REGULAR EXPRESSIONS

Определение

Формальный язык поиска и осуществления манипуляций с подстроками в тексте.

Основан на использовании метасимволов.

Символы

x ==> символ x

\\ ==> обратный слеш

\xhh ==> символ с кодом U+00hh

\xhhhh ==> символ с кодом U+hhhh

\n ==> перевод строки

\r ==> возврат каретки

\t ==> табуляция

Символьные классы

Простой класс:

[abc] ==> a, b или c

Регулярное выражение: [ab]

Входная строка: **a**ccdd**b****a**

Символьные классы

Отрицание:

[^abc] ==> любой символ, кроме a, b, c

Регулярное выражение: [^ab]

Входная строка: a**c****d**ba

Символьные классы

Диапазон:

[a-zA-Z] ==> от a до z или от A до Z

Регулярное выражение: [a-cA-Z]

Входная строка: adAcdh

Символьные классы

Объединение:

[a-z[A-Z]] ==> от a до z или от A до Z

Тоже самое, что и [a-zA-Z]

Символьные классы

Пересечение:

[a-z&&[def]] ==> d, e или f

Регулярное выражение: [a-d&&c-f]

Входная строка: accdddeab

Символьные классы

Вычитание:

[a-z&&[^def]] ==> от a до c или от g до z

Регулярное выражение: [a-d&&[^c-f]]

Входная строка: **a**ccddde**ab**

Символьные классы Java

Эквиваленты методов класса Character:

\p{javaLowerCase} ~ isLowerCase

\p{javaUpperCase} ~ isUpperCase

\p{javaWhitespace} ~ isWhitespace

Регулярное выражение:

\p{javaUpperCase} \p{javaLowerCase}

Входная строка: Текст

Предопределенные классы

. ==> любой символ

Регулярное выражение: ...

Входная строка: **abcdef**gh

Регулярное выражение: ..

Входная строка: **abcd**e

Предопределенные классы

\d ==> цифра, [0-9]

\D ==> не цифра, [^\d]

Регулярное выражение: \d\D

Входная строка: ab**8a**b8

Предопределенные классы

\s ==> пробельный символ, [\t\n\f\r\x0b]

\S ==> непробельный символ, [^\s]

Регулярное выражение: \s\S

Входная строка: ab 8 ab8

Предопределенные классы

\w ==> символ слова, [a-zA-Z_\d]

\W ==> отрицание \w, [^\w]

Регулярное выражение: \w\W

Входная строка: ab*8&ab8

Границы

^ ==> начало строки

Регулярное выражение: ^ab
Входная строка: ababab

$ ==> конец строки

Регулярное выражение: ab$
Входная строка: ababab

Границы

\b ==> граница слова

\B ==> отрицание \b

Регулярное выражение: abc\b

Входная строка: abc abcd

Границы

\A ==> начало ввода

Регулярное выражение: \Aabc

Входная строка:

abc abc

abc abc

Границы

\z ==> конец ввода

Регулярное выражение: abc\z

Входная строка:

abc abc

abc **abc**

Границы

\Z ==> конец ввода, как и \z, но ***может*** включать ограничитель строки

Регулярное выражение: abc\Z

Входная строка:

abc abc

abc **abc<ограничитель строки>**

Ограничители строк

'\n'	==>	LF (новая строка)
'\r'	==>	CR (возврат каретки)
"\r\n"	==>	CR+LF
'\u0085'	==>	следующая строка
'\u2028'	==>	разделитель строки
'\u2029'	==>	разделитель параграфа

Квантификаторы

Квантификатор определяет повторяемость.

Жадный квантификатор определяет максимально возможную подстроку.

Ленивый квантификатор определяет минимально возможную подстроку.

Квантификаторы

X? ==> один или ноль раз (жадный)

Регулярное выражение: ab?

Входная строка: **a**a**b**c**a**bbb

X?? ==> один или ноль раз (ленивый)

Регулярное выражение: ab??

Входная строка: **a**a**b**c**a**bbb

Квантификаторы

X*==> ноль или более раз (жадный)

Регулярное выражение: ab*

Входная строка: **aabcabbb**

X*? ==> ноль или более раз (ленивый)

Регулярное выражение: ab*?

Входная строка: **aa**bc**a**bbb

Квантификаторы

X+ ==> один или более раз (жадный)

Регулярное выражение: ab+

Входная строка: a**ab**c**abbb**

X+? ==> один или более раз (ленивый)

Регулярное выражение: ab+?

Входная строка: a**ab**c**ab**bb

Квантификаторы

X{n} ==> ровно n раз (жадный)

или (совпадает по результату применения)

X{n}? ==> ровно n раз (ленивый)

Регулярное выражение: ab{2} или ab{2}?

