

DEV1 – Laboratoires Java I**TD 10 – Tableaux**

Dans ce TD vous trouverez une introduction aux tableaux.

Les codes sources et les solutions de ce TD se trouvent à l'adresse :

<https://git.esi-bru.be/dev1/labo-java/tree/master/td10-tableaux>

Table des matières

1	Créer et manipuler des tableaux	2
2	Tableaux et tests	3
3	Exercices récapitulatifs	3

1 Créer et manipuler des tableaux

```
1 package esi.dev1.td10;
2
3 public class Tableau {
4
5
6     public static void main(String[] args) {
7         int[] tab = {1, 2, 3, 4, 5};
8         System.out.println("taille: "+tab.length);
9         System.out.println("1er élément: "+tab[0]);
10        System.out.println("dernier élément: "+tab[tab.length-1]);
11        System.out.println("autre élément: "+tab[2]);
12        System.out.println("affiche un drôle de truc: "+tab);
13
14        afficherTab(tab);
15        incrémenter(tab);
16        afficherTab(tab);
17
18        int[] tab2 = new int[10];
19        afficherTab(tab2);
20    }
21
22    static void afficherTab(int[] tab) {
23        for (int valeur : tab) {
24            System.out.print(valeur+" ");
25        }
26        System.out.println(); // on passe à la ligne
27    }
28
29    static void incrémenter(int[] tab) {
30        for (int i = 0; i < tab.length; i++) {
31            tab[i]++;
32        }
33    }
34 }
```

Tableau.java

Exercice 1 TableauUtil

Dans une classe `TableauUtil` écrivez les méthodes suivantes et leur javadoc :

- ▷ `static double min(double[] tab)` qui retourne le minimum du tableau passé en paramètre ;
- ▷ `static double max(double[] tab)` qui retourne le maximum du tableau ;
- ▷ `static double somme(double[] tab)` qui retourne la somme des éléments du tableau ;
- ▷ `static double moyenne(double[] tab)` qui retourne la moyenne des éléments du tableau ;

Exercice 2 Initialisation par défaut

Dans la méthode principale d'une classe `TestInit` :

- ▷ créer un tableau d'entier de taille 10 et afficher-le (comme à la ligne 18 de la classe `Tableau` ci-dessus).
Quelle valeur par défaut initialise chacune des cases du tableau ?
- ▷ créer un tableau de double de taille 10 et afficher-le ;
- ▷ créer un tableau de 10 booléens et afficher-le ;
- ▷ créer un tableau de 10 `String` et afficher-le ;

2 Tableaux et tests

```
1 package esi.dev1.td10;
2
3 import org.junit.Test;
4 import static org.junit.Assert.*;
5
6 public class TableauTest {
7
8     @Test
9     public void testIncrémenter() {
10         int[] tab = {1, 2, 3};
11         Tableau.incrémenter(tab);
12         int[] tabAttendu = {2, 3, 4};
13         assertEquals(tabAttendu, tab);
14     }
15
16     @Test
17     public void testIncrémenterTableauVide() {
18         int[] tab = {};
19         Tableau.incrémenter(tab);
20         int[] tabAttendu = {};
21         assertEquals(tabAttendu, tab);
22     }
23 }
24 }
```

TableauTest.java

Exercice 3 Tests

Créer des tests JUnit pour les méthodes de l'exercice 1. Pour chacune de ces méthodes testez un cas général ainsi que les cas limites (tableau vide, maximum en début/fin de tableau, valeurs négatives, etc).

3 Exercices récapitulatifs

Exercice 4 TableauUtil (suite)

Dans la classe `TableauUtil` ajoutez les méthodes suivantes, leur javadoc ainsi que les tests JUnit correspondant :

- ▷ `int indiceMax(double[] tab)` : retourne l'indice du maximum du tableau passé en paramètre ;
- ▷ `boolean estTrié(double[] tab)` : retourne vrai si le tableau est trié par ordre croissant et faux sinon ;
- ▷ `int indice(int[] tab, int valeur)` : retourne l'indice de cette valeur dans le tableau. Que retournez-vous si la valeur n'apparaît pas ?
- ▷ `boolean contient(String[] tab, String mot)` : retourne vrai si un tableau de `String` contient un mot passé en paramètre ;

A faire :

- ▷ tableaux de string
- ▷ remplissage de tableau par `System.in`
- ▷ retourner un tableau
- ▷ tableau et `String` (`split`, `chars`)
- ▷ tableau de booléens ?