

B - Les fichiers

I. Introduction

A la fin de l'exécution d'un programme, toutes les informations saisies et stockées au niveau de la mémoire centrale seront perdues. Les langages de programmation nous permettent de stocker les données saisies dans **des fichiers** enregistrés sur des supports de stockage.

1. Définition

Un fichier est un ensemble structuré de données de même type nommé et enregistré sur une mémoire de masse.

2. Contenu des fichiers

- Les fichiers de données (binaires typés) : C'est un fichier qui est basé sur une structure de donnée.
- Les fichiers texte : C'est un ensemble de lignes composées de caractères.
- Les fichiers binaires non typés.

3. Types d'accès

- Accès séquentiel
- Accès direct

II. Les fichiers de données

1. Déclaration

En pascal

```
Type nom_fichier = File of type_composants;
```

```
Var nom_logique : nom_fichier;
```

2. Association

- Nom interne (logique) : Nom du fichier par rapport au programme.
- Nom externe (physique) : Nom du fichier par rapport au système d'exploitation.

En analyse

```
Associer (nom_logique, nom_physique)
```

En pascal

```
ASSIGN (nom_logique, nom_physique);
```

3. Ouverture

- *Ouverture et création :*

En analyse

Recréer (nom_logique)

En pascal

ReWrite (nom_logique);

- *Ouverture et remise à zéro :*

En analyse

Ouvrir (nom_logique)

En pascal

ReSet (nom_logique);

4. Ecrire dans un fichier

En analyse

Écrire (nom_logique, objet)

En pascal

Write (nom_logique, objet);

5. Lire à partir d'un fichier

En analyse

Lire (nom_logique, objet)

En pascal

Read (nom_logique, objet);

6. Fermeture

Certains changements sur le fichier physique ne sont fait qu'après sa fermeture.

En analyse

Fermer (nom_logique)

En pascal

Close (nom_logique);

7. Fonction position_fichier

En analyse

Position_fichier (nom_logique)

En pascal

FilePos (nom_logique);

8. Fonction Taille du fichier (nombre d'éléments du fichier)

En analyse

Taille_fichier (nom_logique)

En pascal

Filesize (nom_logique);

9. Procédure Tronquer

Consiste à faire une suppression physique des données situées à partir du pointeur.

En analyse

Tronquer (nom_logique)

En pascal

Truncate (nom_logique);

10. Procédure Effacer (Supprimer le fichier physique du disque)

En analyse

Effacer (nom_logique)

En pascal

Erase (nom_logique);

11. Procédure Renommer (Supprimer le fichier physique du disque)

En analyse

Renommer (nom_logique, nouveau_nom_physique)

En pascal

Rename (nom_logique, nouveau_nom_physique);

12. Directive de compilation {\$I-}

Tous les appels de procédures et de fonctions d'entrées-sorties font automatiquement l'objet d'un contrôle d'erreur. Si une erreur survient, le programme se termine en affichant un message d'erreur à l'exécution (Run-time error). Ce contrôle automatique peut être désactivé et réactivé en utilisant les directives de compilation {\$I-} et {\$I+}.

Lorsque le contrôle d'entrées-sorties est désactivé, c'est-à-dire lorsqu'une série d'instructions est précédée de {\$I-}, une erreur d'entrée-sortie ne cause pas l'arrêt du programme, mais suspend toute autre opération d'entrée-sortie jusqu'à ce que la fonction standard **IOResult** soit appelée. Cette particularité permet de tenter sans risque plusieurs opérations disques en séquence et de ne tester qu'une fois à la fin si tout s'est bien passé.

III. Accès séquentiel

1. Test de fin du fichier

En analyse

Fin_fichier (nom_logique)

En pascal

EOF (nom_logique);

Activité 4 :

On désire créer un fichier de données à accès à accès séquentiel pour stocker la liste des élèves d'une classe. Créer une suite de programme en Pascal pour faire les trois opérations suivantes :

- Création et écriture dans le fichier des élèves.
- Lecture à partir du fichier du premier élève.
- Ajouter des élèves au fichier.

