Chapitre 6: Les types personnalisés

1. Les structures (enregistrements):

- Un enregistrement (ou ***structure***) est un type qui permet de stocker plusieurs données, de même type ou de types différents.
- Un enregistrement comporte des composants appelés ***champs***, chaque champ correspond à une donnée.

Exemple :

Un enregistrement Etudiant contient quatre champs (Nom, Prénom, DateNaissance, moyenne).

Nom
Prénom
DateNaissance
moyenne

1.1. Déclaration d'un enregistrement :

Type

Structure Nom_TypeEnregistrenement
Champ1 : type1;
Champ2 : type2;
.
.
.
Champ n : type n;
FinStructure ;

- Le mot clé **Type** est commun à tous les nouveau type qu'on veut rajouter au compilateur ;
- Le mot clé **Structure** indique le début de la déclaration d'une structure ;
- Nom_TypeEnregistrenement est le nom du nouveau type ;
- Le mot clé **FinStructure** indique la fin de l'enregistrement.

Exemple :

On suppose qu'on veut écrire un programme qui manipule les informations des étudiants à savoir : **le nom**, **le prénom**, **le numéro de carte**, **Numéro de téléphone** et les **8 notes** de chaque étudiant.

Donc, les données de chaque étudiant sont :

- nom : chaine de caractères
- prénom : chaine de caractères
- numéro de carte : entier

- numéro de téléphone : entier
- note : tableau de réel

On déclare la structure **Etudiant** comme suit :

Type

Structure Etudiant

Nom : chaîne ;

Prénom : chaîne **;**

NumCart : entier ;

NumeroTel :Chaîne ;

Note[8] : tableau de réel ;

FinStructure ;

Remarque :

- on peut déclarer plusieurs structures (enregistrements), qui peuvent êtres en relation les un des autres.

Par exemple, on peut décomposer la structure **Etudiant** déclarée précédemment en deux structures comme suit :

Type

Structure IdentEtudiant

Nom : chaîne ;

Prénom :Chaîne **;**

FinStructure ;

Structure Etudiant

NumCart : entier ;

Identité : identEtudiant ;

NumeroTel :Chaîne ;

Note[8] : tableau de réel ;

FinStructure ;

- la structure **Etudiant** contient un champ (Identité) qui est de type **IdentEtudiant** , qu'on a déclaré avant **Etudiant.**

Après la déclaration des types, on fait la déclaration des variables qu'on va l'utiliser :

etud1, etud2, etud3 : Etudiant ;

- Les trois variables **etud1**, **etud2** et **etud3** sont de type **Etudiant** donc chaque un de ces trois variables a la structure suivante :

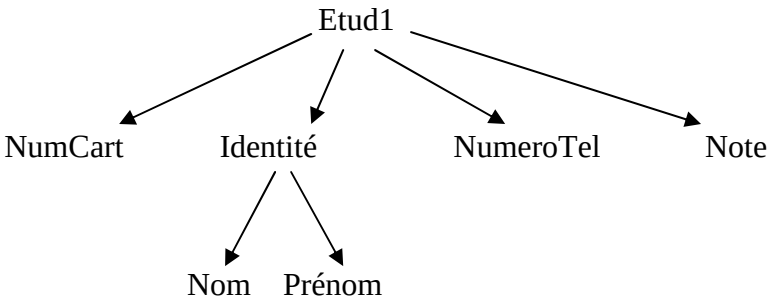
NumCart
Identité Nom Prénom
NumeroTel
Note

1.2. Manipulations des enregistrements :

a) Accès à champs d'un enregistrement :

- On peut représenter une variable de type enregistrement (structure) sous forme d'arbre de champs.

Exemple : La variable Etud1 peut être représentée comme suit :



- On accède à un champ en précisant le nom de la variable de type enregistrement suivie de l'identificateur du champ séparé par un point (.):

Exemple :

- Pour accéder au Numéro de carte de l'étudiant Etud1, on écrit :

Etud1.NumCart

- Pour accéder au nom, on écrit :

Etud1.Identité.Nom

- Pour accéder au troisième élément du tableau Note, on écrit:

Etud1.Note[3]

b) Lecture et écriture des enregistrements :

- Par analogie aux tableaux les champs d'une structure sont lies ou affichés un par un car on ne peut lire ou afficher que les types simple définis par le compilateur.

