TD1 : Programmer ses premières méthodes

Exercice 1 : Ecrire dans une classe `Tableau` trois fonctions :
- `public static int maxTableau(int[] tab);`
- `public static void afficherTableau(int[] tab);`
- `public static int sommeTableau(int[] tab);`

puis tester les dans une fonction main.

Rappels : On peut afficher une information dans la console avec les méthodes `System.out.println()` et `System.out.print()`. On peut concaténer deux Strings avec l'opérateur "+". On peut accéder à la taille d'un tableau avec l'attribut `length`. Ainsi `tab.length` est la longeur du tableau `tab`.

Correction Exercice 1 :

```java
public class Tableau{
    public static int maxTableau(int[] tab) {
        int res = tab[0];
        for(int i = 1; i<tab.length;i++) {
            if(tab[i] > res) res = tab[i];
        }
        return res;
    }

    public static void afficherTableau(int[] tab) {
        System.out.print("[");
        for(int i = 0; i<tab.length;i++) {
            System.out.print("\t" + tab[i] + "\t");
        }
        System.out.println("]");
    }

    public static int sommeTableau(int[] tab) {
        int res = 0;
        for(int i = 0; i<tab.length;i++) {
            res += tab[i];
        }
        return res;
    }

    public static void main(String[] args){
        int[] tab = {3,-2,5,7};
        afficherTableau(tab);
        System.out.println("Max = " + maxTableau(tab));
        System.out.println("Somme = " + sommeTableau(tab));
    }
}
/*En ligne de commande
```

```
javac Tableau.java
java Tableau
*/
```

Exercice 2 : Générer trois nombres aléatoires compris entre 0 et 1000, puis vérifier si vous aves deux nombres pairs suivis par un nombre impair. Recommencer jusqu'à obtenir cette configuration. Enfin afficher le nombre d'essais nécessaires. Pour réaliser cet exercice, regarder la documentation de la classe Random() : https ://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/util/Random.html

Correction Exercice 2 :

```
//Dans un fichier Exercice2.java
import java.util.Random;
public class Exercice2{
    Random rand = new Random();
    int nbTries = 0;
    int int1, int2, int3;
    public static void main(String[] argv){
        do{
            int1 = rand.nextInt(1000);
            int2 = rand.nextInt(1000);
            int3 = rand.nextInt(1000);
            nbTries++;
         }while( !(int1%2 == 0 && int2%2 == 0 && int3%2 == 1));
        System.out.println("Number of tries: "+nbTries);
    }
}

/*En ligne de commande
javac Exercice2.java
java Exercice2
*/
```

Exercice 3 : Soit une température T (qu'on considerera comme un nombre entier) passé en argument du programme. Ecrire une classe qui affiche : froid si $T < 8$; frais si $8 < T < 17$; et chaud sinon.

Correction Exercice 3 : La correction est déjà ici un peu avancée par rapport au cours car elle fait intervenir les notions de classe et d'objet.

```
//Dans un fichier Temperature.java
public class Temperature {
        private int T = 0;

        public Temperature() {}

        public Temperature(int temp) {
            T = temp;
```

```java
        }

        public void printTemp() {
                String strAAfficher;
                if(T < 8) {
                        strAAfficher = "Il fait froid";
                }
                else if (T < 17) {
                        strAAfficher = "Il fait frais";
                }
                else {
                        strAAfficher = "Il fait chaud";
                }
                System.out.println(affichage);
        }
}

//Dans un fichier Test.java
public class Test {

        public static void main(String[] args){
            int val = args[0];
            Temperature temp = new Temperature(val);
            temp.printTemp();
        }
}

/*En ligne de commande
javac Temperature.java
javac Test.java
java Test 12
12 peut être remplacé par une autre valeur
*/
```

Exercice 4 :

1. Réécrire les opérations suivantes avec des parenthèses pour montrer l'ordre dans lequel s'effectuent les opérations.

```
- a + b / c
a *  - b + c % d
- c % d
- a / - b + c
- a / - (b + c)
```

2. Soit les déclarations suivantes :

```
byte b1 = 10, b2 = 20;
short s = 12;
long l = 320;
```

```
4  float f = 7.2f;
5  double d = 3.25;
```

Donner le type et la valeur des expressions suivantes :

```
1  b1 + b2 ;
2  s * b1 ;
3  l + s + b1*b2 ;
4  l*2./f ;
```

Correction Exercice 4 :

```
1  (- a) + (b / c)
2  (a *  (- b)) + (c % d)
3  (- c) % d
4  ((- a) / (- b)) + c
5  (- a) / (- (b + c))
```

```
1  b1 + b2 ; // 30 promotion en int
2  s * b1 ; // 120 promotion en int
3  l + s + b1*b2 ; //532 promotion en int de b1 et b2
4  //puis opérations de gauche à droite avec des conversions en long
5  l*2./f ; // 88,89 conversions en double
```

Exercice 5 : Qu'affiche ce code ? Expliquez pourquoi.

```
1  import java.util.Arrays;
2
3  public class Exo{
4          public static void incr(int i){
5              i++;
6          }
7
8          public static void incr(int[] tab){
9                  for(int i = 0; i< tab.length;i++)
10                         tab[i]++;
11         }
12
13         public static void main(String[] args){
14             int i = 0;
15             int j = i;
16             i = i+1;
17             System.out.println(i);
18             System.out.println(j);
19
20             int[] tab1 = new int[3];
21             for(int l = 0; l < 3; l++)
22                             tab1[l] = 0;
23
```

```
24              int[] tab2 = tab1;
25
26              tab1[1] = 1;
27
28              System.out.println(Arrays.toString(tab1));
29              System.out.println(Arrays.toString(tab2));
30
31              String s1 = "toto";
32              String s2 = s1;
33              s2 += "titi";
34
35              System.out.println(s1);
36              System.out.println(s2);
37
38              incr(j);
39              System.out.println(j);
40
41              incr(tab2);
42              System.out.println(Arrays.toString(tab1));
43          }
44  }
```

Correction Exercice 5 :

```
1  1 // i a été incrémenté
2  0 // pas j car int est un type valeur
3  [0, 1, 0] //tab1[1] = 1
4  [0, 1, 0] //tab2[1] aussi car int[] est un type adresse
5  toto
6  tototiti //le même raisonnement ne marche pas pour le type String qui est immuable
7  0 //le passage d'un paramètre d'une méthode se fait par valeur
8  //et int est un type valeur
9  [1, 2, 1] //int[] est un type adresse donc c'est l'adresse de
10 //tab1 et tab2 qui est aussi utilisé dans la méthode incr.
```

Vous pouvez en profiter pour faire un rappel sur la surcharge de méthode.