

C# Essential

Обработка исключений

C# Essential

Автор курса



Александр Шевчук
MCT



MCID: 9230440

C# Essential

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)

Доступ можно получить через руководство вашего учебного центра



Проверьте как Вы усвоили данный материал на [TestProvider.com](http://testprovider.com)

Обработка исключений

Exceptions

Исключения

Исключения – неожиданные события, прерывающие нормальную работу кода и возникающие во время исполнения программы.

try, catch

Обработка исключений

Обработка исключительных ситуаций в C# организуется с помощью трех ключевых слов: **try**, **catch** и **finally**

```
try
{
    // Попытка деления на ноль.
    a = a / (2 - n);
    Console.WriteLine("a = {0}", a);
}

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

Конструкция **try-catch** позволяет выделить небезопасные участки кода, в которых могут возникать исключения, и предоставляет код для их обработки, если они происходят.



Если в конструкции **try** возникает исключение, то выполняется конструкция **catch**.

Исключения

Обработка исключений

```
static void Main()
{
    Exception ex = new Exception("Мое Исключение");

    try
    {
        throw ex;
    }
    catch (Exception e)
    {
        Console.WriteLine("Обработка исключения.");
        Console.WriteLine(e.Message);
    }
}
```

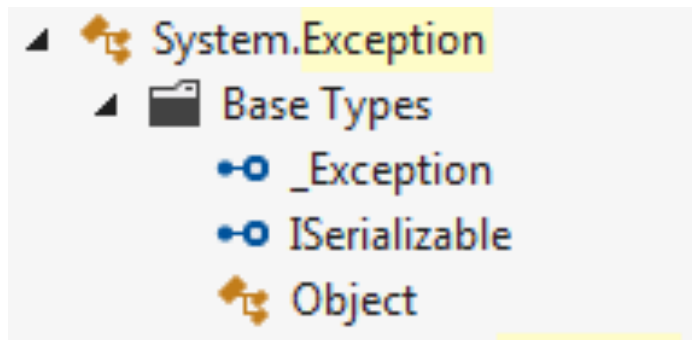


Ключевое слово `throw` — генерирует исключение

Исключения

Обработка исключений

Для создания пользовательского исключения, требуется наследование от `System.Exception`.



```
class UserException : Exception
{
    public void Method()
    {
        Console.WriteLine("Exception");
    }
}
```


Исключения

Обработка исключений

Блок `finally` используется зачастую для установки настроек в первоначальное значение и выполняется всегда. Является не обязательным.

```
try
{
    throw new Exception("Мое Исключение");
}
catch (Exception e)
{
    Console.WriteLine("Обработка исключения.");
}
finally
{
    Console.WriteLine("Выведется всегда")
}
```



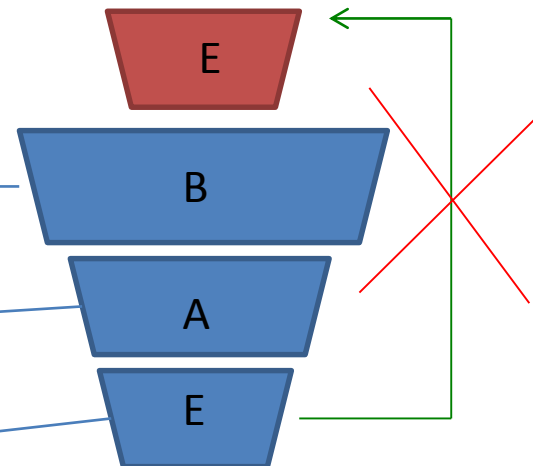
`finally` – не срабатывает в случае возникновения исключения `StackOverflowException`.

`finally` – не срабатывает, если не завершается работа блока `catch`.

Исключения

Обработка нескольких конструкций catch

```
try
{
    //throw new Exception("Exception");
    throw new MyExceptionA("MyExceptionA");
    //throw new MyExceptionB("MyExceptionB");
}
catch (MyExceptionB e)
{
    Console.WriteLine(e.Message);
}
catch (MyExceptionA e)
{
    Console.WriteLine(e.Message);
}
catch (Exception e)
{
    Console.WriteLine(e.Message);
}
```



Несколько конструкций `catch` используется для отлавливания нескольких видов ошибок.

Смотрите наши уроки в видео формате

ITVDN.com



Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://itvdn.com) для закрепления пройденного материала.

Все курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics



Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на TestProvider.com

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



C# Essential

Q&A

Информационный видеосервис для разработчиков программного обеспечения

