

Travaux Pratiques VARI1 Séance 3

Thèmes : Les fichiers séquentiels, les exceptions.

On pourra se reporter au cours de D.Enselme Fichiers sur la page VARI.

Généralités

Un fichier est une collection de données dont le nombre est à priori inconnu.

Un fichier est conservé en mémoire secondaire (disque par exemple), il est désigné par un nom : chaîne de caractères et est extérieur au programme. On parlera de fichier externe.

Lorsqu'un programme veut manipuler les données d'un fichier, il doit déclarer une variable de type fichier et l'associer à un nom de fichier externe. Ce sont les opérations de création et d'ouverture de fichier.

Nous nous intéresserons par la suite uniquement aux fichiers binaires à accès séquentiel.

Les fichiers binaires

Un fichier binaire contient du code binaire représentant chaque élément, un tel fichier ne peut être manipulé que par un programme.

L'accès séquentiel à un élément se fait en parcourant le fichier jusqu'à trouver l'élément recherché, l'écriture d'un nouvel élément se fait après le dernier élément.

Le traitement d'un fichier binaire à accès séquentiel s'effectue au moyen du paquetage `Ada.Sequential_IO`.

Création d'un fichier de comptes bancaires

1. Définir le type des éléments du fichier

```
Subtype banque is integer range 100..500 ;
Subtype numero is integer range 10000..30000 ;
Type Entete is record
  Ba : banque ;
  Num : numero ;
  Solde : float :=0.0 ;
End record ;
Type compte is record
  E :entete ;
  Fl : string(1..11) ;
  --Fl nom du fichier de lignes comptables associé au compte
End record ;
```
2. Créer le paquetage `Ada.Sequential_IO` pour le type compte :

```
Package Compt_Seq_Io is new Ada.Sequential_IO(Compte) ;
Use Compt_Seq_Io ;
```
3. Déclarer une variable fichier dont les éléments sont de type compte

```
F_compte : Compt_Seq_IO.File_Type ;
```
4. créer le fichier de nom « comptes.dat »

```
create(File=>F_Compte, name=> « comptes.dat ») ;
```

Ouverture et fermeture d'un fichier binaire

L'ouverture d'un fichier permet de faire le lien entre le fichier externe et la variable fichier, l'ouverture précise également le mode d'utilisation du fichier: lecture (read), écriture(write), adjonction(append).

```
Open(File =>F_Compte,mode =>out_file, name=> « comptes.dat »)
```

Ouverture du fichier en écriture, positionne sur le premier élément du fichier.

Le mode `Append_file` positionne en fin de fichier et permet d'ajouter un élément au fichier

Le mode `In_file` positionne sur le premier élément et permet la lecture

La fermeture du fichier supprime l'association variable/fichier externe

```
Close(File=>F_Compte)
```

Lecture et écriture des éléments d'un fichier

Soit C une variable de type `Compte`

La lecture d'un élément de `F_Compte` se fait par :

```
Read(File=>F_Compte, Item=> C) ;
```

L'écriture par :

```
Write(File=>F_Compte,Item=>C) ;
```

Autres fonctions

```
Function End_of_file(file : in File_type) return boolean ;
```

Le fichier doit être ouvert en lecture , la fonction retourne true si la fin de fichier est atteinte, false sinon.

Questions

- 1- Écrire une procédure qui crée un fichier de comptes
- 2- Écrire une procédure qui crée un nouveau compte dans le fichier de compte
- 3- Écrire une procédure qui recherche et lit un compte identifié par le code banque et le numero de compte
- 4- Écrire un programme qui crée un fichier de comptes, crée 3 comptes, recherche un compte donné et affiche ce compte