

Support de cours Les enregistrements.

Introduction

Les types élémentaires de l'algorithmique (*entier, réel, booléen, et caractère*) ne permettent pas d'implémenter la sauvegarde des données du monde réel. Un étudiant par exemple est caractérisé par des informations de différents types tel que, nom et prénom de type chaîne, la moyenne de type réel et etc. Il est possible de créer des types composés de plusieurs données de différents types appelés enregistrements. Contrairement aux tableaux, un enregistrement est un type composé de plusieurs données pouvant être de types différents.

Déclaration

Créer un enregistrement en algorithmique consiste à créer un nouveau type avec le mot clé *type*. Les données incluses dans l'enregistrement sont des champs. La définition d'un nouveau type enregistrement implique la définition de chacun de ses champs. Un champ d'un enregistrement est défini par son nom et son type.

syntaxe	
Algorithmique	Langage C
<pre> Type nom_Enreg = Enregistrement id_ch1 : type1 id_ch2 : type2 ... id_chN : typeN Fin;</pre>	<pre> struct nom_struct{ type1 id_ch1 ; type2 id_ch2 ; ... typeN id_chN; } ;</pre>
Exemples	
<pre> Type Etudiant = enregistrement Nom: chaîne[30]; Prenom : chaîne[30] ; Moyenne : reel ; Fin; Module = enregistrement intitule: chaîne[100]; coef : reel; credit : entier; Fin;</pre>	<pre> struct Etudiant{ char Nom[30]; char Prenom[30]; float Moyenne ; } ; struct Module{ char intitule[100]; float coef ; int credit ; } ;</pre>

Une fois le type enregistrement est défini, *nom_Enreg* devient un type et il est possible de déclarer une variable de type enregistrement et également un tableau d'enregistrement de la même façon que l'on déclare des variables d'un type primitif.

Algorithmique	Langage C
<pre> Var id_var : nom_Enreg; Var E: Etudiant; M: Module; T : tableau[1..50]de Etudiant;</pre>	<pre> struct nom_struct id_struct; struct Etudiant E; struct Module M; struct Etudiant T[50];</pre>

Chaque champ d'une variable de type enregistrement est représenté par une case mémoire.

	Nom	Prenom	Moyenne
E :			
	Intitule	Coef	Credit
M :			

Support de cours Les enregistrements.

	1	2	3				
T :	Nom Prenom Moyenne	Nom Prenom Moyenne	Nom Prenom Moyenne				
	[] [] []	[] [] []	[] [] []				

En langage C, pour éviter la répétition du mot `struct` à chaque déclaration d'une variable de type enregistrement, il est préférable de donner un synonyme au type structure au moment de sa définition :

```
typedef struct {
type-1 membre-1;
type-2 membre-2;
...
type-n membre-n;
} nom_struct;
```

Exemples :

Sans le mot clés <i>typedef</i>	Avec le mot clés <i>typedef</i>
<pre>struct personne{ char nom [50]; char prenom [50]; int age; }; struct personne p;</pre>	<pre>typedef struct { char nom [50]; char prenom [50]; int age; } personne; personne p;</pre>

Un champ d'enregistrement peut être de type enregistrement à condition que cet enregistrement soit défini bien avant. Dans un algorithme ou dans programme langage C, il faut respecter l'ordre de définition des enregistrements

Exemple :

Algorithmique	Langage C
<pre>Type Date = enregistrement; Jour : 1..31 ; Moi : 1..12 ; Annee : entier ; Fin ; employe = enregistrement ; nom, prenom : chaine[30] D_naissance : date ; Fin ;</pre>	<pre>typedef struct { int jour, mois, annee; } Date; typedef struct { char nom[30], prenom[30]; Date D_naissance ; } employe;</pre>

Manipulation d'un enregistrement

La manipulation d'un enregistrement se fait à travers ses champs. Comme pour les tableaux, il n'est pas possible de manipuler un enregistrement globalement. Par exemple, pour afficher un enregistrement il faut afficher tous ses champs un par un. Les champs d'un enregistrement sont accessibles à travers leur nom, grâce à l'opérateur `'.'` de la même façon en algorithmique et en langage

C. Nom_variable.nom_champ ;

Support de cours Les enregistrements.

Exemples	
Instruction	Rôle
Lire (E.Nom, E.Prenom, E.Moyenne);	Lecture des champs de la variable E de type Etudiant
Lire (T[1].Moyenne);	Lecture du champ Moyenne du premier élément du tableau T.
Ecrire(M.credit) ;	Affiche le contenu du champ credit de la variable M de type Module
Scanf('%S' ,&E.Nom);	Lecture du nom de l'étudiant E (langage C)

Dans le cas d'un tableau d'enregistrement, chaque élément du tableau est un enregistrement. Les champs d'un élément d'un tableau T sont accessibles avec `T[i].nom_champ`. Avec `i` est l'indice de l'élément du tableau à manipuler.

La répétition du nom de la variable enregistrement pour manipuler chacun des champs de l'enregistrement est ennuyeuse. Il est possible d'utiliser l'opérateur '**Avec**'. A l'intérieur de l'instruction Avec, on peut manipuler directement les champs de l'enregistrement sans l'opérateur '.' Par contre l'opérateur '**Avec**' n'existe pas en langage C.

Exemple :

Avec (M) faire Lire(intitule) ; Lire(coef) ; Lire(credit) ; finAvec ;	Avec (T[i]) faire Lire(Nom) ; Lire(Prenom) ; Lire(Moyenne) ; finAvec ;
---	--

Les enregistrements et les tableaux

Comme nous l'avons vu précédemment, il est possible de déclarer un tableau d'enregistrement en respectant la syntaxe de déclaration d'un tableau.

```
Var nom_Tab : tableau[inf . . . sup] de type_enregistrement ;
```

A condition que `type_enregistrement` soit défini avant. Il est préférable de créer un type synonyme pour le type tableau et surtout lorsque le type tableau est utilisé comme paramètre de sous-programmes. Il est également possible qu'un champ d'un enregistrement soit un tableau.

Exemple :

```
Type Etudiant = enregistrement
    Nom: chaine[30];
    Prenom : chaine[30] ;
    Note : tableau[1..8] de reel ;
    Moyenne : reel ;
Fin;
Tab = tableau [1..50] de Etudiant ;
Var T : tab ;
```

Remarque : des exemples d'application sont donnés en séances de cours.