

Département Mathématique et Informatique

# Compte rendu du TP1

**Filière :**

**« Génie du Logiciel et des Systèmes Informatiques  
Distribués »**

**GLSID**

**Programmation Orientée Objet Java :  
Les tableaux et les chaînes de caractères**

**Réalisé le 30/09/2022**

Réalisé par :

- Achraf TAFFAH

Encadrant d'entreprise :

- M. Abdelmajid BOUSSELHAM

**Année Universitaire : 2022-2023**

## Exercice 1 :

1. L'importation des classes dans lequel je vais les utiliser dans les méthodes que je dois développer.

```
import java.util.Arrays; //l'importation de la classe Arrays
import java.util.Scanner; // l'importation de la classe Scanner
```

2. Cette méthode permet d'insérer les notes avec l'utilisateur, le nombre des notes est donné en paramètre.

```
public float[] insertNotesValues(int length){
    float notes[] = new float[length];
    for(int i=0;i<length;i++) {
        System.out.println("Entrer la note "+i+" : ");
        notes[i]=sc.nextFloat();
    }
    return notes;
}
```

3. Cette méthode permet de trier et afficher un tableau des notes

```
public void TrieAndShow(float notes[]) {
    if(notes.length!=0) {
        Arrays.sort(notes);
        for(float note : notes) {
            System.out.println(note);
        }
    }
    else {
        System.out.println("Ce tableaux est vide");
    }
}
```

4. Cette méthode permet d'afficher la moyenne d'un tableau

```
public void displayMoy(float notes[]) {
    if(notes.length!=0){
        float sum=0;
        for(float note : notes) {
            sum+=note;
        }
        System.out.println("La moyenne est : "+ sum/(notes.length));
    }
    else {
        System.out.println("Le tableaux est vide");
    }
}
```

5. Cette méthode permet d'afficher la plus grande note qui existe dans un tableau

```
public void displayMax(float notes[]) {
    if(notes.length!=0) {
        System.out.println("La valeur maximal est : "+notes[notes.length-1]);
    }
    else {
        System.out.println("Ce tableaux est vide");
    }
}
```

6. Cette méthode permet d'afficher la plus moins note qui existe dans un tableau

```
public void displayMin(float notes[]) {
    if(notes.length!=0) {
        Arrays.sort(notes);
        System.out.println("La valeur minimal est : "+notes[0]);
    }
    else {
        System.out.println("Ce tableaux est vide");
    }
}
```

7. Cette méthode permet d'afficher le nombre des notes qui ont supérieur à une note donner

```
public void nbNotesEqualsNote(float notes[],float n) {
    if(notes.length!=0) {
        int nb=0;
        for(float note:notes) {
            if(note == n) {
                nb++;
            }
        }
        System.out.println("Le nombre d'étudiants ayant une vaut "+n+" est "+nb);
    }
    else {
        System.out.println("Ce tableaux est vide");
    }
}
```

8. Dans la classe Main j'ai testé ces méthodes

```
import java.util.Scanner;

public class Main{
    static Scanner sc = new Scanner(System.in);

    public static void main(String args[]) {
        /* Exercice1 */
        Exercice1 ex1 = new Exercice1();

        System.out.println("Entrer svp la longueur du tableau : ");
        int length=sc.nextInt();

        float notes[] = new float[length];
        notes = ex1.insertNotesValues(length);
        ex1.TrieAndShow(notes);
        ex1.displayMoy(notes);
        ex1.displayMax(notes);
        ex1.displayMin(notes);
        System.out.println("Entrer svp une note : ");
        ex1.nbNotesEqualsNote(notes,sc.nextFloat());
    }
}
```

9. Résultat de l'exercice 1

```
Entrer svp la longueur du tableau :
3
Entrer la note 0 :
11
Entrer la note 1 :
11
Entrer la note 2 :
17
11.0
11.0
17.0
La moyenne est : 13.0
La valeur maximal est : 17.0
La valeur minimal est : 11.0
Entrer svp une note :
11
Le nombre d'étudiants ayant une vaut 11.0 est 2
```

## Exercice 2 :

10. Cette méthode permet de conjuguer un verbe du premier group

```
public class Exercice2 {  
    public void conjuguer(String verbe) {  
        if(verbe.endsWith("er")){  
            String radical=verbe.substring(0,verbe.length()-2);  
            System.out.println("Je "+radical+"e");  
            System.out.println("Tu "+radical+"es");  
            System.out.println("Il "+radical+"e");  
            System.out.println("Nous "+radical+"ons");  
            System.out.println("Vous "+radical+"ez");  
            System.out.println("");  
        }  
        else {  
            System.out.println("Entrez un verbe de premier group , '"+verbe+"' n'est pas un verbe de premier group!!");  
        }  
    }  
}
```

