

# Editeur graphique d'expressions arithmétiques

## Travaux pratique

On souhaite réaliser un éditeur graphique pour expressions arithmétiques. L'éditeur se présente comme une fenêtre où apparaît une palette offrant 5 options (cf. Fig. 1) :

- Une option permet de dessiner sous forme de cercle un opérateur (les opérateurs sont + - \* /)
- Une option permet de dessiner sous forme de carré une variable qui est définie par son nom et sa valeur
- Une option permet de dessiner sous forme de triangle une constante définie par sa valeur
- Une option permet de définir l'opérande gauche d'un opérateur
- Une option permet de définir l'opérande droit d'un opérateur

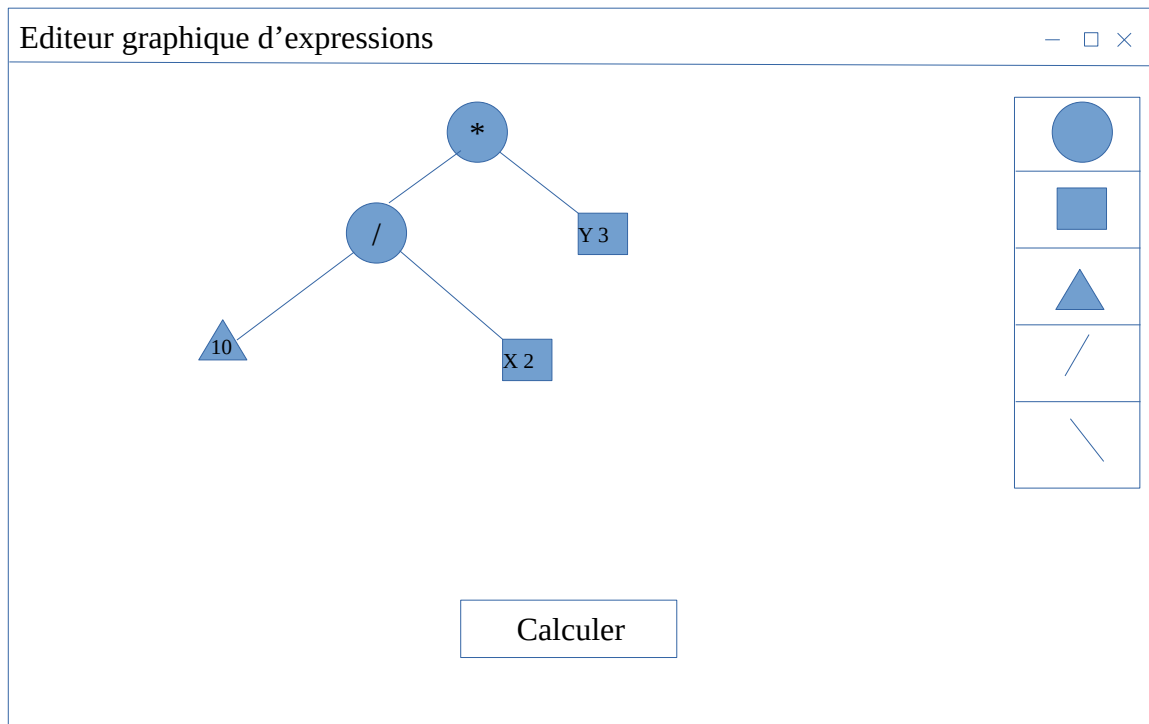


Fig. 1 : Interface de l'éditeur graphique

Ainsi la figure 1 montre l'éditeur où l'expression  $(10/x)*y$  a été dessinée avec x valant deux et Y trois. Le click sur le bouton Calculer doit fournir le résultat 15.

On exige que l'éditeur soit conforme à l'architecture MVC. Pour cela, on introduit 3 modules :

- Le module Vue qui prend en charge la présentation. Tous les éléments de l'interface utilisateur seront pris en charge par ce module. On impose d'utiliser le package tkinter pour implanter l'interface graphique. Enormément de ressources sont disponibles pour tkinter, en particulier, Il est possible de consulter <https://tkdocs.com>
- Le module Modèle stocke les données de l'application. La vue manipule des éléments graphique, elle ne se préoccupe pas de la sémantique d'une expression. Alors que le modèle

manipule les expressions, le modèle a en charge l'évaluation de l'expression.. Pour cela, il implante le diagramme de classe suivant :

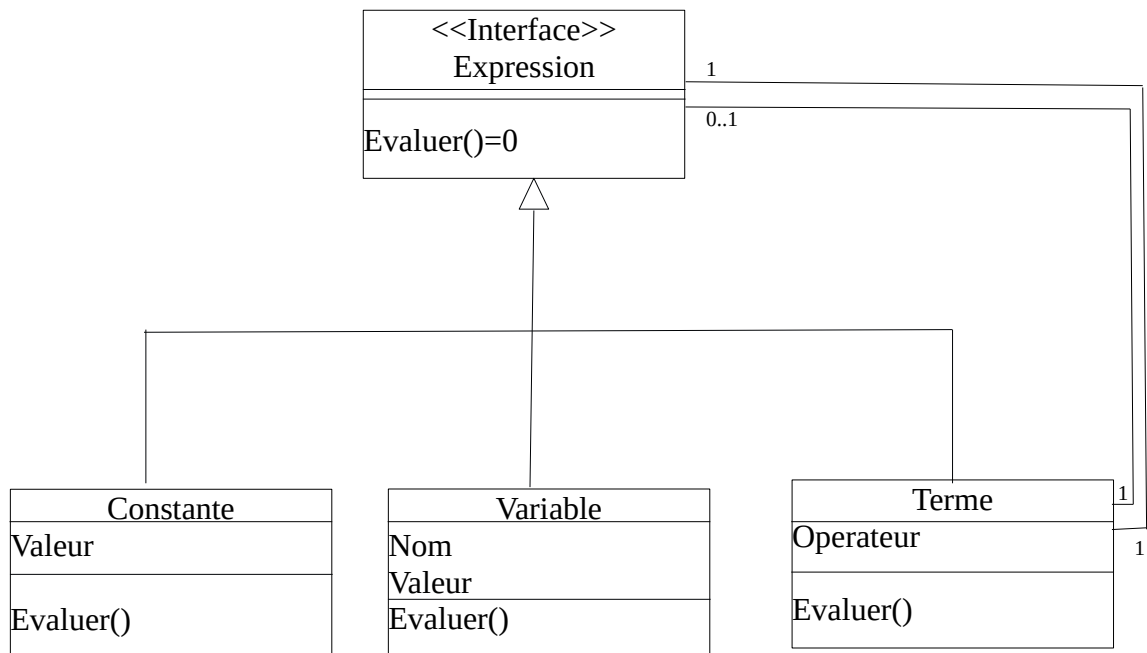


Fig2. Diagramme des classes du modèle

- Le module controleur : L'architecture MVC impose que le modèle et la vue soit totalement indépendant l'un de l'autre. La vue n'a aucune connaissance sur le modèle et vice-versa. Le controleur a pour rôle d'acheminer l'information depuis le modèle vers la vue et en sens inverse. Le controleur dépend de la vue et il dépend du modèle, l'inverse n'est pas vrai: la vue et le modèle sont indépendants du controleur. La vue et le modèle sont chacun réutilisable indépendamment de tous les autres éléments de l'application.

Il est demandé de remettre 3 fichiers source en python : `vue.py`, `modele.py` et `controleur.py` correspondants aux 3 modules et un rapport qui explique le contenu des fichiers.