SEQUENCE 1 – PERSISTANCE DE DONNEES
SEANCE 2 - FORMAT TXT, CSV, JSON, XML

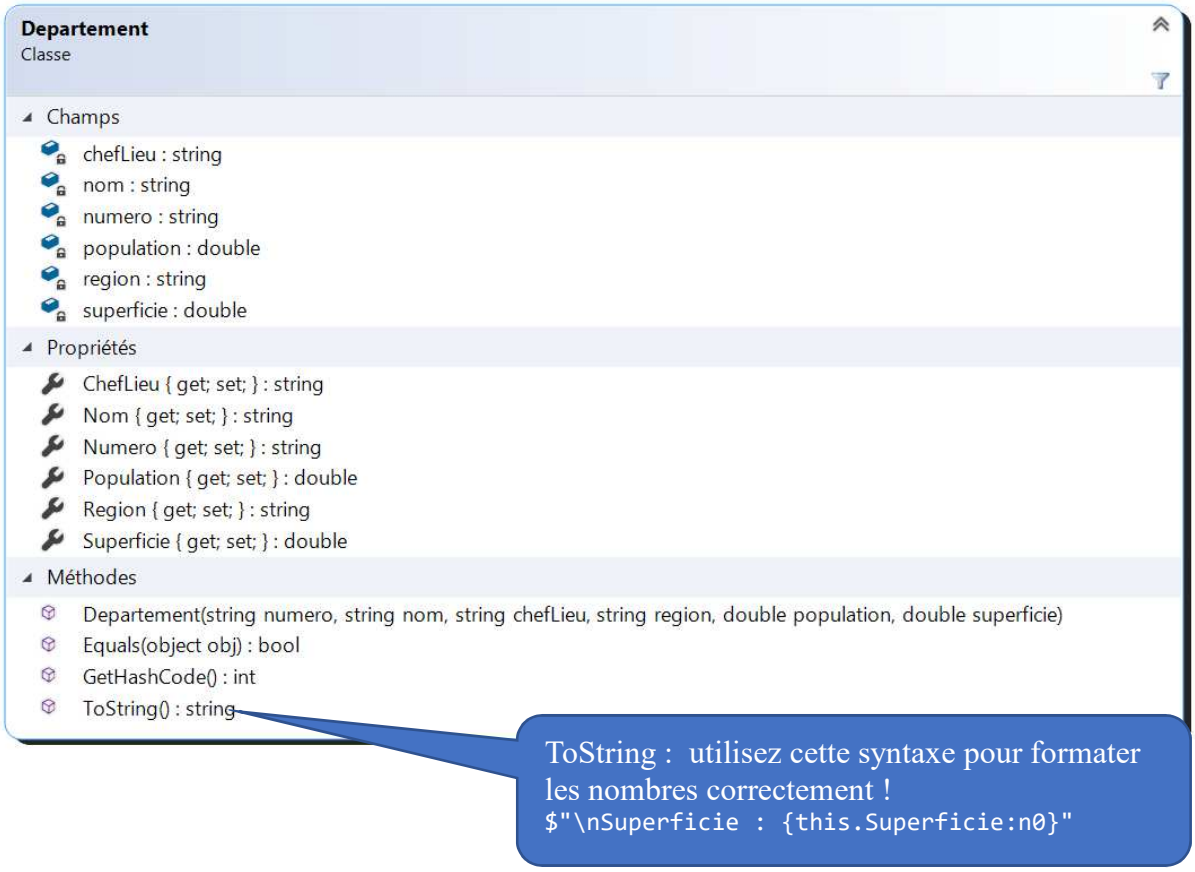
OBJECTIFS

- Utiliser différents types de collection : List, Dictionnary, …
- Maitriser la sérialisation , la désérialisation d'objets.
- Revoir les notions de base d'une classe

EXERCICE 1 :

1. Créez un projet « Exo1_List_Departements » dans une solution « **Seance2_Approfondissement** » au sein de votre répertoire « **Sequence_1_PersistanceCollection** »

2. Définissez la classe Departement :



1. A partir des 2 fichiers mis sur le réseau : Vous chargerez les régions « **regions.csv** » dans une liste de String et les départements « **departementsExo1.json** » dans une liste de Département.

