Matching via Character classes

Victor Grazi
ORACLE JAVA CHAMPION, SPEAKER AND GEEK

@vgrazi

Introduction

Predefined character classes \d, \s, \w

and their inverses \D, \S, \W

Defining your own character classes

Ranges of characters

Negation of home made character classes

Unions and intersections of character classes

Alternation

POSIX character classes (US-ASCII only)

`\p{Lower}`	A lower-case alphabetic character: `[a-z]`
`\p{Upper}`	An upper-case alphabetic character:`[A-Z]`
`\p{ASCII}`	All ASCII:`[\x00-\x7F]`
`\p{Alpha}`	An alphabetic character
`\p{Digit}`	A decimal digit: `[0-9]`
`\p{Alnum}`	An alphanumeric character
`\p{Punct}`	Punctuation:
`\p{Graph}`	A visible character: `[\p{Alnum}\p{Punct}]`
`\p{Print}`	A printable character: `[\p{Graph}\x20]`
`\p{Blank}`	A space or a tab: `[ \t]`
`\p{Cntrl}`	A control character: `[\x00-\x1F\x7F]`
`\p{XDigit}`	A hexadecimal digit: `[0-9a-fA-F]`
`\p{Space}`	A whitespace character: `[ \t\n\x0B\f\r]`

java.lang.Character classes (simple java character type)

`\p{javaLowerCase}`	Equivalent to java.lang.Character.isLowerCase()
`\p{javaUpperCase}`	Equivalent to java.lang.Character.isUpperCase()
`\p{javaWhitespace}`	Equivalent to java.lang.Character.isWhitespace()
`\p{javaMirrored}`	Equivalent to java.lang.Character.isMirrored()

Classes for Unicode scripts, blocks, categories and binary properties

`\p{IsLatin}`	A Latin script character (script)
`\p{InGreek}`	A character in the Greek block (block)
`\p{Lu}`	An uppercase letter (category)
`\p{IsAlphabetic}`	An alphabetic character (binary property)
`\p{Sc}`	A currency symbol
`\P{InGreek}`	Any character except one in the Greek block (negation)
`[\p{L}&&[^\p{Lu}]]`	Any letter except an uppercase letter (subtraction)

Numeric digits: 0123456789

“Word” characters

Alphanumeric or “_”

Whitespace (“ “, tabs, or new lines)

Escaped characters \

Character set []

Grouping ()

Concatenation

Anchoring ^$

Alternation |

```
public void findFile() {
    File dir = new File(".");
    FileFilter fileFilter = new RegexFileFilter
            ("^.*[mM]y-file(-\\d+)?\\.java$");
    File[] files = dir.listFiles(fileFilter);
    for (int i = 0; i < files.length; i++) {
        System.out.println(files[i]);
    }
}
```

replaceAll()

Replaces all occurrences of the supplied regex

```
“one + one = 2”.replaceAll(“one”, “1”)
> 1 + 1 = 2
```

replaceFirst()

Replaces just the first occurrence of the supplied regex

```
“one + one = 2”.replaceFirst(“one”, “1”)
> 1 + one = 2
```

Type	Example	RegEx
Digit:	1	\d
whiteSpace:	(Tab, space, new line)	\s
Word character	Alphanumeric or underscore (_)	\w
Anything but a Digit:	a	\D
Anything but a whiteSpace:	1	\S
Anything but a Word character	-	\W

String class Regex support

matches()

split()

replaceAll()

replaceFirst()

*** But not replace()**

Summary

- File searching
- Validation
- Text searching
- Text extraction
- Expression reformulating
- Text cleansing

Summary

Predefined character classes \d, \s, \w

and their inverses \D, \S, \W

Defining your own character classes

Ranges of characters

Negation of home made character classes

Unions and intersections of character classes

Alternation