Регулярные выражения

Лекции по курсу

Языки программирования

семестр 2

Что такое регулярное выражение

- Язык регулярных выражений – язык поиска и замены подстрок в строке
- Для поиска используется строка-образец (pattern, шаблон), содержащая метасимволы (wildcards)
- Виды регулярных выражений:
  - Базовые регулярные выражения POSIX
  - Расширенные регулярные выражения POSIX
  - Регулярные выражения, совместимые с Perl (C#)

Основные классы .NET

using System.Text.RegularExpressions;

Класс Regex

Статические методы:

- Match Regex.Match(s,pattern)
- bool Regex.IsMatch(s,pattern)
- MatchCollection Regex.Matches(s,pattern)
- string Regex.Replace(s,pattern,replace_s)

или экземплярные методы

- var r = new Regex(pattern);
- r.Match(s) r.IsMatch(s) r.Matches(s) r.Replace(s,replace_s)

Класс **Match –** методы и свойства

- m.Success, m.Value, m.Index, m.Length, m.NextMatch

Класс **MatchCollection –** свойство Count

Основной цикл

```
using System.Text.RegularExpressions;

var s = "red  green  gray blue";

foreach (Match m in Regex.Matches(s,@"\w+"))
    WriteLine($"{m.Value}  {m.Index}");
```

Примеры регулярных выражений

"кот|кит"
"к[ио]т"
"к[а-я]т"
"ко+т "
"кор?т"
"к.т"
"к\.т"
"к.*т"
"ко{3}т"
"ко{2,4}т"
"ко[^ъыь]т"
"\b кот\b"

Управляющие символы

\ * + ? | { [( ) ^ $. #

Имеют особый смысл в регулярном выражении

Экранируются символом \

Опции регулярных выражений

```
s = "АБракадабРА";
Regex.Matches("бра", RegexOptions.IgnoreCase)
```

Квантификаторы

* - 0 или более повторений
+ - 1 или более повторений
? - 0 или 1 повторение
{N} - ровно N повторений
{N,} - не менее N повторений
{N,M} - от N до M повторений

Директивы нулевой длины

^ - начало строки
$ - конец строки
\b - позиция на границе слова

Классы символов

[abcd] - один из символов, указанных в списке
[^abcd] - любой из символов, кроме тех, которые указаны в списке
[a-d]   - один из символов, лежащих в указанном диапазоне
[^a-d] - любой из символов, кроме тех, которые лежат в указанном диапазоне
\d - десятичная цифра: [0-9]
\D - любой символ, кроме десятичной цифры
\w - буква, цифра или символ подчеркивания
\W - любой символ, не являющийся словообразующим;
\s - пробельный символ, т. е. [\t\r\n];
\S - любой непробельный символ;
. - любой символ

Группы

```
s = " prepod@sfedu.ru   prepod2@sfedu.ru ";

foreach (Match m in Regex.Matches(s,@"(\w+)@(\w+)\.(\w+)"))
  WriteLine($"{m.Groups[0]},
                    {m.Groups[1]},
                    {m.Groups[2]},
                    {m.Groups[3]} ");
```

Вывод:
prepod@sfedu.ru, prepod, sfedu, ru
prepod2@sfedu.ru, prepod2, sfedu, ru

```
Regex.Matches(s,@"(т[оае])\1*");
```
\1 – ссылка на первую группу

Замена с помощью регулярного выражения

```
s = Regex.Replace(s,@"\b\w+\b", "<$0>");
s = Regex.Replace(s,@"\w+",m=>m.Value.ToUpper());
s = Regex.Replace(s,@"\w+",m=>m.Value+"("+m.Length+")")

s = "10+2=12";
s = Regex.Replace(s,@"\d+", "<$0>") // <10>+<2>=<12>
s = Regex.Replace(s,@"\d+",
    m => (int.Parse(m.Value)*2).ToString()); // 20+4=24

s = " prepod@sfedu.ru   prepod2@sfedu.ru ";
s = Regex.Replace(s, @" (\w+)@(\w+)\.(\w+)", @"$1-$2-$3")
```

Q & A