

**Université Sultan Moulay Slimane**

**Beni-Mellal**



**École nationale des sciences appliquées**

**Khouribga**



# COMPTE RENDU :

## TP2 JAVA POO

Filière : Informatique et Ingénierie des données (iid1)

Réalisé par :

- LAHMAMA Fatima-Zahraa
- EL FATHI Zakaria

Encadré par :

- Mr. GHERABI Noredline

# EXERCICE 1 :

Un programme qui demande à l'utilisateur un texte (T) et un mot (A) !

- Si le mot A= 'ENSA', le mot A sera supprimé du texte (T) !
- Si le mot A= 'FST', le mot A sera remplacé par le mot 'ENSA' dans le texte (T) !
- Si le mot A='ENCG', le mot A sera inséré à la fin du texte (T)

En s'appuyant sur l'instruction Switch :

*Code :*

```
package tp2;
import java.util.Scanner;
import java.lang.String;

public class Ex1{
    public static void main(String[] args) {
        String mot;
        String text;
        System.out.println("-----entrer un texte-----");
        Scanner T= new Scanner(System.in);
        text=T.nextLine();
        do {
            System.out.println("-----pour quitter taper '0'-----");
            System.out.println("-----entrer un mot-----");
            Scanner A = new Scanner(System.in);
            mot = A.nextLine();
            int z = mot.length();
            int l = text.length();
            switch(mot) {
                case "ENSA":

                    StringBuffer i = new StringBuffer(text);
                    String e="ENSA";
                    int pos=text.indexOf(e);
                    i.delete(pos , pos +z);
                    System.out.println("-----le nouveau texte est " +i);

                    break;

                case "FST":
                    String s2 = text.replace("FST" , "ENSA");
                    System.out.println("-----le nouveau texte est " +s2);
                    break;
            }
        }
    }
}
```

## TRAVAUX PRATIQUES 2

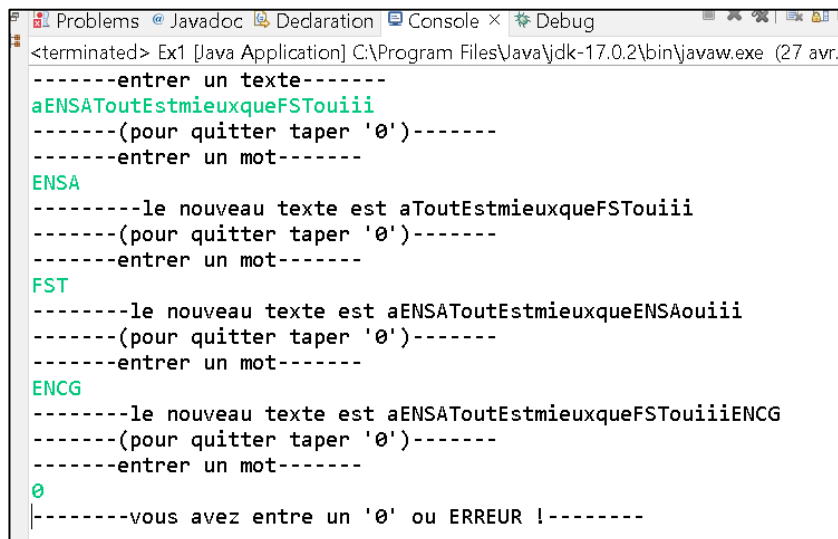
### – JAVA POO

```
        case "ENCG":
            StringBuffer m = new StringBuffer(text);
            m.append("ENCG");
            System.out.println("-----le nouveau texte est " +m);
            break;

        default:
            System.out.println("-----vous avez entre un '0' ou ERREUR ! ");
            break;
    }

    }while(mot.equals("0")!=true);
}}
```

*Exécution :*



```
<terminated> Ex1 [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (27 avr.
-----entrer un texte-----
aENSAToutEstmieuxqueFSTouiii
------(pour quitter taper '0')-----
-----entrer un mot-----
ENSA
-----le nouveau texte est aToutEstmieuxqueFSTouiii
------(pour quitter taper '0')-----
-----entrer un mot-----
FST
-----le nouveau texte est aENSAToutEstmieuxqueENSAouiii
------(pour quitter taper '0')-----
-----entrer un mot-----
ENCG
-----le nouveau texte est aENSAToutEstmieuxqueFSTouiiiENCG
------(pour quitter taper '0')-----
-----entrer un mot-----
0
|-----vous avez entre un '0' ou ERREUR !-----
```

