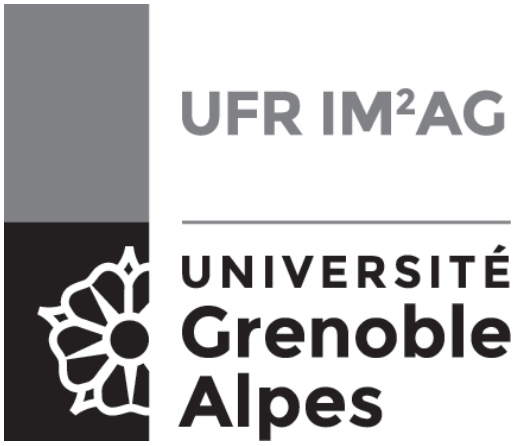


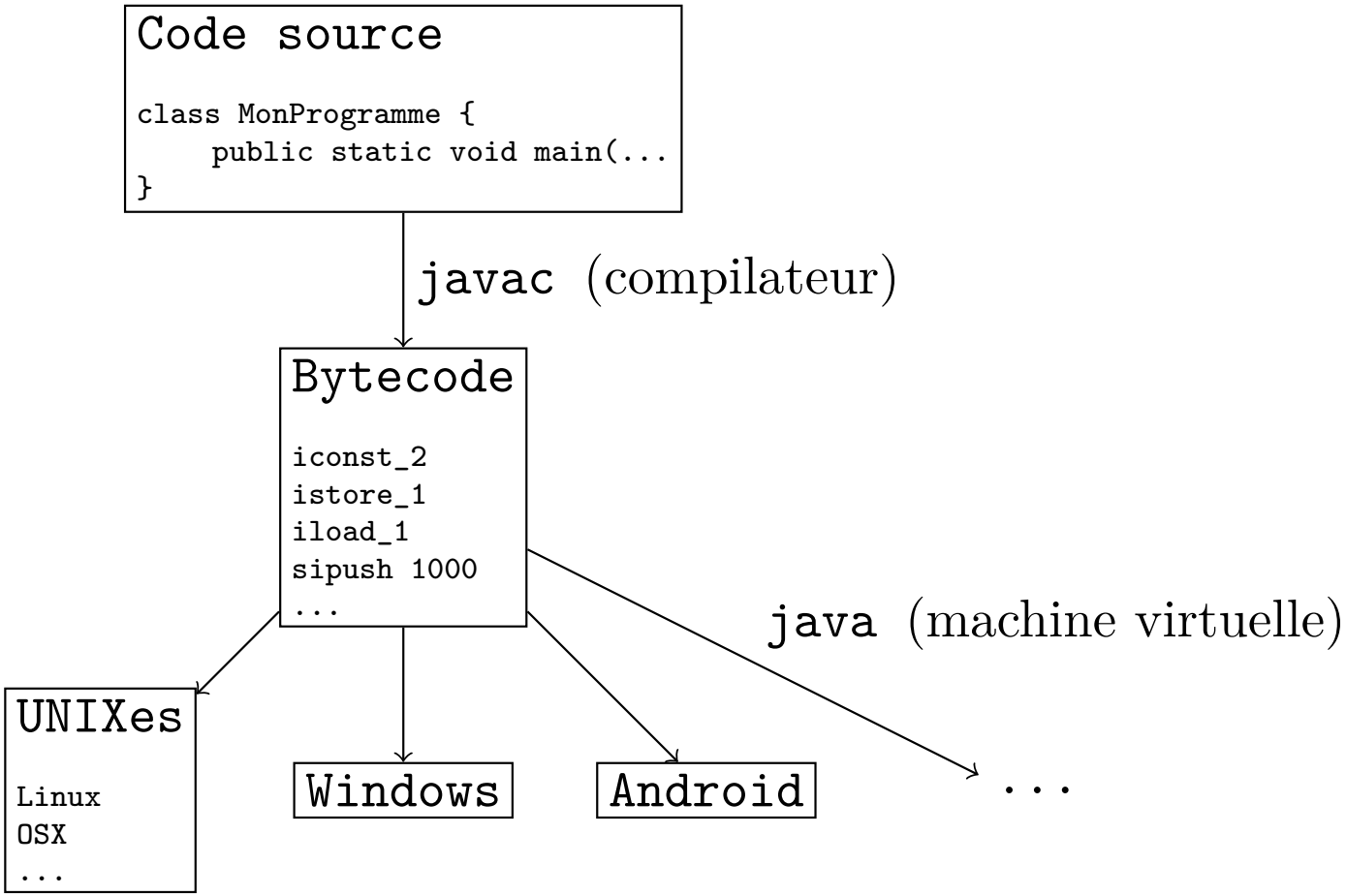
DU ISN - Présentation java, tableaux

Guillaume Huard

Retour sur le langage java

Origines

Créé par James Gosling and Bill Joy dans les années 90
Evolution d'Oak (systèmes embarqués) pour l'internet et le web



Caractéristiques

Moderne

- orienté objet
- exceptions
- généricité
- reflexivité
- threads

Simple (trop ?)

- pas de pointeurs (ni arithmétique, ni adresses, *garbage collector*)
- pas d'héritage multiple
- pas de surcharge d'opérateurs
- un seul schéma d'allocation mémoire

Equilibré

- "efficace" (compilation *just-in-time* adaptative)
- "robuste et sur" (machine virtuelle et politiques de sécurité)
- "portable" (bytecode interprété, indépendant de l'architecture)

#** Compilation et exécution

LaTeX

Tableaux

Définition

Structure de données contenant N éléments de même type

- N est la taille du tableau
- chaque élément est associé à un indice parmi $0, \ldots, N-1$
- chaque indice est associé à un unique élément

Création

```
// tab est une référence (position en mémoire)
// ici c'est une référence à un tableau d'entiers
// nous reviendrons sur cette notion de référence
int [] tab;
// création d'un tableau de 10 entiers
tab = new int[10];
```

Accès aux éléments d'un tableau

```
// On met i au carré dans la case d'indice i
for (int i=0; i<10; i++)
      tab[i] = i*i;
// On affiche les éléments
for (int i=0; i<10; i++)
      System.out.print(tab[i]+" ");
System.out.println();
```

Infos utiles

- l'élément d'indice *i* est aussi appelé case d'indice *i*
- la taille d'un tableau est constante

Caractéristiques d'un tableau

Un tableau est un objet (nous reviendrons sur les objets)

