

TD 6 : héritage Algorithmie Master EEA 1^{ère} année 2006

1. Introduction

1.1. Edition du fichier

Afin d'écrire votre script C++, vous pouvez utiliser n'importe quel éditeur de texte, cependant xemacs ou nedit paraissent être un bon choix. On lance l'éditeur à partir d'une console (« nedit & »).

Le fichier aura une extension .cpp.

1.2. Compilation

Ouvrez un terminal et placez-vous dans le répertoire dans lequel vous avez enregistré votre script C. Afin de compiler votre script, utiliser la commande :

```
g++ nom_du_fichier.cpp -o nom_executable
```

Ce qui donne, par exemple, si votre script du exo1 se nomme exo1.cpp et que vous voulez que votre exécutable se nomme exo1:

```
g++ exo1.cpp -o exo1
```

Le compilateur génère alors un fichier exo1 que vous pouvez exécuter en utilisant la commande ./exo1

1.3. En tête de fichier

L'en-tête du fichier d'extension « .cpp » commencera par :

```
#include <iostream>
using namespace std;
```

La première ligne fait appel à la librairie relative aux fonctions d'entrées sorties.

La seconde ligne définit au compilateur que l'on va utiliser les librairies standard du langage C++.

2. Exercice 1

1. Qu'est ce que l'héritage et qu'est ce qu'une classe mère et une classe fille ?
2. Comment déclare t on qu'une classe B dérive d'une classe A ?
3. De quoi hérite la classe fille ?
4. Complétez le tableau suivant définissant l'accès aux membres d'une classe :

Droit d'accès	public	protected	private
Classe fille			
Classe indépendante			

5. Si on implémente un constructeur au niveau de la classe fille, est ce que le constructeur de la classe mère est systématiquement appelé en premier ?

6. Si la classe fille ne définit pas de constructeur, est ce que le constructeur par défaut de la classe mère est appelé ?
7. Comment définir le constructeur d'une classe fille en spécifiant l'appel d'un constructeur différent de celui par défaut de la classe mère ?
8. Quand est-il pour le destructeur ?
9. Peut-on redéfinir une méthode de la classe mère ?

3. Exercice 2

```
class CPersonne
{
protected:
    char * nom;
    char * adresse;
public:
    CPersonne(const char * name, const char * adress);
    void setNom(const char * name);
    void setAdresse(const char * address);
    void affiche ();
};
-----
CPersonne::CPersonne(const char * name, const char * address) {
    this->setNom(name);
    this->setAdresse(address);
}
void CPersonne::setNom(const char * name) {
    if(name) {
        this->nom = new char[strlen(name)];
        strcpy(this->nom, name);
    }
    else
        nom = NULL;
}
void CPersonne::setAdresse(const char * address) {
    if(address) {
        this->adresse = new char[strlen(address)];
        strcpy(this->adresse, address);
    }
    else this->adresse = NULL;
}
void CPersonne::affiche() {
    cout<< "Nom : " << this->nom << endl;
    cout<< "Adresse : " << adresse << endl;
}
```

1. On veut créer une classe CEtudiant. Un étudiant est une personne, peut on dériver la classe CEtudiant de la classe CPersonne ?
2. Comment créer une classe CEtudiant avec comme paramètre une note ?
3. Comment implémenter le constructeur de CEtudiant ?
4. Peut on créer une méthode affiche membre de CEtudiant ?

5. On définit qu'une personne peut avoir un handicap mais qu'au niveau de l'étudiant, on n'a pas besoin de connaître son handicap, par contre, on pourra connaître son niveau d'handicap. Comment l'implémenter ?
6. Créer une méthode qui estime le temps d'examen d'un étudiant en tenant compte de son niveau d'handicap. On imagine une règle simple (temps x niveau d'handicap).

4. Exercice 3

Créer un fichier « vehicule.h » contenant la déclaration de classe suivante :

```
const float MAX_SPEED = 5000;
class CVehicle {
protected:
    float speedLimit;
    float cx;                //between 0 and 1
public:
    CVehicle();
    ~CVehicle();
    void affiche();
};
```

Créer un fichier « vehicule.cpp » contenant l'implémentation suivante :

```
#include <iostream>
#include "vehicule.h"

using namespace std;

CVehicle::CVehicle(){
    this->speedLimit = 0;
    return;
}
CVehicle::CVehicle(float cx){
    this->cx = cx;
    this->speedLimit = 0;
    return;
}
CVehicle::~~CVehicle(){
}
void CVehicle::affiche() {
    cout << "C'est avant tout un véhicule " << endl;
    return;
}
```

1. On veut créer une classe CCar qui décrit une voiture. Une voiture est caractérisée par le fait qu'elle possède 4 roues. Implémenter cette classe.
2. Créez une méthode affiche de la classe CCar. Créer une fonction main et créez un objet car de CCar. Appelez la méthode affiche.
3. Via l'objet car, appelez la méthode affiche de CVehicle.
4. On estime que le calcul de la vitesse d'un véhicule dépend de son nombre de roue. La formule suivante permet de d'avoir la vitesse maximum : $v = (1-cx) * MAX_SPEED / \text{nombre de roue}$. Créez la méthode permettant de calculer cette vitesse et appelez la à partir du main.
5. On veut maintenant créer un classe CPlane.