**Reg**ular **Ex**pressions or Regex (in short) in Java is an API for defining String patterns that can be used for searching, manipulating, and editing a string in Java. Email validation and passwords are a few areas of strings where Regex is widely used to define the constraints. Regular Expressions in Java are provided under **java.util.regex** package. This consists of **3 classes and 1 interface**. The **java.util.regex** package primarily consists of the following three classes as depicted below in tabular format as follows:

test12\_test@test.co.in

## **Regex Classes and Interfaces**

**Regex in Java provides 3 classes and 1 interface which are as follows:**

1. Pattern Class
2. Matcher Class
3. PatternSyntaxException Class
4. MatchResult Interface

| **S. No.** | **Class/Interface** | **Description** |
| --- | --- | --- |
| 1 | Pattern Class | Used for defining patterns |
| 2 | Matcher Class | Used for performing match operations on text using patterns |
| 3 | PatternSyntaxException Class | Used for indicating syntax error in a regular expression pattern |
| 4 | MatchResult Interface | Used for representing the result of a match operation |

**Methods that are often used with Regex:**

* Pattern.matches(String regex, CharSequence input) - It is used to compile the given regular expression and attempts to match the given input against it.

boolean java.util.regex.Pattern.matches(String regex, CharSequence input)

* String strVariable;

strVariable.matches(String regex)

* boolean java.lang.String.matches(String regex)
* String strVariable;

strVariable.replaceAll(String regex, String replacement)

String java.lang.String.replaceAll(String regex, String replacement)

| **Character Class** | **Description** |
| --- | --- |
| **[xyz]** | x,y or z |
| **[^xyz]** | Any characters other than x,y or z |
| **[a-zA-Z][0-9]** | characters from range a to z or A to Z and range 0 to 9 |
| **[a-f[m-t]]** | Union of a to f and m to t. |
| **[a-z && p-y]** | All the range of elements intersection between two ranges |
| **[a-z && [^bc]]** | a to z union with except b and c |
| **[a-z && [^m-p]]** | a to z union with except range m to p |

| **Metacharacter** | **Description** |
| --- | --- |
| | | Find a match for any one of the patterns separated by | as in: cat|dog|fish |
| . | Find just one instance of any character |
| ^ | Finds a match as the beginning of a string as in: ^Hello |
| $ | Finds a match at the end of the string as in: World$ |
| \d | Find a digit. Any digits, short of [0-9] |
| \s | Find a whitespace character |
| \b | Find a match at the beginning of a word like this: \bWORD, or at the end of a word like this: WORD\b |
| \uxxxx | Find the Unicode character specified by the hexadecimal number xxxx |
| \D | Any non-digit, short for [^0-9] |
| \S | Any non-whitespace character, short for [^\s] |
| \w | Any word character, short for [a-zA-Z\_0-9] |
| \W | Any non-word character, short for [^\w] |
| \b | A word boundary |
| \B | A non word boundary |
|  |  |
| **Quantifier** | **Description** |
| n+ | Matches any string that contains at least one n |
| n\* | Matches any string that contains zero or more occurrences of n |
| n? | Matches any string that contains zero or one occurrences of n |
| n{x} | Matches any string that contains a sequence of X n's |
| n{x,y} | Matches any string that contains a sequence of X to Y n's |
| n{x,} | Matches any string that contains a sequence of at least X n's |