

## TD 3 PROGRAMMATION ORIENTÉE OBJET EN C++

### Objectif de la séance

L'objectif de cette séance est d'étudier les relations de composition et d'agrégation entre les classes. On s'appuiera pour cela sur l'exemple de l'application de gestion des notes vue en cours et dans les exercices. Vous pourrez donc vous référer aux corrections correspondantes.

Au fur et à mesure de la séance, vous devez faire les manipulations proposées sur votre PC (sous Visual Studio) et faire les diagrammes de classes dans les cadres prévus à cet effet. A la fin de la séance, **vous devrez déposer l'ensemble des fichiers sources créés** sur le portail (les diagrammes ne seront pas déposés sur le portail).

On rappelle qu'il s'agit d'écrire une application capable de saisir les notes d'un étudiant pour un ensemble de matières que suit cet étudiant, puis de calculer la moyenne générale de l'étudiant. Les différents cas d'utilisation identifiés pour l'application sont :

1. définition des matières que suit l'étudiant,
2. saisie des notes pour l'ensemble des matières,
3. calcul de la moyenne générale.

On veut concevoir deux classes permettant de répondre aux 3 cas d'utilisation :

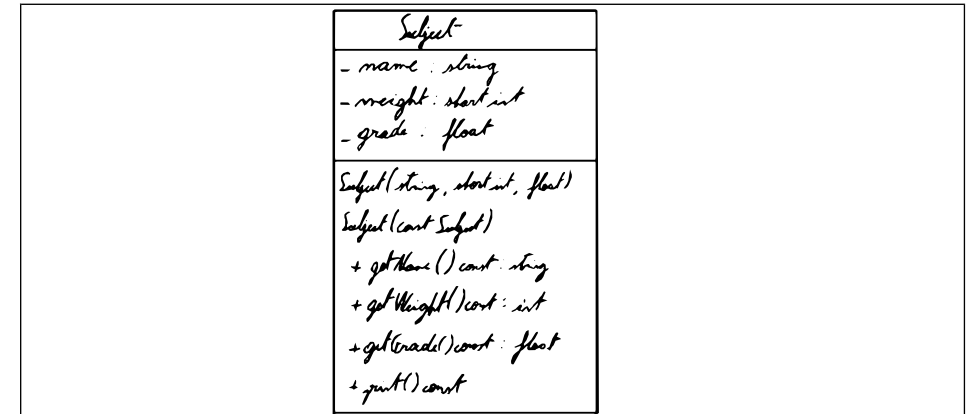
- la classe *Matiere* qui est conçue pour gérer les caractéristiques d'une matière : nom de la matière, coefficient, note obtenue pour cette matière ;
- la classe *Semestre* qui permet de regrouper les matières que suit un étudiant, puis d'en calculer la moyenne.

Les différents fichiers de l'application seront insérés dans un projet de type « Application console Win32 » intitulé *Calcul\_moyenne* qui devra être créé au début de la séance. Pour chaque classe, vous devrez créer un fichier de déclaration (.h) et le fichier de définition correspondant (.cpp). Ces fichiers porteront le même nom que la classe. Les programmes de test de chaque classe seront conservés dans des fichiers séparés (le fichier test de la classe *truc* sera intitulé *test\_truc.cpp*). Vous devrez conserver tous les fichiers sources qui seront réutilisés dans d'autres séances de TD.

### Partie 1 : la classe Matière

On s'intéresse ici à la conception de la classe *Matiere* qui est conçue pour gérer les caractéristiques d'une matière. Cette classe sera en composition avec la classe *string* pour gérer le nom de la matière.

a) Dans le cadre ci-dessous, faire le diagramme de cette classe en prévoyant la composition ainsi que les champs et les méthodes nécessaires aux trois scénarios d'utilisation. Prévoir aussi un constructeur avec les paramètres permettant d'initialiser tous les champs.

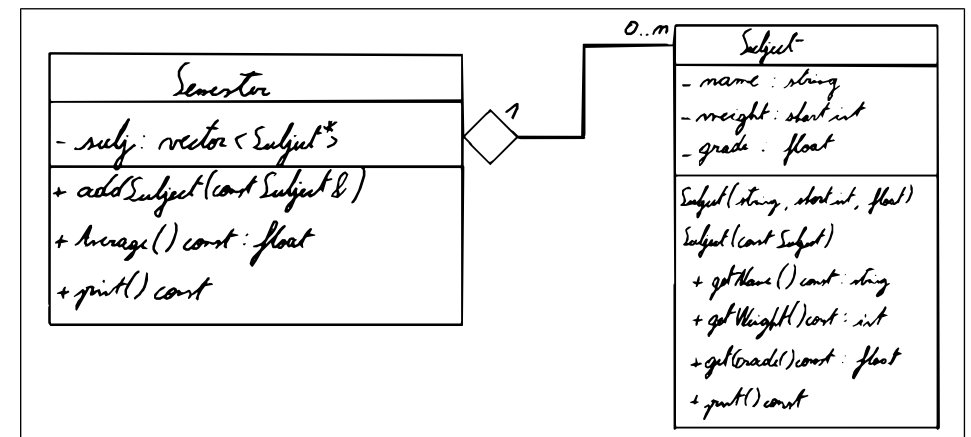


- Ecrire un programme de test de cette classe.
- Déclarer et définir la classe et toutes ses méthodes.
- Compiler et tester.

