



École nationale des sciences appliquées  
Khouribga

Informatique et Ingénierie des Données  
IID1

# COMPTE RENDU TP1

## POO- JAVA

ENCADRE PAR  
MR. GHERABI NOREDDINE

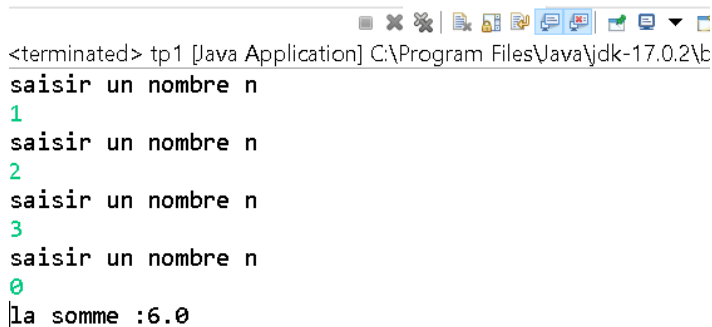
REALISE PAR  
LAHMAMA FATIMA-ZAHRAA  
EL FATHI ZAKARIA

**1/ Un programme qui demande a l'utilisateur de saisir une série de nombres réels. Pour arrêter la saisie on entre la valeur 0. Puis le programme affiche la somme de la partie entière des valeurs saisie.**

```
import java.util.Scanner; // import de la classe Scanner
public class tp1 {

    public static void main(String[] args) {
        float somme; // declaration de la variable somme de type reel
        float n; // declaration de la variable n de type reel
        somme = 0; // initialisation de somme avec la valeur 0
        Scanner p = new Scanner (System.in); // declaration de la variable p de
type scanner
        do
        { System.out.println("saisir un nombre n"); // message d'interaction
avec l'utilisateur
            n = p.nextFloat(); // reception de la valeur saisie dans n
            somme = (int)somme + n; // incrementation de somme en ajoutant n et
cast de somme pour obtenir la partie entiere
        }while (n != 0); // condition d'arret (lorsque n=0 )
        System.out.println("la somme :"+somme);
    }
}
```

**Exécution :**



```
<terminated> tp1 [Java Application] C:\Program Files\Java\jdk-17.0.2\bin
saisir un nombre n
1
saisir un nombre n
2
saisir un nombre n
3
saisir un nombre n
0
la somme :6.0
```

**2/ Un programme qui affiche le carré des nombres entiers allant de -3 a +3 en utilisant  
CONTINUE :**

```
public class qst2 {

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

```

    int i=-3; //declaration et initialisation de i de type entier avec la
valeur -3
    int carre;
    while(i>=-3) { // le programme entre la boucle tant que la valeur de i
est superieure a -3
        carre = i*i;

        if(i>3) { // condition pour utiliser l'instruction continue;
            continue;// on continuer a sortir des iterations ou i>0
        }
        i++;//incrementation de i avec une valeur de 1
        System.out.println(carre);
    }
}
}

```

Exécution :

```

qst2 [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe
9
4
1
0
1
4
9

```

**3/ Un programme qui affiche la racine carrée des nombres entiers pairs allant de 0 a 10 :**

```

public class qst3 {
    public static void main(String[] args) {
        int i ;
        for(i=0;i<=10;i++) { //boucle d'iteration qui commence de 0 et se termine
pour i=10 avec une incrementation de i de 1
            if(i%2 == 0) { // condition pour verifier que i est paire(le reste de
la division sur 2 est nul)
                System.out.println("la racine carree de "+i+" est:"+Math.sqrt(i));
            }
        }
    }
}

```

Exécution :

```

<terminated> qst3 [Java Application] C:\Program Files\Java\jdk-1
la racine carree de 0 est:0.0
la racine carree de 2 est:1.4142135623730951
la racine carree de 4 est:2.0
la racine carree de 6 est:2.449489742783178
la racine carree de 8 est:2.8284271247461903
la racine carree de 10 est:3.1622776601683795

```

**4/ Un programme qui demande a l'utilisateur de saisir une serie de nombres entiers positifs. Pour arrêter la saisie, on entre 0. Puis le programme affiche le maximum, le minimum et la somme des entiers saisis :**

