INF1120 – Programmation 1

Variables et opérateurs

Jacques Berger

© 2018 Jacques Berger

Objectifs

Introduire les principaux types de données

Introduire les principaux opérateurs

© 2018 Jacques Berger

Prérequis

Introduction

© 2018 Jacques Berger

Types

Les variables doivent avoir un type

Le type correspond à la nature de l'information que l'on stocke dans la variable

© 2018 Jacques Berger

Types

Il existe plusieurs types pour différents besoins

Utiliser le bon type simplifie la programmation

© 2018 Jacques Berger

Nombre entier

Type int

Pour conserver une valeur entière, positive ou négative

Peut contenir les nombres de -2^{31} à $2^{31}-1$

Taille de 4 octets en mémoire

© 2018 Jacques Berger

Nombre réel

Type double

Pour conserver une valeur réelle, positive ou négative

Nombre avec partie fractionnaire

Taille de 8 octets

© 2018 Jacques Berger

Nombres

Opérateurs sur nombres entiers et réels

+ : addition
- : soustraction
* : multiplication
/ : division
% : modulo (donne le reste d'une division)

© 2018 Jacques Berger

Nombres

La division a un comportement différent dépendamment du type des opérandes

Si les deux opérandes sont des entiers, elle fait une division entière, sans partie fractionnaire

Si un des deux opérandes est un réel, le résultat sera aussi un nombre réel, avec partie fractionnaire

© 2018 Jacques Berger

Nombres

Il est possible de combiner une opération mathématique avec une opération d'affectation

Exemple : +=

```
variable += 5;
```

équivaut à :

```
variable = variable + 5;
```

© 2018 Jacques Berger

Nombres

Les opérateurs combinant une opération et une affectation :

+= -= *= /= %=

© 2018 Jacques Berger

Nombres

L'opérateur ++ sert à incrémenter la variable de 1
L'opérateur -- sert à décrémenter la variable de 1

Exemple :

```
i = i + 1;
```

devient :

```
i++;
```

© 2018 Jacques Berger

Nombres

La position de l'opérateur ++ ou -- a un impact sur la priorité de l'opération

Avec i++, l'incrémentation est faite après le reste de l'instruction

Avec ++i, l'incrémentation est faite avant le reste de l'instruction

© 2018 Jacques Berger

Cast

Il est possible de transformer un entier en réel et vice versa

C'est ce qu'on appelle un transtypage (ou cast en anglais)

Dans certains cas, un cast implicite est fait (automatiquement par le compilateur), sinon il faut le spécifier nous même

© 2018 Jacques Berger

Cast implicite

Lorsque le type source est plus petit que le type de destination

```
int variable = 4;
double reel = variable; // cast implicite
```

© 2018 Jacques Berger

Cast explicite

Le compilateur refuse d'affecter la variable car la source est plus grande que la destination

```
double reel = 33.3;
int valeur = (int) reel;  // cast explicite
```

© 2018 Jacques Berger

boolean

Une variable booléenne ne peut avoir que 2 valeurs possibles : true et false

Utile pour conserver un état ou une condition

© 2018 Jacques Berger

boolean

Opérateurs

! : Inversion
&& : ET logique
|| : OU logique

© 2018 Jacques Berger

Priorité des opérateurs

Pour connaître la priorité des opérateurs en Java

https://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html

© 2018 Jacques Berger

Constantes

Une constante est une variable dont la valeur ne peut pas changer

Une constante peut être déclarée à l‘extérieur d‘une fonction

```
static final int MAXIMUM_AGE = 115;
```

© 2018 Jacques Berger

Casse

Un nom de variable doit être en camelCase

Un nom de constante doit être en UPPER_SNAKE_CASE

© 2018 Jacques Berger

Structure conditionnelle

Une variable booléenne va souvent servir à vérifier une condition

À l'aide d'une structure conditionnelle, on peut changer le chemin d'exécution du programme afin d'exécuter certaines instructions uniquement si une condition est respectée ou non

© 2018 Jacques Berger

if

L'instruction if permet d'exécuter son contenu uniquement si la condition donnée est vraie

```
boolean condition = true;
if (condition) {
   // placer ici les instructions…
}
```

© 2018 Jacques Berger

if

La partie entre les accolades se nomme un bloc d'exécution

© 2018 Jacques Berger

if

On utilise l'opérateur ! lorsqu'on veut exécuter des instructions si une condition est fausse

```
boolean condition = false;
if (!condition) {
   // placer ici les instructions…
}
```

© 2018 Jacques Berger

if

On place une clause else lorsqu'une condition nous oblige à choisir entre 2 chemins d'exécution

```
boolean condition = true;
if (condition) {
    System.out.println("La condition est vraie.");
} else {
    System.out.println("La condition est fausse.");
}
```

© 2018 Jacques Berger

Comparaison

L'opérateur == retourne un booléen indiquant si deux variables ont la même valeur

L'opérateur != retourne un booléen indiquant si deux variables ont une valeur différente

Ces opérateurs sont habituellement utilisés avec une structure conditionnelle

© 2018 Jacques Berger

Comparaison

```
double reel = 3.0;
int entier = 3;

if (reel == entier) {
    System.out.println("Les valeurs sont égales.");
}
```

© 2018 Jacques Berger

Comparaison

Opérateurs de comparaison

== : Égalité
!= : Inégalité
< : Plus petit que
<= : Plus petit ou égal
> : Plus grand que
>= : Plus grand ou égal

© 2018 Jacques Berger

Comparaison

Avec les opérateurs && et ||, on peut construire des conditions plus complexes

```
if (reel < 1 && reel > 0) {
    System.out.println("Le nombre est entre 0 et 1.");
}
```

© 2018 Jacques Berger

char

Le type char sert à stocker un caractère

Le caractère doit être placé entre apostrophes

```
char lettre = 'r';
char espace = ' ';
```

© 2018 Jacques Berger

char

Certaines valeurs particulières s'écrivent avec 2 caractères, mais ça ne compte que pour un caractère

'\n' : Saut de ligne
'\t' : Tabulation

© 2018 Jacques Berger

char

On peut également affecter un nombre entier à un char

Chaque caractère possède une valeur numérique (Unicode)

Les valeurs de 0 à 127 correspondent à la table ASCII

© 2018 Jacques Berger

Autres types

Entiers :

- byte : 1 octet
- short : 2 octets
- long : 8 octets

Réel :

- float : 4 octets

© 2018 Jacques Berger

Plus loin...

Table ASCII
    http://www.asciitable.com/

Unicode
    http://unicode-table.com/fr/

Le livre de Java premier langage, 11ème édition
Anne Tasso
Eyrolles
Chapitres 1 et 3

© 2018 Jacques Berger