**4 Обработка исключительных ситуаций**

Задание 1. Вычислить значение выражений. В каждой разработанной программе должна быть осуществлена обработка исключительных ситуаций. Отдельно обработаны исключения DivideByZeroException, FormatException. Выражения для задания представлены на рисунке 4.1.

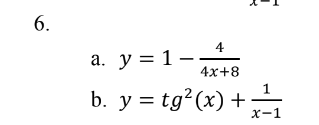


Рисунок 4.1 – Выражения для вычисления

Источник: задание 1

Листинг программы для выражения A:

try {

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

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

if (x == -2)

{throw new DivideByZeroException("Знаменатель не может быть равен 0"); }

double y = 1 - (4 / (4 \* x + 8));

Console.WriteLine("Значение выражения a. y = {0}", y); }

catch (DivideByZeroException ex) {Console.WriteLine(ex.Message);}

catch (FormatException ex) { Console.WriteLine("Ошибка ввода: " + ex.Message); }

catch (Exception ex) {Console.WriteLine("Произошла ошибка: " + ex.Message); }

Таблица 4.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| ff | Ошибка ввода: Input string was not in a correct format. |

Источник: собственная разработка

Анализ результатов:

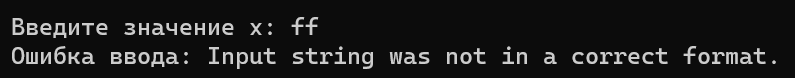


Рисунок 4.2 – Результат работы программы

Источник: собственная разработка

Листинг программы для выражения B:

try {

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

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

if (x == 1) {

throw new DivideByZeroException("Знаменатель не может быть равен 0");}

double y = Math.Pow(Math.Tan(x), 2) + (1 / (x - 1));

Console.WriteLine("Значение выражения b. y = {0}", y); }

catch (DivideByZeroException ex){Console.WriteLine(ex.Message);}

catch (FormatException ex){Console.WriteLine("Ошибка ввода: " + ex.Message);}

catch (Exception ex){Console.WriteLine("Произошла ошибка: " + ex.Message);}

Таблица 4.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1 | Знаменатель не может быть равен 0 |

Источник: собственная разработка

Анализ результатов:

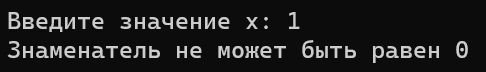


Рисунок 4.3 – Результат работы программы

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.

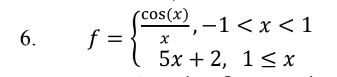


Рисунок 4.4 – Функция f

Источник: задание 2

Листинг программы:

Таблица 3.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| -5; 5; 1 | f(-5)=0; f(-4)=0; f(-3)=0; f(-2)=4; f(-1)=1; f(0)=0; f(1)=0; f(2)=0; f(3)=0; f(4)=0; f(5)=0; |

Источник: собственная разработка

Анализ результатов:

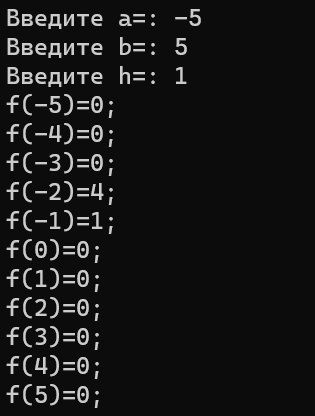


Рисунок 3.3 – Результат работы программы

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 DigitCountSum(K, C, S), находящую количество C цифр целого положительного числа K, а также их сумму S (K — входной, C и S — выходные параметры целого типа). С помощью этого метода найти количество и сумму цифр для каждого из пяти данных целых чисел.

Листинг программы:

Таблица 3.3 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Матрица a:  1 2  3 4  Матрица b:  5 6  7 8  Сумма элементов главной диагонали в матрице a меньше, чем в матрице b |

Источник: собственная разработка

Анализ результатов:

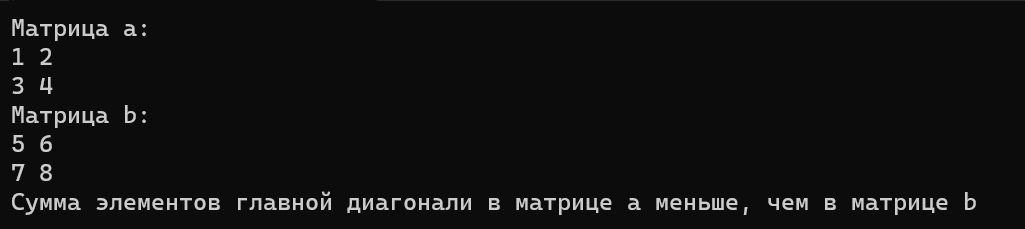


Рисунок 3.4 – Результат работы программы

Источник: собственная разработка