Ministère de l'enseignement supérieur et de la recherche scientifique
Université Djillali BOUNAAMA - Khemis Miliana (UDBKM)
Faculté des Sciences et de la Technologie
Département de Mathématiques et d'Informatique

Chapitre 3

Les enregistrements et les Fichiers

MI-L1-UEF221 : Algorithmiques et Structures de Données II

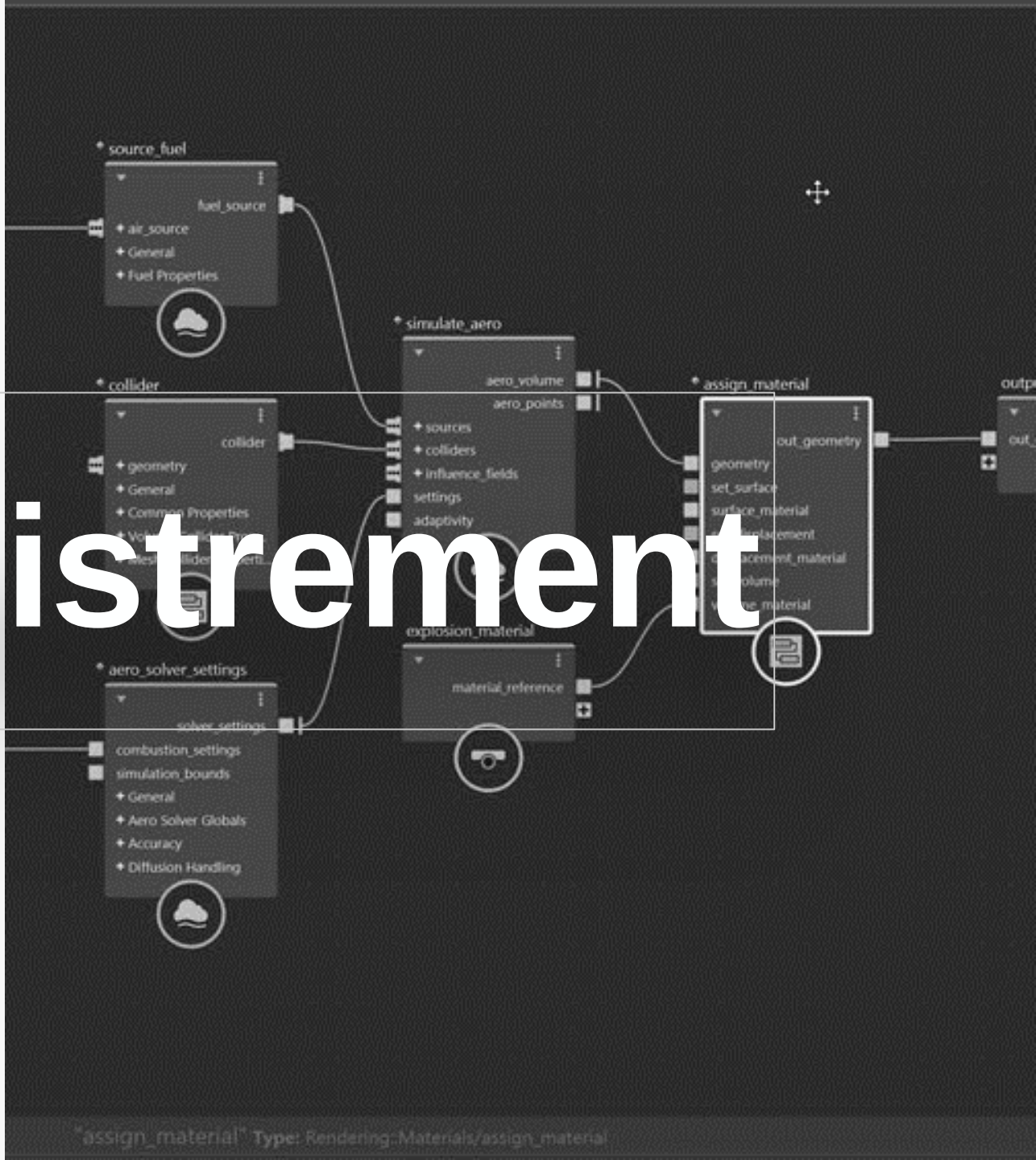
Noureddine AZZOUZA

n.azzouza@univ-dbkm.dz

Plan du Cours

1. Enregistrements
    - 1.1 Définition
    - 1.2 Déclaration
    - 1.3 Manipulation
2. Les Fichiers
    - 1.1 Définition
    - 1.2 Fichiers de données et fichiers textes
    - 1.3 Operations sur les fichiers
    - 1.3 Accès séquentiel et accès direct

Les Enregistrement



Définition

- ✓ Un enregistrement est un ensemble de données pouvant être de types différents, appelés **Champs** et regroupés sous un même nom **d'objet** défini par l'utilisateur.

