Université A. Mira de Béjaia
Faculté des Sciences Exactes
Département d'Informatique

Chapitre 7 : Les fichiers

Contenu du chapitre :

1. Définition d'un fichier.
2. Fichier typé.
3. Exemples illustratifs d'utilisation d'un fichier typé
4. Fichier texte.
5. Exercice

1. Définition

Un fichier est une collection de données de même type stockée en général sur un support externe (disquette, disque dur, disque optique, flash disque, etc.).

Les fichiers permettent de stocker des informations de manière permanente. Un fichier peut être :

- **Un fichier typé** : déclaré avec les mots clés « **FICHIER DE** ».
- **Un fichier texte :** déclaré avec le mot clé « **TEXTE** ».

2. Fichier typé

Utilisé pour stocker des données structurées (enregistrements, tableaux, etc.). Généralement, les données sont stockées dans le fichier sous le format binaire.

Syntaxe de déclaration :

```
TYPE  Nom_type_enregistrement = ENREGISTREMENT
      Champ1 : Type1 ;
      Champ2 : Type2 ;
      …
      ChampN : TypeN ;
      FIN ;
VAR nom_fichier : FICHIER DE Nom_type_enregistrement ;
```

Exemple : Déclaration d'un fichier qui contient des élèves.

```
TYPE eleve = ENREGISTREMENT
       Nom : Chaine de caractères ;
       Age : Entier ;
       FIN ;
VAR Fichier_eleves : FICHIER DE eleve ;
```

Nom logique vs. Nom physique :

- **Nom logique :** c'est le nom de fichier utilisé dans un programme. **Exemple :** Fichier_eleves.
- **Nom physique :** c'est le nom de fichier stocké sur le disque. **Exemple :** "D:\TP\MonFichier.dat".

2.1. Opération d'association (assignation)

C’est une opération qui consiste à associer un nom logique d’un fichier à son nom physique. Elle se fait en utilisant le mot clé *« ASSOCIER »*.

Syntaxe de déclaration :
ASSOCIER (Nom_logique, Nom_Physique) ;

Exemple : Associer le nom logique *« Fichier_eleves »* au nom physique « D:\TP\MonFichier.dat ».
ASSOCIER (Fichier_eleves, "D:\TP\MonFichier.dat") ;

2.2. Ouverture d’un fichier

L’ouverture d’un fichier se fait soit en écriture, soit en lecture/écriture.

2.2.1. L’instruction « REECRIRE »

Cette instruction permet d'ouvrir un fichier **en écriture**, c’est-à-dire de créer le fichier, et d'autoriser des opérations d'écriture dans ce dernier. Si le fichier existe déjà, son contenu sera effacé.

Syntaxe de déclaration :
REECRIRE (Nom_logique) ;

2.2.2. L’instruction « RELIRE »

Cette instruction permet de renvoyer le pointeur au début du fichier pour pouvoir **lire** ou **écrire** à partir du début du fichier.

Syntaxe de déclaration :
RELIRE (Nom_logique) ;

2.3. Ecriture dans un fichier

L’instruction « *ECRIRE* » permet d’écrire ou de modifier une valeur ou un enregistrement dans un fichier.

Syntaxe de déclaration :
ECRIRE (Nom_logique, variable) ;

Exemple : Ecrire l’enregistrement « Enreg » dans le fichier « Fichier_eleves »
ECRIRE (Fichier_eleves, Enreg) ;

2.4. Lecture à partir d’un fichier

L’instruction « **LIRE** » permet de lire une valeur (ou un enregistrement) à partir d’un fichier et de la (le) mettre dans une variable.

Syntaxe de déclaration :
LIRE (Nom_logique, variable) ;

Exemple : Lire un enregistrement à partir de « Fichier_eleves » et le mettre dans la variable « Enreg ».
LIRE (Fichier_eleves, Enreg) ;

2.5. Fermeture d'un fichier

L'instruction « FERMER » permet de fermer un fichier.

Syntaxe de déclaration :
FERMER (Nom_logique) ;

Exemple : Fermer le fichier « Fichier_eleves».
FERMER (Fichier_eleves) ;

2.6. Test de fin de fichier

L'instruction « *FDF* » permet de savoir si on a atteint la fin d'un fichier ou non. C'est une fonction qui retourne un booléen. En fait, elle retourne VRAI si on a atteint la fin du fichier, sinon elle retourne FAUX.

