R2.03 – SEANCES 5 ET 6 - TDD
TEST DRIVEN DEVELOPPEMENTS

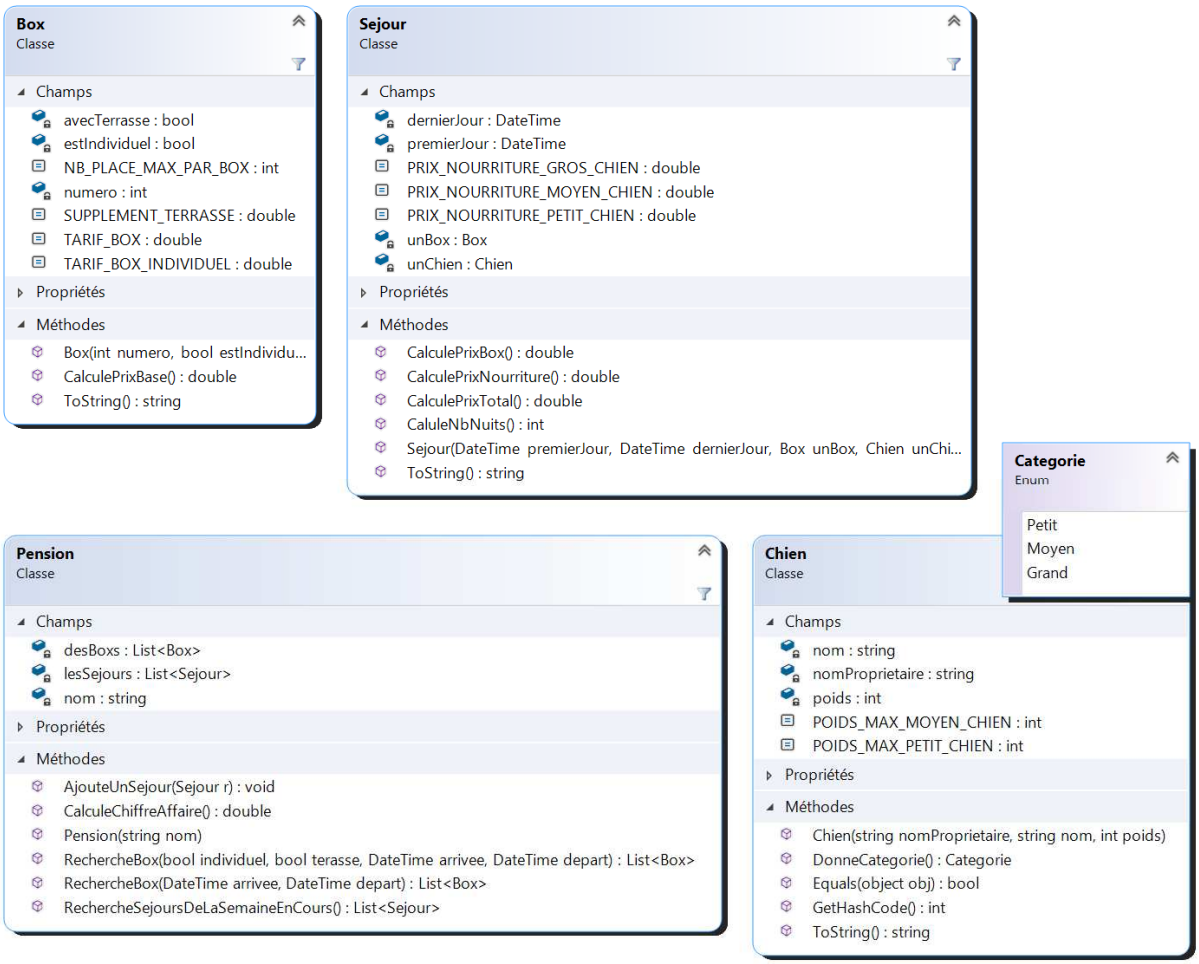
OBJECTIFS

Pratiquer le TDD : écrire les tests unitaires avant d'écrire le code source d'un logiciel.

