

Chapitre 1

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Les enregistrements et les fichiers

1^{ère} Partie : Les enregistrements

1- Définition :

Un enregistrement est une structure de donnée composée, définie par le programmeur, qui permet de grouper (dans une même case mémoire) un nombre fini de données, simples et/ou composées, de même ou de différents types.

On parle, alors, d'un nouveau type non prédéfini appelé **enregistrement** ou **article** susceptible de grouper plusieurs variables de même et/ou de différents types.

2- Notion de champ :

Un enregistrement est un ensemble de valeurs se trouvant chacune dans une variable appelée **CHAMP**. Un champ est une partie de l'enregistrement, identifié par un nom, contenant une valeur de type prédéfini (entier, réel, caractère, chaîne de caractères ou booléen) ou de nouveau type défini par le programmeur.

Schématiquement parlant :

Une variable simple

Elle ne peut contenir qu'une seule valeur de type simple : **entier, réel, caractère, chaîne de caractères ou booléen.**

Par contre un enregistrement est présenté comme suit :



Une variable de type enregistrement groupant n champs

N.B : Un enregistrement n'est pas un tableau !

: 6 : 'G569F'; î

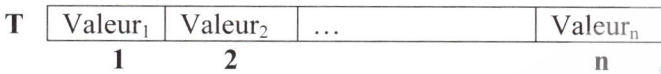
3- Tableau d'enregistrements :

Comme il est mentionné, un enregistrement n'est pas un tableau de valeurs, mais le contraire est possible !

D'après notre pré acquis un tableau est un ensemble de cases de même type, contenant chacune une valeur. Mais, un tableau d'enregistrements est un ensemble de cases de même type, représentant chacune un enregistrement de données.

↳ Schématiquement parlant :

T est un tableau de n éléments simples, représenté comme suit :



Cependant, E est un tableau d'enregistrements dont chaque case (E[i]) est un enregistrement de p Champs :



4- Déclaration d'une variable de type enregistrement :

↳ Au niveau de l'analyse

T.D.O.U

Objet	Nature/Type	Rôle
Nom_variable	Nouveau_Type	Contenir

T.D.N.T

Type
Nouveau_type = Enregistrement
Champ ₁ : Type ₁
....
Champ _p : Type _p
Fin Nouveau_type

↳ En Pascal

TYPE

```
Nouveau_type = RECORD
  Champ1 : Type1 ;
  ....
  Champp : Typep ;
End ;
```

VAR

```
Nom_variable : Nouveau_type ;
```

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

Remarques : Pour cibler un champ d'une variable de type enregistrement, on met le nom de la variable et le nom du champ séparés par un **point**.

5- Traitements simples sur les enregistrements :

a- Lecture d'une valeur dans un champ d'un enregistrement :

- ✓ Algorithme : Lire (Nom_variable.Nom_champ)
- ✓ Pascal : Readln (Nom_variable.Nom_champ) ;

b- Affectation d'une valeur dans un champ d'un enregistrement :

- ✓ Algorithme : Nom_variable.Nom_champ ← Valeur
- ✓ Pascal : Nom_variable.Nom_champ := Valeur ;

c- Affichage d'une valeur d'un champ d'un enregistrement :

- ✓ Algorithme : Ecrire (Nom_variable.Nom du champ)
- ✓ Pascal : Write (Nom_variable.Nom du champ) ;

Remarques :

- ✓ La lecture d'une valeur dans un enregistrement se fait champ par champ.
- ✓ L'écriture des valeurs d'un enregistrement se fait champ par champ.
- ✓ L'affectation des valeurs d'un enregistrement dans une variable de même type peut être réalisée champ par champ ou d'une façon entière.

Exemple : $a \leftarrow b$ avec a et b deux enregistrements de même type.

- ✓ La structure Avec .. Faire (With .. Do en Pascal) nous épargne d'indiquer à chaque fois le nom de la variable de type enregistrement.

2^{ème} partie : Les fichiers

1- Définition :

Un fichier est un ensemble structuré de données, de même ou de différents types, **sauvegardé** sur un support d'informations quasi-permanent (Disquette, Disque dur, Flash disque,...) ce fichier est appelé **fichier physique** représenté toujours au cours des traitements par un fichier de travail appelé **fichier logique**.