- **Lecture :**

Exemple :

- Pour remplir la variable Etud1 de l'exemple précèdent depuis l'entrée standard on procède comme suit :

```
Ecrire('Donnez SVP le numéro de carte :') ;
Lire(Etud1.NumCart) ;
Ecrire('Donnez SVP le nom :') ;
Lire(Etud1.Identité.Nom) ;
Ecrire('Donnez SVP le Prénom :') ;
Lire(Etud1.Identité.Prénom) ;
Ecrire('Donnez SVP le Numero de Telephone :') ;
Lire(Etud1.NumTel) ;
Pour i allant de 1 à 8 faire
        Ecrire('Donnez SVP la ', i ,'ieme note ' ) ;
        Lire(Etud1.Note[i]);
FinPour
```

- **Ecriture :**

➢ Pour afficher la variable Etud1 de l'exemple précèdent on procède comme suit :

Exemple :

```
Ecrire ('Numero de carte :', Etud1.NumCart) ;
Ecrire ('Nom :' , Etud1.Identité.Nom) ;
Ecrire ('Prénom :' , Etud1.Identité.Prénom) ;
Ecrire ('Numero de Telephone :' ,Etud1.NumTel) ;
Pour i allant de 1 à 8 faire
        Ecrire ('La ', i,' ieme Note est :', Etud1.Note[i])
FinPour
```

Remarques:

➢ A la différence des tableaux, Il n'y a pas la possibilité d'utiliser une boucle pour manipuler tous les éléments d'un enregistrement.

➢ Dans la pratique, le nombre de champs est très réduit (5 à 20 champs).

➢ Un enregistrement peut faire l'objet d'une affectation (avec une variable de même type) :

```
Etud2←Etud1 ;   /* Ceci veut dire que tous les champs de Etud1 sont copiés vers les
                   mêmes champs de Etud2*/
```

2.4. Tableaux d'enregistrements :

➢ Il est possible de déclarer un tableau dont les éléments sont de type enregistrement.

➢ Donc, On défini d'abord la structure, puis on déclare l'existence d'un tableau dont les éléments sont de ce type.

```
Type
    Structure Nom_TypeEnregistrenement
        variable 1 : type1;
        variable 2 : type2;
              .....
        variable n : type n;
    FinStructure

Nom_tableau[taille] : Tableau de Nom_Typenregistrement ;
```

- Pour sélectionner le troisième champ du quatrième élément du tableau on utilise la syntaxe :
  Non_Tableau [4]. variable3.
- **L'écriture:**
  Lire (Tab [2]. Nom) ;
  Tab [4].moyenne ← 10.5 ;
- **La lecture :**
  Ecrire (Tab [2]. Nom)
  X← Tab [2]. Prenom ;
- **Comparaison :**
  Si (Tab [2].moyenne < 10) alors

Exercice:

On veut enregistrer les résultats des étudiants du première année MI (on a 300 étudiant) pour faire quelques statistiques, pour cela, on doit connaitre pour chaque étudiant son nom, prénom, groupe, et sa moyenne générale.

1) Ecrire l'algorithme qui permet de lire les informations des étudiants puis afficher le **nom** et le **prénom** des étudiants du groupe N°8 ?

Solution :

```
Algorithme statistique
Type
Structure étudiant
    nom : chaine;
    prénom : chaine;
    groupe : entier;
    moyenne: réel
FinStructure ;

E [300]:  tableau de étudiants
Début
// la lecture des informations des étudiant
Pour i allant de 1 à 300 faire
Lire (E [ i ].nom);
Lire (E [ i ].prénom);
Lire (E [ i ].groupe);
Lire (E [ i ].moyenne);
Finpour
// l'affichage du nom et prénom des étudiants du groupe 8
Pour i allant de 1 à 300 faire
 Si (E[i].groupe = 8) alors
  Ecrire (E [ i ].nom);
  Ecrire (E [ i ].prénom);
 finsi
Finpour
Fin.
```

2) modifier l'algorithme précédant pour qu'il affiche **les étudiants admis** du groupe **8 ?**

Réponse :

```
// l'affichage du nom et prénom des étudiants admis du groupe 8
Pour i allant de 1 à 300 faire
 Si (E[i].groupe = 8) et (E[i].moyenne >= 10) alors
  Ecrire (E [ i ].nom);
  Ecrire (E [ i ].prénom);
 finsi
Finpour
```

2) modifier l'algorithme précédant pour qu'il affiche **le majeur de la promo ?**

```
// dans la partie déclaration on doit définir une variable max de type Etudiant
// l'affichage du nom et prénom du le majeur de la promo représenté par la variable max
Max ← E[1] ;
Pour i allant de 2 à 300 faire
 Si (E[i].moyenne > max.moyenne) alors
  Max ← E[i] ;
 finsi
Finpour
Ecrire (max.nom);
Ecrire (max.prénom);
```

3) modifier l'algorithme précédant pour qu'il affiche **le majeur du groupe 8 ?**

```
// Premièrement en doit chercher le premier étudiant du groupe 8 dans le tableau pour cela on va
//utiliser la boucle tant que avec une variable booléen
i←1 ; b←vrai ;
Tantque ( b= vrai) faire
Si (E[i].groupe = 8 ) alors
 Max ← E[i] ;
 B← faux ;
Finsi
i← i+1 ;
Fin tantque
// après cette boucle, max contient le premier étudiant du groupe 8 mais pas le meilleur,
// maintenant on va chercher le meilleur par une autre boucle pour comme suit
Pour i allant de 1 à 300 faire
 Si ((E[i].groupe = 8) et (E[i].moyenne > max.moyenne)) alors
  Max ← E[i] ;
 finsi
Finpour
Ecrire (max.nom);
Ecrire (max.prénom);
Fin.
```