Chapitre 2 : Les variables

I. Le rôle des variables

Les variables servent comme un lieu de stockage pour un programme informatique.

Pour employer une image, une variable est une boîte, que le programme (l'ordinateur) va repérer par une **étiquette** (nom). Pour avoir accès au **contenu de la boîte**, il suffit de la désigner par son étiquette.

Une variable désigne en fait un **emplacement mémoire** dont le contenu peut changer au cours d'un programme (d'où le nom variable).

Les variables doivent être **déclarées avant d'être utilisées**, elle doivent être caractérisées par :

- Un nom (Identificateur)
- Un type (entier, réel, caractère, chaîne de caractères, …)

Exemple :

Pour réaliser un algorithme pour faire l'addition entre deux nombre. Nous avons besoin des emplacements pour stocker les valeurs de ces deux pour que l'ordinateur réaliser l'opération souhaitée.

II. Déclaration des variables

La déclaration des variables permet de réserver une espace mémoire (boite) qui va être utilisé par son nom dans le corps de l'algorithme.

La forme générale pour déclarer une variable est la suivante :

Nom_de_la_variable : Type_de_la_variable

1.1. Nom des variables

Le choix des noms de variables est soumis à quelques règles qui varient selon le langage, mais en général:

- Un nom doit commencer par une lettre alphabétique

  Exemple valide : A1

  Exemple invalide : 1A

- Doit être constitué uniquement de lettres, de chiffres et du soulignement _ (Eviter les caractères de ponctuation et les espaces)

  Exemple valides : SMIP2007, SMP_2007

  Exemple invalides : SMP 2005 , SMI-2007, SMP;2007

- Doit être différent des mots réservés du langage (par exemple en Java: **int, float, else, switch, case, default, for, main, return**, …).

- La longueur du nom doit être inférieure à la taille maximale spécifiée par le langage utilisé.

Conseil : pour la lisibilité du code choisir des noms significatifs qui décrivent les données manipulées.

Exemples: TotalVentes2004 , Prix_TTC , Prix_HT.

1.2. Les types des variables

Le type d'une variable détermine l'ensemble des valeurs qu'elle peut prendre, les types offerts par la plus part des langages sont:

- Type numérique (entier ou réel)

  Byte (codé sur 1 octet): de 0 à 255

  Entier court(codé sur 2 octets) : -32 768 à 32 767

  Entier long (codé sur 4 ou 8 octets)

  Réel simple précision (codé sur 4 octets)

  Réel double précision (codé sur 8 octets)

  - Remarque: pour le type numérique on va se limiter aux entiers et réels sans considérer les sous types

- Type logique ou booléen:

  Deux valeurs VRAI ou FAUX

- Type caractère:

  Lettres majuscules, minuscules, chiffres, symboles, …

  Exemples: 'A', 'a', '1', '?', …

- Type chaîne de caractère:

Toute suite de caractères, **Exemples: " Nom, Prénom", "code postale: 1000", ...**

Exemple :

```
Algorithme Exemple_Declaration
Variables
        Var1 : Entier
        Var2, Var3 : booléen
        Var4, var5 : caractére
        Var6 : chaine
  Début


  Fin
```

III. Affectation

L'affectation consiste à attribuer une valeur à une variable (ça consiste en fait à remplir où à modifier le contenu d'une zone mémoire)

En algorithme, l'affectation se note avec le signe ←

Var← e : attribue la valeur de e à la variable Var

- e peut être une valeur, une autre variable ou une expression

- Var et e doivent être de même type ou de types compatibles
- L'affectation ne modifie que ce qui est à gauche de la flèche.
- l'affectation n'est pas commutative : A=B est différente de B=A.

