

Leçon 3: Le type scalaire énuméré et le type intervalle

I-Le type scalaire énuméré:

1-Définition: Le type scalaire par énumération définit un ensemble ordonné fini de valeurs désignées par des identificateurs.

Activité:

Définir le type jour contenant les jours de la semaine.

Tableau de déclaration des nouveaux types.

| Types |
|---|
| Jour=(Lundi, Mardi, Mercredi, Jeudi, Vendredi, Samedi,Dimanche) |

*On peut appliquer la fonction **Ord** sur ses valeurs pour déterminer leurs numéro d'ordre.

Exemple: Ord(Lundi)=0 Ord(Mardi)=1

*Les opérateurs applicables à ses valeurs sont:

- Les opérateurs de relation >, <, <=, >=, =, <>
- Les opérateurs **Pred** et **Succ**

Déclaration en Pascal:

Type

<nom_type>= (constante_1,constante_2,,constante_n);

var <nom_variable> : nom_type ;

Exemple:

Type

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

var j: jour;

Remarque : On ne peut ni lire (saisir) ni écrire (afficher) une variable de type scalaire énuméré.

~~WriteLn(j)~~

~~ReadLn(j)~~

II-Type intervalle:

1-Définition:

Un intervalle est décrit par la donnée de deux constantes « Borne inf » et « borne sup » (tq borne inf <borne sup) appartenant à un type scalaire discret ordonné (entier, caractère et scalaire énuméré)

Activité: Définir le type mois, le type weekend et le type lettres.

Tableau de déclaration des nouveaux types.

| Types |
|---|
| Jour=(Lundi, Mardi, Mercredi, Jeudi, Vendredi, Samedi,Dimanche) |
| week_end = samedi..dimanche |
| mois = 1..12 |
| lettres="a".."z" |

Déclaration en Pascal:

Type <nom_type>=borne_inf..borne_sup ;

var <nom_variable>: nom_type ;

Exemple:

type

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

week_end=samedi..dimanche;

mois=1..12; lettres ='a'..'z' ;

var j: jour; m : mois ; w : week_end ; l : lettres ;

Leçon 4:

Les Tableaux

Activité :

On veut écrire un programme qui permet de saisir puis afficher la moyenne de 5 élèves d'une classe dans une matière donnée.

Définition : un tableau est une structure de données permettant de ranger un nombre fini d'éléments de même type et selon une disposition bien définie.

Remarque: On se limite à l'étude des tableaux unidimensionnels appelés aussi vecteur.

I-Déclaration d'un tableau:

→ Au niveau de l'analyse et de l'algorithme :

Tableau de déclaration des objets

| Objet | Type/Nature |
|---------------|--------------------------------------|
| Ident_tableau | Tableau de taille et de type_element |

→ Au niveau du Pascal :

Var Ident_tableau : ARRAY [B_inf..B_sup] of type_element;

Exemple : Tableau de déclaration des objets

| Objet | Type/Nature | Rôle |
|---------|--------------------|-------------------------------|
| Moyenne | Tableau de 5 réels | Stocker les notes de 5 élèves |

En pascal :

Var Moyenne : ARRAY [1..5] OF REAL ;

Remarques :

-Pour accéder au ième élément du tableau, il suffit de donner l'identificateur de tableau et l'indice i .

-Nous pouvons lire et écrire un élément du tableau

Moyenne[4]← 13.5

-Les opérations possible sur un élément d'un tableau sont les même que celles définies sur une variable de même type.

Exercice:

On a deux classes pour les quels on veut stocker le nom de l'élève sa moyenne et son rang,

II-Le type tableau:

Autres déclarations d'un tableau :

Tableau de déclaration de nouveau types

| TYPES |
|--|
| Classe=tableau de 20 chaînes de caractères |
| T_Moyenne= tableau de 20 réels |
| T_rang=tableau de 20 entiers |

Tableau de déclaration des objets :

| Objet | Type/Nature | rôle |
|---------|-------------|--|
| CL1 | Classe | Tableaux des noms des élèves |
| CL2 | Classe | |
| moyCl1 | T_moyenne | Tableaux des moyennes de cl1et cl2 |
| moyCl2 | T_moyenne | |
| rangCl1 | T_rang | Tableaux des rangs des élèves de cl1et cl2 |
| rangCl2 | T_rang | |

En pascal:

Type classe=array [1..5] of real;

var cl1:classe;

Exercice :

Soit la séquences suivante :

- 1)T[1]←30
- 2)T[2]← 7
- 3)T[3]← T[1] div T[2]
- 4)T[4]← T[3]*10
- 5)T[5]← T[1]+T[3]*T[4]

1) Déclarer le tableau T.

2) Quel est le contenu de chaque élément du tableau?