VILNIUS UNIVERSITY
ŠIAULIAI ACADEMY

BACHELOR PROGRAMME SOFTWARE ENGINEERING

Object Oriented Programming

Practical 7 (Seven).

Student: Sunday Emmanuel Sanni

Lecturer: Prof. Donatas Dervinis

Šiauliai,

28/04/2024

1. Create C# OOP a program (class and method) to calculate the mathematical fibonacci sequence1,2,3,5,8, (or other https://mathigon.org/course/sequences/introduction)

To solve the problem above, I created a class called **Fibonacci** that has 3 properties which are:

_input for user input,

_maxValue for the maximum value of the sequence, and

_Count for the number of iterations in the Fibonacci sequence.

There are also 4 other methods in the class associated with generating the needed Fibonacci sequence for a given number. They are:

GenerateSequence: This is the main method responsible for generating the Fibonacci sequence. It accepts a parameter which is the input number by the user. The method is a list of the type int. It stores all the values of the Fibonacci sequence in a list and returns the list.

GetMaxValue: it returns thee maximum value

GetIterationCount: returns the number of iterations

PrintTable: this is a static method that prints out the values of the generated Fibonacci sequence. It accepts a parameter of type integer which is a list.

The code is shared below:

```
class Fibonacci
    {
        public int _input;
        private int _maxValue;
        private int _Count = 0;

        public List<int> GenerateSequence(int input)
        {
            _input = input;
            List<int> SingleTable = [];
            if (_input == 0)
            {
                SingleTable.Add(0);
                _Count = 0;
                _maxValue = 0;
            }
            else if (_input == 1)
            {
                SingleTable.Add(0);
                SingleTable.Add(1);
                SingleTable.Add(1);
                _Count = 2;
                _maxValue = 1;
            }
            else
            {
                SingleTable.Add(0);
                SingleTable.Add(1);
                _Count = 1;
                _maxValue = SingleTable[_Count];
                while ((_maxValue + SingleTable[_Count - 1]) < input)
                {
                    SingleTable.Add(SingleTable[_Count - 1] + SingleTable[_Count]);
                    _maxValue = SingleTable[_Count + 1];
                    _Count++;
                }
                _Count = SingleTable.Count;
            }
            return SingleTable;
        }

        public static void PrintTable(List<int> Table)
        {
            for (int i = 0; i < Table.Count; i++)
            {
                Console.Write(Table[i]);
                if (i != Table.Count - 1)
                {
                    Console.Write(", ");
                }
            }
        }

        public int GetMaxValue()
        {
            return _maxValue;
        }
        public int GetIterationCount()
        {
            return _Count;
        }
}
```

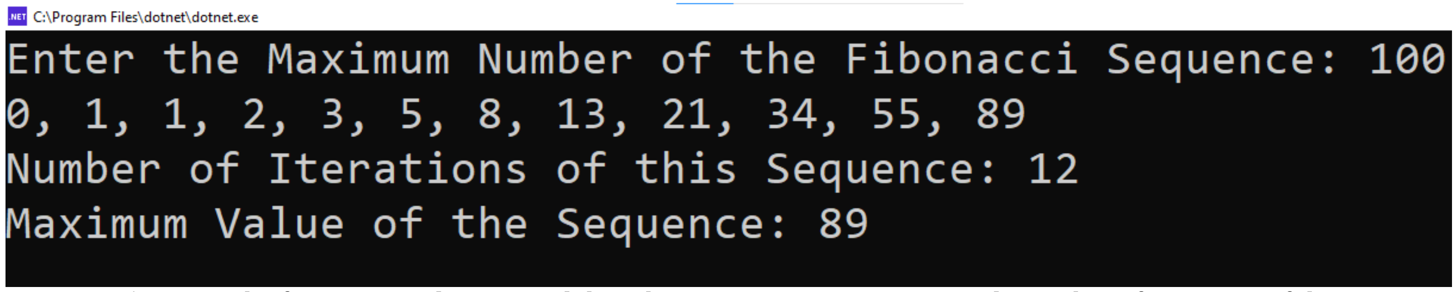
```
class Program
   {
       static void Main(string[] args)
       {
           Console.Write("Enter the Maximum Number of the Fibonacci Sequence: ");
           int entry = Convert.ToInt32(Console.ReadLine());
           Fibonacci number = new();
           List<int> Table = number.GenerateSequence(entry);
           Fibonacci.PrintTable(Table);
           int MaxIteration = number.GetIterationCount();
           int MaxValue = number.GetMaxValue();
           Console.WriteLine("\nNumber of Iterations of this Sequence: " + MaxIteration);
           Console.WriteLine("Maximum Value of the Sequence: " + MaxValue);

           Console.WriteLine("\n--------------------------------");
            }
```

In the main program class, the console asks user for input. This input is converted to an integer. An instance of the class is created, and the method responsible for generating a Fibonacci sequence is called with the user’s input as the parameter.

