

Bài 7.1: Tổng quan về Regular expression

- ✓ Khái niệm, mục đích sử dụng
- ✓ Các kí tự đặc biệt và siêu kí tự
- ✓ Lớp Regex
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Khái niệm, mục đích sử dụng

- ✓ Regular expression dịch ra là biểu thức chính quy. Thường được sử dụng trong so khớp mẫu khi cần kiểm tra dữ liệu đầu vào trong các string.
- ✓ Từ đó ta có thể thực hiện các hành động chuyển đổi kiểu dữ liệu, thay thế, tìm kiếm, sửa đổi dữ liệu theo mục đích sử dụng.
- ✓ Ví dụ: sử dụng regular expression để kiểm tra dữ liệu họ và tên có đúng định dạng không. Kiểm tra định dạng mã sinh viên, email, số điện thoại, đầu điểm...

Các kí tự đặc biệt

✓ Các kí tự đặc biệt bắt đầu với dấu \ sau đó là 1 hoặc nhiều kí tự khác.

Kí tự	Mô tả
\a	Khớp với kí tự thể hiện tiếng chuông \u0007.
\b	Khớp với backspace \u0008.
\t	Khớp với kí tự dấu tab \u0009.
\r	Khớp với 1 dấu xuống dòng \u000D, khác \n.
\v	Khớp với 1 dấu tab dọc \u000B.
\f	Khớp với 1 nguồn cấp dữ liệu.
\n	Khớp với 1 dòng trống \u000A.
\e	Khớp với kí tự thoát \u001B.
\nnn	Sử dụng hệ cơ số 8 để xác định một kí tự. 3 chữ n tương ứng 3 chữ số. Ví dụ \040.
\u nnnn	Khớp với kí tự unicode bằng cách sử dụng biểu diễn của hệ 16. 4 chữ n tương ứng với 4 chữ số ví dụ \u0020.
\	Khi đứng trước các kí tự không phải kí tự đặc biệt, nó có nghĩa là khớp với kí tự đó hoặc ý nghĩa của siêu kí tự đó.

Các siêu kí tự và mô tả

\z	Việc so khớp phải xảy ra tại cuối chuỗi.
\G	Việc so khớp phải xảy ra tại vị trí chuỗi so khớp trước đó kết thúc.
\b	Việc so khớp phải xảy ra trong phạm vi giữa \w và \W.
\B	Việc so khớp không được xảy ra trong phạm vi của \b.
*	Khớp phần tử trước dấu * 0 hoặc nhiều lần.
+	Khớp phần tử trước dấu + 1 hoặc nhiều lần.
?	Khớp phần tử trước dấu ? 0 hoặc 1 lần. Tức là có hoặc không có phần tử trước đó.
{n}	Khớp phần tử trước đó chính xác n lần.
{n,}	Khớp phần tử trước đó tối thiểu n lần.
{m,n}	Khớp phần tử trước đó từ m tới n lần. Nhưng không quá n lần.
*?	Khớp phần tử trước đó 0 hoặc nhiều lần nhưng càng ít càng tốt.
+?	Khớp phần tử trước đó 1 hoặc nhiều lần nhưng càng ít càng tốt.
??	Khớp phần tử trước đó 0 hoặc 1 lần nhưng càng ít càng tốt.
{n}?	Khớp phần tử trước đó chính xác n lần.
{n,}?	Khớp phần tử trước đó ít nhất n lần nhưng càng ít càng tốt.
{m,n}?	Khớp phần tử trước đó từ m đến n lần nhưng càng ít càng tốt.
()	Gom nhóm các mẫu so khớp.
	Khớp với một trong các phần tử ở bên trái hoặc bên phải kí tự .
(?(exp)yes no)	Khớp với vế yes nếu exp được thỏa mãn và no nếu ngược lại.

