

Лабораторная работа 05. Обработка исключительных ситуаций

Справочный материал:

<https://metanit.com/sharp/tutorial/2.14.php>

<https://metanit.com/sharp/tutorial/2.28.php>

<https://metanit.com/sharp/tutorial/2.29.php>

<https://metanit.com/sharp/tutorial/2.31.php>

<https://metanit.com/sharp/tutorial/3.17.php>

<https://metanit.com/sharp/tutorial/3.17.php>

Задания:

1) Создать статический метод, возвращающий

$$f(x) = \begin{cases} x + \sin^2\left(\frac{1}{x-a} + 4\right), & x < 0 \\ \frac{ax}{\sqrt{a^2 - x^2}}, & x \geq 0 \end{cases}$$

причем величина x передается через параметр метода,

а значение величины a вычисляется внутри метода как случайное целое число из диапазона [-30, 30].

Протестировать работу метода, при получении не числовых значений (NaN, Infinity) программа должна выводить подходящие текстовые сообщения. Использовать try-catch.

В тестах для значений x от -20 до 20 вычислять $f(x)$

а) прекращать выполнение при первом же неудачном вычислении $f(x)$;

Если все $f(x)$ рассчитаны, то вывести их сумму

б) при неудачном вычислении $f(x)$ игнорировать это значение и продолжать вычисление остальных.

Вызвести количество найденных $f(x)$ и их среднее арифметическое

2) В программе требуется вводить четные отрицательные числа.

Создать статический метод для ввода одного такого числа (с проверкой четности и отрицательности).

В диалоге с пользователем обеспечить не более 3 попыток ввести это число. Использовать try-catch с фильтрами и рекурсивный вызов этого метода.

В программе вводить числа до тех пор, пока это удается. Вызвести количество правильно введенных чисел и сумму.

3) Создать статические методы, для

а) ввода с консоли части элементов одномерного вещественного массива (в диапазоне индексов от $k1$ до $k2$ включительно; индексы передаются через параметры метода),
остальные элементы массива не меняются

б) заполнения случайными числами части одномерного вещественного массива (в диапазоне индексов от $k1$ до $k2$ включительно; индексы передаются через параметры),
остальные элементы массива не меняются

Протестировать методы на подходящих примерах

4) Создать собственные классы исключений для обработки ошибок, возникающих в методах из пп 1) и 2).

Создать реализацию методов из пп 1) и 2) с использованием этих классов. Продемонстрировать их применение на подходящих примерах.