```
import java.util.Scanner; // import de la classe Scanner
public class qst4 {
    public static void main(String[] args) {
        int n;
        int i=1;// declaration et initialisation de i avec 1
        int max;
        int min;
        int somme;
        Scanner p = new Scanner(System.in);// declartion de p de type scanner
        System.out.println("entrer n");
        n = p.nextInt();// stocker la valeur entree dans n
        max = n;// initialisation de max par la valeur initiale de n
        min = n;//initialisation de min par la valeur initiale de n
        somme = n;//initialisation de somme par la valeur initiale de n
        while(n != 0) {// boucle qui itere tant que n differente de 0
            System.out.println("entrer n");
            n = p.nextInt();// stocker la valeur entrer dans n
            if (n>max) {// condition si qui s'execute seulement si n >max;
                max = n;// on affecte la value n a la variable max
            }
            if (n<min && n != 0) {// condition si la value de n est inferieure
a min et n differe de 0
                min = n;// affectation de n a la variable min
            }
            somme += n;// on incremente la somme a chaque fois d'une valeur n
        }
        System.out.println("\n la somme est "+somme+"\n le max est
"+max+"\n le min est "+min);// affichage des valeur prises
    }
}
```

**Exécution :**

```
<terminated> qst4 [Java Application] C:\Program Files\Java\jdk
entrer n
1
entrer n
2
entrer n
3
:entrer n
0
|
la somme est 6
le max est 3
le min est 1
```

**5/ Un programme qui permet d'initialiser 3 valeurs puis calcule le résultat selon un opérateur arithmétique, tout en utilisant l'instruction SWITCH:**

```
import java.util.Scanner; // import de la classe scanner
public class question5 {

    public static void main(String[] args) {
        int a, b, c; //declaration de a,b,c de types entiers
        int result;

        String operation; // declaration de la variable operation de type
chaîne de caracteres
        System.out.println("entrer a,b,c");
        Scanner p=new Scanner(System.in); // declaration de p de type Scanner

        Scanner u=new Scanner(System.in);
        a=p.nextInt();// on stocke la 1ere valeur entree dans a
        b=p.nextInt();// on stocke la 2eme valeur entree dans a
        c=p.nextInt();
        System.out.println("entrer l operation");
        operation=u.nextLine();// on stocke la valeur entree dans la variable
operation
        switch(operation) { // boucle qui depend de la valeur de la variable
operation
            case "+": // le cas ou operation ='+'
                result = a+b+c; // on renvoie la somme des trois entiers
                System.out.println("resultat =" +result);
                break;// on quitte la boucle apres affichage
            case "*":
                result = a*b*c;
                System.out.println("resultat =" +result);
                break;
            case "-":
                result = a-b-c;
                System.out.println("resultat =" +result);
                break;
        }
    }
}
```

**Exécution :**

```
<terminated> question5 [Java Application] C:\Program File
entrer a,b,c
1
2
3
entrer l operation
*
|resultat =6
```

6/ Soit la classe *Produit*:

**Classe Produit {Code int, Intitule String, Prix Float, Marque String}**

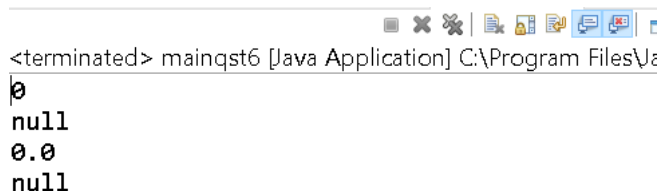
*a/ Implémentation de la classe <<Produit>>*

```
public class Produit {  
    static int Code;  
    static String Intitule;  
    static float Prix;  
    static String Marque;  
}
```

*b/ création d'un constructeur par défaut*

