**4 ОБРАБОТКА ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ**

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

Формулы, реализованные в задании представлены на рисунке 4.1.

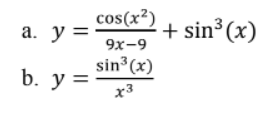


Рисунок 4.1 – Формулы

Источник: собственная разработка

Листинг программы:

try

{

Console.Write("Введите х = ");

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

double y1 = ((Math.Cos(Math.Pow(x, 2))) / (9 \* x - 9)) + Math.Pow(Math.Sin(x), 3);

//if (double.IsInfinity(y1))

//{

// Console.WriteLine("Inf");

//}

double y2 = (Math.Pow(Math.Sin(x), 3)) / (Math.Pow(x, 3));

Console.WriteLine($"a = {y1}");

Console.WriteLine($"b = {y2}");

}

catch (DivideByZeroException ex)

{

Console.WriteLine(ex.Message);

}

catch (FormatException ex)

{

Console.WriteLine(ex.Message);

}

Таблица 4.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| х = 4 | a = -0,4689275116224442  b = -0,006772791280950584 |

Анализ результатов:

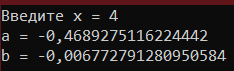


Рисунок 4.2 – Результаты работы программы

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. Задание, реализованное в задании представлено на рисунке 4.3.

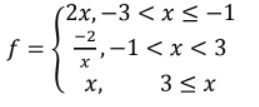


Рисунок 4.3 – Задание для реализации

Источник: собственная разработка

Листинг программы:

try

{

Console.Write("Введите х = ");

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

if (x > -3 && x <= -1)

{

var f = 2 \* x - 3;

Console.WriteLine($"Result = {f}");

}

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

{

var f = -2 / x;

Console.WriteLine($"Result = {f}");

}

else if (x >= 3)

{

var f = x;

Console.WriteLine($"Result = {f}");

}

else

{

throw new Exception("Выход за пределы диапозона");

}

}

catch (DivideByZeroException ex)

{

Console.WriteLine(ex.Message);

}

catch (FormatException ex)

{

Console.WriteLine(ex.Message);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Таблица 4.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| х = 10 | Result = 10 |

Анализ результатов:



Рисунок 4.4 – Результаты работы программы

Источник: собственная разработка