

# Support de cours

## INITIATION A L'ALGORITHME

**L1 - MIAGE**

**Equipe Pédagogique Informatique**

*M. ASSOHOUN Egomli Stanislas*

# Plan

- CHI – Généralités et structures de contrôle
- CHII – Types énumérés, Tableaux et enregistrements
- CHIII – Algorithme procédural
- CHV- Fichiers

# Objectifs:

**Dans ce cours, nous allons:**

- Décrire les structure de données de type énuméré, tableaux et enregistrements.
- Ecrire des algorithmes pour les opérations de base sur ce type de structure
- Utiliser ces structures de données pour résoudre des problèmes concrets

# **CH I I– Types énumérés, Tableaux et enregistrements**

# CH II - Types énumérés, Tableaux et enregistrements

## I- Types énumérés

**I-1- Motivation :** Parfois, dans un programme on est obligé de prendre des conventions (des codages) pour représenter certaines informations. Par exemple, pour représenter un mois de l'année on peut décider de prendre des entiers avec 1 pour janvier, 2 pour février, etc.

Le principe d'un type énuméré est de donner un nom symbolique pour chacune des valeur que peut prendre une variable.

## I-2- Définitions

Les types énumérés permettent à l'utilisateur de définir son propre type en indiquant explicitement l'ensemble de ses valeurs..

# CH II - Types énumérés, Tableaux et enregistrements

## I-3- syntaxe

### Type

`nom_type_enumere = (val1, val2, ..., valn)`

### ❑ Exemple :

#### Type

`Mois = (janvier, février, ..., décembre)`

`Jour = (lundi, mardi, mercredi, jeudi, vendredi, samedi, dimanche)`

`Couleur = (rouge, jaune, vert, bleu, violet)`

**Remarque:** le programmeur utilisera le nom symbolique pour désigner une valeur particulière du type

# CH II - Types énumérés, Tableaux et enregistrements

## I-3- syntaxe (suite)

□ Exemple :

Var

m: Mois

c: Couleur

DEBUT

m <- "MAI "

c <- "VIOLET"

m <- 'c' - - interdit car de types différents !

...

FIN

# CH II - Types énumérés, Tableaux et enregistrements

## I-3- syntaxe (suite)

### ❑ Exemple :

Var

m: Mois

c: Couleur

DEBUT

m <- MAI

c <- VIOLET

m <- c -- interdit car de types  
différents !

...

FIN

### ❑ Remarque :

Les booléens sont un exemple de type énuméré avec deux valeurs FAUX et VRAI.

### Type

Booléen = (FAUX, VRAI)



# **CH II - Types énumérés, Tableaux et enregistrements**

## **II- Les tableaux**

### **II-1- Contexte et justification**

Jusqu'à présent, nous avons uniquement utilisé des variables simples qui correspondaient à un unique emplacement en mémoire d'un type particulier. Cependant, il est fréquent de devoir manipuler un grand nombre (par forcément très grand d'ailleurs) de données du même type auxquelles on applique les mêmes traitements.

### **II-2- Définition**

Un tableau est une variable qui permet de rassembler sous un même nom (celui de la variable) un nombre fini d'éléments ayant tous le même type.

# CH II - Types énumérés, Tableaux et enregistrements

## II- Les tableaux

### II-2- Définition (suite)

Un élément particulier d'un tableau est désigné en précisant son **indice**. On parle alors de tableau à une dimension.

Il est peut être nécessaire de préciser plusieurs indices pour accéder à une donnée. On parle alors de tableau à plusieurs dimensions.

Lorsqu'un tableau est défini, le nombre d'éléments qu'il peut contenir doit être précisé (en général au travers de la spécification des valeurs possibles des indices) et il ne pourra pas être changé par la suite. Ce nombre d'éléments est appelé la capacité du tableau.

# CH II - Types énumérés, Tableaux et enregistrements

## II- Tableaux de dimension 1

### II-1- Définition

Un tableau à une dimension unidimensionnel ou est un tableau dont on accède aux éléments (pour « lire » ou modifier leur valeur) en utilisant un seul indice (ou index).

