

19 ДОКУМЕНТИРОВАНИЕ В ФОРМАТЕ XML

Задание 1. В темах: 3 (Процедуры и функции – методы класса), 12 (Делегаты), 16 (Работа с файлами) выполнить XML документирование кода.

Листинг программы:

```
static void Main(string[] args)
{
    Console.Write("a=");
    double a = double.Parse(Console.ReadLine());
    Console.Write("b=");
    double b = double.Parse(Console.ReadLine());
    Console.Write("h=");
    double h = double.Parse(Console.ReadLine());
    for (double i = a; i <= b; i += h)
        Console.WriteLine("f({0:f2})={ 1:f4}", i, f(i));
    Console.WriteLine("Переверззка!-----");
    double y = 0;
    for (double i = a; i <= b; i += h)
    {
        f(i, out y);
        Console.WriteLine("f({0:f2})={ 1:f4}", i, y);
    }
}
/// <summary>
/// Function value calculation
/// </summary>
/// <param name="x">value for function definition</param>
/// <returns>function result</returns>
public static double f(double x)
{
    double y;
    if (x > 1) y = 1 / Math.Pow(1 + x, 2);
    else if (x < 1) y = Math.Pow(Math.Pow(x, 2) - 1, 2);
    else y = 0;
```

					УП 2-40 01 01.31 ТП.2471.22.19			
Изм.	Лист	№ докум	Подп.	Дата				
Разраб.	Мушинский М.С.				ДОКУМЕНТИРОВАНИЕ В ФОРМАТЕ XML		Лит	Лист
Пров.	Толочко П.С.							
Н.контр.							Гродненский ГКТТид	
Утв.								

```

        return y; }
    /// <summary>
    /// Function value calculation, overload
    /// </summary>
    /// <param name="x">value for function definition</param>
    /// <param name="y"></param>
    public static void f(double x, out double y)
    {
        if (x > 1) y = 1 / Math.Pow(1 + x, 2);
        else if (x < 1) y = Math.Pow(Math.Pow(x, 2) - 1, 2);
        else y = 0;
    }
    double radius;
    double length, area, volume;
    Figures fG;
    fG = new Figures();
    CalcFigures cF;
    Console.Write("Введите радиус окружности: ");
    radius = Convert.ToDouble(Console.ReadLine());
    cF = fG.Get_Length;
    length = cF(radius);
    Console.WriteLine($"Длина окружности = {length}");
    cF = fG.Get_Area;
    area = cF(radius);
    Console.WriteLine($"Площадь круга = {area}");
    cF = fG.Get_Volume;
    volume = cF(radius);
    Console.WriteLine($"Объём шара = {volume}");
    /// <summary>
    /// Figures class
    /// </summary>
    class Figures
    {
        /// <summary>
        /// Get length this figure
        /// </summary>
        /// <param name="r">radius</param>
        /// <returns>calculate length</returns>
        public double Get_Length(double r)

```

					УП 2-40 01 01.31 ТП.2471.22.19	Лист
Изм.	Лист	№ докум.	Подп.	Дата		

```

    {
        double length;
        length = Math.PI * 2 * r;
        return length;
    }
    /// <summary>
    /// Get area.
    /// </summary>
    /// <param name="r">radius</param>
    /// <returns>calculate length</returns>
    public double Get_Area(double r)
    {
        double area;
        area = Math.PI * Math.Pow(r, 2);
        return area;
    }
    /// <summary>
    /// Get volume.
    /// </summary>
    /// <param name="r">radius</param>
    /// <returns>volume</returns>
    public double Get_Volume(double r)
    {
        double volume;
        volume = 4.0 / 3.0 * Math.PI * Math.Pow(r, 3);
        return volume;
    }
}
/// <summary>
/// Delegate calculated figures
/// </summary>
/// <param name="r">radius</param>
/// <returns>volume, area, length</returns>
delegate double CalcFigures(double r);

```

Листинг программы:

```
class Program
```

```

{
    /// <summary>

```

Изм.	Лист	№ докум.	Подп.	Дата

```

/// Print menu.
/// </summary>
static void ShowMenu()
{
    Console.WriteLine();
    Console.WriteLine("Меню:");
    Console.WriteLine("0 - Выйти из программы");
    Console.WriteLine("1 - Вывести на экран все компоненты
файла, меньшие заданного числа.");
    Console.WriteLine("2 - Вывести на экран все положительные
компоненты файла");
    Console.WriteLine("3 - Вычислить среднее арифметическое
компонентов файла, стоящих на чётных позициях");
    Console.WriteLine();
}

```