Contents

[**Java Overview** 1](#_Toc509962806)

[**Regular Expression** 1](#_Toc509962807)

|  |
| --- |
| **Java Overview** |
| **Regular Expression** |
|  |
| Which pakage contains Regular Expressions related methods? |
| * A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern. * Java provides the java.util.regex package for pattern matching with regular expressions. * The java.util.regex package primarily consists of the following three classes −  1. Pattern Class –   Pattern class provides no public constructors.  To create a pattern, you must first invoke one of its public static compile() methods, which will then return a Pattern object.   1. Matcher Class –   Like the Pattern class, Matcher defines no public constructors. You obtain a Matcher object by invoking the matcher() method on a Pattern object.  A Matcher object is the engine that interprets the pattern and performs match operations against an input string.   1. PatternSyntaxException –   A PatternSyntaxException object is an unchecked exception that indicates a syntax error in a regular expression pattern. |
| List some of the Subexpression ? |
| |  |  | | --- | --- | | **Subexpression** | **Matches** | | **^** | **Matches the beginning of the line.** | | **$** | **Matches the end of the line.** | | **.** | **Matches any single character except newline. Using m option allows it to match the newline as well.** | | **[...]** | **Matches any single character in brackets.** | | **[^...]** | **Matches any single character not in brackets.** | | \A | Beginning of the entire string. | | \z | End of the entire string. | | \Z | End of the entire string except allowable final line terminator. | | **re\*** | **Matches 0 or more occurrences of the preceding expression.** | | **re+** | **Matches 1 or more of the previous thing.** | | **re?** | **Matches 0 or 1 occurrence of the preceding expression.** | | **re{ n}** | **Matches exactly n number of occurrences of the preceding expression.** | | **re{ n,}** | **Matches n or more occurrences of the preceding expression.** | | **re{ n, m}** | **Matches at least n and at most m occurrences of the preceding expression.** | | a| b | Matches either a or b. | | (re) | Groups regular expressions and remembers the matched text. | | (?: re) | Groups regular expressions without remembering the matched text. | | (?> re) | Matches the independent pattern without backtracking. | | **\w** | **Matches the word characters.** | | **\W** | **Matches the nonword characters.** | | **\s** | **Matches the whitespace. Equivalent to [\t\n\r\f].** | | **\S** | **Matches the nonwhitespace.** | | **\d** | **Matches the digits. Equivalent to [0-9].** | | **\D** | **Matches the nondigits.** | | \A | Matches the beginning of the string. | | \n | Back-reference to capture group number "n". | | \b | Matches the word boundaries when outside the brackets. Matches the backspace (0x08) when inside the brackets. | | \B | Matches the nonword boundaries. | | \n, \t, etc. | Matches newlines, carriage returns, tabs, etc. | | \Q | Escape (quote) all characters up to \E. | | \E | Ends quoting begun with \Q. | |
| Methods of the Matcher Class |
| 1. Index Methods : Index methods provide useful index values that show precisely where the match was found in the input string        1. Study Methods: Study methods review the input string and return a Boolean indicating whether or not the pattern is found          1. Replacement Methods :Replacement methods are useful methods for replacing text in an input string |
| List PatternSyntaxException Class Methods ? |
| * A PatternSyntaxException is an unchecked exception that indicates a syntax error in a regular expression pattern. * The PatternSyntaxException class provides the following methods to help you determine what went wrong |