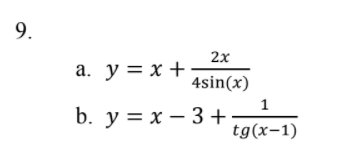
**Обработка исключительных ситуаций.**

**Задание №1.** Вычислить значение выражений. В каждой разработанной программе должна быть осуществлена обработка исключительных ситуаций. Отдельно обработаны исключения DivideByZeroException, FormatException.



Листинг программы:

static void Main(string[] args)

{

CultureInfo.CurrentCulture = new CultureInfo("ru-RU");

Thread.CurrentThread.CurrentCulture = new CultureInfo("ru-RU", true);

Thread.CurrentThread.CurrentUICulture = new CultureInfo("ru-RU", true);

try

{

Write("Введите X: ");

int x = ToInt32(ReadLine());

WriteCalculated(MakeFirstCalculate(x));

}

catch (FormatException ex)

{

WriteLine(ex.Message);

}

catch (DivideByZeroException ex)

{

WriteLine("Ошибка: деление на 0!");

}

try

{

Write("Введите X: ");

int x = ToInt32(ReadLine());

WriteCalculated(MakeSecondCalculate(x));

}

catch (FormatException ex)

{

WriteLine(ex.Message);

}

catch (DivideByZeroException ex)

{

WriteLine("Ошибка: деление на 0!");

}

ReadKey();

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| X = 5  X = 6 | 3,08  2,70 |

Анализ результатов:

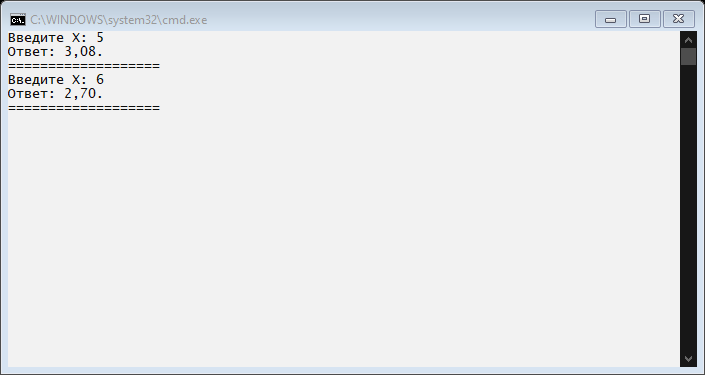
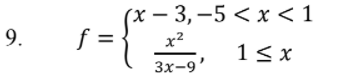


Рисунок 1.1 – Результат работы программы

**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:

static void Main(string[] args)

{

try

{

Write("Введите X: ");

int x = ToInt32(ReadLine());

WriteCalculated(MakeCalculate(x));

}

catch(IndexOutOfRangeException ex)

{

WriteLine(ex.Message);

}

catch (FormatException ex)

{

WriteLine(ex.Message);

}

catch (DivideByZeroException ex)

{

WriteLine("Ошибка: деление на 0!");

}

ReadKey();

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| X = 12 | 567,00 |

Анализ результатов:

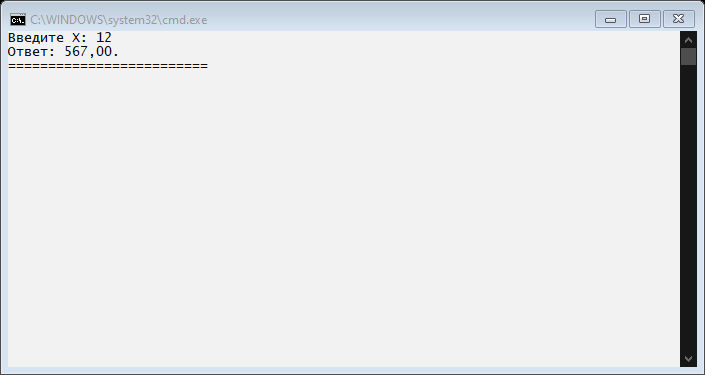


Рисунок 1.2 – Результат работы программы

**Задание № 3.** Описать метод AddLeftDigit(D, K ), добавляющий к целому положительному числу K слева цифру D (D — входной параметр целого типа, лежащий в диапазоне 1–9, K — параметр целого типа, являющийся одновременно входным и выходным). С помощью этого метода последовательно добавить к данному числу K слева данные цифры D1и D2, выводя результат каждого добавления.

Листинг программы:

static void Main(string[] args)

{

try

{

Write("Введите целое число К: ");

int k = ToInt32(ReadLine());

for (int i = 1; i < 3; i++)

{

Write($"Введите цифру D{i}: ");

int d = CheckRange(ToInt32(ReadLine()));

k = AddLeftDigit(d, k);

WriteLine($"\nK{i}({k})+ D{i}({d}) = {k}\n" +

$"=========================\n");

}

}

catch (OverflowException ex)

{

WriteLine("Число слишком большое для INT32.");

}

catch (IndexOutOfRangeException ex)

{

WriteLine(ex.Message);

}

catch (FormatException ex)

{

WriteLine(ex.Message);

}

ReadKey();

}

Таблица 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| К: 652  Д1: 9  Д2: 8 | 9652  89652 |

Анализ результатов:

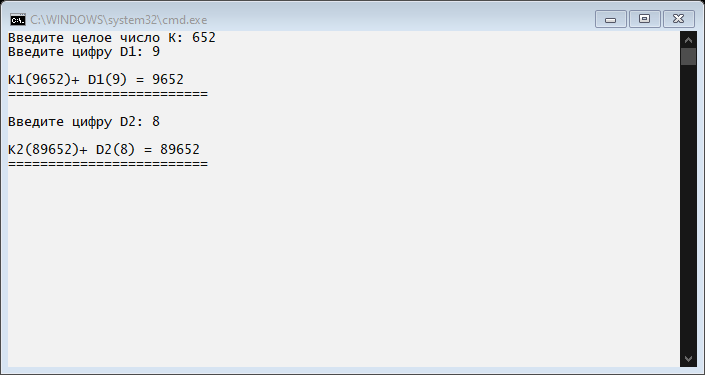


Рисунок 1.3 – Результат работы программы