

Для генерации исключительной ситуации используется оператор **throw**:

```
throw <объект класса исключительной ситуации>;
```

Объект, указанный после **throw**, должен быть объектом класса исключительной ситуации. В C# классами исключительных ситуаций являются класс **System.Exception** и все его наследники. В языках, отличных от C#, можно (хотя и не рекомендуется) генерировать исключения, не являющиеся производными от **Exception**. В таком случае CLR автоматически поместит объект исключения в класс-оболочку **RuntimeWrappedException**, который наследуется от **Exception**.

Класс **Exception** – это базовый класс для представления исключительных ситуаций. Основными элементами этого класса являются:

- свойство только для чтения **Message**, содержащее строку с описанием ошибки;
- перегруженный конструктор с одним параметром-строкой, записываемым в свойство **Message**;
- строковое свойство **StackTrace**, описывающее содержимое стека вызова, в которой первым отображается самый последний вызов метода.
- свойство **InnerException** – объект **Exception**, описывающий ошибку, вызывающую текущее исключение.
- коллекция-словарь **Data** с дополнительной информацией об ошибке.

В пространстве имён **System** содержится несколько классов для описания наиболее распространённых исключений:

- **ArgumentException** - генерируется, когда методу передаётся недопустимый аргумент.
- **ArgumentNullException** (наследник **ArgumentException**) - генерируется, когда методу передаётся аргумент, равный **null**.
- **ArgumentOutOfRangeException** (наследник **ArgumentException**) - генерируется, когда методу передаётся аргумент, выходящий за допустимый диапазон.
- **InvalidOperationException** - сигнализирует о том, что состояние объекта препятствует выполнению метода (пример: запись в файл, который открыт только для чтения).
- **NotImplementedException** - сигнализирует о том, что функциональная возможность не реализована.

Разработчик может создать собственный класс для представления информации об исключительной ситуации. Единственным условием для этого класса в C# является прямое или косвенное наследование от класса **Exception**.

Опишем возможности по обработке исключительных ситуаций. Для перехвата исключительных ситуаций служит блок **try – catch – finally**. Синтаксис блока следующий:

```
try
{
    [операторы, способные вызвать исключительную ситуацию]
}
[один или несколько блоков catch]
catch [(<тип ИС> [<идентификатор объекта ИС>])]
{
    <операторы обработки исключительной ситуации>
}
Finally { <операторы из секции завершения> }
```

Если один из операторов, расположенных в блоке **try**, вызвал исключительную ситуацию, управление немедленно передается на блоки **catch**. Здесь *<идентификатор объекта ИС>* – это некая временная переменная, которая может использоваться для извлечения информации из объекта исключительной ситуации. Отдельно описывать эту переменную нет необходимости. Если указать в блоке **catch** оператор **throw**, это приведёт к тому, что обрабатываемая исключительная ситуация будет сгенерирована повторно.