

## **Chapitre 6 : Les enregistrements**

Contenu du chapitre :

1. Définition d'un enregistrement
2. Manipulation des variables de type enregistrement
3. Tableau d'enregistrements
4. Enregistrement membre d'un autre enregistrement
5. La structure AVEC - FAIRE
6. Passage d'un enregistrement à une procédure ou fonction
7. Exercices

### **1. Définition**

Un enregistrement est un ensemble de variables (de champs) de types différents constituant une entité.

#### **Syntaxe de déclaration :**

```
TYPE nom_type_enregistrement = ENREGISTREMENT  
    Champ1 : Type1 ;  
    Champ2 : Type2 ;  
    ...  
    ChampN : TypeN ;  
FIN ;
```

En langage C :

```
STRUCT nom_type_enregistrement  
{  
    Type1 Champ1 ;  
    Type2 Champ2 ;  
    ...  
    TypeN ChampN ;  
};
```

**Exemple :**

**TYPE** Facture = **Enregistrement**

NoFacture : Entier ;

Montant : Réel ;

**FIN ;**

Une fois l'enregistrement est défini, on peut déclarer des variables de type de cet enregistrement.

**Exemple :**

**VAR** Facture1, Facture2 : Facture ;

Il est à noter que « Facture1 » et « Facture2 » sont deux variables de type « Facture ». Chacune de ces variables contient un numéro de facture et un montant.

Dans un enregistrement, **chaque champ doit avoir un identificateur unique**. Mais, pour des enregistrements différents, **on peut réutiliser le même identificateur de champ** avec un type différent ou identique.

**Exemple :**

**TYPE** Enregistrement1 = **ENREGISTREMENT**

a : Entier ;

b : Réel ;

c : Char ;

**FIN ;**

**TYPE** Enregistrement2 = **ENREGISTREMENT**

a : Char ;

b, c : Réel ;

**FIN ;**

## **2. Manipulation des variables de type enregistrement**

La manipulation des variables de type enregistrement se fait champ par champ. Et l'accès à un champ donné se fait comme suit :  
« *Variable\_enregistrement.champ* ».

**Exemple :**

**ALGORITHME** exemple1;

```

TYPE Compte = ENREGISTREMENT
    NoCompte : Entier ;
    Nom : Chaîne de caractères ;
FIN ;
VAR Client : Compte ;
DEBUT
    Ecrire ('Entrez le numéro de compte et nom du client');
    Lire (Client.NoCompte) ;
    Lire (Client.nom) ;
    Ecrire ('Le numéro de compte du client est : ', Client.NoCompte);
    Ecrire ('Le nom du Client est : ', Client.Nom);
FIN.

```

### 3. Tableau d'enregistrements

Il est possible de déclarer un tableau de sorte que chaque case du tableau est un élément de type enregistrement.

#### Syntaxe de déclaration :

```

VAR Nom_tableau : Tableau [1 .. dimension] de Nom_type_enregistrement ;

```

L'accès à un champ d'un tableau d'enregistrements se fait comme suit : « *Nom\_tableau [indice].Champ* ». Où « indice » désigne l'indice de l'<sup>i</sup><sup>ème</sup> élément du tableau.

#### Exemple :

```

ALGORITHME exemple2 ;
CONST dim = 3 ;
TYPE Compte = ENREGISTREMENT
    NoCompte : Entier ;
    Nom : Chaîne de caractères ;
FIN ;
VAR
    C : Tableau [1.. dim] de Compte ;

```

```

    i : Entier ;
DEBUT
    Pour i allant de 1 à dim faire
        Ecrire ('Entrez le numéro de compte du client') ;
        Lire (C[i].NoCompte) ;
        Ecrire (' Entrez le nom du client') ;
        Lire (C[i].Nom) ;
    Fin Pour ;
    Ecrire ('Voici le tableau d'enregistrements saisi : ') ;
    Pour i allant de 1 à dim faire
        Ecrire ('Numéro de compte du client ', i, ' est : ', C[i].NoCompte) ;
        Ecrire (' Entrez le nom du client') ;
        Ecrire ('Le nom du client ', i, ' est : ', C[i].Nom) ;
    Fin Pour ;
FIN.

```

#### 4. Enregistrement membre d'un autre enregistrement

Un champ d'un enregistrement peut lui-même être de type enregistrement. Dans ce cas, le type de l'enregistrement champ doit être déclaré avant le type de l'enregistrement qui le contient.

De plus, on accède à un membre de sous-enregistrement comme suit : « *variable\_enregistrement.variable\_sous\_enregistrement.Champ* ».

