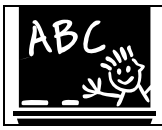
Cours n° 5 : Les tableaux à une dimension



Objectifs :.Apprentissage des techniques de manipulation des tableaux à une dimension

Difficultés : .Aucune difficulté majeure, juste de la pratique. ☺☺

C# comme tout langage de programmation, permet de stocker des variables dans des tableaux. Un tableau est composé d'un nombre déterminé de variables de **même type** (primitif ou objet).

Un tableau est déclaré à partir d'un type de données, par exemple :

```
String [] tableaudechaines // tableaudechaines est donc un tableau contenant des chaînes de caractères
Int [] tableaudentier // tableaudentier est donc un tableau d'entier
```

Les crochets permettent de spécifier qu'il s'agit d'un tableau. Toutefois à ce niveau, les tableaux sont déclarés, mais non créés. En fait ils n'existent pas concrètement en mémoire, vous venez de définir une entité virtuelle. Pour pouvoir les manipuler il faut donc les construire :

```
tableaudechaines= new String [100] ;
tableaudentier= new int[6252];
```

tableaudechaine est maintenant défini et manipulable, comme un tableau de 100 cellules pouvant contenir une chaîne de caractère.

La taille du tableau est définie au moment de sa création, et ne peut plus être changée par la suite. Si on manque de place dans un tableau, il faut donc obligatoirement en créer un nouveau plus grand.

Une fois crée, le tableau est vide, les cellules sont initialisée à 0 pour les tableaux de valeurs numériques, à false pour les booléens, et à null pour les tableaux d'objets.

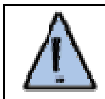
Pour résumer :
Type [] nomtableau ; // Définition du tableau
nomtableau= new Type [taille] ; // Création du tableau
nomtableau[numerocellule]=valeurdetype // Affectation d'une valeur

L'accès aux cellules se fait en spécifiant un nombre entier (de type byte, char, short, ou int) **indexé à partir de 0**.

L'index ne peut pas être de type long, c'est une vérification faite à la compilation.

Exemples divers :

```
tableaudentiers[0]=24; // on place 24 dans la première cellule.
int i=8;
tableaudentiers[1]=i ; // Maintenant  la deuxième cellule à reçu 8.
i=tableaudentier[0] ; // Maintenant i est à 24.
tableaudechaine[6]="coucou" ; // Là je pense que vous avez compris !?
```



Remarque importante : Les tableaux sont alloués dynamiquement. Leurs tailles peuvent donc être le résultat d'une expression. Exemple :

```
personne [] famille;
famille=new persone[2+nombreenfants];.
```

Exemple d'utilisation d'un tableau en C

Ecrire un programme qui crée un tableau comportant les valeurs des carrés des n premiers nombres impairs, la valeur de n étant lues au clavier et qui affiche les valeurs sous la forme suivante :

Combien de valeurs : 5
1 a pour carre 1
3 a pour carre 9
5 a pour carre 25
7 a pour carre 49
9 a pour carre 81

Fichier "tp51.cs"

```
using System;

public class tp51
{
  static void Main(string[] args)
  {
      int[] car;
    int n ;
      Console.WriteLine("combien de valeurs : ");
      n = System.Convert.ToInt32(System.Console.ReadLine());
      car = new int[n] ;
    for (int i=0 ; i<n ; i++)
    car[i] = (2*i+1)*(2*i+1) ;
    for (int i=0 ; i<n ; i++)
      Console.WriteLine((2*i+1) + " a pour carre " + car[i]);
  }
}
```

Exercices applicatifs

Exercice 1 :

Ecrire un programme (n'oubliez pas qu'il est plus facile de commencer par un algorithme) qui :

Lit dans un tableau 5 valeurs fournies au clavier
En calcule et en affiche la moyenne, la plus grande et la plus petite valeur

Exercice 2 :

Supposons que nous souhaitions déterminer, à partir de 10 notes d'élèves (fournies en données), combien d'entre elles sont supérieures à la moyenne de la classe. S'il ne s'agissait que de calculer simplement la moyenne de ces notes, il nous suffirait d'en calculer la somme, en les cumulant dans une variable, au fur et à mesure de leur lecture. Mais, ici, il nous faut à nouveau pouvoir consulter les notes pour déterminer combien d'entre elles sont supérieures à la moyenne ainsi obtenue. Il est donc nécessaire de pouvoir "mémoriser" ces 10 notes. Pour ce faire, il paraît peu raisonnable de prévoir vingt variables scalaires différentes (méthode qui, de toute manière, serait difficilement transposable à un nombre important de notes).Le tableau va nous offrir une solution convenable à ce problème

Exercice 3 :

Voici un algorithme :

```
Programme Fairesomme
Var
        ValeurLue : Nombre
        Lasomme : Nombre

Début
        Lasomme =0

        Ecrire(ecran, " Entrez une série de valeurs")
        Lire(clavier, Valeurlue)
Tant que ValeurLue<>0 Faire
        Lasomme=LaSomme+ValeurLue
        lire(clavier, ValeurLue)
Fin tanque

Afficher " La somme des valeurs saisie est ", Lasomme
fin
```

Posons maintenant comme postulat que l'on connaît le nombre maxi de nombre saisis (5). Modifiez cet algorithme et produisez le programme correspondant en employant un tableau.

Exercice 4:

On demande à l'utilisateur de saisir une série de nombres entiers positifs ou négatifs. A chaque saisie on place la valeur lue dans une case d'un tableau. La saisie s'arrête lorsque l'utilisateur tape 0 ou lorsque le tableau est plein.
Lorsque la saisie est terminée, le programme affiche le contenu du tableau.

Exercice 5

Les Palindromes : On veut afficher à l'écran si un mot saisi est un palindrome ou non. La définition d'un palindrome est un mot qui peut se lire de manière identique dans un sens ou dans l'autre, exemple : radar, elle, anna…..