

Cours « Algorithmique »

2^{ème} année Licence

Chapitre 3 (Suite):

Notions d' « Algorithmique »

Structure d'un algorithme

un algorithme se présente sous la structure suivante :

Algorithme nom_algorithme //entête de l'algorithme

Var <expressions> //partie déclaration

Début

instruction 1

instruction 2

⋮

instruction n

//Corps de l'algorithme

Fin

Les parties d'un algorithme

Entête de l'algorithme

permet simplement d'identifier un algorithme.

Déclaration

permet de déclarer les variables à utiliser

Corps de l'algorithme

c'est la partie qui se trouve entre **début** et **fin**,
dans cette partie sont placées les instructions à exécuter.

Commentaire

permet de documenter l'algorithme et faciliter sa compréhension,

// pour écrire un commentaire sur une seule ligne
(* ... *) pour un commentaire multi-lignes.

Partie déclaration :

Déclaration des constantes : une constante désigne un espace mémoire dont la valeur ne peut changer dans le temps.

Cette valeur doit être fixée dès la déclaration,

Exemple : **const** PI $\leftarrow$ 3.14 ;

Déclaration des variables : une variable désigne un emplacement mémoire qui permet de stocker une valeur.

Une variable est définie par :

- Un nom unique qui la désigne.
- Un type de définition.
- Une valeur attribuée et modifiée au cours du déroulement de l'algorithme.

Nom d'une variable (identifiant) : le nom d'une variable est un nom qui permet de l'identifier de manière unique dans l'algorithme. **Exemple : var x, y : entier ;**

La déclaration d'une variable réserve un espace mémoire dans la mémoire centrale de taille égale à celle spécifiée par son type.

Caractéristiques du nom de variable :

- Il est préférable que le nom soit significatif de l'information qu'elle désigne.
- Le nom doit obligatoirement commencer par une lettre.
- Le nom peut contenir des lettres, des chiffres et le caractère « _ ».
- Le nom ne doit pas contenir des espaces.

Type d'une variable :

Le type d'une variable indique l'ensemble des valeurs que la variable peut prendre ainsi que la taille de l'espace mémoire à retenir.

Il existe plusieurs types :

- entier,
- réel,
- caractère,
- chaîne,
- booléen

Le type numérique :

Les variables de type numérique utilisées dans l'algorithmique ont comme domaines usuels : **réel (réel)** ou **entier (entier)**.

Les opérateurs arithmétiques utilisables dans ces domaines sont : l'addition (+), la soustraction (-), le produit (*), la division (/) et l'exponentiel (^).

On peut aussi appliquer sur les éléments de type entier ou réel, les opérateurs de comparaison : $>$, $<$, $\neq$, $=$, $\geq$, $\leq$.

Fonctions arithmétiques standards pour les types numériques:

Fonction	Description	Exemple
Tronc(x)	Extraire la partie entière de x.	Tronc(-5.125) vaut -5 Tronc(3.14) vaut 3
Arrondi(x)	Arrondit une valeur réelle à l'entier le plus proche.	Arrondi(7.499) vaut 7 Arrondi(7.50) vaut 8 Arrondi(7.99) vaut 8
Abs(x)	renvoie la valeur absolue de x.	Abs(-3) vaut 3
Carrée(x)	donne le carré de x.	Carré(3) vaut 9
RacineCarrée(x)	donne la racine carrée de x	RacineCarré(9) vaut 3.0

Type caractère et chaîne de caractères :

Il s'agit d'un type constitué des caractères **alphabétiques**, **numériques** et de **punctuation** et les autres symboles &, #, ...etc.

Les opérations élémentaires pour les éléments de type **caractère (char)** ou **chaîne de caractères (chaîne)** sont les opérations de comparaison : >, <, ≠, =, ≥, ≤ et le + pour la concaténation.

Exemple :

x : char ; // x ← 'A'

Nom : chaîne ; // nom ← "Guelma" ;

Type booléen :

Le type des booléens (**booléen**) est l'ensemble formé des deux seules valeurs {vrai, faux}.

Les opérations admissibles sur les éléments de ce type sont réalisées à l'aide de tous les opérateurs logiques :

- **ET** (et logique),
- **OU** (ou logique),
- **OUEX** (ou exclusif)
- **NON** (négation logique),
- ainsi que les deux opérateurs de comparaison **=**, **≠**.

