**Sử dụng regex trong java**

Regular Expression hay còn gọi là biểu thức chính quy được dùng để xử lý chuỗi nâng cao thông qua biểu thức riêng của nó, những biểu thức này sẽ có những nguyên tắc riêng và bạn phải tuân theo nguyên tắc đó thì biểu thức của bạn mới hoạt động được. Ngoài cái tên gọi Regular Expression ra thì nó còn có thể viết tắt thành Regex.

Nguyên tắc hoạt động của biểu thức Regex là so khớp dựa vào khuôn mẫu, khuôn mẫu được xây dựng từ các quy tắc căn bản của biểu thức Regex. Các ký tự thường dùng trong Regex:

. : đại diện cho 1 ký tự bất kỳ trừ ký tự xuống dòng \n.

\d : ký tự chữ số tương đương [0-9]

\D : ký tự ko phải chữ số

\s : ký tự khoảng trắng tương đương [ \f\n\r\t\v]

\S : ký tự không phải khoảng trắng tương đương [ ^\f\n\r\t\v]

\w : ký tự word (gồm chữ cái và chữ số, dấu gạch dưới \_ ) tương đương [a-zA-Z\_0-9]

\W : ký tự không phải ký tự word tương đương [^a-zA-Z\_0-9]

^ : bắt đầu 1 chuỗi hay 1 dòng

$ : kết thúc 1 chuỗi hay 1 dòng

\A : bắt đầu 1 chuỗi

\z : kết thúc 1 chuỗi

| : ký tự ngăn cách so trùng tương đương với phép or (lưu ý cái này nếu muốn kết hợp nhiều điều kiện)

[abc] : khớp với 1 ký tự nằm trong nhóm là a hay b hay c.

[a-z] so trùng với 1 ký tự nằm trong phạm vi a-z, dùng dấu - làm dấu ngăn cách.

[^abc] sẽ không so trùng với 1 ký tự nằm trong nhóm, ví dụ không so trùng với a hay b hay c.

() : Xác định 1 group (biểu thức con) xem như nó là một yếu tố đơn lẻ trong pattern .ví dụ ((a(b))c) sẽ khớp với b, ab, abc.

? : khớp với đứng trước từ 0 hay 1 lần. Ví dụ A?B sẽ khớp với B hay AB.

: khớp với đứng trước từ 0 lần trở lên . A\*B khớp với B, AB, AAB

: khớp với đứng trước từ 1 lần trở lên. A+B khớp với AB, AAB.

{n} : n là con số, Khớp đúng với n ký tự đúng trước nó . Ví dụ A{2}) khớp đúng với 2 chữ A.

{n, } : khớp đúng với n ký tự trở lên đứng trước nó , A{2,} khớp vói AA, AAA ...

{m,n} : khớp đùng với từ m->n ký tự đứng trước nó, A{2,4} khớp vói AA,AAA,AAAA

——————————————————————————————————————————

Username Validation Chúng ta cần kiểm tra chuối string input nhập vào, là uername hay không, nếu là uername nó cần thỏa mãn các điều kiện sau:

Có từ 3 - 5 kí tự

Bao gồm các kí tự thường a -> z

các chữ số 0 - 9 và một số kí tự đặc biệt: "\_", "-", "."

Pattern được sử dụng là: ^[a-z0-9.\_-]{3,15}$

^ : bắt đầu chuỗi

[a-z0-9.\_-] Check kí tự xuất hiện trong chuỗi là a-z, 0-9, \_, - hoặc .

{3,15} : có từ 3 - 15 kí tự

$ : kết thúc chuỗi

Password Complexity Validation Điều kiện để chuỗi đầu vào là password cần thỏa mãn các điều kiện sau:

Có độ dài từ 6 - 15 kí tự

Có ít nhất 1 kí tự thường, 1 kí tự viết hoa và 1 chữ số

Có 1 trong các kí tự đặc biệt sau (! # $ @ \_ + , ? . - )

Pattern được sử dụng là: ((?=.d)(?=.[a-z])(?=.[A-Z])(?=.[!.#$@\_+,?-]).{8,50})

( Start of group

(?=.\*d) must contains one digit from 0-9

(?=.\*[a-z]) must contains one lowercase characters

(?=.\*[A-Z]) must contains one uppercase characters

(?=.\*[!.#@\_+,?-]) must contains one special symbols in the list "!.#@\_+,?-" . match anything with previous condition checking

{8,50} length at least 8 characters and maximum of 50

) End of group

<https://regexr.com/>

https://regexone.com/lesson/capturing\_groups