1. **«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:

namespace task1

{

internal class Program

{

static void Main(string[] args)

{

try

{

Console.Write("Введите х: ");

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

Console.WriteLine($"A = {A(x):f2}");

Console.WriteLine($"B = {B(x):f2}");

}

catch (FormatException ex)

{

Console.WriteLine(ex.Message);

}

catch (DivideByZeroException ex)

{

Console.WriteLine(ex.Message);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

static double A(int x)

{

return (2 \* x) / Math.Sin(x) - 4 \* x; ;

}

static double B(int x)

{

return x - 3 + 1 / Math.Tan(x - 1);

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 6 | -66.95 2.70 |

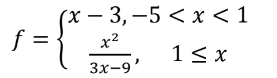
Анализ результатов:



Рисунок 1.1 – Результат работы программы

Источник: собственная разработка.

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:

namespace task2

{

internal class Program

{

static void Main(string[] args)

{

try

{

Console.WriteLine("Введите х:");

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

Console.WriteLine(F(x));

}

catch (FormatException ex)

{

Console.WriteLine(ex.Message);

}

catch (DivideByZeroException ex)

{

Console.WriteLine(ex.Message);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

public static double F(double x)

{

if (x >= 1)

{

return Math.Pow(x, 2) / (3 \* x - 9);

}

else if (x > -5)

{

return x - 3;

}

else

{

throw new Exception("вышло за рамки диапазона");

}

}

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 67 | 23,380208333333332 |

Анализ результатов:

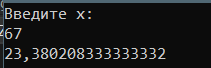


Рисунок 1.2 – Результат работы программы

Источник: собственная разработка.

Задание 3. Описать метод AddLeftDigit(D, K ), добавляющий к целому положительному числу K слева цифру D (D — входной параметр целого типа, лежащий в диапазоне 1–9, K — параметр целого типа, являющийся одновременно входным и выходным). С помощью этого метода последовательно добавить к данному числу K слева данные цифры D1и D2, выводя результат каждого добавления.

Листинг программы:

namespace task\_3

{

internal class Program

{

static void Main(string[] args)

{

try

{

Console.Write("Введите k: ");

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

Console.Write("Введите d: ");

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

AddLeftDigit(d, ref k);

Console.WriteLine(k);

}

catch (FormatException ex)

{

Console.WriteLine(ex.Message);

}

catch (DivideByZeroException ex)

{

Console.WriteLine(ex.Message);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

static void AddLeftDigit(int d, ref int k)

{

for (int i = 0; i < k.ToString().Length; i++)

{

d \*= 10;

}

k += d;

}

}

}

Таблица 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 6 2 | 26 |

Анализ результатов:

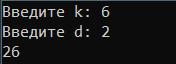


Рисунок 1.3 – Результат работы программы

Источник: собственная разработка.