11. Dans la classe Main j'ai testé la méthode Conjuguer

```
import java.util.Scanner;  
public class Main{  
    static Scanner sc = new Scanner(System.in);  
  
    public static void main(String args[]) {  
        /* Exercice1 */  
        /*Exercice1 ex1 = new Exercice1();  
  
        System.out.println("Entrer svp la longueur du tableau : ");  
        int length=sc.nextInt();  
  
        float notes[] = new float[length];  
        notes = ex1.insertNotesValues(length);  
        ex1.TrieAndShow(notes);  
        ex1.displayMoy(notes);  
        ex1.displayMax(notes);  
        ex1.displayMin(notes);  
        ex1.nbNotesEqualsNote(notes,7); */  
  
        /* Exercice2 */  
        Exercice2 ex2 = new Exercice2();  
        ex2.conjuguer("Chanter");  
        ex2.conjuguer("Voir");  
    }  
}
```

12. Résultat de l'exercice 2

```
Je Chante  
Tu Chantes  
Il Chante  
Nous Chantons  
Vous Chantez  
  
Entrez un verbe de premier group , 'Voir' n'est pas un verbe de premier group!!
```

## Exercice 3 :

13. Cette méthode permet de donner a l'utilisateur la possibilité d'entrer une chaine et retourné cette chaine.

```
Scanner sc = new Scanner(System.in);  
  
public StringBuilder saisir() {  
    StringBuilder chaine = new StringBuilder();  
    System.out.println("Entrer svp une chaine de caracter : ");  
    chaine.append(sc.nextLine());  
    return chaine;  
}
```

14. Cette méthode permet d'afficher une chaine de type **StringBuilder**

```

        public void afficher(StringBuilder chaine) {
            System.out.println(chaine);
        }

```

15. Cette méthode permet de retourner l'inverse d'une chaîne de type **StringBuilder**

```

public StringBuilder inverser(StringBuilder chaine) {
    return chaine.reverse();
}

```

16. Cette méthode permet de retourner le nombre des mots qui existent dans une chaîne de type **StringBuilder**

```

public int nombreDeMots(StringBuilder chaine) {
    return chaine.toString().split(" ").length;
}

```

17. Cette méthode permet de tester toutes les méthodes de l'exercice 4 à l'aide d'un menu

```

public void tester() {
    StringBuilder chaine = new StringBuilder();

    System.out.println("");
    System.out.println("");
    System.out.println("1) Saisir une chaîne ");
    System.out.println("2) Afficher la chaîne ");
    System.out.println("3) Inverser la chaîne ");
    System.out.println("4) Afficher le nombre de mots qui contient cette chaîne ");
    System.out.println("5) Quitter ");

    String choix="";
    do {
        System.out.println("Entrer svp un choix : ");
        choix = sc.nextLine();

        switch(choix) {
            case "1" : {
                chaine = saisir();
                break;
            }
            case "2" : {
                afficher(chaine);
                break;
            }
            case "3" : {
                afficher(inverser(chaine));
                break;
            }
            case "4" : {
                System.out.println("Cette chaîne contient "+nombreDeMots(chaine)+" mots");
                break;
            }
            default : {
                break;
            }
        }
    } while (!choix.equals("5")); // quitting the program
}

```

18. Dans la classe Main j'ai testé la méthode tester ()

```

/*Exercice3 */
Exercice3 ex3 =new Exercice3();
ex3.tester();

```

#### 19. Le résultat de l'exercice 3

```
1) Saisir une chaine
2) Afficher la chaine
3) Inverser la chaine
4) Afficher le nombre de mots qui contient cette chaine
5) Quitter
Entrer svp un choix :
1
Entrer svp une chaine de caracter :
achraf taffah glsid2 ensetm
Entrer svp un choix :
2
achraf taffah glsid2 ensetm
Entrer svp un choix :
3
mtesne 2dislg haffat farhca
Entrer svp un choix :
4
Cette chaine contient 4 mots
Entrer svp un choix :
```

#### Exercice 4 :

#### 20. Cette méthode permet d'afficher le nombre d'occurrence des caractères qui engendre une chaine

```
class Exercice4 {
    void getOccuringChar(String str){
        str=str.replace(" ","");
        int countString = str.length();
        for(int i=0;i<countString;i++) {
            if(!str.substring(0,i).contains(""+str.charAt(i)))
                System.out.println(str.charAt(i)+"--> "+(str.length() - str.replace(Character.toString(str.charAt(i)),"").length()));
        }
    }
}
```

#### 21. J'ai testé cette méthode dans classe Main

```
/*Exercice4 */
Exercice4 ex4 = new Exercice4();
ex4.getOccuringChar("Achraf TAFFAH GLSID2 ENSETM JAVA POO");
```

#### 22. Le résultat de l'exercice 4

```
A--> 5
c--> 1
h--> 1
r--> 1
a--> 1
f--> 1
T--> 2
F--> 2
H--> 1
G--> 1
L--> 1
S--> 2
I--> 1
D--> 1
2--> 1
E--> 2
N--> 1
M--> 1
J--> 1
V--> 1
P--> 1
O--> 2
```