

07/10/2025

# Bloc 1 – TP6

## Procédure et fonction Java

### Sommaires

Introduction .....	1
I. Conception .....	1
II. Réalisation .....	2
Exercice 1 - Écrire un programme qui affiche la factorielle d'un nombre .....	2
Exercice 2 - Table de multiplication .....	3
Exercice 3 – Choix entre factorielle ou table de multiplication .....	4
Exercice 4 - Moyenne de la classe et meilleure note .....	5
Exercice 5 : Ecrire une fonction qui permet de retourner la valeur maximale d'un tableau de valeur .....	6
Exercice 7 – Calculer la moyenne et la valeur maximale d'un tableau .....	7
III. Retour d'expérience .....	8
Conclusion .....	8

## Introduction

L'objectif du TP est nous faire apprendre dans le code java des fonction et procédure, entre fonction et valeur. Au point jusqu'à faire des programmes structuré en plusieurs partie utilisable

### I. Conception

Avant de commencer le code, j'ai pris le temps d'analyser chaque exercice pour comprendre ce qu'il fallait faire.

Voici les choses retenu :

- Pour le premier exercice, j'ai pensé à utiliser une **boucle for** pour multiplier les nombres entre 1 et le nombre choisi, afin d'obtenir la factorielle.
- Pour le deuxième exercice, j'ai prévu une **procédure** qui affiche directement la **table de multiplication** du nombre saisi.
- Dans le troisième, j'ai ajouté un **menu** pour laisser le choix à l'utilisateur entre calculer une factorielle ou afficher une table de multiplication.
- Les exercices suivants portaient sur les tableaux : j'ai conçu un programme qui demande plusieurs notes, puis qui calcule la **moyenne** et recherche la **meilleure note**.
- Enfin, les derniers exercices reprennent ces fonctions pour les combiner dans un même programme.

J'ai choisi de travailler avec des **tableaux de type double** pour stocker les notes et d'utiliser des **boucles for** pour parcourir les éléments. La conception globale est donc simple : on lit les données, on les traite avec une fonction ou une boucle, puis on affiche le résultat à l'écran. Le tout au code **Java**.

## II. Réalisation

Tout mon travail a été faite sur **Java**, dans le même fichier mais dans des class séparés (Test le fichier puis exo 1 à l'exo 7 de chaque class). Chaque programme commence par la lecture des données avec un **Scanner** pour récupérer les valeurs entrées. Ensuite, j'entre le code des fonctions ou procédures correspondantes selon l'exercice.

### Exercice 1 - Écrire un programme qui affiche la factorielle d'un nombre

Le programme commence par demander à l'utilisateur d'entrer un nombre entier. La fonction **factorielle()** prend ce nombre en paramètre et effectue le calcul à l'aide d'une boucle **for**. La variable **resultat** est initialisée à 1 puis multipliée successivement par chaque nombre jusqu'au nombre **n**. Le résultat est ensuite affiché dans la console.

Voilà le programme :

```
package test;

import java.util.Scanner;

public class exo1 {

    static int factorielle(int n) {
        int resultat = 1;
        for (int i = 1; i <= n; i++) {
            resultat = resultat * i;
        }
        return resultat;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Entrez un nombre : ");
        int n = sc.nextInt();

        System.out.println("La factorielle de " + n + " est : " + factorielle(n));
        sc.close();
    }
}
```

Résultat :

```
Entrez un nombre : 8
La factorielle de 8 est : 40320
```

## Exercice 2 - Table de multiplication

J'ai écrit une procédure qui affiche la table de multiplication complète d'un nombre choisi.

Le programme va demander la valeur de quel table de multiplication puis la limite

désigner, puis affiche chaque ligne de calcul.

La boucle **for** va rendre le code plus simple et efficace.

