Reguläre Ausdrücke

Ingo Köster

Diplom Informatiker (FH)

Reguläre Ausdrücke

- Text nach bestimmten Mustern durchsuchen
- Klasse `Regex`
- Aus dem Namensraum `System.Text.RegularExpressions`
- Einige Methoden
  - `Regex.IsMatch()`
  - `Regex.Match()`
  - `Regex.Replace()`
  - `Regex.Split()`
  - `Regex.Matches()`

Reguläre Ausdrücke (Auswahl)

Regex	Bedeutung
`[a-d]`	ein Zeichen aus der Menge a, b, c oder d
`[^abc]`	kein Zeichen aus der Menge a, b oder c
`^`	Anfang einer Zeichenkette
`$`	Ende einer Zeichenkette
`.` *(Punkt)*	genau ein beliebiges Zeichen (auch Leerzeichen)

Reguläre Ausdrücke (Auswahl)

Regex	Bedeutung
*	Platzhalter für das vorherige Zeichen; mögliche Anzahl 0 bis ∞
?	Platzhalter für das vorherige Zeichen; mögliche Anzahl 0 bis 1
+	Platzhalter für das vorherige Zeichen; mögliche Anzahl 1 bis ∞

Reguläre Ausdrücke (Auswahl)

Regex	Bedeutung
\b	Wortgrenze. Position zwischen einem Wort und einem Leerzeichen. "er\b" passt auf "er" in "lieber", nicht auf "er" in "herb"
\s	Jedes Leerzeichen, Tabulator, etc.
\w	Wort-Zeichen. Entspricht "[A-Za-z0-9_]“. Inklusive Unterstrich
\d	Ziffer. Entspricht “[0-9]“
\S , \W & \D	Negation der Bedeutung

Beispiel - IsMatch

- Zwischen Anfang und Ende genau 5 Ziffern
  - \d Eine beliebige Ziffer ([0-9])
  - {n} Anzahl der Wiederholungen

```
Regex.IsMatch("12345", @"^\d{5}$");  // true
```

```
Regex.IsMatch("1234", @"^\d{5}$");   // false
```

Beispiel - IsMatch

› Regex für eine E-Mail-Adresse

```
Regex.IsMatch(text,
@"\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*")
```

› Hinweis: Reguläre Ausdrücke am besten immer mit einem @ beginnen

Überladung mit RegexOptions (Auswahl)

- Optionen für Regex-Methoden festlegen

- Unter:
  - http://msdn.microsoft.com/de-De/library/system.text.regularexpressions.regexoptions.aspx

- Beispiel:
  - `Regex.IsMatch("Hallo", @"^H");  // true`
  - `Regex.IsMatch("hallo", @"^H");  // false`
  - `Regex.IsMatch("hallo", @"^H", RegexOptions.IgnoreCase);  // true`

Regex.Match

› Durchsucht eine Zeichenfolge nach einer Teilzeichenfolge, die mit einem Muster eines regulären Ausdrucks übereinstimmt

```
string input = "Company Name: Contoso, Inc.";
Match m = Regex.Match(input, "Company Name: (.*$)");
Console.WriteLine(m.Groups[0]);
Console.WriteLine(m.Groups[1]);
```

› Gibt aus:
  › Company Name: Contoso, Inc.
  › Contoso, Inc.

Beispiel - Replace

```
string element = "H4ll0 W3lt";
string text = Regex.Replace(element, @"\d", "");
```

› Inhalt von Text:
  › Hll Wlt

Referenz

- MSDN Referenz zu Regulären Ausdrücken
- https://docs.microsoft.com/de-de/dotnet/standard/base-types/regular-expression-language-quick-reference

Encoding

Klasse `Encoding`

- Verschiedene Zeichensätze
  - UTF-7, UTF-8, ASCII, etc.
- Unterstütze Encodings anzeigen:
- `EncodingInfo[] infos = Encoding.GetEncodings();`
  - Attribute: `CodePage, Name, DisplayName`

- Encoding beim Lesen und Schreiben angeben
- `StreamReader sr = new StreamReader("dateiname", Encoding.UTF8);`
  - Ohne Angabe des Encodings wird passendes Encoding gewählt