2- Types de fichiers :

- ✓ Fichier **Typé** ou fichier **Binaire** : peut contenir tous les types possibles de données,
- ✓ Fichier **Texte** ou fichier **ASCII** : ne contient qu'une succession de lignes de type chaîne de caractères.

☞ **Déclaration d'un fichier Binaire :**

☞ **Au niveau de l'analyse**

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

T.D.O.U

Objet	Nature/Type	Rôle
Nom_Fichier_logique	Nom_Nouveau_type	Contenir

T.D.N.T

Type
Nom_Nouveau_type = Fichier de Nom_Type

☞ **En Pascal**

TYPE Nom_Nouveau_type = **File of Nom_Type** ;

VAR Nom_Fichier_logique : **Nom_Nouveau_type** ;

☞ **Déclaration d'un fichier Texte :**

☞ **Au niveau de l'analyse**

T.D.O.U

Objet	Nature/Type	Rôle
Nom_Fichier_logique	Texte	Contenir

↪ **En Pascal**

VAR Nom_Fichier_logique : TEXT ;

www.BAC.org.tn
Page: BAC-TUNISIE
Tél: 25 361 197 / 53 371 502

3- Types d'accès :

Il existe deux types d'accès au contenu des fichiers :

🔗 Accès Séquentiel :

Dit aussi linéaire, **on ne peut** passer à un élément du fichier **qu'après avoir** passé par ceux qui le précèdent. Ce type d'accès est possible pour les deux types de fichiers (**Binaire ou Texte**).

🔗 Accès Direct :

On peut accéder **directement** à un élément du fichier, **sans** passer par les éléments qui le précèdent. Ce type d'accès n'est possible que pour les fichiers **Binaires**.

4- Notion de pointeur :

Le pointeur est un dispositif logique de lecture/écriture des éléments d'un fichier qui sont numérotés de 0 à un entier représentant (le nombre des éléments -1).

La position du pointeur est, par défaut, sur le 1^{er} élément du fichier et le pas de déplacement du pointeur suite à une opération de lecture ou écriture est, automatiquement, d'un élément du fichier vers celui qui le succède de 0 à 1, de 1 à 2 etc.

Pour les fichiers binaires, il est possible de déplacer le pointeur vers n'importe quel élément du fichier en utilisant des fonctions d'accès direct appropriées.

5- Modules prédéfinis pour la gestion des fichiers Binaires :

a) Association du fichier logique au fichier physique :

Algorithme : **Assigner** (Nom_logique , Nom_Physique)

Pascal : **Assign** (Nom_logique , Nom_Physique) ;

b) Ouverture d'un fichier :

Deux modes d'ouverture peuvent être utilisés à savoir :

↪ Mode **nouveau** fichier :

Algorithme : **Recréer** (Nom_logique)

Pascal : **Rewrite** (Nom_logique) ;

↳ Mode fichier **existant** :

Algorithme : **Ouvrir** (Nom_logique)

Pascal : **Reset** (Nom_logique) ;

c) **Fermeture d'un fichier** :

Algorithme : **Fermer** (Nom_logique)

Pascal : **Close** (Nom_logique) ;

d) **Lecture d'un élément du fichier** :

Algorithme : **Lire** (Nom_logique, Variable_intermédiaire)

Pascal : **Read** (Nom_logique, Variable_intermédiaire) ;

N.B :

✓ La lecture d'un élément du fichier est réalisée par une variable intermédiaire de même type que les éléments du fichier.

✓ Cette opération de lecture engendre systématiquement le déplacement du pointeur vers l'élément successeur.

✓ On ne peut pas utiliser « Readln » avec les éléments d'un fichier binaire (utiliser **Read**).

e) **Ecriture d'un élément dans un fichier** :

Algorithme : **Ecrire** (Nom_logique, Variable_intermédiaire)

Pascal : **Write** (Nom_logique, Variable_intermédiaire) ;

N.B :

✓ L'écriture d'un élément sur le fichier est réalisée par une variable intermédiaire de même type que les éléments du fichier.

✓ Cette opération d'écriture engendre systématiquement le déplacement du pointeur vers l'élément successeur.

