**19. Документирование в формате XML**

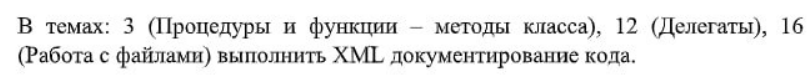
 Задание 1.

Рисунок 19.1 – Условие задания

Источник: собственная разработка

Листинг программы:

namespace zad1

{

internal class Program

{

/// <summary>

/// Calculation methods

/// </summary>

/// <param name="x">x side value</param>

/// <param name="n">y side value</param>

/// <returns>

/// <param name="y">return out param y </param>

/// </returns>

static void F(double x, double n)

{

double y;

y = (Math.Pow(x, n)) / (n);

Console.WriteLine(y);

}

static void Main(string[] args)

{

double n;

double x;

Console.Write("Введите x: ");

x = Double.Parse(Console.ReadLine());

Console.Write("Введите n: ");

n = Double.Parse(Console.ReadLine());

F(x, n);

Console.ReadLine();

}

}

}

Таблица 19.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Предыдущие задания | Документирование |

Анализ результатов:

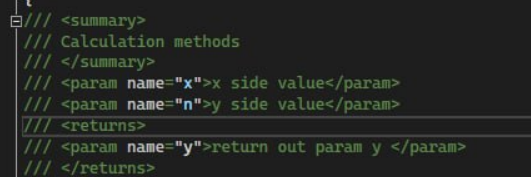


Рисунок 19.2 – Результат работы программы

Источник: собственная разработка

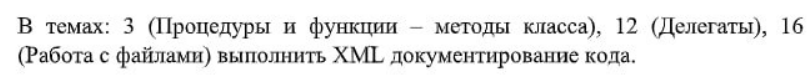
Задание 2.

Рисунок 19.3 – Условие задания

Источник: собственная разработка

Листинг программы:

using System;

namespace zad2

{

public class Array

{

public int[,] Arr;

public Array(int row, int col)

{

this.Arr = new int[row, col];

}

/// <summary>

/// Method of Get Value for array

/// </summary>

/// <param name="row"></param>

/// <param name="col"></param>

/// <returns>

/// Arr Array

/// </returns>

public int GetValue(int row, int col)

{

return Arr[row, col];

}

/// <summary>

/// Method of Print array

/// </summary>

/// <param name="row"></param>

/// <param name="col"></param>

/// <param name="val"></param>

/// <returns>

/// <param name="val">return param uot of Print</param>

/// </returns>

public int SetValue(int row, int col, int val)

{

return Arr[row, col] = val;

}

/// <summary>

/// Method of Getting Length for Array

/// </summary>

/// <param name="d"></param>

/// <returns>

/// d value

/// </returns>

public int GetLength(int d)

{

return Arr.GetLength(d);

}

/// <summary>

/// Method of Getting array

/// </summary>

/// <param name="random"></param>

/// <returns>

/// <param name="i">return param uot of Print</param>

/// <param name="j">return param uot of Print</param>

/// </returns>

public Array GetArr(Array arr)

{

Random random = new Random();

for (int i = 0; i < arr.GetLength(0); i++)

{

for (int j = 0; j < arr.GetLength(1); j++)

{

SetValue(i, j, random.Next(10));

Console.Write("{0,4}", arr.GetValue(i, j));

}

Console.WriteLine();

}

return arr;

}

public static Array operator +(Array Arr, int num)

{

for (int j = 0; j < Arr.GetLength(0); j++)

{

Arr.SetValue(j, 0, Arr.GetValue(j, 0) + Arr.GetValue(j, num));

}

return Arr;

}

/// <summary>

/// Method of Print array

/// </summary>

/// <param name="Arr"></param>

/// <returns>

/// <param name="i">return param uot of Print</param>

/// <param name="j">return param uot of Print</param>>

/// </returns>

public void Print(Array Arr)

{

for (int i = 0; i < Arr.GetLength(0); i++)

{

for (int j = 0; j < Arr.GetLength(1); j++)

{

Console.Write("{0,4}", Arr.GetValue(i, j));

}

Console.WriteLine();

}

}

}

class Program

{

static void Main(string[] args)

{

Array Arr = new Array(3, 4);

Arr.GetArr(Arr);

Console.WriteLine();

Arr.Print(Arr + 2);

Console.ReadLine();

}

}

}

Таблица 19.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Предыдущие задания | Документирование |

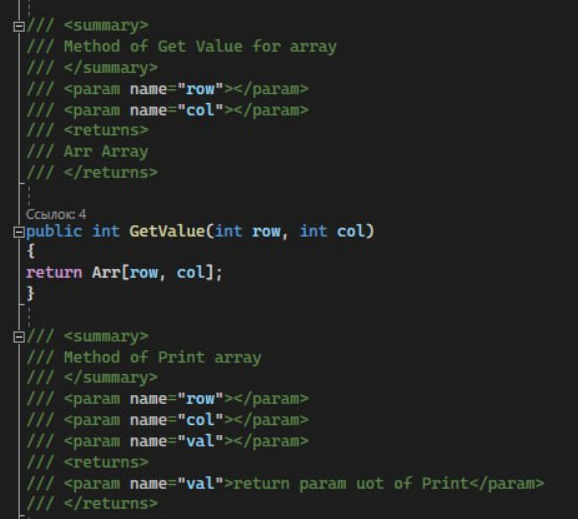
Анализ результатов:

Рисунок 19.4 – Результат работы программы

Источник: собственная разработка