PW 1: Arrays and Strings

PREPARED BY :
BRAIMI HAMZA

JAVA OBJECT ORIENTED PROGRAMMING .

Java Introduction

What is Java?

Java is a popular programming language, created in 1995.
It is owned by Oracle, and more than **3 billion** devices run Java.
It is used for:

- Mobile applications (especially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection
- And much, much more!

Why Use Java?

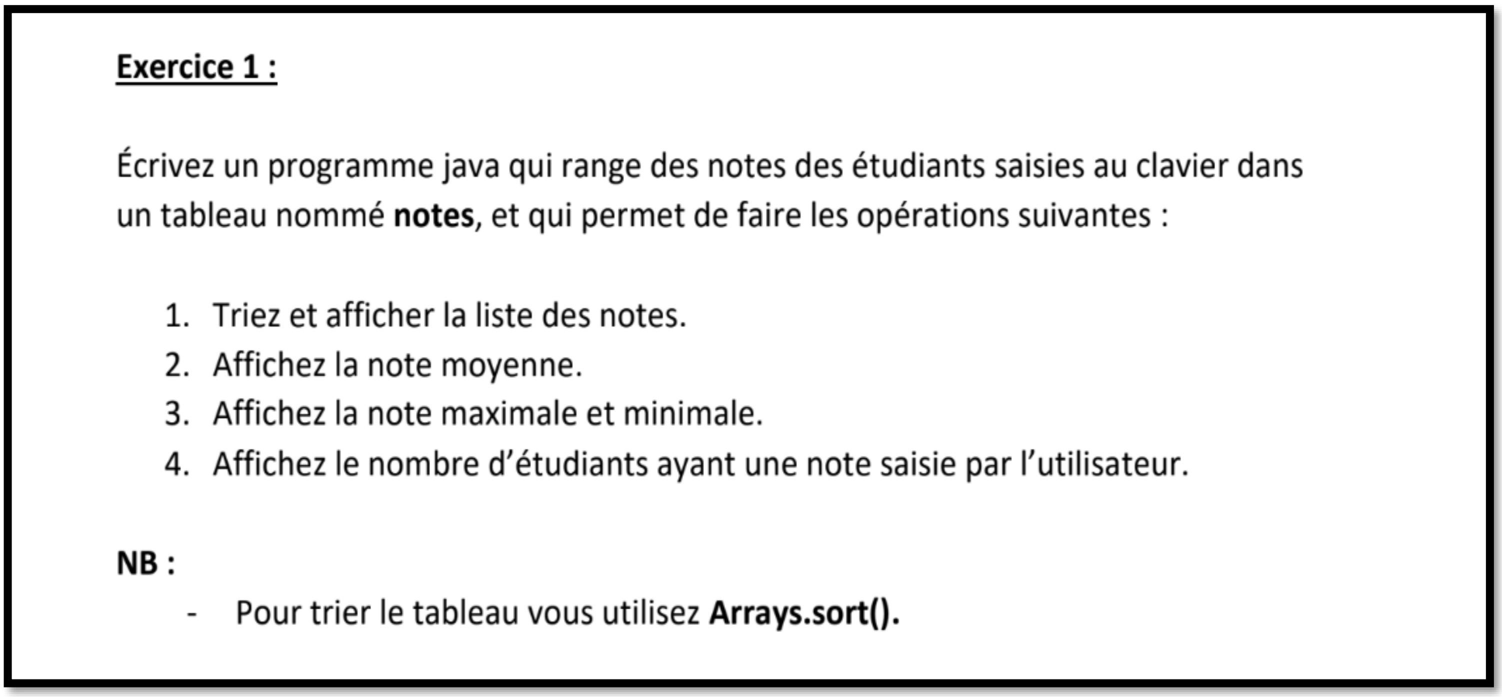
Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
It is one of the most popular programming languages in the world
It has a large demand in the current job market
It is easy to learn and simple to use
It is open-source and free
It is secure, fast and powerful
It has a huge community support (tens of millions of developers)
Java is an object-oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs
As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa

Exercise 1:

The first exercise is about learning how to use arrays and their different methods in Java Such as:

Array. Length: In **Java**, the **array length** is the number of elements that an array can holds. There is no predefined method to obtain the length of an array.
Array. Sort: The Arrays class of 'java.util' package provides the sort method that takes an array as an argument and sorts the array. This is a direct sorting method, and you can sort an array with just one method call.

Math. Min: The Java.lang.math.min() function is an inbuilt function in java that returns the minimum of two numbers
Math.max : The Java.lang.math.min() function is an inbuilt function in java that returns the maximum Of two numbers .



Exercice 1 :

Écrivez un programme java qui range des notes des étudiants saisies au clavier dans un tableau nommé **notes**, et qui permet de faire les opérations suivantes :

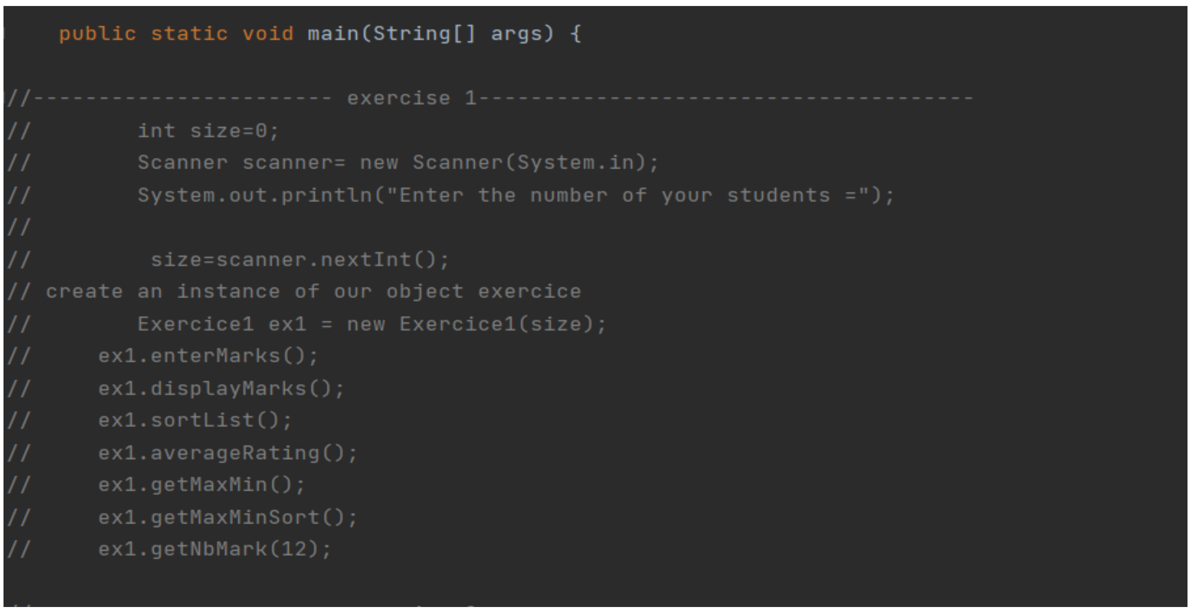
1. Triez et afficher la liste des notes.
2. Affichez la note moyenne.
3. Affichez la note maximale et minimale.
4. Affichez le nombre d'étudiants ayant une note saisie par l'utilisateur.

NB :

- Pour trier le tableau vous utilisez **Arrays.sort().**

Figure 1: Exercise 1

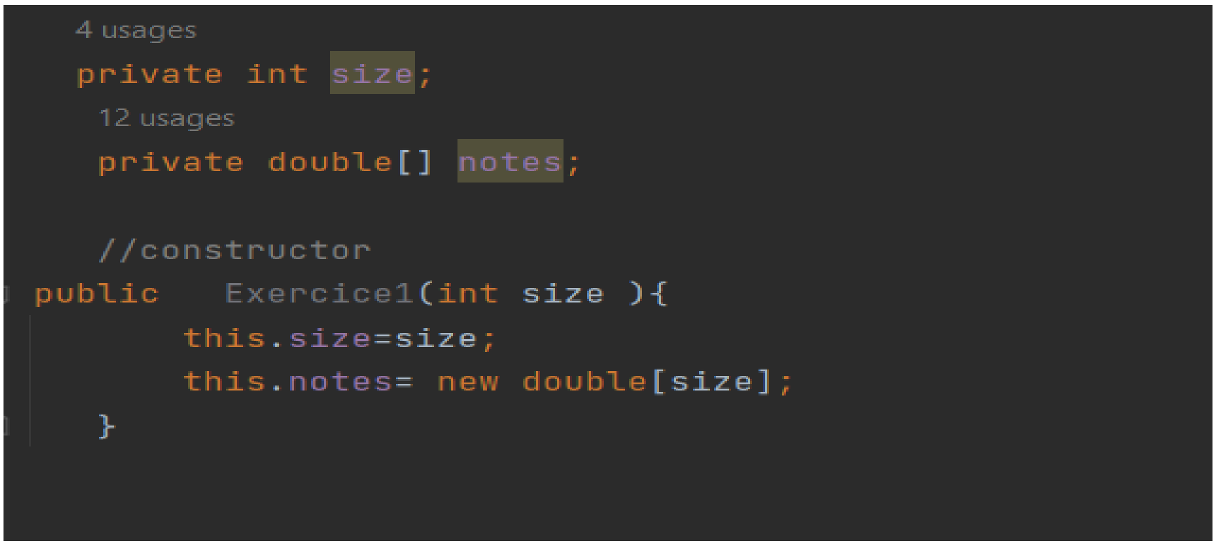
The first step we need to do is create an instance (object) of our Class 1 exercise. Then scan to get the number of students the user wants to enter their grades.



```
    public static void main(String[] args) {

//----------------------- exercise 1-------------------------------------
//         int size=0;
//         Scanner scanner= new Scanner(System.in);
//         System.out.println("Enter the number of your students =");
//
//          size=scanner.nextInt();
// create an instance of our object exercice
//         Exercice1 ex1 = new Exercice1(size);
//      ex1.enterMarks();
//      ex1.displayMarks();
//      ex1.sortList();
//      ex1.averageRating();
//      ex1.getMaxMin();
//      ex1.getMaxMinSort();
//      ex1.getNbMark(12);
```

