

## Les tableaux à une dimension

Un tableau est utilisé à la place d'une série de variables de même type et de même nature. Un tableau est un objet qui possède des attributs (entre autres, sa longueur).

### 1. Déclaration d'un tableau

La déclaration d'un tableau permet de préciser à l'ordinateur le nom du tableau ainsi que le type des éléments qui seront contenus dans le tableau.

#### Syntaxe en Java :

```
type[] nomTableau;  
    ou  
type nomTableau[];    // les [] indiquent un tableau
```

#### Exemples :

```
double[] tabSalaires;    // tableau tabSalaires de type double  
    ou  
double tabSalaires[];  
  
int[] nombres;           // tableau nombres de type int  
    ou  
_____  
  
char codes[];           // tableau codes de type char  
  
String noms[];          // tableau noms de type String  
    ou  
_____
```

## 2. Création d'un tableau

L'étape suivante est de préciser combien d'éléments maximum contiendra le tableau (allocation de l'espace mémoire) : cette étape est appelée la *création du tableau*.

Un tableau est un objet qui peut être créé soit à l'aide de l'opérateur **new** ou par initialisation lors de la déclaration du tableau.

### 1. Création par new :

Pour allouer de la place en mémoire, on instancie le type qu'on a utilisé pour déclarer le tableau et on indique le nombre maximal d'éléments que doit contenir le tableau.

#### Syntaxe en Java :

```
nomTableau = new type[max];    // max = nombre de places à réserver
```

#### Exemples :

```
// tableau numeros contenant MAX nombres entiers
final int MAX = 75;
int[] numeros;           // déclaration du tableau
numeros = new int[MAX];  // création du tableau

// tableau tabSalaires contenant 6 nombres réels
double tabSalaires[];    // déclaration du tableau
tabSalaires = new double[6]; // création du tableau

// même exemple mais écrit en une seule ligne
double tabSalaires[] = new double[6];

// tableau codes contenant 10 caractères
char codes[] = new char[10];

// tableau noms contenant 12 chaînes de caractères
String noms[] = new String[12];
```

#### Exercices : Déclarez et créez les tableaux demandés

```
// tableau destiné à contenir les 365 températures de l'année
_____

// tableau destiné à contenir les adresses de 10 employés
_____

// tableau destiné à contenir les lettres de l'alphabet
_____
```

**Remarques importantes :**

- ☞ La mémoire peut être réservée pour plusieurs tableaux de même type à l'aide d'une seule déclaration :

```
String tabJours[] = new String[7],
      tabMois[]   = new String[12];
```

- ☞ Quand les tableaux sont créés (qu'on a réservé l'espace en mémoire), les éléments des tableaux sont automatiquement initialisés à 0 pour les tableaux dont le type est numérique primitif (char, int, byte, short, long, float, double), à **false** pour le type boolean ou à **null** pour les types non-primitifs (comme String).
- ☞ Une fois le tableau créé, on peut connaître le nombre d'éléments du tableau à l'aide de la variable d'instance **length**. Par exemple :

```
// Ceci va afficher Le nombre de salaires est de 6
System.out.println("Le nombre de salaires est de " +
                  tabSalaires.length);
```

## 2. **Création par initialisation du tableau lors de sa déclaration :**

On utilise une liste de valeurs séparées par une virgule et mises entre accolades en respectant le type du tableau. On peut laisser un espace après chaque virgule mais ce n'est pas obligatoire.

**Exemples :**

```
// tableau de 5 entiers multiples de 10
int multiplesDix[] = {10, 20, 30, 40, 50};

// ici on n'a pas mis d'espaces après les virgules !
char voyelles[] = {'A', 'E', 'I', 'O', 'U', 'Y'};

// tableau saisons contenant les noms des 4 saisons
String saisons[] = {"printemps", "été", "automne", "hiver"};
```

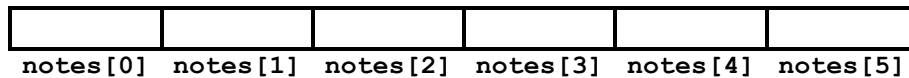
### 3. Accès aux éléments d'un tableau

Chaque élément d'un tableau est accédé par l'expression `nomTableau[indice]` où `indice` est une valeur entière (constante, variable, expression de type `int`) indiquant la position de l'élément dans le tableau. En Java, l'indice varie entre **0** et **max - 1** (max étant le nombre de places réservées pour le tableau).

#### Exemples :

```
double[] notes = new double[6];
```

En mémoire, on aura les 6 cases suivantes dont les indices varieront de **0** à **5** :



`notes[0]` représente le 1<sup>er</sup> élément du tableau

`notes[5]` représente le 6<sup>ième</sup> élément du tableau

#### Note :

Lors de l'exécution, l'exception (erreur) **ArrayIndexOutOfBoundsException** est générée si la valeur de l'indice est plus grande que la limite du tableau ou inférieure à 0.

### 4. Utilisation d'un tableau

Chaque élément d'un tableau est considéré comme une variable. Une fois le tableau déclaré et l'espace mémoire alloué, on peut effectuer les mêmes opérations sur un élément du tableau que sur une variable, en y accédant via le nom du tableau.