## EXERCICE 2 :

Une procédure qui affiche tous les éléments d'un tableau passé en paramètre

Donc on crée une classe, ici de nom « **Tableau** » qui comprend la procédure d’affichage du tableau comme suit :

```
package tp2;
public class Tableau {
    public void afficheTab(int [] tab) {
        for (int i=0;i<tab.length;i++) {
            System.out.println("T["+i+"]="+tab[i]);
        }
    }
}
```

## TRAVAUX PRATIQUES 2

### – JAVA POO

Maintenant pour des valeurs d'un tableau de taille et valeurs saisies par l'utilisateur, on essaie de bien appelé la procédure dans la méthode main donc pour ce faire on commence par l'instanciation d'un objet de type « **Tableau** »

*Code :*

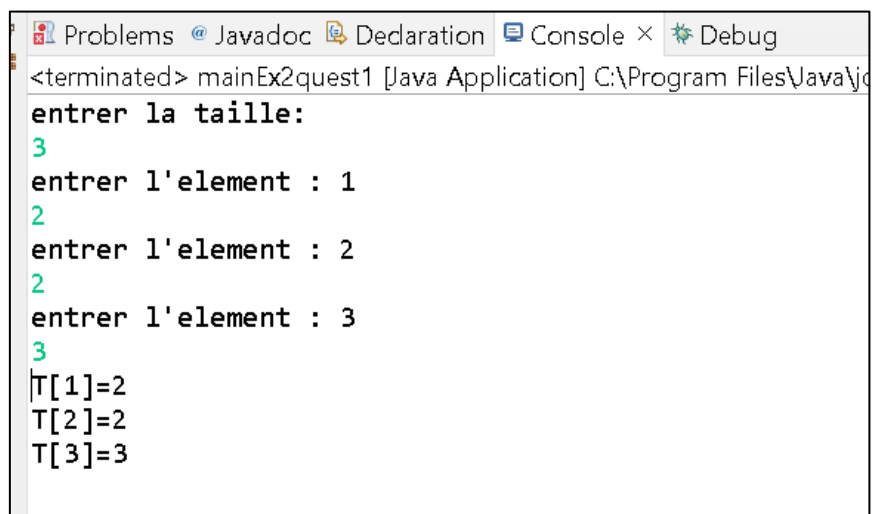
```
package tp2;

import java.util.Scanner;

public class mainEx2quest1 {

    public static void main(String[] args) {
        System.out.println("entrer la taille: ");
        Scanner s=new Scanner(System.in);
        int n;
        n=s.nextInt();
        int[] tab = new int[n];
        for (int i=0;i<n;i++) {
            System.out.println("entrer l'element : "+(i+1));
            tab[i]=s.nextInt();
        }
        s.close();
        Tableau T = new Tableau();
        T.afficheTab(tab);
    }
}
```

*Exécution :*



```
<terminated> mainEx2quest1 [Java Application] C:\Program Files\Java\j
entrer la taille:
3
entrer l'element : 1
2
entrer l'element : 2
2
entrer l'element : 3
3
T[1]=2
T[2]=2
T[3]=3
```

## EXERCICE 3 :

La classe Produit définie comme suit:

Produit
- Code : int + Marque : String - Prix : float
+ toString + equals

1. Implémentation en JAVA la classe « Produit »

```
package tp2;  
// Q1 IMPLEMENTATION DE LA CLASSE  
public class Produit{  
    private int Code;  
    public String Marque;  
    private float Prix;  
}
```

2. Création deux constructeurs de la classe (avec et sans paramètres)

Dans la même classe « Produit » on écrit :

```
//Q2 SANS PARAMETRES  
public Produit() {  
    Code = 1;  
    Marque = "noMarques";  
    Prix = 1;  
}  
//Q2 AVEC PARAMETRES  
public Produit(int C,String M,float P) {  
    Code = C;  
    Marque = M;  
    Prix = P;  
}
```

