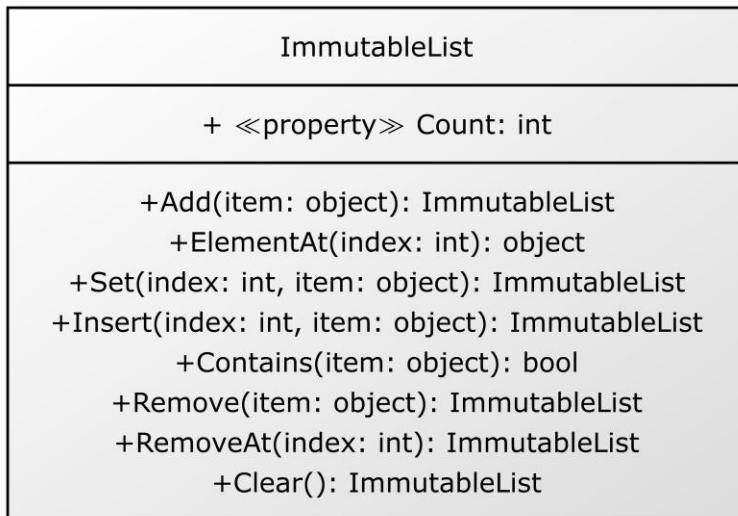


# Práctica 2 - Programación Orientada a Objetos

## Lista inmutable (de objects)



CREATED WITH YUML

Figura 1: Diagrama UML de la lista inmutable.

Implementa una lista inmutable similar a una lista enlazada. Las operaciones de añadir y eliminar elementos no modifican la lista original. En su lugar, cada método devuelve una nueva lista con los cambios aplicados.

Por ejemplo:

```
var list1 = new ImmutableList();      // Lista vacía
var list2 = list1.Add("1");          // list1 sigue vacía, list2 = ["1"]
var list3 = list2.Add(null);         // list2 = ["1"], list3 = ["1", null]
var list4 = list3.Add(2);            // list3 = ["1", null], list4 = ["1",
null, 2]
if (list4.Contains(null))           // se imprime "TRUE"
    Console.WriteLine("TRUE");
var list5 = list4.Remove(null);       // list4 = ["1", null, 2], list5 = ["1",
2]

Console.WriteLine(list1.Count);      // 0
Console.WriteLine(list2.Count);      // 1
Console.WriteLine(list3.Count);      // 2
Console.WriteLine(list4.Count);      // 3
Console.WriteLine(list5.Count);      // 2
```

Implementa un proyecto de prueba para la clase `ImmutableList`. Las pruebas deberán validar el correcto funcionamiento de los métodos implementados, empleando distintos tipos de datos y contemplando casos normales y casos límite (por ejemplo, valores nulos, índices fuera de rango o estructuras vacías).

Esta actividad deberá subirse al Campus Virtual antes del comienzo de la siguiente sesión de laboratorios.