

## PCOO – TD 6

Nous souhaitons écrire un programme qui prend en paramètre une expression en notation postfixée composée de nombres, de '+' et de '-' et qui calcule la valeur de l'expression. En notation postfixée (ou polonaise inversée), on place l'opérateur à droite des deux opérandes. Par exemple, l'expression  $(2 * 5) + 3 * (4 + 7)$  s'écrit en notation postfixée `2 5 * 3 4 7 + * +` (les parenthèses sont inutiles).

Pour réaliser cette calculatrice, nous partons des programmes sur les formules des TD précédents. Par conséquent, nous supposons que les classes du diagramme 1 sont implémentées.

1. Ajoutez une classe **Constant** afin de représenter les constantes. Cette classe implémente l'interface **Formula**. Dans l'expression `2 5 * 3 4 7 + * +`, les valeurs 2, 5, 3, 4 et 7 seront représentées par des instances de cette classe.
2. L'expression est passée en paramètre du programme de façon à avoir le comportement suivant :

```
$ java Calculator 2 5 * 3 4 7 + * +
43.0
```

Implémentez la classe **Calculator** en instanciant directement les classes **Constant**, **Product** et **Sum**.

3. Que doit-on modifier dans la classe **Calculator** si on utilise la version avec le patron de conception **Strategy** pour réaliser le produit et la somme (voir le diagramme 2). Est-ce satisfaisant ?
4. Écrivez une fabrique afin de séparer le module de calcul du module de lecture. La fabrique de formules est une classe qui permet de construire une constante, un produit ou une somme. Vous devez implémenter le diagramme 3.
5. Mettez en place une fabrique abstraite afin de faire cohabiter les deux implémentations du module de calcul (représentées par les diagrammes 1 et 2). Vous devez implémenter le diagramme 4 et le connecter aux autres classes.
6. Modifiez la classe **Calculator** afin qu'il soit possible de passer d'une implémentation du module de calcul à l'autre en modifiant seulement une ligne de la classe **Calculator**.

