<http://www.nakov.com/blog/2015/03/14/regular-expressions-qa-challenge-accepted/>

Наков – за регулярни изрази

Регулярните изрази в .NET Framework и в частност в C# са дефинирани от класа [System.Text.RegularExpression.Regex](http://msdn.microsoft.com/en-us/library/system.text.regularexpressions.regex.aspx). Най-често употребявани са три статични метода на класа:

* [public static MatchCollection Matches](http://msdn.microsoft.com/en-us/library/system.text.regularexpressions.regex.matches.aspx) (string input, string pattern, с или без RegexOptions) – приема като входни параметри стринг съдържащ оригиналния текст, в който ще се търси и стринг съдържащ шаблона за търсене и връща обект от тип MatchCollection, където са групирани всичките открити съвпадения, като при overload на метода, който приема и параметър RegexOptions може например да се зададе RegexOptions.IgnoreCase, за търсене без значение от регистъра на буквите. Тази колекция може да се обхожда и извежда например с foreach цикъл;
* [public static Match Match](http://msdn.microsoft.com/en-us/library/system.text.regularexpressions.regex.match.aspx)(string input, string pattern, с или без RegexOptions) – приема същите входни параметри като предходния метод, но връща само първото открито съвпадение като обект от тип Match, информация, за който може да се получи например чрез полето му Value или Index;
* [public static MatchCollection Replace](http://msdn.microsoft.com/en-us/library/system.text.regularexpressions.regex.replace.aspx)(string input, string pattern, MatchEvaluator evaluator)  – приема като входни параметри стринг съдържащ оригиналния текст, в който ще се търси, стринг съдържащ шаблона за търсене и обект от тип MatchEvaluator, който всъщност дава функцията за преобразуване на намерения стринг съвпадащ с шаблона, в нов стринг, който трябва да го замести. За задаването на тази функция много често  се използва ламбда израз, но може да се направи и чрез  дефиниране на обикновен метод. Съществува и overload на този метод, който приема като трети аргумент не обект от тип MatchEvaluator, а директно новия стринг, с който трябва да се направи заместването ;

Най – важния, но и най – трудния момент при работата с регулярни изрази всъщност е задаването на шаблона за търсене.  Затова ще се спра по-подробно на този момент. За задаването на шаблона се ползват определени предварително дефинирани символи, които могат да се прилагат заедно или поотделно и така да образуват доста сложно конструкции. Настоящата статия има за цел да опише само най-често употребяваните , защото езика на регулярните изрази може да бъде изключително сложен*.*

Символи заместващи други символи или групи от символи:

**.** – означава множеството от всички възможни символи, например шаблона “h.t” ще открие съвпадение при “hit”, “hat“, “h1t”, “h$t” и т.н.;

**|**  – означава алтернатива „или“ например шаблона “art|box“, означава, че се търси или  стринга „art“ или стринга „box“;

**[]**– търси единичен символ, който съвпада с множеството от символи описани вътре в скобите, например: шаблона  “1[0-9]0” ще открие съвпадение при “100“, “110“, “120“, “130“, “140“, “150“, “160“, “170“, “180“ и “190“, а шаблона ”b[aiu]t ” ще намери съвпадение само с “bat“,“bit“ и “but“.

**[^ ]** – означава изключване на определен символ посочен вътре в скобите [], например шаблона “ b[^ie]t“  ще открие съвпадение при ”bat” и ”but”, но не и при ”bit” и ”bet”;

**–**  –  означава  определен интервал от символи, например: [a-z] означава всички малки букви, а [0-9] всички цифри, а когато е използвано в конструкцията [a-z-[bc]] означава изваждане на две множества, тоест примера означава всички малки букви от a до z без буквите b и c;

**^** – когато е без скоби означава, че се търси показания шаблон само когато той е в началото на реда, например шаблона “^b[ie]t” означава, че ще се търсят думите ”bit” и ”bet”, но само когато с тях започва изречението;

**$** – означава, че се търси показания шаблон само когато той е в края на реда, например шаблона “b[ie]t$” означава, че ще се търсят думите ”bit” и ”bet”, но само когато с тях завършва изречението;

