

I- Enregistrements

1- *Notion d'Enregistrement*

Un enregistrement est une structure (complexe) de données permettant de représenter un ensemble de données (dites champs) hétérogènes, c'est-à-dire de types différents.

On peut considérer un enregistrement comme étant un tableau à une dimension, où chaque case (composante) du tableau représente un champ dans l'enregistrement. La différence réside dans :

- Une case de tableau est accessible à travers un indice entier, par contre un champ est repéré par un nom (identificateur) ;
- Les cases du tableau sont toutes du même type (structure homogène), par contre, les champs de l'enregistrement ne sont pas obligatoirement du même type (structure hétérogène).

Exemple

Pour représenter les informations liées à un produit quelconque dans une même structure, on peut utiliser un enregistrement constitué de champs suivants : (la désignation du produit, la référence du produit, sa quantité en stock, son prix unitaire, etc...) Les informations précédentes (les champs de l'enregistrement) sont de types différents, elles ne peuvent être décrites par un tableau (dont les cases sont du même type). Les types de ces informations sont respectivement : (caractères, chaîne de caractères, entier, réel, etc...)

2. *Déclaration d'un Enregistrement*

La syntaxe générale pour déclarer un enregistrement est comme suit :

En Algorithme	En Pascal
<b>Type</b> <Nom_Enreg> = <b>Enregistrement</b> <Nom_ch1> : <type1> <Nom_ch1> : <type2> ... <i_chN> : <typeN> <b>Fin</b>	<b>Type</b> <Nom_Enreg> = <b>RECORD</b> <Nom_ch1> : <type1> <Nom_ch1> : <type2> ... <Nom_chN> : <typeN> <b>End;</b>

Pour l'exemple du Produit précédent, la déclaration de l'enregistrement sera comme suit :

En Algorithme	En Pascal
<b>Type</b> <i>Produit</i> = <b>Enregistrement</b> Designation : Chaîne Reference : Chaîne Quantite : Entier Prix_U : Réel <b>Fin</b>	<b>Type</b> <i>Produit</i> = <b>RECORD</b> Designation : String [30]; Reference : String[13]; Quantite : Integer; Prix_U : Real; <b>End;</b>

3. *Accès aux champs d'un enregistrement*

Pour référencer ou accéder à un champ quelconque d'un enregistrement, on utilise la notation pointée. La notation pointée utilise la variable enregistrement (identificateur) suivie d'un point puis du nom du champ

auquel on veuille accéder. Soit : <Variable\_Enregistrement>•<Nom du champ>

**Exemple :**

Un nombre complexe peut être décrit par un enregistrement soit :

Un nombre complexe peut alors être décrit par un enregistrement de deux champs : X est la partie réelle et Y est la partie imaginaire. Par exemple :

Algorithme	Pascal
<b>PROGRAM</b> complexe <b>Type</b> COMPLEX = <b>Enregistrement</b> X, Y : Réel <b>Fin</b> <i>variables</i> NBR : Complex <b>DEBUT</b> NBR.X ← 10 NBR.Y ← 2 Écrire(NBR.X,'+i',NBR.Y) <b>FIN</b>	<b>PROGRAM</b> Ecole; <b>uses</b> wincrt; <b>Type</b> COMPLEX = <b>record</b> X,Y : real; <b>End;</b> <b>var</b> NBR: Complex ; <b>BEGIN</b> NBR.X := 10; NBR.Y := -2; write( NBR.X,'+i',NBR.Y); <b>END.</b>

#### 4- L'instruction WITH

L'instruction PASCAL WITH, permet d'éviter les répétitions de la variable enregistrement lors de l'accès à ses différents champs. Sa syntaxe générale est :

```
With <var_enreg> Do
Begin
<instructions_champs_enreg>;
End;
```

Dans le cas des deux affectations précédentes, on peut écrire :

**With NBR Do**

**Begin**

X := 10 ;

Y :=2

**End;**

#### Exemple 1

Soit un enregistrement « Produit » contenant les champs suivants : Designation (chaines de caractères de taille 30), Reference (chaines de caractères de taille 13), Quantite (entier) et Prix\_U (Réal).

Ecrire un programme Pascal qui permet de saisir et d'afficher un Produit.

#### Solution :

```
PROGRAM Stock;
uses wincrt;
Type Produit = RECORD
Designation : String[30];
Reference : String[13];
Quantite : Integer;
Prix_U : Real;
End;
var
P:produit;
begin
with P do
begin
Designation:= 'Ordinateur' ;
Reference := 'HP Z3098' ;
Quantite := 10 ;
Prix_U := 29000.00 ;
end;
with P do
write( 'designation=',designation,' reference', reference,' Quantite', Quantite,' Prix_U ',Prix_U);
end.
```

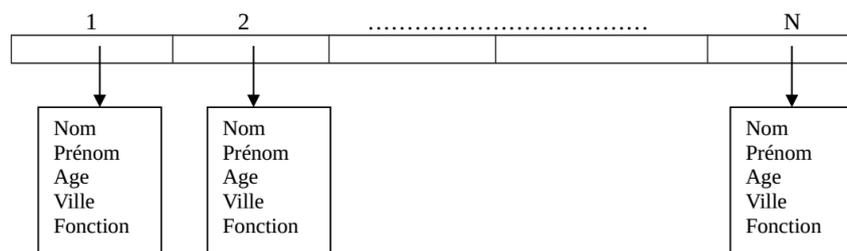
#### *Exemple (Tableau d'enregistrement)*

Considérons une base de données comportant des individus décrits par les informations suivantes :

- Nom
- Prénom
- Age
- Ville
- Fonction

On veut organiser ces informations de telle sorte qu'on puisse sélectionner les individus ayant un certain âge, ou habitant une certaine ville ou ayant une fonction donnée etc... De façon générale gérer la base de données. Proposer une structure de donnée.

