

Chapitre 6 : Les structures

6.1 Structures simples

Une structure est un type contenant une ou plusieurs variables groupées sous le même nom pour être traitées comme une seule entité :

1. Les variables d'une structure peuvent être de type différents contrairement à un tableau qui est de type homogène
2. Chaque variable est appelée membres ou champs, chaque membre correspond à une donnée
3. Une structure peut contenir n'importe quel type, y compris des tableaux et des structures

6.1.1 Définition

```
struct coord {  
    int x;  
    int y;  
};
```

Le mot clef struct identifie le début de la définition. Le nom de la structure (coord) sera donnée immédiatement après.

Les accolades identifient les limites de la structure. Comme toutes instructions en C, la définition d'une structure se termine par un point virgule après l'accolade fermante.

Chaque membre est défini par son type comme une déclaration standard en C. On ne donne pas de valeur à ces membres dans la définition de la structure.

6.1.2 Initialisation

```
struct coord pointA;
```

On utilise de nouveau le mot clef struct pour initialiser une structure existante. Ce mot clef sera suivi du nom de la structure (coord dans notre cas), puis du nom de la variable qui fera référence à la structure (pointA dans notre cas).

6.1.3 Variante

Il existe une variante pour la définition d'une structure qui permet une initialisation un peu plus simple.

```
typedef struct coord {  
    int x;  
    int y;  
};
```

En utilisant le mot clef `typedef` (que l'on a déjà vu lors de la création de type personnalisé), on simplifie l'initialisation de la structure. En effet, celle ci se déclare alors comme ceci :

```
coord pointA;
```

6.1.4 Utilisation d'une structure

Pour accéder au membre d'une structure, on utilise l'opérateur «.» suivi du nom du membre.

```
pointA.x = 5;  
pointA.y = 7;
```

Le couple `NomStructure.NomMembre` s'utilise comme une variable classique en C.

On peut comparer directement deux structures entre elle. L'opération de comparaison se fera membre par membre dans l'ordre de déclaration de ceux-ci.

L'opérateur «.» est prioritaire à tout les autres opérateurs. On peut donc très bien comparer deux membres de structure entre eux.

6.2 Les structures complexes

Les structures complexes sont des structures intégrant des structures (ou des tableaux) parmi leurs membre. Exemple:

```
struct rectangle {  
    coord hautGauche;  
    coord basDroite;  
};
```

L'initialisation ne change pas d'un structure classique. Pour accéder aux membres des structures membres de la structure complexe, on utilise autant de fois que nécessaire l'opérateur «.» . Dans notre exemple, on fera comme ceci:

```
struct rectangle page; // Définition  
page.hautGauche.x = 7;  
page.hautGauche.y = 6;
```