**Chapitre 12 : Les types structures et les enregistrement**

# Introduction

Contrairement aux tableaux dont tous les éléments sont de même type, **les enregistrements (Les structures) sont des structures de données dont les éléments peuvent être de type différent**.

Les éléments qui composent un enregistrement sont appelés **champs**.

Avant de déclarer une variable enregistrement, il faut avoir au préalable définit son type, **c'est à dire le nom et le type des champs qui le compose**. Le type d'un enregistrement est appelé type structuré. (Les enregistrements sont parfois appelé structures, en analogie avec le langage C)

# Les structures

## En algorithme

### Déclaration d’une structure

La déclaration des types structurés se fait dans une section spéciale des algorithmes appelée Type, qui précède la section des variables (et succède à la section des constantes).

**La syntaxe de déclaration d’une structure est :**

**Type**

**Structure** nom\_type

Nom\_champ1 : type\_de\_champ

Nom\_champ2 : type\_de\_champ

…

Nom\_champN : type\_de\_champ

**FinStruct**

**Exemple 1 : structure pour représenter des personnes**

**Type**

**Structure t**Personne

Nom : **chaine**

Prenom : **chaine**

Age : **int**

**FinStruct**

**Exemple 2 : structure pour représenter des produits**

**Type**

**Structure t**Produit

Code : **chaine**

Libelle : **chaine**

Prix\_Achat : **réel**

Prix\_Vente **: réel**

**FinStruct**

### Déclaration d'un enregistrement à partir d'un type structuré.

Une fois qu'on a défini un type structuré, on peut déclarer des variables enregistrements exactement de la même façon que l'on déclare des variables d'un type primitif.

**Syntaxe :**

**Variable**

nom\_var : nom\_type\_structure

**Exemple :**

**Variable** pers1, pers2 : tpersonne

Représentation : Les variables pers1 et pers2 sont composé de 3 zones mémoire :

Pers 1 :

nom prénom âge

|  |  |  |
| --- | --- | --- |
|  |  |  |

Pers 2 :

nom prénom âge

|  |  |  |
| --- | --- | --- |
|  |  |  |

### Accès aux champs d’un enregistrement

Alors que les éléments d'un tableau sont accessibles au travers de leur indice, les champs d'un enregistrement sont accessibles à travers leur nom, grâce à l'opérateur '.'

**nom\_enregistrement . nom\_champ**

Par exemple, pour accéder à l'âge de la variable pers2, on utilise l'expression:

**pers2.âge**

### Exemple complet

**Algorithme** exemple1\_Personne

**Type**

Structure tPersonne

Nom : chaine

Prenom : chaine

Age : int

FinStruct

**Variable** pers1, pers2: tPersonne

**Début**

Ecrire("Saisir les données de la personne 1 :")

Ecrire("Donner le nom :")

Lire(pers1.nom)

Ecrire("Donner le prénom : ")

Lire(pers1.prenom)

Ecrire("Donner l’age :")

Lire(pers1.age)

Ecrire("Saisir les données de la personne 2 :")

Ecrire("Donner le nom :")

Lire(pers2.nom)

Ecrire("Donner le prénom : ")

Lire(pers2.prenom)

Ecrire("Donner l’age :")

Lire(pers2.age)

Ecrire("Personne 1 : ",pers1.nom, " – ", pers1.prenom," – ", pers1.age)

Ecrire("Personne 2 : ",pers2.nom, " – ", pers2.prenom," – ", pers2.age)

**Fin**

### Passage d'un enregistrement en paramètre d'un sous-programme

Il est possible de passer tout un enregistrement en paramètre d'une fonction ou d'une procédure (on n'est pas obligé de passer tous les champs uns à uns, ce qui permet de diminuer le nombre de paramètres à passer), exactement comme pour les tableaux.

**Exemple 1 :**

Voilà une fonction qui renvoie la différence d'age entre deux personnes

**Fonction** différence (p1, p2 : tPersonne) : entier

**Début**

Si p1.age > p2.age Alors

Retourne ( p1.age – p2.age )

Sinon

Retourne ( p2.age – p1.age )

FinSi

**FinFonct**

**Exemple 2 :**

Voilà une procédure qui affiche les informations d’une personne :

**Precedure** différence (p1: tPersonne)

**Début**

Ecrire("Les informations de la personne sont : ")

Ecrire("Le nom de la personne : ",pers1.nom)

Ecrire("Le nom de la personne : ", pers1.prenom)

Ecrire("Le nom de la personne : ", pers1.age)

**FinFonct**

## En langage C

Nous avons trois façons pour déclarer une structure :

**Méthode 1**:

La syntaxe de déclaration est :

struct nom\_structure

{

type\_champ1 nom\_champ1 ;

type\_champ2 nom\_champ2 ;

….

