

2-7/2-7_code.cs

```
using System;
using System.Collections;
using System.Security.Cryptography.X509Certificates;
class Test{
    public static void Main(){
        int i;
        var mc = new MyClass(10);
        for (i = 0; i < mc.Length; i+=1) {
            mc[i] = i + 1;
        }
        Console.WriteLine(mc.bruh);

        //foreach (var c in mc){
            //Console.WriteLine(c);
        //}

        //Lambda function
        mc.ForEach(v => Console.WriteLine(v));

        //mc.Mutate(v => return_dub(v));
        mc.ForEach(v => Console.WriteLine(v));

    }
    public static double return_dub(double b){
        return b + 1;
    }
}
```

```
class MyClass : IEnumerable {
    private double[] _data;
    public string bruh = "bruh";
    public MyClass(int capacity){
        _data = new double[capacity];
    }
    public MyClass() : this(1) {
    }
    public int Length {
        get => _data.Length;
    }
    public double this[int index]{
        get {
            return _data[index];
        }
        // For a property or indexer "value" in the set block has a special meaning
    }
}
```

```
        set => _data[index] = value;
    }

    public IEnumerator GetEnumerator() {
        return new MyClassEnumerator(_data);
    }

    //labmda stuff
    public delegate void ImmutableFunc(double v);

    public void ForEach(ImmutableFunc f){
        int i;
        for(i = 0; i < Length; i+=1) {
            f(_data[i] );
        }
    }

    public delegate double MutableFunc(double v);
    public void Mutate(MutableFunc f){
        int i;
        for (i = 0; i < Length; i+=1){
            _data[i] = f(_data[i]);
        }
    }

}

class MyClassEnumerator : IEnumerator {
    private double[] _dataref;
    private int _pos;
    public MyClassEnumerator(double[] data){
        _dataref = data;
        Reset();
    }
    public void Reset(){
        _pos = -1;
    }
    public object Current{
        get => _dataref[_pos];
    }
    public bool MoveNext(){
        _pos += 1;
        return (_pos < _dataref.Length);
    }
}
```