

# **Informatique - Java**

Nicolas Rousset

# Passage par valeur ou par référence

# Types primitifs et type objet

Il existe deux catégories de type en Java :

- les types primitifs (int, long, double, char)
- les types objets (tous les autres)

Par convention les noms des primitifs commencent par une minuscule, ceux des types objets par une majuscule.

Les classes définies par l'utilisateur sont toujours de type objet. Il n'est pas possible de définir de nouveaux types primitifs.

# int VS Integer

Les Integer sont donc l'équivalent objet des int. Pourquoi existent-ils ? Entre autres parce que certaines manipulations nécessitent un type objet, on ne peut utiliser les containers génériques usuels avec un int :

```
Map< ?, ? >  
Set< ? >  
List< ? >
```

Tous ces types utiliseront donc le type Integer.

# Notion de référence

Toutes les variables informatiques sont stockées en mémoire, et il est important de distinguer deux choses :

- l'adresse de la variable dans la mémoire
- la valeur de la variable stockée à cette adresse

Deux variables dont les valeurs sont stockées dans le même endroit de la mémoire sont dites **identiques** (notion d' **identité**).

Comme elles désignent le même emplacement mémoire, la même valeur y est stockée.

Par contre, deux variables peuvent désigner des endroits différents qui contiennent la même valeur.

La variable sont dites alors **égale** mais ne sont pas **identiques**.

# Passage par valeur ou par référence

Il existe deux façons de passer une variable à une fonction en informatique :

- passage par valeur, on fait une copie de la valeur de la variable, qui ne peut donc pas être modifiée par l'effet de la fonction
- passage par référence, on ne copie la valeur de la variable, mais on donne son adresse mémoire. La variable peut donc (potentiellement) être modifiée.

Les types primitifs sont passés par valeur, les types objets par référence.