Figure 2: main method

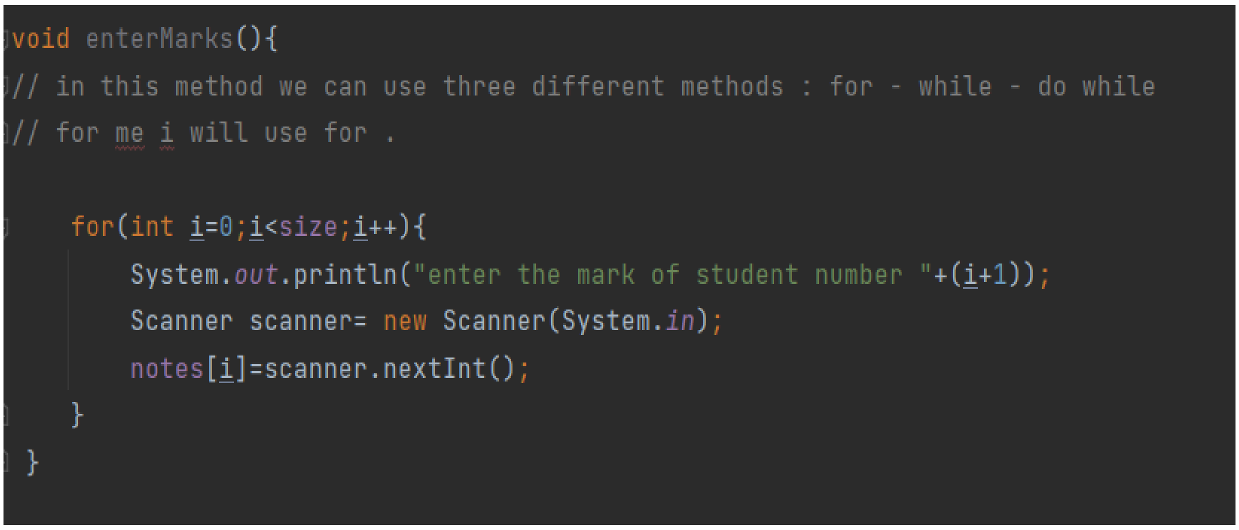
Implementation of constructor will be the second step to build and fix the size of notes array



```java
    4 usages
    private int size;
     12 usages
     private double[] notes;

     //constructor
  public    Exercice1(int size ){
           this.size=size;
           this.notes= new double[size];
     }
```

Figure 3: constructor of class Exercise1

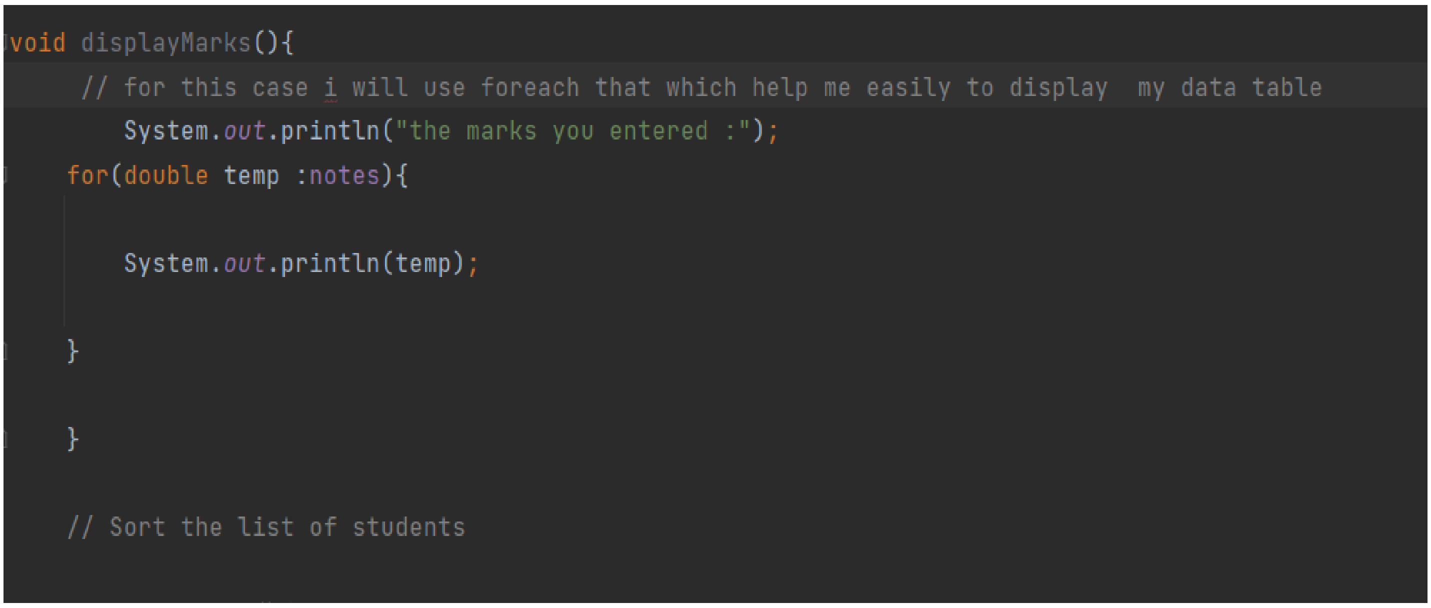
Then we should implement the method which will help us to enter marks of students



```java
void enterMarks(){
// in this method we can use three different methods : for - while - do while
// for me i will use for .

    for(int i=0;i<size;i++){
        System.out.println("enter the mark of student number "+(i+1));
        Scanner scanner= new Scanner(System.in);
        notes[i]=scanner.nextInt();
    }
 }
```

Figure 4: Enter Marks Method

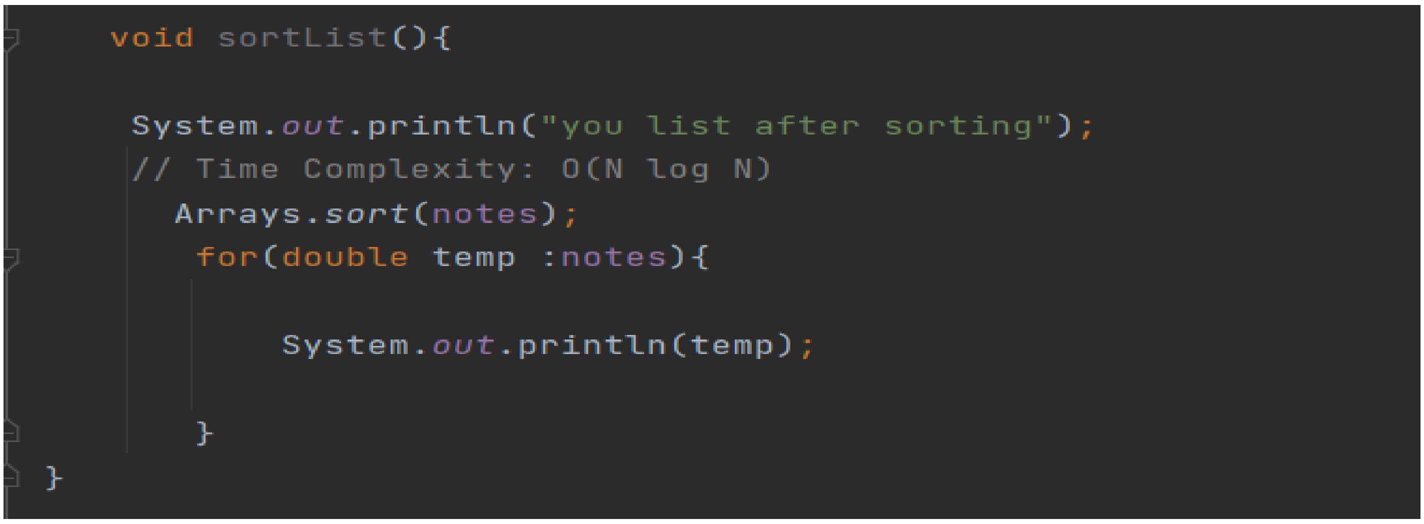
To display the data that we are stocked in notes array we should get the size of array to use it in loop for.
Array. Length: is the method to get the size of an array



```
void displayMarks(){
    // for this case i will use foreach that which help me easily to display  my data table
        System.out.println("the marks you entered :");
    for(double temp :notes){

        System.out.println(temp);

    }

    }

    // Sort the list of students
```