Входная строка: aabc**abb**b

Квантификаторы

X{n,} ==> не менее n раз (жадный)

Регулярное выражение: ab{2,}

Входная строка: aabc**abbb**

X{n,}? ==> не менее n раз (ленивый)

Регулярное выражение: ab{2,}?

Входная строка: aabc**abb**b

Квантификаторы

X{n,m} ==> от n до m раз (жадный)

Регулярное выражение: ab{1,2}

Входная строка: a**ab**c**abb**b

X{n,m}? ==> от n до m раз (ленивый)

Регулярное выражение: ab{1,2}?

Входная строка: a**ab**c**ab**bb

Сверхжадные квантификаторы

При поиске в строке **aab** с помощью рег. выражения **a+b** шаги анализатора:

a+ ==> a (соответствует)

a+ ==> aa (соответствует)

a+ ==> aab (не соответствует)

откат назад (возврат b) к последнему соответствию (aa) и проверка a+b:

a+b ==> aab (соответствует)

Сверхжадные квантификаторы

Сверхжадный квантификатор действует как жадный, но никогда не откатывается назад.

a++ ==> a (соответствует)
a++ ==> aa (соответствует)
a++ ==> aab (не соответствует)

Последний символ ввода (b) прочтен, соответствие не найдено.

Сверхжадные квантификаторы

Чтобы сделать жадный квантификатор сверхжадным достаточно добавить + справа от квантификатора:

X? ==> X?+ X{n} ==> X{n}+

X* ==> X*+ X{n,} ==> X{n,}+

X+ ==> X++ X{n,m} ==> X{n,m}+

Сверхжадные квантификаторы работают как правило быстрее, чем жадные.

Логические операции

XY ==> X за которым следует Y (AND)

X|Y ==> X илиY (OR)

Приоритет AND выше чем OR.

Регулярное выражение: aa|b

Входная строка: **aa**b**b**ca**b**b**b**

Группы

Выражение в круглых скобка - группа.

Каждая группа имеет номер.

Группы нумеруются слева направо, начиная с единицы (номер может быть больше 9)

Чтобы группа не нумеровалась, она должна начинаться с **(?:**

Группы

(A)(B(C)(?:D))

(A) ==> группа номер 1
(B(C)(?:D)) ==> группа номер 2
(C) ==> группа номер 3
(?:D) ==> группа **без номера**

Группы

Группы могут быть использованы по номеру в регулярном выражении с помощью синтаксиса: \НОМЕР_ГРУППЫ

Регулярное выражение: (aab)\W\1

Входная строка: **aab aab**

Экранирование символов

Для представления специальных символов:

\ . | + * ?

[ ] ( ) { }

^ $

используют экранирование с помощью обратного слеша:

\\ \. \\ \+ * \?

\[\] \(\) \{ \}

\^ \$

Экранирование символов

Для указания диапазона экранирования можно использовать \Q и/или \E

\Q==> начало диапазона

\E ==> окончание диапазона

Регулярное выражение: \Q\(*\E(a)\1

Входная строка: ab**\(*aa**

Упреждающий просмотр вперед

Позитивный: (?=X)

Регулярное выражение: a(?=b)

Входная строка: **a**bac**a**b

Негативный: (?!X)

Регулярное выражение: a(?!b)

Входная строка: ab**a**cab

Просмотр назад

Позитивный: (?<=X)

Регулярное выражение: (?<=b)a

Входная строка: ab**a**cab

Негативный: (?<!X)

Регулярное выражение: (?<!b) a

Входная строка: **a**bac**a**b

Режимы

Влияют на работу регулярных выражений.

Каждый режим имеет буквенный код.

- COMMENTS ==> x
- CASE_INSENSITIVE ==> i
- UNIX_LINES ==> d
- DOTALL ==> s
- UNICODE_CASE ==> u
- MULTILINE ==> m

Режимы

Чтобы включить режим, достаточно предварить регулярное выражение комбинацией: (?КОД_РЕЖИМА).

(?m) (?s)

Если нужно включить сразу несколько режимов, можно писать несколько кодов:

(?iu)

Режимы

COMMENTS ==> x

Режим комментариев. Пробельные символы игнорируются, после символа # можно писать комментарий к рег. выражению.

Регулярное выражение: (?x)a bc #comment

Входная строка: **abc**ab

Режимы

CASE_INSENSITIVE ==> i

Игнорирует регистр символов.

UNIX_LINES ==> d

Разделитель строк только CR (\r)

DOTALL ==> s

Точка (.) может включать \n

Режимы

UNICODE_CASE ==> u

Игнорирует регистр символов.

MULTILINE ==> m

Многострочный режим (по умолчанию $ - конец ввода).