Ex :     : Notes 

		.....	
--	--	-------	--

### II-2- Syntaxe

**TABLEAU** Nom\_du\_tableau [nombre de composants] : Type\_des\_composants

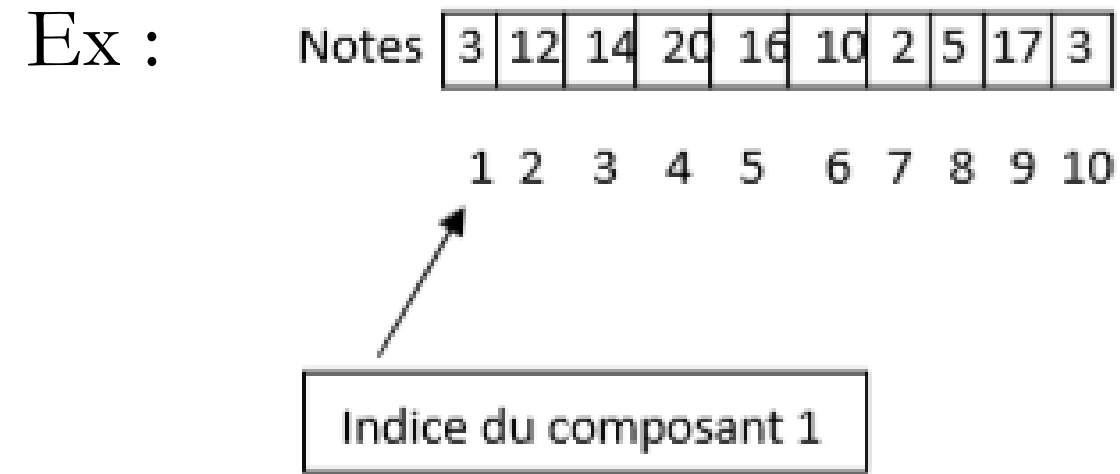
Ex : TABLEAU Notes [10] : Réel

# CH II - Types énumérés, Tableaux et enregistrements

## II- Tableaux de dimension 1

### II-3- Accès aux composants d'un tableau

Dans un tableau à une dimension un composant est repéré par un indice



i : Entier

Notes [3]  $\leftarrow$  14 - Notes [10]  $\leftarrow$  3

Notes[i] représente la composante d'indice i .

# CH II - Types énumérés, Tableaux et enregistrements

## II- Tableaux de dimension 1

### II-4- Saisie des composants d'un tableau à une dimension

Syntaxe

```
POUR i ← 1 à 10 FAIRE  
|   SAISIR(Notes[i])  
FPOUR
```

### II-5- Saisie des composants d'un tableau à une dimension

Syntaxe

```
POUR i ← 1 à 10 FAIRE  
|   AFFICHER (Notes[i])  
FPOUR
```

**Application :** Ecrire un programme qui permet de saisir 5 notes dans un tableau et de les afficher.

# CH II - Types énumérés, Tableaux et enregistrements

## II- Tableaux de dimension 1

### II-6- Opérations sur tableau de dimension 1

#### II-6-1- Recherche séquentielle

La recherche séquentielle encore appelée méthode du drapeau utilise un drapeau pour initier la recherche

#### ❑ Méthode:

**Etape1** : baisser le drapeau

**Etape2** : parcourir le tableau pour rechercher un élément

**Etape3** : lever le drapeau dès que élément recherché est trouvé

**Etape 4** : à la fin du parcours tester l'état du drapeau. S'il est levé l'élément recherché est dans le tableau sinon la recherche est considéré comme infructueuse

