Strings

Ionut
Spalatelu

Manipulating Strings

Method	Description
String **substring**(int beginIndex, int endIndex)	Returns a new string that is a substring of this string. The substring begins at the specified beginIndex and extends to the character at index endIndex - 1.
String **substring**(int beginIndex)	Returns a new string that is a substring of this string. The integer argument specifies the index of the first character. Here, the returned substring extends to the end of the original string.
String[] **split**(String regex)	Searches for a match as specified by the string argument (which contains a regular expression) and splits this string into an array of strings accordingly.
String **trim**()	Returns a copy of this string with leading and trailing white space removed.
String **toLowerCase**() String **toUpperCase**()	Returns a copy of this string converted to lowercase or uppercase. If no conversions are necessary, these methods return the original string.

Searching for Characters and Substrings

Method	Description
int **indexOf**(int ch) int **lastIndexOf**(int ch)	Returns the index of the first (last) occurrence of the specified character.
int **indexOf**(int ch, int fromIndex) int **lastIndexOf**(int ch, int fromIndex)	Returns the index of the first (last) occurrence of the specified character, searching forward (backward) from the specified index.
int **indexOf**(String str) int **lastIndexOf**(String str)	Returns the index of the first (last) occurrence of the specified substring.
int **indexOf**(String str, int fromIndex) int **lastIndexOf**(String str, int fromIndex)	Returns the index of the first (last) occurrence of the specified substring, searching forward (backward) from the specified index.
boolean **contains**(CharSequence s)	Returns true if the string contains the specified character sequence.

Replacing Characters and Substrings

Method	Description
String **replace**(char oldChar, char newChar)	Returns a new string resulting from replacing all occurrences of oldChar in this string with newChar.
String **replace**(CharSequence target, CharSequence replacement)	Replaces each substring of this string that matches the literal target sequence with the specified literal replacement sequence.
String **replaceAll**(String regex, String replacement)	Replaces each substring of this string that matches the given regular expression with the given replacement.
String **replaceFirst**(String regex, String replacement)	Replaces the first substring of this string that matches the given regular expression with the given replacement.

Comparing Strings and Portions of Strings

Method	Description
boolean **endsWith**(String suffix) boolean **startsWith**(String prefix)	Returns true if this string ends with or begins with the substring specified as an argument to the method.
boolean **startsWith**(String prefix, int offset)	Considers the string beginning at the index offset, and returns true if it begins with the substring specified as an argument.
int **compareTo**(String anotherString)	Compares two strings lexicographically. Returns an integer indicating whether this string is greater than (result is > 0), equal to (result is = 0), or less than (result is < 0) the argument.
int **compareToIgnoreCase**(String str)	Compares two strings lexicographically, ignoring differences in case. Returns an integer indicating whether this string is greater than (result is > 0), equal to (result is = 0), or less than (result is < 0) the argument.
boolean **equals**(Object anObject)	Returns true if and only if the argument is a String object that represents the same sequence of characters as this object.
boolean **equalsIgnoreCase**(String anotherString)	Returns true if and only if the argument is a String object that represents the same sequence of characters as this object, ignoring differences in case.
boolean **matches**(String regex)	Tests whether this string matches the specified regular expression.

StringBuilder

Method	Description
StringBuilder **append**(String s)	Appends the argument to this string builder. The data is converted to a string before the append operation takes place.
StringBuilder **delete**(int start, int end) StringBuilder **deleteCharAt**(int index)	The first method deletes the subsequence from start to end-1 (inclusive) in the StringBuilder's char sequence. The second method deletes the character located at index
StringBuilder **insert**(int offset, String s) void **setCharAt**(int index, char c)	Inserts the second argument into the string builder. The first integer argument indicates the index before which the data is to be inserted.
StringBuilder **reverse**()	Reverses the sequence of characters in this string builder

Note: You can use any String method on a StringBuilder object by first converting the string builder to a string with the toString() method of the StringBuilder class. Then convert the string back into a string builder using the StringBuilder(String str) constructor.

Bibliography

- https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/String.html