Fonctions logiques standards pour les types booléens :

X	Y	x ET y	x OU y	x OU EX y	NON x
Vrai	Vrai	Vrai	Vrai	Faux	Faux
Vrai	Faux	Faux	Vrai	Vrai	Faux
Faux	Vrai	Faux	Vrai	Vrai	Vrai
Faux	Faux	Faux	Faux	Faux	Vrai

Les types énumérés :

un type énuméré est un type dont les variables associées n'auront qu'un nombre limité de valeurs à prendre. Définir un type énuméré consiste à énumérer la liste de ses valeurs possibles.

Exemples :

Type Jour = (Dimanche, Lundi, Mardi, Mercredi, Jeudi, Vendredi, Samedi) ;

Type Couleur = (Rouge, Bleu, Vert, Jaune, Blanc, Noir) ;

Var

j : Jour ;

c : Couleur ;

Les opérations applicables aux valeurs de type énumérés sont : $>$, $<$, $\neq$, $=$, $\geq$, $\leq$.

De plus, on peut utiliser les fonctions suivantes :

Fonction	Description	Exemple
Succ(x)	Fournit le successeur de la valeur donnée	Succ(Lundi) = Maridi
Pred(x)	Fournit le prédécesseur de la valeur donnée	Pred(Lundi) = Dimanche
Ord(x)	Fournit l'ordre de la valeur donnée	Ord(Lundi) = 2

Le type Tableau :

Un tableau est une structure de données qui désigne par une seule variable (nom du tableau) un ensemble de valeurs de même type accessibles via leurs positions dans le tableau.

La dimension et le type du tableau doivent être précisés lors de sa définition.

Pour accéder aux éléments du tableau, un indice est utilisé, ce dernier indique le rang de l'élément.

Tableau à une dimension :

Const n $\leftarrow$ 10 ;

Var tab : tableau[1..n] d'entier ;

tab	22	-7	56	12	0	23	-4	1	57	35
Indice	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]

Nous constatons que:

tab[6] = 23,

tab[3] = 56,

tab[9] = 57,

Tableau à deux dimensions :

```
Const  n_lignes ← 3 ;  
       n_colonnes ← 3 ;
```

```
Var  
tab : tableau[1 .. n_lignes][1 .. n_colonnes] d'entier ;
```

[1][1]	[1][2]	[1][3]
[2][1]	[2][2]	[2][3]
[3][1]	[3][2]	[3][3]

Exemple 1

Algorithme Somme

Var

a:entier, b : entier, résultat: entier ;

Debut

a ← 5 ;

b ← 7 ;

résultat ← a+b ;

Fin

Exemple 2

Algorithme

Var

a:entier, résultat: entier ;

Debut

$a \leftarrow 5$;

 résultat $\leftarrow a * a$;

Fin

Exemple 2: écrire un algorithme qui calcule le carré d'un entier, et arrondit le résultat.

Algorithme Carré_arrondissement;

Var

a:reel, résultat_carr: réel ;

Resultat_arr:entier;

Debut

 Lire(a);

 Résultat_carr $\leftarrow$ carré(a) ;

 Resultat_arr $\leftarrow$ arrondi(résultat_carr);

 Ecrire (resultat_carr);

 Ecrire (resultat_arr);

Fin

Exemple 2

Algorithme

Var

a, b, c: entier, résultat: entier ;

Debut

a ← 12.5 ;

b ← 15 ;

c ← 10.5 ;

résultat ← (a+b+c)/3 ;

Fin

Exemple 2

Algorithme Moyenne

Var

a, b, c: entier, résultat: entier ;

Debut

a ← 12.5 ;

b ← 15 ;

c ← 10.5 ;

résultat ← (a+b+c)/3 ;

Fin

Notion de « fichier »

- Les données et les programmes sont stockés dans des fichiers
- Un fichier est identifié par un **nom** et une **extension** (fichier.doc, pgcd.c, texte.tex, etc.)
- Un fichier est caractérisé par des **attributs**:
taille, date de création, date de modification, etc....
- Les **opérations** de base sur un fichier sont :
ouvrir, fermer un fichier, lire, écrire
- L'utilisateur peut créer, détruire, organiser, lire, écrire, modifier et copier des fichiers ou parties de fichier (enregistrements)