

Objectif

Le but de ce TP est d'assimiler les notions suivantes :

- La notion de classe
- La stratégie D.D.U
- La portée des membres de classe

Exercice 1

On veut manipuler des points définis par une abscisse et une ordonnée (des réels). Pour un point, on peut calculer la distance entre 2 points et le milieu de 2 points.

Vous allez donc définir une classe *Point* définie par un fichier *.h* et un fichier *.cpp* ainsi que le programme principal *main.cpp*.

La classe point comporte alors des membres publiques qui sont :

- Deux données réelles : abscisse x et ordonnée y.
- Deux méthodes :
 - *double distance(Point &P);*
 - *Point milieu(Point &P);*

Le fichier *point.h* utilise des directives de compilation *#ifndef*, *#define* et *#endif* pour gérer les inclusions multiple du fichier header.

Rappel :

La distance entre le point A et le point B se traduit comme suit :

$$AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

Exercice 2 encapsulation

Veuillez interdire l'accès aux données membres x et y en utilisant le mot clé *private*.

Pour cela, rajouter les Accesseurs et mutateurs et apporter les modifications nécessaires à votre programme.

Exercice 3 Gestion des données des étudiants

On veut développer une application qui permet de saisir les données des étudiants et de calculer leurs moyennes. Vous allez commencer tout d'abord par la création d' :

- Une classe intitulée *Matiere* avec :
 - Attributs : *intitule* de type string, *coefficient* de type double, *note* de type double
 - Méthodes :
 - Les accesseurs (getters & setters)
 - Un constructeur

- Une classe intitulée Etudiant avec :
 - Attributs : *Num_carte* de type int, *téléphone* de type int, *Moyenne* de type double, *nbr_matières* de type int et *matieres* est un tableau de type *Matiere*
 - Méthodes :
 - Les accesseurs (getters & setters)
 - Méthode void *afficher()* : permet d'afficher les données d'un étudiant
 - Méthode double *calcul_moyenne()* : permet de calculer la moyenne de l'étudiant
 - Méthode boolean *reussi()* : permet de déduire si l'étudiant en question a réussi ou pas s'il a une moyenne supérieure à 10
 - Un constructeur
- Une classe intitulée Filiere avec :
 - Attributs : *id_filiere* de type int, *nbr_etudiants* de type int et *etudiants* de type tableau d'Etudiant
 - Méthodes :
 - Les accesseurs (getters & setters)
 - Méthode void *afficher()* : permet d'afficher les données de tous les étudiants de la filières
 - Un constructeur

Les attributs seront privés et les méthodes publiques.

Questions

1. Ecrivez le code des classes demandées.
2. Ecrire un programme principal en créant 4 étudiants, 3 matières et 2 filières
 - a. Afficher l'ensemble des étudiants et leurs matières des deux filières
 - b. Calculer la moyenne des étudiants
 - c. Afficher la liste des étudiants qui ont réussi