- ✓ Un **enregistrement** est un **type** de données décrivant une information composite, constituée de **plusieurs valeurs** qui peuvent être elles-mêmes de **types différents**

Définition

Un Enregistrement

Champ1	Champ2	Champ3	Champ4	Champ5
Type1	Type2	Type3	Type4	Type5

❑ ***Exemples***

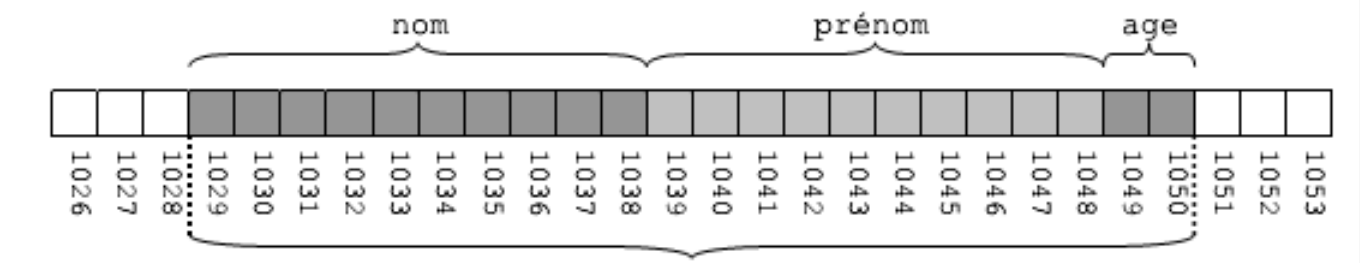
Personne

nom	prenom	age
Chaîne	Chaîne	Entier

Produit

Référence	Désignation	Quantité	Prix d'Achat	Prix de vente
Chaîne	Chaîne	Entier	Réel	Réel

❑ ***Représentation mémoire***



Déclaration

✓ La déclaration d'un enregistrement d'effectue en précisant le mot clé « **enregistrement** », suivi de la description des champs et terminé par le mot clé « **Fin** ».

```
Type    nom_type = Enregistrement
                   Champ1 : type 1
                   ...
                   Champ n : type n
          Fin nom_type
```

➢ Les types des champs peuvent être ***prédéfinis*** ou ***définis par l'utilisateur***

➢ Un champ a exactement les mêmes propriétés qu'une variable de même type

➢ Le champ d'une variable enregistrement peut être lui même un enregistrement

Déclaration : *Exemples*

```
Type date = enregistrement;
        jour: entier;
        mois: entier;
        annee: entier;
    Fin

Var date_entree: date;
```

```
Type personne = enregistrement;
        nom : Chaîne [20];
        prenom : Chaîne [20];
        date_naissance: date ;
    Fin

Var P1, P2 : personne;
```

```
Type produit = enregistrement;
        ref: Chaîne [10];
        design: Chaîne [20];
        qte: entier;
        PA, PV: réel;
    Fin

Var  prod: produit;
     T: Tableau[1..100] de produit;
```

Déclaration : *Implémentation*

PASCAL

```
Type date = record
        jour: integer;
        mois: integer;
        annee: integer;
        end;

    personne = record
        nom: String;
        prenom: String;
        date_naissance: date;
        end;

    produit = record
        ref: String[10];
        design: String[25];
        qte: integer;
        PA, PV: Real;
        end;
```

C

```
struct date
{   int jour;
    int mois;
    int annee;
};

struct
{   char nom[10];
    char prenom[10];
    struct date date_naissance;
};

struct produit
{   char ref[10];
    char design[25];
    int qte;
    float PA, PV;
};
```

Manipulation

✓ L'accès à un champs se fait en précisant le **nom de l'enregistrement** suivi du **nom du champs séparé** par un **point**.

