

## № 7 Обработка исключений

### Задание

Дополнить предыдущую лабораторную работу № 6.

Создать иерархию классов исключений (собственных) – 3 типа и более. Сделать наследование пользовательских типов исключений от стандартных классов .Net (например Exception).

Сгенерировать и обработать как минимум пять различных исключительных ситуаций на основе своих и стандартных исключений. Например, не позволять при инициализации объектов передавать неверные данные, обрабатывать ошибки при работе с памятью и ошибки работы с файлами, деление на ноль, неверный индекс, нулевой указатель и т. д.

В конце поставить универсальный обработчик catch.

Обработку исключений вынести в main. При обработке выводить специфическую информацию о месте, диагностику и причине исключения. Последним должен быть блок, который отлавливает все исключения (finally). Добавьте код в одной из функций макрос Assert. Объясните что он проверяет, как будет выполняться программа в случае не выполнения условия. Объясните назначение Assert.

Пример:

```
int[] aa= null;  
Debug.Assert(aa !=null, "Values array cannot be null");
```

Не забудьте подключить `using System.Diagnostics;`

Далее приведен перечень заданий.

### Вопросы

1. Расскажите как генерируется исключение.
2. Расскажите методику обработки исключений.
3. Какое ключевое слово служит для обозначения блока кода, в котором можно генерировать исключение?
4. Какие ключевые слова используются для обработки и генерации исключений? Расскажите об механизме обработке исключения?
5. Что будет, если в программе нет предложения catch, способного обработать исключение?
6. Что такое фильтры исключения? Приведите пример
7. Могут ли исключения быть вложенными?
8. Какой синтаксис нужно использовать в C# для отлова любого возможного исключения?
9. Чем следует руководствоваться при размещении обработчиков исключения?
10. Что будет выведено на консоль в результате выполнения фрагмента листинга?

```
static void Main(string[] args)  
{  
    string[] str = new string[5];  
    try  
    {  
        str[4] = "anything";  
        Console.WriteLine("It's OK");  
    }  
    catch (IndexOutOfRangeException e)  
    {
```

```
        Console.WriteLine("IndexOutOfRangeException");
    }
    catch (Exception e)
    {
        Console.WriteLine("Exception");
    }
}
```

11. Как повторно сгенерировать то же самое исключение в блоке обработчике catch?
12. .Какие методы содержатся в классе Exception? Где и как их можно использовать?