

1 Les tableaux

1.1 Principe de base

Un tableau est un ensemble de valeurs de même type, occupant des emplacements contigus en mémoire, désigné par une variable.

On peut définir un tableau de taille 10 en écrivant :

```
int [ ] unTableau = new int[10];
```

--	--	--	--	--	--	--	--	--	--

C'est l'**opérateur new** qui doit être utilisé pour allouer le tableau.

Le premier indice est 0. Chaque élément est valorisé en utilisant un indice :

```
unTableau[5] = 9;
```

Toutefois, on peut initialiser un tableau à sa création sans utiliser l'opérateur **new** :

```
int[] autreTableau = { 12, 15, 5, 78, 5 }; // ce tableau contient 5 éléments
```

--	--	--	--	--

A l'exécution, Java opère une vérification de taille ; le programme s'arrêtera si l'on tente d'accéder à un indice hors tableau.

On peut affecter une variable tableau à une autre variable tableau :

```
int[] encoreUnTableau = autreTableau;
```

Mais attention dans ce cas les deux tableaux font référence au même espace mémoire.

Exercice 1

Une application Java permet de gérer des produits de papeterie. Un tableau contient les noms des produits :

Crayon	Gomme	Stylo	Equerre	Règle
--------	-------	-------	---------	-------

indice : 0

1

2

3

4

// Code Java

// tableau de String « noms » qui contient la liste des noms des produits :

```
String[] noms = {"Crayon", "Gomme", "Stylo", "Equerre", "Règle"};
```

// Travail à faire : 1.1 - afficher le premier élément du tableau et le dernier

```
. . . .
```

```
. . . .
```

// 1.2 - Affecter la valeur « compas » à la case d'indice 2

```
. . . .
```

1.2 Parcours d'un tableau

Exemple :

```
// Déclaration et instantiation du tableau
int[] tab = { 12, 15, 5, 78, 5 };
```

--	--	--	--	--

```
// longueur du tableau = nombre d'éléments
// obtenue grâce à l'attribut length : . . .
int longueur = tab.length;

// boucle for
for (int i = 0 ; i < longueur ; i++)
{
    // affichage
    System.out.println(tab[i]);
}
```

Exercice 2 - Gestion des produits de papeterie, suite

On dispose des produits suivants (on commence à numéroté à 0 exprès) :

<i>indice (Numéro du produit)</i>	<i>noms (Nom du produit)</i>	<i>prix (Prix du produit)</i>
0	Crayon	1.5
1	Gomme	1
2	Stylo	2
3	Equerre	1.5
4	Règle	1.8

Dans le programme Java on utilise deux tableaux : un pour les noms et l'autre pour les prix.

```
// tableau de String « noms » qui contient la liste des noms des produits :
String[] noms = {"Crayon", "Gomme", "Stylo", "Equerre", "Règle"};
```

```
// tableau de float « prix » qui contient la liste des prix des produits :
float[] prix = { 1.5f, 1f, 2f, 1.5f, 1.8f };
```

```
// Travail à faire : 2.1 - affichez tous les noms des produits
```

```
. . . . .
```

```
// Travail à faire : 2.2 - afficher tous les prix des produits
```

```
. . . . .
```

```
// Travail à faire : 2.3 - trouver le prix du produit le plus cher (= le max des prix) et l'afficher
```

```
// le prix max
```

```
float prixMax = . . . . .
```

```
// parcours du tableau
```

```
. . . . .
```

```
. . . . .
```

```
// Travail à faire : 2.4 - trouver le produit le moins cher ; afficher son prix et son nom
```

```
// le prix min
```

```
float prixMin = . . . . .
```

```
// l'indice du produit le moins cher
```

```
int iMin = . . . . .
```

```
// Travail à faire : 2.5 - afficher le prix moyen des produits
```

```
// somme totale des prix
```

```
float total = . . . . .
```

```
. . . . .
```

Exercice 3 – Recherche dans un tableau (suite du 2)

But : on veut pouvoir trouver un produit et son prix à partir de la saisie de son nom.

Instructions - le programme fait les choses suivantes :

- L'utilisateur saisit un nom de produit,
- Le programme recherche ce nom dans le tableau et il affiche le prix du produit correspondant.
- Si ce produit est inexistant, un message d'erreur est affiché.

Attention ! En Java, pour comparer deux chaînes de caractère, on va utiliser la méthode (fonction) `equals` et non pas l'opérateur `==` :

```
String var1 = "abcdef" ;           // une variable de type String
String var2 = "abcdef" ;           // une autre
if ( var1.equals(var2) )           // dans ce cas, c'est vrai
    { // faire quelque chose     }
```

Travail à faire :

3.1 – Ecrire le programme de recherche d'un produit avec une boucle **for**. La boucle doit s'arrêter si on trouve le produit.

3.2 – Ecrire le programme avec une boucle **while**. La boucle doit s'arrêter si on trouve le produit.

// PROGRAMME main

// tableau de **String** « **noms** » qui contient la liste des noms des produits :

```
String[] noms = {"Crayon", "Gomme", "Stylo", "Equerre", "Règle"};
```

// tableau de **float** « **prix** » qui contient la liste des prix des produits :

```
float[] prix = { 1.5f, 1f, 2f, 1.5f, 1.8f };
```

.