IUT Villetaneuse
M3105 : Conception et programmation objet avancées
Année 2019/2020, Formation Initiale

TD5 : Le design pattern Composite

Nous allons utiliser le design pattern composite pour représenter et calculer des expressions arithmétiques contenant des additions, soustractions, multiplications et/ou divisions.
Noter qu'une expression arithmétique peut se représenter sous la forme d'une arborescence, par exemple (3+4) / (5 * (10-6)) correspond à l'arborescence :

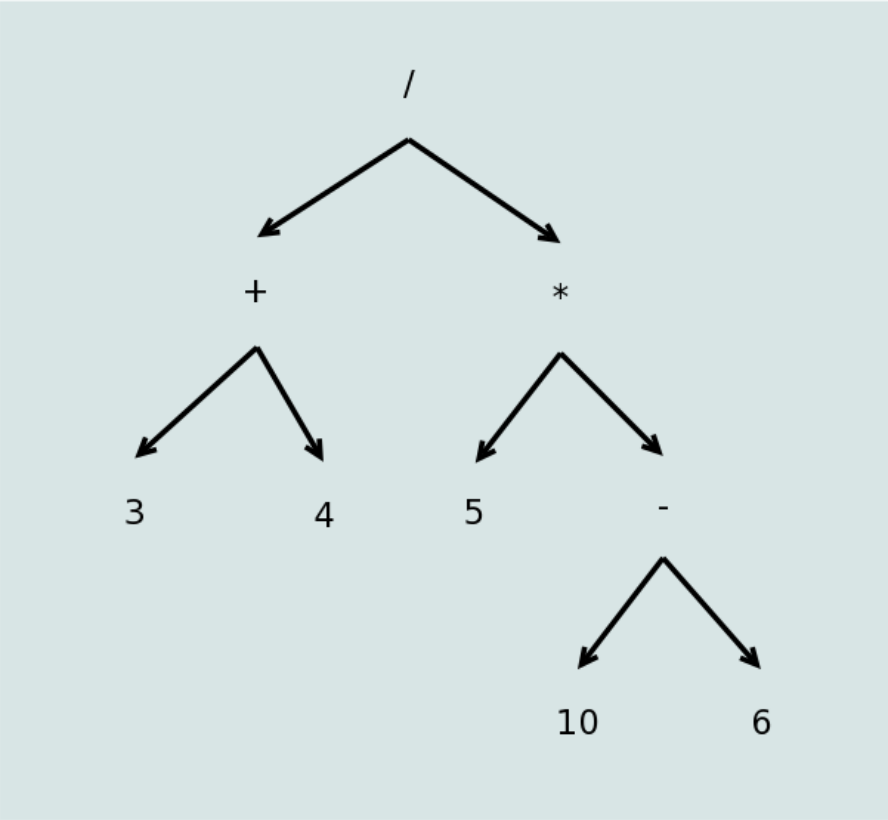

FIGURE 1 – Arborescence

I UML

1° Dans un premier temps nous n'utiliserons pas le design pattern composite.
Sur l'arborescence précédente on constate qu'une opération possède deux types d'opérandes : les nombres, représentés par les feuilles, et les opérations, représentées par les nœuds. Ainsi dans l'opération 5 * (10 – 6), la première opérande est le nombre 5, et la seconde est l'opération 10-6.

a) Compléter le diagramme des classes suivant de façon a représenter toutes les situations possibles concernant les 2 arguments des opérations :

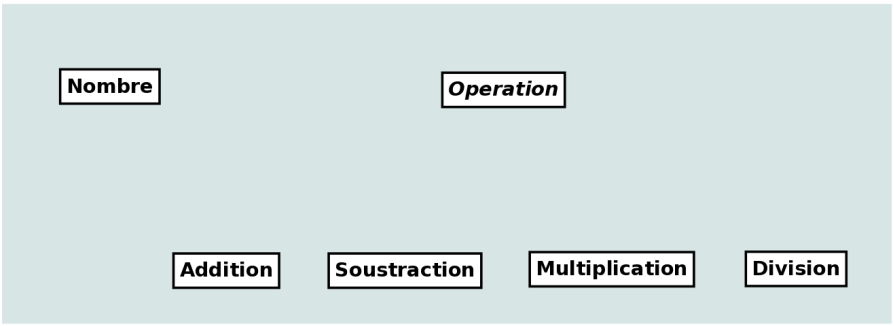

FIGURE 2 – Ebauche de diagramme

Noter qu'on doit utiliser un commentaire pour représenter le fait qu'une opération a exactement 2 opérandes.

b) Compléter le diagramme précédent de façon à munir les classes de 2 méthodes : valeur() qui retourne une valeur de type double représentant la valeur du Nombre ou de l'Operation, et une redéfinition de toString() retournant la représentation du Nombre ou de l'Operation.

2° Dans le cadre du design pattern composite on introduit une classe Expression dont les classes Nombre et Operation héritent.

a) Modifier le diagramme de la question précédente en utilisant le design pattern composite. Tenir compte du fait qu'on n'aura pas besoin d'ajouter ou retirer des opérandes puisque qu'une opération possède toujours 2 opérandes.

b) Modifier le diagramme précédent en tenant compte des informations suivantes :

- La valeur d'un nombre est représentée par un attribut valeurNombre (la méthode valeur retournera cette valeur).
- Ajouter un constructeur permettant d'initialiser un Nombre avec une valeur de type double.
- Ajouter à Expression 2 méthodes getOperande1() et getOperande2() qui retournent respectivement la première et la seconde opérande d'une Expression (par défaut on retournera null afin de prendre en compte le fait que l'Expression courante peut être nombre).
- Ajouter à Operation un constructeur permettant d'initialiser une opération avec 2 opérandes op1 et op2 qu'on affectera à deux variables operande1 et operande2.
- Redéfinir getOperande1() et getOperande2() dans Operation.

c) Donner les instructions de la méthode valeur() de la classe Addition qui retourne la valeur de l'addition de ses deux opérandes. Indication : utiliser la méthode valeur() de chaque opérande. Donner les instructions de la méthode toString() de cette même classe.

II Java

a) Traduire le diagramme des classes de la question c) en java et tester les programmes dans une classe Calculatrice en faisant des opérations simples, puis en calculant l'expression numérique donnée dans l'arborescence de la Figure1.