Le programme :

```
package test;

import java.util.Scanner;

public class exo2 {

    static void tableMultiplication(int n, int limite) {
        System.out.println("Table de " + n + " :");
        for (int i = 1; i <= limite; i++) {
            System.out.println(n + " x " + i + " = " + (n * i));
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Entrez un nombre : ");
        int nombre = sc.nextInt();

        System.out.print("Jusqu'à combien voulez-vous aller ? ");
        int limite = sc.nextInt();

        tableMultiplication(nombre, limite);

        sc.close();
    }
}
```

Résultat:

```
Entrez un nombre : 80
Jusqu'à combien voulez-vous aller ? 3
Table de 80 :
80 x 1 = 80
80 x 2 = 160
80 x 3 = 240
```

## Exercice 3 – Choix entre factorielle ou table de multiplication

L'objectif de cet exercice était de réunir les deux programmes précédents dans un seul code plus interactif.

Pour cela, j'ai utilisé un **menu** qui affiche deux options :

- la première pour calculer la factorielle d'un nombre,
- la seconde pour afficher sa table de multiplication.

L'utilisateur choisit une option en entrant 1 ou 2. Le programme lit ensuite la valeur avec un Scanner et appelle la fonction correspondante.

La **boucle while** va permettre de recommencer le programme tant que l'utilisateur répond "O" (pour oui) ou par non.

Le programme :

```
public class Exo3 {
    static int factorielle(int n) {
        int resultat = 1;
        for (int i = 1; i <= n; i++) {
            resultat = resultat * i;
        }
        return resultat;
    }

    static void tableMultiplication(int n, int limite) {
        for (int i = 1; i <= limite; i++) {
            System.out.println(n + " x " + i + " = " + (n * i));
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String continuer = "O";

        while (continuer.equalsIgnoreCase("O")) {
            System.out.println("\n1 - Calculer une factorielle");
            System.out.println("2 - Afficher une table de multiplication");
            System.out.print("Votre choix : ");
            int choix = sc.nextInt();

            if (choix == 1) {
                System.out.print("Entrez un nombre : ");
                int n = sc.nextInt();
                System.out.println("Résultat : " + factorielle(n));
            } else if (choix == 2) {
                System.out.print("Entrez un nombre : ");
                int n = sc.nextInt();
                System.out.print("Jusqu'à combien ? ");
                int limite = sc.nextInt();
                tableMultiplication(n, limite);
            } else {
                System.out.println("Choix invalide !");
            }

            System.out.print("Voulez-vous recommencer ? (O/N) : ");
            continuer = sc.next();
        }

        System.out.println("Programme terminé !");
        sc.close();
    }
}
```

## Exercice 4 - Moyenne de la classe et meilleure note

Dans ce quatrième exercice, j'ai réalisé un programme qui demande à l'utilisateur d'entrer cinq notes différentes d'une classe. Les notes sont ensuite stockées dans un tableau pour être traitées puis le programme va additionner les notes pour calculer la moyenne, tout en gardant la plus grande valeur pour déterminer la meilleure note. Une fois le calcul terminé, la moyenne et la meilleure note sont affichées.

Le programme :

```
import java.util.Scanner;

public class exo4 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        double[] notes = new double[5];
        double somme = 0;
        double max = 0;

        for (int i = 0; i < notes.length; i++) {
            System.out.print("Entrez la note " + (i + 1) + " : ");
            notes[i] = sc.nextDouble();
            somme += notes[i];
            if (notes[i] > max) {
                max = notes[i];
            }
        }

        double moyenne = somme / notes.length;

        System.out.println("-----");
        System.out.println("Moyenne : " + moyenne);
        System.out.println("Meilleure note : " + max);
        System.out.println("-----");

        sc.close();
    }
}
```

Résultat :

```
Entrez la note 1 : 15
Entrez la note 2 : 10
Entrez la note 3 : 7
Entrez la note 4 : 20
Entrez la note 5 : 12
-----
Moyenne : 12.8
Meilleure note : 20.0
```

## Exercice 5 : Ecrire une fonction qui permet de retourner la valeur maximale d'un tableau de valeur

Dans ce cinquième exercice, le but était de trouver la valeur la plus grande parmi plusieurs nombres saisis par l'utilisateur. J'ai d'abord créé un tableau pour enregistrer cinq valeurs, puis j'ai programmé une fonction maximum() qui parcourt ce tableau. Le programme compare chaque valeur et garde toujours la plus élevée. Enfin, la valeur maximale est affichée à l'écran.

