

Programmation Procédurale 1

Equipe Algorithmique & Programmation

Année universitaire :2021 - 2022



Les structures & Tableaux de structures



Plan



- Les structures
 - ✓ Définition
 - ✓ Syntaxe et Déclaration
- Les tableaux de structures
 - ✓ Définition
 - ✓ Syntaxe et Déclaration
 - ✓ Accès à un champ



Objectifs



Objectifs



A la fin de ce chapitre, l'apprenant sera capable de:

- Définir une structure, la déclarer, l'initialiser et l'utiliser.
- Accéder et manipuler les champs d'une structure.
- Déclarer une structure en utilisant *typedef*.
- Manipuler les structures imbriquées et les tableaux de structures







Les Structures

Définition

Le tableau permet de désigner **sous un seul nom** un ensemble de **valeurs de même type**, repérées par des indices.

TabAg
e

22	21	22	19
----	----	----	----

TabMoyenn
e

12.25	18.50	16.75	15.25
-------	-------	-------	-------

Un enregistrement, appelé ***structure*** en langage C, est une variable complexe qui permet de désigner **sous un seul nom** un ensemble de valeurs pouvant être **de type différent**.

Etudiant

Nour	22	12.25
------	----	-------

Déclaration d'une Structure (1/2)

Syntaxe

```
struct <Nom_Structure> {  
  <type_champ1> <Nom_Champ1>;  
  <type_champ2> <Nom_Champ2>;  
  <type_champ3> <Nom_Champ3>;  
  ...  
};
```

Noms
différents

Tous les types

Exemple

```
struct etudiant  
{  
  char nom[30];  
  int age;  
  float moyenneScolaire;  
};
```



Déclaration d'une Structure (2/2)



Remarques :

Deux champs d'une même structure ne peuvent pas avoir le même nom.

- Les données peuvent être de n'importe quel type hormis le type de la structure dans laquelle elles se trouvent.
- La déclaration d'une structure ne fait que donner l'allure de la structure, donc elle n'entraîne pas la réservation d'espace mémoire pour une variable structurée (variable de type structure).
- Il faut donc définir une (ou plusieurs) variable(s) structurée(s) après avoir déclaré la structure.

► Déclaration d'une Variable (1/3)



- Une variable structurée doit être définie comme suit :

struct <Nom_Structure> <Var_Structure>;

Exemple:

```
struct etudiant {  
    char nom[30];  
    int age;  
    float moyenneScolaire;  
};
```

```
struct etudiant E1, E2;
```



Exemple:

```
struct etudiant {  
    char nom[30];  
    int age;  
    float moyenneScolaire;  
} E1, E2;
```

► Déclaration d'une Variable : Utilisation de typedef (2/3)

Syntaxe

```
typedef struct <Nom_Structure> <nom_Definition>;
```

Exemple:

```
struct etudiant {  
    char Nom[30];  
    int Age;  
    float MoyenneScolaire;  
};
```

```
typedef struct etudiant Etudiant;
```

Déclaration

```
<Nom_Définition> <Var_Structure>;  
Etudiant E1;
```



► Déclaration d'une Variable : Utilisation de typedef (3/3)

Syntaxe

```
typedef struct {  
    <type_champ1> <Nom_Champ1>;  
    <type_champ2> <Nom_Champ2>;  
    <type_champ3> <Nom_Champ3>;  
    ...  
} <Nom_Structure> ;
```

Exemple

```
typedef struct {  
    char Nom[30];  
    int Age;  
    float MoyenneScolaire;  
} Etudiant;
```

Déclaration

```
<Nom_Structure> <Var_Structure>;  
Etudiant E1;
```



Initialisation d'une Structure



Initialiser une structure en indiquant entre accolades {} la liste des valeurs séparées par des virgules.

