

Exercice 02 – Suite de Fibonacci

Objectif

Appréhender la manipulation du framework de test **NUnit** dans la réalisation de plusieurs tests unitaires utilisant les particularités du framework par rapport à **MSTest** à partir d'une classe donnée : **Fibo.cs**.

Sujet

Vous réaliserez plusieurs tests par rapport à la **classe fournie** (dont vous ne modifierez pas le code). Pour ce faire, vous utiliserez un projet de type **NUnit** qui utilisera une **déclaration de la classe de façon unique** et qui sera la même pour les différents tests. Vous ferez attention à respecter les **conventions de nommage** dans le but de rendre vos tests clairs et lisibles pour vos potentiels collègues.

Les tests, quant à eux, viseront à couvrir plusieurs scénarios, tel que :

- Lors du déclenchement de la fonction **GetFiboSeries()** avec un **Range** de 1
 - Le résultat **n'est pas vide**
 - Le résultat est **trié de façon ascendante**
 - Le résultat correspond à une **liste qui contient {0}**
- Lors du déclenchement de la fonction **GetFiboSeries()** avec un **Range** de 6
 - Le résultat **contient le chiffre 3**
 - Le résultat **contient 6 éléments**
 - Le résultat **n'a pas le chiffre 4 en son sein**
 - Le résultat correspond à une liste qui contient **{0, 1, 1, 2, 3, 5}**

```
3 références
public class Fibo
{
    5 références | 2/2 ayant réussi
    public int Range { get; set; }
    1 référence
    public Fibo()
    {
        Range = 5;
    }
    2 références | 2/2 ayant réussi
    public List<int> GetFiboSeries()
    {
        List<int> fiboSeries = new();
        int a = 0, b = 1, c = 0;
        if (Range == 1)
        {
            fiboSeries.Add(0);
            return fiboSeries;
        }
        fiboSeries.Add(0);
        fiboSeries.Add(1);
        for (int i = 2; i < Range; i++)
        {
            c = a + b;
            fiboSeries.Add(c);
            a = b;
            b = c;
        }

        return fiboSeries;
    }
}
```