- attribut `length` contenant sa taille
- méthodes
  - `clone()` pour en créer une copie
  - `toString()` pour permettre à java de l'afficher

Exemple

```
// On met i au carré dans la case d'indice i
for (int i=0; i<tab.length; i++)
      tab[i] = i*i;
// java utilise toString pour afficher un objet
System.out.println(tab);
// Mais n'affiche pas les éléments :'(
// Nous reviendrons sur toString...
```

Accès aux éléments d'un tableau (2)

Les indices forment un intervalle avec

- une borne inférieure (0)
- une borne supérieure (taille - 1)

Tout accès en dehors des bornes lève une exception (erreur)

```
Exception in thread "main"
        java.lang.ArrayIndexOutOfBoundsException: -1
at Exemple.main(Exemple.java:7)
```

Avantages / inconvénients

- aide fortement à éviter les bugs
- léger surcout à l'exécution

Manipulation par référence

Un tableau est manipulé par référence

- la variable associée ne désigne qu'un endroit en mémoire
- plusieurs variables peuvent désigner le même endroit

```
// Création + initialisation à la déclaration
int [] tab = {1, 42};
int [] tab2;
int [] tab3;

tab2 = tab;
tab3 = tab.clone();
for (int i=0; i<tab2.length; i++)
      tab2[i] = i+42;
System.out.println(tab[0] + " et " + tab3[1]);
```

Manipulation par référence (2)

Un tableau est manipulé par référence

- on peut passer un tableau en paramètre d'une méthode
- la méthode reçoit une copie de la référence

```
static void echange(int [] t) {
        int tmp;
        tmp = t[0];
        t[0] = t[1];
        t[1] = tmp;
}

public static void main(String args[]) {
        int [] tab = { 1, 42 };

        echange(tab);
        System.out.println(tab[0]);
}
```

Exemple

Génération d'un tableau d'entiers aléatoires cf. ExempleTableau.java