

C++: TP05 L'héritage

DAKKAR Borhen-eddine

Lycée le Corbusier

BTS SN

September 28, 2021

1 Objectifs du TP

Dans ce TP nous allons étudier les caractéristiques des fonctions membres d'une classe.

2 Logiciels à utiliser

Sous Linux les outils nécessaire à la compilation du C++ sont déjà présents, il s'agit de "gcc" et d'un éditeur de texte "gedit" par exemple.

Pour compiler un fichier source "main", il suffit d'utiliser la commande suivante:

```
g++ *.cpp -o main
```

Pour l'exécution de fichier ".exe", il suffit d'utiliser la commande suivante:

```
./main
```

3 Exercice 1:

Classe abstraite: est une classe qui ne sera pas utilisée telle quelle car elle est destinée à donner naissance, par héritage, à des classes dérivées.

1. On dispose d'un fichier nommé Point.h contenant la déclaration suivante de la classe point :

```
#include <iostream>
using namespace std;
class Point
{
    private:
        float x,y;
    public:
        Point(float,float);
        float getx();
        float gety();
        void affiche ();
};
```

Définir le constructeur **Point** et les méthodes **float getx()**, **float gety()**, **void affiche ()** qui permettent d'initialiser les coordonnées d'un point, retourner x, retourner y et de les afficher respectivement.

2. Créer une classe **point_polaire**, dérivée de point comportant deux fonctions membres nommées **rho_point** et **theta_point**, fournissant les valeurs des coordonnées polaires d'un point.

- $r = \sqrt{x^2 + y^2}$
- $\theta = \text{atan2}(y, x)$

3. En supposant maintenant que les membres **x** et **y** ont été déclarés protégés (protected) dans point, modifier les deux fonctions **rho_point** et **theta_point**.

4 Exercice 2:

1. Ecrire une classe **Moteur** qui à:
 - Trois attributs privés:
 - Entiers **cylindre** et **puissance**(privés).
 - Char **carburant** (privé).

- Un constructeur qui permet d'initialiser le nombre de **cylindre**, la **puissance** et le type de **carburant**.
- Une méthode **demarrer** le moteur.
- Une méthode **getcylindre()** qui retourne le nombre de cylindre.
- Une méthode **getpuissance()** qui retourne le nombre la puissance d'un moteur.
- Une méthode **getcarburant()** qui retourne le type de carburant utilisé par le moteur.

2. Ecrire une classe **Modele__voiture** qui à:

- Deux attributs:
 - Char **mod__voiture** (privé).
 - Char **couleur** (privé).
- Un constructeur qui permet d'initialiser le modèle de la voiture.
- Une méthode **getmod__voiture()** qui retourne le modèle de la voiture.
- Une méthode **getcouleur()** qui retourne la couleur d'un modèle de voiture.

3. Ecrire une classe **Voiture** qui hérite des deux classes définies précédemment qui possède:

- La date de fabrication **date__fab** comme attribut privé, elle est représentée par un vecteur de 3 entiers qui définit le jour le mois et l'année.
- La **marque** du la voiture qui est un attribut char.
- La **catégorie** du modèle: citadines, berline, break, monospace et SUV.
- Un constructeur qui permet d'initialiser tout les attributs.
- Les méthode **getdate__fab**, **getmarque** et **getcatégorie** qui permettent de retourner les membres privés de la classe **Voiture**.
- Une methode **afficher** qui affiche toutes les caractéristiques d'une voiture.

4. Ecrire une fonction **main** qui ouvre un fichier "Informations_voiture.txt" et qui enregistre toutes les informations d'une nouvelle voiture créée.

Obj	mod_voit	coul	Cyl	puiss	carb	date	Marque	Catégorie
Voiture 1	3008	Bleu	4	100ch	Diesel	01/01/18	Peugeot	SUV
Voiture 2	3008	Noire	4	120ch	Diesel	10/03/19	Peugeot	SUV
Voiture 3	308	Verte	4	120ch	essence	10/03/20	Peugeot	Citadine
Voiture 4	Golf	Rouge	4	136ch	Diesel	12/03/19	volkswagen	Citadine
Voiture 5	Passat	Rouge	4	150ch	Diesel	1/05/21	volkswagen	Berline

5. Utiliser le fichier "Informations_voiture.txt" de la question 3 pour lire les informations d'une voiture, puis créer un vecteur qui contient des copies de cette voiture.
6. Enregistrer par la suite toutes les informations des copies de voiture dans un fichier "copie_voiture.txt".