Các siêu kí tự và mô tả

Kí tự	Mô tả
[abc]	Khớp với bất kỳ kí tự đơn nào nằm bên trong cặp móc vuông [].
[^cde]	Khớp với bất kỳ kí tự nào không nằm trong cặp móc vuông [], sau kí tự ^.
[first-last]	Khớp với các kí tự trong đoạn first đến last. Ví dụ mẫu [a-z] sẽ khớp bất kì kí tự thường nào trong bảng chữ cái từ a-z.
.	Khớp bất kỳ kí tự nào trừ khoảng trắng \n.
\p{name}	Khớp kí tự đơn trong name ở dạng phân loại unicode hoặc khối được đặt tên xác định bởi name. Ví dụ \p{Lu} sẽ khớp với 'C' và 'L' trong "City Lights".
\P{name}	Khớp với bất kỳ kí tự nào không thuộc phân loại unicode hoặc khối được đặt tên chỉ định bởi name. Ví dụ \P{Lu} khớp với 'i', 't', 'y' trong "City".
\w	Khớp với bất kỳ kí tự chữ cái hoặc chữ số nào.
\W	Khớp với bất kỳ kí tự nào không phải chữ cái, chữ số.
\s	Khớp với kí tự dấu cách.
\S	Khớp với kí tự không phải dấu cách.
\d	Khớp với 1 kí tự chữ số.
\D	Khớp với bất kỳ kí tự nào không phải chữ số.
^	Việc so khớp phải bắt đầu ở đầu dòng hoặc đầu chuỗi kí tự.
\$	Việc so khớp phải kết thúc ở cuối chuỗi kí tự hoặc trước \n tại cuối dòng, cuối chuỗi.
\A	Việc so khớp phải xảy ra tại vị trí bắt đầu của chuỗi kí tự.
\Z	Việc so khớp phải xảy ra tại vị trí kết thúc của chuỗi hoặc trước \n cuối chuỗi.

Lớp Regex

- ✓ Lớp này sử dụng để thể hiện một biểu thức chính quy bất biến.
- ✓ Các trường dữ liệu của enum `RegexOptions` sử dụng trong constructor của lớp `Regex`:
 - ✓ **IgnoreCase**: so khớp không phân biệt chữ hoa/thường.
 - ✓ **Compiled**: chỉ định biểu thức chính quy biên dịch vào code MSIL thay vì thông dịch. Tăng tối đa hiệu năng trong thời gian chạy.
 - ✓ **Multiline**: bật chế độ so khớp cho `^` và `$` khớp ở đầu và cuối bất kì dòng nào.
 - ✓ **None**: không có tùy chọn nào được thiết lập.
 - ✓ **RightToLeft**: chỉ định việc tìm kiếm sẽ diễn ra theo thứ tự từ phải sang trái.
 - ✓ **Singleline**: chỉ định chế độ áp dụng trên 1 dòng. Thay đổi ý nghĩa của siêu kí tự `.` để cho phép khớp mọi kí tự.
- ✓ Khi sử dụng lớp này ta khai báo `using System.Text.RegularExpressions;` ở đầu file chương trình.

Các phương thức thường dùng của lớp Regex

Phương thức và mô tả
<i>Regex()</i> : khởi tạo một đối tượng mới của lớp Regex.
<i>Regex(string pattern)</i> : khởi tạo một đối tượng của lớp Regex với biểu thức mẫu được chỉ định trong tham số.
<i>Regex(string pattern, RegexOptions options)</i> : tạo mới đối tượng Regex với biểu thức mẫu và tùy chọn được cho sẵn.
<i>bool IsMatch(string input)</i> : xác định liệu biểu thức so khớp đã chỉ định có khớp với input không. <i>bool IsMatch(string input, int startPos)</i> : tương tự phương thức trên nhưng bắt đầu việc so khớp tại vị trí startPos của chuỗi đầu vào. <i>bool IsMatch(string input, string pattern)</i> : xác định xem mẫu so khớp pattern có khớp với chuỗi đầu vào input không. <i>bool IsMatch(string input, string pattern, RegexOptions options)</i> : xác định xem mẫu so khớp có khớp với chuỗi đầu vào sử dụng tùy chọn so khớp cho sẵn.
<i>Match Match(string input)</i> : tìm sự xuất hiện đầu tiên của mẫu so khớp trong input. <i>Match Match(string input, int startAt)</i> : tìm sự xuất hiện đầu tiên của mẫu so khớp trong input bắt đầu tại vị trí startAt. <i>Match Match(string input, string pattern)</i> : tìm sự xuất hiện đầu tiên của pattern trong input. <i>Match Match(string input, string pattern, RegexOptions options)</i> : tương tự phương thức trên nhưng bổ sung thêm tùy chọn so khớp trong tham số options.
<i>MatchCollection Matches(string input)</i> : tìm tất cả các lần xuất hiện của mẫu so khớp trong input. <i>MatchCollection Matches(string input, int startAt)</i> : tìm tất cả các lần xuất hiện của mẫu so khớp trong input bắt đầu tại vị trí startAt. <i>MatchCollection Matches(string input, string pattern)</i> : tìm tất cả các lần xuất hiện của pattern trong input. <i>MatchCollection Matches(string input, string pattern, RegexOptions options)</i> : tương tự phương thức trên nhưng bổ sung thêm tùy chọn so khớp trong tham số options.

