

CHAPITRE 6 : LES STRUCTURES DES DONNÉES

LEÇON 1 : LES CONSTANTES ET LES VARIABLES

I. INTRODUCTION :

Activité : Ecrire un programme qui permet de calculer le périmètre d'un cercle.

Pré-analyse

Résultat : Ecrire (P)

$P \leftarrow PI * D$

D = Donnée

Algorithme :

0) **Début** périmètre

1) Lire (D)

2) $P \leftarrow PI * D$

3) Ecrire (P)

4) **Fin** périmètre

Remarque :

On remarque qu'un programme utilise des objets qui peuvent être des **constantes** ou des **variables**.

II. LES CONSTANTES :

1. Définition

- On appelle une constante un objet ayant une valeur fixe tout le long d'un algorithme.
- Une constante est caractérisée par *son nom* et *sa valeur*.

2. Déclaration au niveau analyse :

Tableau de déclaration des objets

Objets	Type / Nature	Rôle
Ident_constante	Constante = valeur_constante	Rôle

Exemple :

Objets	Nature / Type	Rôle
PI	Constante = 3.14	Constante prend la valeur 3.14

3. Déclaration au niveau PASCAL :

CONST <ident_constante> = <valeur de constante> ;

Exemple : **CONST** PI = 3,14 ;

III. LES VARIABLES :

1. Définition

- On appelle un variable tout objet pouvant prendre différentes valeurs tout le long d'un algorithme.
- Une variable est caractérisée par *son nom*, *son type* et *son contenu*

2. Déclaration au niveau analyse :

Tableau de déclaration des objets :

Objets	Type / Nature	Rôle
Ident_variable	Type de la variable	Rôle

Exemple :

Objets	Nature / Type	Rôle
P	Réel	Périmètre du cercle
D	Réel	Diamètre du cercle

3. Déclaration au niveau PASCAL :

VAR <ident_variable> : <type_variable> ;

Exemple : **VAR** P, D : Real ;

4. Remarques :

- Une variable est un moyen pour stocker les données
- L'opération permettant de charger le contenu d'une variable est l'affectation

IV. APPLICATION:

Soit la séquence d'affectation suivante :

1. $m \leftarrow 10$
2. $n \leftarrow 20$
3. $p \leftarrow m$
4. $m \leftarrow n$
5. $n \leftarrow p$

Donnez le résultat d'exécution de cette séquence :

N° de l'instruction	m	n	p
1	10	-	-
2	10	20	-
3	10	20	10
4	20	20	10
5	20	10	10

Quelles sont les valeurs finales de m et n

$M = 20$ et $N = 10$

Quel est le rôle de cette séquence d'affectation

Permutation de contenu de deux variables m et n

Leçon 2 : Les types des données et les expressions

I. LES TYPES DE DONNEES :

1. Le type entier :

a) Domaine :

Une variable de type entier contient un nombre appartenant à un sous ensemble Z, souvent c'est l'intervalle [-32768 ... 32767]

b) Les opérateurs sur les entiers :

Type	Opérateurs
Opérateurs arithmétiques	+, -, *, DIV, MOD
Opérateurs relationnels	<, >, <=, >=, =, <>

NB :

MOD : Donne le reste de la division entière

DIV : Donne le quotient de la division entière

c) Déclaration :

↪ Au niveau analyse : (Tableau de déclaration des objets)

Objets	Type / Nature	Rôle
Ident_variable	Entier	Rôle

↪ Au niveau PASCAL : **VAR <nom de variable> : INTEGER ;**

Exemples : **VAR x : INTEGER ;**

d) Application:

Evaluer les expressions suivantes

$((30 \text{ MOD } 7) + (40 \text{ DIV } 22)) \rightarrow 2 + 1 = 3$

$(100 \text{ MOD } 90) \text{ DIV } 5 \rightarrow 10 \text{ DIV } 5 \rightarrow 2$

2. Le type réel :

a) Domaine :

Une variable de type réel contient un nombre appartenant à un sous ensemble de IR. Exp. : 0; 1.25 ; 2E+5; 15E-2

b) Déclaration :

↪ Au niveau analyse : (Tableau de déclaration des objets)

Objets	Type / Nature	Rôle
Ident_variable	Réel	Rôle

↪ Au niveau PASCAL : **VAR <nom de variable> : REAL ;**

Exemples : **VAR x : REAL ;**

c) Les fonctions arithmétiques standard :

