

---

# FICHE D'EXERCICES

## TABLEAUX ET MÉTHODES

### Séance Java 6 (TP)

---

#### Objectifs :

- Comprendre ce qu'est un tableau
- Réaliser des opérations classiques sur les tableaux à une dimension (parcours, recherche)
- Écrire des méthodes selon les spécifications (noms, arguments, type de retour) fournies

### Exercice 1 : Affichage et statistiques d'un tableau

On se propose d'écrire une classe `StatsTableaux` qui permet de calculer des statistiques sur un tableau d'entiers. Vous prendrez soin à **tester dans le main** chacune des méthodes implémentées.

#### 1.1 Affichage du tableau

Ecrivez la méthode `afficheTableau` qui affiche tous les éléments d'un tableau passé en paramètre. Les éléments du tableau devront être affichés sur une même ligne : 1 2 3 4 5 6

```
/** Methode afficheTableau                                1
 * @param t Le tableau                                  2
 */                                                      3
public static void afficheTableau(int [] t)              4
```

**Aide :** Vous pourrez utiliser les méthodes suivantes :

- `System.out.print(x + " ");` affiche la valeur de la variable `x` suivie d'un espace sans aller à la ligne;
- `System.out.println();` retourne à la ligne sans afficher de caractère.

Ecrivez la méthode `afficheTableau2` qui affiche les éléments **non nuls** d'un tableau passé en paramètre.

```
/** Methode afficheTableau2                              1
 * @param t Le tableau                                  2
 */                                                      3
public static void afficheTableau2(int [] t)             4
```

#### 1.2 Somme et moyenne du tableau

Ecrivez la méthode `sommeTableau` qui renvoie la somme des éléments d'un tableau d'entiers passé en paramètre.

```
/** methode sommeTableau qui renvoie la somme de tous les elements du tableau  1
 *                                                                              2
 * @param t Le tableau                                                         3
 * @return La somme des elements du tableau                                    4
 */                                                                              5
public static int sommeTableau(int [ ] tableau)                                6
```

Ecrivez la méthode `moyenneTableau` qui renvoie la valeur moyenne des éléments d'un tableau d'entiers passé en paramètre. Cette méthode devra faire appel à la méthode `sommeTableau`.

---

```

/** methode moyenneTableau calculant la moyenne des elements du tableau      1
*                                                                              2
* @param t Le tableau                                                         3
* @return La moyenne des elements du tableau                                 4
*/                                                                              5
public static double moyenneTableau(int[] t)                                  6

```

### 1.3 Ecart type du tableau

On souhaite calculer l'écart type  $\sigma$  d'un ensemble de valeurs  $\{x_i\}_{1 \leq i \leq n}$ . Cette valeur s'obtient de la manière suivante :

$$\sigma = \sqrt{\frac{1}{n} \sum_{k=1}^n (x_i - \bar{x})^2}$$

où  $\bar{x}$  est la moyenne des  $\{x_i\}_{1 \leq i \leq n}$ .

Ecrivez la méthode `ecartTypeTableau` qui renvoie l'écart type des éléments d'un tableau d'entiers passé en paramètre. Cette méthode fera appel à la méthode `moyenneTableau`. Vous pourrez également faire appel aux méthodes suivantes pour calculer les fonctions puissance et racine carrée : `Math.pow(x,y)` et `Math.sqrt(x)`.

```

/** methode ecartTypeTableau calculant l'ecart type des elements du tableau  1
*                                                                              2
* @param t Le tableau                                                         3
* @return L'ecart type                                                       4
*/                                                                              5
public static double ecartTypeTableau(int[] t)                                6

```

## Exercice 2 : Recherche dans un tableau

On souhaite maintenant effectuer la recherche d'un élément dans un tableau. Plus particulièrement, on souhaite obtenir la position dans le tableau de l'élément ayant une certaine valeur.

