

4 Обработка исключительных ситуаций

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

a. $y = 1 - \frac{4}{4x+8}$

b. $y = \operatorname{tg}^2(x) + \frac{1}{x-1}$

Листинг программы:

```
namespace MyNameSpace
```

```
{
```

```
    class Program
```

```
    {
```

```
        static double F(double x)
```

```
        {
```

```
            if (x == -2)
```

```
                throw new DivideByZeroException();
```

```
            return 1 - (5 * x - 7) / (4 * x + 8);
```

```
        }
```

```
        static double FF(float x)
```

```
        {
```

```
            if (x == 1)
```

```
                throw new DivideByZeroException();
```

```
            return Math.Pow(Math.Tan(x), 2) + (1 / (x - 1));
```

```
        }
```

```
        static void Main(string[] args)
```

```
        {
```

```
            try
```

```
            {
```

```
                Console.Write("Введите значение x1: ");
```

```
                float x1 = float.Parse((Console.ReadLine()));
```

```
                Console.WriteLine("Значение функции: " + Math.Round(F(x1), 2));
```

```
                Console.Write("Введите значение x2: ");
```

```
                float x2 = float.Parse((Console.ReadLine()));
```

					УП 2-40 01 01.37ТП.227.23.04			
Изм.	Лист	№ док	Подпись	Дата				
Разраб.		Сорокина Е.А.			Обработка исключительных ситуаций		Лит	Лист
Проверил.		Новик А.И.						20
							Гродненский ГКТТид	
Н.контр.								
Утвердил.								

```

        Console.WriteLine("Значение функции: " + Math.Round(FF(x2), 2));
    }
    catch(DivideByZeroException)
    {
        Console.WriteLine("Деление на 0 запрещено!");
    }
    catch (FormatException)
    {
        Console.WriteLine("Некорректное значение!");
    }
}
}
}

```

Таблица 4.1 – Входные и выходные данные

Входные данные	Выходные данные
4	0,46
fgg	Некорректное значение!

Анализ результатов:

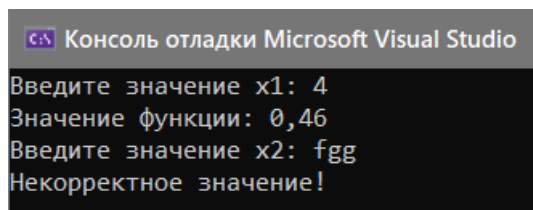


Рисунок 4.1 – Результат работы программы

Задание 2. Описать метод InvertDigits(K), меняющий порядок следования цифр целого положительного числа K на обратный (K — параметр целого типа, являющийся одновременно входным и выходным). С помощью этого метода поменять порядок следования цифр на обратный для каждого из пяти данных целых чисел.

Листинг программы:

```

namespace MyNameSpace
{
    class Not5 : Exception
    {
        public Not5(string message)
            : base(message) { }
    }
}

```

```

class Program
{
    private static int InvertDigits(int k)
    {
        int result = 0;
        while (k != 0)
        {
            result = result * 10 + k % 10;
            k = k / 10;
        }
        return result;
    }

    static void Main(string[] args)
    {
        try
        {
            Console.Write("Введите 5-ти значное число: ");
            int x = Convert.ToInt32(Console.ReadLine());
            string a = x.ToString();

            if (a.Length == 5)
            {
                Console.WriteLine("Переворот числа: " + InvertDigits(x));
            }
            else
            {
                throw new Not5("");
            }
        }
        catch (Not5)
        {
            Console.WriteLine("Вы ввели не 5 чисел!");
        }
        catch (FormatException)
        {
            Console.WriteLine("Некорректное значение!");
        }
    }
}

```

Таблица 4.2 – Входные и выходные данные

Входные данные	Выходные данные
5	0,62
-8	Числа с минусом в диапазон не входят!

Анализ результатов:

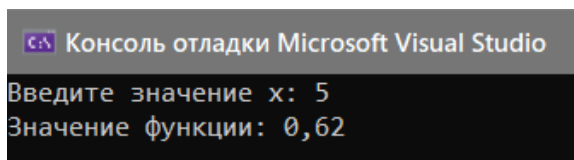


Рисунок 4.2 – Результат работы программы

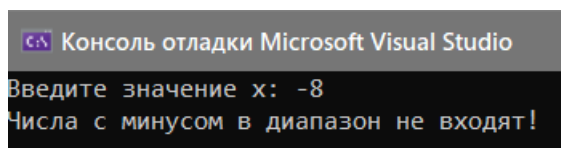


Рисунок 4.3 – Результат исключения