❑ ***Lecture à partir du champ***

valeur ← **Objet.champ**;

```
m ← date_entree.mois;
nom2 ← P2.nom;
annee1 ← P1. date_naissance.annee;
Ecrire('le prix de vente=', prod.PV);
pr1 ← T[i].design;
```

❑ ***Ecriture sur le champ***

Objet.champ ← *valeur;*

```
date_entree.jour ← jj;
Lire (P2.prenom);
P1. date_naissance.annee = 1992;
Prod.qte ← 1500 – qte_vente;
T[12].PV ← 652.50;
```

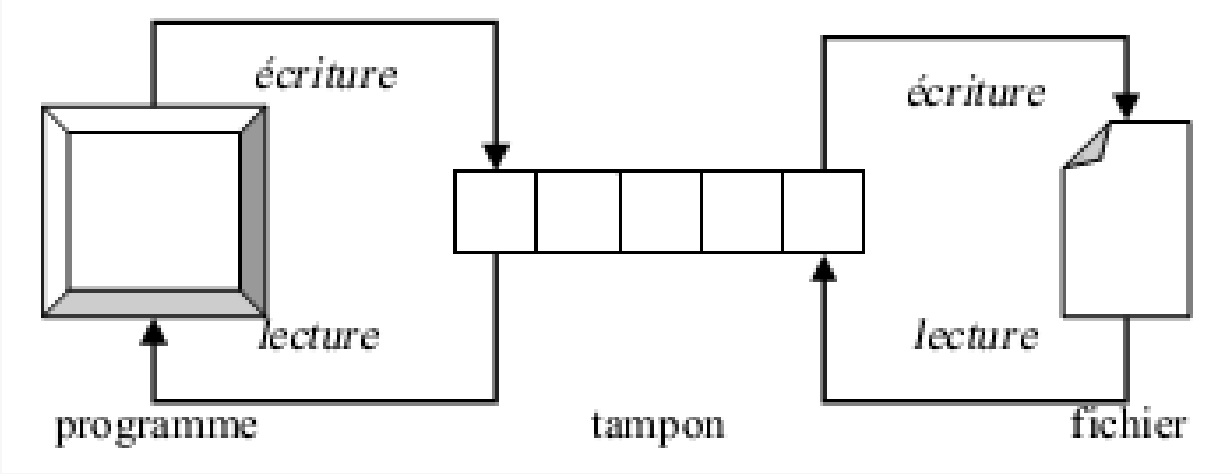
Les Fichiers

Définition

- ✓ Un **fichier** est ensemble de données associées à un **nom** et un **emplacement** sur un **support de stockage** (disque dur, clé USB, disque optique, etc.).
  - ➢ un fichier n'est pas structuré par nature, il est vu comme une séquence d'octets.
  - ➢ Les informations que l'on veut stocker peuvent prendre deux formes :
    1. des données sous forme de composants ou articles
    2. du texte.

Accès au fichiers

- ✓ L'accès aux fichiers est bufférisé, c'est-à-dire qu'on utilise un tampon (buffer) lors des écritures et des lectures.
- ✓ tampon : zone de mémoire utilisée pour y manipuler des données de façon temporaire, avant de les placer ailleurs.



Remarques

1. Comme on a déjà dit un fichier doit être enregistré sur un support externe, donc ce fichier doit avoir un nom et de préférence une extension. Ce nom est appelé le nom externe (ou le **nom physique**)

✓ 2. Le nom de l'objet déclaré dans le tableau des objets comme nom de fichier est le nom interne du fichier (ou aussi le **nom logique**). C'est le nom utilisé dans les instructions du programme.

✓ 3. Les fichiers de données (data) ont une extension **.dat, .fch**

Définition

✓ C'est un ensemble de composants ou d'articles traitant du même sujet et regroupés sous un même nom.

Fichier F

Composant1	Composant2	.....	Composant n	FF

Fin de fichier

Déclaration

- ✓ La déclaration d'un fichier de données se fait en précisant le mot « **FICHIER** » suivi du type des composants.
- ✓ Ce dernier pouvant avoir n'importe quel type (**élémentaire** ou **structuré**).