Nom physique et emplacement : C:\fich_el.dat

Pascal (voir fichier : f_crea.pas)

Pascal (voir fichier : f_read.pas)

Pascal (voir fichier : f_ajout.pas)

Activité 5 : (Suite de l'activité 1)

Créer un programme qui saisi autant de produit que l'utilisateur désire. Ces enregistrements seront stockés dans un fichier nommé "fp.dat" situé sur la racine C:\. (Accès séquentiel).
Afficher le nombre des produits dont le PU est < 10,000 DT. (Découpage modulaire)

a. Analyse principale :

Résultat : Afficher la liste des produits dont PU < 10,000 DT

Traitement :

Saisie des produits.

Création du fichier.

Algorithme

- 0) Début produit_pu
- 1) Proc création (fp1)
- 2) Proc saisie (fp1)
- 3) Ecrire (FN nbr_prod (fp1))
- 4) Fermer (fp1)
- 5) Fin produit_pu

T.D.N.O

Type
Produit = Enregistrement
Référence : Chaîne [10]
Nom : Chaîne [50]
Prix_unitaire : Réel
Quantité : Entier
Exportation : octet
f_prod = fichier de produit

T.D.O. Globaux

Objet	Type	Rôle
fp1	f_prod	Fichier de produit
Création	Procédure	Assignment et ouverture en écriture
Saisie	Procédure	Saisie des produits
nbr_prod	Fonction	Teste et affichage

b. Analyse de la procédure création :

Résultat : Créer et ouvrir le fichier fp1

Traitements :

Associer
Recréer

Algorithme

- 0) Procédure **création** (var fp1 : f_prod)
- 1) Associer (fp1, "C:\fp.dat")
- 2) Recréer (fp1)
- 3) Fin **création**

c. Analyse de la procédure saisie :

Résultat : Produits saisis et enregistrés dans fp1

Traitements :

Une boucle répéter .. Jusqu'à est utilisée pour saisir autant de produits que l'utilisateur désire. L'arrêt se fait lorsque l'utilisateur saisie "N" (Non).

Algorithme

- 0) Procédure **saisie** (var fp1 : f_prod)
- 1) **Répéter**

Avec **p** faire
Lire (Référence)
Lire (Nom)
Lire (Prix_unitaire)
Lire (Quantité)
Lire (Exportation)

Fin Avec

Ecrire (fp1, p)

Ecrire ("Voulez vous saisir un nouveau produit (O / N) : "), Lire (rep)

Jusqu'à rep = "N" ou rep = "n"

Fin **remplir**

T.D.O. Locaux

Objet	Type	Rôle
rep	Caractère	Réponse de l'utilisateur (O / N)
P	Produit	Sert pour la saisie d'un produit

d. Analyse de la fonction affiche :

Résultat : nbr_prod.

Traitements :

```

nbr_prod ← cpt
Ouverture du fichier et remise à zéro du pointeur.
Initialisation de cpt à 0.
Lire le produit
Tester le PU
Incrémenter cpt si le teste est vrai

```

Algorithme

- 0) Fonction **nbr_prod** (var fp1 : f_prod) : octet
- 1) ouvrir (fp1)
- 2) cpt ← 0
- 3) Tant que non (Fin_fichier (fp1)) faire
 - Lire (fp1, p)
 - Si p.pu <10 alors cpt ← cpt + 1
 - Fin si
- Fin tant que
- 4) nbr_prod ← cpt
- 5) Fin **nbr_prod**

T.D.O. Locaux

Objet	Type	Rôle
P	Produit	
cpt	Octet	Nombre de produit

e. Pascal (voir fichier : prod_3_1.pas)

IV. Accès direct

1. Pointeur d'enregistrement

En analyse

Pointer (nom_logique)

En pascal

Seek (nom_logique, numéro);

Activité 6 : (Suite de l'activité 4)

Créer un programme qui saisi un numéro d'ordre d'un élève parmi la liste des élèves créée dans l'activité 4. Le programme doit tester la validité de ce nombre [0.. n], puis afficher le nom de l'élève qui lui correspond.

a. Analyse principale :

Résultat : Affichage du nom de l'élève en question.

Traitement :

Ouvrir le fichier en lecture

Saisie du numéro d'ordre.

