Quelques exercices autour de l'héritage

Christophe Labourdette

Novembre 2016

1 Des oiseaux

On va construire une hiérarchie de classe, à des fins pédagogique, chaque constructeur et chaque destructeur devra indiquer quand il est appelé.

- Créer une classe creature_volante qui a comme donnée membre nombre_ailes et qui a un constructeur, un destructeur et une fonction affiche.
- Créer une classe animal qui a comme données membres nombre_pattes et type_pelage et qui a un constructeur, un destructeur et une fonction affiche.
- Créer une classe oiseau qui dérive publiquement des classes creature_volante et animal. Elle a comme donnée membre nombre_oeufs et qui a un constructeur, un destructeur et une fonction affiche qui affiche la donnée membre et qui fait appel aux fonctions affiche des classes de base.
- Ecrire un programme qui crée un objet de type oiseau et teste ses fonctions.

2 Des formes

La géométrie est souvent un terrain de jeu pour l'héritage.

1. Construire une classe de base, forme en 2d (fichier .hpp et .cpp) les données seront les coordonnées du centre.
2. Construire des fonctions virtuelles usuelles : aire, perimetre, centre gravite , mais aussi une fonction affiche donnant les caratéristiques de la forme.
3. spécialiser la classe pour : triangle, carré, rectangle, en ajoutant le nombre de côtés et leurs longueur.
4. Implementez une fonction booléenne : intersection indiquant si deux formes ont une partie commune.
5. Comment généraliser les classes pour une dimension de l'espace quelconque.

3 Personnes

1. Dans deux fichiers Personne.hpp et Personne.cpp, écrire le code d'une classe Personne.
   Cette classe a un attribut privé nom de type string.
   Dans la classe Personne,écrire un unique constructeur qui prend une instance de string en argument.
   Ecrire egalement une méthode publique getnom() retournant la valeur de l'attribut.
   Dans le programme principale, écrire la fonction **void** afficheNom(Personne& p) qui affiche sur la sortie standard le nom de la personne.
2. Ecrire une classe Etudiant dérivée de Personne.
   Cette classe dispose d'un attribut filiere de type string (par exemple : "Info1", "Info2" , . . .) et d'un attribut enseignement (par exemple : "algo1,prog1" ou "poo, c++, prog. sys").
   De plus, elle a trois méthodes : filiere() (retournant la valeur de l'attribut),
   enseignement(...) (retournant la valeur de l'attribut),
   setEnseignement(...) (positionnant la valeur de l'attribut).
   Redéfinir la méthode getnom() afin que celle-ci renvoie le nom précédé de la mention "Eleve :" .
   Appliquer la fonction afficheNom() sur une instance de cette classe. Comment écrire la méthode getnom() si son prototype renvoie une référence constante.

3. Ecrire la classe Enseignant , sous-classe de Personne .
   Cette classe dispose d'un attribut service de type int indiquant le nombre d'heures d'enseignement effectuées et une méthode nbHeure() retournant la valeur de cet attribut.
   Redéfinir le méthode getnom() afin que celle-ci renvoie le nom précédé de la mention "Enseignant :".
4. Elève vacataire Certains élèves peuvent ffectuer des enseignements (vacations).
   En utilisant l'héritage multiple, Ecrire une classe EleveVacataire héritant de Enseignant et Eleve .
   La redéfinition de la méthode getnom() appel la méthode nom() des deux classes de base.
   Appliquer la fonction afficheNom() sur une instance de cette classe.
   Expliquer le problème rencontré et la manière de le résoudre.