Figure 5: Display Marks Method

In this case I use foreach to display marks. Foreach is faster because the local variable that stores the value of the element in the array is faster to access than an element in the array

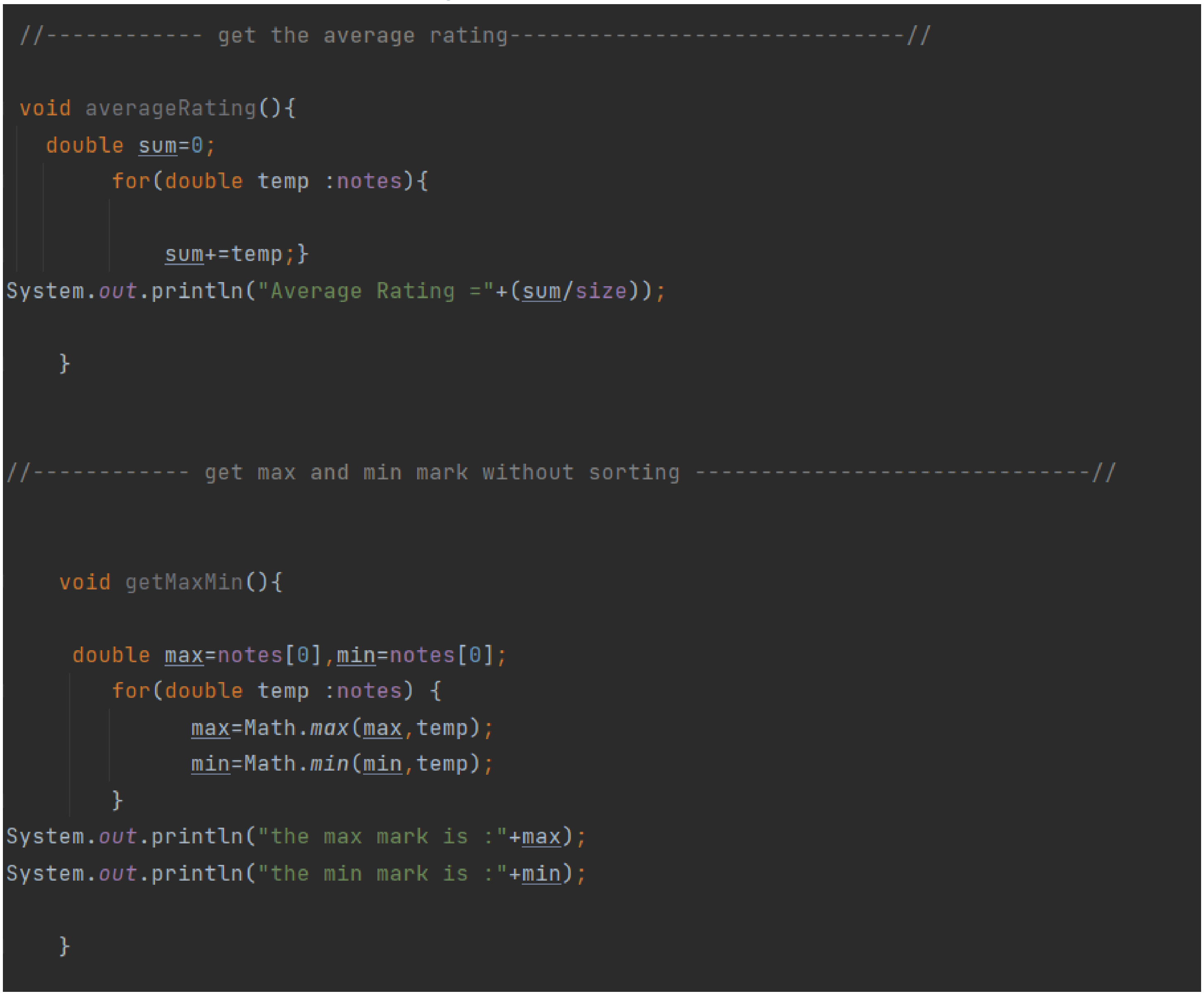


```
    void sortList(){

     System.out.println("you list after sorting");
     // Time Complexity: O(N log N)
       Arrays.sort(notes);
        for(double temp :notes){

            System.out.println(temp);

        }
 }
```

Figure 6: Sort Marks Method

The sort() method sorts the elements of an array in place and returns the reference to the same array, now sorted. The default sort order is ascending, built upon converting the elements into strings, then comparing their sequences of UTF-16 code units' values.

Figure 6: Sort Marks Method



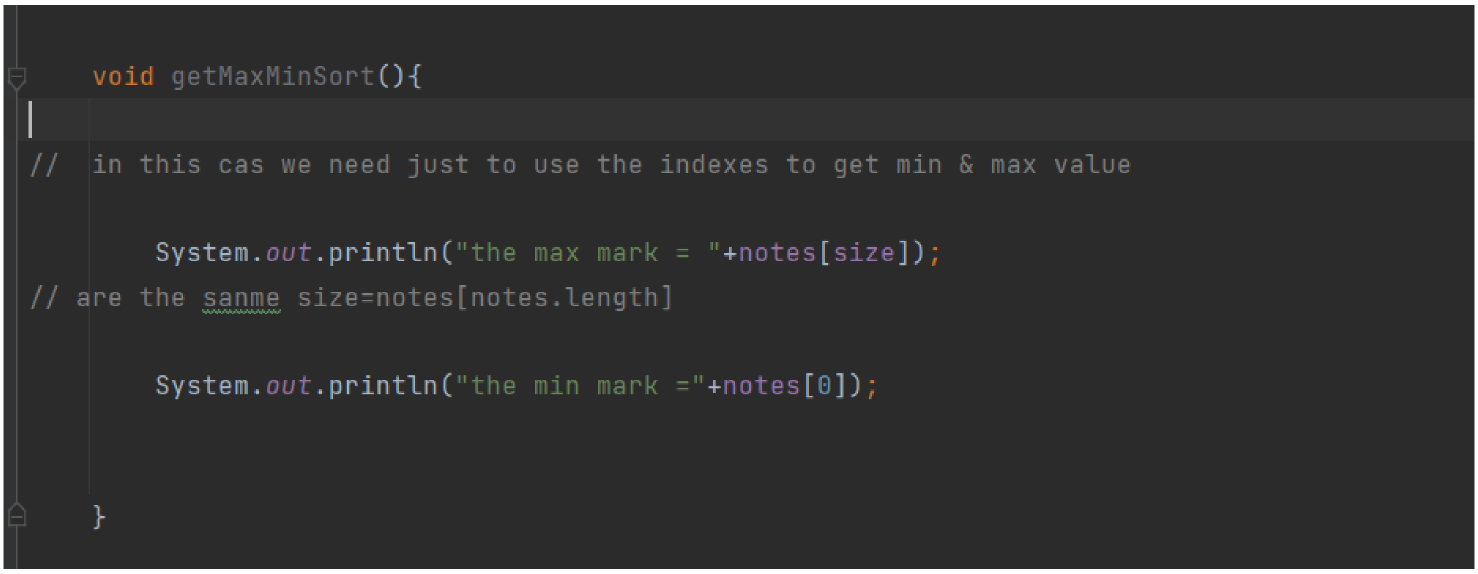
```
 //------------ get the average rating------------------------------//

 void averageRating(){
   double sum=0;
        for(double temp :notes){

            sum+=temp;}
System.out.println("Average Rating ="+(sum/size));

    }


//------------ get max and min mark without sorting ------------------------------//


    void getMaxMin(){

     double max=notes[0],min=notes[0];
        for(double temp :notes) {
              max=Math.max(max,temp);
              min=Math.min(min,temp);
        }
System.out.println("the max mark is :"+max);
System.out.println("the min mark is :"+min);

    }
```