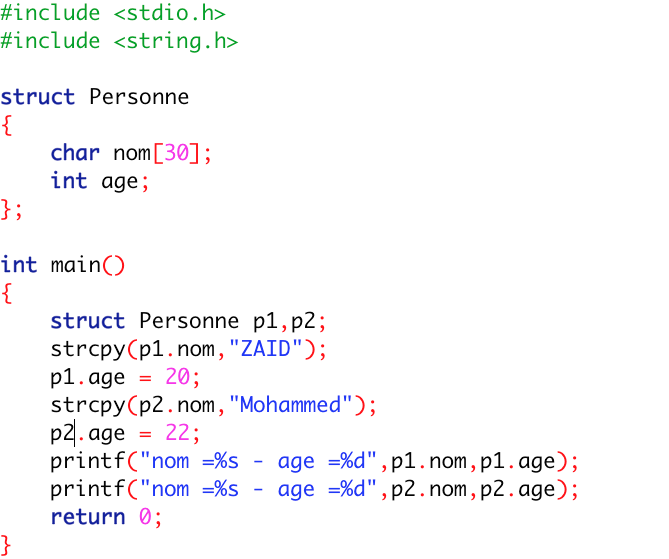
type\_champN nom\_champN ;

} ;

Déclaration des enregistrements

**struct nom\_structure nom\_enregistrement**

Exemple de déclaration des variables avec cette méthode :



**Méthode 2**:

La syntaxe de déclaration est :

struct nom\_structure

{

type\_champ1 nom\_champ1 ;

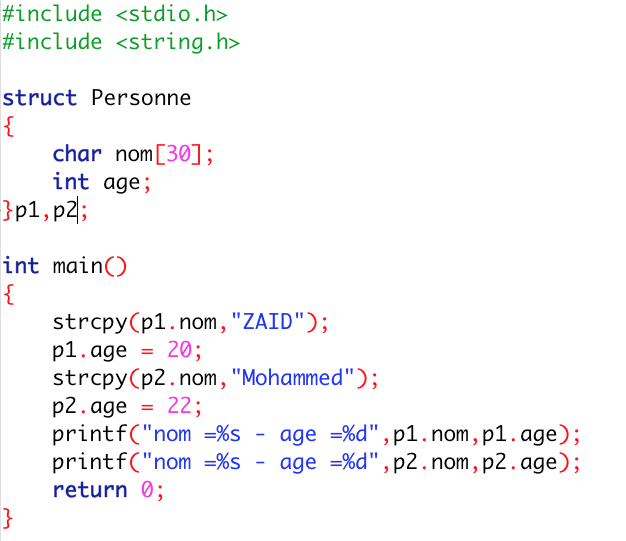
type\_champ2 nom\_champ2 ;

….

type\_champN nom\_champN ;

} nom\_enregistrement1,…;

Exemple de déclaration des variables avec cette méthode :



**Méthode 3**:

La syntaxe de déclaration est :

typedef struct

{

type\_champ1 nom\_champ1 ;

type\_champ2 nom\_champ2 ;

….

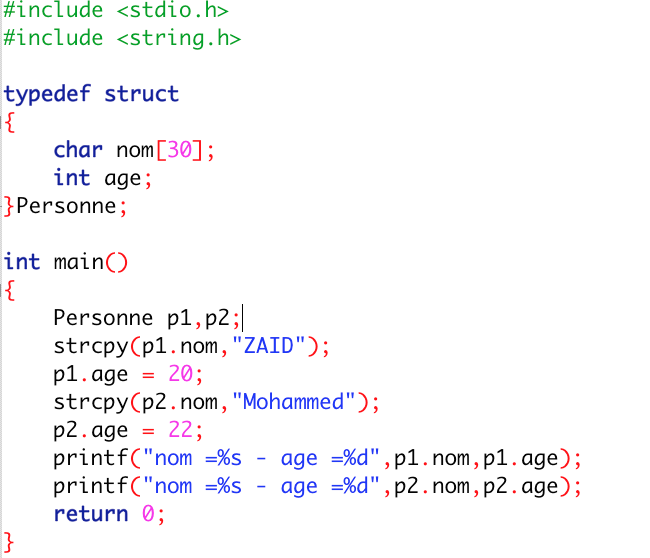
type\_champN nom\_champN ;