Type *nom_type* = **Ficher** *de type;*

```
Type
    T : Tableau [1..20] d'entier;
    F: Fichier de T;

    contact= enregistrement;
        nom: chaîne [25];
        Notel: entier;
      Fin
    r: Fichier de contact;
```

```
Var
    F1: Fchier d'entier;
    Enr : F;
    Repertoire : r
```

Déclaration : *Implémentation*

PASCAL

```
Var  nom_var =  File of type;
```

C

```
FILE      * nom_var ;
```

Définition

- ✓ C'est une suite de caractères du jeu ASCII découpée sous forme de lignes, pouvant être de longueurs différentes.
- ✓ Chaque ligne de terminant par un "retour chariot" éventuellement suivi d'un changement de ligne.

Déclaration

- ✓ la déclaration d'un fichier texte se fait simplement en précisant le type « **TEXTE** ».

```
Var      nom_var : Texte;
```

```
Var
        livre: texte;
```

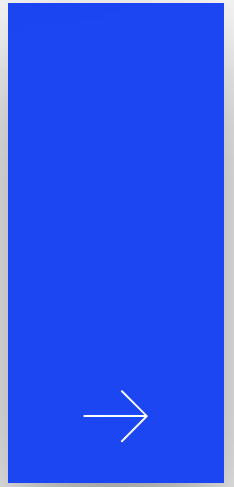
Déclaration : *Implémentation*

PASCAL

```
Var  nom_var =  Text;
```

C

```
FILE      * nom_var ;
```



1. Liaison

- ✓ La liaison entre nom de fichier interne (logique) et nom de fichier externe (physique).
- ✓ Cette liaison se fera grâce à une procédure d'assignation.

ASSIGN (Variable_fichier_interne , Nom_ de_fichier_externe) ;

- ➢ *Variable_fichier_interne* : Fichier
- ➢ *Nom_de_fichier_externe* : chaîne de caractères

2. Ouverture

Après l'étape d'assignation, un fichier doit être ouvert. Deux cas sont possibles :

a) ***Le fichier n'existe pas*** : nous allons le créer. Il s'agit donc d'un nouveau fichier. Dans ce cas l'ouverture se fait à l'aide de la procédure :

REWRITE (f)

b) ***Le fichier existe déjà :*** nous voulons le consulter ou le mettre à jour (création de nouveaux composants, suppression de composants, modification de composants) alors l'ouverture se fera par la procédure:

RESET (f)

3. Fermeture

- Lorsque le traitement d'un fichier est terminé il ne faut fermer le fichier:

CLOSE (f)

4. Fin de ficher

- On peut détecter la fin de fichier en utilisant la fonction suivante:

EOF (f)

5. Positionner

- permet de positionner la tête de lecture/écriture à une position spécifique du fichier : **SEEK (f , pos)**

5. Lecture / Ecriture

- lire ou écrire un composant d'un fichier à l'aide les procédures WRITE et WRITELN, et READ et READLN.
- pour lire ou écrire dans un fichier il suffit de préciser son nom.

WRITE (f , paramètre1 , paramètre2 ,……., paramètren)

WRITELN (f , paramètre1 , paramètre2 ,……., paramètren)

READ (f , paramètre1 , paramètre2 ,……., paramètren)

READLN (f , paramètre1 , paramètre2 ,……., paramètren)

1. Fichiers à accès séquentiel

1. Dans les fichiers à accès séquentiel, les composants ou articles sont rangés les uns à la suite des autres de telle manière à ce que si l'on veut accéder à un composant nous sommes obligés de lire tous ceux qui se trouvent avant.

2. Lors de la lecture dans un fichier à accès séquentiel après les étapes d'assignation et d'ouverture, le pointeur se place automatiquement sur le premier composant (position 0) , et aussitôt que vous faites une lecture le pointeur se place automatiquement sur le composant suivant.

1. Fichiers à accès direct

1. Dans les fichiers à accès direct, il est possible d'accéder directement à un composant. Dans ce cas la répartition des composants, lors de le création du fichier ou de leur recherche dans des opérations de consultation ou de mise à jour, se fait grâce à l'utilisation de formule de transformation de la forme ***F(indicatif) = Adresse physique*** ou adresse relative ou grâce à des tables d'index dans lesquelles on aura des couples (indicatif , position).

2. Dans les fichiers à accès direct, une fois que vous avez la position du composant (qui est fournie par votre formule ou votre table index) vous pouvez y accéder directement en utilisant la procédure ***SEEK (f , POS).***