NOM ALGO	NOM PASCAL	ROLE	TYPE DONNEES	TYPE RESULTAT	EXEMPLES
TRUNC (X)	TRUNC (X)	Permet d'extraire la partie entière de X	Réel	Entier	TRUNC (1.1) = 1 TRUNC (-2.9) = -2
ARRONDI (X)	ROUND (X)	Retourne l'entier le plus proche de X	Réel	Entier	ROUND (2.49) = 2 ROUND (2.5) = 3 ROUND (2.1) = 2
ABS (X)	ABS (X)	Retourne la valeur absolue de X	Entier ou Réel	Même type que X	ABS (-5) = 5 ABS (4.25) = 4.25
CARRE (X)	SQR (X)	Retourne le carré de X	Entier ou Réel	Même type que X	SQR (2) = 4 SQR (2.5) = 6.25
RACINECARRE (X)	SQRT (X)	Retourne la racine carrée de X si (X > 0)	Entier ou Réel	Même type que X	SQRT (4) = 2 SQRT (6.25) = 2.5
FRAC (X)	FRAC (X)	Retourne la partie décimale de X	Réel	Réel	FARC (2.65) = 0.65 FRAC (1.1) = 0.1

INT (X)	INT (X)	Retourne la partie entière de X	Réel	Réel	INT (6.8) = 6.0
SIN (X)	SIN (X)	Retourne le sinus de X (X en Radians)	Réel	Réel	SIN (PI/2) = 1.0 SIN (PI) = 0.0
COS (X)	COS (X)	Retourne le cosinus de X (X en Radians)	Réel	Réel	COS (PI/2) = 0.0 COS (PI) = 1.0

3. Le type booléen :

a) Domaine :

Une variable de type booléen ne peut prendre que l'une des valeurs suivantes : **VRAI** (TRUE) ou **FAUX** (FALSE)

b) Opérateurs logiques sur les booléen :

NON (NOT) : négation

ET (AND) : conjonction

OU (OR) : disjonction

OUex (XOR) : ou exclusif

X	Y	NON(X)	X ET Y	X OU Y	X Ouex Y
F	F	V	F	F	F
F	V	V	F	V	V
V	V	F	V	V	F
V	F	F	V	V	V

Remarque :

Il existe un ordre de priorité entre les opérateurs logiques :

1. NON
2. ET
3. OU, OUex

c) Déclaration :

↪ **Au niveau analyse :** _(Tableau de déclaration des objets)

Objets	Type / Nature	Rôle
Ident_variable	Booleen	Rôle

↪ **Au niveau PASCAL :** **VAR** <nom de variable> : **BOOLEAN** ;

Exemple : **VAR** trouve : **BOOLEAN** ;

Application: soit la séquence d'affectation suivante :

- 1) X ← 2
- 2) Y ← 9
- 3) Z ← NON(X > Y)
- 4) M ← (X < Y) ET (X < 0)
- 5) N ← (X < Y) OU NON (5 IN [0..6])
- 6) T ← (Y > X * 4) OU (X > 5) ET Z
- 7) P ← $\sqrt{2 * Y} + 2$

Z = vrai

M = faux

N = vrai

T = vrai

P = vrai

4. Le type caractère :

a) Domaine :

Un caractère est présenté par le caractère lui-même placé entre apostrophes.

Une variable de type caractère peut contenir l'un des caractères suivants :

- Les lettres de l'alphabet majuscule ou minuscule : "A", "B", ..., "Z" ou "a", "b", ..., "z"
- Les chiffres : "1", "2", "3", ..., "9"
- Les signes de ponctuation : ";", " ", " . ", "?",
- Les caractères spéciaux : "&", "\$", "#",

Remarque :

↪ L'espace est un caractère « blanc »

↪ Tous ces caractères sont ordonnés selon leur code ASCII

b) Les opérateurs sur les caractères :

Type	Opérateurs
Opérateurs relationnels	<, >, <=, >=, =, <>, Dans(IN)

c) Déclaration :

↪ Au niveau analyse : (Tableau de déclaration des objets)

Objets	Type / Nature	Rôle
Ident_variable	Caractère	Rôle

↪ Au niveau PASCAL :

VAR <nom de variable> : CHAR ;

Exemple : VAR c1, c2 : CHAR ;

d) Les fonctions prédéfinies sur les caractères:

NOM	ROLE	TYPE DONNEE	TYPE RESULTAT	EXEMPLES
Maj (X) UPCASE (X)	Convertie le caractère contenu dans X en majuscules	Caractère	Caractère	UPCASE ('a') = 'A' UPCASE ('A') = 'A'
ORD (X)	Retourne le code ASCII du caractère existant dans X	Caractère	Entier	ORD ('A') = 65 ORD ('a') = 97
CHR (X)	Retourne le caractère dont le code ASCII est X	Entier	Caractère	CHR (65) = 'A' CHR (68) = 'D'
PRED (X)	Retourne le prédécesseur (ce qui précède) de X s'il existe	Scalaire	De même type que X	PRED (1) = 0 PRED (B) = 'A' PRED (TRUE) = FALSE
SUCC (X)	Retourne successeur (le suivant) de X s'il existe	Scalaire	De même type que X	SUCC (FALSE) = TRUE SUCC (29) = 30 SUCC ('A') = 'B'

5. Le type chaîne de caractère :

a) Définition :

Une chaîne de caractère est une entité composée d'une suite de n caractères, avec compris entre 0 et 255.