```
public class mainqst6 {  
    public static void main(String[] args) {  
        Produit produit1 = new Produit();  
        System.out.println(produit1.Code);  
        System.out.println(produit1.Intitule);  
        System.out.println(produit1.Prix);  
        System.out.println(produit1.Marque);  
    }  
}
```

**Exécution :**

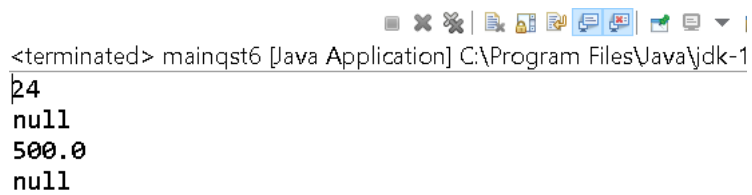


```
<terminated> mainqst6 [Java Application] C:\Program Files\Java\jre\bin\java.exe  
0  
null  
0.0  
null
```

*c/ création d'un constructeur qui utilise le Code et le Prix comme argument :*

```
public Produit() {  
    Code = 24;  
    Prix = 500;  
}  
public Produit(int a ,float z) {  
    Code = a;  
    Prix = z;  
}
```

**Exécution :**



```
<terminated> mainqst6 [Java Application] C:\Program Files\Java\jdk-1
24
null
500.0
null
```


*d/ création d'une méthode qui accède au prix et du produit :*

```
public static void price(float p) {
    p = Produit.Prix;
    System.out.println(p);
}
```

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

    price(Prix = 500);
}
```

**Exécution :**



```
<terminated> Produit [Java Application] C:\Program Files\Java\jdk-1
500.0
```

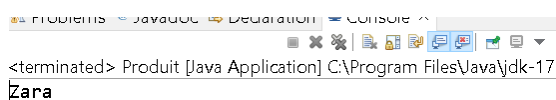
*e/ création d'une méthode qui accède a la marque du produit :*

```
public static void brand(String m) {
    m = Produit.Marque;
    System.out.println(m);
}
```

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

    brand(Marque = "Zara");
}
```

**Exécution :**



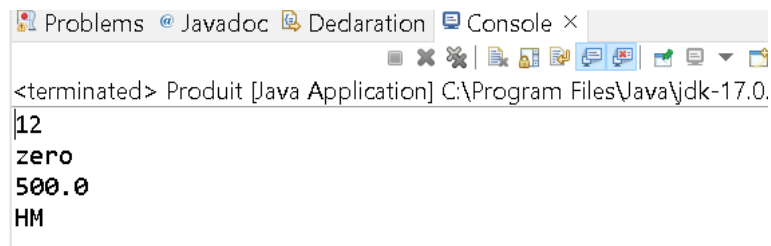
```
<terminated> Produit [Java Application] C:\Program Files\Java\jdk-17
Zara
```

*f/ création d'une méthode appelée 'afficher' qui affiche les valeurs des attributs du produit :*

```
public static void afficher(int n,String b,float c,String x) {  
    n = Produit.Code;  
    b = Produit.Intitule;  
    c = Produit.Prix;  
    x = Produit.Marque;  
    System.out.println(n);  
    System.out.println(b);  
    System.out.println(c);  
    System.out.println(x);  
}
```

```
public static void main(String[] args) {  
  
    afficher(Code = 12,Intitule = "zero" ,Prix = 500 , Marque = "HM");  
}
```

**Exécution :**



The screenshot shows a Java IDE window with a console tab. The console output is as follows:

```
<terminated> Produit [Java Application] C:\Program Files\Java\jdk-17.0.  
12  
zero  
500.0  
HM
```

*g/ création d'une méthode appelée 'augmenter' qui augmente le prix par un montant M :*

```
public static void augmenter(float M) {  
    M = Prix * 5/100;  
    Prix = Prix + M;  
    System.out.println(Produit.Prix);  
}
```

```
public static void main(String[] args) {  
    augmenter(Prix = 450);  
}
```



Exécution :

```
<terminated> Produit [Java Application] C:\Progra  
472.5
```

*h/ création d'une méthode appelée 'diminuer' qui retire N% du prix :*

```
public static void diminuer(float N) {  
    N = Prix * 7 / 100;  
    if (Prix > 1000) {  
        Prix = Prix - N;  
    }  
    System.out.println(Produit.Prix);  
}
```

