**4 Обработка исключительных ситуаций**

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

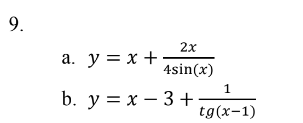


Рисунок 4.1 – Значение выражения

Листинг программы:

namespace zad1

{

internal class Program

{

static void Main(string[] args)

{

try

{

double y1;

double y2;

Console.WriteLine("Введите X: ");

int X = Convert.ToInt32(Console.ReadLine());

y1 = X + (2 \* X) / (4 \* Math.Sin(X));

Console.WriteLine("Первый Y = " + y1);

y2 = X - 3 + (1 / (Math.Tan(X - 1)));

Console.WriteLine("Второй Y = " + y2);

}

catch (DivideByZeroException)

{

Console.WriteLine("Деление на ноль запрещено! ");

}

catch (FormatException)

{

Console.WriteLine("Не введено значение! ");

}

finally

{

Console.WriteLine("Завершение программы! ");

}

Console.ReadLine();

}

}

}

Таблица 4.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| X = 2 | Первый Y = 3,099  Второй Y = -0,35 |

Анализ результатов:

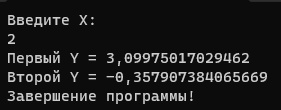


Рисунок 4.2 – Результат работы программы

Источник: собственная разработка

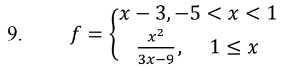
Задание 2. Для данного вещественного x найти значение следующей функции f, принимающей вещественные значения. В каждой разработанной программе должна быть осуществлена обработка исключительных ситуаций. Отдельно обработаны исключения DivideByZeroException, FormatException. Сгенерируйте пользовательское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

Рисунок 4.3 – Функция f

Источник: собственная разработка

Листинг программы:

namespace zad2

{

internal class Program

{

static void Main(string[] args)

{

try

{

double f;

Console.WriteLine("Введите вещественный X: ");

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

if (X < 1 && X > -5)

{

f = X - 3;

Console.WriteLine(f);

}

else if (X >= 1)

{

f = (Math.Pow(X,2)) / ((3 \* X) - 9);

Console.WriteLine(f);

}

else throw new Exception("Ошибка: выход из диапозона допустимых значений X");

}

catch (DivideByZeroException DfX)

{

Console.WriteLine("Деление на ноль запрещено! " + DfX.Message);

}

catch (FormatException fX)

{

Console.WriteLine("Не введено значение X! " + fX.Message);

}

finally

{

Console.WriteLine("Программа завершена! ");

}

Console.ReadLine();

}

}

}

Таблица 4.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 3,4 | 9,6 |

Анализ результатов:

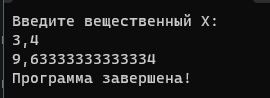


Рисунок 4.4– Результат работы программы

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.

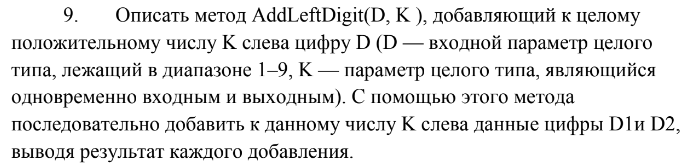


Рисунок 4.5 – Условие задания

Источник: собственная разработка

Листинг программы:

namespace zad3

{

internal class Program

{

static void Main(string[] args)

{

try

{

int K = 2345;

Console.WriteLine("Введите целочисленное число D1: ");

int D1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите целочисленное число D2:");

int D2 = Convert.ToInt32(Console.ReadLine());

AddLeftDigit(D1, ref K);

Console.WriteLine(K);

AddLeftDigit(D2, ref K);

Console.WriteLine(K);

Console.ReadLine();

}

catch (DivideByZeroException dv)

{

Console.WriteLine(dv.Message + "Деление на ноль! ");

}

catch (FormatException fe)

{

Console.WriteLine(fe.Message + "Не верно введен формат! ");

}

Console.ReadLine();

}

public static void AddLeftDigit(int D, ref int K)

{

if (D <= 0 || D > 9 || K <= 0)

throw new ArgumentException("Неверное значение параметров");

int n = K;

while (n != 0)

{

n /= 10;

D \*= 10;

}

K += D;

}

}

}

Таблица 4.3 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| D1 = 2  D2 = 4 | 422345 |

Анализ результатов:

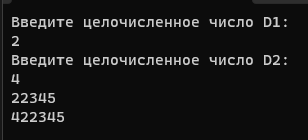


Рисунок 4.6 – Результат работы программы

Источник: собственная разработка