C# Регулярные выражения

Мухортова Н.Н.

Цель

Научиться создавать шаблоны строк и использовать регулярные выражения.

Регулярные выражения
Класс Regex

```
string s = "Бык тупогуб, тупогубенький бычок, у быка губа бела была тупа";

Regex regex = new Regex(@"туп(\w*)");

MatchCollection matches = regex.Matches(s);

if (matches.Count > 0){

   foreach (Match match in matches)

      Console.WriteLine(match.Value);

}
```

Параметр RegexOptions

Compiled: при установке этого значения регулярное выражение компилируется в сборку, что обеспечивает более быстрое выполнение

CultureInvariant: при установке этого значения будут игнорироваться региональные различия

IgnoreCase: при установке этого значения будет игнорироваться регистр

IgnorePatternWhitespace: удаляет из строки пробелы и разрешает комментарии, начинающиеся со знака #

Multiline: указывает, что текст надо рассматривать в многострочном режиме. При таком режиме символы "^" и "$" совпадают, соответственно, с началом и концом любой строки, а не с началом и концом всего текста

RightToLeft: приписывает читать строку справа налево

Singleline: устанавливает однострочный режим, а весь текст рассматривается как одна строка

Параметр RegexOptions

```
Regex regex = new Regex(@"туп(\w*)", RegexOptions.IgnoreCase);
```

При необходимости можно установить несколько параметров

```
Regex regex = new Regex(@"туп(\w*)", RegexOptions.Compiled | RegexOptions.IgnoreCase);
```

Синтаксис регулярных выражений

^: соответствие должно начинаться в начале строки (например, выражение @"^пр\w*" соответствует слову "привет" в строке "привет мир")

$: конец строки (например, выражение @"\w*ир$" соответствует слову "мир" в строке "привет мир", так как часть "ир" находится в самом конце)

.: знак точки определяет любой одиночный символ (например, выражение "м.р" соответствует слову "мир" или "мор")

Синтаксис регулярных выражений

*: предыдущий символ повторяется 0 и более раз

+: предыдущий символ повторяется 1 и более раз

?: предыдущий символ повторяется 0 или 1 раз

\s: соответствует любому пробельному символу

\S: соответствует любому символу, не являющемуся пробелом

Синтаксис регулярных выражений

\w: соответствует любому алфавитно-цифровому символу

\W: соответствует любому не алфавитно-цифровому символу

\d: соответствует любой десятичной цифре

\D : соответствует любому символу, не являющемуся десятичной цифрой

Проверка на соответствие строки формату

Нахождение телефонного номера в формате 111-111-1111

```
string s = "456-435-2318";

Regex regex = new Regex(@"\d{3}-\d{3}-\d{4}");
```

Если известно, сколько определенных символов должно быть, то можно явным образом указать их количество в фигурных скобках: \d{3} - то есть в данном случае три цифры

Проверка на соответствие строки формату

Переменная pattern задает регулярное выражение для проверки адреса электронной почты.

Данное выражение предлагается Microsoft на страницах msdn.

Пример в файле

Вывод

Регулярное выражение – это некий шаблон, составленный из символов и спецсимволов, который позволяет находить подстроки соответствующие этому шаблону в других строках.