Syntaxe de déclaration :
FDF (Nom_logique) ;

Exemple : Vérifier si on a atteint la fin du fichier « Fichier_eleves ».
FDF (Fichier_eleves) ;

3. Exemples illustratifs d'utilisation d'un fichier typé

3.1 Ecriture dans un fichier typé

Algorithme qui permet de créer un fichier « Fichier_eleves » avec les champs : nom et âge de l'élève.

```
ALGORITHME Fichier_ecriture ;
TYPE eleve = ENREGISTREMENT
        Nom : Chaine de caractères ;
        Age : Entier ;
        FIN ;
VAR
   Fichier_eleves : FICHIER DE eleve ;
   Enreg : eleve ;
   Choix : Entier ;

DEBUT
   {Création du fichier « Fichier_éléves » : ouverture en écriture}
   ASSOCIER (Fichier_eleves, "D:\TP\MonFichier.dat") ;
   REECRIRE (Fichier_eleves) ;
   REPETER
      ECRIRE ('Nom de l'élève ? ») ;
```

```
        LIRE (Enreg.Nom) ;
        ECRIRE ('Age de l'élève ? ») ;
        LIRE (Enreg.Age) ;
        {On écrit sur le fichier}
        ECRIRE (Fichier_eleves, Enreg) ;
        ECRIRE ('Autre saisie ? 1. OUI 2.NON') ;
        LIRE (Choix) ;
    JUSQU'A (Choix = 2) ;
   FERMER (Fichier_eleves) ;
FIN.
```

3.2 Lecture à partir d'un fichier typé

Algorithme qui permet de consulter le fichier « Fichier_eleves » crée précédemment.

```
ALGORITHME Fichier_lecture ;
TYPE eleve = ENREGISTREMENT
          Nom : Chaine de caractères ;
          Age : Entier ;
          FIN ;
VAR
    Fichier_eleves : FICHIER DE eleve ;
    Enreg : eleve ;

DEBUT
    {Consulation du fichier « Fichier_éléves » : ouverture en lecture}
    ASSOCIER (Fichier_eleves, "D:\TP\MonFichier.dat") ;
    RELIRE (Fichier_eleves) ;
   TANT QUE (NON FDF (Fichier_eleves)) FAIRE
        LIRE (Fichier_eleves, Enreg) ; {On lit dans le fichier}
        ECRIRE (Enreg.Nom, '      ', Enreg.Age) ;
    FIN TANT QUE;
   FERMER (Fichier_eleves) ;
FIN.
```

4. Fichier texte

C'est un cas particulier des fichiers. Un fichier texte est formé par des caractères (texte) qui sont organisés en lignes. Chaque ligne se termine par une marque de fin de ligne (caractère 10 du code ASCII), et le fichier lui-même se termine par une marque de fin de fichier (caractère 13 du code ASCII). Les deux marques de fin sont transparentes au programmeur.

Syntaxe de déclaration :

VAR Nom_fichier : **TEXTE** ;

4.6. Manipulation d'un fichier texte

En lecture comme en écriture, la manipulation d'un fichier texte se passe très naturellement, de la même façon que la lecture au clavier ou l'écriture à l'écran. En fait, le clavier et l'écran sont tout simplement considérés comme des fichiers texte.

4.7. Exemple illustratif d'utilisation d'un fichier texte

```
ALGORITHME Lecture_texte ;
VAR
        F : TEXTE ;
        S : Chaine de caractères ;
DEBUT
  ASSOCIER (F, "D:\TP\MonFichier.txt") ;
  RELIRE (F) ;
  TANT QUE (NON FDF(F)) FAIRE
     LIRE (F, S);    {Lire une ligne complète dans « F »}
     ECRIRE (S) ;  {Ecrire « S » sur l'écran}
  FIN TANT QUE ;
  FERMER (F) ;
FIN.
```

De la même manière, on peut lire et/ou écrire des caractères, entiers, réels, etc.

Remarque :

Les valeurs numériques sont stockées dans le fichier texte comme des chaines de caractères. La conversion nombre/chaîne de caractères est faite automatiquement par le compilateur. Par contre, si le type d'une variable ne correspond pas avec la donnée lue dans le fichier et qu'aucune conversion n'est possible alors cela produit une erreur.

5. Exercice

Ecrire un algorithme qui permet de :

A. Créer un fichier « étudiants » avec les champs suivants : *numéro d'identification de l'étudiant*, *nom de l'étudi*ant, et *moyenne de l'étudiant*.

   La création du fichier doit se faire dans une procédure.

B. Consulter la moyenne d'un étudiant donnée avec son numéro d'identification. Ceci doit être réalisé par une fonction.

   L'algorithme demande d'entrer le numéro d'identification de l'étudiant recherché, le passe comme paramètre à une fonction, celle-ci retourne la moyenne de l'étudiant au programme principal qui l'affiche.

Orientation et remédiation :

- *Pour la résolution de l'exercice l'étudiant à besoin des notions du chapitre 6.*
- *Si l'étudiant n'arrive pas à faire la partie « A » de l'exercice. Il est conseillé de refaire la lecture et analyse des Sections 1 et 2. Plus exactement, les Sous Sections 2.1, 2.2, et 2.3.*

- *Si l'étudiant n'arrive pas à faire la partie « B » de l'exercice. Il est conseillé de refaire la lecture et analyse des Sous Sections 2.4 et 2.5.*
- *Si l'étudiant a pu terminer l'exercice, il est conseillé de passer au chapitre suivant.*