pas de désallocation (~~free~~, ~~delete~~)
 - Garbage collector
- Pas de pointeur
 - Références

Machine virtuelle

- Clé de la portabilité



Machine virtuelle

- Portabilité
 - Un programme Java : navigateur, client lourd...
- Presque aussi performante que du code compilé
- Pour gagner en performance
 - Interprétation et compilation à la volée (JIT)
 - Compilation adaptative

Hello World

```
import java.io.*;

/* ma premiere classe */
public class Exemple {
    public static void main(String[] argv) {
        // afficher un message
        System.out.println("Hello World");
    }
}
```

- Méthode main
 - Obligatoire en mode standalone
 - Point d'entrée unique du programme

```
java {-cp <classpath>} Exemple parametres_ligne_commande
```

Plan

Le langage Java

Notions de base et syntaxe

- Fichier source

- Processus de compilation et lancement

- Package

- Syntaxe

- Concepts de base

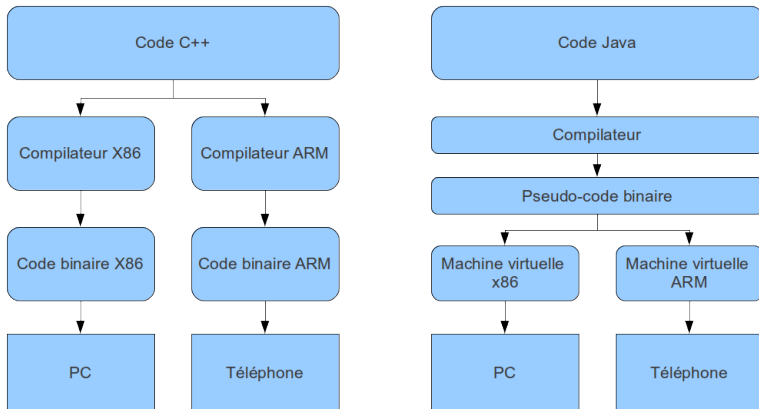
- Tableaux

- Chaînes de caractères

Fichier source

- Extension : .java
- Nom du fichier = nom de la classe publique
- Respecter la casse (**E**xemple.java)
- 1 classe publique par fichier
- Toutes les classes du même fichier sont *friend*
- Pas de point virgule en fin de classe
- Mélange déclaration + implémentation + commentaires

Compilation



Compilation

- Fichier compilé : .class
- Pseudo-code (byte-code)
- != Code machine
- Compilateur
 - `javac {-classpath <classpath>} Exemple.java`
 - `man javac`

Exécution

- `java {-classpath <classpath>} Exemple.class`
- Pseudo-code interprété par la Java Virtual Machine

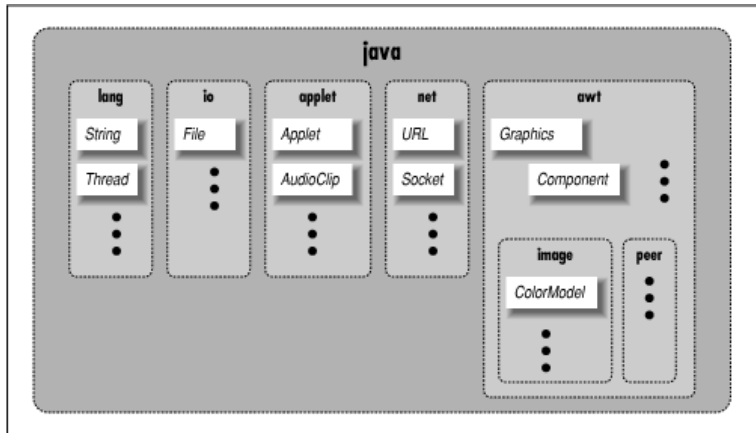
Java ARchive

- Zip contenant
 - Définitions de classes java
 - Métadonnées
 - Fichiers
- Sert à la distribution d'un programme
- `jar [opt] fichier.jar pointEntree c1.class c2.class`
- `man jar`

Package

- Ensemble de classes/fichiers rassemblés pour une finalité (besoin fonctionnel)
 - Equivalent namespace C++
- Rangement des classes Java
- Un package = un répertoire
- Nom en chemin d'un package
 - Mécanisme arborescent comme les répertoires
 - Séparateur : . (fr.isima.monpackage)
- Comment retrouver le chemin ?
 - Classpath