Exemple:

```
struct etudiant {  
    char nom[30];  
    int age;  
    float moyenneScolaire;  
};  
  
struct etudiant E1={" test ",21,12};
```



Utilisation d'une structure (1/2)



Accès aux champs:

Pour accéder aux champs d'une structure on utilise l'opérateur de champ (un simple **point**) placé entre le nom de la variable structurée et le nom du champ.

Syntaxe:

<Var_Structure>.<Nom_Champ> ;

Pour affecter des valeurs à la variable **E1**, on pourra écrire:

```
E1.Niveau = 2;                // affectation d'un champ
```

```
printf ("donner la moyenne \n ") ;
```

```
scanf ("%f", &E1.Moyenne) ; //stockage dans un champ
```




Utilisation d'une structure (2/2)



Soit E1 et E2 deux variables de type Etudiant:

Il existe deux types d'affectation:

- ❑ Soit d'une façon individuelle (champs par champs), sur chacun de leurs champs:

E1.niveau = E2.niveau;

E1.moyenne = E2.moyenne;

- ❑ Soit d'une manière globale sur toute la structure.

E1 = E2;



Structures contenant des tableaux



Une structure peut contenir des champs de type chaîne de caractères ou bien de type tableau.

Exemple:

```
struct Etudiant
{
    char nom [30];
    char prenom [30];
    float notes [4];
};
struct Etudiant E1,E2;
```

Notation:

E1.nom[0] : désigne le premier caractère du champ nom de l'étudiant E1

E2.notes[3] : désigne la quatrième note du tableau notes de l'étudiant E2



Structures imbriquées

Une structure imbriquée est une structure qui contient aussi un ou plusieurs champ(s) de type structure. Cette dernière doit être différente de la première.

Exemple:

```
struct Date {  
    int jour;  
    int mois;  
    int annee;  
};  
Struct Etudiant {  
    char nom [30] ;  
    char prenom [30] ;  
    struct Date date_naissance ;  
};
```

NB: La structure **Date** doit aussi être déclarée et définie avant la structure **Etudiant**



Application



Exercices 1 et 2 de la série d'exercices



Tableaux de structures



Définition et Déclaration



Un tableau de structures est un tableau dont les éléments sont de type structure.

Syntaxe

```
struct <Nom_Structure>
{
    <type_champ1> <Nom_Champ1>;
    <type_champ2> <Nom_Champ2>;
    <type_champ3> <Nom_Champ3>;
};
struct <Nom_Structure> tab [taille];           // tab est le nom du tableau
```

Exemple

```
struct Etudiant
{
    char nom [30];
    char prenom [30];
    float notes [4];
};
struct Etudiant Classe[10];
```



Accès à un champ



Notation

`Classe[i].note[1]= 15`: désigne l'affectation de la valeur 15 au champ `note[1]` de l'élément de rang `i` du tableau `Classe`.

Remarque : ~~`Classe`~~`note[i]` \Rightarrow N'a pas de sens!



Exemple

```
#include<stdio.h>

typedef struct
{
    char nom [30];
    char prenom [30];
    float notes [4];
} Etudiant;

void main()
{
    Etudiant Classe[5];
    int i,j;
    float Moy;
```

```
/*remplissage des 5 éléments du tableau Classe */
for(i = 0 ; i <5 ; i++)
{
    printf(" Nom:" );scanf("%s",Classe[i].nom);
    printf(" Prenom:" );scanf("%s",Classe[i].prenom);
    for(j=0;j<4;j++)
    {
        printf(" Note %d:" ,j );scanf("%f",&Classe[i].notes[j]);
    }
}

/*Affichage des éléments du tableau Classe*/
for(i = 0 ; i <5; i++)
{
    Moy=0;
    for(j=0;j<4;j++)
        Moy+=Classe[i].notes[j];
    printf(" %s %s a une moyenne de %f" , Classe[i].nom,
        Classe[i].prenom,Moy/4);
}
}
```




Application



Exercice 3 de la série d'exercices