Регулярные выражения

Основной цикл

s := 'red green gray blue';

foreach var m in s.Matches('\w+') do

 Print(m.Value, m.Index);

m: Match: m.Value, m.Index

Опции регулярных выражений

s := 'AБракадабРА';

s.Matches('бра', RegexOptions.IgnoreCase)

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

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

Они экранируются дополнительным \

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 var m in s.Matches('(\w+)@(\w+).(\w+)') do

 Println(m.Groups[0], m.Groups[1], m.Groups[2], m.Groups[3]);

s.Matches('(т[оае])\1\*')

Замена с помощью регулярного выражения

s.Replace('\b\w+\b', '<$0>')

s := s.Replace('\w+',m->m.Value.ToUpper);

s := s.Replace('\w+',m->m.Value+'('+m.Length+')')

s := '10+2=12';

s.Replace('\d+', '<$0>') // <10>+<2>=<12>

s.Replace('\d+', m -> (m.Value.ToInteger\*2).ToString) // 20+4=24

s := ' prepod@sfedu.ru prepod2@sfedu.ru ';

s.Replace('(\w+)@(\w+).(\w+)', '$1-$2-$3')