

Lycée Secondaire Raccada



Chapitre1

Structures de contrôle 1

Lecon1

Les Structures de données

4^{ème} Sc

Enseignant: Sami MEJRI

2015-2016

Introduction

Lancer l'application « Calculatrice » puis faire l'opération 6+8.

1. On a saisi des nombres (6 et 8) : ce sont des données.

2. 6 est effacé de l'écran, mais l'ordinateur se souvient de la valeur.

➤ **Le nombre 6 est sauvegardé dans la mémoire (RAM), dans un objet.**

➤ **Les applications manipulent des données contenues dans des objets.**

On distingue deux catégories d'objets : les variables et les constantes.

1. Les constantes et les variables

1- Les constantes

Définition

On appelle constante un objet ayant une valeur inchangée tout le long d'un algorithme

Caractéristiques

Une constante est caractérisé par:

- **Son nom (un identificateur unique)**
- **Sa valeur (le contenu inchangé dans la case mémoire)**

Exemple: $\text{Pi} = 3.14$

Déclaration

▪ Déclaration algorithmique (TDO)

Objets	Type/Nature
nom de la constante	Constante = valeur

Exemple

Objets	Type/Nature
Pi	Pi = 3.14

▪ Déclaration en Pascal

```
Const  
nom_constante = valeur_constante;
```

Exemple

```
Const  
Pi = 3.14;
```

2- Les variables

Définition

On appelle variable tout objet pouvant prendre différents valeurs tout le long d'un algorithme

Caractéristiques

Une variable est caractérisé par:

- **Son nom (un identificateur unique)**
- **Son type**
- **Son contenu**

Déclaration

▪ Déclaration algorithmique (TDO)

Objets	Type/Nature
Nom_variable	Type_variable

Exemple

Objets	Type/Nature
rang	Entier

▪ Déclaration en Pascal

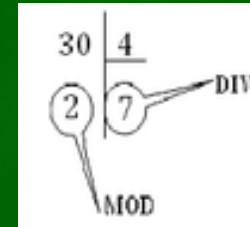
```
Var  
nom_variable : type_variable;
```

Exemple

```
Var  
rang : integer;
```


a- le type entier

- ❖ Les variables du type entier forment un sous-ensemble de l'ensemble des nombres entiers relatifs (\mathbb{Z}).
- ❖ Un entier est représenté sur 16 bits (2 octets) et il admet une borne minimale et une borne maximale. **Max= 32767** **Min= -32768**
- ❖ Les operateurs arithmétiques sur les entiers sont: +, -, *, /
- ❖ Les operateurs de la division entière sont: DIV (*retourne le quotient*), MOD (*retourne le reste*)
- ❖ Les opérateurs relationnels sont: <, >, =, <=, >=, <>



Déclaration

▪ Déclaration algorithmique (TDO)

Objets	Type/Nature
nom_variable	entier

▪ Déclaration en Pascal

```
Var  
nom_variable :integer;
```

Exemple

Objets	Type/Nature
x,y	Entier

```
Var  
x,y : integer;
```

b- le type réel

- ❖ Le réel forme un sous-ensemble de l'ensemble de nombres réels \mathbb{R}
- ❖ Les opérateurs arithmétiques applicables sur le réel sont: +, -, *, /
- ❖ Les opérateurs relationnels applicables sur le réel sont: <, >, =, <=, >=, < >

Déclaration

▪ Déclaration algorithmique (TDO)

Objets	Type/Nature
nom_variable	réel

▪ Déclaration en Pascal

```
Var  
nom_variable :real;
```

Exemple

Objets	Type/Nature
moy	Réel

```
Var  
moy: real;
```

- ❖ les fonctions arithmétiques standards

Application

On considère la fonction mathématique suivante

$$F(x) = \frac{|x^2 - 5x|}{\sqrt{3x + 2}}$$

On suppose que $f(x) = y$

En algorithmme

$Y \leftarrow Abs (carré(x) - 5 * x) / racine\ carré (3*x + 2)$

En pascal

$Y := Abs (sqr(x) - 5 * x) / sqrt (3*x + 2)$

c- le type booléen

Le type booléen contient les deux valeurs logiques Vrai (True) et Faux (False)

Déclaration

▪ Déclaration algorithmique (TDO)

Objets	Type/Nature
nom_variable	booléen

▪ Déclaration en Pascal

```
Var  
nom_variable :boolean;
```

Application

Remplir le tableau suivant :

<i>Expression</i>	<i>Résultat</i>
$(5 > 6) \text{ AND } (5 \text{ DIV } 6 = 5)$	F
$(3 = \text{SQRT}(9)) \text{ OR } (14 < 4)$	V

d- le type caractère

Une variable de type caractère est représentée par un seul caractère alphanumérique placé entre deux guillemets en algorithmique et deux apostrophes (' ') en Pascal.