**Exemple :** Algorithme qui permet de saisir une variable de type enregistrement, puis l'affiche. L'enregistrement utilisé contient lui-même un champ de type enregistrement.

```

ALGORITHME exemple3 ;
    TYPE Date = ENREGISTREMENT
        Jour : 1 .. 31 ;
        Mois : 1 .. 12 ;
        Année : 2000 .. 2021 ;
    FIN ;

```

**TYPE** Compte = **ENREGISTREMENT**

NoCompte : Entier ;

Nom : Chaîne de caractères ;

DateOuverture : Date ;

**FIN ;**

**VAR** Client : Compte ;

**DEBUT**

Ecrire ('Entrez le numéro de compte du client') ;

Lire (Client.NoCompte) ;

Ecrire ('Entrez le nom du client') ;

Lire (Client.Nom) ;

Ecrire ('Entrez le jour d'ouverture du compte ') ;

Lire (Client.DateOuverture.Jour) ;

Ecrire ('Entrez le mois d'ouverture du compte ') ;

Lire (Client.DateOuverture.Mois) ;

Ecrire ('Entrez l'année d'ouverture du compte ') ;

Lire (Client.DateOuverture.Année) ;

Ecrire ('Voici l'enregistrement saisi') ;

Ecrire (Client.NoCompte, ' ', Client.Nom) ;

Ecrire (Client.DateOuverture.Jour, ' / ', Client.DateOuverture.Mois, ' / ', Client.DateOuverture.Année) ;

**FIN.**

## **5. La structure AVEC - FAIRE**

La structure AVEC-FAIRE est utilisée pour parcourir les champs d'un enregistrement d'une manière automatique.

### **Syntaxe de déclaration :**

**AVEC** enregistrement **FAIRE** instruction(s) ;

### **Exemple :**

```

AVEC Client FAIRE
    Ecrire (NoCompte) ;
    Ecrire (Nom) ;
    Ecrire (DateOuverture.Jour,' / ', DateOuverture.Mois,' / ', DateOuverture.Année) ;
FIN AVEC ;

```

Il est à noter que la structure AVEC-FAIRE n'existe pas en langage C.

## 6. Passage d'un enregistrement à une procédure ou fonction

Une procédure ou une fonction peut recevoir une variable de type enregistrement comme paramètre.

### Syntaxe de déclaration :

```

TYPE Nom_type_enregistrement = ENREGISTREMENT
    Champ1 : Type1 ;
    Champ2 : Type2 ;
    ...
    ChampN : TypeN ;
FIN ;
PROCEDURE nom_procedure (nom_variable : nom_type_enregistrement); (Ou FONCTION nom_fonction (nom_variable : nom_type_enregistrement) : type ;)
DEBUT {Début de la procédure ou fonction}
    Instruction(s) ;
FIN ; {Fin de la procédure ou fonction}

```

### Exemple :

```

ALGORITHME exemple4 ;
TYPE Compte = ENREGISTREMENT
    NoCompte : Entier ;
    Nom : Chaîne de caractères ;
    Solde : Réel ;
FIN ;
VAR Client : Compte ;
PROCEDURE Proc_affichage (x : Compte) ;

DEBUT

```

```

Ecrire ('Affichage dans la procédure : ');
Ecrire (' Le numéro de compte du client : ', x.NoCompte);
Ecrire (' Le nom du client : ', x.Nom);
Ecrire (' Le solde du client : ', x.Solde);

```

**FIN ;**

**DEBUT**

```

Ecrire ('Affichage dans l'algorithme principal');

```

**Avec Client faire**

```

Lire (NoCompte);

```

```

Lire (Nom);

```

```

Lire (Solde);

```

**Fin Avec ;**

```

Proc_affichage (Client);

```

**FIN.**

La procédure « proc\_affichage » reçoit la variable « Client » de type compte, la copie dans sa variable locale « x », puis l'affiche ;

On ne peut pas retourner une variable de type enregistrement à l'aide d'une fonction, autrement que par adresse. Ce point sera abordé en détails dans le chapitre des pointeurs.

## 7. Exercices

- A.** Soit un enregistrement « Client » contenant les champs suivants : *Nom*, *Prénom*, et *Téléphone*. Ecrire un algorithme qui permet de saisir et d'afficher un client.
- B.** Soit un enregistrement « étudiant » contenant les champs suivants : *Nom*, *Prénom*, et *Moyenne* (comprise entre 0 et 20), et *Adresse*. L'enregistrement « Adresse » contient les champs suivants : *Numéro de rue*, *Nom de rue*, et *Code postal*. Ecrire un algorithme qui permet de saisir et d'afficher un tableau d'étudiants de dimension 3. L'affichage doit se faire en utilisant un sous-algorithme.

### Orientation et remédiation :

- Si l'étudiant n'arrive pas à faire la partie « A » de l'exercice. Il est conseillé de refaire la lecture et analyse de la partie « Type composés » et les Sections 1 et 2 de la partie « Enregistrements » de ce chapitre.
- Si l'étudiant n'arrive pas à faire la partie « B » de l'exercice. Il est conseillé de refaire la lecture et analyse des Sections 3, 4, 5, et 6 de la partie « Enregistrements » de ce chapitre.

- *Si l'étudiant a pu terminer l'exercice, il est conseillé de passer au chapitre suivant.*