}  nom\_structure;

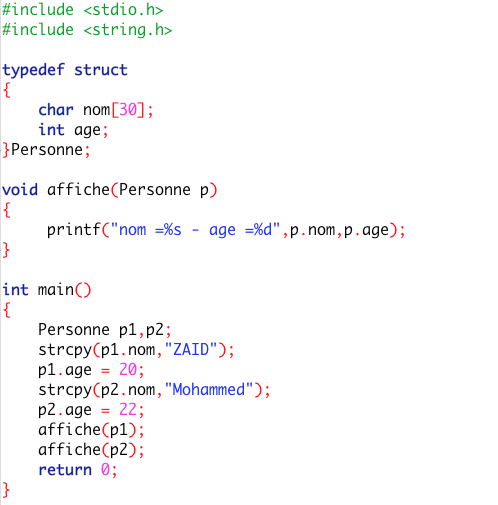
Déclaration des enregistrements

**nom\_structure nom\_enregistrement**

Exemple de déclaration des variables avec cette méthode :



**Exemple d’utilisation d’une structure en paramètre :**

****

# Exercices d’application

## Exercice 1 :

Traduire l’algorithme suivant en langage C :

**Algorithme** exemple1\_Personne

**Type**

Structure tPersonne

Nom : chaine

Prenom : chaine

Age : int

FinStruct

**Variable** pers1, pers2: tPersonne

**Début**

Ecrire("Saisir les données de la personne 1 :")

Ecrire("Donner le nom :")

Lire(pers1.nom)

Ecrire("Donner le prénom : ")

Lire(pers1.prenom)

Ecrire("Donner l’age :")

Lire(pers1.age)

Ecrire("Saisir les données de la personne 2 :")

Ecrire("Donner le nom :")

Lire(pers2.nom)

Ecrire("Donner le prénom : ")

Lire(pers2.prenom)

Ecrire("Donner l’age :")

Lire(pers2.age)

Ecrire("Personne 1 : ",pers1.nom, " – ", pers1.prenom," – ", pers1.age)

Ecrire("Personne 2 : ",pers2.nom, " – ", pers2.prenom," – ", pers2.age)

**Fin**

## Exercice 2 :

1. Définir une structure « Point » dont les champs sont :

|  |  |
| --- | --- |
| Nom champs | Type champs |
| Nom\_point | Caractére |
| X | entier |
| Y | Entier |

1. Déclarer trois points ; donner la main à l’utilisateur pour saisir leurs données ;
2. Déclarer une procédure qui permet d’afficher un point sous la forme :

Point A de coordonner 2 , 3

3. Tester la procédure dans la fonction main.

## Exercice 3 :

Soit la structure Personne définie par les champs : nom, prénom, âge ;

1. Déclarer deux personnes ; donner la main à l’utilisateur pour saisir leurs données ;
2. Ajouter une procédure qui permet d’afficher les informations d’une personne sous la forme suivante.

Nom, prénom (age)

1. Afficher les 2 personnes triées selon leur âge par ordre décroissant ;
2. Afficher les 2 personnes triées selon leur nom par ordre décroissant.

## Exercice 4 :

1. Définir une structure « Date » dont les champs sont :

|  |  |
| --- | --- |
| Champ | Type |
| Jour | Entier |
| Mois | Entier |
| Année | Entier |

1. Saisir deux dates (on supposera qu’il s’agit de dates valides) et afficher si la première date est antérieure ou postérieure à la seconde.
2. Ajouter une fonction qui accepte comme paramètre 2 dates et qui retourne la différence entre ces deux dates en mois ;

# L’imbrication d’enregistrement

Supposons que dans le type personne, nous ne voulions plus l'âge de la personne, mais sa date de naissance. Une date est composée de trois variables (jour, mois, année) indissociables. Une date correspond donc à une entité du monde réel qu'on doit représenter par un type enregistrement à 3 champs.

Si on déclare le type date au préalable, on peut l'utiliser dans la déclaration du type personne pour le type de la date de naissance.

Un type structuré peut être utilisé comme type pour des champs d'un autre type structuré

**En algorithme :**

Type

Structure Date

Jour : entier

Mois : entier

Annee : entier

FinStruct

Structure Personne

Cin : chaine

Nom : chaine

Prenom : chaine

Date\_naissance : Date

FinStruct

**En langage C :**

Typedef struct

{

int jour, mois, annee ;

} Date ;

Typedef struct

{

char cin[50],nom[50],prenom[50] ;

Date date\_naissance ;

}

Pour accéder à l'année de naissance d'une personne, il faut utiliser deux fois l'opérateur '.'

pers1. date\_naissance .annee

Il faut lire une telle variable de droite à gauche : l'année de la date de naissance de la personne 1.

