

TD 1 : Familiarisation avec Java

```
// Fichier Hello.java
public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello World !") ;
    }
}
```

Exercice 1 Introduction

1. Recopiez le programme ci-dessus dans un fichier nommé `Hello.java`. Pour le compiler utilisez la commande `javac Hello.java` et pour l'exécuter, `java Hello`. Peut-on l'exécuter directement sans utiliser la commande `java` ? Que se passe-t-il si le fichier s'appelle `Hello2.java` ?

Exercice 2 Lire la ligne de commande

Lire la ligne de commande s'avère beaucoup plus simple en Java qu'en C. On reçoit comme argument de `main` un seul tableau (habituellement nommé `args`). On obtient le nombre d'éléments du tableau par `args.length`. Les arguments de la ligne de commande seront donc dans

`args[0] ... args[args.length-1]`.

1. Modifiez ce programme pour que `java Hello Toto` produise l'affichage :
2. `Hello Toto !`

Que se passe-t-il si on ne donne pas d'argument sur la ligne de commande ? A l'aide d'une conditionnelle, résolvez ce problème.

3. Modifiez maintenant le programme pour que pour chaque nom donné sur la ligne de commande, il affiche une ligne de salutation.

Exercice 3

1. Écrivez un programme `Somme` effectuant la somme du tableau : `{1, 23, 12, -4, 6, 89, 2}` et affichant le résultat.
2. Définissez une méthode `static`, qui prendra en argument le tableau, qui effectuera le calcul et qui retournera la valeur obtenue. `main` ne sera responsable que de l'appel de la fonction et de l'affichage. Le prototype de la méthode est :
`public static int somme(int[] tab)`

3. Ajoutez la méthode `produit` et la méthode `moyenne` au programme.

Exercice 4 Familiarisation

Essayer de provoquer les erreurs suivantes en modifiant un programme, observez le résultat.

1. Erreurs de compilation :
 - a. Variable non déclarée.
 - b. Mauvais arguments passés lors de l'appel d'une méthode.
2. A l'exécution :
 - a. Pas de fonction `main`.
 - b. Dépassement des limites d'un tableau.
 - c. Division par zéro.

Exercice 5

Cet exercice ne doit pas être fait sur machine !

Qu'affichent les programmes suivants :

```
1. public class Ex1a {
2.     static int f1(int a, int b) {
3.         a++ ; b++ ;
4.         System.out.println(a+" "+b) ;
5.         a=a+b ;
6.         return a ;
7.     }
8.     public static void main(String[] args) {
9.         int a=2, b=3 ;
10.        System.out.println(a+" "+b) ;
11.        f1(b,a) ;
12.        System.out.println(a+" "+b) ;
13.        b = f1(b,a) ;
14.        System.out.println(a+" "+b) ;
15.    }
16. }
17.
18. public class Ex2a {
19.     static void f(int[] a,int i,int k) {
20.         a[i] = k ;
21.     }
22.     public static void printtab(int[] tab) {
23.         for(int i=0;i<tab.length;i++)
24.             System.out.print(tab[i]+" ") ;
25.         System.out.println() ;
26.     }
27.
28.     public static void main(String[] args) {
29.         int[] t = {1,2,3,4,5,6} ;
30.         int[] u ;
31.         int[] v ;
```

```

32.         printtab(t) ;
33.         f(t,2,12) ;
34.         printtab(t) ;
35.         u = t ;
36.         f(u,1,0) ;
37.         printtab(u) ;
38.         printtab(t) ;
39.         v = (int [])t.clone() ;
40.         f(v,0,0) ;
41.         printtab(v) ;
42.         printtab(t) ;
43.
44.     }
45. }
46.

```

Exercice 6

Écrire un programme `Occ.java` lisant tous les arguments passés sur la ligne de commande et recherchant si le premier argument existe dans la suite de la ligne. Par exemple `java Occ aaaa bbb AA cc` affichera :
Non trouve
et `java Occ aaaa bbb AA aaaa cc` affichera :
Trouve

Exercice 7

On appelle palindrome un mot pouvant se lire indifféremment dans les deux sens : par exemple *Laval*, *ici*, le mot vide, etc. Plus formellement, un mot de longueur l est un palindrome si pour tout $i < l$, le caractère à la position i est égal au caractère à la position $l-i-1$. Cette notion peut également s'appliquer à une séquence d'entiers.

1. Écrivez sur papier un algorithme vérifiant si un mot est un palindrome.
2. Écrivez une fonction en Java prenant en argument un tableau d'entier et renvoyant un booléen indiquant si le tableau est un palindrome. Vous écrirez également la méthode statique `main` permettant de tester cette fonction.
3. En utilisant la méthode `charAt`, écrivez une autre fonction prenant cette fois ci une `String` en argument et vérifiant si c'est un palindrome. La chaîne vérifiée sera récupérée par `main` sur la ligne de commande.

Exercice 8

Écrire la classe `Tri.java` contenant une méthode `public static void bubble(int[] t)` implémentant l'algorithme de tri à bulle : Soit l la longueur du tableau t Pour i allant de 0 à l -

1 :

Pour j allant de 1 à $l-i-1$:

si $t[j] < t[j+1]$ permuter les deux valeurs.