

Calculatrice de polynômes

Projet de programmation orientée objet en Java

Wael BELHADJ
Matthieu POIDEVIN
Nils JEAN-ZÉPHIRIN

BUT Science des Données
IUT de Sophia Antipolis

Table des matières

1	Introduction	2
2	Analyse du problème	2
2.1	Objectifs fonctionnels	2
2.2	Contraintes	2
3	Choix de modélisation	2
3.1	Classe Monome	2
3.2	Classe Polynome	2
3.3	Classe LecteurPolynome	3
3.4	Classe Main	3
4	Solution algorithmique	3
4.1	Ajout d'un monôme	3
4.2	Soustraction	3
4.3	Multiplication	3
5	Implémentation Java	3
6	Tests et exemples	4
6.1	Exemple	4
7	Limites et améliorations	4
7.1	Limites	4
7.2	Améliorations possibles	4
8	Conclusion	4

1 Introduction

Ce projet a pour objectif de concevoir une calculatrice de polynômes multivariés en Java. Le programme permet de lire plusieurs polynômes saisis par l'utilisateur, d'effectuer des opérations algébriques (addition, soustraction et multiplication) et d'afficher le résultat final sous une forme simplifiée.

Ce travail s'inscrit dans le cadre du cours de programmation orientée objet et permet de mettre en pratique les notions de modélisation, d'algorithmique et de manipulation de structures de données.

2 Analyse du problème

2.1 Objectifs fonctionnels

Le programme doit :

- lire plusieurs polynômes saisis sous forme textuelle ;
- représenter ces polynômes sous une forme exploitable par le programme ;
- appliquer une suite d'opérations algébriques ;
- simplifier automatiquement les résultats ;
- afficher le polynôme final.

2.2 Contraintes

Les contraintes principales sont :

- rester dans un cadre pédagogique simple ;
- ne pas utiliser de bibliothèques de calcul symbolique ;
- gérer des polynômes comportant plusieurs variables.

3 Choix de modélisation

3.1 Classe Monome

Un monôme est défini par :

- un coefficient réel ;
- une ou plusieurs variables ;
- les puissances associées à ces variables.

Les variables et puissances sont stockées dans des tableaux, accompagnés d'un entier indiquant le nombre réel de variables utilisées.

3.2 Classe Polynome

Un polynôme est modélisé comme une collection de monômes. Il est représenté par :

- un tableau dynamique de monômes ;
- un entier indiquant le nombre de monômes présents.

Une méthode centrale permet d'ajouter un monôme tout en fusionnant automatiquement les monômes similaires.

3.3 Classe LecteurPolynome

Cette classe est chargée de convertir une chaîne de caractères saisie par l'utilisateur en une structure de type **Polynome**. La lecture est volontairement simple afin de rester lisible et compréhensible.

3.4 Classe Main

La classe principale gère l'interaction avec l'utilisateur :

- lecture des polynômes ;
- lecture des opérations ;
- affichage du résultat final.

4 Solution algorithmique

4.1 Ajout d'un monôme

Lorsqu'un monôme est ajouté à un polynôme, le programme vérifie s'il existe déjà un monôme similaire. Si c'est le cas, les coefficients sont additionnés.

Algorithm 1: Ajout d'un monôme avec fusion

```
Input: Polynôme P, Monôme m
for i ← 0 to P.taille – 1 do
    if P.monomes[i] est similaire à m then
        additionner les coefficients
    return
```

Ajouter m à la fin du tableau

4.2 Soustraction

La soustraction repose sur l'identité mathématique :

$$A - B = A + (-B)$$

Chaque monôme du second polynôme est donc ajouté avec un coefficient opposé.

4.3 Multiplication

La multiplication utilise la distributivité :

$$(A_1 + A_2)(B_1 + B_2) = A_1B_1 + A_1B_2 + A_2B_1 + A_2B_2$$

5 Implémentation Java

La séparation en classes permet de distinguer clairement la représentation mathématique (**Monome**, **Polynome**) de la logique d'entrée/sortie (**LecteurPolynome**, **Main**). Cette organisation facilite la lecture et la maintenance du code.

```
for (int i = 0; i < this.taille; i++) {
    for (int j = 0; j < p.taille; j++) {
        r.addMonome(this.monomes[i].multiply(p.monomes[j]));
    }
}
```

6 Tests et exemples

6.1 Exemple

Entrée :

```
Polynome 1 : x^2 + x + 1
Polynome 2 : x + 1
Opération : +
```

Résultat obtenu :

$$x^2 + 2x + 2$$

7 Limites et améliorations

7.1 Limites

- pas de gestion des parenthèses ;
- lecture volontairement simple des expressions ;
- nombre limité de variables par monôme.

7.2 Améliorations possibles

- améliorer la robustesse du lecteur d'expressions ;
- trier les monômes pour un affichage plus standard ;
- ajouter d'autres opérations algébriques.

8 Conclusion

Ce projet a permis de concevoir une calculatrice de polynômes multivariés en Java en respectant une approche orientée objet claire et pédagogique. Les structures de données et les algorithmes mis en place permettent d'effectuer correctement les opérations demandées tout en restant accessibles et lisibles.