CONTEXTE

Vous devez faire une appli pour une pension pour chiens.
Le prix d'un box pour une nuit est de 18€ ou 28€ s'il est individuel, à cela peut s'ajouter un supplément de 8 €, s'il est équipé d'une terrasse. Un box peut au plus accueillir 2 chiens.
Le prix d'un séjour dépend du prix du box, du nombre de jours mais aussi du prix de la nourriture qui dépend de la catégorie du chien : petit <10kg, moyen <25, gros >=25
Le prix de la nourriture pour une journée est de 3€ pour les petits chiens, 6€ pour les moyens et de 10€ pour les gros.

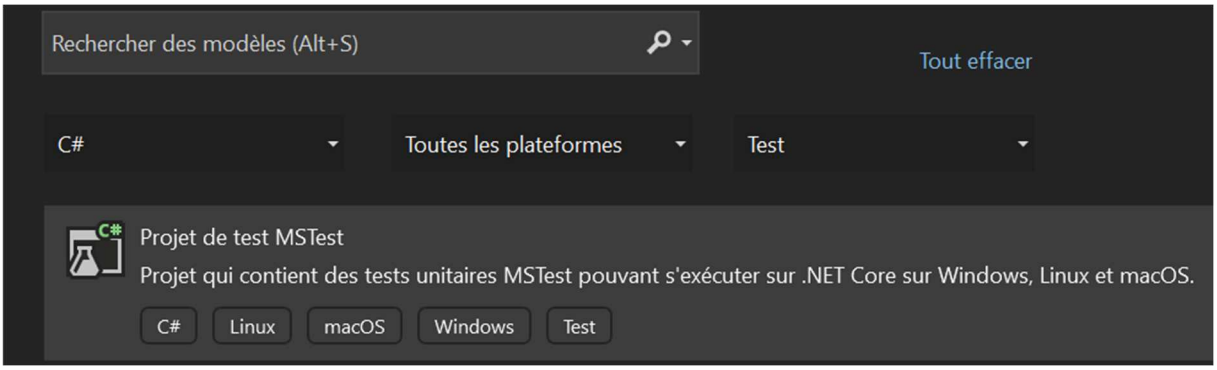
Le nombre de places maximal dans un box sera de 2.



1. **Vous devrez dans un 1er temps réfléchir aux jeux de tests pour les classes : Chien, Box puis Sejour et Pension. Répartissez vous le travail à 2 sur un fichier tableur partagé, une classe par onglet.**

2. **Sur votre bureau, au sein d'une solution nommée « Seance3_4_GestionPensions » , créez 2 projets :**

