

## POO & LANGAGE JAVA

### Sujet 5 - Héritage & méthodes abstraites

**Expressions Numériques** Dans cet exercice, il s'agit de spécifier et d'implémenter les différentes classes permettant de construire, d'évaluer et d'afficher des expressions numériques représentées sous forme d'arbre. Par exemple, l'expression  $3 + 2 * 6 - 2 * (4 + 9)$  pourrait être représentée par l'arbre de la figure 1 (c'est une des solutions possibles) :

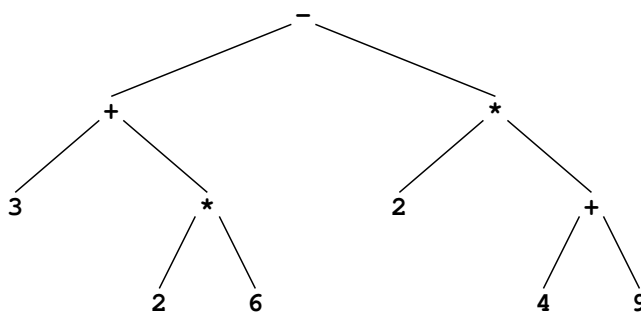


FIGURE 1 – Expression numérique

\* \* \*

Il faut pour cela commencer par définir une classe `Expression` de laquelle hériteront les classes suivantes :

- classe `Constante` : les instances de cette classe seront des constantes réelles
- classe `OperateurBinaire` : les instances de cette classe seront composées d'une expression gauche, d'un opérateur et d'une expression droite; bien évidemment (sic), l'opérateur ne sera pas explicitement codé dans un attribut quelconque, mais sera plutôt représenté implicitement par différentes sous-classes `Plus`, `Moins`, `Multiplier`, `Diviser`, etc...

Au final, le graphe de classes pourrait ressembler à celui de la figure 2. Ce ne sont que les grandes lignes. Libre à vous de le compléter comme bon vous semble en utilisant au mieux les différentes notions vues en cours et en TD (en vrac : méthodes et classes abstraites, méthodes et classes finales, méthodes génériques, constructeurs spécialisés, redéfinition de méthodes, surcharge de méthodes,...). Et n'oubliez pas de faire un programme de test **pertinent** !

Une fois ceci terminé et testé, vous vous attacherez ensuite à ajouter la classe `Variable`. Une variable est représentée par son nom, sa valeur pouvant varier selon le contexte. Vous devrez donc mettre en œuvre une classe `Contexte` permettant d'associer une valeur à une variable, de modifier la valeur associée à une variable, etc... Ensuite il faut faudra définir et implémenter (proprement et correctement) une nouvelle méthode permettant d'évaluer une expression numérique en fonction d'un contexte donné (celui-ci étant utilisé par les variables pour connaître leur valeur).

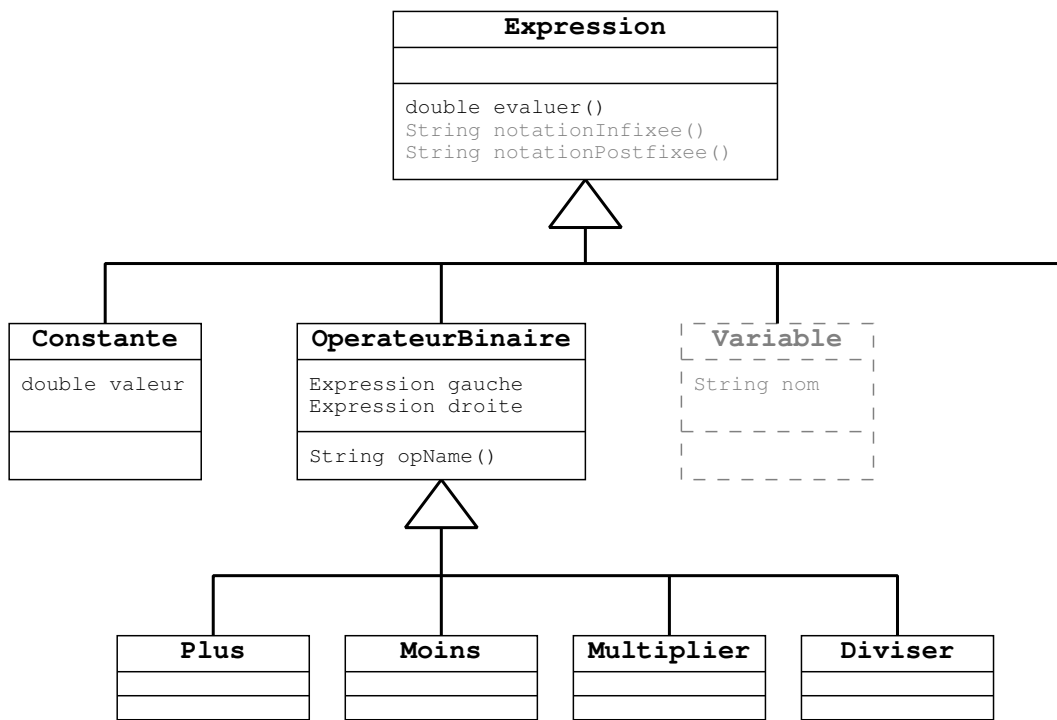


FIGURE 2 – Graphe de classes