

# IFT1015 Programmation 1

## Variables

Sébastien Roy et F. Duranleau

Département d'Informatique et de recherche opérationnelle  
Université de Montréal



6 septembre 2007  
Université de Montréal



## Au programme

- Qu'est-ce qu'une **variable**?
- Valeurs
- Types
- Nommer des variables
- Déclaration

## Références

Tasso: Chapitre 1



## Exemple

```
public class Cercle
{
    public static void main(String [] args)
    {
        // rayon: le rayon lu; perim: le perimetre du cercle
        double rayon, perim;

        // Lecture du rayon
        rayon = Double.parseDouble(args[0]);

        // Calcul du perimetre
        perim = 2 * Math.PI * rayon;

        // Afficher le resultat
        System.out.println("Le cercle de rayon " + rayon
            + " a le périmètre " + perim);
    }
}
```



## Concepts

- Après l'analyse du problème à résoudre, la première étape est d'identifier les éléments manipulés
- Exemple  
Pour le calcul du périmètre du cercle, les éléments manipulés sont le [rayon](#) et le [périmètre](#)
- Dans un programme, il faut prévoir de l'espace en mémoire pour stocker ces éléments



## Définitions

### Variable

*"Symbole ou terme auquel on peut attribuer plusieurs valeurs distinctes, à l'intérieur d'un domaine défini."*

— Le Robert

En programmation: une variable est un emplacement mémoire où une valeur est stockée.

⇒ "Nom du tiroir"



## Définitions

### Valeur

*"Une information fondamentale simple ou composée qui peut être manipulée par un programme."*

Exemples

- 13 , 'a' , -5 , 12.456 , "abc"
- une adresse mémoire



## Définitions

### Type

*"Un ensemble de valeurs apparentées avec les opérations qui peuvent être effectuées avec elles."*

Exemples

- nombre entier, nombre réel, caractère
- adresse mémoire
- strings, comptes bancaires, objets géométriques



## Déclaration d'une variable

### Syntaxe

```
<type> <nom>;  
ou  
<type> <nom1>, <nom2>, ..., <nomN>;
```

<nom>

Étiquette servant de référence à un emplacement en mémoire

<type>

Sert à déterminer la dimension de l'espace mémoire et la façon dont est traduit le code binaire de la valeur qui y est stockée.

- Il faut déclarer une variable avant de l'utiliser



## Exemple

Dans l'exemple de la circonférence du cercle...

### Le rayon

<b>Nom</b>	rayon
<b>Type</b>	double
	(nombre réel)

### Le périmètre

<b>Nom</b>	perim
<b>Type</b>	double
	(nombre réel)



## Nom d'une variable

Choix d'un nom significatif

- Important pour la lisibilité du programme
- Permet d'identifier le rôle et le contenu de la variable

Restrictions

- sensible à la casse (aux majuscules/minuscules)
- commence par une lettre, \_ ou \$
- chaque caractère suivant est soit une lettre, un chiffre, \_ ou \$
- ne doit pas être un mot-clé du langage



## Mots-clé de Java

abstract	default	goto	null	synchronized
boolean	do	if	package	this
break	double	implements	private	throw
byte	else	import	protected	throws
case	extends	instanceof	public	transient
catch	false	int	return	true
char	final	interface	short	try
class	finally	long	static	void
const	float	native	super	volatile
continue	for	new	switch	while



## Nom d'une variable

<u>noms valides</u>	<u>noms invalides</u>
compte	Compte#un
r	2001espace
super\$Inner	une-variable
_variable	double
UNE.CONSTANTE	UNE CONSTANTE
année	l'année



## Directives pour choisir un bon nom

Choisir un nom qui décrit la variable

- rayon plutôt que x1

Éviter les noms trop longs

- rayon plutôt que leRayonDuCercle

Éviter les abréviations, ou être consistant dans leur usage

- employeeRecord et masterRecord ou employeeRec et masterRec mais pas employeeRec et masterRecord

La règle d'or

Mon programme est-il plus lisible?



## Directives pour choisir un bon nom

Être le plus spécifique possible

- perimetre plutôt que resultat

Être descriptif pour distinguer les rôles des variables

- ancienRayon et nouveauRayon plutôt que rayon1 et rayon2

Ne pas incorporer le type de la variable dans son nom

- rayon plutôt que rayonNombreRéal

Convention informelle: nomDeParametre



## Type d'une variable

Définit la quantité de mémoire que prend une variable

- limite sur le nombre de valeurs différentes

La plus petite unité adressable est l'octet (*byte*): un groupe de 8 bits

→ l'espace mémoire est toujours un multiple de 8 bits

L'espace octroyé à chaque type dépend du langage et/ou de l'architecture de la machine

- En Java cet espace est indépendant de l'architecture



## Types simples et composés

### Types simples

aussi appelés types de base (*built-in*)

- ne contiennent qu'une seule valeur
- prédéfinis dans le langage
- Ex: nombres, caractères

### Types composés

aussi appelés types structurés

- contiennent plusieurs valeurs
- définis par le programmeur
- représente des valeurs plus complexes
- Ex: date, chaîne de caractère, compte bancaire



## Types simples de Java

type	Mot-clé Java	Taille (bits)
octet	byte	8
entier "court"	short	16
entier	int	32
entier "long"	long	64
nombre réel	float	32
nombre réel "long"	double	64
caractère	char	16
booléen	boolean	8

◀ ▶ ⏪ ⏩ 🔍 ↺

## Nombres entiers (int)

### Représentation

- $n$  bits:  $[0, 2^n - 1]$
- un bit pour le signe:  $[-2^{n-1}, 2^{n-1} - 1]$
- 32 bits:  $[-2147483648, 2147483647]$

### Valeurs littérales

- suite de chiffres sans espace, avec ou sans - avant le nombre
- exemples valides: 25, 0, 10191, 1234567, -45
- exemples invalides: 10, 191, 1, 234, 567, 3.14
- attention (octal): 0123 est 123 en base 8, donc 83...

◀ ▶ ⏪ ⏩ 🔍 ↺

## Nombres réels (double)

### Représentation

- par le standard IEEE-754: stocker la base et l'exposant
- 8 octets (double):  $\sim [-10^{300}, 10^{300}]$ , précision de 15 chiffres

### Valeurs littérales

- nombres en point flottant, notation scientifique
- exemples: 3.14, 250.0, -0.00333, 3.56E-13

◀ ▶ ⏪ ⏩ 🔍 ↺

## Caractères (char)

### Représentation

- 16 bits, Unicode (une extension d'ASCII)

### Valeurs littérales

- un seul caractère entre guillemets simples: 'a'
- caractères spéciaux (*escape sequences*): '\n' (fin de ligne), '\t' (tabulation), '\\' (barre oblique)

◀ ▶ ⏪ ⏩ 🔍 ↺

## Chaînes de caractères (String, en bref)

String est un type composé, mais très utilisé. Le langage permet donc de l'utiliser un peu comme un type simple.

### Valeurs littérales

- séquence de caractères délimitée par les guillemets, "abc"
- "Hello, World!"
- "a"

Plus de détails sur ceci et les types composés plus tard ...



## Booléens (boolean)

Représente une valeur de vérité (vrai ou fausse)

### Valeurs littérales

Deux valeurs possibles: true et false

On reviendra plus tard sur l'utilisation des booléens ...



## Déclaration

### Syntaxe

```
<type> <nom>;
```

ou

```
<type> <nom1>, <nom2>, ..., <nomN>;
```

### Exemples

```
int age;  
double rayon,perimetre;  
String message;  
boolean carotteCuite;
```