Si n = 0 on dit que la chaîne est vide

Remarque :

- Les valeurs de chaîne de caractères sont définies entre guillemets.
- Si une apostrophe doit figurer dans une chaîne de caractères, **il faut la doubler en PASCAL.**
Exemple : 'l''informatique'

b) Déclaration :

↳ **Au niveau analyse :** (Tableau de déclaration des objets)

Objets	Type / Nature	Rôle
Identificateur_chaine	Chaîne	Rôle
Identificateur_chaine	Chaîne [Lmax]	Rôle

Remarques :

- **Lmax** : est un entier compris entre 0 et 255.
- Une déclaration de chaîne sans spécification de longueur équivalent à une déclaration de type chaîne[255].

Exemple : Tableau de déclaration des objets :

Objets	Type / Nature	Rôle
Nom	Chaîne [10]	Chaîne pouvant contenir jusqu'à 10 caractères
ch1	chaîne	Chaîne pouvant contenir jusqu'à 255 caractères

↳ **Au niveau PASCAL :**

```
VAR < Identificateur_chaine > : STRING;
    < Identificateur_chaine > : STRING[Lmax] ;
VAR nom : STRING[10] ;
    ch1 : STRING;
```

c) Accès aux éléments d'une chaîne:

On pourra accéder en lecture et en écriture au $i^{\text{ème}}$ caractère d'une chaîne CH en utilisant la notation CH[i] où $1 \leq i \leq \text{long}(\text{CH})$.

Exemple :

```
Nom ← "Informatique"
Nom[1] donne le caractère "I"
Nom[4] donne le caractère "o"
```

d) Les fonctions et les procédures standard sur les chaînes:

NOM ALGORITHME	NOM PASCAL	ROLE	EXEMPLES
LONG (CH)	LENGTH (CH)	Retourne le nombre de caractère CH	L := LENGTH ('Informatique') ; ⇒ L := 12
CONCAT (CH1, ..., CHn)	CONCAT (CH1, ..., CHn)	Retourne une chaîne qui est la réunion (somme) de plusieurs chaînes dans l'ordre indiqué	CH := CONCAT ('LYCEE ', 'PILOTE') ⇒ CH := 'LYCEE PILOTE'
SOUS-CHAINE (CH, P, N)	COPY (CH, P, N)	Retourne une chaîne de N caractères extraite de la chaîne CH à partir de la position P	CH := COPY ('Un Journaliste', 4, 7) ⇒ CH := 'Journal'
POS (CH1, CH2)	POS (CH1, CH2)	Retourne un entier donnant la position de la première occurrence de CH1 dans CH2	P := POS ('on', 'concaténation') ⇒ P := 2 P := POS ('y', 'cercle') ⇒ P := 0
EFFACE (CH, P, N)	DELETE (CH, P, N)	Enlève N caractères de la chaîne CH à partir de la position P	CH := 'formatage' ; DELETE (CH, 7, 3) ; CH sera égal à format
INSERE (CH1, CH2, P)	INSERT (CH1, CH2, P)	Insérer CH1 dans CH2 à partir de la position P	CH := 'DIDACIEL' INSERT ('TIC', CH, 6) CH sera égal à 'DIDACTICIEL'

CONVCH (N, CH)	STR (N, CH)	Convertir une valeur numérique N en une chaîne de caractère CH	STR (2000, CH) CH sera égal à '2000'
VALEUR (CH, N, e)	VAL (CH, N, e)	Convertir une chaîne de caractères CH en une valeur numérique X	VAL ('2000', N, e) N sera égal à 2000 et e sera égal à 0 VAL ('12/01/2004', N, e) N sera égal à 0 et e sera égal à 3 (première occurrence de l'erreur)

1. Les expressions

1. Définition

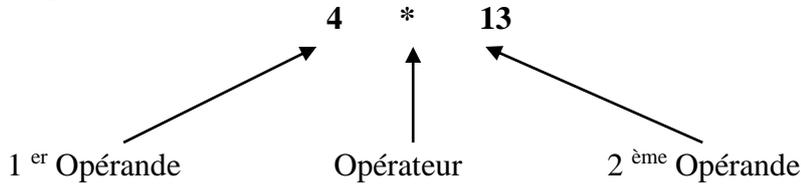
On appelle expression toute composition d'opérandes et d'opérateurs réalisant un calcul déterminé.

2. Les opérandes

L'opérande est l'élément sur lequel on applique l'opération.

Dans une expression les opérandes utilisés doivent être de même type.

Expression :



Les opérandes peuvent être des constantes, des variables, des valeurs ou des résultats envoyés par des fonctions.