3. Redéfinition de la méthode toString permettant de retourner les propriétés de la classe « Produit »

```
//Q3 REDEFINITION DE LA METHODE toString  
public String toString() {  
    return "Code : "+Code+"\nMarque : "+Marque+"\nPrix : "+Prix;  
}
```

## TRAVAUX PRATIQUES 2

### – JAVA POO

#### 4. Les getters et les setters de la classe « Produit »

```
//Q4 getters and setters
public int getCode() {
    return Code;
}
public void setCode(int code) {
    Code = code;
}
public float getPrix() {
    return Prix;
}
public void setPrix(float prix) {
    Prix = prix;
}
```

#### 5. Création de la classe principale puis ajout de la méthode principale « main »

```
package tp2;

//Q5
public class mainexo3 {
    public static void main(String[] args) {
```

#### 6. Dans la même classe principale on crée la méthode statique « CalcPrix » qui permet de calculer la somme des prix d'un tableau des produits (avec le getter getPrix) :

```
//Q6
static float CalcPrix(Produit[] prod) {
    float somme = 0;
    for (int i = 0 ; i<prod.length ; i++) {
        somme += prod[i].getPrix();
    }
    return somme;
}
```

#### 7. Dans la même fonction principale on crée un tableau des produits, les valeurs des attributs sont saisies par l'utilisateur (avec les setters) :

*Code :*

```
//Q7
System.out.println("Saisir la taille du tableau des produits ");
Scanner scan=new Scanner(System.in);
Produit [] p = new Produit[scan.nextInt()];
for (int i = 0 ; i< p.length ; i++) {
    p[i] = new Produit();
    System.out.println("\nCode :");
    p[i].setCode(scan.nextInt());
```

## TRAVAUX PRATIQUES 2

### – JAVA POO

```
System.out.println("prix : ");
p[i].setPrix(scan.nextFloat());
scan.nextLine();
System.out.println("marque : ");
p[i].Marque = scan.nextLine();
}
```

*Exécution :*

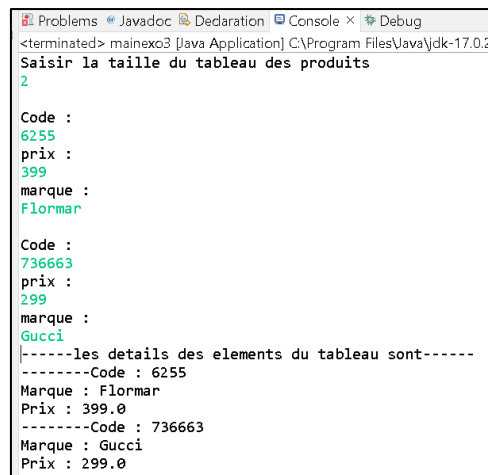


```
<terminated> mainexo3 [Java Application] C:\Program Files\Java\
Saisir la taille du tableau des produits
3
Code :
455
prix :
99
marque :
Chanel
Code :
635
prix :
199
marque :
Gucci
Code :
726
prix :
399
marque :
Flormar
```

8. Affichage des valeurs des attributs de chaque produit, A l'aide de la méthode toString() :

```
//Q8
System.out.println("les details des elements du tableau sont:");
for (int i = 0 ; i<p.length ; i++) {
    System.out.println(p[i].toString());
}
```

*Exécution :*



```
<terminated> mainexo3 [Java Application] C:\Program Files\Java\jdk-17.0.2
Saisir la taille du tableau des produits
2
Code :
6255
prix :
399
marque :
Flormar
Code :
736663
prix :
299
marque :
Gucci
|-----les details des elements du tableau sont-----
|-----Code : 6255
|Marque : Flormar
|Prix : 399.0
|-----Code : 736663
|Marque : Gucci
|Prix : 299.0
```

