

TP4 SLAM2 : GESTION IMMOBILIERE

POO : BILAN CLASSES, OBJETS ET COLLECTION

Objectifs :

- A partir d'un existant, mise en place de nouvelles fonctionnalités
- Rappel des concepts de base de la POO : classes, accesseurs, constructeurs multiples, méthodes, objet et collection d'objets.

Document joint : application commencée en C# sous VS2019 dans l'archive « TP4 POO Bilan (A completer) »

Durée : 1 séance (12 Mars 2021)

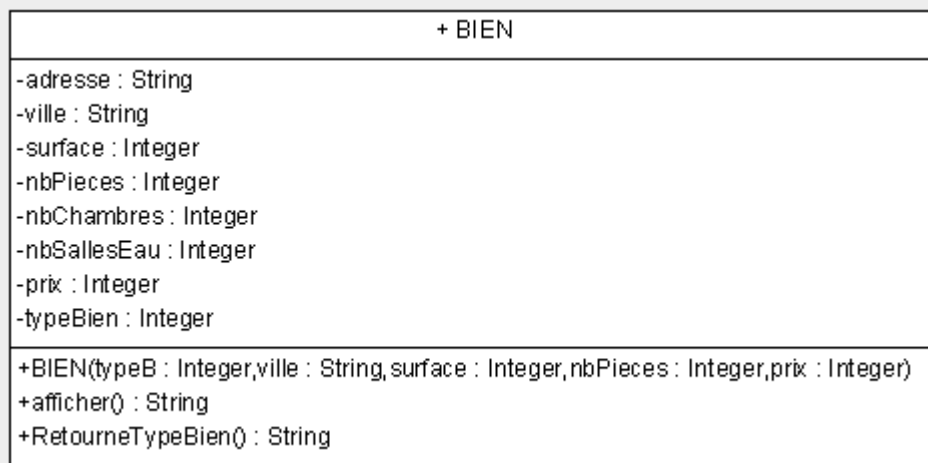
Objectifs de l'application

Cette application doit permettre de définir un bien immobilier, de les lister, de les modifier, de les supprimer afin de créer un catalogue de biens (type Appartement ou Maison) pour une agence immobilière.

L'application fournie met à disposition une interface de gestion des biens, l'objectif est de la compléter pour rendre les fonctionnalités opérationnelles.

PARTIE 1 : CREATION DE LA CLASSE BIEN

Un bien est définie selon les informations recueillies dans le diagramme de classes UML suivant :



Un type de bien vaut 1 si c'est un appartement, 2 si c'est une maison.

La surface, le nb de pièces et le prix doivent être des valeurs positives ou nulles.

Le nb de chambres et de salle d'eau doivent être positives ou nulles et inférieures ou égales au nb de pièces.

La méthode Afficher() retourne une chaîne avec toutes les informations indispensables relatives à un bien du style « L'appartement situé à Angers d'une surface de 60 m2 de 3 pièces pour un prix de 155 000 € ». Vous pouvez compléter cette méthode pour qu'elle retourne plus d'informations si vous le souhaitez.

La méthode RetourneTypeBien() retourne une chaîne correspondant au libellé du type du bien :

si le type de bien est 1, la méthode retourne « Appartement »,

si le type de bien est 2, la méthode retourne « Maison ».

Ce système par type en entier nous permettra facilement d'autre type de bien.

- **La classe est en partie faite, à compléter avec les accesseurs manquants, le constructeur et la méthode RetourneTypeBien()** (voir les commentaires // A COMPLETER)
- Ajouter un **deuxième constructeur** compatible avec l'existant qui vous semble opportun.

PARTIE 2 : AJOUT D'UN BIEN

L'interface de gestion des biens est déjà mise en place :

The screenshot shows a Windows application titled "Gestion des biens immobiliers". The interface is divided into several functional areas. On the left side, there is a "Type du bien" section with two radio buttons: "Appartement" (which is selected) and "Maison". Below this is a "Caractéristiques" section containing input fields for "Surface habitable", "Adresse", and "Ville", along with spinners for "Nb de pièces", "Nb de chambres", and "Nb de salles d'eau". At the bottom left, there is another "Type du bien" section with a "Prix voulu" input field. On the right side, there is a "Consultation des biens" section featuring a "Liste des biens" dropdown menu and two buttons: "MODIFIER" and "SUPPRIMER". At the bottom right of the window, there are three buttons: "AJOUTER" (highlighted in red), "ANNULER", and "QUITTER".

- Les boutons ANNULER et QUITTER sont déjà codés, ainsi que la vérification des caractères saisis sur la surface et le prix (que des entiers et touche BACK sur leur évènement KeyPress).
- Vous disposez d'une collection de biens déclarée et instanciée et d'un objet unBien.

```
// données privées déclarées de la FormBiensImmobiliers
private Bien unBien;
List<Bien> lesBiens = new List<Bien>();
```

- **Compléter l'évènement Click du bouton AJOUTER** avec l'instanciation de l'objet unBien et son ajout à la collection

PARTIE 3 : MODIFICATION ET SUPPRESSION D'UN BIEN

Sur l'interface à droite, la partie « Consultation des biens » permet de lister les biens enregistrés, d'en sélectionner un pour avoir tous ces détails, le modifier ou le supprimer.

Etude de l'existant :

- La méthode `private void CbListeBiens_SelectedIndexChanged(object sender, EventArgs e)` est déjà écrite, elle se déclenche dès qu'un bien est sélectionné dans la liste déroulante (contrôle comboBox CbListeBiens) : bien la comprendre.
- La méthode `private void FormBiensImmobiliers_Load(object sender, EventArgs e)` est déjà écrite : bien la comprendre, que fait cette méthode ? quand est-elle déclenchée ?

Compléter :

- **Compléter les évènements Click des boutons MODIFIER et SUPPRIMER** qui permettent de modifier les informations du bien sélectionné ou de le supprimer.

On souhaite faire évoluer cette application pour noter au fur et à mesure si le bien a été vendu et à quel prix.

1. Modifier en conséquence la classe BIEN pour savoir si le bien est vendu ou non et le prix de vente (penser aux propriétés, accesseurs, modification du ou des constructeurs, ajout éventuel d'une méthode...)
2. Modifier l'interface pour permettre la saisie de la vente d'un bien.
3. Coder la mise en vente d'un bien.