

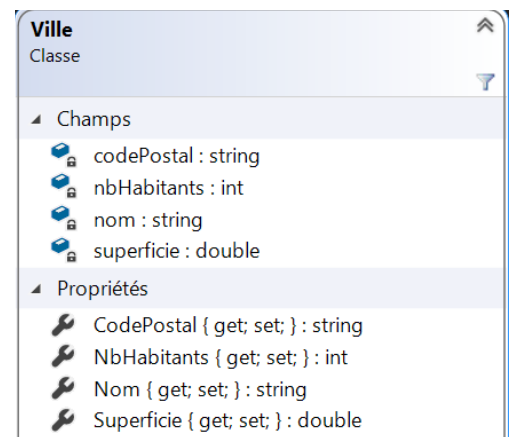
SEANCE 2 - CLASSE VILLE**OBJECTIFS**

- Restituer les automatismes de construction d'une classe
- Comprendre le principe d'encapsulation

CONSIGNES

1. Dans votre projet, ajoutez une classe pour définir le nouveau type Ville. (clic droit sur projet / ajouter nouvelle classe – indiquer bien le nom du fichier Ville.cs)

2. Définissez les 4 champs privés : codePostal (considéré comme une chaîne), nom, nbHabitants, superficie. Puis générez les 4 **propriétés**. Rappel : le champ commence par une minuscule, la propriété publique commence par une majuscule :



3. Générez un **constructeur** pour initialiser tous les champs à partir de 4 paramètres en utilisant les propriétés.

4. **Substituez** la méthode **ToString** ()

5. Ajoutez la constante pour stocker l'unité de mesure de superficie **km2** améliorez ToString.

6. Dans le main, **instanciez** les villes d'Annecy et de Chambéry et affichez les.

7. Améliorez les **propriétés** : ajoutez des vérifications pour contrôler l'intégrité des données :

- Le nom de la ville ne doit être ni nul, ni vide et doit être formaté comme un nom propre : 1ere lettre en majuscule et la suite en minuscule. Sinon lancez une exception de type ArgumentException
- La superficie et le nombre d'habitants doivent être > 0 sinon lancez une exception de type ArgumentOutOfRangeException
- Le code postal ni null, ni vide et composé de 5 chiffres sinon lancez une exception du type FormatException. Rem : utilisez **une expression régulière**

Ici , vous pouvez consulter toutes les exceptions standards

<https://learn.microsoft.com/fr-fr/dotnet/api/system.exception?view=net-7.0#Standard>

8. **Substituez** la méthode **Equals et GetHashCode** ainsi que **les opérateurs**. Attention : ici, Nom et Code Postal suffisent à tester l'égalité et à les différencier. Testez si vos 2 villes sont identiques.

9. **Surchargez** les opérateurs `>` `>=` `<` `<=` : vous testerez le nombre d'habitants . Puis dans le main, comparez vos 2 villes.
10. Définissez la méthode **APlusDeSuperficie** - **public bool APlusDeSuperficie (Ville v)** : renvoie true si la ville a plus de superficie que celle passée en paramètre, false sinon. Dans le main, comparez vos 2 villes.
11. Définissez la méthode **CalculDensite** - **public double CalculDensite ()** : renvoie la densité arrondie à un chiffre après la virgule. Puis utilisez la au sein du main, pour afficher la densité des 2 villes. Ajoutez une constante pour stocker l'unité de mesure de la densité **hab/km2**.
12. Définissez la méthode **EstPlusDense** - **public bool EstPlusDense (Ville v)** : renvoie true si la ville est plus dense que celle passée en paramètre, false sinon. Comparez vos 2 villes.
13. Définissez méthode **EstPrefecture** : return true si les 3 derniers chiffres du code postal sont à 000, sinon false. Testez.
14. Définissez la méthode **EstDansLeMemeDepartement** :
public bool EstDansLeMemeDepartement (Ville v) - renvoie true si la ville est dans le même département que celle passée en paramètre. Vous travaillerez sur les 2 premiers chiffres du code postal.
15. Définissez la méthode **GetDepartement** - **public int GetDepartement ()** : renvoie le numéro sous forme numérique du département . Utilisez-la pour tester.
16. Définissez la méthode **EstDansLeDepartement**- **public bool EstDansLeDepartement (int departement)** : renvoie true si la ville fait partie du département passée en paramètre. Attention : le departement spécifié en paramètre doit être compris entre 1 et 95. Sinon elle lance un `ArgumentOutOfRangeException`.
17. Surchargez l'opérateur + afin de pouvoir fusionner des villes . Ex : l'agglomération d'Annecy c'est la fusion de plusieurs villes. On veut pouvoir écrire : Ville agglomération = ville1 + ville2. Cela cumulera le nombre d'habitants ainsi que la superficie et gardera pour code postal et le nom de ville les éléments de la 1ère ville.
18. Définissez une propriété **Categorie** de type String (sans champ associé), en lecture seule : pas de set, juste la partie get : elle renvoie la catégorie de la ville en fonction de son nombre d'habitants : penser à créer des constantes :
 - bourg : entre 2.000 et 5.000 habitants
 - petite ville : entre 5.001 et 20.000
 - ville moyenne : entre 20.001 et 50.000
 - grande ville : entre 50.001 et 200.000
 - métropole : au-delà.
19. Améliorez l'accessor set de **CodePostal** pour les 2 premiers chiffres forment une combinaison comprise entre 01 et 95.