Dla większości prostych gramatyk można w łatwy sposób napisać wyrażenie regularne które będzie służyło do sprawdzania poprawności zdań z tą gramatyką.

Celem niniejszego laboratorium będzie zapoznanie się z wyrażeniami regularnymi (Regex) występującymi w języku C# oraz napisanie odpowiednich wyrażeń regularnych do sprawdzania poprawności zdań z zadanymi gramatykami.

1. Podstawy budowania wyrażeń regularnych (Regex)

Znaki:

**. $ ^ { [(|)] } * + ? **

są w wyrażeniach regularnych znakami specjalnymi.

Przy użyciu poniższych symboli można zaznaczyć dopasowanie do 'klasy' znaków.

.	Dowolny znak
[abc]	Dowolny znak ze wskazanego zbioru
[^abc]	Dowolny znak różny od tych w zbiorze
[a-z0-9]	Dowolny znak z zakresu
\w	Dowolny znak typu: litera, cyfra, podkreślnik
\W	Dowolny znak inny niż: litera, cyfra, podkreślnik
\s	Dowolny biały znak (spacja, …)
\S	Dowolny znak różny niż znak biały
\d	Dowolna cyfra
\D	Dowolny znak inny niż cyfra

Przy użyciu poniższych symboli można zaznaczyć miejsce w tekście gdzie ma dojść do dopasowania.

^	Dopasowanie musi się pojawić na początku tekstu
$	Dopasowanie musi się pojawić na końcu tekstu
\b	Dopasowanie musi się znajdować na początku lub końcu słowa
\B	Dopasowanie nie może się znajdować na początku ani końcu słowa

Za pomocą poniższych symboli można określić ile razy dany znak (lub grupa znaków) ma wystąpić.

*	Zero lub więcej razy
+	Jeden lub więcej razy
?	Zero lub jeden raz
{n}	Dokładnie n razy
{n,}	Co najmniej n razy
{n,m}	Co najmniej n razy ale nie więcej niż m razy

Poniżej przedstawiono kilka innych elementów.

\	Ukośnik jest również znakiem ucieczki, tzn. likwiduje znaczenie znaków specjalnych, np. **\.** Znaczy dosłownie „kropka”
\0x20	Kod szesnastkowy znaku ASCII
\u0020	Kod szesnastkowy znaku Unicode
( )	Nawiasy zwykłe używane są do grupowania symboli
|	Pionowa kreska oznacza alternatywę

Uwaga: Wyrażenia regularne są wpisywane jako łańcuchy znakowe (String), w których znak ukośnika (\) ma również specjalne znaczenie, dlatego wpisując wyrażenie regularne należy podwójnie wpisywać znak ukośnika jeśli ma on pełnić funkcje znaku ucieczki. Tzn. zamiast wpisać „\\” wpisujemy „\\\\”, ale jeżeli znak ukośnika nie pełni funkcji znaku ucieczki to wpisujemy normalnie np. „\d”.

Przykład działania wyrażeń regularnych:

- sprawdzanie kodu pocztowego: `^\d{2}-\d{3}$`
- sprawdzenie adresu www: `^www\.[-\w]+(\.[-\w]+)+$`
- sprawdzanie adresu IP: `^0*(25[0-5]|2[0-4]\d|1?\d\d?)(\.0*(25[0-5]|2[0-4]\d|1?\d\d?)){3}$`

Jeżeli chcemy dokonać dokładnego sprawdzenia zgodności wpisanego tekstu z wyrażeniem regularnym to na początku wyrażenia regularnego musimy umieścić znak ^ a na jego końcu znak $.

Więcej informacji o wyrażeniach regularnych:
http://www.codeguru.pl/article-696.aspx
http://www.radsoftware.com.au/articles/regexlearnsyntax.aspx

2. Jak przejść z definicji gramatyki na wyrażenia regularne

Do przejścia na wyrażenia regularne najlepiej nadają się gramatyki zapisane w notacji MBNF w której dochodzą dodatkowe symbole:

[ ]	Oznacza wystąpienie jakiegoś symbolu zero lub jeden raz
{ }	Oznacza wystąpienie jakiegoś symbolu nieokreśloną ilość razy
( )	Oznacza grupę symboli

Przejście z gramatyki na wyrażenie regularne zostanie przedstawione na przykładzie prostej gramatyki dla adresu e-mail.