### 2.1 Recherche de la position d'un élément

Ecrivez la méthode `rechercheTableau` qui retourne la position d'un entier dans un tableau. La méthode renvoie -1 si l'élément n'est pas dans le tableau. Il est possible que la valeur apparaisse plusieurs fois dans le tableau ; dans ce cas la méthode renverra la position du premier élément en partant de la gauche.

```

/** Methode rechercheTableau                                                 1
* @param t Le tableau                                                         2
* @param v La valeur a chercher                                             3
* @return La position de l'element                                          4
*/                                                                              5
public static int rechercheTableau(int[] t, int v)                            6

```

#### Exemple :

La recherche de la valeur 4 dans le tableau suivant devra retourner 3 :

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 2 | 5 | 1 | 4 | 8 | 3 | 7 |
|---|---|---|---|---|---|---|

---

## 2.2 Le plus grand élément du tableau

Ecrivez la méthode `valMaxTableau` qui renvoie le plus grand élément d'un tableau d'entiers passé en paramètre.

```
/** Methode valMaxTableau                                     1
 * @param t Le tableau                                       2
 * @return La valeur du plus grand element                 3
 */                                                         4
public static int valMaxTableau(int [] t)                  5
```

Ecrivez la méthode `posMaxTableau` qui renvoie la position du plus grand élément d'un tableau d'entiers passé en paramètre.

```
/** Methode posMaxTableau                                    1
 * @param t Le tableau                                       2
 * @return La position du plus grand element               3
 */                                                         4
public static int posMaxTableau(int [] t)                  5
```

## Exercice 3 : Ordre du tableau

### 3.1 Croissant et décroissant

On souhaite tester si un tableau est trié par ordre croissant (resp. décroissant).

Ecrivez la méthode `estCroissant` qui prend en paramètre un tableau et qui retourne vrai quand le tableau contient des valeurs triées par ordre croissant.

```
/** Methode estCroissant                                    1
 * @param t Le tableau                                       2
 * @return true si le tableau est croissant, false sinon  3
 */                                                         4
public static boolean estCroissant(int [] t)               5
```

En vous inspirant de cette méthode, écrivez la méthode `estDecroissant` qui retourne vrai quand le tableau passé en paramètre contient des valeurs triées par ordre décroissant.

### 3.2 Tableau trié ?

A partir des deux méthodes précédentes, écrivez une méthode `estTrie` qui retourne 1 si le tableau donné en paramètre est trié par ordre croissant, -1 s'il est trié par ordre décroissant et 0 sinon (pas trié).

```
/** Methode estTrie                                         1
 * @param t Le tableau                                       2
 * @return 1 si tableau croissant, -1 si decroissant, 0 sinon 3
 */                                                         4
public static int estTrie(int [] t)                        5
```

## Exercice 4 : Bonus : Distribution des valeurs du tableau

On souhaite maintenant pouvoir afficher la distribution des valeurs des éléments dans un tableau. On supposera que les valeurs contenues dans le tableau sont des entiers compris dans l'intervalle  $[1, 6]$ .

---

## 4.1 Affichage numérique

Ecrivez la méthode `afficheDistribution` qui affiche la distribution des valeurs des éléments d'un tableau passé en paramètre.

```
/** Methode afficheDistribution                                1
 * @param t Le tableau                                     2
 */                                                         3
public static void afficheDistribution(int [] t)             4
```

L'appel de la méthode sur le tableau

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 2 | 5 | 1 | 4 | 6 | 4 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|

devra produire l'affichage suivant :

```
1 : 4
2 : 1
3 : 0
4 : 2
5 : 1
6 : 1
```

## 4.2 Diagramme bâton

On veut maintenant obtenir un affichage plus "graphique" de cette distribution en utilisant un diagramme *bâton* horizontal. Par exemple, pour le tableau précédent, on souhaite obtenir un affichage de la forme :

```
1 : ****
2 : *
3 :
4 : **
5 : *
6 : *
```

Ecrivez la méthode `diagrammeBaton` qui affiche le diagramme bâton de la distribution des éléments d'un tableau passé en paramètre.

```
/** Methode diagrammeBaton                                  1
 * @param t Le tableau                                     2
 */                                                         3
public static void diagrammeBaton(int [] t)                 4
```