

Tableaux et collections - C#

Programmation en C#

Tableaux

Collection d'éléments

Éléments de même type (de même ancêtre)

Accès indicé

Tableaux statiques – Taille et contenu fixés à la compilation

```
type [] identifiant = {énumération des valeurs,...};
```

```
class Program
{
    public static void Main(string[] args)
    {
        //déclaration, initialisation
        double [] valeurs = {0.1, 2.5, 0.6, 0.8, 1.6};

        //accès indicé
        Console.WriteLine("Premiere valeur : " + valeurs[0].ToString());
        Console.WriteLine("Dernière valeur : " + valeurs[4].ToString());

        //taille du tableau
        Console.WriteLine("Taille du tableau : " + valeurs.Length.ToString());

        Console.Write("Press any key to continue . . . ");
        Console.ReadKey(true);
    }
}
```

Premier indice : 0



Outils spécifiques pour la manipulation des tableaux

`Length` indique le nombre d'éléments

`foreach` est une boucle spécifique pour les tableaux (collections)

```
class Program
{
    public static void Main(string[] args)
    {
        //déclaration, initialisation
        double [] valeurs = {0.1, 2.5, 0.6, 0.8, 1.6};

        ..

        //énumération des valeurs
        Console.WriteLine("Toutes les valeurs :");
        foreach(double v in valeurs){
            Console.WriteLine(v);
        }

        //classiquement avec un accès indicé|
        Console.WriteLine("Toutes les valeurs encore:");
        for (int i = 0; i < valeurs.Length; i++){
            Console.WriteLine(valeurs[i]);
        }

        Console.Write("Press any key to continue . . . ");
        Console.ReadKey(true);
    }
}
```

Type de l'élément

Forcément dans l'ordre des valeurs
Uniquement pour la lecture

On a le choix du sens du parcours (ascendant ou descendant)
Peut servir pour l'écriture

Tableaux dynamiques – Taille et valeurs définies à l'exécution

Déclaration : `type_de_donnée [] nom_de_variable;`

Initialisation : `nom_de_variable = new type_de_donnée [taille];`

Déclaration et initialisation

(Note : On peut redimensionner le tableau après coup avec la méthode `Resize()` de la classe `Array`)

Saisie : forcément accès indicé

Lecture : on peut utiliser `foreach`

Il n'est pas nécessaire de détruire le tableau, le « garbage collector » s'en charge !

```
class Program
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Taille du tableau : ");
        int n = int.Parse(Console.ReadLine());

        //déclaration + initialisation
        double [] tab = new double[n];

        //saisie
        for (int i = 0; i < tab.Length; i++) {
            tab[i] = double.Parse(Console.ReadLine());
        }

        //affichage
        Console.WriteLine("Les valeurs : ");
        foreach (double v in tab) {
            Console.WriteLine(v);
        }

        Console.WriteLine("Press any key to continue . . . ");
        Console.ReadKey(true);
    }
}
```

Les objets – Classes en C#

Voir les slides sur les classes !
(jusqu'à encapsulation)

Tableaux d'objets

Toujours une collection d'éléments de même type !

```
namespace TableauVoiture
{
    public class Voiture
    {
        private string marque;
        private string modele;

        public Voiture()
        {
        }

        public void saisie()
        {
            this.marque = Console.ReadLine();
            this.modele = Console.ReadLine();
        }

        public void affichage()
        {
            Console.WriteLine(this.marque + ", " + this.modele);
        }
    }
}
```

Classe Voiture

Création de deux temps :

- (1) La structure tableau
- (2) Chaque élément (objet) du tableau

foreach marche aussi pour les objets

```
namespace TableauVoiture
{
    class Program
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Nombre de voitures : ");
            int n = int.Parse(Console.ReadLine());

            //création du tableau (conteneur)
            Voiture [] tab = new Voiture[n];

            //création de chaque voiture et saisie
            for (int i = 0; i < tab.Length; i++)
            {
                //création de l'objet n°i
                tab[i] = new Voiture();
                //saisie des paramètres de l'objet
                tab[i].saisie();
            }

            //affichage de chaque voiture
            foreach (Voiture v in tab) {
                v.affichage();
            }

            Console.WriteLine("Press any key to continue . . . ");
            Console.ReadKey(true);
        }
    }
}
```

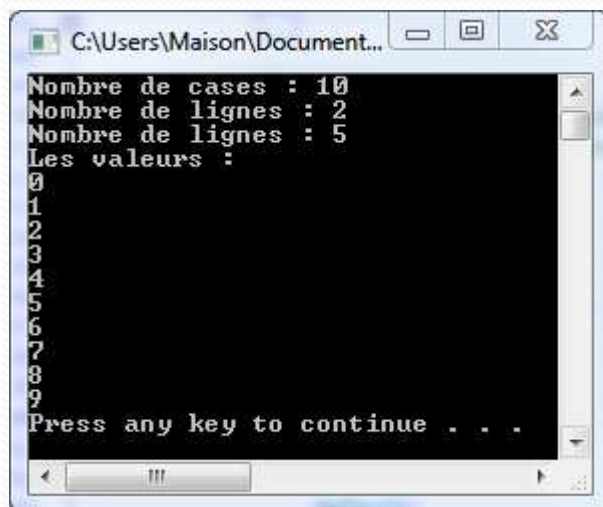
« Programme principal »

Tableaux à 2 dimensions (ou plus)

Tableau de 2 lignes et 5 colonnes

Les indices aux extrémités
du tableau

Remplissage ligne par ligne
(pour chaque ligne, remplir
toutes les colonnes, etc.)



```
C:\Users\Maison\Document...
Nombre de cases : 10
Nombre de lignes : 2
Nombre de lignes : 5
Les valeurs :
0
1
2
3
4
5
6
7
8
9
Press any key to continue . . .
```

```
class Program
{
    public static void Main(string[] args)
    {
        //deux dimensions
        double [,] tab = new double[2,5];

        //première case
        tab[0,0] = 1.0;

        //dernière case
        tab[1,4] = 2.0;

        //nombre total de cases
        Console.WriteLine("Nombre de cases : " + tab.Length.ToString());

        //nombre de lignes
        Console.WriteLine("Nombre de lignes : " + tab.GetLength(0).ToString());

        //nombre de colonnes
        Console.WriteLine("Nombre de colonnes : " + tab.GetLength(1).ToString());

        //parcours en 2 dim - remplissage
        for (int i = 0; i < tab.GetLength(0); i++) {
            for (int j = 0; j < tab.GetLength(1); j++) {
                tab[i,j] = i * tab.GetLength(1) + j;
            }
        }

        //parcours simple
        Console.WriteLine("Les valeurs : ");
        foreach (double v in tab) {
            Console.WriteLine(v.ToString());
        }

        Console.Write("Press any key to continue . . . ");
        Console.ReadKey(true);
    }
}
```

Utilisation de
foreach pour le
parcours



FIN...

Les mêmes concepts sont - à peu de choses près - présents dans tous les langages de programmation objet (Java, Delphi, C++,...)