

TP Informatique 4 – 8h

Les tableaux

But :

Mise en oeuvre des tableaux en C#

Boucle for

Listbox

Traitements un peu plus complexes

Tableaux

Un tableau est une structure de données qui contient un certain nombre de variables du même type. Les tableaux sont déclarés avec un type :

```
type[] arrayName;
```

En fait un tableau en C# est un objet, comme String. Mais, on peut aussi, s'en servir de façon très similaire à C++.

Un tableau possède les propriétés suivantes :

- Un tableau peut être Unidimensionnel, Multidimensionnel ou En escalier.
- La valeur par défaut des éléments de tableau numériques est égale à zéro, et les éléments de référence ont la valeur null.
- Un tableau en escalier est un tableau de tableaux (non abordé ici).
- Les tableaux sont indexés à partir de zéro : un tableau avec n éléments est indexé de 0 à $n-1$.
- Les éléments du tableau peuvent être de n'importe quel autre type, y compris un type tableau (cas des tableaux en escalier).
- Les types de tableaux sont des types référence dérivés du type de base abstrait Array.

Exemples :

```
int[] tabEntiers; // déclaration d'un tableau d'int sans le créer
String[] tabMots; // déclaration d'un tableau de string sans le créer
tabEntiers=new int[20]; // creation tableau de 20 entiers
tabMots = new String[100]; // creation tableau de 100 string

int[] tabTempo = new int[20]; // déclaration et création tableau de 20 int

String[,] grilleMots = new String[10, 20]; // un tableau 2 dimensions : 10
//lignes , 20 colonnes
int[, , ] tabInt3D = new int[10, 10, 3]; // un tableau 3 dimensions
```

Utilisation :

```
tabMots[0] = "coucou"; //élément n° 0
tabInt3D[2, 2, 1] = 55; // élément n° 2,2,1

tabMots[100] = "salut"; // ERREUR à l'exécution, dépasse le tableau ...
                        //(pensez au try catch!!)
```

Valeurs pré définies :

```
int[] tabValeurs = new int[5] { 0, 1, 2, 3, 4 } ; //tableau avec valeurs
//prédéfinies
String[] tab2 = new String[4] { "je" , "tu", "il", "elle" } ;
```

Boucle for(...)

On avait vu la répétitive avec while(). Le for est une instruction plus complexe :

For (initialisation ; test condition sortie ; action incrémentation)

Exemple :

```
for (int i = 0; i < 10; i++) // répète pour I allant de 0 à 9
{
    tabValeurs[i] = 0 ;
}
```

Equivalent en while

```
while (i < 10) // condition sortie
{
    tabValeurs[i] = 0;
    i++; // action incrémentation
}
```

Le for est bien pratique pour balayer un tableau, on l'utilise souvent avec une variable d'indice i, j ou k .

On peut aussi balayer depuis la fin :

```
for (int i = 9; i >= 0; i--) // répète pour I allant de 9 à 0
{
    tabValeurs[i] = 0 ;
}
```

ListBox

C'est un composant graphique qui permet d'afficher plusieurs lignes de texte.

```
listBox1.Items.Add("coucou"); // ajoute une ligne de texte
listBox1.Items.Add("salut");

listBoxAffichage.Items.Clear(); // efface tout
```

la propriété Items est en fait un tableau de String :

```
listBoxAffichage.Items.Count; // le nombre de chaînes actuel
s=listBoxAffichage.Items[2] //on met la deuxième ligne dans s
```

Manipulation :

Ecrire une application windows , qui mette en œuvre des traitements simples sur tableau.

On donne le début du programme.

Une fois crée l'application sous Visual Studio, on ajoute

Une **constante** *taille* qui vaudra 20 : de cette façon en changeant la valeur de cette constante, on change le nombre d'élément du tableau à travers tout le programme.

Un tableau *tabValeurs* , qui déclaré dans Form1, mais au dessus des méthodes, sera vue par toutes les méthodes.

! ne pas créer une autre variable tabVleur en local dans une méthode !

