



ISN

NOTION DE FONCTION

Une **fonction** est un sous-programme, qui permet d'isoler une instruction, lorsque celle-ci revient plusieurs fois dans un programme.

Cela permet :

- › d'éviter de retaper des morceaux de code lorsque ceux-ci interviennent plusieurs fois ;
- › en cas d'erreur, de corriger une seule fois dans la fonction au lieu d'autant de fois que le code apparaît.

Il existe des fonctions prédéfinies ; par exemple, en Java, **length**, qui renvoie la longueur d'un tableau, **charAt()**, qui extrait un caractère d'une chaîne, **Math.pow()**, qui calcule une puissance, ...

On peut aussi en créer ; par exemple la fonction **arrondi** utilisée dans le TP conversion Celsius-Fahrenheit.

On distingue deux types de fonctions :

- › celles qui ne renvoient pas de valeur (parfois appelées procédure) ; si, par exemple, son but est d'afficher un texte.
- › celles qui renvoient une valeur, comme la fonction **arrondi** citée plus haut.

Une fonction peut contenir des arguments (ou paramètres).

Exemple 1 : fonction qui ne renvoie pas de valeur

Dans l'exemple suivant, on affiche les notes obtenues par Marie et Joshua, lesquelles sont stockées dans 2 tableaux distincts.

```
public class TP_fct {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        double[] notesM = {11,12,16,8,12.5};  
        double[] notesJ = {13,10,17,11};  
        for (int i=0;i<notesM.length;i++) {  
            System.out.println("Note "+i+": "+notesM[i]);  
        }  
        for (int i=0;i<notesJ.length;i++) {  
            System.out.println("Note "+i+": "+notesJ[i]);  
        }  
    }  
}
```

Le bloc encadré en pointillés est répété deux fois afin d'afficher chacun des tableaux. Nous allons donc créer une fonction **AfficheNotes**, qui à partir d'un tableau, affiche tous ses éléments.

```
public class TP_fct {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        double[] notesM = {11,12,16,8,12.5};  
        double[] notesJ = {13,10,17,11};  
        AfficheNotes(notesM);  
        AfficheNotes(notesJ);  
    }  
  
    static void AfficheNotes(double[] tab) {  
        for (int i=0;i<tab.length;i++) {  
            System.out.println("Note "+i+": "+tab[i]);  
        }  
        System.out.println();  
    }  
}
```

La fonction **AfficheNotes** se situe à l'extérieur la classe **main**, mais à l'intérieur de la classe **TP_fct**.

Elle ne renvoie pas de valeur, elle débute donc par **static void**, suivi du nom (**AfficheNotes**), puis des arguments entre parenthèses (**double[] tab**).

Dans le programme principal (**main**), on fait appel à la fonction et on passe en paramètre le tableau que l'on veut afficher (**notesM**, puis **notesJ**).

Ouvrez Eclipse avec l'espace de travail commun (T:Commun/travail/T04S spé ISN/workspace). Créer une nouvelle classe dans votre espace et copier le programme TP_fct. Le tester.

Exemple 2 : fonction qui renvoie une valeur

On veut créer une fonction **MaxNote**, qui à partir d'un tableau, donne la note maximale.

Elle renvoie un résultat réel (le maximum), elle débute donc par **static double**.

A la dernière ligne, on trouve le mot clé **return**, suivi du résultat à obtenir.

```
static double MaxNote(double[] tab) {  
    double max = 0; //initialisation du max  
    for (int i=0;i<tab.length;i++) { //détermination du max  
        if (tab[i]>max) {  
            max = tab[i];  
        }  
    }  
    return max ; //valeur retournée par la fonction  
}
```

- a) Indiquer comment faire appel à cette fonction pour connaître la note maximale de Marie :
.....
- b) Compléter le programme précédent en recopiant au bon endroit la définition de la fonction **MaxNote**.
- c) Rajouter dans le programme principal 2 lignes faisant appel à la fonction et permettant d'obtenir l'affichage suivant :

```

La meilleure note de Marie est : 16.0
La meilleure note de Joshua est : 17.0

```

Exercice d'application : (à finir à la maison)

Le but de cet exercice est de simplifier l'algorithme suivant grâce à une ou des fonctions.

Les variables sont les tableaux de réels notesM et notes J, les entiers i et j et les réels SomM, SomJ, MoyM et MoyJ.

Entrées :

```

notesM ← {11,12,16,8,12.5};
notesJ ← {13,10,17,11};

```

Initialisation:

```

SomM ← 0
SomJ ← 0

```

Traitement :

```

Pour i allant de 0 à (longueur de notesM - 1) faire
  SomM ← SomM + notesM[i]
FinPour
MoyM ← SomM/(longueur de notesM)
Pour j allant de 0 à (longueur de notesJ - 1) faire
  SomJ ← SomJ + notesJ[j]
FinPour
MoyJ ← SomJ/(longueur de notesJ)

```

Sorties :

Afficher les moyennes de Marie et de Joshua

- a) A quoi sert cet algorithme ?
.....
.....
.....
- b) Entourer les morceaux de code qui se répètent dans le traitement.
- c) Compléter le pseudo-code suivant, définissant la fonction **CalculeMoy**, qui à partir d'un tableau de nombres, retourne la moyenne des valeurs contenues dans ce tableau.

```

Fonction CalculeMoy ( ..... ) {
  Som ← 0
  Pour i allant de .. à ..... faire
    Som ← .....
  FinPour
  Retourne .....
}

```

- d) Réécrire la partie initialisation + traitement en utilisant la fonction **CalculeMoy**.

```

.....
.....
.....
.....

```

- e) Traduire la fonction **CalculeMoy** en Java :

```

.....
.....
.....
.....
.....
.....

```

- f) Compléter le programme TP_fct par la définition de la fonction **CalculMoy**, ainsi que par un morceau de code permettant d'afficher deux phrases donnant la moyenne de chaque élève.