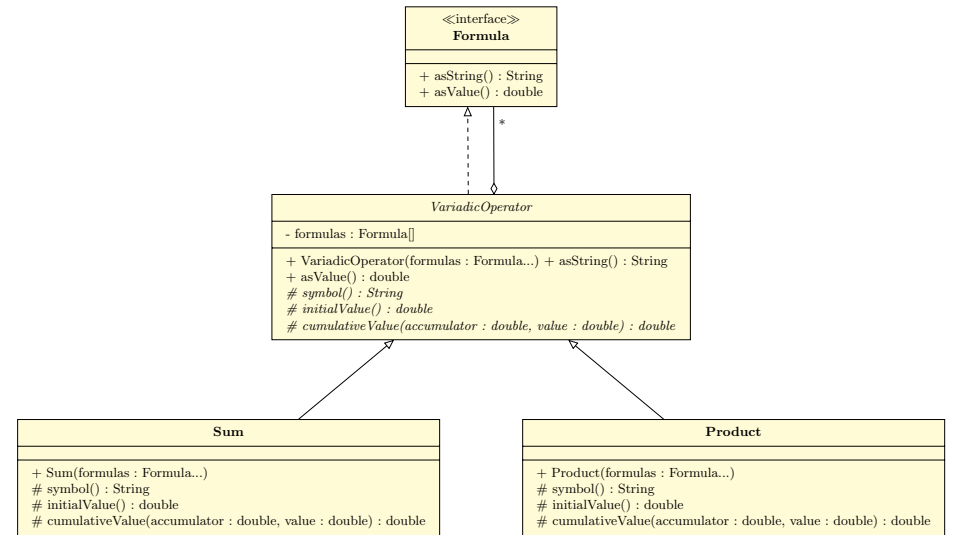


Diagramme 1 – Avec des extensions

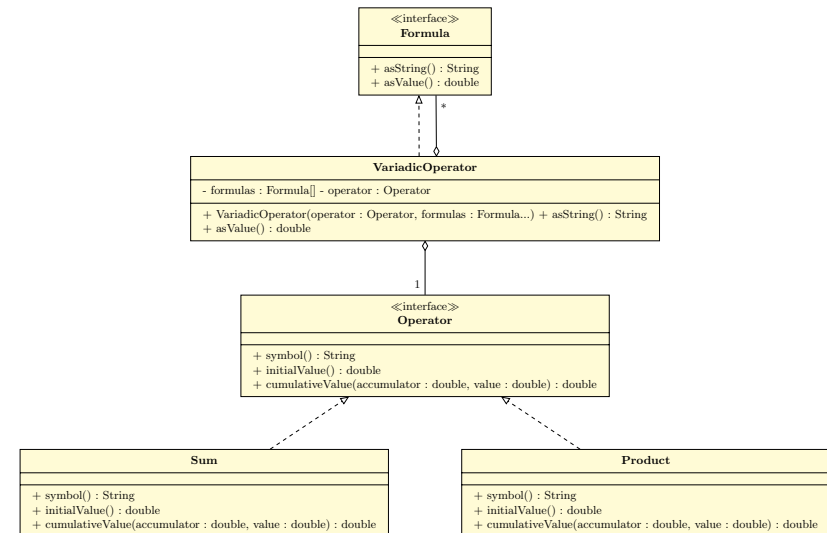


Diagramme 2 – Avec des stratégies

