

Langages à Objets Avancés

M1 - INFO | 2016 - 2017

TP 6 - Héritage

Université Paris-Diderot

1 Automatic Trading

1.1 Actifs et options

Un actif financier est un contrat qui possède un intitulé (nom) et un prix d'achat. Il est aussi intéressant de connaître à tout moment sa valeur sur le marché. Nous supposons que cette valeur est aléatoire à cause des fluctuations des marchés. Écrire une classe `Asset` qui modélise un actif financier, qui possède une méthode `CurrentValue()` qui renvoie un nombre aléatoire modulo 30, ainsi que tous les membres nécessaires avec leurs accesseurs.

Une option est aussi un actif financier. Il est basé sur un autre actif `A` et confère au détenteur de l'option le droit d'achat de `A` dans le futur à un prix convenu `AgreedPrice`. Écrire la classe `Option` modélisant cet actif et contenant l'actif `A` sur lequel est indexée l'option.

Il est naturel que la valeur de l'option dépende de la valeur de l'actif sur lequel elle est indexée. Modifiez votre classe pour que cela soit le cas. Le choix de la dépendance entre la valeur des deux est libre.

1.2 Trader

On souhaite avoir un `Trader` automatique qui possède une vision du marché représentée par un tableau contenant des actifs financiers (des actifs simples mais aussi des options), un horizon (un entier représentant le nombre de jours de trading ou de présence sur les marchés) durant lequel il possèdera un portefeuille de transaction contenant les actifs qu'il aura acheté sur ce marché ainsi qu'un capital représentant la valeur qu'il est autorisé à dépenser.

Chaque jour le `Trader` examine la valeur des actifs sur le marché et si un actif a une valeur plus élevée que son prix d'achat il l'achète en le mettant dans le portefeuille. Cette nouvelle valeur devient alors le nouveau prix d'achat pour cet actif et l'ancien prix est déduit du capital du `Trader`. Écrivez la classe `Trader` ainsi que ses méthodes `Buy`, `Sell` et `trade`

2 Contrôle d'accès (Partie non obligatoire)

Soient les classes suivantes :

```
1 class X{
2     int priv;
3 public:
4     int pub;
```

```
5     void m();
6 };
7
8 class Y : public X{
9     void my();
10};
```

Est-ce que le code suivant est correct ?

```
1 // Premier -----
2 void X::m(){
3     priv = 1;
4     pub = 3;
5 }
6
7
8 // Deuxieme ----
9 void Y::my() {
10     priv=1;
11     pub=3;
12 }
13
14 // Troisieme -----
15
16 void f(Y * p){
17     p->priv=1;
18     p->pub=3;
19 }
```