Figure 7: Average-rating & Marks Method

The second way to get min and max values is after sorting array

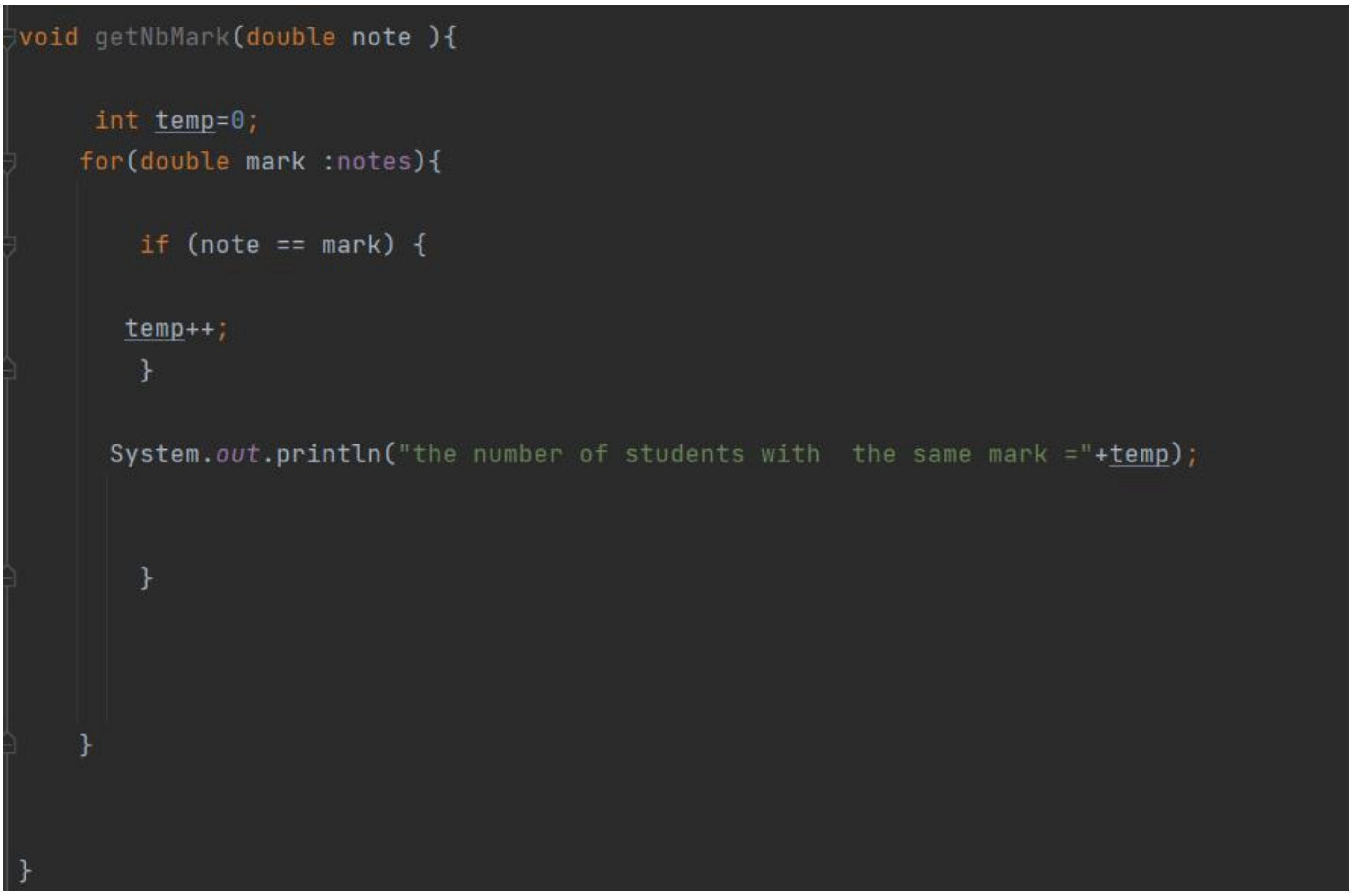
Min: the first element

Max : will be the last element after sorting



```
    void getMaxMinSort(){

//  in this cas we need just to use the indexes to get min & max value

        System.out.println("the max mark = "+notes[size]);
// are the sanme size=notes[notes.length]

        System.out.println("the min mark ="+notes[0]);


    }
```

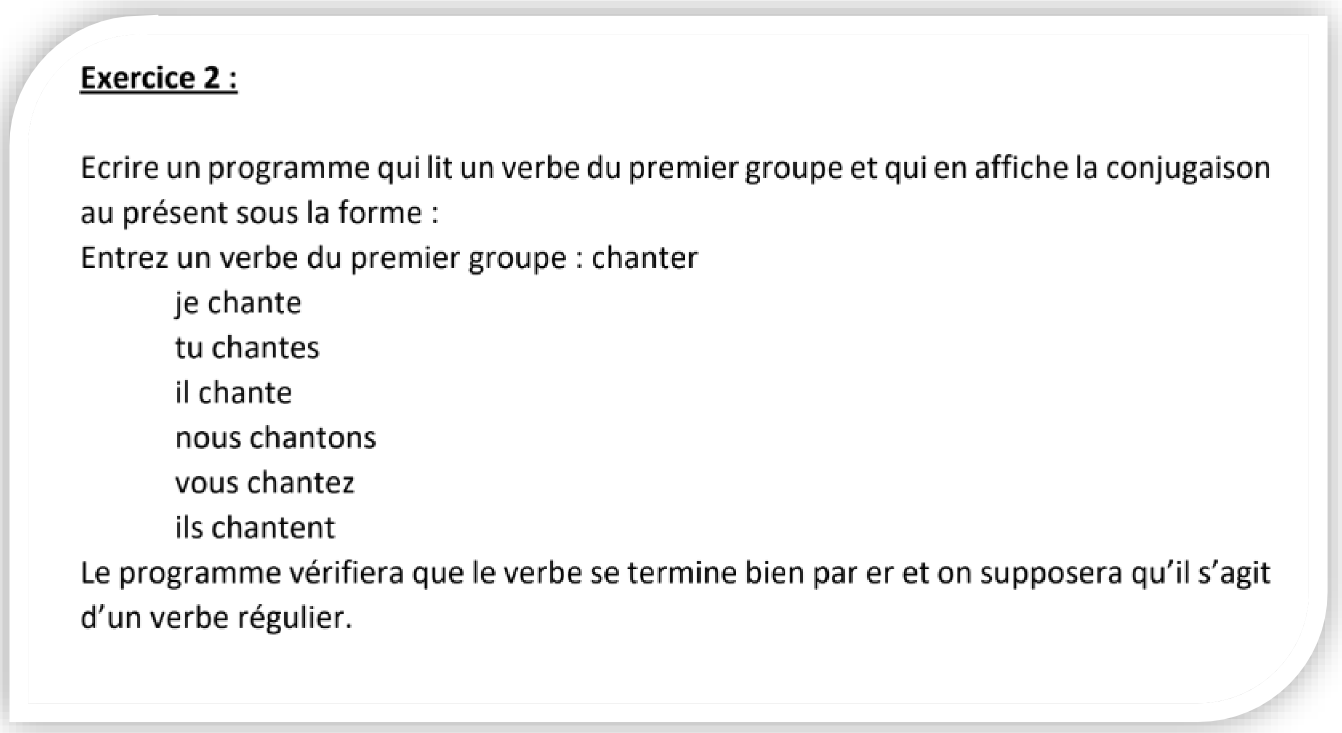
Figure 8: Min - Max Marks Method



```
void getNbMark(double note ){

     int temp=0;
    for(double mark :notes){

        if (note == mark) {

       temp++;
        }

      System.out.println("the number of students with  the same mark ="+temp);



        }



    }



}
```

Figure 9: get NbMarks Method

Exercise 2:



Exercice 2 :

Ecrire un programme qui lit un verbe du premier groupe et qui en affiche la conjugaison au présent sous la forme :

```
Entrez un verbe du premier groupe : chanter
        je chante
        tu chantes
        il chante
        nous chantons
        vous chantez
        ils chantent
```

Le programme vérifiera que le verbe se termine bien par er et on supposera qu'il s'agit d'un verbe régulier.

