

TP de C++ n°3

Exercice 1 : Classe, méthodes, constructeurs, accesseurs et mutateurs

Ecrire une classe `Rectangle` comportant deux attributs de type réel double précision : `largeur` et `hauteur`.

Cette classe doit également disposer :

- d'un constructeur qui fait également office de constructeur par défaut (`largeur` et `hauteur` prendront alors la valeur 0.0 par défaut) ;
- d'une méthode `perimetre()` retournant le périmètre du rectangle ;
- d'une méthode `surface()` retournant la surface du rectangle ;
- d'une méthode `affiche()` qui affiche à l'écran « `Rectangle L:l H:h` » où `l` et `h` sont les valeurs des attributs `largeur` et `hauteur` respectivement.

Attention, les méthodes demandées doivent pouvoir être appliquées à une instance non modifiable de rectangle (`const Rectangle`).

On ne souhaite pas que l'utilisateur de cette classe (fonction externe ou autre classe) puisse accéder directement aux attributs. Proposez les mécanismes nécessaires pour assurer cette protection des données mais permettre quand même de consulter et de modifier ces valeurs (via des accesseurs et mutateurs).

Écrire un programme principal dans un autre fichier qui demande à l'utilisateur de saisir la taille d'un rectangle, crée une instance de la classe `Rectangle` de la taille saisie et affiche le périmètre et la surface de ce rectangle.

Exercice 2 : Classe, tableau, allocation dynamique

Créer une classe `Etudiant` permettant de représenter un étudiant par son nom et son prénom (type `string`), son identifiant (type `unsigned int`) et ses notes (tableau dynamique de `float`).

La classe `Etudiant` doit disposer d'un constructeur permettant d'initialiser son nom, son prénom et son identifiant ainsi que le nombre de notes prévues. Le nombre de notes prévues aura une valeur par défaut de 5. Ce constructeur aura entre autre pour rôle d'allouer dynamiquement l'espace nécessaire pour le tableau de notes.

Ajouter les accesseurs et mutateurs pour le nom, le prénom et l'identifiant.

La classe `Etudiant` doit disposer d'une méthode `ajouterNote(float note)` permettant d'ajouter une nouvelle note (attention à la taille maximum du tableau). De quel(s) autre(s) attribut(s) a-t-on besoin d'ajouter à la classe `Etudiant` pour pouvoir gérer correctement l'ajout d'une note ? Programmer cette méthode.

Ajouter un constructeur de copie qui permet de créer une nouvelle instance de la classe `Etudiant` à partir d'une instance existante en recopiant son état (attention au traitement du tableau de notes).

Ajouter le destructeur de la classe `Etudiant`. Que doit faire ce destructeur avant disparition de l'instance ?

Écrire un programme principal qui utilise toutes les fonctionnalités de la classe `Etudiant`.