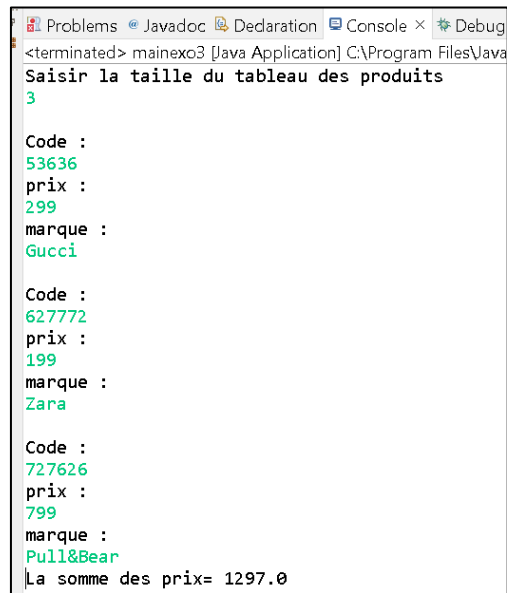
9. Affichage de la somme des prix de tous les produits, En utilisant la méthode « CalcPrix » :

```
//Q9
System.out.println("La somme des prix= "+CalcPrix(p));
```

## TRAVAUX PRATIQUES 2

### – JAVA POO

*Exécution :*



```
<terminated> mainexo3 [Java Application] C:\Program Files\Java
Saisir la taille du tableau des produits
3

Code :
53636
prix :
299
marque :
Gucci

Code :
627772
prix :
199
marque :
Zara

Code :
727626
prix :
799
marque :
Pull&Bear
La somme des prix= 1297.0
```

10. A l'aide de la méthode ***equals***, on cherche les produits ayant même état (càd mêmes valeurs des attributs.), pour ce faire il faut redéfinir la méthode ***equals*** dans la classe « ***Produit*** » comme suit :

```
//Q10
public boolean equals(Produit P) {
    if (this.Prix == P.Prix && this.Marque.equals(P.Marque) && this.Code ==
P.Code) {
        return true;
    }
    else {
        return false;
    }
}
```

On passe maintenant à la classe main pour continuer l'exercice

*Code :*

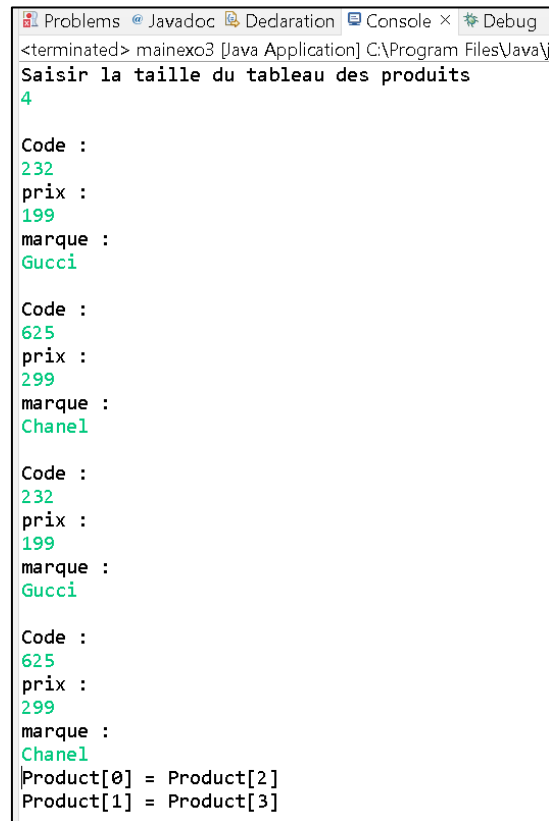
```
for (int i = 0; i < p.length/2;i++) {
    for(int j = i+1; j< p.length;j++) {
        if (p[i].equals(p[j])) {
            System.out.println("Product["+i+"] = Product["+j+"]");
        }
    }
}
```



## TRAVAUX PRATIQUES 2

### – JAVA POO

*Exécution :*



```
<terminated> mainexo3 [Java Application] C:\Program Files\Java\
Saisir la taille du tableau des produits
4
Code :
232
prix :
199
marque :
Gucci
Code :
625
prix :
299
marque :
Chanel
Code :
232
prix :
199
marque :
Gucci
Code :
625
prix :
299
marque :
Chanel
Product[0] = Product[2]
Product[1] = Product[3]
```

# FIN !