S ::= A@A.W
A ::= W{.W}
W ::= L{L}
L ::= a|b|c|d|e|...|x|y|z

Następnie należy kolejno po podstawiać do poszczególnych symboli. W rezultacie otrzymamy:

```
S::=(a|b|c|...|z){(a|b|c|...|z)}{.(a|b|c|...|z){(a|b|c|...|z)}}
    @(a|b|c|...|z){(a|b|c|...|z)}{.(a|b|c|...|z){(a|b|c|...|z)}}
    .(a|b|c|...|z){(a|b|c|...|z)}
```

Z kolei ten zapis jest już bez mała wyrażeniem regularnym:

```
^[a-z]+(\.[a-z]+)*@[a-z]+(\.[a-z]+)*\.[a-z]+$
```

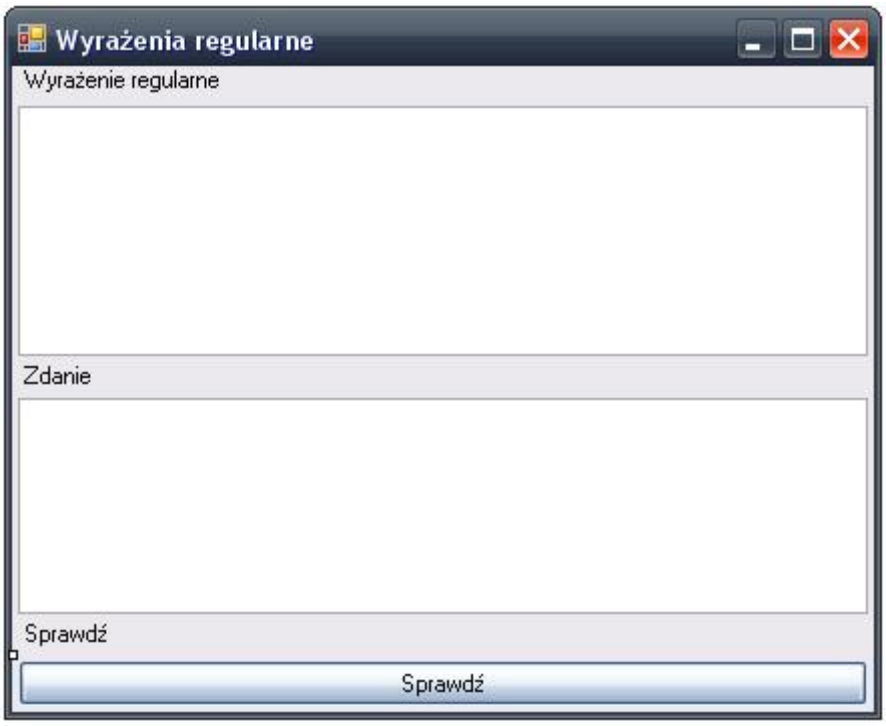
Niestety jeżeli w gramatyce występuje rekurencyjne wywołanie jakiegoś symbolu, to nie można dokonać podstawienia i nie da się zbudować wyrażenia regularnego do sprawdzenia tej gramatyki. Np.

$$S ::= W;\{W;\}$$
$$W ::= POP\{OP\}$$
$$P ::= L \mid (W)$$
$$L ::= C\{C\}$$
$$C ::= 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid 0$$
$$O ::= * \mid / \mid + \mid - \mid \hat{}$$

Gdy podstawimy P do W to w W będziemy mieli rekurencyjne odwołanie do W. i nie będzie możliwe dokonanie dalszych podstawień.

3. Program

W trakcie laboratorium można korzystać z przygotowanego programu w którego jednym polu należy wpisać wyrażenie regularne a w drugim zdanie do sprawdzenie przy pomocy tego wyrażenia regularnego.



4. Zadania

Napisać wyrażenia regularne dla sprawdzania zgodności zdań z zadanymi gramatykami:

$S ::= W;\{W;\}$
$W ::= LOL\{OL\}$
$L ::= C\{C\}$
$C ::= 1|2|3|4|5|6|7|8|9|0$
$O ::= *|/|+|-|\hat{}$

```
12+2;3*8^12-2/3;
```

$S ::= A[.L][eA]$
$A ::= [-]L$
$L ::= C\{C\}$
$C ::= 1|2|3|4|5|6|7|8|9|0$

```
-15.6e-3
```

Wymyślić gramatykę i napisać dla niej wyrażenie regularne.