

C#

Chapitre 4 – Partie 2: Liste

Nous allons ici passer en revue les tableaux, listes et liste de structures.

Les listes :

Ensemble de données qui sont de même type.

Liste = Collections = Tableau dynamique

Taille de la liste n'est pas indispensable lors de la déclaration, elle s'adapte automatiquement au nombre de données à l'intérieur

Chaque case du tableau possède un numéro => indice.

Ce numéro commence de 0 jusqu'à la taille du tableau -1.

Déclaration de la liste :

```
List<typeVariable> <nomListe>;
```

```
Ex : List<int> lesEntrees ;
```

Création de la liste :

```
<nomListe> = new List<typeVariable>();
```

```
Ex : lesEntrees = new List<typeVariable>();
```

Déclaration et création d'un tableau en une seule instruction :

```
List<typeVariable> <nomListe> = new List<typeVariable>() ;
```

Ajouter une valeur :

```
<nomListe>.Add();
```

Afficher la capacité d'une liste :

```
<nomListe>.Capacity
```

Parcourir un tableau :

2 options sont disponibles pour parcourir un tableau, les boucles FOR et FOREACH :

La boucle FOR nécessite de connaître un minimum la taille du tableau et s'écrit de la manière suivante :

```
for (ind = 0; ind < <nomDuTableau>.Count; ind++)  
{  
    Console.WriteLine( <nomListe[ind]> );  
}
```

On définit une variable (ici ind) qui nous servira à boucler. Elle est initialisée et déclarée dans la boucle FOR, on définit qu'elle sera strictement inférieure à la taille du tableau, et sera automatiquement incrémentée de 1 en 1.

- La boucle FOREACH qui elle ne nécessite que l'initialisation d'une variable. Elle parcourra toutes les valeurs du tableau et s'écrit de la manière suivante :

```
Foreach ( int nb in <nomListe>)
```

```
{  
    Console.WriteLine(nb)  
}
```

On définit une variable (ici nb) qui nous servira à boucler, elle nécessite seulement d'être initialisée dans la boucle.

Nombre d'éléments dans une liste :

```
<nomListe>.Count
```

Ajout de valeur :

```
<nomListe>.Add(<valeurAjoutée>) ;
```

→ Se rajoute à la suite des autres valeurs

Modification de valeur :

```
<nomListe>[<indiceVise>] = <valeurAjoutée>
```

→ Permet de modifier une valeur seulement si une valeur est déjà existante dans l'indice ciblé

Affichage d'une valeur :

```
Console.WriteLine(<nomListe>[indiceVisé]);
```

D'autres méthodes utiles :

`<nomListe>.Sum()` Somme des éléments enregistré dans la liste

`<nomListe>.Max()` permet d'obtenir la plus grande valeur

`<nomListe>.Min()` permet d'obtenir la plus petite valeur

`<nomListe>.Count(ele → ele > 200 )` nombre d'élément dans la liste supérieur à 200)

