

Feuille de TD N° 4

Classes en C++

Exercice 1 : Classe Voiture

Soit la classe voiture avec les méthodes tourne, accélère, allumeBougie, sortPiston, coinceRoue, changeVitesse, injecteEssenceDansCylindre.

Quelles sont celles qui font partie de l'interface et celles qui font partie de l'implémentation ?

Exercice 2 : Rectangle

Ecrire un programme utilisant une classe rectangle dont le constructeur prend deux paramètres, largeur et hauteur et qui offre les fonctions suivantes :

- calcul du périmètre
- calcul de la surface
- affichage

Ainsi que les accesseurs et mutateurs triviaux (lecture et modification de la largeur et de la hauteur).

Exercice 3 : Pile d'entiers

Une pile est un ensemble dynamique d'éléments où le retrait se fait d'une façon particulière. En effet, lorsque l'on désire enlever un élément de l'ensemble, ce sera toujours le dernier inséré qui sera retiré. Un objet pile doit répondre aux fonctions suivantes:

-Initialiser une pile

-Empiler un élément sur la pile (push)

-Dépiler un élément de la pile (pop)

pour cela nous allons supposer que les éléments à empiler sont de type int.

Le programme main comprend la définition d'une classe pile et un programme de test qui crée deux piles p1 et p2, empile dessus des valeurs entières et les dépile pour vérifier les opérations push et pop.

Exercice 4 : Une classe point

Réaliser **une classe point** permettant de manipuler un point d'un plan. on prévoira:

-un point est défini par ses coordonnées x et y (des membres privés)

-un constructeur (vous pouvez également implémenter les trois types de constructeur)

-une fonction membre déplace effectuant une translation définie par ses deux arguments dx et dy (double)

-une fonction membre affiche se contentant d'afficher les coordonnées cartésiennes du point.

-une fonction membre saisir se contentant de saisir les coordonnées cartésiennes du point.

-une fonction membre distance effectuant calculant la distance entre deux point.

-une fonction membre milieu donnant le milieu d'un segment.

on écrira séparément:

-un fichier source constituant la déclaration et la définition de la classe.

-un petit programme d'essai (main) gérant la classe point.

Exercice 5 : la classe Compteur

On souhaite implémenter une classe représentant un compteur entier. Un tel objet se caractérise par :

Une valeur entière, positive ou nulle, nulle à l'origine.

Le fait qu'il ne peut varier que par pas de 1 (incrémentation ou décrémentation). On convient qu'une décrémentation d'un compteur nul est

sans effet.

Il s'agit de créer une classe Compteur pour rendre le service demandé. On écrira en outre un petit programme de test qui :

1. créera un compteur et affichera sa valeur;
2. l'incrémentera 10 fois, puis affichera à nouveau sa valeur;
3. le décrémentera 20 fois, puis affichera une troisième fois sa valeur

La sortie de ce programme doit donner (quelque chose comme) "0 10 0"

Exercice 6 : La classe étudiant

On voudrait gérer les étudiants d'une institution à l'aide d'une classe Etudiant définie par :

les attributs suivants :

- nom : nom d'un étudiant
- prénom: prénom d'un étudiant
- tabnotes : tableau contenant les notes d'un étudiant, sachant qu'un étudiant a au total 10 notes.

les méthodes suivantes :

- void saisie (), permettant la saisie d'un étudiant
- void affichage (), permettant l'affichage d'un étudiant
- float moyenne (), retourne comme résultat la moyenne des notes d'un étudiant.
- int admis (), retourne comme résultat la valeur 1, si un étudiant est admis et la valeur 0, sinon. Un étudiant est considéré comme étant

admis lorsque la moyenne de ses notes est supérieure ou égale à 10.

- int exae_quo (Etudiant E), retourne comme résultat la valeur 1, si deux étudiants ont la même moyenne et la valeur 0, sinon.

Ecrire la classe Etudiant dans le langage C++.

Exercice 7 : Classe Cercle

Créer 3 fichiers pour l'énoncé suivant : (h, cpp et principal).

Créer une classe cercle contenant une donnée membre privée r (le rayon) et 3 méthodes publiques :

- un constructeur qui initialise les données membres et qui donne comme valeur par défaut à r 1
- surface qui renvoie la surface d'un cercle et qui a comme paramètre le rayon (entier)
- glsurface qui a comme paramètre le côté d'un carré et qui renvoie la surface d'un carré, elle utilise une fonction globale (externe à la classe), nommée surface, qui renvoie la surface d'un carré et qui a comme paramètre un entier (le côté du carré).

Créer un programme appelant les différentes méthodes et affichant les différentes surfaces.

Exercice 8 : File d'attente

Créer une classe `file_attente` qui permet de gérer des objets fonctionnant sur le mode "premier entré-dernier sorti". Cette classe comprendra un constructeur permettant d'initialiser les données membres (un entier et un pointeur sur des entiers), une méthode qui permet d'ajouter un élément à la pile et une méthode qui permet de retirer un élément à la pile.

Créer un programme permettant de tester la classe.

Exercice 9 : Point à 3 dimensions

Créer une classe `point` pour représenter des points dans un espace à 3 dimensions (x, y, z). Inclure un constructeur par défaut, un constructeur initialisant les données membres, un constructeur par copie, une fonction `oppose` qui transforme un point en son opposé, une fonction `norme` qui retourne la distance du point par rapport à l'origine (0, 0, 0) et une fonction `affiche`. Créer un programme permettant de tester les différents constructeurs et fonctions.

Modifier les constructeurs pour qu'ils acceptent des lignes d'initialisation

Remplacer la fonction `affiche` en utilisant 3 fonctions d'accès c'est-à-dire des fonctions qui retournent les valeurs de x, de y et de z.

Modifier le constructeur pour qu'il initialise, par défaut, le point avec les coordonnées (1,2,3).