

Corrigé d'exercices de TD POO

Série n°3 – Modificateurs et Niveaux de visibilité

S. BOUKHEDOUMA

**USTHB – FEI – département d'Informatique
Laboratoire des Systèmes Informatiques -LSI**

sboukhedouma@usthb.dz

Exercice 3 – Série n°3

Exercice 3

Dans cet exercice, on s'intéresse à la manipulation de polynômes à une seule variable. Un polynôme est décrit par la somme d'un ensemble de monômes. Un monôme est décrit par un coefficient réel et un exposant entier (exemple $(-4, 3)$ représente le monôme $-4x^3$).

1. Donner l'implémentation d'une classe **Monôme** comportant un constructeur, une méthode *lire* pour la saisie au clavier et une méthode *afficher* qui affiche le monôme sous la forme usuelle, par exemple $-4x^3$.
2. Dans la même classe, écrire une méthode *somme* qui calcule le monôme représentant la somme de deux monômes de même degré (exposant).
3. Ajouter à la classe, une méthode *valeur* qui calcule la valeur du monôme pour une valeur réelle x donnée ($x \neq 0$), par exemple si $x = 2$, la valeur du monôme $-4x^3$ est égale à -32 .

Exercice 3 – Série n°3

```
import java.util.Scanner;

public class Monome {
    private float coef;
    private int expt;

    // constructeurs

    public Monome() { }

    public Monome (float coef, int expt) {
        this.coef = coef;  this.expt = expt; }

    // méthodes

    public void lire() {
        Scanner e = new Scanner(System.in);
        System.out.print("Saisir le coefficient: ");
        this.coef = e.nextFloat();
        System.out.print("Saisir l'exposant: ");
        this.expt = e.nextInt(); }
}
```

Exercice 3 – Série n°3

```
public void afficher()
{   System.out.print(this.coef + "x" + this.expt); }

public Monome somme(Monome m2) {
    return new Monome(this.coef + m2.Coef, this.expt);
}

public float valeur(float x) {
    return (this.coef * (float)Math.pow(x, this.expt));
}

                                // les accesseurs

public float getCoef () {return (coef); }
public void setCoef (float co) {coef = co; }
public int getExpt () {return (expt); }
public void setExpt (int ex) {expt = ex; }  } // fin de la classe Monome
```

Exercice 3 – Série n°3

Exercice 3

Un polynôme est décrit par un ensemble de monômes.

4. Donner l'implémentation d'une classe **Polynôme** comportant un attribut degré (entier) et un tableau de monômes de taille **degré+1**. La classe polynôme doit comporter un **constructeur**, une méthode **lire** pour la saisie de tous les monômes constituant le polynôme et une méthode **afficher** qui affiche le polynôme.

NB: On suppose que dans le tableau, les monômes sont rangés dans l'ordre décroissant de leurs degrés.

5. Ecrire une méthode **somme** qui calcule le *polynôme somme* de deux polynômes.

6. Ajouter une méthode **valeur** qui calcule la valeur du polynôme pour une valeur x donnée.

Exercice 3 – Série n°3

```
import java.util.Scanner;

public class Polynome {
    private int degré;
    private Monome [] M; // tableau de Monômes
                        // constructeurs

    public Polynome() { }

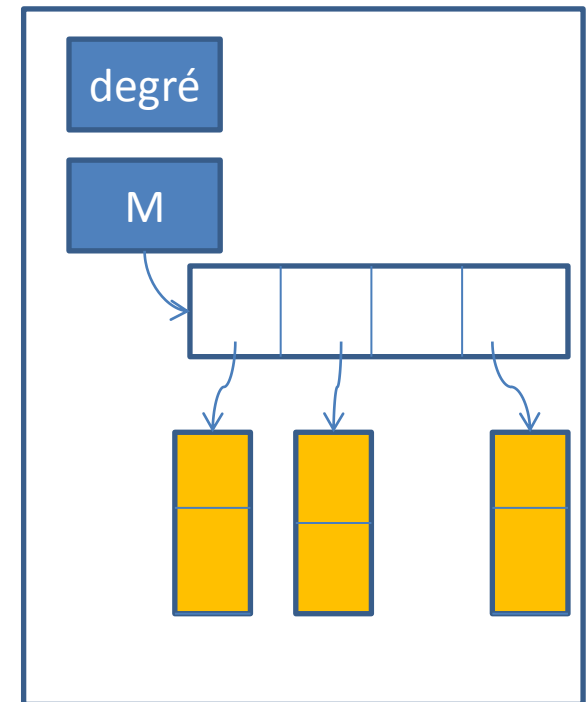
    public Polynome (int degré) {
        this.degré = degré;

        // création du tableau de Monômes
        M = new Monome [degré+1];

        // création des objets Monômes
        for (int i = 0; i <= degré; i++)
            M[i] = new Monome();}
```