```
public static void main(String[] args) {  
    diminuer(Prix = 1200);  
}
```

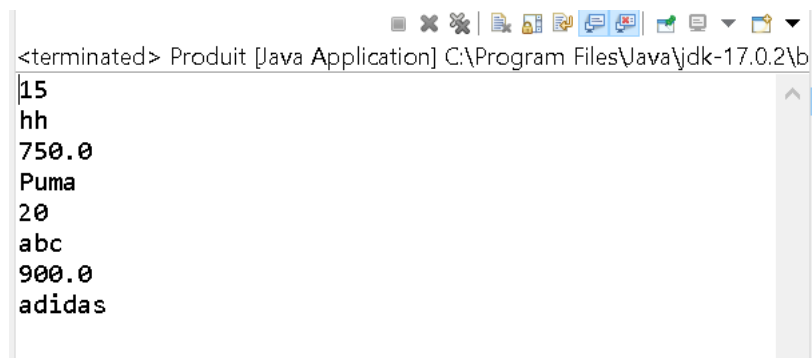
Exécution :

```
<terminated> Produit [Java Application] C:\Program Files\Java\jdk-17  
1116.0
```

*i/ création de deux objets 'P1' et 'P2' de la classe 'Produit':*

```
public static void main(String[] args) {  
    Produit P1 = new Produit();  
    P1.Code = 15;  
    P1.Intitule = "hh";  
    P1.Prix = 750;  
    P1.Marque = "Puma";  
    System.out.println(P1.Code);  
    System.out.println(P1.Intitule);  
    System.out.println(P1.Prix);  
    System.out.println(P1.Marque);  
    Produit P2 = new Produit();  
    P2.Code = 20;  
    P2.Intitule = "abc";  
    P2.Prix = 900;  
    P2.Marque = "adidas";  
    System.out.println(P2.Code);  
    System.out.println(P2.Intitule);  
    System.out.println(P2.Prix);  
    System.out.println(P2.Marque);  
}
```

Exécution :



```
<terminated> Produit [Java Application] C:\Program Files\Java\jdk-17.0.2\bin
15
hh
750.0
Puma
20
abc
900.0
adidas
```

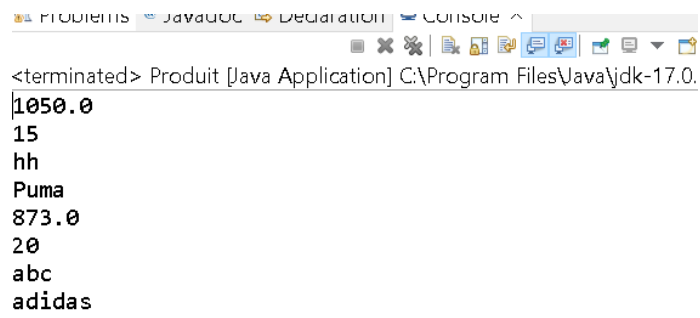
*j/ création de deux fonctions l'une ajoute 300dh au prix de produit1 l'autre retranche 3% au prix de produit2 :*

```
public void add(float l) {
    Prix = Prix + 300;
    System.out.println(Produit.Prix);
}
public void retrancher(float k) {
    k = Prix * 3/100;
    Prix = Prix -k;
    System.out.println(Produit.Prix);
}
```

*j/Les propriétés des produits P1 ET P2 après avoir ajoute 300dh au prix du premier et retrancher 3% au prix du deuxième:*

```
public static void main(String[] args) {
    Produit P1 = new Produit();
    P1.Code = 15;
    P1.Intitule ="hh";
    P1.add(Prix = 750);
    P1.Marque = "Puma";
    System.out.println(P1.Code);
    System.out.println(P1.Intitule);
    System.out.println(P1.Marque);
    Produit P2 = new Produit();
    P2.Code = 20;
    P2.Intitule ="abc";
    P2.retrancher(Prix = 900);
    P2.Marque = "adidas";
    System.out.println(P2.Code);
    System.out.println(P2.Intitule);
    System.out.println(P2.Marque);}
```

## Exécution :



The screenshot shows a Java IDE with a console window titled "Console". The console output is as follows:

```
<terminated> Produit [Java Application] C:\Program Files\Java\jdk-17.0.  
1050.0  
15  
hh  
Puma  
873.0  
20  
abc  
adidas
```

# FIN !