# Les tableaux d’enregistrements

Il arrive souvent que l’on veuille traiter non pas un seul enregistrement mais plusieurs. Par exemple, on veut pouvoir traiter un groupe de personne. On ne va donc pas créer autant de variables du type personne qu’il y a de personnes. On va créer un tableau regroupant toutes les personnes du groupe. Il s’agit alors d’un tableau d’enregistrements.

**En algorithme**

**Type**

Structure Personne

Nom : chaine

Age : entier

FinStruct

**Variable** tableau groupe[50] : personne

**En langage C**

Typedef struct

{

char nom[50] ;

int age ;

} Personne;

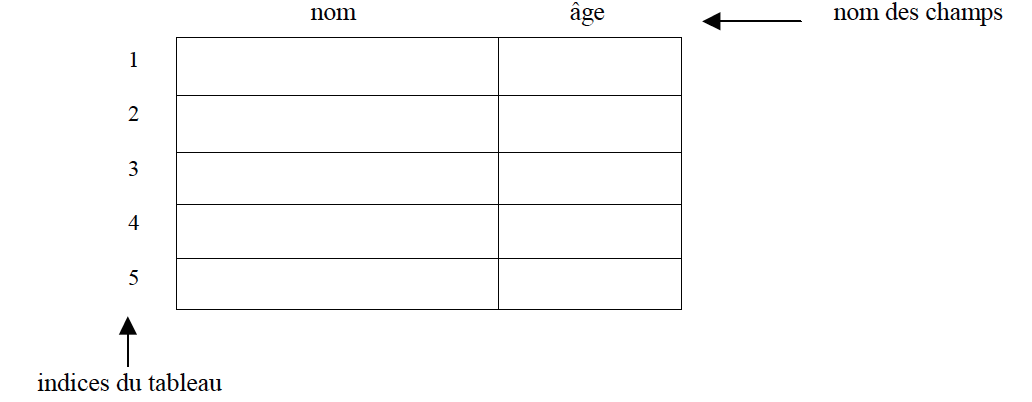
Déclaration de tableau :

Personne groupe[50] ;

Chaque élément du tableau est un enregistrement, contenant plusieurs variables de type différent. On accède à un enregistrement par son indice dans le tableau.

groupe[2] représente la deuxième personne du groupe

groupe[2].nom représente le nom de la deuxième personne du groupe



**Attention!**

groupe.nom[3] n'est pas valide.

Pour accéder au nom de la troisième personne du tableau, il faut écrire groupe[3].nom

# Exercice d’application

**Exercice 1:**

1. Définir une structure « Date » dont les champs sont :

|  |  |
| --- | --- |
| Champ | Type |
| Jour | Entier |
| Mois | Entier |
| Année | Entier |

* 1. Définissez la structure « Stagiaire » constituée des champs suivants :

|  |  |
| --- | --- |
| Champ | Type |
| Nom | Chaîne |
| Prénom | Chaîne |
| DateN | Date |

* 1. Ecrivez ensuite l’algorithme qui permet de lire et après afficher le nom, prénom et date de naissance d’un stagiaire.

**Exercice 2 :**

1. On souhaite gérer les notes d’un étudiant. Pour cela on va définir la structure « Etudiant » dont les champs sont :

|  |  |
| --- | --- |
| **Champ** | **Type** |
| Nom | Chaîne |
| Prénom | Chaîne |
| Note | Tableau de 3 éléments |
| Moyenne | Réel |

1. Ecrire l’algorithme qui permet de lire les informations d’un étudiant (nom, prénom et notes), de calculer sa moyenne et d’afficher à la fin un message sous la forme suivante :

« La moyenne de l’étudiant Dinar Youssef est : 12.45 »

Où « Dinar » et « Youssef » sont les noms et prénoms lus et 12.45 est la moyenne calculée.

1. Modifier l’algorithme de l’exercice précédent de façon que l’on puisse gérer les notes de 20 étudiants.
2. Trier puis afficher la liste des étudiants selon leur moyenne par ordre décroissant (utiliser le tri par sélection).
3. Afficher le nom, prénom, la liste des notes de l’étudiant majorant de la classe.