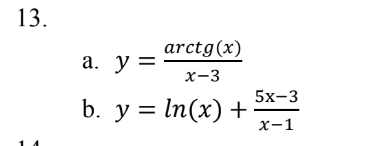
**4 «ОБРАБОТКА ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ».**

Задание 1. Вычислить значение выражений. В каждой разработанной программе должна быть осуществлена обработка исключительных ситуаций. Отдельно обработаны исключения DivideByZeroException, FormatException.



Листинг программы:

using System;

namespace ExpressionCalculator

{

class Program

{

static void Main(string[] args)

{

try

{

Console.WriteLine("Введите значение x:");

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

if (double.IsNaN(x))

{

throw new ArgumentException("Неверный формат числа!");

}

double y1 = Math.Atan(x) / (x - 3);

double y2 = Math.Log(x) + (5 \* x - 3) / (x - 1);

Console.WriteLine($"Значение выражения y1 = arctg(x) / (x - 3) при x = {x} равно {y1}");

Console.WriteLine($"Значение выражения y2 = ln(x) + (5x - 3) / (x - 1) при x = {x} равно {y2}");

}

catch (FormatException)

{

Console.WriteLine("Ошибка: Неверный формат числа!");

}

catch (DivideByZeroException)

{

Console.WriteLine("Ошибка: Деление на ноль!");

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка: {ex.Message}");

}

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2 | Значение выражения y1 = arctg(x) / (x - 3) при x = 2 равно -1,1071487177940904  Значение выражения y2 = ln(x) + (5x - 3) / (x - 1) при x = 2 равно 7,693147180559945 |

Анализ результатов:

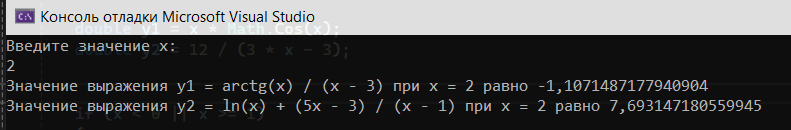
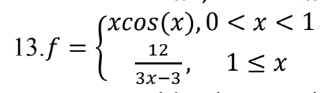


Рисунок 1.1 – Результат работы программы

Задание 2. Для данного вещественного x найти значение следующей функции f, принимающей вещественные значения. В каждой разработанной программе должна быть осуществлена обработка исключительных ситуаций. Отдельно обработаны исключения DivideByZeroException, FormatException. Сгенерируйте пользовательское исключение с помощью оператора throw при проверке условий выхода за диапазон.



Листинг программы:

using System;

namespace ExpressionCalculator

{

class Program

{

static void Main(string[] args)

{

try

{

Console.WriteLine("Введите значение x:");

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

if (double.IsNaN(x))

{

throw new ArgumentException("Неверный формат числа!");

}

double y1 = x \* Math.Cos(x);

double y2 = 12 / (3 \* x - 3);

if (x < 0 || x >= 1)

{

throw new ArgumentOutOfRangeException("Значение x должно быть в диапазоне (0, 1)");

}

Console.WriteLine($"Значение выражения y1 = x \* cos(x) при x = {x} равно {y1}");

Console.WriteLine($"Значение выражения y2 = 12 / (3x - 3) при x = {x} равно {y2}");

}

catch (FormatException)

{

Console.WriteLine("Ошибка: Неверный формат числа!");

}

catch (DivideByZeroException)

{

Console.WriteLine("Ошибка: Деление на ноль!");

}

catch (ArgumentOutOfRangeException ex)

{

Console.WriteLine($"Ошибка: {ex.Message}");

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка: {ex.Message}");

}

}

}

} Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2 | Ошибка: Specified argument was out of the range of valid values. (Parameter 'Значение x должно быть в диапазоне (0, 1)') |

Анализ результатов:

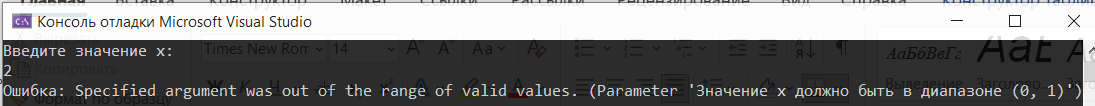


Рисунок 1.2 – Результат работы программы

Задание 3. В каждой разработанной программе должна быть осуществлена

обработка исключительных ситуаций. Отдельно обработаны исключения DivideByZeroException, FormatException. По возможности сгенерируйте пользовательское исключение с помощью оператора.

Описать метод SortDec3(A, B, C), меняющий содержимое переменных A, B, C таким образом, чтобы их значения оказались упорядоченными по убыванию (A, B, C — вещественные параметры, являющиеся одновременно входными и выходными). С помощью этого метода упорядочить по убыванию два данных набора из трех чисел: (A1, B1,C1) и (A2, B2, C2).

Листинг программы:

using System;

namespace SortingApp

{

class Program

{

static void Main(string[] args)

{

try

{

double A1 = 5, B1 = 2, C1 = 8;

double A2 = 3, B2 = 7, C2 = 1;

Console.WriteLine("Original values for Set 1: A1 = {0}, B1 = {1}, C1 = {2}", A1, B1, C1);

Console.WriteLine("Original values for Set 2: A2 = {0}, B2 = {1}, C2 = {2}", A2, B2, C2);

SortDec3(ref A1, ref B1, ref C1);

SortDec3(ref A2, ref B2, ref C2);

Console.WriteLine("Sorted values for Set 1: A1 = {0}, B1 = {1}, C1 = {2}", A1, B1, C1);

Console.WriteLine("Sorted values for Set 2: A2 = {0}, B2 = {1}, C2 = {2}", A2, B2, C2);

}

catch (FormatException)

{

Console.WriteLine("Ошибка: Неверный формат числа!");

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка: {ex.Message}");

}

}

static void SortDec3(ref double A, ref double B, ref double C)

{

if (A < B)

Swap(ref A, ref B);

if (B < C)

Swap(ref B, ref C);

if (A < B)

Swap(ref A, ref B);

}

static void Swap(ref double a, ref double b)

{

double temp = a;

a = b;

b = temp;

}

}

}

Таблица 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Original values for Set 1: A1 = 5, B1 = 2, C1 = 8  Original values for Set 2: A2 = 3, B2 = 7, C2 = 1  Sorted values for Set 1: A1 = 8, B1 = 5, C1 = 2  Sorted values for Set 2: A2 = 7, B2 = 3, C2 = 1 |

Анализ результатов:

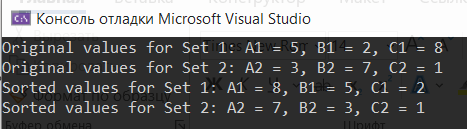


Рисунок 1.3 – Результат работы программы