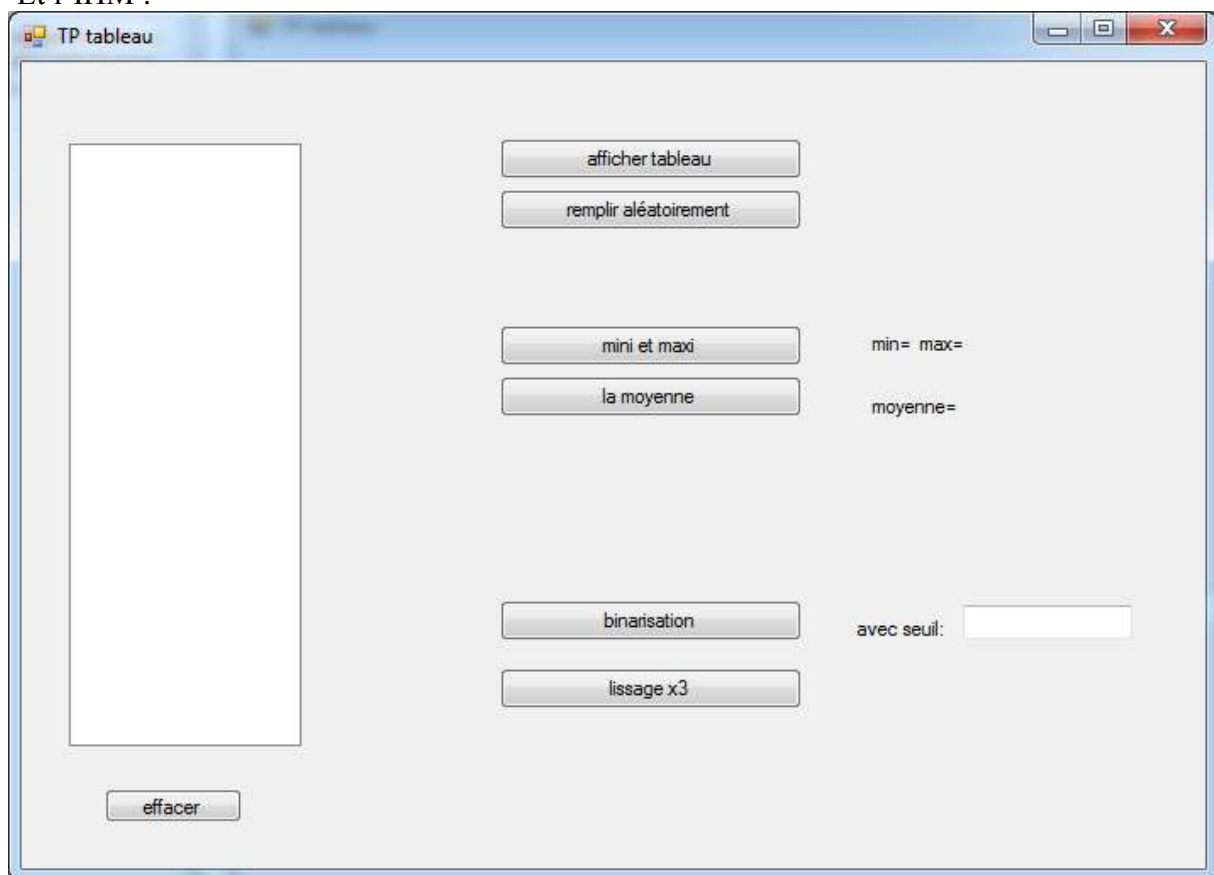
```

public partial class Form1 : Form    // on écrit une classe Form1
{
    public const int taille=20; // taille du tableau
    int[] tabValeurs;    // le tableau vu par toutes les méthodes
//onclick()

    public Form1()    // la méthode qui porte le nom de la classe est
//le constructeur : appelé au démarrage
    {
        InitializeComponent();
        tabValeurs = new int[taille] { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
10, 11, 12, 13, 14, 15, 16, 17, 18, 19 }; // un tableau avec valeur pré
définies
    }
}