Package Java standard



- isima/
 - ui/
 - UiProcessing.java
 - UiSettings.java
 - model/
 - Image.java
- Utilisation d'un package
 - Directement `new isima.ui.UiSettings();`
 - Importation de toutes les classes d'un package
 - `import isima.ui.*;`
 - `new UiProcessing();`
 - `new UiSettings();`
- Importation d'une classe d'un package
 - `import isima.ui.UiProcessing;`
 - `new UiProcessing();`
 - `new isima.ui.UiSettings();`

Syntaxe

- Comme en C
- Indentation différente :

```
/* Commentaires sur plusieurs  
* lignes  
*/  
if (4 == i) {  
    system.out.println("Egal");  
} else {  
    system.out.println("Non_Egal");  
}
```

- Quelques particularités

```
for (Client c : listClient) {  
    // for each  
}
```

20 / 32

Concepts de base

- Java est un langage orienté objet : tout appartient à une classe
- Pour accéder à une classe il faut en déclarer une instance de classe ou objet.
- Une classe comporte sa déclaration, des attributs et la définition de ses méthodes.

Types d'attributs / variables

- Objet
 - Prédéfini ou utilisateur
- Chaîne de caractères
 - String
 - Manipulation par références
- Scalaires / Primitives
 - entier / réel / booléen
 - caractère
- Pour l'efficacité

Types de données scalaires

- char
 - Type caractère
 - != String
- boolean
 - true
 - false
- Entiers
 - byte
 - short
 - int
 - long
- Réels
 - float
 - double

Types de données scalaires

- char (16 bits)
 - Type caractère
 - != String
- boolean (flag)
 - true
 - false
- Entiers
 - byte (8 bits)
 - short (16 bits)
 - int (32 bits)
 - long (64 bits)
- Réels
 - float (32 bits)
 - double (64 bits)

Types de données scalaires

- Normalisé (portabilité)
- Objets equivalents
 - Byte
 - Integer
 - Short
 - ...
- Méthodes
 - toString(), parseInt(), conversions...

Déclaration de variables

```
MaClasse objet;  
int i=0;  
char c = 'A';  
String s;  
double[] tableau;
```

- N'importe où dans le bloc
- Variable locale non initialisée

Tableaux

```
// creation d'un tableau de 10 entiers  
int[] t1 = new int[10];  
  
// declaration d'un tableau (vide)  
int[] t2;  
  
for(int i=0; i<t1.length; ++i){  
    System.out.println(t1[i]);  
}
```

- Taille fixe, donnée par le champ length
- Premier indice de tableau : 0
- Il y a une vérification des indices
 - Exception : `OutOfBoundsException`

Tableaux

- Initialisation du tableau
 - Par des valeurs scalaires (0 pour des int)
 - null pour des objets
- Tableaux multidimensionnels

```
String[][] chaines = new String[10][5]  
Color[][][] cubeRGB = new Color[3][3][3];
```

Chaîne de caractères

- String != char[]
- Statique
 - String
- Dynamique
 - StringBuffer
 - StringBuilder
- Méthodes incluses
 - Comparaisons : equals(), compareTo()
 - Recherche : indexOf(), contains()...
 - Extraction : substring(), split(), regex...

Chaîne de caractères

- Création de chaîne

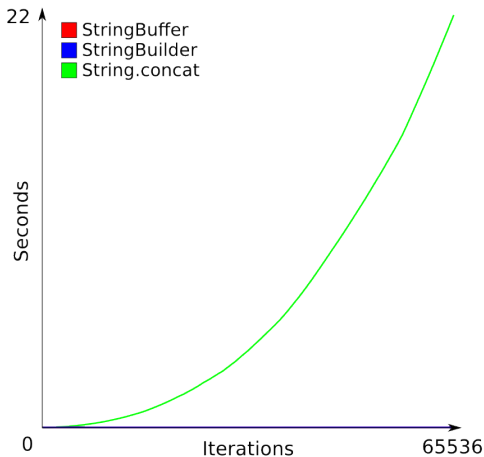
```
String s1 = "hello";  
String s2 = new String("hello");  
String s3 = null;
```

- Que se passe-t-il ?

```
String s3 = s1 + " " + s1;
```

Chaîne de caractères

Ajouter 65535 fois une lettre à une String



Références utilisées pour le cours

- [http ://fr.wikipedia.org](http://fr.wikipedia.org)
- [http ://stackoverflow.com](http://stackoverflow.com)
- Introduction à Java (Pat Neimeyer et Jonathan Knudsen, éditions O'Reilly)
- Cours de Java de Loic Yon (ISIMA)
- [http ://kaioa.com/](http://kaioa.com/) pour le schéma StringBuilder/StringBuffer/String