Các phương thức thường dùng của lớp Regex

string Replace(string input, string replacement): thay thế tất cả các vị trí xuất hiện của biểu thức chính quy trong tham số input bởi nội dung của tham số replacement.

string Replace(string input, string replacement, int count): thay thế tối đa count vị trí xuất hiện của biểu thức chính quy trong tham số input bởi nội dung của tham số replacement.

string Replace(string input, string pattern, string replacement): thay thế tất cả các vị trí xuất hiện của mẫu pattern trong tham số input bởi nội dung của tham số replacement.

string Replace(string input, string pattern, string replacement, RegexOptions options): thay thế tất cả các vị trí xuất hiện của mẫu pattern trong tham số input bởi nội dung của tham số replacement. Bổ sung tùy chọn so khớp trong tham số options.

string[] Split(string input): tách chuỗi đầu vào trong input thành mảng các chuỗi con tại vị trí xuất hiện của mẫu so khớp trong phương thức khởi tạo.

string[] Split(string input, int count): tách chuỗi đầu vào trong input thành mảng tối đa count chuỗi con tại vị trí xuất hiện của mẫu so khớp trong phương thức khởi tạo.

string[] Split(string input, string pattern): tách chuỗi đầu vào trong input thành mảng các chuỗi con tại vị trí xuất hiện của mẫu so khớp trong tham số pattern.

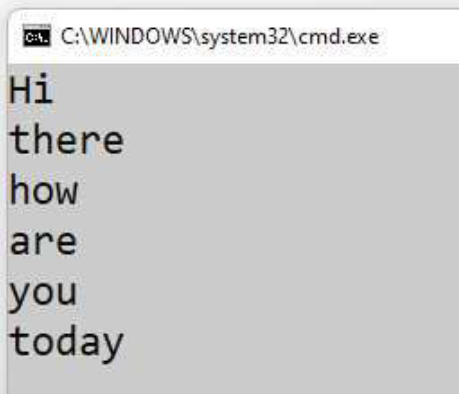
string[] Split(string input, int count, int startAt): tách chuỗi đầu vào trong input thành mảng tối đa count chuỗi con tại vị trí xuất hiện của mẫu so khớp trong phương thức khởi tạo. Việc so khớp bắt đầu tại vị trí startAt của chuỗi input.

string[] Split(string input, string pattern, RegexOptions options): tách chuỗi đầu vào trong input thành mảng các chuỗi con tại vị trí xuất hiện của mẫu so khớp trong phương thức khởi tạo. Bổ sung tùy chọn so khớp trong tham số options.

Ví dụ minh họa

- ✓ Tách từ tại vị trí có một hoặc nhiều dấu cách:

```
0 references
static void Main(string[] args)
{
    var input = "Hi there      how are      you      today      ";
    var pattern = @"\s+";
    var regex = new Regex(pattern);
    var words = regex.Split(input);
    // in kết quả
    foreach (var item in words)
    {
        Console.WriteLine(item);
    }
}
```



Ví dụ minh họa

- ✓ Thay thế một hoặc nhiều dấu cách bằng 1 dấu cách:

```
0 references
static void Main(string[] args)
{
    var input = "Hi there      how are      you      today      ?      ";
    var pattern = @"\s+";
    var regex = new Regex(pattern);
    var result = regex.Replace(input, " ");
    // in kết quả
    Console.WriteLine("==> Truoc khi thay the: ");
    Console.WriteLine(input);
    Console.WriteLine("==> Sau khi thay the: ");
    Console.WriteLine(result);
}
```

C:\WINDOWS\system32\cmd.exe

```
==> Truoc khi thay the:
Hi there      how are      you      today      ?
==> Sau khi thay the:
Hi there how are you today ?
Press any key to continue . . .
```



Nội dung tiếp theo

So khớp địa chỉ email