Figure 9: the second exercise

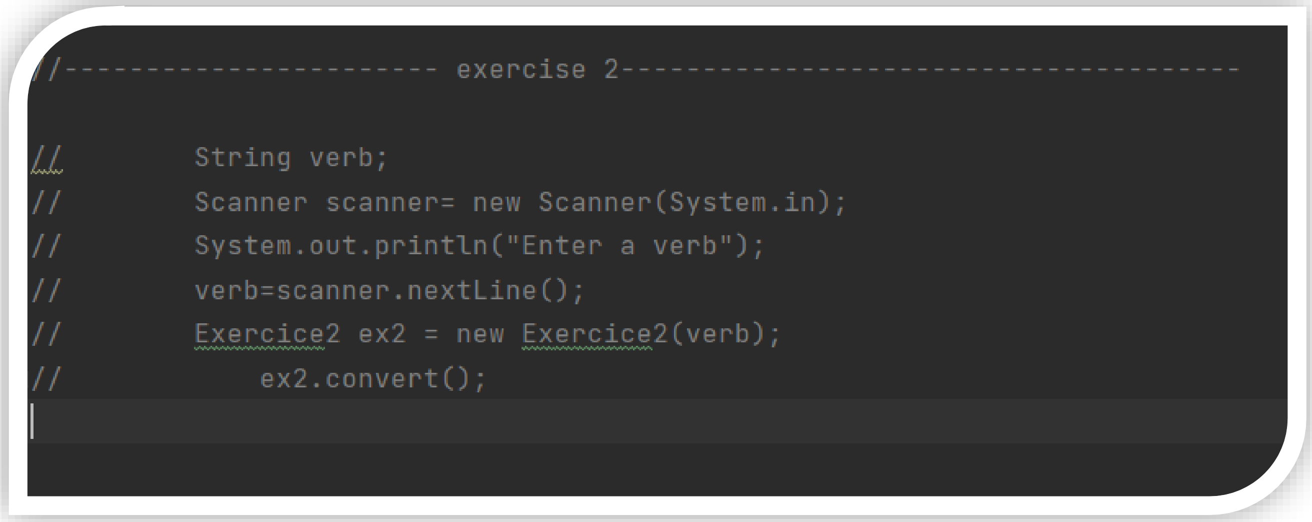
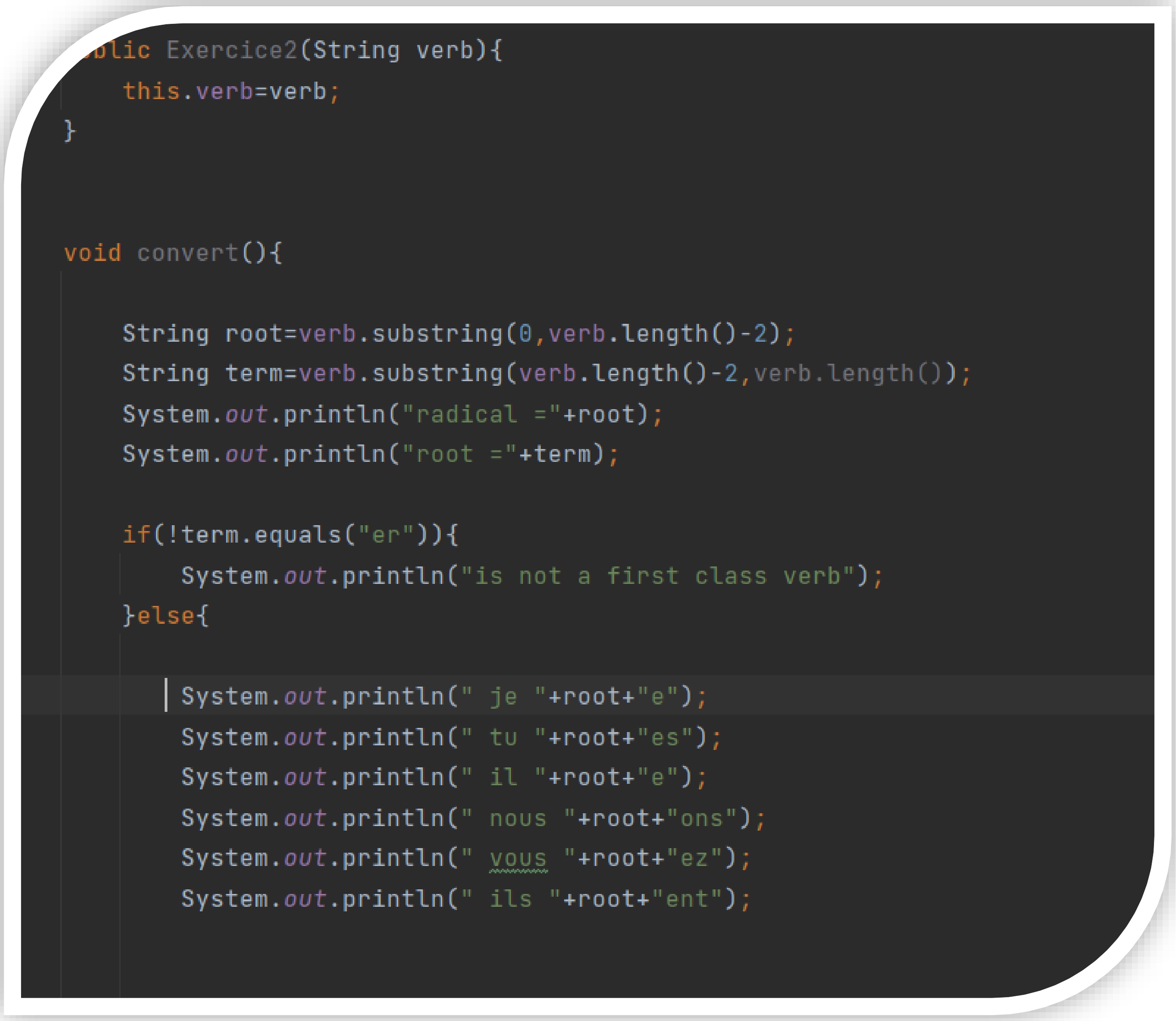

```
//--------------------- exercise 2-----------------------------------

//        String verb;
//        Scanner scanner= new Scanner(System.in);
//        System.out.println("Enter a verb");
//        verb=scanner.nextLine();
//        Exercice2 ex2 = new Exercice2(verb);
//            ex2.convert();
```

Figure 10: the main method

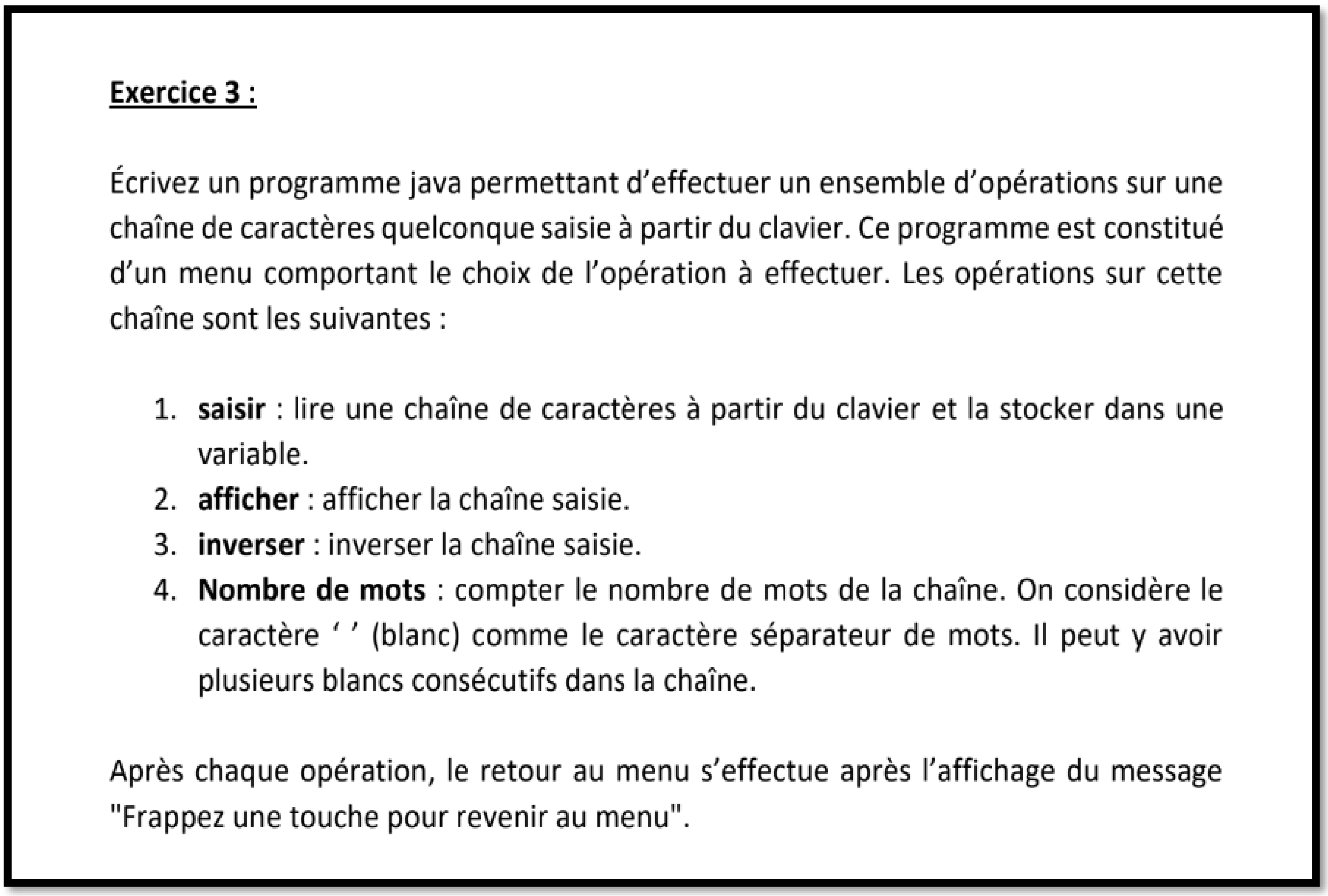


```
blic Exercice2(String verb){
    this.verb=verb;
}


void convert(){

    String root=verb.substring(0,verb.length()-2);
    String term=verb.substring(verb.length()-2,verb.length());
    System.out.println("radical ="+root);
    System.out.println("root ="+term);

    if(!term.equals("er")){
        System.out.println("is not a first class verb");
    }else{

        System.out.println(" je "+root+"e");
        System.out.println(" tu "+root+"es");
        System.out.println(" il "+root+"e");
        System.out.println(" nous "+root+"ons");
        System.out.println(" vous "+root+"ez");
        System.out.println(" ils "+root+"ent");
```

Figure 11: convert method

This program reads a verb of the first group and displays its conjugation, but before that we need to check if the verb response to the condition.

