Section 17: Regular Expressions

15 July 2021 19:23

Section 17: Regular Expressions

Regular expressions are way to describe a string or a pattern. Regular expressions are often used to search strings for a specific pattern or to validate that user input matched a specific pattern.

```
String alphanumeric = "abcDeeeF12Ghhiiiijkl99z";
System.out.println(alphanumeric.replaceAll(".","Y"));       // will replace all character of string with Y

System.out.println(alphanumeric.replaceAll("^abcDeee", "YYY"));    // will only replace starting occurance of abcDeee with YYY

System.out.println(alphanumeric.matches("^abcDeee"));    // will return false, return true only when whole string is matched and should be same

System.out.println(alphanumeric.replaceAll("ijkl99z$", "THE END"));    // will replace last of ijkl99z with THE END (opposite of ^)

System.out.println(alphanumeric.replaceAll("[aei]", "X"));    // all occurance of a, e, I will replace with X

System.out.println(alphanumeric.replaceAll("[aei][Fj]", "X"));    // going to perform a replacement if of the three letters A, E, I is actually followed by F or a  J

System.out.println(newAlphanumeric.replaceAll("[^ej]", "X"));    // will replace every letter with X except e and j

System.out.println(newAlphanumeric.replaceAll("[a-fA-F3-8]", "X"));    // can use - character to specify the range.

System.out.println(newAlphanumeric.replaceAll("(?i)[a-f3-8]", "X"));    // using (?i) we can turn of case sensitivity

System.out.println(newAlphanumeric.replaceAll("\\d", "X"));    // replace all digits with X

System.out.println(newAlphanumeric.replaceAll("\\D", "X"));    // replace all non-digits with X

System.out.println(hasWhitespace.replaceAll("\\s", ""));    // with remove all spaces, tabs and new line

System.out.println(hasWhitespace.replaceAll("\t", "X"));    // will replace all tabs

System.out.println(hasWhitespace.replaceAll("\\S", ""));    // will replace all non-whitespace characters

System.out.println(newAlphanumeric.replaceAll("\\w", "X"));    // will replace a-z, A-Z, 0-9 and _

System.out.println(hasWhitespace.replaceAll("\\b", "X"));    // each word will surrounded by X, which means hello string will be XhelloX.
```

Quantifiers : It specifies how often an element in a regular expression can occur.

Using "^abcDeee" we can use this "^abcDe{3}" or "^abcDe+" (followed with e) or "^abcDe*" (followed by any character after abcD) or "^abcDe{2,5}"

.replaceAll("h+i*j", "Y")) // replacing all occurrences of h followed by any number of I's followed by at least one j with Y.

Class Pattern :-
https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html

Using Matcher Method :-

```
 StringBuilder htmlText = new StringBuilder("<h1>My Heading</h1>");
htmlText.append("<h2>Sub-heading</h2>");
htmlText.append("<p>This is a paragraph about something.</p>");
htmlText.append("<p>This is another paragraph about something else.</p>");
htmlText.append("<h2>Summary</h2>");
htmlText.append("<p>Here is the summary.</p>");

String h2Pattern = ".*<h2>.*";  // . will match every character and star means zero or more
Pattern pattern = Pattern.compile(h2Pattern);
Matcher matcher = pattern.matcher(htmlText);
System.out.println(matcher.matches());  // will return true or false if pattern is exist or not
```

How to count number of occurrence:-

```
matcher.reset();        // matcher only can used once, so reset it and String h2Pattern = "<h2>";
      int count = 0;
      while(matcher.find()) {
         count++;
        }
```

Class Matcher :-
https://docs.oracle.com/javase/8/docs/api/java/util/regex/Matcher.html

Using Pattern and Matcher :-

```
String challenge11 = "{0, 2}, {0, 5}, {1, 3}, {2, 4}";
      Pattern pattern11 = Pattern.compile("\\{(.+?)\\}");
      Matcher matcher11 = pattern11.matcher(challenge11);
      while(matcher11.find()) {
         System.out.println("Occurrence: " + matcher11.group(1));
      }

String challenge12 = "11111";
String challenge13 = "11111-1111";
System.out.println(challenge12.matches("^\\d{5}(-\\d{4})?$"));
System.out.println(challenge13.matches("^\\d{5}(-\\d{4})?$"));
```