Символи означаващи количество

**“+”**                означава едно или повече повторения, например шаблона “ b[aiu]+t“ ще открие съвпадение при  ”bat”, ”baat”, ”bait”, ”buauuiit”, ”bit” и т.н.;

“\*“                 означава нула или повече повторения, например шаблона “ b[aiu]\*t“ ще открие съвпадение при  ”bat”, ”baat”, ”bait”, ”buauuiit”, ”bit”, но и при “bt“, за разлика от предишния пример;

**“?“**                                означава нула или едно повторения, например шаблона “ b[aiu]?t“ ще открие съвпадение при  ”bat”, но не и при ”baat”, при ”bait”, но не и при ”buauuiit”, при ”bit” , а също и при „bt“;

**“ {n, m}“**     означава поне n повторения, но не повече от m, например шаблона “b[aiu]{2,3}t“  ще открие съвпадение при  ”baat”, но не и при ”bat”, при  ”baaauuiit”, но не и при ”bait”. В този случай може да се изпусне m, тоест конструкцията става {n,} и означава поне n повторения, или пък да стане {n}, което означава точно n повторения;

**“()“**                               означава групиране на няколко символа, като след това върху тази група могат да бъдат поставени например символи за количество , например шаблона “(01)+“ ще открие съвпадение при стринговете “01“ “0101“ “010101“, но не и при “001“ и “111“ за разлика от шаблона “[01]+“, или шаблона “(01){3}“ ще открие съвпадение само при стринга “010101“. Тази група затворена между скобите може да се достъпва след това и по номер, например шаблона “(ab)/1“ ще открие съвпадение при стринга “abab“, защото “(ab)“ означава групата ”ab”, a ”/1” означава първата група от символи, което отново е “ab“;

Escaping при регулярни изрази

Символите, които имат специално значение в езика на регулярните изрази наричаме метасимволи. Това са: **. \* ? + [] () {} ^ $ и \** Другите символи, които участват в стринга на шаблона наричаме литерали и те се интерпретират буквално. Ако искаме да търсим в текст стринг съдържащ някой от метасимволите трябва да укажем изрично, че това не е метасимвол в случая, а че трябва да се интерпретира буквално, тоест като литерал. Това става с така наречения escaping. При него преди всеки един от метасимволите трябва да се сложи ‘\‘, за да се интерпретира той като литерал. Например шаблона “[0-9]\*/.[0-9]+“  ще открие съвпадение при “23.75“, а  шаблона “[0-9]\*.[0-9]+“  ще открие също съвпадение при “23.75“, но и при “23$75“ и “23 75“.

Символи представляващи предварително дефинирани множества

**/w**    означава всички цифри, букви и ‘\_‘,  тоест [a-zA-Z0-9\_];

**/W**   означава всички символи, които са различни от букви, цифри и ‘\_‘  или [^a-zA-Z0-9\_];

**/b**     означава границата между буквен  ‘/w’ и небуквен символ ‘/W’, тоест това са границите между думите в изречението например;

**/d**     означава всички цифри или [0-9];

**/D**    означава всичко, което не е цифра или [^0-9];

**/s**     означава всички празни пространства или [ \t\r\n\v\f];

**/S**     означава всичкo, което не е празно пространство или [^ \t\r\n\v\f];

1. За дата XX-XX-XXXX

(?<day>[1-9]|0[1-9]|[12][0-9]|3[01])\-(?<month>[1-9]|0[1-9]|1[012])\-(?<year>20[01][0-9])

([1-9]|0[1-9]|[12][0-9]|3[01])\-([1-9]|0[1-9]|1[012])\-(20[01][0-9])

1. За име и фамилия

([А-Я][а-я]\*\s){2}

1. За email

[A-Za-z0-9\_\-\.][+@[a-z]{3,5}\.[a-z]{2,3}](mailto:+@[a-z]%7b3,5%7d\.%5ba-z%5d%7b2,3%7d)

1. Аритметичен израз

[-]?([1-9][0-9]\*[\+\-\\*/])+ [1-9][0-9]\*