

Работа метода **.Parse()**

Для начала ссылка на нашу полученную строку помещается в стек.

(для примера: IL_0001: ldstr "hello")

Далее вызывается метод **.Parse()**, который в качестве аргумента извлекает данные из стека.

(для примера: IL_0006: call int32 [System.Runtime]System.Int32::Parse(string))

В случае успешного завершения работы метода возвращаемое значение отправляется вызывающей стороне и помещается в стек.

Оттуда это значение помещается в локальную переменную.

(для примера: IL_000b: stloc.0 // num1)

В случае неудачного преобразования мы получаем System.FormatException и приложение заканчивает работу.

Работа метода **.TryParse()**

Для начала ссылка на нашу полученную строку помещается в стек.

Далее вызывается метод **.TryParse()**, который в качестве аргумента извлекает данные из стека.

В случае успешного завершения работы метод возвращает преобразованное значение в out параметре.

Затем оно отправляется вызывающей стороне и помещается в стек.

Оттуда это значение помещается в локальную переменную.

В случае неудачного преобразования метод возвращает значение false, что говорит о неудачном преобразовании аргумента.

Затем мы получаем возможность обработать ошибку.

Заключение:

Метод **.Parse()** можно использовать ТОЛЬКО в случае, если мы уверены в корректности обрабатываемых данных.

В любом другом случае лучше использовать метод **.TryParse()**, так как использование этого метода даёт нам уверенность, что наше приложение не «крашнется» из-за некорректных обрабатываемых данных, а также даёт нам возможность отреагировать на ошибку (например, использовать какие-то данные по умолчанию или просить пользователя ввести данные до тех пор, пока не будут введены корректные).