

Langages objets

Les types primitifs

M2 Pro CCI, Informatique
Emmanuel Waller, LRI, Orsay

- Vu : int, double
- Types primitifs : manipuler entiers, flottants, caractères, booléens
- Seuls types du langage qui ne sont pas des classes

- La notion de type
- Les types entiers : byte, short, int, long
- Les types flottants : float, double
- Le type caractère
- Le type booléen
- Initialisation et constantes
 - Variables non initialisées

la notion de type

- mémoire, octets, adresses
- comprendre octet nécessite savoir quel codage utilisé

les types entiers (entiers relatifs)

- représentation mémoire des entiers : un codage
- les différents types d'entier
 - type, taille (octets), valeur minimale, valeur maximale
 - byte, 1, -128 (Byte.MIN_VALUE), 127 (...MAX_...)
 - short, 2, -32 768, 32 767
 - int, 4, -2 147 483 648, 2 147 483 647
 - long, 8, 19 chiffres..., idem
- taille ne dépend pas de l'environnement et de l'architecture (JVM)

- choix : selon application (le plus proche des besoins)
- notation des constantes entières : usuelle (+123, -4567), hexadécimale, octale

les types flottants

- les différents types et leur représentation en mémoire
 - un nombre réel r est représenté par : signe, mantisse, exposant : $s M \cdot 2^E$, qui est une approximation de r (il n'existe pas nécessairement M et E dans Java pour r)
 - précision : limitée par erreur de troncature (ex : M et E trop petits, ou r nombre infini de chiffres après virgule)
 - domaine couvert : ensemble des réels représentés

- les différents types flottants
 - type, taille (octets), précision (chiffres significatifs), valeur absolue minimale, valeur absolue maximale
 - float, 4, 7, $-1.40239846E-45$ ($\sim 10^{-45}$) (Float.MIN...), $\sim 3.10^{38}$
 - double, 8, 15, $\sim 5.10^{-324}$, $\sim 2.10^{308}$
- CCI-LO
 - flottants et erreurs : ça marche souvent, mais prudence, domaine de recherche en informatique
 - détails pour calcul scientifique : voir documentation et livre(s) selon besoins
 - utiliser double, pas float

- notation des constantes flottantes
 - décimale : 12.34, -0.38, 4., .27, 0. (4 : entier)
 - exponentielle : 4.25e-4, etc.

le type caractère

- 2 octets : Unicode (65 536 combinaisons)
- écriture : 'a', 'E', '+' (CCI-LO : éviter caractères bizarres)
- subtilités

le type booléen

- valeurs logique vrai/faux : true, false
- ex :

```
int n, p;
```

```
boolean b;
```

```
b = (n < p);
```

initialisation et constantes

- initialisation d'une variable

- `int n = 15;`

- équivalent à : `int n; n = 15;`

- `n` modifiable ensuite bien sûr

- `int n;`

- `n = ... lecture clavier, etc. ...`

- `int p = 2 * n; // ok`

- cas de variables non initialisées
 - impossible utiliser variables n'ayant pas reçu de valeur (rejeté compilation)
 - ex :

```
int n, p;  
if (...)  
    n = 30;  
    p = 2 *n; // erreur : pas initialisée dans tous les cas
```
 - la notion de valeur d'initialisation par défaut n'a donc pas de sens

- variable final : ne peut recevoir qu'une seule fois une valeur, n'importe quand au cours du programme (y compris initialisation) : détecté par compilateur
- expression constante : calculable par le compilateur

notion de conversion implicite

- opérateurs opèrent sur arguments de même type
- ex : `double x = 3 + 4.5;`
 - 3 converti en double
 - puis + (double) appliqué
- il faut connaître règles conversion (LO/CCI : non)
- en général marche comme on pense
- mais... en cas bug dans un calcul : bien afficher tous les résultats intermédiaires

les types primitifs : récapitulatif

- La notion de type
- Les types entiers : byte, short, int, long
- Les types flottants : float, double
- Le type caractère
- Le type booléen
- Initialisation et constantes
 - Variables non initialisées
- (Delannoy chapitre 3)