The print method is called with the output of the sequence generated as its parameter, and the result is displayed.

2. First test the program with a single number and print all the values, find max value (Highest number) and count of iterations (Total stopping time).



C:\Program Files\dotnet\dotnet.exe

```
Enter the Maximum Number of the Fibonacci Sequence: 100
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89
Number of Iterations of this Sequence: 12
Maximum Value of the Sequence: 89
```

Figure1: The figure entered is 100 and the Fibonacci sequence is generated. Number of iterations of the sequence and the maximum value of the sequence is also printed.

3. Create extra method for calculating functions values (max value and count of iterations) of numbers range (e.g., 10 to 10^99). The console must be prompted for range values (start and finish). As results store in array: numbers , maximum (highest) values and the number of iterations of them.

This extra method for using a range of values for generating a Fibonacci sequence is called:

GetArrayOfResults: It takes two parameters which are the lowerBound and upperBound. It uses the function **GenerateSequence** from the first solution to generate fibonacci sequence for the upperBound. This sequence is stored into a list. The list iterated through so that numbers within the range of values entered by the user is stored into another list, and the first list is deleted.

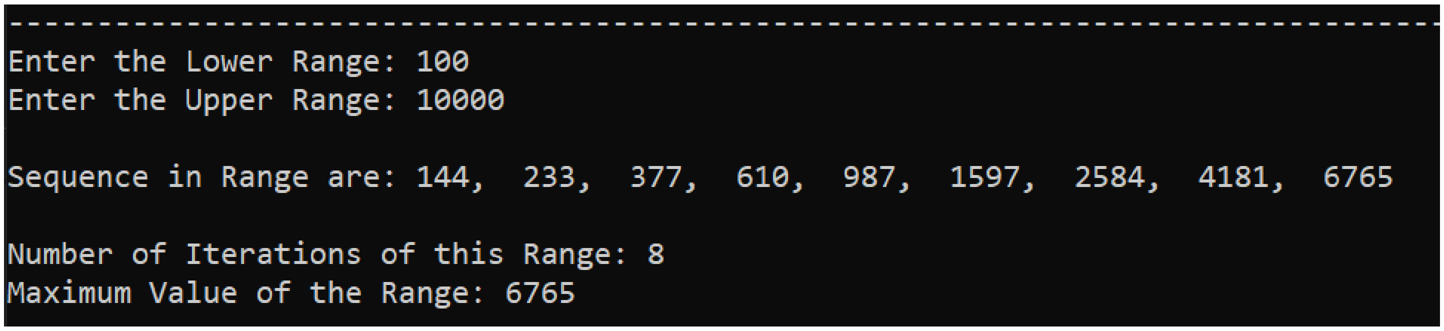
PrintGrid: This is a static method that accepts a list and prints out the values stored in it.

```
public List<int> GetArrayOfResults(int lowerBound, int upperBound)
    {
        List<int>? ExternalTable = GenerateSequence(upperBound);
        List<int> FinalTable = [];

        for (int i = 0; i < ExternalTable.Count; i++)
        {
            if (ExternalTable[i] >= lowerBound)
            {
                FinalTable.Add(ExternalTable[i]);
            }
        }
        ExternalTable = null;
        return FinalTable;
    }

    public static void PrintGrid(List<int> _result)
    {
        foreach (var value in _result)
        {
            Console.Write(value);
            if (value != _result[_result.Count - 1])
                Console.Write(",  ");
        }
        Console.WriteLine();
    }

  }
```



```
--------------------------------------------------------------------------------
Enter the Lower Range: 100
Enter the Upper Range: 10000

Sequence in Range are: 144,  233,  377,  610,  987,  1597,  2584,  4181,  6765

Number of Iterations of this Range: 8
Maximum Value of the Range: 6765
```

Figure 2: A range of 100 – 1,000 is entered and the result is displayed above with number of iterations and maximum number in range

```
Console.Write("Enter the Lower Range: ");
        int lowerRange = Convert.ToInt32(Console.ReadLine());

        Console.Write("Enter the Upper Range: ");
        int upperRange = Convert.ToInt32(Console.ReadLine());

        Console.Write("\nSequence in Range are: ");
        List<int> Result = number.GetArrayOfResults(lowerRange, upperRange);
        Fibonacci.PrintGrid(Result);
        Console.WriteLine("\nNumber of Iterations of this Range: " + (Result.Count - 1));
        Console.WriteLine("Maximum Value of the Range: " + Result[Result.Count - 1]);
```

The code above is the implementation of the extra method, in the main function. The range of values are accepted from the users and converted to integers. Since an instance of the class had earlier been created, the extra method is called with parameters provided by the user. Then then static print method is called to show the output of the list used.

In conclusion, C# OOP has implementation that varies slightly from PHP. OOP implementation is generally a very efficient way to cod and get solutions.