

Исключения

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

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`

Обработка исключений

Ключевое слово `throw` – генерирует исключение

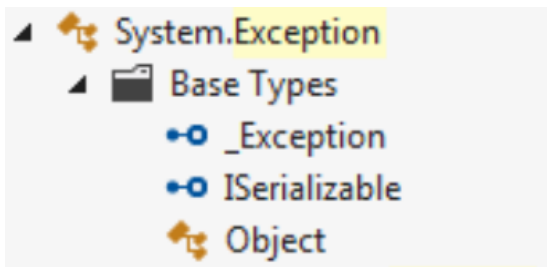
```

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

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

```

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



```

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

```

Обработка исключений

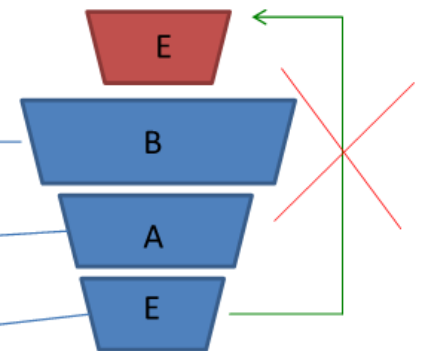
Блок `finally` используется зачастую для установки настроек в первоначальное значение и выполняется всегда. Является не обязательным

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

`Finally` – не срабатывает, если не завершается работа блока `catch`

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