

Expressions arithmétiques

1) Modéliser sous forme d'un diagramme de classes, pour un interpréteur syntaxique, la structure des **expressions arithmétiques** binaires du genre : $(X + (Y/2)) * ((X/3) - Y)$

Placer les opérations :

- `affecteValeur(V)` ; Affecte la valeur "V" à une variable (p.ex. X=5)
- `évaluer()` ; Calcule le résultat d'une expression et retourne sa valeur ; retourne null si toutes les variables n'ont pas été affectées
- `éditer_parenthésé` ; Édite la chaîne de caractère représentant l'expression en notation parenthésée . p.ex. " $(X + (Y/2)) * ((X/3) - Y)$ "
- `éditer_postfixé` ; Édite la chaîne de caractère représentant l'expression mais en notation postfixée. p.ex. la même expression s'écrirait : " $XY2/+X3/Y-*$ "
- `get_opérandes` ; Retourne les opérandes (p.ex. pour $(X/3)$, les 2 opérandes sont X et 3
- `get_opérateur` ; Retourne l'opérateur (p.ex. pour $(X/3)$, l'opérateur est la division : "/")

NB : Attention, tous les opérateurs ne sont pas commutatifs ! Notez aussi que les caractères "parenthèses" ne font pas partie de la structure puisque celles-ci ne sont pas nécessaires en notation postfixée.

Votre structure doit bien sûr pouvoir modéliser des expressions de n'importe quelle longueur (celle qui est indiquée plus haut n'est qu'un exemple) et quel que soit le nombre de variables

Pensez aux précisions suivantes :

- Typez les attributs, les paramètres, les valeurs de retour des opérations
- Pensez aussi aux éléments abstraits (classes et méthodes)
- Documentez en quelques mots les classes, attributs et opérations dans l'onglet "Documentation" (Essayez d'utiliser des termes mathématiquement exacts : variables, constantes, expression, terme, opérande...)

2) Dites ce qu'il faudrait changer et pourquoi, si on voulait aussi pouvoir représenter les opérateurs unaires du genre : $-(X+4) * Y$