# CH II - Types énumérés, Tableaux et enregistrements

## II-6- Opérations sur tableau de dimension 1

### II-6-1- Recherche séquentielle (Suite)

#### Description

T[5] : tableau de 5 composants

i : indice de parcours du table

val: la valeur recherché

Pos : position de l'élément recherché

Drapeau: le drapeau

Algo recherche\_sequentielle

Var

TABLEAU T[5] : Entier

i , val , pos , drapeau : Entier

```
Début
    AFFICHER ("Entrer 5 entiers différents")
    POUR i ← 1 à 5 FAIRE
        SAISIR ( T[i] )
    FPOUR
    AFFICHER ("Entrer la valeur à rechercher")
    SAISIR(val)
    Drapeau ← 0
    Pos ← -1
    POUR i ← 1 à 5 FAIRE
        SI ( T[i] = val ) ALORS
            Drapeau ← 1
            Pos ← i
        FSI
    FPOUR
    SI ( drapeau = 1 ) ALORS
        AFFICHER (val, " est à la position ", Pos , "du tableau")
    SINON
        AFFICHER (val, " n'est pas dans le tableau")
    FSI
Fin
```

# CH II - Types énumérés, Tableaux et enregistrements

## II- Tableaux de dimension 1

### II-6- Opérations sur tableau de dimension 1

#### II-6-2- Recherche Dichotomique

La recherche Dichotomique se base sur le principe de la recherche dans un dictionnaire. Il est utiliser pour rechercher un élément dans un tableau ordonné.

#### ❑ Méthode:

**Etape 1 :** déterminer l'élément médian (au milieu du tableau). Puis le comparer à l'élément recherché.

**Etape 2 :** s'il est égale à l'élément recherché c'est fini

**Etape 3 :** si est plus petit, reprendre les étapes précédente sur la partie à droite de l'élément médian

**Etape 4 :** si est plus grand, reprendre les étapes précédente sur la partie à gauche de l'élément médian



# CH II - Types énumérés, Tableaux et enregistrements

## II-6- Opérations sur tableau de dimension 1

### II-6-2 Recherche Dichotomique

#### Recherche manuelle

Application au tableau ci-contre (T)

-1	8	12	20	32
----	---	----	----	----

Bas	Haut	Mil = <u>(bas+haut)/2</u>	T[mil]	<u>val</u>	Pos
1	5	3	12	-1	-1
1	2	1	-1	-1	1
1	5	3	12	-20	-1
4	5	4	20	20	4
1	5	3	12	-25	-1
1	2	1	-1	-25	-1
1	0	Arrêter la recherche			

# CH II - Types énumérés, Tableaux et enregistrements

## II-6- Opérations sur tableau de dimension 1

### II-6-2 Recherche Dichotomique (Algorithme)

**ALGO** RechercheDichotomique

**VAR**

Tableau T[5] : reel

i,bas,haut,mil,pos,val : entier

**DEBUT**

*(\*remplir le tableau\*)*

AFFICHER ("Entrer cinq nombres ")

POUR (i ← 1 A 5) FAIRE

AFFICHER ("Entrer le nombre N°", i)

SAISIR(T[i])

FPOUR

*(\*saisir l'élément à rechercher\*)*

AFFICHER ("Entrer la valeur à rechercher")

SAISIR (val)

*(\*effectuer la recherche\*)*

Pos ← -1

Bas ← 1

Haut ← 5

**REPETER**

Mil ← (bas + haut) / 2

Si (T[mil] = val) ALORS

Pos ← mil

SINON SI (T[mil] < val) ALORS

Bas ← mil+1

SINON

haut ← mil-1

FSI

IQA(pos <> -1 OU bas > haut)

*(\* afficher le résultat \*)*

SI (pos = -1) ALORS

AFFICHER(Val, " n'est pas dans le tableau")

SINON

AFFICHER(val, " est à la position ", pos)

**FIN**

# CH II - Types énumérés, Tableaux et enregistrements

## II- Tableaux de dimension 1

### II-6- Opérations sur tableau de dimension 1

#### II-6-3- Tri par sélection (du minimum)

La tri par sélection du minimum parcourt le tableau pour rechercher l'indice du minimum. dès qu'il l'a trouvé il le permute avec l'élément en fin de tableau. La taille du tableau est décrémenté puis le même principe est exécuté sur ce nouveau tableau

#### ❑ Méthode:

**Etape 1 :** supposer que le premier élément est le maximum.

**Etape 2 :** parcourir le tableau pour récupérer l'indice du maximum réel

**Etape 3 :** permuter le maximum réel avec l'élément en dernière position

**Etape 4 :** réduire la taille du tableau de 1

**Etape 4 :** reprendre le même principe

# CH II - Types énumérés, Tableaux et enregistrements

## II- Tableaux de dimension 1

### II-6- Opérations sur tableau de dimension 1

#### II-6-3- Tri par sélection (du maximum)

❑ Exemple :