3. Les opérateurs

a. Les opérateurs unaires :

Un opérateur est dit unaire s'il est appliqué à un seul opérande

Opérateur	Opérande	Type résultat
-	Entier/Réel	Entier/Réel

b. Les opérateurs binaires :

Les opérateurs binaires multiplicatifs

Opérateur	Type opérande 1	Type opérande 2	Type résultat
*	Entier	Entier	Entier
	Réel	Réel	Réel
	Réel	Entier	Réel
	Entier	Réel	Réel
/	Entier	Entier	Réel
	Entier	Réel	Réel
	Réel	Entier	Réel
	Réel	Réel	Réel
DIV	Entier	Entier	Entier
MOD	Entier	Entier	Entier

Les opérateurs binaires additifs

Opérateur	Type opérande 1	Type opérande 2	Type résultat
+	Entier	Entier	Entier
	Entier	Réel	Réel
	Réel	Entier	Réel
	Réel	Réel	Réel

Les opérateurs relationnels

Opérateur	Code PASCAL	Type résultat
<, <=, >, >=, =, <>	<, <=, >, >=, =, <>	Logique

Les opérateurs logiques

Opérateur	Type opérande 1	Type opérande 2	Type résultat
ET	Booléen	Booléen	Booléen
OU	Booléen	Booléen	Booléen
OUex	Booléen	Booléen	Booléen

4. Evaluation d'une expression

Pour évaluer une expression il faut respecter l'ordre de priorité des opérateurs.

- 1) Les parenthèses
- 2) Les opérateurs unaires
- 3) Les opérateurs multiplicatifs
- 4) Les opérateurs additifs
- 5) Les opérateurs relationnels

Leçon 3 : Les tableaux

Activité

On se propose de calculer la moyenne en informatique de 5 élèves. Proposez une solution en précisant le type des variables utilisées.

Pour calculer les moyennes des 5 élèves, on peut utiliser 5 variables (moy1, moy2, moy3, moy4, moy5) de type réel.

Si le nombre d'élèves est plus grand (par exemple 30)

Pour résoudre ce problème, on propose de déclarer un tableau de 30 éléments pour stocker les moyennes au lieu d'utiliser 30 variables différentes.

Ces moyennes se distinguent avec des indices allant de 1 à 30

MOY	15.57	15.43	14.98	13.5	12.90	9.78	9.48
	1	2	3	4	5	29	30

15.57 correspond au 1^{er} élément du tableau MOY

14.98 correspond au 3^{ème} élément du tableau MOY

9.48 correspond au 30^{ème} élément du tableau MOY

I- Présentation

Un tableau est une structure de données permettant de ranger un nombre fixe d'éléments de même type. Chaque élément du tableau est désigné par un indice qui doit être forcément de type scalaire (entier, caractère).

Un tableau est caractérisé par :

- ⇒ Son nom (identificateur)
- ⇒ Le nombre de ses éléments
- ⇒ Le type de ses éléments.

II- Déclaration

Au niveau analyse : (Tableau de déclaration des objets)

Objets	Type / Nature	Rôle
Identificateur_Tableau	Tableau de taille de Type éléments	Rôle

Exemple : Tableau de déclaration des objets :

Objets	Type / Nature	Rôle
MOY	Tableau de 30 Réels	Tableau pouvant contenir la moyenne de 30 élèves

↪ Au niveau PASCAL :

VAR < Identificateur_Tableau > : **ARRAY**[Borne_Inf..Borne_Sup] **OF** TYPE;

Exemple

VAR MOY : **ARRAY**[10] **OF** Real;

III- Accès aux éléments d'un tableau:

On pourra accéder en lecture et en écriture au i^{ème} élément d'un tableau T on utilisant la notation T[i] où $1 \leq i \leq$ Taille du Tableau T.

Exemple :

MOY[1] Donne 15.57

MOY[3] Donne 14.98

MOY[4] ← 15.02 le tableau devient :

MOY	15.57	15.43	14.98	15.02	12.90	9.78	9.48
	1	2	3	4	5	29	30

IV- Application

Soit la séquence d'affectation suivante :

1. $A[1] \leftarrow \text{"Tunisie"}$
2. $A[2] \leftarrow \text{"Tozeur"}$
3. $A[3] \leftarrow \text{SOUS-CHINE}(A[1],1,3)$
4. $\text{EFFACE}(A[4], 4,3)$
5. $B[1] \leftarrow \text{LONG}(A[1])$
6. $B[2] \leftarrow \text{LONG}(A[4])$
7. $B[3] \leftarrow B[1] \text{ DIV } B[2]$
8. $B[3] \leftarrow B[3] * 15$

Questions

1. Déclarez les deux tableaux A et B
2. Donnez le contenu de chaque tableau après exécution des affectations précédentes.