Université Abdelmalek Essaadi
Faculté ses Sciences et techniques de Tanger
Département Génie Informatique
LST GI S5 POO C++
2022/2023

Atelier 3

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

EXERCICE 1 :
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.

EXERCICE 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. Puis, définir deux sous-classes triangle et rectangle, qui calculent l'aire de la zone de shape(). Dans la fonction principale main(), définir deux objets : un triangle et un rectangle, puis appelez la fonction area() dans ces deux classes.

EXERCICE 3 :
Effectuer les opérations arithmétiques sur des **nombres 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. (égalité, addition, soustraction, multiplication, division). *Le choix de l'opération peut être fait par un Menu.*

EXERCICE 4 :
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.

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

EXERCICE 6 :
Créer une classe **Personne** qui comporte trois champs privés, nom, prénom et date de naissance. Cette classe comporte un constructeur pour permettre d'initialiser des données. Elle comporte également une méthode polymorphe *Afficher* pour afficher les données de chaque personne.

- Créer une classe **Employe** 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.
- Créer 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.
- Créer 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.

EXERCICE 7 :

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

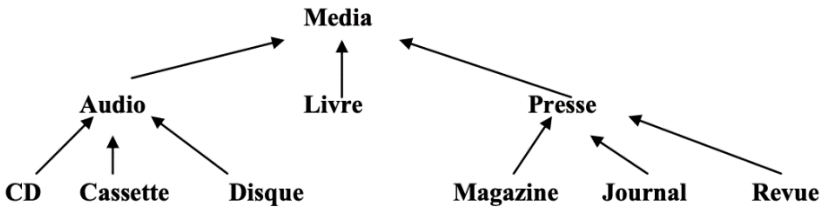
- un constructeur, avec des valeurs par défaut (0),
- une fonction d'affichage des 3 composantes du vecteur, sous la forme : (x, y, z)
- une fonction permettant d'obtenir la somme de 2 vecteurs ;
- une fonction permettant d'obtenir le produit scalaire de 2 vecteurs.
- une fonction coincide permettant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

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

EXERCICE 8 :

En utilisant la notion de **virtual** donner une possible déclaration de la hiérarchie des 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.

EXERCICE 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.