La conception de ce projet nécessite l’utilisation de la programmation orientée objet.

La programmation orientée objet apporte des avantages face à la programmation procédurale.

Tout d’abord elle permet d’obtenir un code évolutif. Les types d’objets créés peuvent servir de base pour d’autres objets.

Les objets ont chacun leur propriétés et méthodes. Le code est plus compréhensible. Il est donc aisé de voir ce qu’une fonction manipule et à quoi correspondent les variables disponibles.

A des fins de confidentialités, les attributs des classes sont privés. Leurs valeur peuvent être accéder et modifier via des accesseurs ( getter & setter ).

Voyons donc la structure de nos classes.

La classe Broker :

Elle contient les attributs suivants : \_\_broker\_name, \_\_broker\_fees.

\_\_broker\_name permet de définir d’identifier le broker

\_\_broker\_fees permet de fixer les frais du broker.

Ces frais sont sous forme suivante :

list(borne inférieur 1, borne supérieur 1, taux fixe, taux variable, borne inférieur 2, borne supérieur 2, taux fixe, taux variable […] )

Elle contient les méthodes suivantes :

comp\_broker\_fees(asset\_quantity, asset\_price)

Cette fonction permet de calculer les frais correspondant à une position (quantité d’actif \* prix de l’actif). Cette fonction est vitale pour l’ensemble du projet, elle est réutilisée dans différentes classes avec différents