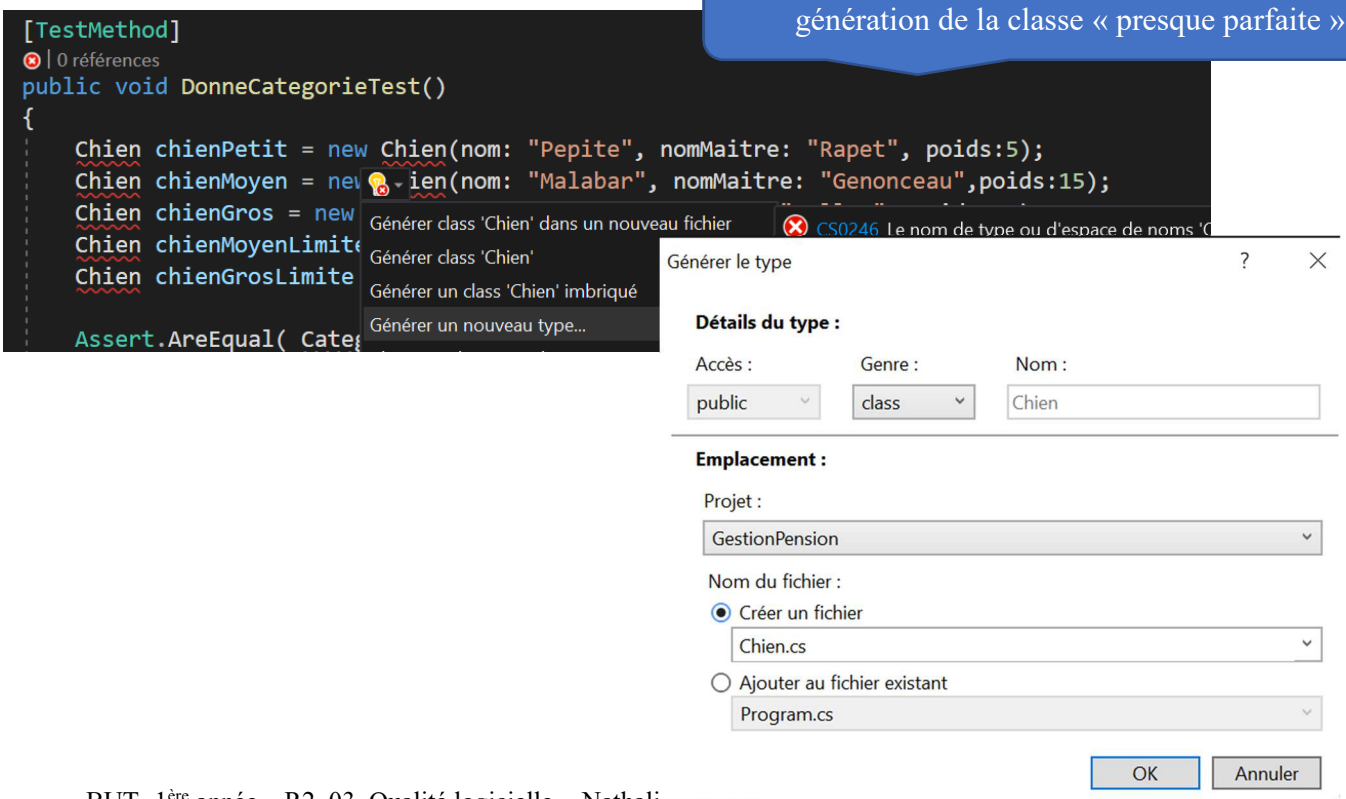
   - **un projet de type application console « GestionPension »**
   - **un projet de tests unitaires « GestionPensionTest ». Attention : bien filtrer C# et Test afin de bien choisir « Projet de test MSTest**



3. **On commencera par tester la classe Chien. Renommez le fichier « UnitTest1.cs » en « ChienTest.cs » et confirmez vouloir changer toutes ses références. Renommez la méthode « TestMethod1 » en « TestDonneCategorie »**

4. **Commencez par coder les tests de DonneCategorie : créez les 5 jeux de tests. Puis cliquez sur le constructeur pour déclencher le menu contextuel ci-dessous :**

En indiquant ici le nom des param, vous aurez une génération de la classe « presque parfaite »



5. Encapsulez les champs de la classe Chien à l'aide du raccourci ctrl + R ctl + E.

6. Au sein de la classe de test, ajoutez le using vers le projet « console »

7. Définissez au dessus de la classe Chien, l'énumération Categorie.

8. **Au sein TestDonneCategorie , faites les 5 Assert nécessaires à tester les 5 jeux de tests (exemple ci-dessous) puis cliquez sur la méthode DonneCategorie pour la générer.**

```
Categorie  categoriePetit = chienPetit.DonneCategorie();
Assert.AreEqual(Categorie.Petit, categoriePetit, "chien petit de 4 kilos <limite ");
```

9. **Puis codez la méthode DonneCategorie avec l'option Live Testing en marche jusqu'à obtenir les coches vertes.**

```
35      [TestMethod]
        0 références
36      public void DonneCategorieTest()
37      {
38          Chien chienPetit = new Chien(nom: "Pepite", nomMaitre: "Rapet", poids:5);
```

10. **Codez les tests d'anomalies sur le constructeur :**
    - **nom, et nomProprio ne doivent être ni nuls ni vides : ArgumentNullException**
    - **poids doit être > 0 : ArgumentOutOfRangeException**

11. **Puis codez les setters**

12. **Faites de même pour Box, Sejour et Pension :** Attention : il faudra veiller à la disponibilité d'un box pour ne pas ajouter un séjour irréalisable.