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:

try

{

Console.Write("Введите x: ");

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

double y = (x + 4) / (x + 8);

double z = Math.Pow(Math.Cos(x), 3) / (x - 1);

Console.WriteLine($"y = {Math.Round(y, 2)}");

Console.WriteLine($"y = {Math.Round(z, 2)}: ");

}

catch (DivideByZeroException e)

{

Console.WriteLine($"Exception: {e}");

}

catch (FormatException e)

{

Console.WriteLine($"Exception: {e}");

}

Таблица 4.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4 | 0, -0.09 |

Анализ результатов:

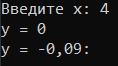


Рисунок 4.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Проверить истинность высказывания: «Цифры данного

трехзначного числа образуют возрастающую или убывающую

последовательность».

Листинг программы:

try

{

Console.Write("Введите x: ");

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

double y = 0;

if (x > 1)

{

y = Math.Pow(x, 2);

}

else if (x > -4 && x <= 1)

{

y = (3 \* x + 2) / (4 \* x + 4);

else

{

throw new Exception("IndexOutOfRange");

}

Console.WriteLine($"y = {Math.Round(y, 2)}");

}

catch (DivideByZeroException e)

{

Console.WriteLine($"Exception: {e}");

}

catch (FormatException e)

{

Console.WriteLine($"Exception: {e}");

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

Таблица 4.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| -5 | IndexOutOfRange |

Анализ результатов:

C:\Users\1234321\Downloads\govno7.png

Рисунок 4.2 – Результат работы программы

Источник: собственная разработка

Задание 3. 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:

try

{

Console.Write($"Введите A: ");

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

Console.Write($"Введите B: ");

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

Console.Write($"Введите C: ");

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

Console.Write($"Введите A1: ");

double a1 = Convert.ToDouble(Console.ReadLine());

Console.Write($"Введите B1: ");

double b1 = Convert.ToDouble(Console.ReadLine());

Console.Write($"Введите C1: ");

double c1 = Convert.ToDouble(Console.ReadLine());

SortInc3(a, b, c, out a, out b, out c);

SortInc3(a1, b1, c1, out a1, out b1, out c1);

Console.WriteLine($"A = {Math.Round(a, 2)}, B = {Math.Round(b, 2)}, C = {Math.Round(c, 2)}");

Console.WriteLine($"A1 = {Math.Round(a1, 2)}, B1 = {Math.Round(b1, 2)}, C1 = {Math.Round(c1, 2)}");

}

catch (DivideByZeroException e)

{

Console.WriteLine($"Exception: {e}");

}

catch (FormatException e)

{

Console.WriteLine($"Exception: {e}");

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

void SortInc3(double a, double b, double c, out double resA, out double resB, out double resC)

{

if (a > b)

{

double temp = a;

a = b;

b = temp;

}

if (a > c)

{

double temp = a;

a = c;

c = temp;

}

if (b > c)

{

double temp = b;

b = c;

c = temp;

}

resA = a;

resB = b;

resC = c;

}

Таблица 4.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| A = 1 B = 3 C = 2  A1 = 4  B1 = 3 C1 = 3 | A = 1  B = 2  C = 3  A1 = 1 B1 = 3 C1 = 4 |

Анализ результатов:

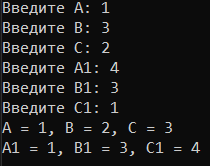


Рисунок 4.3 – Результат работы программы

Источник: собственная разработка