- **Exemple valides :**

```
Algorithme Exemple_Valide_affectation
Variables
   i, j,k : entier
   x, y : réel
   OK: booléen
   ch1, ch2 : chaîne de caractères
Début
   i ←1
   j ←i
   k ←i+j
   x ←10.3
   OK ←FAUX
   ch1 ←"TDI"
   ch2 ←ch1
   x ←4
   x ←j
Fin
```

- **Exemple non valides :**

```
Algorithme Exemple_NONValide_affectation
Variables
    i, j,k : entier
    x, y : réel
    OK: booléen
    ch1, ch2 : chaîne de caractères
Début
    i ←10.3
    OK ←"TDI"
    j ←x
Fin
```

Certains langages de programmation donnent des valeurs par défaut aux variables déclarées. Pour éviter tout problème il est préférable d'initialiser les variables déclarées.

IV. Exercices d'application

1.1. Exercice 1

Donnez les valeurs des variables A, B et C après exécution des instructions suivantes ?

Algorithme Exo1

Variables

A, B, C: **Entier**

Début

A ← 3

B ← 7

A ← B

C ← A

Fin

Solution : A =7 / B=7 / C=7

1.2. Exercice 2

Donnez les valeurs des variables A et B après exécution des instructions suivantes ?

Algorithme Exo2

Variables A, B : **Entier**

Début

A ← 1

B ← 2

A ← B

B ← A

Fin

Solution : A=2 / B=2

Les deux dernières instructions permettent-elles d'échanger les valeurs de A et B ? **Réponse : Non**

1.3. Exercice 3

Ecrire un algorithme qui permet de :

- Déclarer trois variables A, B et C de type entier
- Initialiser les valeurs de A et B par des valeurs de votre choix.
- Echanger les valeurs A et B en utilisant la variable C.

Solution :

Algorithme Echange_valeurs_A_B

Variables A,B,C : entier

Début

A← 5

B←10

C←B

B←A

A←C

Fin

V. Expression et opérateurs

Une expression peut être une valeur, une variable ou une opération constituée de variables reliées par des opérateurs

Exemples: 1, b, a*2, a+ 3*b-c, ...

L'évaluation de l'expression fournit une valeur unique qui est le résultat de l'opération

Les **opérateurs** dépendent du type de l'opération, ils peuvent être :

- **Des opérateurs arithmétiques: +, -, *, /, % (modulo), ^ (puissance)**

  Exemple :

  Algorithme OP_ARTH

  Variables

  A , B : Entier

  Début

  A ← 4 + 3

  B ← A * 2

  Fin

- **Des opérateurs logiques: NON, OU, ET**
- **Des opérateurs relationnels: =, , <, >, <=, >=**
- **Des opérateurs sur les chaînes: & (concaténation)**

  Exemple :

  Algorithme OP_CHAINE

  Variables

  A , B, C : chaine

  Début

  A ← **"TDI"**

  B ← **"TRI"**

  C ← A & B

  Fin

Une expression est évaluée de gauche à droite mais en tenant compte de **priorités**

Pour les opérateurs arithmétiques donnés ci-dessus, l'ordre de priorité est le suivant (du plus prioritaire au moins prioritaire):

- ^ : (élévation à la puissance)
- * , / (multiplication, division)
- % (modulo)
- + , - (addition, soustraction)

Exemple: **2 + 3 * 7 vaut 23**

En cas de besoin (ou de doute), on utilise les parenthèses pour indiquer les opérations à effectuer en priorité

exemple: (2 + 3) * 7 vaut 35

VI. Les exercices d'application

1.1. Exercice 1

Que produit l'algorithme suivant :

Algorithme EXO_1

Variables

A , B , C : chaine

Début

A ← **"432"**

B ← **"15"**

C ← **A & B**

Fin

Solution : C = "43215"

1.2. Exercice 2

Que produit l'algorithme suivant :

Algorithme EXO_2

Variables

A , B , C : chaine

Début

A ← **"432"**

B ← **"15"**

C ← **A + B**

Fin

Solution : Erreur on peut jamais additionner des chaines de caractères

1.3. Exercice 3

Que produit l'algorithme suivant :

Algorithme EXO_3

Variables

A , B : entier

Début

A ← **4**

B ← **1**

A ← A + B

B ← **A – B**

A ← A - B

Fin

Solution : A = 1 / B = 4 (Une autre façon pour échanger les valeurs car on peut considérer que l'expression est comme une variable)