Exercise 3:



Exercice 3 :

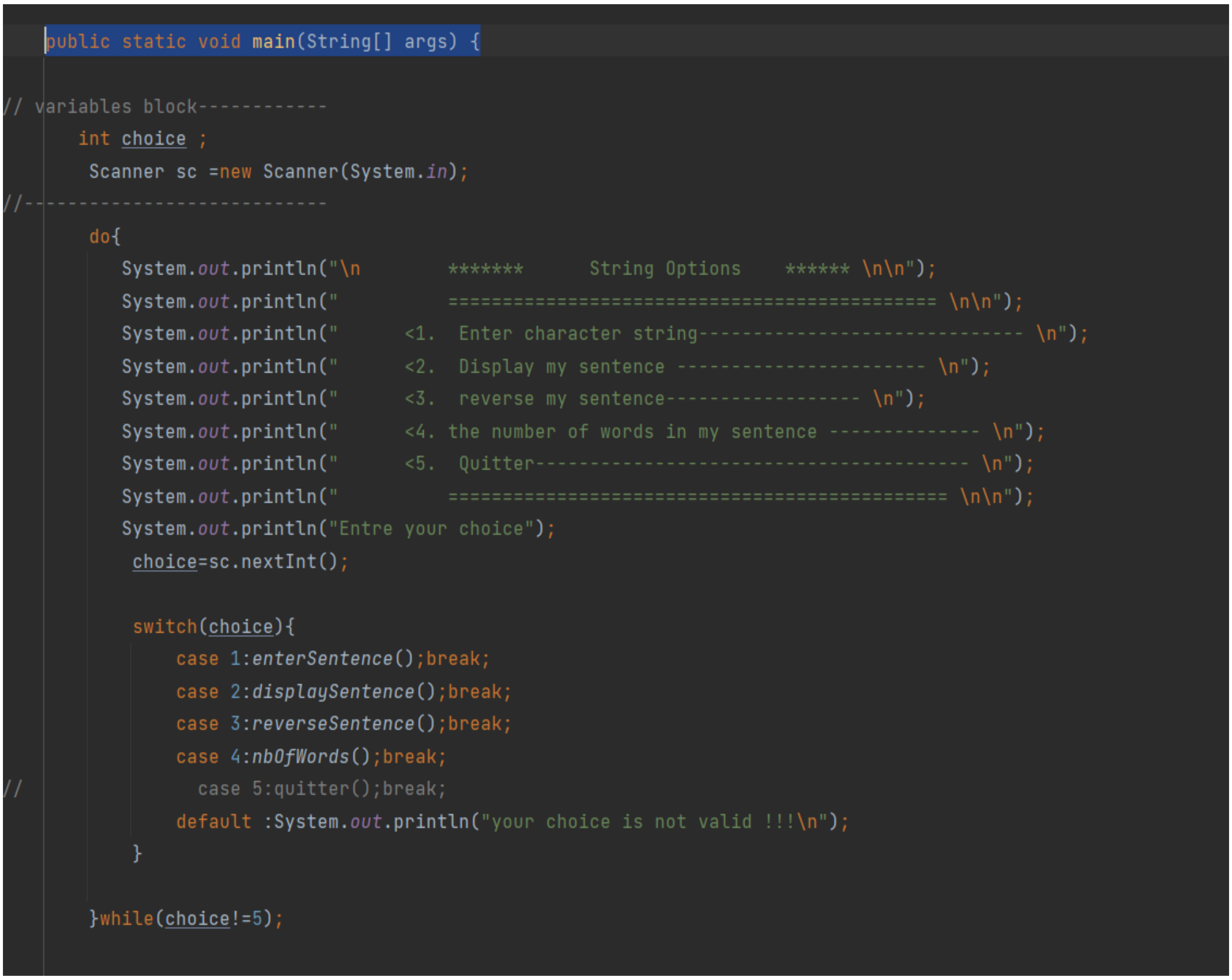
Écrivez un programme java permettant d'effectuer un ensemble d'opérations sur une chaîne de caractères quelconque saisie à partir du clavier. Ce programme est constitué d'un menu comportant le choix de l'opération à effectuer. Les opérations sur cette chaîne sont les suivantes :

1. **saisir** : lire une chaîne de caractères à partir du clavier et la stocker dans une variable.
2. **afficher** : afficher la chaîne saisie.
3. **inverser** : inverser la chaîne saisie.
4. **Nombre de mots** : compter le nombre de mots de la chaîne. On considère le caractère ' ' (blanc) comme le caractère séparateur de mots. Il peut y avoir plusieurs blancs consécutifs dans la chaîne.

Après chaque opération, le retour au menu s'effectue après l'affichage du message "Frappez une touche pour revenir au menu".

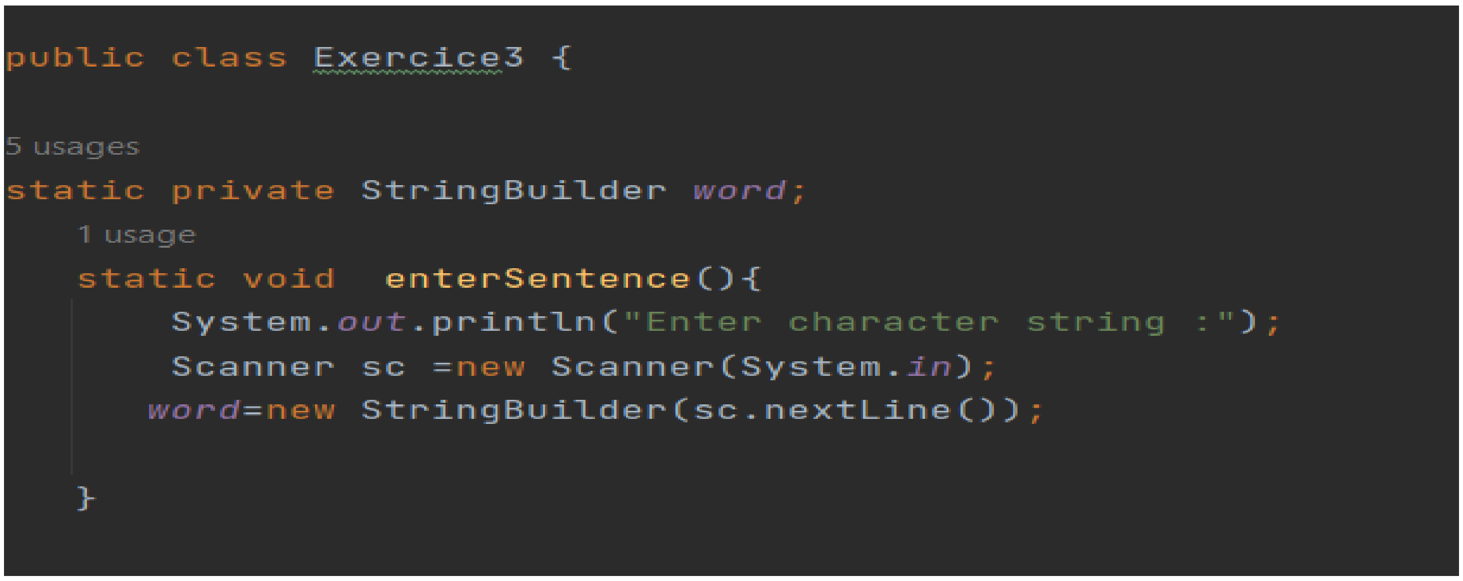
Figure 12: the third exercise

Main method should display to the users a table of options to choose the type of operation wanted



```
    public static void main(String[] args) {

// variables block-----------
        int choice ;
         Scanner sc =new Scanner(System.in);
//---------------------------
         do{
            System.out.println("\n        *******      String Options    ****** \n\n");
            System.out.println("          ============================================= \n\n");
            System.out.println("      <1.  Enter character string------------------------------ \n");
            System.out.println("      <2.  Display my sentence ----------------------- \n");
            System.out.println("      <3.  reverse my sentence------------------ \n");
            System.out.println("      <4. the number of words in my sentence -------------- \n");
            System.out.println("      <5.  Quitter---------------------------------------- \n");
            System.out.println("          ============================================== \n\n");
            System.out.println("Entre your choice");
             choice=sc.nextInt();

             switch(choice){
                 case 1:enterSentence();break;
                 case 2:displaySentence();break;
                 case 3:reverseSentence();break;
                 case 4:nbOfWords();break;
//                 case 5:quitter();break;
                 default :System.out.println("your choice is not valid !!!\n");
             }

         }while(choice!=5);
```

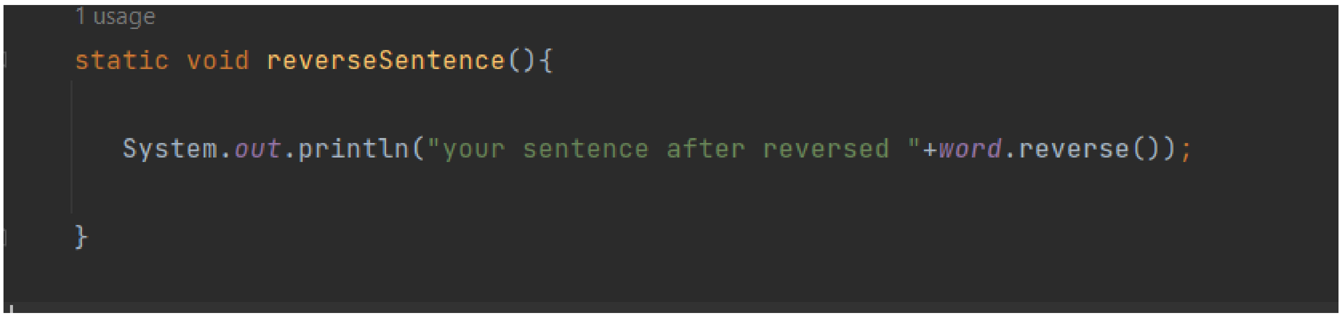
Figure 13: main method



```
public class Exercice3 {

5 usages
static private StringBuilder word;
    1 usage
    static void  enterSentence(){
        System.out.println("Enter character string :");
        Scanner sc =new Scanner(System.in);
      word=new StringBuilder(sc.nextLine());

    }
```