✓ On ne peut pas utiliser « Writeln » avec les éléments d'un fichier Binaire (utiliser **Write**).

f) **Indicateur de la position actuelle du pointeur**

Algorithme : **Position_fichier** (Nom logique)

Pascal : **FilePos** (Nom_logique) ;

g) **Indicateur de la fin du fichier**

Algorithme : **FF** (Nom_logique)

Pascal : **EOF** (Nom_logique) ;



6- Modules d'accès direct aux éléments d'un fichier binaire :

a) Pointer sur un élément du fichier :

Algorithme : **Pointer** (Nom_logique, Numéro_élément)

Pascal : **Seek** (Nom_logique, Numéro_élément) ;

N.B :

✓ Il ne faut pas oublier que les éléments d'un fichier binaire sont numérotés de 0 à un entier représentant (le nombre des éléments -1).

b) Déterminer la taille d'un fichier (nombre d'éléments):

Algorithme : **Taille_fichier** (Nom_logique)

Pascal : **FileSize** (Nom_logique) ;

N.B :

✓ C'est une fonction retournant le nombre d'éléments dans un fichier qui est égal au numéro du dernier élément du fichier qui joue le rôle de l'indicateur de la fin du fichier (EOF).

7- Modules prédéfinis pour la gestion des fichiers Textes :

Les modules : Assigner, Recréer et Fermer utilisés avec les fichiers typés ont le même rôle avec les fichiers textes.

a) Ouverture d'un fichier :

Trois modes d'ouverture peuvent être utilisés à savoir :

↳ Mode nouveau fichier :

Algorithme : **Recréer** (Nom_logique)

Pascal : **Rewrite** (Nom_logique) ;

N.B : Ce mode d'ouverture a le même rôle et effets que celui utilisé avec les fichiers binaires.

↳ Mode fichier **existant** :

Algorithme : **Ouvrir** (Nom_logique)

Pascal : **Reset** (Nom_logique) ;

N.B : Ce mode d'ouverture est différent de celui utilisé avec les fichiers binaires, celui-ci ne permet pas l'écriture sur le fichier.

↳ Mode **ajout de lignes** :

Algorithme : **Ajouter** (Nom_logique)

Pascal : **Append** (Nom_logique) ;

N.B :

- ✓ Le pointeur pointe, par défaut, sur la dernière ligne du fichier,
- ✓ Ce mode permet l'écriture des lignes sur le fichier.

b) Lecture d'un élément du fichier :

Algorithme : **Lire_nl** (Nom_logique, Variable_intermédiaire)

Pascal : **Readln** (Nom_logique, Variable_intermédiaire) ;

N.B :

- ✓ La lecture d'une ligne du fichier est réalisée par une variable intermédiaire de type chaîne de caractères.
- ✓ Cette opération de lecture engendre systématiquement le déplacement du pointeur vers la ligne suivante.
- ✓ On ne peut pas utiliser « Read » avec les lignes d'un fichier texte (utiliser **Readln**).

c) Ecriture d'un élément dans un fichier :

Algorithme : **Ecire_nl** (Nom_logique, Variable_intermédiaire)

Pascal : **Writeln** (Nom_logique, Variable_intermédiaire) ;

N.B :

- ✓ L'écriture d'une ligne sur le fichier est réalisée par une variable intermédiaire de type chaîne de caractères.
- ✓ Cette opération d'écriture engendre systématiquement le déplacement du pointeur vers la ligne suivante.
- ✓ On ne peut pas utiliser « Write » avec les lignes d'un fichier texte (utiliser **Writeln**).

d) Indicateur de la fin d'une ligne :

Algorithme : **Fin_ligne** (Nom_logique)

Pascal : **EOLN** (Nom_logique) ;

e) Accès à la fin d'une ligne :

Algorithme : **Chercher_Fin_ligne** (Nom_logique)

Pascal : **SeekEoln** (Nom_logique) ;

f) Accès à la fin du fichier :

Algorithme : **Chercher_Fin_fichier** (Nom_logique)

Pascal : **SeekEof** (Nom_logique) ;

www.BAC.org.tn
Page BAC-TUNISIE
Tél: 25 361 197 / 53 371 502