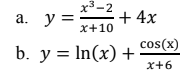
**4 ОБРАБОТКА ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ**

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



Листинг программы:

internal class Program

{

static double Y1(double x)

{

return (Math.Pow(x, 3) - 2 / x + 10) + 4 \* x;

}

static double Y2(double x)

{

return Math.Log(x, Math.E) + Math.Cos(x) / x + 6;

}

static void Main(string[] args)

{

try

{

Console.Write("Input х1: ");

double x1 = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Function result Y1:" + Y1(x1));

}

catch (FormatException ex)

{

Console.WriteLine(ex.Message, ex.StackTrace);

}

try

{

Console.Write("Input х2: ");

double x2 = Convert.ToDouble(Console.ReadLine());

if (Double.IsNaN(Y2(x2)))

{

Console.WriteLine("Not a number.");

}

else

{

Console.WriteLine("Function result Y2:" + Y2(x2));

}

}

catch (DivideByZeroException ex)

{

Console.WriteLine(ex.Message, ex.StackTrace);

}

Console.ReadLine();

}

}

}

Таблица 4.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Х1 = a;  Х2= 3; | Input string was not in a correct format.  Function result Y2: 6,768614789801295 |

Анализ результатов:

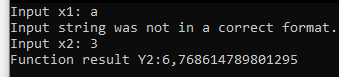


Рисунок 4.1 – Результат работы программы

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:

internal class Program

{

static double F(double x)

{

if ((0 < x) && (x < 1))

{

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

}

else if ((-1 < x) && (x < 3))

{

return -2 / x;

}

else if (1 <= x)

{

return x / x - 1;

}

else

{

throw new IndexOutOfRangeException("OUT OF RANGE!");

}

}

static void Main(string[] args)

{

try

{

Console.Write("Input X: ");

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

Console.WriteLine("Function result:" + F(x));

}

catch (FormatException ex)

{

Console.WriteLine(ex.Message, ex.StackTrace);

}

catch (DivideByZeroException ex)

{

Console.WriteLine(ex.Message, ex.StackTrace);

}

catch (IndexOutOfRangeException dx)

{

Console.WriteLine(dx.Message);

}

Console.ReadLine();

}

}

}

Таблица 4.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| X=1. | Function result:-1 |

Анализ результатов:



Рисунок 4.2 – Результат работы программы

Источник: собственная разработка

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 PowerA3(A, B), вычисляющий третью степень числа A и возвращающую его в переменной B (A — входной, B — выходной параметр; оба параметра являются вещественными). С помощью этого метода найти третьи степени пяти данных чисел.

Листинг программы:

public static double PowerA3(double a, out double b)

{

b = Math.Pow(a, 3);

return b;

}

static void Main(string[] args)

{

try

{

int length = 4;

for (int i = 0; i <= length; i++)

{

Console.Write("Input number: ");

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

double temp;

var b = PowerA3(a, out temp);

Console.WriteLine("Number {0} to the power of 3= {1}", a, b);

}

Console.WriteLine("Program completed, press enter to exit.");

}

catch (FormatException ex)

{

Console.WriteLine(ex.Message, ex.StackTrace);

}

Console.ReadLine();

}

}

}

Таблица 4.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Число = 5  Число = 7  Число = 9 | 125  343  729 |

Анализ результатов:



Рисунок 4.2 – Результат работы программы

Источник: собственная разработка