Figure 14: Enter sentence method

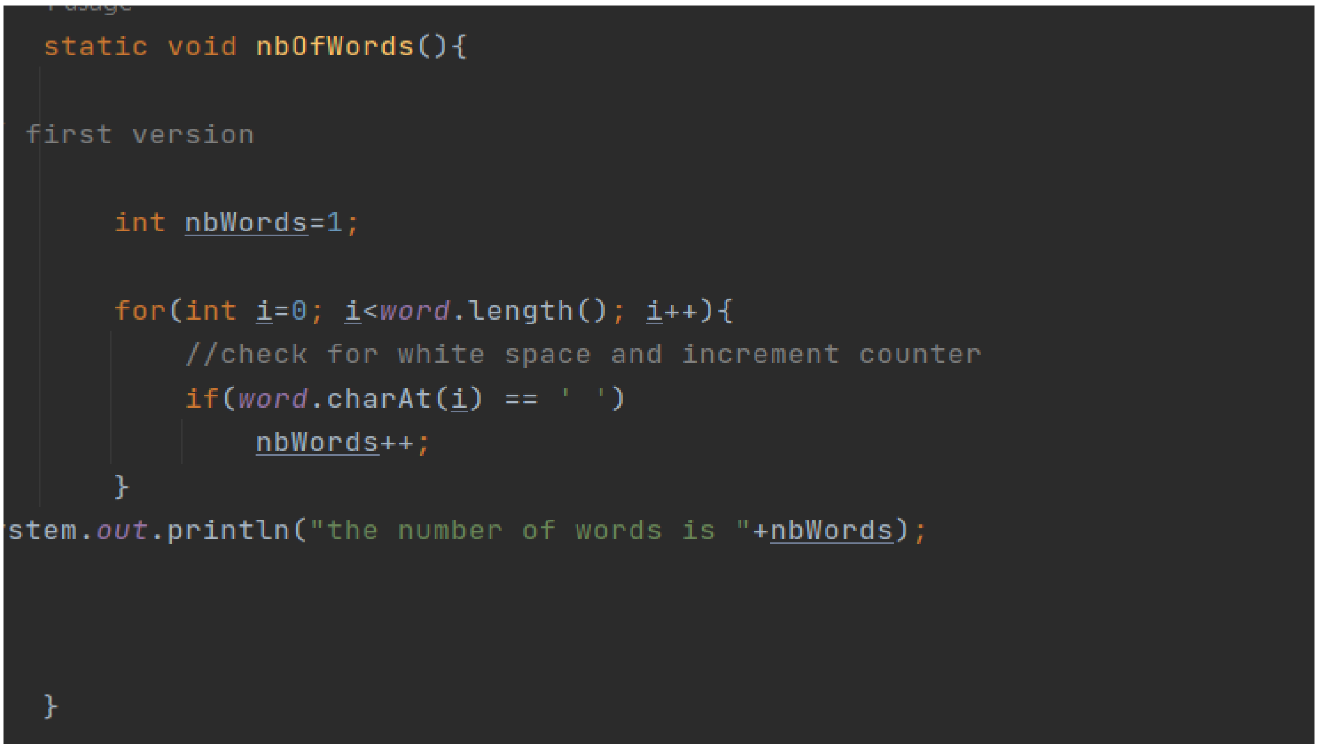
In this exercise I use StringBuilder instead of string for many reasons: StringBuilder in Java is a class used to create a mutable, or in other words, a modifiable succession of characters. Like StringBuffer, the StringBuilder class is an alternative to the Java Strings Class, as the Strings class provides an immutable succession of characters.



```
1 usage
static void reverseSentence(){

    System.out.println("your sentence after reversed "+word.reverse());

}
```

Figure 15: Reverse method

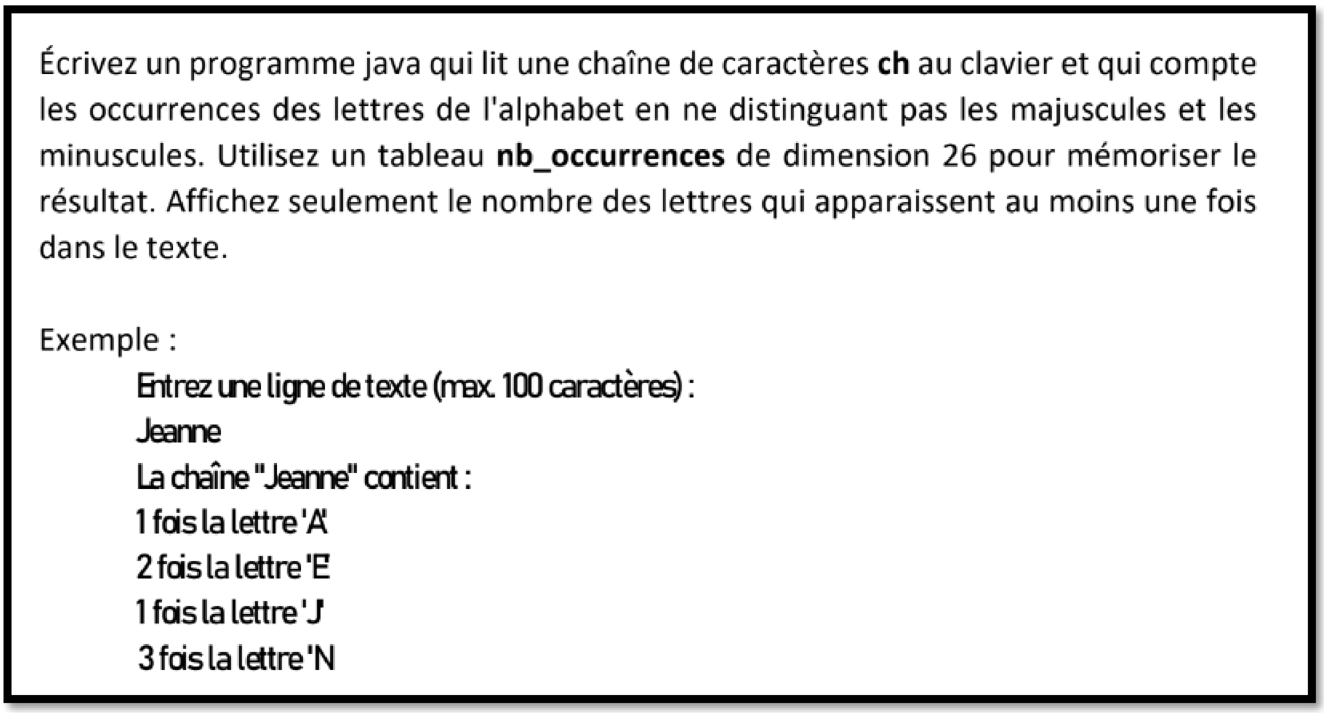
The **reverse()** method of StringBuilder is used to **reverse the characters in theStringBuilder**. The method helps to this character sequence to be replaced by the reverse of the sequence.



```
  static void nbOfWords(){

 first version

      int nbWords=1;

      for(int i=0; i<word.length(); i++){
          //check for white space and increment counter
          if(word.charAt(i) == ' ')
              nbWords++;
      }
stem.out.println("the number of words is "+nbWords);



  }
```

Figure 16: Get Number of words method

Exercise 4:

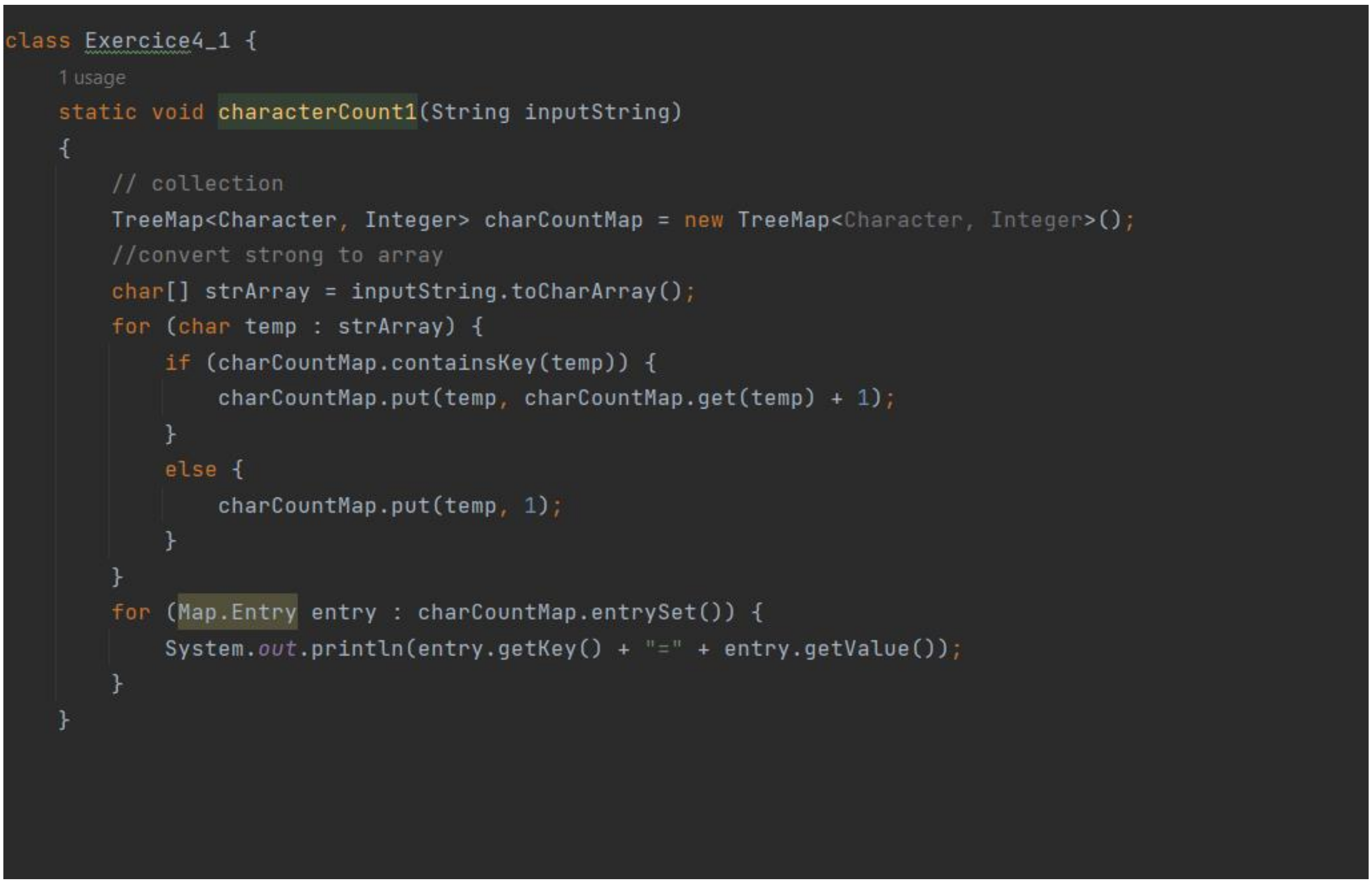


Écrivez un programme java qui lit une chaîne de caractères **ch** au clavier et qui compte les occurrences des lettres de l'alphabet en ne distinguant pas les majuscules et les minuscules. Utilisez un tableau **nb_occurrences** de dimension 26 pour mémoriser le résultat. Affichez seulement le nombre des lettres qui apparaissent au moins une fois dans le texte.

Exemple :

```
Entrez une ligne de texte (max. 100 caractères) :
Jeanne
La chaîne "Jeanne" contient :
1 fois la lettre 'A'
2 fois la lettre 'E'
1 fois la lettre 'J'
3 fois la lettre 'N
```

Figure 17: EXERCISE 4

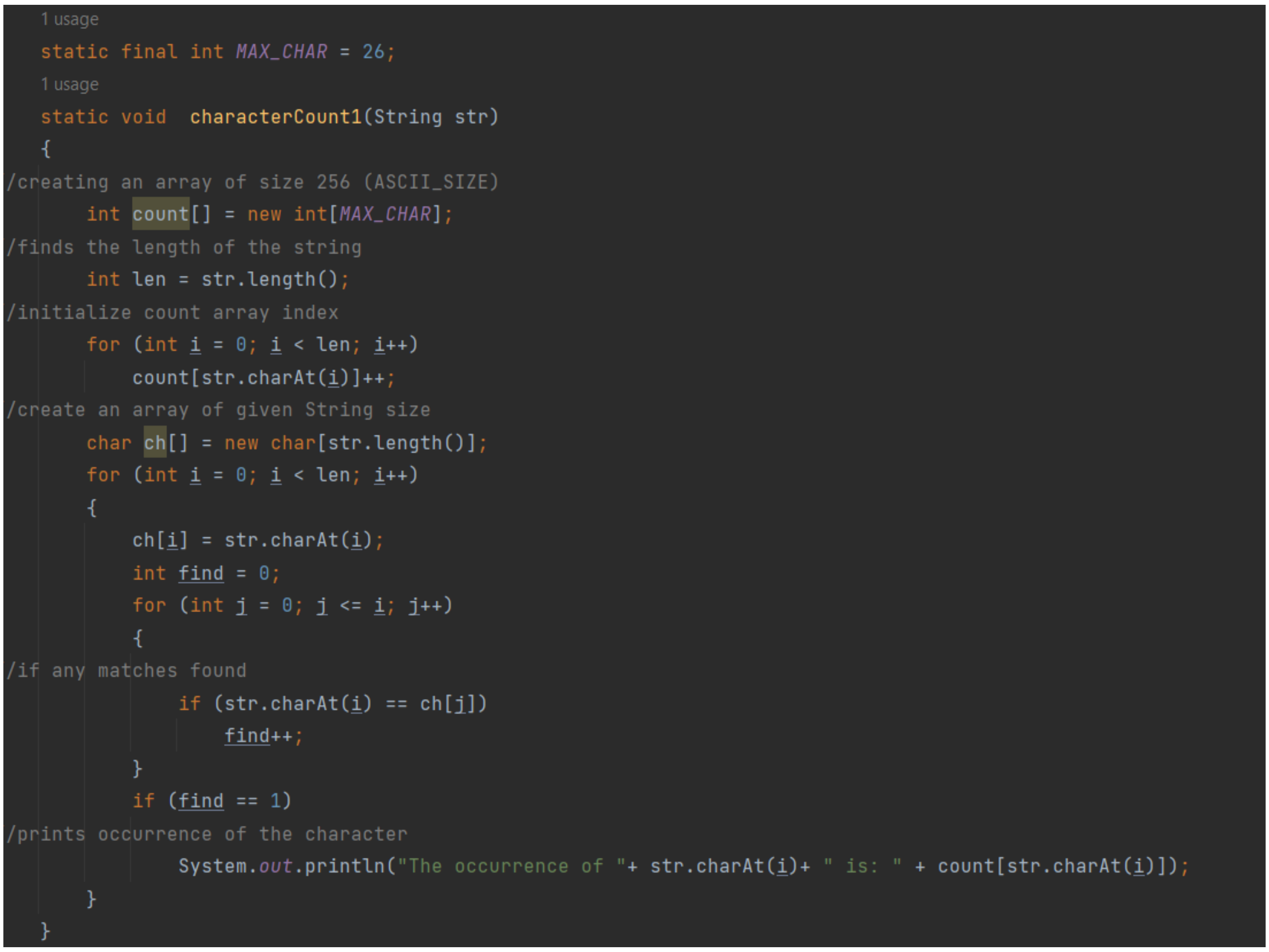


```
class Exercice4_1 {
    1 usage
    static void characterCount1(String inputString)
    {
        // collection
        TreeMap<Character, Integer> charCountMap = new TreeMap<Character, Integer>();
        //convert strong to array
        char[] strArray = inputString.toCharArray();
        for (char temp : strArray) {
            if (charCountMap.containsKey(temp)) {
                charCountMap.put(temp, charCountMap.get(temp) + 1);
            }
            else {
                charCountMap.put(temp, 1);
            }
        }
        for (Map.Entry entry : charCountMap.entrySet()) {
            System.out.println(entry.getKey() + "=" + entry.getValue());
        }
    }
```

Figure 17: Character count method

Given a string, the task is to write a program in Java which prints the number of occurrences of each character in a string.

- Declare a Hashmap in Java of {char, int}.
- Traverse in the string, check if the Hashmap already contains the traversed character or not.
- If it is present, then increase its count using get() and put() function in Hashmap.
- Once the traversal is completed, traverse in the Hashmap and print the character.



```
    1 usage
    static final int MAX_CHAR = 26;
    1 usage
    static void  characterCount1(String str)
    {
/creating an array of size 256 (ASCII_SIZE)
        int count[] = new int[MAX_CHAR];
/finds the length of the string
        int len = str.length();
/initialize count array index
        for (int i = 0; i < len; i++)
            count[str.charAt(i)]++;
/create an array of given String size
        char ch[] = new char[str.length()];
        for (int i = 0; i < len; i++)
        {
            ch[i] = str.charAt(i);
            int find = 0;
            for (int j = 0; j <= i; j++)
            {
/if any matches found
                if (str.charAt(i) == ch[j])
                    find++;
            }
            if (find == 1)
/prints occurrence of the character
                System.out.println("The occurrence of "+ str.charAt(i)+ " is: " + count[str.charAt(i)]);
        }
    }
```

Figure 17: Character count method 2

The second way to get character count is using this way :

- Creating an array to count number of occurrences
- Initialize array of occurrence
    - Create an array of given string size .

Conclusion

Java is one of the most popular programming languages in the world, which you can use to develop web apps, mobile apps, and desktop apps. This lab therefore allows you to have a general idea of the java syntax,