Université Abdelmalek Essaadi
Faculté des Sciences et Techniques de Tanger
Département Génie Informatique
1er Année CI-GI-GEMI | POO C++/Java
Prof. Ikram BENABDELOUAHAB

TD 3 : Héritage et Polymorphisme 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

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 4

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 5 (Extrait Examen 2022-2023)

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 *(constructeur(s))*
- Empiler un élément sur la pile *(push)*
- Dépiler un élément de la pile *(pop)*

Pour simplifier, nous allons supposer que les éléments à empiler sont de type **int**.

Le programme principale 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 dépiler pour vérifier les opérations **push** et **pop**.

EXERCICE 6

Soit une classe vecteur3d définie comme suit :

```
class vecteur3d {
   float x, y, z ;
   public :
   vecteur3d (float c1=0.0, float c2=0.0, float c3=0.0) {
        x = c1 ; y = c2 ; z = c3 ;
   }
} ;
```

Définir les opérateurs == et **!=** de manière qu'ils permettent de tester la coïncidence ou la non-coïncidence de deux points :

a. en utilisant des fonctions membre;
b. en utilisant des fonctions amies.