Le programme :

```
public class exo5 {

    static double maximum(double[] tab) {
        double max = tab[0];
        for (int i = 1; i < tab.length; i++) {
            if (tab[i] > max) {
                max = tab[i];
            }
        }
        return max;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        double[] valeurs = new double[5];
        for (int i = 0; i < valeurs.length; i++) {
            System.out.print("Entrez la valeur " + (i + 1) + " : ");
            valeurs[i] = sc.nextDouble();
        }

        System.out.println("La valeur maximale est : " + maximum(valeurs));

        sc.close();
    }
}
```

Résultat :

```
Entrez la valeur 1 : 156
Entrez la valeur 2 : 25
Entrez la valeur 3 : 34
Entrez la valeur 4 : 256
Entrez la valeur 5 : 4
La valeur maximale est : 256.0
```

## Exercice 7 – Calculer la moyenne et la valeur maximale d'un tableau

Dans ce programme, j'ai combiné deux fonctions : moyenne() et maximum(). La première calcule la moyenne des valeurs d'un tableau, et la seconde recherche la valeur la plus grande. L'utilisateur saisit cinq notes qui sont enregistrées dans un tableau. Ensuite, le programme affiche la moyenne de la classe et la note la plus haute.

Le programme :

```
import java.util.Scanner;

public class exo7 {

    static double maximum(double[] tab) {
        double max = tab[0];
        for (int i = 1; i < tab.length; i++) {
            if (tab[i] > max) {
                max = tab[i];
            }
        }
        return max;
    }

    static double moyenne(double[] tab) {
        double somme = 0;
        for (int i = 0; i < tab.length; i++) {
            somme += tab[i];
        }
        return somme / tab.length;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        double[] notes = new double[5];
        for (int i = 0; i < notes.length; i++) {
            System.out.print("Entrez la note " + (i + 1) + " : ");
            notes[i] = sc.nextDouble();
        }

        System.out.println("-----");
        System.out.println("Moyenne : " + moyenne(notes));
        System.out.println("Note maximale : " + maximum(notes));
        System.out.println("-----");
    }
}
```

Résultat:

```
Entrez la note 1 : 5
Entrez la note 2 : 17
Entrez la note 3 : 15
Entrez la note 4 : 9
Entrez la note 5 : 14
-----
Moyenne : 12.0
Note maximale : 17.0
-----
```

### III. Retour d'expérience

Pendant la réalisation de ce TP, j'ai appris à utiliser les fonctions et les procédures pour rendre un programme plus structuré et facile à comprendre. Au début, certains concepts comme la manipulation des tableaux ou la création de fonctions m'ont semblé un peu complexes, mais à force de pratique, j'ai compris leur utilité.

J'ai aussi remarqué qu'il est plus facile de corriger et d'améliorer un programme quand le code est bien organisé. Ce TP m'a donc permis de mieux comprendre la logique de la programmation en Java et de progresser pas à pas.

## Conclusion

Ce TP m'a permis de renforcer mes bases en **programmation Java**. J'ai appris à utiliser les **fonctions** et les **procédures** pour rendre un programme plus organisé et plus facile à comprendre.

Les différents exercices m'ont aidé à manipuler les **tableaux**, les **boucles** et les **conditions**, tout en apprenant à structurer un programme en plusieurs parties distinctes.

Grâce à cela, j'ai pu progresser dans la manière de réfléchir à le faire et faire le codage plus propre.

Pour certain partie des exercices, j'ai utilisé de IA pour comprendre certaines lignes de code et pour avoir des explications plus claires. Ça m'a permis de mieux apprendre, de corriger mes erreurs et mieux comprendre la signification.