



Atelier 3

Objectif : Les Bases de la programmation POO en C++.

Programme 1 :

Écrivez un programme qui définit une classe appelée MyClass avec un constructeur par défaut et un destructeur définis par l'utilisateur.

Définissez à la fois le constructeur et le destructeur en dehors de la classe.

Les deux fonctions membres généreront un libre choix du texte sur la sortie standard.

Créez un objet d'une classe dans la fonction main.

Programme 2 :

Écrire un programme en C++ qui définit une classe shape avec un constructeur qui donne une valeur à la largeur et à la hauteur. Ils définissent deux sous-classes triangle et rectangle, qui calculent l'aire de la zone de forme(). Dans l'ensemble, définir deux variables un triangle et un rectangle, puis appelez la fonction area() dans ces deux variables.

Programme 3 :

Effectuer les opérations arithmétiques sur des données complexes à l'aide d'une classe et d'un objet. Le programme doit demander la partie réelle et imaginaire de deux nombres complexes et afficher les parties réelle et imaginaire de l'opération demandée.

Programme 4 :

Ecrire un programme en C++ avec une classe mère et une classe fille héritée. Les deux doivent avoir une méthode void display() qui affiche un message (différent pour la mère et la fille). En gros, définir la fille et appelez la méthode display (), utiliser cette classe fille dans la méthode main.

Programme 5 :

Ecrire un programme en C++ avec une classe mère Animal. À l'intérieur, définir des variables nom et d'âge, et la fonction set_value(). Créer ensuite deux sous classes de base Zebra et Dolphin qui écrivent un message indiquant l'âge, le nom et donnant des



Université Abdelmalek Essaadi
Faculté des Sciences et techniques de Tanger
Département Génie Informatique
LST GI S5
Algorithmique et POO C++



informations supplémentaires (par exemple, le lieu d'origine), Créer 2 variables un de type Zebra et l'autre Dolphin puis appeler la méthode `et_value()` pour chaque instance.

Programme 6 :

une classe `Personne` qui comporte trois champs privés, `nom`, `prénom` et `date de naissance`.

Cette classe comporte un constructeur pour permettre d'initialiser les données. Elle comporte également une méthode polymorphe `Afficher` pour afficher les données de chaque personne.

une classe `Employé` qui dérive de la classe `Personne`, avec en plus un champ `Salaire` accompagné de sa propriété, un constructeur et la redéfinition de la méthode `Afficher`.

une classe `Chef` qui dérive de la classe `Employé`, avec en plus un champ `Service` accompagné de sa propriété, un constructeur et la redéfinition de la méthode `Afficher`.

une classe `Directeur` qui dérive de la classe `Chef`, avec en plus un champ `Société` accompagné de sa propriété, un constructeur et la redéfinition de la méthode `Afficher`.

Programme 7 :

Réaliser une classe C++ "`vecteur3d`" permettant de manipuler des vecteurs à 3 composantes (de type `float`). On y prévoira :

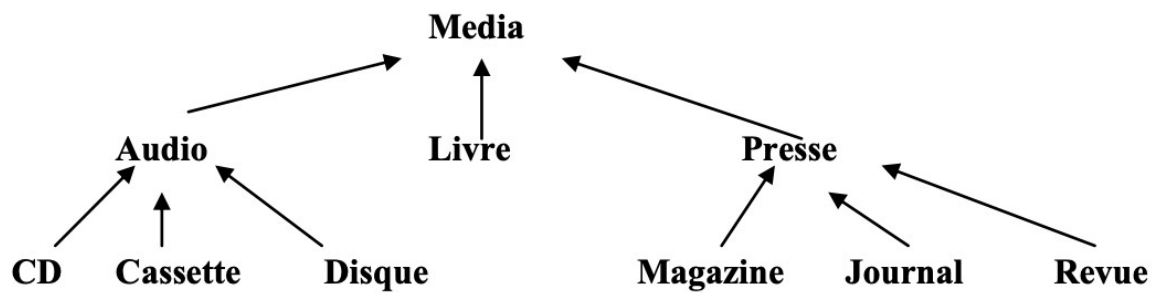
- un constructeur, avec des valeurs par défaut (0),
- une fonction d'affichage des 3 composantes du vecteur, sous la forme : (x, y, z)
- une fonction permettant d'obtenir la somme de 2 vecteurs ;
- une fonction permettant d'obtenir le produit scalaire de 2 vecteurs.
- une fonction `coincide` permettant de savoir si 2 vecteurs ont mêmes composantes.
- une fonction qui renvoie la norme du vecteur

•une fonction nommée `normax` permettant d'obtenir, parmi deux vecteurs, celui qui a la plus grande norme. On prévoira trois situations :

1. le résultat est renvoyé par valeur ;
2. le résultat est renvoyé par adresse, l'argument étant également transmis par adresse.
3. le résultat est renvoyé par référence, l'argument étant également transmis par référence.

Programme 8 :

En utilisant la notion de virtual donner une possible déclaration de l'hérarchie de classes suivante :



La classe Media possède deux méthodes « `void imprimer()` ; `char *id()` », et un attribut titre, par contre tous les autres classes possèdent aussi ces propres attributs et méthodes qu'il faut ajouter selon la spécificité de chaque sous classe.

Programme 9 :

Ecrire un programme en C++ qui vérifie combien de fois une fonction « `call` » d'une classe Test a été appelée à partir du programme principal, main.

Note : penser à utiliser une variable static.



Université Abdelmalek Essaadi
Faculté des Sciences et techniques de Tanger
Département Génie Informatique
LST GI S5
Algorithmique et POO C++