Algorithme

- 0) Début numero_eleve
- 1) Proc ouvrir (elevel)
- 2) Proc saisie (num)
- 3) Proc recherche (elevel, num)
- 4) Fin numero_eleve

T.D.N.O

Type
f_eleve = fichier de chaîne

T.D.O. Globaux

Objet	Type	Rôle
elevel	f_eleve	Fichier des noms des élèves
num	octet	Numéro de l'élève recherché
Ouvrir	Procédure	Assignation et ouverture
Saisie	Procédure	Saisie du numéro
Recherche	Procédure	recherche et affichage du nom

b. Analyse de la procédure ouvrir :

Résultat : Ouvrir et repositionner le pointeur du fichier elevel

Traitements :

Associer

Test de l'existence du fichier puis Ouverture.

Algorithme

- 0) Procédure **ouvrir** (var elevel : f_eleve)
- 1) Associer (elevel, "C:\fich_el.dat")
- 2) {\$I-}
Ouvrir (elevel)
Si IOResult < > 0 alors Recréer (elevel)
Fin Si
{\$I+}
- 3) Fin **ouvrir**

c. Analyse de la procédure saisie :

Résultat : numéro saisi

Traitements :

Saisie contrôlée de num qui doit être encadré entre 0 et le nombre d'éléments du fichier.

Algorithme

0) Procédure **saisie** (var num: octet)

1) **Répéter**

 Lire (num)

Jusqu'à num dans [0.. taille_fichier (elevel) – 1]

2) Fin **saisie**

d. Analyse de la procédure rechercher :

Résultat : Afficher l'élève.

Traitements : Pointer l'élève

 Lire le nom pointé.

Algorithme

0) Procédure **recherche** (var elevel : f_eleve, num: octet)

1) Pointer (elevel, num)

2) Lire (elevel, ch)

3) Ecrire ("l'élève numéro ", num, " est : ", ch)

4) Fin **recherche**

T.D.O. Locaux

Objet	Type	Rôle
ch	chaîne	

e. Pascal (voir fichier : f_direct.pas)

Activité 7 : (Suite de l'activité 4)

Créer un programme qui saisi un numéro d'ordre d'un élève parmi la liste des élèves créée dans l'activité 4. Le programme doit tester la validité de ce nombre [0.. n], puis supprimer l'élève en question du fichier. (Accès direct)

Pascal (voir fichier : f_suppr.pas)

Pascal (voir fichier : f_rename.pas)

Pascal (voir fichier : f_erase.pas)

I. Les fichiers texte

1. Déclaration

En analyse

Objet	Type	Rôle
nom_logique	Texte	Fichier de type texte

En pascal

Var nom_logique : **Text**;

2. Procédures et Fonction prédéfinies

- Associer..... **assign**
- Recréer **Rewrite**
- Ouvrir **Reset**
- Ajout..... **Append**
- Lire / Lire_nl **read / readln**
- Ecrire / écrire_nl..... **write / writeln**
- Fin_fichier**EOF**
- Chercher_fin_fichier**SeekEOF**
- Fin_ligne..... **EoLn**
- Chercher_fin_ligne..... **SeekEoLn**
- Effacer **Erase**
- Renommer **Rename**
- Fermer **Close**

Activité 8 :

Créer un programme nommé fichier_texte qui effectue les tâches suivantes :

- Création et saisie d'un fichier texte enregistré sur C:\ sous le nom "FichierOriginal.txt".
- Copier le contenu de ce fichier dans un deuxième nommé "FichierCopie.txt" avec numérotation des lignes. (Modulaire)

Pascal (voir fichier : copie.pas)

Exercice révision

Une société désire créer un fichier pour gérer les crédits accordés à ses employés. Un crédit est constitué par :

- Un numéro
- Une date : La date est constituée de trois champs (jour, mois et année). Par défaut, lors de la création d'un crédit, la date est insérée automatiquement dans l'enregistrement à partir du système. (Procédure Getdate(), bibliothèque windos)
- Code employé
- Montant
- Durée
- Taux d'intérêt : Doit être compris entre 5% et 12 %.

T.A.F

- 1) Les données sont stockées sur le disque dur c:\, le nom du fichier physique est : credit.dat.
- 2) Les différents modes d'accès sont acceptés.
- 3) Créer un module permettant de remplir le fichier.
- 4) Créer un module qui permet de d'effectuer la mise à jour suivante :
 - Si la durée est comprise entre 8 et 10 alors le taux est incrémenté de 2. (+ 2%)
 - Si la durée est comprise entre 6 et 8 alors le taux est décrétementé de 1. (- 1 %)

Pascal (voir fichier : credit.pas)