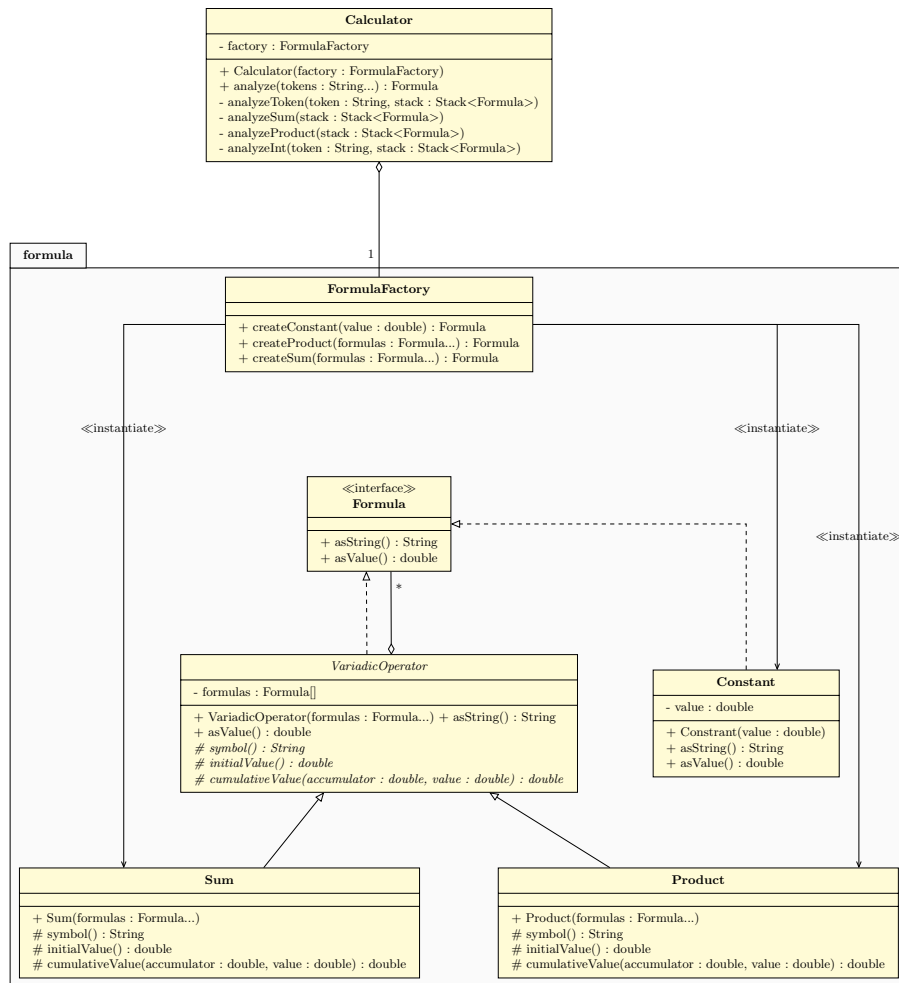


Diagramme 3 – Avec une fabrique

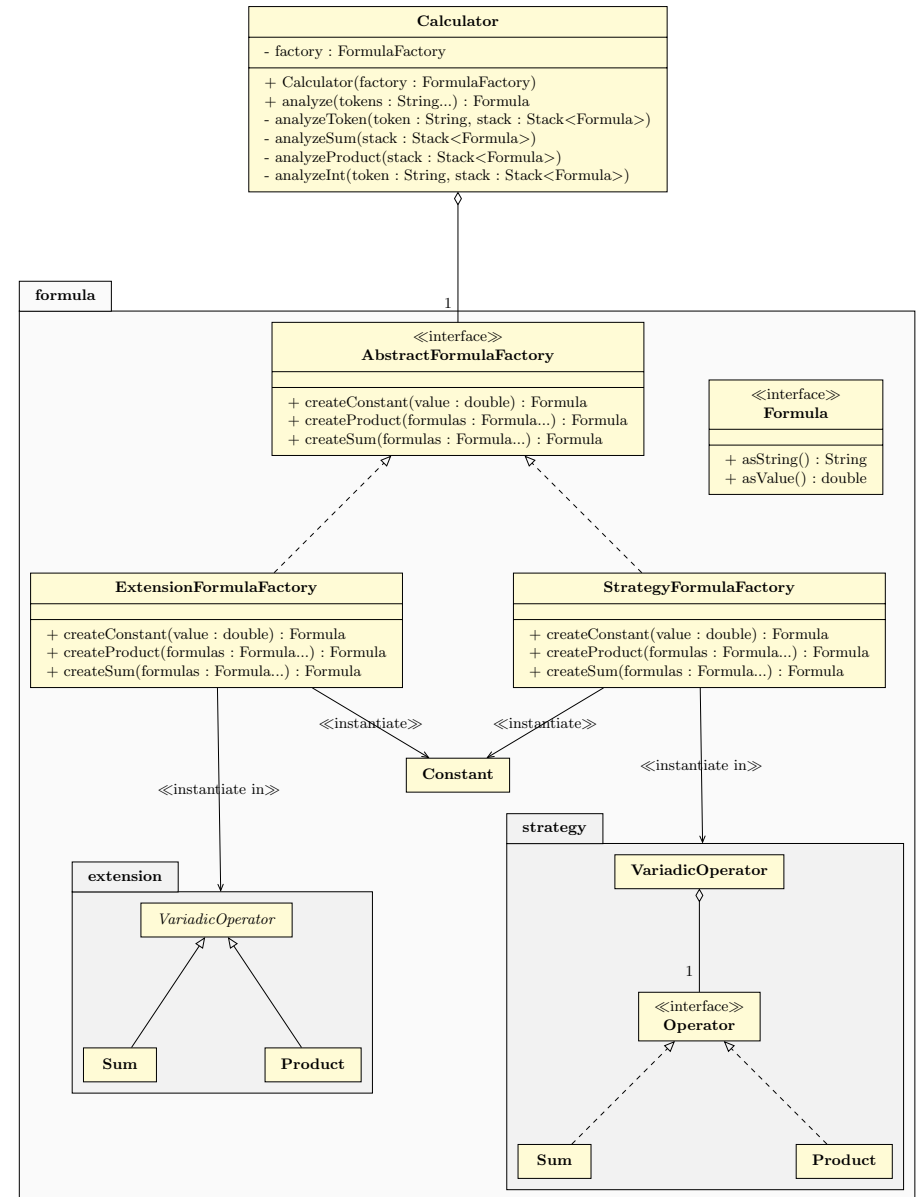


Diagramme 4 – Avec une fabrique abstraite