### Partie 2 : la classe Semestre

On s'intéresse maintenant à la classe *Semestre* qui permet de regrouper les matières que suit un étudiant, puis d'en calculer la moyenne. Cette classe sera conçue comme un **agrégat multiple** d'objets de classe *Matiere*

a) Faire ci-dessous le diagramme de classe de la classe *Semestre* en faisant apparaître la relation d'agrégation. Ne pas faire figurer les champs et les méthodes de la classe *Matiere*.



- Ecrire un programme de test de la classe *Semestre* qui correspond au scénario 1 de définition des matières.
- Déclarer et définir la classe et les méthodes intervenant dans le scénario précédent.
- Compiler et tester.

### Partie 3 : Le calcul de la moyenne

a) Modifier le programme de test pour que les matières ne soient plus saisies par l'utilisateur, mais qu'elles soient définies directement dans le programme avec des valeurs fixées pour le nom et le coefficient de chaque matière.

b) Ajouter dans le programme de test les instructions pour répondre aux scénarios 2 et 3.

c) Définir et tester les méthodes permettant de répondre aux scénarios 2 et 3.

### Partie 4 : L'application de calcul de la moyenne

On s'intéresse maintenant à l'application permettant de répondre aux 3 scénarios définis dans l'énoncé. Le programme principal correspondant à cette application sera écrit dans un fichier *main.cpp*.

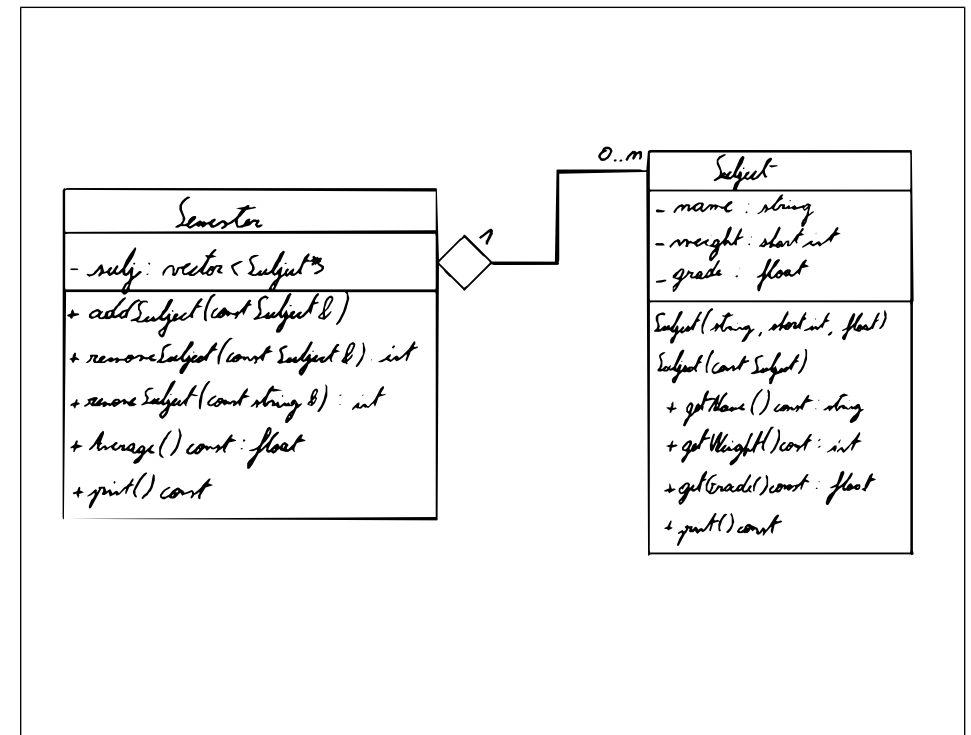
a) Ecrire une fonction *menu* chargée d'afficher un menu présentant les options correspondant aux 3 scénarios de l'énoncé plus une option quitter permettant de sortir de l'application. Cette fonction devra demander à l'utilisateur de faire un choix et elle devra retourner un entier correspondant au choix qu'il a fait.

b) Ecrire le programme principal de l'application et tester le résultat (utiliser une instruction *switch* pour traiter le choix de l'utilisateur).

### Partie 5 : Supprimer des matières (facultatif)

Pour faciliter la saisie et la modification des matières, on veut modifier le scénario 1 en permettant à l'utilisateur d'ajouter ou de supprimer des matières dans le semestre. Ainsi, lorsqu'il choisira l'option correspondante dans le menu, il pourra voir la liste des matières déjà saisies et il lui sera proposé d'en ajouter ou d'en supprimer une (l'ajout se fera systématiquement à la fin alors que la suppression se fera n'importe où).

a) Donner le nouveau diagramme de la classes *Semestre* pour pouvoir répondre au nouveau scénario.



b) Modifier la déclaration et la définition de la classe *Semestre*.

c) Modifier le programme principal pour répondre au nouveau scénario.