2. Faites les traitements associés au menu ci-dessous :

```
static void Main(string[] args)
{
int choix;
 do
  {
       Console.WriteLine("-----------------------------------------------");
       Console.WriteLine(" 0. Quitter");
       Console.WriteLine(" 1. Voir  en détail un departement");
       Console.WriteLine(" 2. Voir les départements d'une région ");
       Console.WriteLine(" 3. Voir les stats: superficie et population moyenne, min et max");
       Console.WriteLine(" 4. Voir les 10 départements les plus habités ");
       Console.WriteLine(" 5. Voir les 10 départements les plus grands ");
       Console.WriteLine("-----------------------------------------------");
      choix = Program.SaisieInt(0, 5);
       switch (choix)
       {
              case 0: break;
              case 1:
                     Console.WriteLine("Numéro du departement ?");
                     String numDep = Console.ReadLine();
                     Console.WriteLine("-----------------------------------------------");
                     Console.WriteLine(" A FAIRE 1 ");
                     break;
              case 2: Console.WriteLine(" A FAIRE 2 ");  break;
              case 3: Console.WriteLine(" A FAIRE 3 ");  break;
              case 4: Console.WriteLine(" A FAIRE 4 ");  break;
              case 5: Console.WriteLine(" A FAIRE 5 ");  break;
       }
       if (choix != 0)
       {
              Console.ReadLine();
              Console.Clear();
       }
    }  while (choix != 0);
   Console.WriteLine("FIN");
}
private static int SaisieInt(int min, int max)
   {
    int nb =0; bool ok;
    String choix = Console.ReadLine();
    do
       {
         ok = true;
          if (!(int.TryParse(choix, out nb) && nb >= min && nb <= max))
                {
                    Console.WriteLine($"Erreur- Choix entre {min} et {max} :");
                    choix = Console.ReadLine();
                    ok = false;
                }
         } while (!ok);
       return nb; }
```

En suivant les consignes ci-dessous : pour l'option :

1. Vous demanderez à l'utilisateur le numéro du département. Pensez à afficher un message si celui donne un numéro qui ne correspond à aucun département. **Utilisez la méthode : Find sur la liste des départements.**
   Exemple :

```
Departement d = lesDepartements.Find(x => x.ChefLieu == "Annecy");
Find recherche un département dans la liste ayant pour ChefLieu Annecy.
X est utilisé de manière générique pour exprimer une condition  sur un
objet x de classe Departement faisant parti de la liste.
```

2. Vous afficherez dans un 1er temps les différentes régions(et un numéro pour faire un sous menu) à l'aide de la liste des régions afin de faciliter la saisie ! **Et ensuite, vous utiliserez la méthode : FindAll pour trouver les départements de la région selectionnée.**

```
2
------------------------------
1 . Auvergne-Rhône-Alpes
2 . Bourgogne-Franche-Comté
3 . Bretagne
4 . Centre-Val de Loire
5 . Corse
6 . Grand Est
7 . Guadeloupe
8 . Guyane
9 . Hauts-de-France
10 . Ile-de-France
11 . La Réunion
12 . Martinique
13 . Mayotte
14 . Normandie
15 . Nouvelle-Aquitaine
16 . Occitanie
```

3. **Vous devrez parcourir la liste pour calculer moyennes, min et max**

```
Superficie moyenne : 6 268 ( min :  105 - max : 83 534 )
Population moyenne : 657 889 ( min : 76 309 - max : 2 605 238 )
```

4. **Conseil : faites un tri sur la population puis affichez les 10 derniers en partant de la fin. Remarque : étant donné qu'il y aura 2 tris possibles ici au lieu de définir CompareTo qui ne fonctionnera que pour un seul tri ! Vous pouvez définir des méthodes de comparaison static que vous passerez à la méthode Sort ! ex :**

```
les 10 départements les plus habités
--------------------------------------
1.Nord-2 605 238
2.Paris-2 206 488
3.Bouches-du-Rhône-2 016 622
4.Rhône-1 821 995
5.Hauts-de-Seine-1 601 569
6.Seine-St-Denis-1 592 663
7.Gironde-1 548 478
8.Pas-de-Calais-1 472 648
9.Yvelines-1 427 291
10.Seine-et-Marne-1 390 121
```

Dans Departement.cs

```
public static int CompareDepartementParPopulation(Departement d1, Departement d2)
       {
            return d1.Population.CompareTo(d2.Population);
        }
```

Dans Program.cs

```
       lesDepartements.Sort(Departement.CompareDepartementParPopulation);
```

EXERCICE 2 :

1. Ajoutez un projet Exo2_Dictionary à votre solution.

2. Vous prendrez cette fois-ci la version V2 du fichier « **departementsExo2.json** ».
   Et vous définirez non pas une liste mais un dictionary pour désérialiser les données contenues dans le fichier. Puis affichez les données contenus dans le dictionary.

```
Dictionary<String, Departement> lesDepartements = new Dictionary<string, Departement>();
```

3. Refaites l'option 1 du menu précédent.

4. Vous envisagez de refaire les autres options du menu de l'exo 1. Vous aviez utilisé FindAll, et Sort pour alléger votre code. Cela est-il possible avec un Dictionary ? Consultez la doc …

CONCLUSION

On choisit un type de collections : List, Dictionary, …en fonction de son usage !