

Exercice sur les classes en C#

- 1- Démarrez un nouveau projet C# en mode console. Appelez ce projet **ONU**.
- 2- Ajoutez une classe, **CoordGeo**, possédant deux « automatic properties » :
 - `public double Latitude { get; set;}`
 - `public double Longitude { get; set;}`
- 3- Ajoutez un constructeur avec deux paramètres de type double. Ils initialiseront la latitude et la longitude. Les degrés de la coordonnée correspondront à la portion entière du double et les minutes à la portion décimale. Ignorez les secondes. Si une latitude est au Sud de l'Équateur, mettez le double négatif, idem pour les longitudes à l'Ouest de Greenwich.
- 4- Ajoutez à cette classe une méthode publique **Afficher()** qui affichera la Latitude (position nord-sud) et la Longitude (position est-ouest). Testez le constructeur et l'affichage avec les coordonnées de Montréal:

```
namespace ONU
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("projet ONU");

            CoordGeo coord = new CoordGeo(45.3, -73.33);
            coord.Afficher();
        }
    }
}
```

- 5- Vous devriez avoir le résultat suivant :

```
Projet ONU!
Lat: 45,3 Nord
Long:73,33 Ouest
```

- 6- Ajoutez une seconde classe à votre projet : **Ville**. Cette classe aura trois propriétés :
 - Nom** (string)
 - Position** (CoordGeo)
 - Population** (int)
- 7- Ajoutez un constructeur à trois paramètres pour la classe **Ville** :
 - param1 : string (pour initialiser le Nom)
 - param2 : CoordGeo (pour initialiser Position)
 - param3 : int (pour initialiser Population)
- 8- Ajoutez à la classe **Ville** une méthode publique **Afficher()** pour afficher toutes ses informations. Pour afficher la Position d'une ville, utilisez la méthode déjà définie au point 4- .

- 9- Effectuez une recherche sur Google pour trouver les coordonnées géographiques réelles des villes suivantes:

Greenwich
Saint-Jérôme
Vancouver
Quito
Jakarta

- 10- Dans la fonction **Main()** de votre programme, testez votre classe Ville en instantiant cinq objets de la classe Ville, avec les paramètres spécifiés précédemment.
- 11- Définissez une troisième classe : **Pays**, qui possédera trois propriétés :
- Nom** (type string),
 - Capitale** (classe Ville),
 - ListeVilles** (de type List<Ville>)
- 12- List<> est une classe offerte par l'environnement C#, c'est une liste générique et vous devez lui fournir le type qui sera mis dans la liste. Ici on veut une liste de villes alors l'attribut sera de type **List<Ville>**
- 13- Ajoutez un constructeur à la classe **Pays** possédant deux paramètres : un de type string et le second de classe **Ville**. Le nom du pays sera le paramètre string, et le paramètre **Ville** sera la capitale du pays. Initialiser ListeVilles ainsi :
- ```
ListeVilles = new List<Ville>();
```
- 14- Ajoutez à la classe Pays une méthode **AjouterVille()** qui recevra un objet Ville en paramètre et qui ajoutera cette ville à **listeVilles** du **Pays**.
- 15- Ajoutez une méthode **CalculerPopulation()** à la classe Pays, cette méthode parcourra **ListeVilles** et fera la somme de toutes les populations de chaque ville du pays et retournera cette somme
- 16- Ajoutez une méthode **Afficher()** toutes les informations d'un pays : son nom, sa population totale, ses villes;
- 17- Trouvez les noms, positions géographiques et populations des 10 plus grandes villes du Canada puis instanciez tous les objets nécessaires à l'instanciation de ce pays et afficher ses informations
- 18- Ajoutez deux paramètres à la méthode Afficher() d'un Pays : le premier sera le critère de tri désiré pour trier les villes (population ou position géographique), le deuxième sera un paramètre facultatif (qui possède une valeur par défaut) qui indiquera si on a un ordre croissant ou décroissant (pour la position géo : croissant = Est vers l'Ouest vice versa pour décroissant)
- 19- Modifier la méthode d'affichage des villes pour tenir compte des paramètres de tri et faites afficher les infos pour le Canada
- 20- Ajoutez une méthode pour écrire les informations d'un pays dans un fichier. Le fichier créé devra être dans un dossier ONU et s'appeler **nomDuPays.txt**. Chaque ligne de ce

fichier sera une ville de ce pays. Par exemple, le contenu du fichier Canada.txt :

Montréal;2000000;45.3;-73.33

Ottawa;1017000;45.25;-75.41

- 21- Ajoutez une méthode capable de charger un pays en mémoire à partir du fichier texte
- 22- Créez une classe ONU qui contiendra une List<Pays>. Ajoutez une méthode à cette classe capable de charger tous les fichiers se trouvant dans le dossier ONU et d'initialiser la List<Pays>
- 23- Ajoutez à la classe ONU une méthode pour afficher le sommaire des pays de sa liste. Vous indiquez combien de pays la liste contient, combien de villes et quelle est la population totale. Puis pour chaque pays de liste vous affichez le nom du pays, sont nombres de villes et sa population totale
- 24- Ajoutez à la classe ONU une méthode pour trier sa liste de pays. On veut les possibilités de tri suivantes :
  - Par ordre alphabétique de nom de pays
  - Par ordre de grandeur de la population totale des pays
  - Par ordre de position NordSud de la capitale des pays
  - Par ordre de position EstOuest de la capitale des pays
  - Par ordre de grandeur du nombre de villes dans les pays
- 25- Chacun des critères de tri pourra être croissant ou décroissant