



# TD-6 Classe, constructeur, propriétés et méthodes

## Objectif du TD

Revoir les principes de base de la POO et manipuler la notion de composition de classe. Une région est composée de communes.

## Exercice 1

Une commune est caractérisée par son nom, son département (int), son pays (string), son maire et le nombre d'habitants. (Les attributs de la classe sont privés)

1. Dans un premier temps, vous créez la classe Commune. Notamment, vous définirez un constructeur permettant d'attribuer toutes les caractéristiques d'une commune.
  - a. Sachant que la population et le département sont des nombres strictement positifs. Dans le cas contraire assignez la valeur -1 qui sera considérée comme un code d'erreur
  - b. Mettre le nom de la commune et pays en lettres majuscules
  - c. Mettre le nom du maire en lettre minuscule
2. Implémentez les méthodes d'instance
  - a. toString qui renvoie une chaîne de caractères décrivant une commune
  - b. equals qui teste si 2 communes sont identiques. On dit que 2 communes sont identiques quand elles ont la même population
3. Implémentez les méthodes statiques d'égalité entre 2 communes
  - a. equalStatic qui teste si 2 communes sont identiques selon le même critère que ci-dessus
4. Créez, au minimum, les propriétés telles que
  - a. La population d'une commune puisse être modifiée avec la même contrainte que celle énoncée dans le constructeur.
  - b. Le nom du maire puisse évoluer
5. Dans le programme principal,
  - a. Tester cette classe, créez les communes Paris, Rouen par exemple
  - b. Changez la population de ces communes
  - c. Affichez les caractéristiques de chacune de ces communes

```
Commune Paris = new Commune("Paris",75, "France","Hidalgo",2200000) ;  
Commune Rouen = new Commune("Rouen",76, "France","Robert",111000) ;  
Paris.Population = 2220000 ;  
string s = Paris.toString() ;
```



```
bool b = Paris.equals(Rouen);  
Console.WriteLine(s);
```

## Exercice 2

- Créer une classe région, qui est composée d'un ensemble de communes. Vous attribuerez un préfet et un chef-lieu pour chaque région
- Créer une méthode toString qui retourne une chaîne décrivant la région
- Vous créez une méthode qui calcule la population d'une région sachant que celle-ci est la somme des populations des communes qui appartiennent à la région
- Créer une méthode qui teste l'appartenance d'une commune dans une région (si le nom de la commune est dans la région)
- Créer une méthode qui trie les communes d'une région par ordre alphabétique (sans utiliser la méthode sort). Vous pouvez adapter la méthode travaillée au TD4 en utilisant la méthode suivante

Vous utiliserez la fonction Compare : `public static int Compare(string stra, string strb);` Qui compare deux objets string spécifiés et retourne un entier qui indique leur position relative dans l'ordre de tri.

Si l'entier retourné est

- inférieur à zéro alors stra est inférieur à strb.
  - zéro alors stra est égal à strb.
  - supérieur à zéro stra est supérieur à strb.
- 
- Dans le programme principal,
  - Créer 2 régions Ile\_De\_France et Normandie qui seront composées de différentes communes
  - Calculez la population de chacune des régions
  - Testez si une commune appartient à la région
  - Triez les communes par ordre alphabétique

Ci-dessous un exemple de Main que vous devez tester



```
static void Main()
{
    Commune Paris = new Commune("Paris",75,"France","Hidalgo",2200000) ;
    Commune Marly = new Commune("Marly",78,"France","Lamy",10000) ;
    Commune Versailles = new Commune("Versailles", 78, "France", "Toto", 100000);
    Commune Rouen = new Commune("Rouen",76,"France","Robert",111000) ;
    Paris.Population = 2220000 ;
    string s = Paris.ToString() ;
    bool b = Paris.equals(Rouen);
    Console.WriteLine(s);

    Commune[] tab = { Marly, Versailles,Paris };

    Region IleDeFrance = new Region("IleDeFrance", tab, "", "Paris");
    Console.WriteLine(IleDeFrance.existeCommune("Marly"));
    IleDeFrance.triCommune();
    s = IleDeFrance.ToString();
    Console.WriteLine(s);
    Console.ReadKey();
}
```