```

Et l'IHM :



Compléter :

- 1°) On veut pouvoir afficher tabVal dans la ListBox sur le click bouton « afficher ». Utiliser une boucle For.
- 2°) on veut remplir le tableau de *taille* valeur aléatoire comprise entre 0 et 99 en cliquant sur le bouton adéquat. Une recherche sur le WEB donne cet exemple de génération de nombre aléatoire :

The following code snippet shows how to generate a random number between a range in C#, where min and max are minimum and maximum range of the new number.

```
Random random = new Random(); // create random object
int value = random.Next(min, max); // generate random number between min and max
```

3°) Ecrire le code qui cherche et affiche les valeurs mini et maxi.

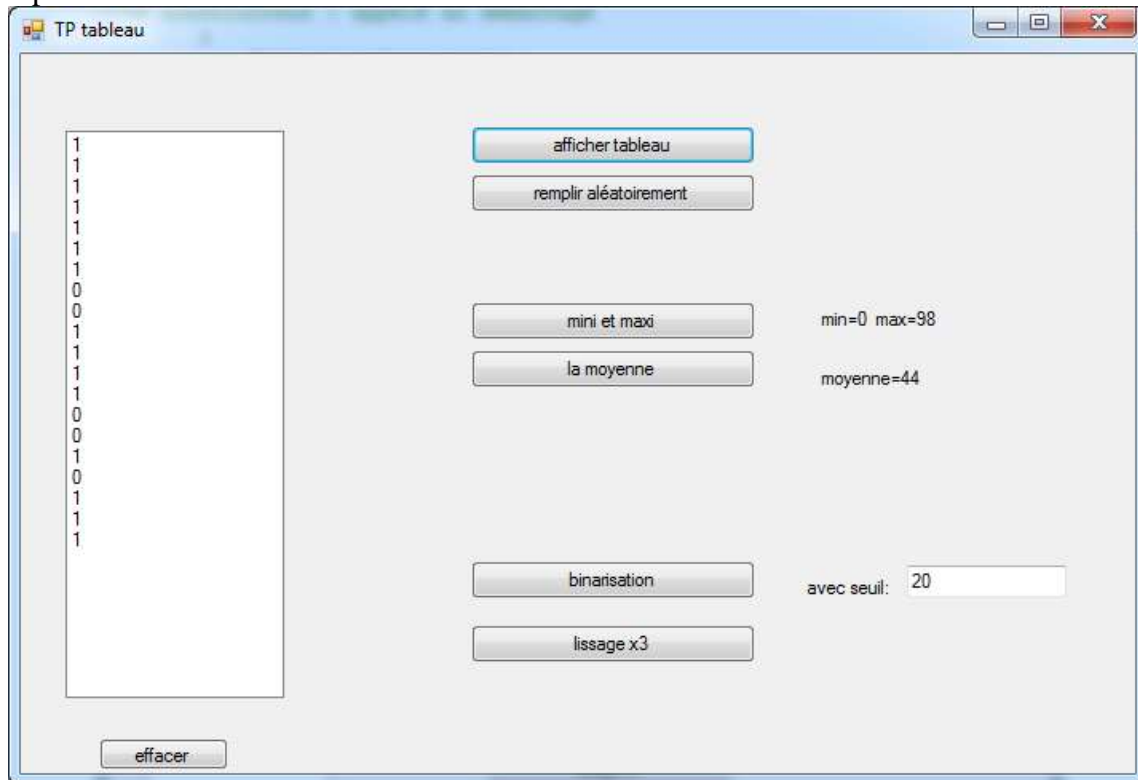
4°) Ecrire le code qui affiche la valeur moyenne.

5°) La binarisation est une opération de flitrage sur le tableau qui consiste à tester chaque élément du tableau par rapport à une valeur de seuil. Si cet élément est supérieur au seuil, alors il prend la valeur 1 sinon il prend la valeur 0. Le tableau ne contient en sortie de ce traitement que des 0/1 (ou 0/ 255 ou 0/ 65535)

Sur une image, cela consiste en une transformation en image noir/blanc.