Exemples : "A", "g", "5", "!",...

❖ On accorde à chaque caractère un code ASCII variant de 0 à 255 pour l'arranger.

Déclaration

▪ Déclaration algorithmique (TDO)

Objets	Type/Nature
nom_variable	Caractère

▪ Déclaration en Pascal

```
Var  
nom_variable :char;
```

❖ Puisque les valeurs de type caractère sont ordonnées on peut y appliquer les opérateurs relationnels :

= ≠ < ≤ > ≥

Comparer :

"a" > "A" puisque 97 > 65

"," < ";" puisque 44 < 59

On peut appliquer l'opérateur « + » sur deux caractères, c'est la concaténation.

Exemple :

"O"+ "2" = "O2"

Le type du résultat de la concaténation n'est pas caractère.

Les fonctions prédéfinies :

Application

Donner le contenu des variables C et D suivantes :

C <-- Majus (Succ (Chr (97)))

D <-- Ord (C) - 64

e- le type chaîne de caractère

Une chaîne de caractères est une entité composée d'une suite de n caractères ($0 \leq n \leq 255$).

❖ Si n est nul on dit que la chaîne est vide.

Exemple : CH <-- "PASCAL"

- Une chaîne de caractères est écrite entre deux guillemets.
- Quel est le 3^{ème} caractère de la chaîne ? C'est : "S"
- Pour désigner le 3^{ème} caractère de la chaîne CH on écrit : CH [3]

Déclaration

▪ Déclaration algorithmique (TDO)

Objets	Type/Nature
nom_variable	Chaîne de caractère
nom_variable	Chaîne de n caractère

▪ Déclaration en Pascal

```
Var  
nom_variable :string;  
nom_variable :string[n];
```


a- Les opérateurs relationnels :

Les opérateurs sont : = ≠ < ≤ > ≥

Exemple : C ← "projet" < "programme"

C reçoit la valeur Faux

b- Fonctions et procédures prédéfinis

Application

1. Déclarer une variable pouvant contenir une adresse de 60 caractères au maximum.

2. Soit Ch une variable de type chaîne de caractères : Ch ← "BAC"

- Remplacer le 3^{ème} caractère de la chaîne Ch par "S".

Ch[3] ← "S"

- Afficher sur l'écran le 2^{ème} caractère de la chaîne.

Ecrire (Ch [2])

Remarque

Il est possible d'accéder en lecture ou en écriture au i^{ème} caractère d'une chaîne CH donnée en utilisant la notation CH[i] ou $1 \leq i \leq \text{long}(ch)$

f- Le type scalaire énuméré :

C'est un ensemble ordonné et fini de valeurs désignées par des identificateurs.

Déclaration

▪ Déclaration algorithmique

Tableau de déclaration des nouveaux types : **TDNT**

Type
$Nom_Type = (valeur1, valeur2, valeur3, \dots)$

TD0

Objets	Type/Nature
nom_variable	Nom_type

▪ Déclaration en Pascal

```
Type  Nom_Type = (valeur1, valeur2, vailleur3, ...);  
Var   Nom_variable : Nom_type ;
```

g- Le type intervalle

Le type intervalle est décrit par la donnée de deux constantes « borne inférieur » et « borne supérieur » appartenant à un type scalaire discret ordonné telle que $B_{inf} < B_{sup}$.

Déclaration

▪ Déclaration algorithmique

TDNT

Type
$Nom_Type = Borne_inf .. Borne_sup$

TDO

Objets	Type/Nature
nom_variable	Nom_type

▪ Déclaration en Pascal

```
Type  Nom_Type = Borne_inf .. Borne_sup ;  
Var   Nom_variable : Nom_type ;
```

h- Le type Tableau :

On veut écrire un programme permettant de calculer la somme de 3 puis de 100 nombres.

Combien d'objets on va utiliser à chaque fois ?

Est-ce que on va s'amuser à déclarer 100 variables ?

On doit utiliser une nouvelle structure de données capable de regrouper toutes ces variables : tableau

Définition

Un tableau est une structure de données permettant de ranger un nombre fini d'éléments de même type.

Un tableau est caractérisé par : un nom, une taille et le type de ses éléments.

Déclaration

Première formulation :

TDO

Objets	Type/Nature
nom_tableau	Tableau de taille - type

Var

Nom_tableau : array [Bi..Bs] of type ;

Deuxième formulation

TDNT

Type
<i>Nom_type = Tableau de taille - type</i>

TDO

Objets	Type/Nature
nom_tableau	Nom_type

Type *Nom_type = array [Bi..Bs] of type ;*
Var *Nom_tableau : Nom_type*

**Merci pour votre
attention**

leçon prochaine

Les structures simples