Tri par sélection

-9	10	12	-4	2	8
-9	10	8	-4	2	12
-9	2	8	-4	10	12
-9	2	-4	8	10	12
-9	-4	2	8	10	12
-9	-4	2	8	10	12

Tableau des indices

i	lmax	Fin
1	1	6
2	2	
3	3	
4	3	
5	3	
6	3	
1	1	5
2	2	
3	2	
4	2	
5	2	
1	1	4
2	2	
3	3	
4	3	
1	1	3
2	2	
3	2	
1	1	2
2	2	

# CH II - Types énumérés, Tableaux et enregistrements

## II- Tableaux de dimension 1

### II-6- Opérations sur tableau de dimension 1

#### II-6-3- Tri par sélection (algorithme)

Algo Tri\_selection

Var

T[10] : Entier

i, j, max, temp : Entier

```
Début
  AFFICHER ("Entrer 10 entiers")
  POUR i ← 1 à 10 FAIRE
    SAISIR (T[i])
  FPOUR
  POUR i ← 1 à 9 FAIRE
    Max ← i
    POUR j ← i+1 à 10 FAIRE
      SI (T[j] > T[max]) ALORS
        max ← j
      FSI
    FPOUR
    temp ← T[max]
    T[m] ← T[j]
    T[j] ← temp
  FPOUR
  AFFICHER ("Le tableau trié dans l'ordre décroissant est : ")
  POUR i ← 1 à 10 FAIRE
    AFFICHER(T[i])
  FPOUR
Fin
```

# CH II - Types énumérés, Tableaux et enregistrements

## II- Tableaux de dimension 1

### II-6- Opérations sur tableau de dimension 1

#### II-6-4- Tri bulle (**voir TD**)

## II- Tableaux de dimension 2

### II-1- définition

Un tableau à 2 dimensions est interpréter comme un tableau à une dimension de L composants dont chaque composant correspond lui-même à un tableau à une dimension de C composants. On appelle L le nombre de lignes et C le nombre de colonnes du tableau. Un tableau à 2 dimension contient donc  $L \times C$  composants.

# CH II - Types énumérés, Tableaux et enregistrements

## III- Tableaux de dimension 2

### III-1- définition (suite)


L lignes

C colonnes

### III-2- Déclaration

**TABLEAU** Nom\_du\_tableau[Nombre\_de\_lignes,Nombre\_de\_colonnes] : Type\_des\_composants

Ex : **TABLEAU** A[10,5] : Entier

# CH II - Types énumérés, Tableaux et enregistrements

## III- Tableaux de dimension 2

### III-3- Saisie des éléments d'un tableau à 2 dimensions

TABLEAU A[5,3] : Réel

i, j : Entier

POUR i ← 1 à 5 FAIRE

POUR j ← 1 à 3

SAISIR(A[i,j])

FPOUR

FPOUR



# CH II - Types énumérés, Tableaux et enregistrements

## III- Tableaux de dimension 2

### III-3- Affichage des éléments d'un tableau à 2 dimensions

```
POUR i ← 1 à 5 FAIRE
|
|   POUR j ← 1 à 3 FAIRE
|   |
|   |   AFFICHER (A[i,j])
|   |
|   FPOUR
|
FPOUR
```

# CH II - Types énumérés, Tableaux et enregistrements

## IV- Enregistrements

### IV-1- Motivation

Un type enregistrement permet de définir des types structurés (ou complexes) qui ne peuvent pas être représentés par les types élémentaires que nous avons déjà vus (Entier, Booléen, Réel, etc.), ni par un tableau.

