

TP Programmation C++ pour la finance

Partie 1 : Actif sans risque

Ecrire la classe *ASR* (actif sans risque) permettant de calculer le gain d'un placement d'une somme S à un taux t durant une durée d . On considère que la périodicité avec laquelle les intérêts sont calculés est une donnée partagée par toutes les instances dont la valeur par défaut est 12. Cette valeur peut être modifiée mais reste un multiple de 12, dans le cas contraire la valeur par défaut n'est pas modifiée.

Par défaut ce type de produit comprend un taux d'intérêt de 2% pour une durée de 12 unités de temps.

Il existe deux types d'actifs sans risques, ceux dont les gains sont calculés sur la base d'intérêts simples et ceux dont les intérêts sont calculés sur la base d'intérêts composés.

Proposez une structure permettant le calcul générique d'actifs des deux types, et implémenter. Les calculs de gains sont réalisés sur un nombre entier de périodicité. La périodicité par défaut est celle de la classe mais peut aussi être un paramètre du traitement qui dans ce cas n'est pas nécessairement un multiple de 12.

Partie 2 : Action

2.1 : Ecrire une classe *Dividende* qui enregistre dans un vecteur les différents montants pour un nombre d'unité de temps. Cette classe devra permettre :

- de créer un objet en spécifiant le nombre d'unité de temps maximum,
- d'ajouter un nouveau dividende si le maximum n'est pas atteint,
- de remplacer un dividende à une unité de temps donnée en paramètre (si l'unité de temps est correcte) par une nouvelle valeur donnée en paramètre,
- de calculer la somme des dividendes entre 2 unités de temps (par défaut entre 0 et le nombre de dividendes enregistrés si la valeur de fin est supérieure au nombre de montants),
- de retrouver le montant maximum,
- d'afficher le contenu des dividendes dont la valeur est connue selon le format suivant :
montant : <le montant> pour le moment : <unité de temps>.

Vous donnerez un programme principal dans lequel vous devrez créer un objet *Dividende* de 3 unités de temps. Vous lui ajouterez 4 montants et ferez afficher le résultat.

2.2 : Ecrire la classe *Action* permettant de calculer le gain associé à l'action en prenant en compte, le nombre d'actions possédés et les dividendes encaissés. Pour chaque action on enregistre un vecteur de cotations. Chacune enregistre les cotations pour une unité de temps.

La classe *Action* a les spécifications suivantes :

- Un constructeur qui crée une instance de *Action* pour un nombre d'unités de temps, un code enregistrée comme un entier et le nombre d'actions possédées. Un objet *Dividende* est alors créé du même nombre d'unités de temps ainsi qu'un vecteur de *Cotation*.
- Une méthode permet d'ajouter un montant de dividende et une *Cotation correspondant* à la même unité de temps.

- Une méthode permet de retourner la première *Cotation* enregistrée (première valeur du premier objet *Cotation*) et une autre la dernière (dernière valeur du dernier objet *Cotation* enregistrée).
- Une méthode permet d'afficher la variabilité de l'action, c'est-à-dire la différence entre la cotation la plus élevée et la plus basse sur l'ensemble des cotations enregistrées.
- Une méthode d'affichage qui affiche les informations sur l'action selon le format suivant :
 Action : <code action> Volume : <nombre actions possédées>
 Dividende :
 montant : <le montant> pour le moment : <unité de temps>. (autant de dividende connu)
 Première Cotation : <première cotation>
 Dernière Cotation : <dernière cotation>
- Une méthode donnant le gain associé à cette action comme le résultat de la somme des dividendes plus le nombre d'actions fois la différence de cotation.
- Une méthode donnant le rendement associé à cette action (gains en fonction du coût initial).

Partie 3 : Portefeuille

On souhaite que ces classes (actifs sans risques et actions) puisse être enregistrées dans un *Portefeuille* afin de prendre en compte des objets de tous types dans le calcul du gain. Modifier les classes en conséquence.

Proposer un exemple de calcul.