**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 3.1.

static void Main(string[] args)

{

double PentagonSquare = Square(new double[] { 5, 6, 7 }) + Square(new double[] { 5, 6, 7 }) + Square(new double[] { 5, 6, 7 });

Console.WriteLine($"Traiangle square - {Math.Round(Square(new double[] { 5, 6, 7 }), 4)}");

Console.WriteLine($"Pentagon square - {Math.Round(PentagonSquare, 4)}");

}

/// <summary>

/// Calculate square.

/// </summary>

/// <param name="parametrs">sides array</param>

/// <returns>square of any figure</returns>

static double Square(double[] parametrs)

{

double p = 0;

foreach (var item in parametrs)

{

p += item;

}

p /= 2;

return Math.Sqrt(p \* (p - parametrs[0]) \* (p - parametrs[1]) \* (p - parametrs[2]));

}

Листинг программы 12.3.

try

{

Console.Write("Введите строку: ");

string str = Console.ReadLine();

Message msg = ChangeReg;

msg += Substr;

msg += Separate;

msg(str);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

/// <summary>

/// ChangeReg for making Upper case in string

/// </summary>

/// <param name="str">string</param>

void ChangeReg(string str)

{

Console.WriteLine(str.ToUpper());

}

/// <summary>

/// Substr for displaying string from 2 index

/// </summary>

/// <param name="str">string</param>

void Substr(string str)

{

Console.WriteLine(str.Substring(2));

}

/// <summary>

/// Separate for separating string with spaces

/// </summary>

/// <param name="str">string</param>

void Separate(string str)

{

string[] words = str.Split(' ');

foreach (string word in words)

Console.WriteLine(word);

}

/// <summary>

/// Delegat for work with string

/// </summary>

/// <param name="str">some text</param>

delegate void Message(string str);

Листинг программы 16.5.

class Program

{

/// <summary>

/// 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\nна чётных позициях");

Console.WriteLine();

} // ShowMenu()

/// <summary>

/// Display all components of a file less than a given number.

/// </summary>

/// <param name="fromFile">file</param>

static void Task1(BinaryReader fromFile) // Вывести на экран все компоненты файла, меньшие заданного числа

{

double userNum; // Опорное число

double item; // Элемент файла

// Получаем от пользователя опорное число

do // 1

{

Console.Write("Введите опорное число: ");

try

{

userNum = double.Parse(Console.ReadLine());

break;

}

catch (FormatException)

{

Console.WriteLine("Ошибка! Неверный формат данных.");

}

} while (true); // do 1

// Читаем файл и, собственно, выполняем задание

Console.WriteLine("Числа, меньшие заданного вами:");

try // 1

{

fromFile.BaseStream.Seek(0, SeekOrigin.Begin); // Указатель файлового потока - в начало.

while (fromFile.BaseStream.Position < fromFile.BaseStream.Length)

{

item = fromFile.ReadDouble();

if (item < userNum) Console.WriteLine(item);

}

}

catch (IOException A)

{

Console.WriteLine("Внимание! Возникло исключение при чтении файла.\n" + A.Message);

}

// end try 1

} // Task1()

/// <summary>

/// Display all positive components of a file.

/// </summary>

/// <param name="fromFile">file</param>

static void Task2(BinaryReader fromFile) // Вывести на экран все положительные компоненты файла

{

double item; // Элемент файла

// Читаем файл и, собственно, выполняем задание

Console.WriteLine("Положительные компоненты файла:");

try // 1

{

fromFile.BaseStream.Seek(0, SeekOrigin.Begin); // Указатель файлового потока - в начало.

while (fromFile.BaseStream.Position < fromFile.BaseStream.Length)

{

item = fromFile.ReadDouble();

if (item > 0.0) Console.WriteLine(item);

}

}

catch (IOException A)

{

Console.WriteLine("Внимание! Возникло исключение при чтении файла.\n" + A.Message);

}

// end try 1

} // Task2()

/// <summary>

/// Calculate the arithmetic mean of the file components at even positions.

/// </summary>

/// <param name="fromFile">file</param>

static void Task3(BinaryReader fromFile) // Вычислить среднее арифметическое компонентов файла, стоящих на чётных позициях

{

int i = 0;

double summ = 0.0;

double item; // Элемент файла

// Читаем файл и, собственно, выполняем задание

Console.WriteLine("Среднее арифметическое компонентов файла, стоящих на чётных позициях:");

try // 1

{

fromFile.BaseStream.Seek(0, SeekOrigin.Begin); // Указатель файлового потока - в начало.

while (fromFile.BaseStream.Position < fromFile.BaseStream.Length)

{

item = fromFile.ReadDouble();

summ += item; // Суммируем

i++; // Подсчитываем количество нужных нам элементов

fromFile.BaseStream.Position += 8; // Одно число, типа double занимает 8 байт.