La structure de donnée :



a) Écrivons d'abord la séquence d'initialisation de la base de données :

### Exemple 2

Soit un T un tableau d'enregistrement « Individu » contenant les champs suivants : Nom, Prenom (chaines de caractères de taille 20), Age (entier) et Ville, Fonction (chaines de caractères de taille 25).

Ecrire un programme Pascal qui permet de saisir et d'afficher un tableau d'individu.

### Solution

```
PROGRAM LireEcrire;
uses wincrt;
Type Individu = RECORD
Nom, Prenom : string[20] ;
Age : integer ;
Ville, Fonction : string[25] ;
End;
Var
T: array[1..10] of Individu ;
I,N : integer ;
BEGIN
Writeln('Introduire le nombre d'' individus : ');
readln (N); { N est le nombre d'individus }
Writeln('Introduire Nom,Prénom,Age,Ville, Fonction de chaque Individu ');

For I := 1 to N do
begin
Write('Nom: ' );
Readln (T[I].Nom);
Write('prénom: ' );
Readln(T[I].Prenom);
Write('Age: ');
Readln(T[I].Age);
Write('Ville: ');
Readln(T[I].Ville);
Write('Fonction: ' );
Readln(T[I].Fonction);
End;
For I := 1 to N do
begin
Writeln('Nom: ',T[I].Nom);
Writeln('prénom: ',T[I].Prenom);
Writeln('Age: ',T[I].Age);
Writeln('Ville: ',T[I].Ville);
Writeln('Fonction: ',T[I].Fonction);
end;
END.
```

## II- Les Fichiers

Un fichier est une collection de données. PASCAL permet de créer et de manipuler 3 types de fichiers.

- Les Fichiers à 'accès directe' de type **FILE OF**.
- Les Fichiers à accès séquentiel de type **TEXT**
- Les Fichiers 'sans type' de type **FILE**.

Les fichiers de type **FILE OF** sont constitués d'enregistrements de même type, alors que les fichiers **TEXT** admettent des enregistrements de type différents.

Déclaration de fichier en PASCAL :

```
<Nom Fichier> : FILE OF <Type_Enregistrement> ;
```

```
<Nom Fichier> : TEXT ;
```

```
<Nom Fichier> : FILE ;
```

Dans la suite, on s'intéressera uniquement aux fichier de type **FILE OF**.

### II.1 Les fichiers de type FILE OF

**Exemple** : Soit la séquence PASCAL suivante :

```
Type ETUDIANT = RECORD
```

```
Nom : string[20] ;
```

```
Prenom : string[20] ;
```

```
Age : integer ;
```

End ;

Var

F1 : FILE OF ETUDIANT ; F2 : FILE OF Char ;

F3: FILE OF Real;

On a ainsi défini trois fichiers, le premier fichier F1 est constitué d'enregistrement de type ETUDIANT, et le deuxième F2 est constitué de caractères, et le troisième F3 comporte des réels.

## II.2. Les fonctions et procédures sur les fichiers de type FILE OF

### i) *RESET (Fichier)*

Procédure qui ouvre un fichier déjà existant et le prépare pour une lecture ou une écriture. Le pointeur de fichier est positionné sur le premier enregistrement du fichier. (ouverture en lecture/écriture)

### ii) *READ, WRITE, READLN, WRITELN*

Lecture ou écriture sur un fichier de type FILE OF.

#### Exemple

READ(Fichier, Var1, Var2, ..., Varn);

WRITE(Fichier, Var1, Var2, ..., Varn);

Idem pour READLN et WRITELN.

### iii) *SEEK (Fichier, N) :*

Procédure permettant de positionner le pointeur de Fichier sur l'enregistrement Numéro N.

### iv) *FILEPOS(Fichier)*

Fonction qui retourne la position du pointeur du fichier spécifié.

### v) *FILESIZE (Fichier)*

Fonction qui retourne le nombre total d'enregistrements du fichier.

### vi) *REWRITE(Fichier) :*

Procédure qui crée le fichier spécifié s'il n'existait pas. Dans le cas où il existait, il sera détruit. (ouverture du fichier en écriture)

### vii) *CLOSE(Fichier)*

Procédure qui ferme le fichier.

### viii) *EOF(Fichier)*

Fonction qui retourne TRUE si la fin de fichier est rencontrée sinon retourne FALSE.

### ix) *ASSIGN(Fichier, 'Nom\_Physique') :*

Procédure permettant d'associer le nom logique du fichier 'Fichier' au nom physique du fichier.

#### Exemples

##### II.1.1. Exemple 1 : Ouverture et fermeture des fichiers

*ASSIGN(F1, 'CLIENT.DAT') ;*

*REWRITE(F1) ; {création et ouverture du fichier F1 en écriture}*

... ..

*{ initialisation du fichier avec READ/WRITE }*

... ..

*CLOSE(F1) ; {Fermeture du fichier}*

*RESET(F1); {Ouverture du fichier en lecture/écriture}*

... ..

... ..

*{Manipulation du fichier READ,WRITE,SEEK etc... }*

... ..

... ..

*CLOSE(F1) ; {fermeture du fichier}*