**19 Документирование в формате xml**

Задание 1. 3 (Процедуры и функции – методы класса), 12 (Делегаты), 16

(Работа с файлами) выполнить XML документирование кода.

Листинг программы:

/// <summary>

/// создание Делегата

/// </summary>

/// <param name="a"></param>

/// <returns></returns>

public delegate string Stroka(string a);

class Program

{

static void Main()

{

Console.WriteLine("Введите действие: \n1 Удалить первое слово \n2 Замена всех букв о на ОК \n3 Удаление точек ");

string a = Console.ReadLine();

Console.WriteLine("Введите строку: ");

string x = Console.ReadLine();

switch (a)

{

case "1":

Stroka calc = (c) => c;

string del = x.Remove(0, x.IndexOf(' ') + 1);

Console.WriteLine("Ответ:{0}", del);

break;

case "2":

Stroka calc1 = (c) => c;

string del1 = x.Replace("о", "ОК");

Console.WriteLine("Ответ:{0}", del1);

break;

case "3":

Stroka calc2 = (c) => c;

string zamena = x.Replace(".", "");

Console.WriteLine("Ответ:{0}", zamena);

break;

}

Console.ReadKey();

}

}

}

Листинг программы (12.3)

/// <summary>

/// Запись в файл

/// </summary>

/// <param name="filePath"></param>

/// <param name="minValue"></param>

/// <param name="maxValue"></param>

/// <param name="countNumbers"></param>

private static void FillFile(string filePath, int minValue = -5, int maxValue = 6, int countNumbers = 10)

{

var random = new Random();

using (var writer = new StreamWriter(filePath, append: false, encoding: Encoding.UTF8))

{

while (countNumbers > 0)

{

writer.WriteLine(random.Next(minValue, maxValue));

countNumbers--;

}

}

}

/// <summary>

/// Чтение файла

/// </summary>

/// <param name="filePath"></param>

/// <returns></returns>

private static IEnumerable<int> ReadFile(string filePath)

{

using (var reader = new StreamReader(filePath, Encoding.UTF8))

{

var input = String.Empty;

while ((input = reader.ReadLine()) != null)

{

yield return int.Parse(input);

}

}

}

Листинг программы (16.1)

class Program

{

/// <summary>

/// Вычисление функции

/// </summary>

/// <param name="arg"></param>

/// <returns></returns>

static int func(int arg)

{ return (arg / 10) % 10; }

/// <summary>

/// Метод Main

/// </summary>

/// <param name="args"></param>

static void Main(string[] args)

{

int a = 123456, b = 345678, c = 987694, result;

result = func(a) + func(b) - func(c);

Console.Write($" Сумма вторых справа цифр = { result } ");

Console.ReadKey();

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 9 | Tan2 = 0,205  Sin2 = 0,17  Cos2 = 0,83 |

Анализ результатов:

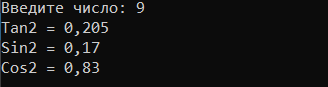


Рисунок 1.1 – Результат работы программы

Задание 2. На основе своей программы создать метод, в котором делегат будет параметром.

Листинг программы:

using System;

namespace zad2

{

class Program

{

public delegate double Calc(double X);

delegate int Operation(int a, int b);

static void Main(string[] args)

{

Console.Write("Введите число: ");

double ch = Convert.ToDouble(Console.ReadLine());

Calc C = new Calc(Calculator.Get\_Area);

C += Calculator.Get\_Sin2;

C += Calculator.Get\_Cos2;

C(ch);

Console.ReadLine();

DoOperation(7, 2, Add);

}

static void DoOperation(int a, int b, Operation operation)

{

int res = a + b;

Console.WriteLine(res);

}

static int Add(int a, int b)

{

return a + b;

}

}

}

public class Calculator

{

public static double Get\_Area(double x)

{

double Tan2;

Tan2 = Math.Pow(Math.Tan(x), 2);

Console.WriteLine($"Tan2 = { Math.Round(Tan2, 3) }");

return Tan2;

}

public static double Get\_Sin2(double x)

{

double Sin2;

Sin2 = Math.Pow(Math.Sin(x), 2);

Console.WriteLine($"Sin2 = { Math.Round(Sin2, 3) }");

return Sin2;

}

public static double Get\_Cos2(double x)

{

double Cos2;

Cos2 = Math.Pow(Math.Cos(x), 2);

Console.WriteLine($"Cos2 = { Math.Round(Cos2, 3) }");

return Cos2;

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 8 | Tan2 = 46,236  Sin2 = 0,979  Cos2 = 0,021 |

Анализ результатов:

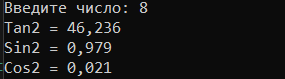


Рисунок 1.2 – Результат работы программы

Задание 3. Для класса MyInfo содержащего поле name (Ваше имя) создать событие выводящее оповещение в случае изменения значения поля name.

Листинг программы:

using System;

namespace задание\_3

{

class Program

{

static void Main(string[] args)

{

Console.Write("Первое имя: ");

string a = Console.ReadLine();

Console.Write("Второе имя: ");

string b = Console.ReadLine();

info Men = new info(a);

Console.Write("Неизменное имя: ");

Console.WriteLine(Men.imya);

Men.inform += vivod;

Men.Put(b);

Console.WriteLine($"Измененное имя: {Men.imya}");

Men.inform -= vivod;

Console.Read();

}

private static void vivod(string message)

{

Console.WriteLine(message);

}

}

}

class info

{

public delegate void muinfo(string message);

public event muinfo inform;

public info(string imya)

{

this.imya = imya;

}

public string imya { get; private set; }

public void Put(string ima)

{

imya = ima;

inform?.Invoke($"Поменять имя на: {ima}");

}

}

Таблица 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Яна  Аня | Неизменное имя: Яна  Поменять имя на: Аня  Измененное имя: Аня |

Анализ результатов:

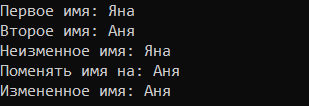


Рисунок 1.3 – Результат работы программы