IFT 1015 - Variables

Professeur:

Stefan Monnier

B. Kégl, S. Roy, F. Duranleau, S. Monnier

Département d'informatique et de recherche opérationnelle

Université de Montréal

hiver 2006

Au programme

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

Références

- [Tasso: Chapitre 1]
- [Niño: 2.1, 2.2, 5.1.2, Chapitre 3]

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);
  }
}
```

Conception

- 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*

- 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 à virgule flottante (“réel”), caractère
  - adresse mémoire
  - strings, comptes bancaires, objets géométriques

Déclaration d'une variable

- Syntaxe:
  - `<type> <nom>;` (ou `<type> <nom1>,...,<nomN>;`)
  - Il faut déclarer une variable avant de l'utiliser
- Nom
  - Étiquette servant à repérer l'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

Exemple

- Problème de la circonférence du cercle

Le rayon

Nom	`rayon`
Type	`double`
	(nombre “réel”)

Le périmètre

Nom	`perim`
Type	`double`
	(nombre “réel”)

Nom d'une variable

- Choisir un nom significatif:
  - facilement identifier le rôle ou le contenu d'une 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

noms valides	noms non valides
compte	Compte#un
r	2001espace
super$Inner	une-variable
_variable	double
UNE_CONSTANTE	UNE CONSTANTE

Directives pour choisir un bon nom

- Choisir un nom qui décrit la variable
  - `rayon` plutôt que `x1`
- Éviter des noms trop longs
  - `rayon` plutôt que `leRayonDuCercle`
- Éviter les abréviations, ou être consistant dans leurs usages
  - `employeeRecord` et `masterRecord` ou `employeeRec` et `masterRec` mais pas `employeeRec` et `masterRecord`

Directives pour choisir un bon nom

- Être le plus spécifique possible
  - `perimetre` plutôt que `resultat`
- Être descriptif pour distinguer des variables reliées
  - `ancienRayon` et `nouveauRayon` plutôt que `rayon1` et `rayon2`
- Ne pas incorporer le type de la variable dans son nom
  - `rayon` plutôt que `rayonNombreRéel`
- Convention “formelle”: `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*): une suite de 8 bits
  - l'espace mémoire sera 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 (*built-in types*)
  - ne contiennent qu'une valeur
  - prédéfinis dans le langage
  - nombres, caractères
- Types composés/structurés
  - contiennent plusieurs valeurs
  - définis par le programmeur
  - chaînes, comptes bancaires

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 virgule flottante	`float`	32
nombre virgule flottante “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: $[-2\,147\,483\,648, 2\,147\,483\,647]$
- 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: `0123` $= 123_8 = 83$

Nombres “réels” à virgule flottante (`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'` (tab), `'\\'` (backslash)

Chaînes de caractères (`String`, en bref)

- `String` est une classe (type composé)
  - n'est pas un type simple
- Valeurs littérales
  - séquence de caractères délimitée par `"  "`
  - `"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: `true` et `false`

  Plus sur ceci quand on parlera des conditions . . .

Déclaration

- Syntaxe:
  - `<type> <nom>;`
  - `<type> <nom1>, ..., <nomN>;`
  - Il faut déclarer une variable avant de l'utiliser
- Exemples:
  - `double rayon, perimetre;`
  - `String greeting;`
  - `boolean soupeCuite;`