The screenshot shows a Java Swing window titled "TP tableau". On the left, there is a vertical list of numbers: 78, 54, 40, 23, 74, 30, 56, 4, 20, 31, 38, 97, 77, 0, 10, 58, 16, 30, 60, 98. To the right of the list are several buttons: "afficher tableau", "remplir aléatoirement", "mini et maxi", "la moyenne" (which is highlighted with a blue border), "binarisation", "lissage x3", and "effacer" at the bottom left. To the right of the "mini et maxi" button, the text "min=0 max=98" is displayed. To the right of the "la moyenne" button, the text "moyenne=44" is displayed. To the right of the "binarisation" button, the text "avec seuil:" is followed by an empty text input field.

Après



6°) Lissage 3x3

Le lissage consiste à remplacer chaque $tab[n]$ par une valeur moyenne :

$(tab[n-1] + tab[n] + tab[n+1]) / 3$

Cette opération a tendance à lisser le tableau, à gommer les variations brusques. Cette opération est utilisée en traitement d'images.

Exemple

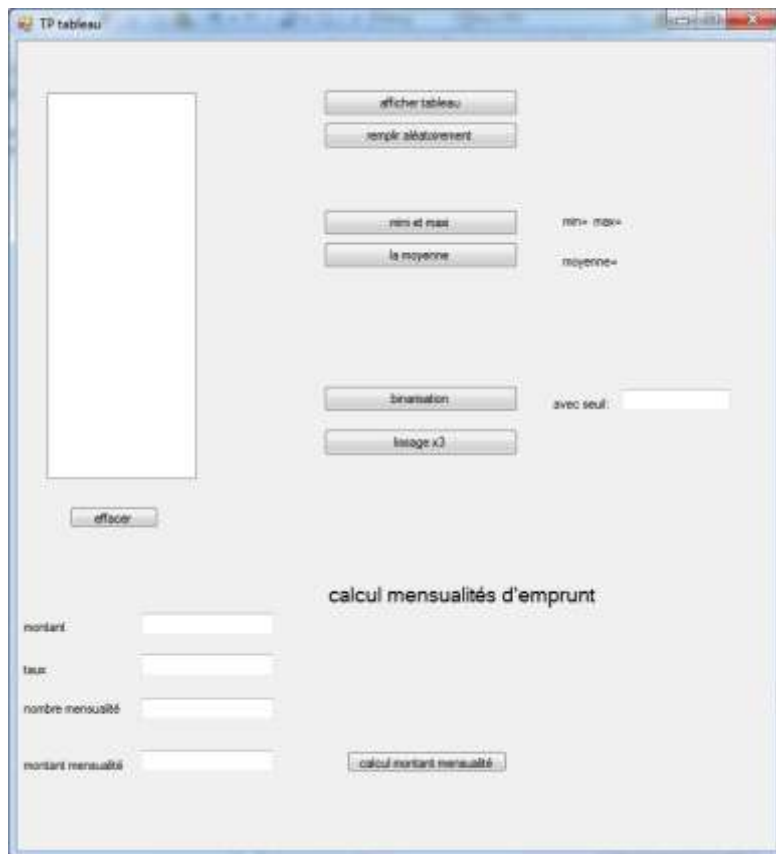
Avant : 10 20 60 20 0 10

Après : ? 30 33 26 10 ?

Pour le cas de la première et de la dernière valeur, on décide de copier le voisin.

D'où : **30** 30 33 26 10 **10**

7°) extra job...



Ecrire le bout de code qui permet de calculer le montant de la mensualité.

Critères :

Respect consignes, autonomie, sauvegarde :3

Nommage : 2

Commentaire : 1

Affichage tableau : 2

Valeurs aléatoires :3

Min et max :2

Moyenne :2

Binarisation :3

Lissage :2