

# Пространство имен RegularExpressions

Таблица 10.6 – Поля перечисления RegexOptions

<i>Поле</i>	<i>Значение</i>
<b>Compiled</b>	Указывает на то, скомпилировано-ли регулярное выражение в сборку (Assembly). Это ускоряет процесс применения регулярного выражения, но увеличивает время загрузки программы.
<b>CultureInvariant</b>	Указывает на то, следует-ли пропускать культурные различия в тексте (например, в Англии цвет — colour, а в Америке — color).
<b>ECMAScript</b>	Включает есмаскрипт-совместимость регулярных выражений. Обычно используется в связке с опцией ignorecase.
<b>IgnoreCase</b>	Указывает на то, что в обрабатываемом тексте не следует учитывать регистр букв.
<b>IgnorePatternWhitespace</b>	Удаляет из тела регулярного выражения неэкранированный пробел и дает возможность использовать комментарии после символа #.
<b>Multiline</b>	Многострочный режим. Позволяет использовать директивы ^ и \$ как символы начала и конца каждой линии.
<b>None</b>	Пустая опция.
<b>RightToLeft</b>	Указывает на то, что поиск следует производить справа налево, а не слева направо.
<b>Singleline</b>	Однострочный режим. Изменяет значение точки (.) Так, что она может представлять любой символ (включая символ переноса строки \n).

Таблица 10.7 – Способы передачи данных в конструктор метода `IsMatch()`

<i>Method</i>	<i>Описание</i>
<code>IsMatch(string input)</code>	<code>input</code> – обрабатываемая строка. Применяется регулярное выражение, ранее переданное в конструктор класса <code>Regex</code> .
<code>IsMatch(string input, int start)</code>	<code>input</code> – обрабатываемая строка, а <code>start</code> – позиция в строке, с которой следует начинать поиск. Применяется регулярное выражение, ранее переданное в конструктор класса <code>Regex</code> .
<code>IsMatch(string input, string pattern)</code>	<code>input</code> – обрабатываемая строка, <code>pattern</code> – регулярное выражение.
<code>IsMatch(string input, string pattern, RegexOptions options)</code>	<code>input</code> – обрабатываемая строка, <code>pattern</code> – регулярное выражение, <code>options</code> – набор операций проверки.

Таблица 10.8 – Способы передачи данных в конструктор метода `Matches()`

<i>Метод</i>	<i>Описание</i>
<code>Matches</code> ( <code>string input</code> , <code>string pattern</code> )	Находит все вхождения в строку <code>input</code> , которая является обрабатываемым текстом, фрагментов, соответствующих <code>pattern</code> – тексту регулярного выражения.
<code>Matches</code> ( <code>string input</code> , <code>string pattern</code> , <code>RegexOptions options</code> )	Аналогичен предыдущему, но позволяет задать опции поиска.
<code>Matches</code> ( <code>string input</code> , <code>int start</code> )	Выполняет поиск с позиции <code>start</code> в строке <code>string</code> . Применяется регулярное выражение, переданное в конструктор класса <code>Regex</code> .

Таблица 10.9 – Способы передачи данных в конструктор метода `Replace()`

<i>Метод</i>	<i>Описание</i>
<code>Replace(string input, string pattern, string replacement)</code>	<code>input</code> – обрабатываемая строка, <code>pattern</code> – текст регулярного выражения, а <code>replacement</code> – текст, на который надо заменить найденные области.
<code>Replace(string input, string pattern, string replacement, RegexOptions options)</code>	<code>input</code> – обрабатываемая строка, <code>pattern</code> – текст регулярного выражения, <code>replacement</code> – текст, на который надо заменить найденные области. <code>options</code> – набор опций поиска.
<code>Replace(string input, string replacement)</code>	<code>input</code> – обрабатываемая строка, <code>replacement</code> – заменяющий текст. Применяется регулярное выражение, ранее переданное в конструктор класса <code>Regex</code> .
<code>Replace(string input, string replacement, int count)</code>	Полностью аналогичен предыдущему примеру, за исключением параметра <code>count</code> , который обозначает максимальное количество выполняемых замен.
<code>Replace(string input, string replacement, int count, int start)</code>	Полностью аналогичен предыдущему примеру, за исключением параметра <code>start</code> , который обозначает с какой позиции начать.

Таблица 10.10 – Способы передачи данных в конструктор метода `Split ()`

<i>Method</i>	<i>Описание</i>
<code>Split (string input, string pattern)</code>	Разбивает строку <code>input</code> на части, описанные в регулярном выражении <code>pattern</code> .
<code>Split (string input, int count)</code>	Разбивает строку <code>input</code> на части, описанные в регулярном выражении, которое было передано в конструктор класса <code>Regex</code> . Возвращает максимум <code>count</code> подстрок.