**EXEMPLES** : Voici quelques exemples d'enregistrements :

- Un complexe est défini par une partie réelle et une partie imaginaire.
- Une date est composée d'un numéro de jour (1..31), d'un numéro de mois (1..12) et d'une année (strictement positive).
- Un stage peut être défini comme un intitulé (une chaîne de caractères), une date de début et une date de fin (deux dates), un nombre de places (entier).
- Une fiche bibliographique est définie par le titre du livre (Chaîne), les auteurs (noms et prénoms), la date de parution, l'éditeur, le numéro ISBN (Chaîne), etc.

# CH II - Types énumérés, Tableaux et enregistrements

## IV- Enregistrements

### IV-2- Définition d'un type enregistrement

Un enregistrement est un type  $T$  qui est le produit cartésien de plusieurs types  $T_1, T_2, \dots, T_n$ . À chaque composante de type  $T_i$  est associé un identificateur choisi par le programmeur. Il permet de sélectionner la composante. Le couple (identificateur, Type) est appelé un champ de l'enregistrement

#### Types

```
T_Enregistrement =  -- Le nom du type
    Enregistrement  -- sa définition
        nom_champ1: T1                -- un champ, son type et son rôle
        ...
        nom_champn: Tn
    FinEnregistrement
```

Par exemple, le type correspondant à une date peut être décrit ainsi :

```
Date =
    Enregistrement
        jour: 1..31
        mois: 1..12
        annee: Entier  -- annee > 0
    FinEnregistrement
```

# CH II - Types énumérés, Tableaux et enregistrements

## IV- Enregistrements

### IV-2- Définition d'un type enregistrement (suite)

Intuitivement, un enregistrement permet donc de regrouper dans une même structure un ensemble d'informations ayant entre elles un lien logique.

Intérêt : Les enregistrements permettent de manipuler simultanément un ensemble de données logiquement reliées.

Par exemple, pour avoir deux dates, il suffit de déclarer deux variables d1 et d2 du type Date plutôt que j1, m1, a1 et j2, m2, a2 de type Entier.

### IV-3- Déclaration d'une variable de type enregistrement

Un type enregistrement est avant tout un type. Il permet donc de déclarer des variables de ce type.

#### **Variables**

```
d1: Date  -- d1 est une variable de type Date.  
          -- Elle occupe 3 entiers en mémoire.
```

# CH II - Types énumérés, Tableaux et enregistrements

## IV- Enregistrements

### IV-4- Manipulation d'une variable de type enregistrement

Sur les variables de type enregistrement, un seul opérateur existe l'affectation. Il consiste à copier toutes les composantes de l'enregistrement.

Tous les autres traitements doivent être explicitement réalisés par le programmeur. Pour ce faire, il peut accéder individuellement à chaque composante grâce à son identifiant. La composante (le champ) se comporte alors exactement comme tout autre variable du même type que cette composante.

**Remarque :** Il est conseillé d'écrire des sous-programmes réalisant les opérations usuelles sur le type enregistrement : voir sous-programmes et types abstraits de données.

Par exemple, si `d` est une variable de type `Date`, on peut accéder au jour de cette date en écrivant **`d.jour`**. On peut alors l'écrire, le lire, le comparer... et plus généralement lui appliquer toutes les opérations définies sur le type.

**Attention :** Lorsqu'on utilise « `.` », il est nécessaire que l'expression à gauche soit d'un type Enregistrement et qu'à droite se trouve un identificateur correspondant à un nom de champ du dit enregistrement. Ces vérifications sont réalisées par le compilateur.

# CH II - Types énumérés, Tableaux et enregistrements

## IV- Enregistrements

### IV-4- Manipulation d'une variable de type enregistrement (algorithme)

**ALGO** EnregistrementDate

**TYPE**

Date = Enregistrement

Jour : entier

Mois entier

Annee : entier

FinEnregistrement

**VAR**

d1, d2 : date

**DEBUT**

AFFICHER ("Entrer la date jour mois et année :")

AFFICHER ("Entrer le jour : ")

SAISIR(d1.jour)

AFFICHER ("Entrer le mois : ")

SAISIR(d1. mois)

AFFICHER ("Entrer l'année : ")

SAISIR(d1.annee)

AFFICHER (" Les informations saisie sont : ", d1.jour," -", d1.mois," -", d1.annee)

**FIN**

---

**Merci de votre  
attention**