Le losange désigne
l'agrégation (composition)



Exercice 3 – Série n°3

```
public void lire() {  
    Scanner e = new Scanner(System.in);  
    System.out.print("Introduire les monômes  
                    dans l'ordre décroissant des exposants ");  
    for (int i = 0; i <= degré; i++)  
        M[i].lire(); // méthode lire de la classe Monôme  
}  
  
public void afficher() {  
    System.out.println("le polynôme" );  
        M[0].affiche(); // méthode affiche de la classe Monôme  
    for (int i = 1; i <= degré; i++)  
        System.out.println("+" ); M[i].affiche(); // méthode affiche de la classe Monôme  
}
```

Exercice 3 – Série n°3

```
public Polynome somme (Polynome P2) {  
    Polynome S; int i, j, k;  
    // le polynôme Somme aura le degré max  
    int dmax = Math.max (this.degré, P2.degré);  
    S = new polynôme (dmax);      // création du polynôme Somme  
    i =j=k=0;  
    while (i<= this.degré && j <= P2.degré)  
        { if (M[i].getExpt () > P2.M[j].getExpt()) // comparer les exposants  
            {S.M[k] = new Monome (M[i].getCoef(), M[i].getExpt()); i++;}  
        else  
            if (M[i].getExpt () < P2.M[j].getExpt())  
                {S.M[k] = new Monome (P2.M[j].getCoef(), P2.M[j].getExpt()); j++;}  
            else // égalité des exposants  
                {S.M[k] = M[i].somme (P2.M[j]); // méthode somme de  
                                                         la classe Monôme  
                i++; j++;}  
            k++; } } }
```


Exercice 3 – Série n°3

```
//suite...
```

```
// le reste des monômes
```

```
while (i<= this.degré)
```

```
{ S.M[k] = new Monome (M[i].getCoef(), M[i].getExpt());  
  i++; k++;}
```

```
while (j <= P2.degré)
```

```
{S.M[k] = new Monome (P2.M[j].getCoef(), P2.M[j].getExpt());  
  j++; k++;}
```

```
return (S);
```

```
}
```

Exercice 3 – Série n°3

```
public float valeur(float x) {  
    float v = 0;  
    for (int i = 0; i <= degré; i++)  
        v += M[i].valeur(x); // méthode valeur de la classe Monome  
    return (v);  
}  
  
                // les accesseurs  
  
public int getDegré () {return (degré); }  
public Monome [] getTabMonome() {return (M);}  
public Monome getMonome(int i) {return (M[i]);}  
                // retourne la référence du monôme d'ordre i  
  
} // fin de la classe Polynôme
```

Exercice 3 – Série n°3

Exercice 3

7. Ecrire un programme qui *crée* deux polynômes P1 et P2 de degrés respectifs n1 et n2, calcule et *affiche* leur *somme* sous forme d'un polynôme. Calcule ensuite, la *valeur* du polynôme somme pour une valeur x donnée.

Exercice 3 – Série n°3

```
import java.util.Scanner;

class ProgPolynome
{
    public static void main ( String args[])
    {
        Scanner e = new Scanner (System.in);
        System.out.println ("donner les degrés des polynômes:");
        int n1 = e.nextInt(); int n2 = e.nextInt();

        // création des polynômes
        Polynome P1 = new Polynôme (n1);
        Polynome P2 = new Polynôme (n2);

        System.out.println("Introduire le polynôme P1" ); P1.lire();
        System.out.println("Introduire le polynôme P2" ); P2.lire();
    }
}
```

Exercice 3 – Série n°3

// calcul de la somme

Polynome S = P1.somme(P2);

// méthode somme de la classe Polynome

System.out.println("le polynôme somme est:"); **S.affiche();**

// calcul de la valeur pour x donné

System.out.println("donner la valeur x");

float x = e.nextFloat();

System.out.println("la valeur du polynôme pour x est:" +

S.valeur(x));

// méthode valeur de la classe Polynome

}} // fin de la classe ProgPolynome