}

}

catch (IOException A)

{

Console.WriteLine("Внимание! Возникло исключение при чтении файла.\n" + A.Message);

}

// end try 1

Console.WriteLine("{0:F3}", summ / i);

} // Task3()

static void Main(string[] args)

{

int i = 0; // Переменная цикла

int seqLength = 0; // Длина последовательности

double currNum; // Очередное число

double[] arr; // Ссылка на массив вещественных чисел (пока пустая)

string fileName = "output.dat"; // Имя файла

FileStream fs; // Базовый файловый поток

BinaryWriter dataOut; // Бинарный поток на запись

BinaryReader dataIn; // Бинарный поток на чтение

Console.WriteLine("Ввод последовательности из n вещественных чисел:");

do // 1 Получение длины последовательности

{

Console.Write("Укажите длину последовательности: ");

try

{

seqLength = int.Parse(Console.ReadLine());

if (seqLength <= 0) throw new FormatException();

}

catch (FormatException)

{

Console.WriteLine("Ошибка! Неверный формат данных.");

}

} while (seqLength <= 0); // do 1

arr = new double[seqLength]; // Подключаем ссылку к массиву из заданного количества элементов

// Заполняем массив

do // 2

{

Console.Write("[{0}]: ", i);

try

{

currNum = double.Parse(Console.ReadLine());

arr[i] = currNum;

i++;

}

catch (FormatException)

{

Console.WriteLine("Ошибка! Неверный формат данных.");

}

} while (i < arr.Length); // do 2

/\* Выводим полученный массив для обозрения пользователя.

\* Не очень понимаю зачем это надо. Ведь на консоли и так всё отображается. Но в вашем коде это есть.

\* ...Хотя, если пользователь вводит с ошибками, то экран засоряется сообщениями об ошибках и наглядность теряется.

\* \*/

Console.WriteLine("Полученная последовательность:");

for (i = 0; i < arr.Length; i++)

{

Console.WriteLine("{0}", arr[i]); // Здесь я вывожу не в строку, а в столбец. Мне решилось, что так нагляднее.

} // for

// Пишем всё это дело в файл.

// Открываем файл.

try // 1

{

// Инициируем потоки

fs = new FileStream(fileName, FileMode.Create, FileAccess.Write);

dataOut = new BinaryWriter(fs);

}

catch (IOException A)

{

Console.WriteLine("Внимание! Возникло исключение при открытии файла {0}.", fileName);

Console.WriteLine(A.Message);

Console.WriteLine("Аварийное завершение программы.");

return;

}

// end try 1

// Пишем данные

try // 2

{

for (i = 0; i < arr.Length; i++)

{

dataOut.Write(arr[i]);

}

Console.WriteLine("Файл записан.");

}

catch (IOException B)

{

Console.WriteLine("Внимание! Возникло исключение при записи файла.\n" + B.Message);

}

finally

{

dataOut.Close(); // Закрываем пишущий поток

}

// end try 2

// Выполняем задание

// Открываем файл для чтения

try // 3

{

fs = new FileStream(fileName, FileMode.Open, FileAccess.Read);

dataIn = new BinaryReader(fs);

}

catch (IOException C)

{

Console.WriteLine("Внимание! Возникло исключение при открытии файла {0}.", fileName);

Console.WriteLine(C.Message);

Console.WriteLine("Аварийное завершение программы.");

return;

}

// end try 3

ShowMenu(); // Показываем меню. Его можно вставить внуть цикла do\_3. Тогда оно будет выводиться после выполнения каждого задания.

bool IsExit; // Флаг завершения работы программы

byte itemMenu;

try // 4

{

do // 3

{

IsExit = false;

try // 5

{

Console.WriteLine("Введите номер пункта меню");

itemMenu = byte.Parse(Console.ReadLine());

switch (itemMenu) // 1

{

case 0:

IsExit = true;

break;

case 1:

Task1(dataIn);

break;

case 2:

Task2(dataIn);

break;

case 3:

Task3(dataIn);

break;

default:

Console.WriteLine("Нет такого пункта меню.");

break;

} // switch 1

}

catch (FormatException)

{

Console.WriteLine("Ошибка! Неверный формат при вводе пункта меню.");

}

// end try 5

} while (!IsExit); // do 3

}

finally

{

dataIn.Close();

}

// end try 4

Console.WriteLine("Конец программы");

} // Main()

}