TP MOPO – Agence immobilière

Préambule

Le but de ce TP est de mettre en place une application programmée en langage C++, et basée sur un modèle objet permettant de stocker les informations relatives au fonctionnement d'une agence immobilière.

Définition du modèle objet

Les données à traiter sont les suivantes :

- Les clients
- Les biens

Données à stocker :

- Client : Nom, prénom, adresse, téléphone, type (acheteur/vendeur/loueur/locataire)
- Biens : Appartement, maison, ou immeuble

Tous les biens possèdent une surface.

Les maisons et les appartements possèdent un certain nombre de pièces, ainsi qu'un statut (en vente, à louer, loué).

Les immeubles possèdent un certain nombre d'appartements.

Mettre en place les classes nécessaires au stockage de ces données à l'intérieur d'une application C++ compilable avec un fichier 'cmake'.

Bien que non obligatoire, il est fortement recommandé d'utiliser la STL, et plus particulièrement std::vector pour le stockage des tableaux dynamiques. Cela vous permettra de facilement créer ou détruire des entrées à l'aide des méthodes 'push_back' et 'erase'.

Implémentation de l'interface utilisateur

Voici les fonctionnalités à implémenter pour permettre à un utilisateur d'interagir avec l'application :

- Ajout d'un nouveau client (demande à l'utilisateur des données associées).
- Ajout d'un bien.
- Suppression d'un client
- Suppression d'un bien

- Affichage du nombre total de clients de l'agence
- Affichage du nombre total de biens de l'agence

- Affichage de la liste des clients
- Affichage synthétique de la liste des biens (Type + surface)

- Affichage d’un bien en particulier

Note :

Vous pouvez utiliser la classe template « std::vector » pour le stockage d’un tableau :

```
#include <vector>

class CClient
{
        // ...
}

std::vector<CClient> aClient;

aClient.push_back(CClient(“jean”, “dupont”, “16 rue Einstein, 25000 Besancon”, “0381500000”));
```

Suppression du 3ème élément d’un std::vector<CClient> :

```
#include <vector>

class CClient
{
        // ...
}

std::vector<CClient> aClient;

aClient.erase(aClient.begin() + 3);
```