#### Exemples :

```
// déclarer et créer un tableau de 15 notes
double notes[] = new double[15];

// affecter la valeur 75 à la première note
notes[0] = 75;

// lecture au clavier des 15 notes
for (int i = 0; i < notes.length; i++)
    notes[i] = Double.parseDouble(JOptionPane.showInputDialog(
        "Entrez la note de l'étudiant " + (i + 1)));

// afficher la deuxième note si elle dépasse 60
if (notes[1] > 60)
    System.out.println(notes[1] + " dépasse 60");
```

**Exercices 1 :**

Référez-vous aux tableaux créés précédemment

Instruction	Explication
	Met à <b>50000</b> le 3 <sup>ième</sup> salaire du tableau <i>tabSalaires</i>
<code>int longueur =</code>	Retourne la longueur du tableau <i>noms</i> (on s'attend à obtenir une longueur de 12)
	Initialise le dernier caractère du tableau <i>codes</i> à <b>'m'</b>
	Initialise le 2 <sup>ième</sup> nom du tableau <i>noms</i> à <b>Bob</b>

**Exercices 2 :**

Instruction	Explication
	Déclare et crée un tableau de 50 nombres entiers nommé <i>nbPoints</i>
	Affiche le 4 <sup>ième</sup> élément du tableau <i>nbPoints</i>
	Additionne <b>4</b> au premier élément du tableau <i>nbPoints</i>
	Met la lettre <b>'S'</b> dans le dernier élément d'un tableau nommé <i>lettres</i> (on ne connaît pas le nombre d'éléments du tableau)

**Exercices 3 :**

Instruction	Explication
	Déclare et initialise un tableau nommé <i>reponses</i> avec les valeurs <b>3, 9, 2, 6</b> et <b>8</b>
	Déclare et initialise un tableau nommé <i>jours</i> avec les valeurs <b>DIM, LUN, MAR, MER, JEU, VEN</b> et <b>SAM</b>
	Déclare et initialise un tableau nommé <i>enPanne</i> avec les valeurs <b>true, false</b> et <b>false</b>
	Déclare et initialise un tableau nommé <i>pourboires</i> avec les valeurs <b>2.50, 8.75, 3.00, 0.75</b> et <b>10.00</b>

**Exercice 4 :** Pour chaque programme, indiquez ce qui sera affiché sur la console

	affichage à l'exécution
<pre> public class Exercice1A {     public static void main(String args[])     {         int tab[] = new int[5];          tab[0] = 10;          for (int k = 1; k &lt; tab.length; k++)             tab[k] = 2 * tab[k - 1];          System.out.print("Le dernier vaut " +             tab[4]);     } } </pre>	
<pre> public class Exercice1B {     public static void main(String args[])     {         int tab[] = new int[10];          int k = 2;          tab[0] = 10;         tab[1] = 2;          for (int j = 1; j &lt; 5; j++)         {             tab[k] = tab[k - 1] / 2;             tab[k + 1] = tab[k] * (k + 1);             k = k + 2;         }          System.out.print("****" +             tab[tab[2]] + "****" + tab[9]);     } } </pre>	

**Exemple complet :**

Voici un exemple de programme qui permet de lire 10 salaires au clavier et de les mémoriser dans un tableau. Le programme calcule ensuite le salaire moyen, le nombre de salaires au dessus de 30000 et le nombre de salaires de moins que 20000.

```
import javax.swing.*;
public class ExempleTableau
{
    public static void main(String args[])
    {
        final int NB_SAL = 10;

        int nbPlus = 0,           // compteur des salaires de plus de 30000
            nbMoins = 0,         // compteur des salaires de moins que 20000
            totalSalaire = 0,    // accumulateur des salaires
            moySalaire;          // le salaire moyen
        int tabSalaires[] = new int[NB_SAL];    // le tableau des salaires

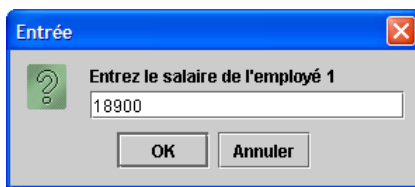
        // lecture des salaires
        for (int i = 0; i < tabSalaires.length; i++)
            tabSalaires[i] = Integer.parseInt(JOptionPane.showInputDialog(
                "Entrez le salaire de l'employé " + (i + 1)));
        // calcul des statistiques
        for (int i = 0; i < tabSalaires.length; i++)
        {
            totalSalaire += tabSalaires[i];
            if (tabSalaires[i] > 30000)
                nbPlus++;
            else if (tabSalaires[i] < 20000)
                nbMoins++;
        }
        moySalaire = totalSalaire / NB_SAL;

        // affichage des résultats
        System.out.println("*****Résultats");
        System.out.print("Salaires : " + tabSalaires[0]);

        for (int i = 1; i < tabSalaires.length; i++)
            System.out.print(", " + tabSalaires[i]);

        System.out.println("\n" + nbPlus + " salaires au dessus de 30000");
        System.out.println("\n" + nbMoins + " salaires de moins que 20000");
        System.out.println("\nSalaire moyen : " + moySalaire);
        System.exit(0);
    }
}
```

Ici `tabSalaires.length` vaut la même chose que `NB_SAL`, on pourrait utiliser l'un à la place de l'autre



```
Console [C:\Program Files\Java\j2re1.4.2\bin\javaw.exe (06-09-18 15:31)]
*****Résultats
Salaires : 18900, 30000, 25761, 63581, 24589, 15000, 34000, 27891, 55555, 26100
3 salaires au dessus de 30000
2 salaires de moins que 20000
Salaire moyen : 32137
```