

JAVA.LANG.STRINGBUFFER CLASS

http://www.tutorialspoint.com/java/lang/java_lang_stringbuffer.htm

Copyright © tutorialspoint.com

Introduction

The **java.lang.StringBuffer** class is a thread-safe, mutable sequence of characters. Following are the important points about StringBuffer:

- A string buffer is like a String, but can be modified.
- It contains some particular sequence of characters, but the length and content of the sequence can be changed through certain method calls.
- They are safe for use by multiple threads.
- Every string buffer has a capacity.

Class declaration

Following is the declaration for **java.lang.StringBuffer** class:

```
public final class StringBuffer
    extends Object
    implements Serializable, CharSequence
```

Class constructors

S.N.	Constructor & Description
1	StringBuffer This constructs a string buffer with no characters in it and an initial capacity of 16 characters.
2	StringBufferCharSequenceseq This constructs a string buffer that contains the same characters as the specified CharSequence.
3	StringBufferintcapacity This constructs a string buffer with no characters in it and the specified initial capacity.
4	StringBufferStringstr This constructs a string buffer initialized to the contents of the specified string.

Class methods

S.N.	Method & Description
1	StringBuffer appendbooleanb

This method appends the string representation of the boolean argument to the sequence

2

[StringBuffer appendCharc](#)

This method appends the string representation of the char argument to this sequence.

3

[StringBuffer appendchar\[\]lstr](#)

This method appends the string representation of the char array argument to this sequence.

4

[StringBuffer appendchar\[\]lstr, intoffset, intlen](#)

This method appends the string representation of a subarray of the char array argument to this sequence.

5

[StringBuffer appendCharSequences](#)

This method appends the specified CharSequence to this sequence.

6

[StringBuffer appendCharSequences, intstart, intend](#)

This method appends a subsequence of the specified CharSequence to this sequence.

7

[StringBuffer appenddoubled](#)

This method appends the string representation of the double argument to this sequence.

8

[StringBuffer appendfloatf](#)

This method appends the string representation of the float argument to this sequence.

9

[StringBuffer appendinti](#)

This method appends the string representation of the int argument to this sequence.

10

[StringBuffer appendlonglng](#)

This method appends the string representation of the long argument to this sequence.

11

[StringBuffer appendObjectobj](#)

This method appends the string representation of the Object argument.

12

[StringBuffer appendStringstr](#)

This method appends the specified string to this character sequence.

13	<u>StringBuffer appendStringBuffersb</u>	This method appends the specified StringBuffer to this sequence.
14	<u>StringBuffer appendCodePointintcodePoint</u>	This method appends the string representation of the codePoint argument to this sequence.
15	<u>int capacity</u>	This method returns the current capacity.
16	<u>char charAtintindex</u>	This method returns the char value in this sequence at the specified index.
17	<u>int codePointAtintindex</u>	This method returns the character <i>Unicodecodepoint</i> at the specified index
18	<u>int codePointBeforeintindex</u>	This method returns the character <i>Unicodecodepoint</i> before the specified index
19	<u>int codePointCountintbeginIndex, intendIndex</u>	This method returns the number of Unicode code points in the specified text range of this sequence
20	<u>StringBuffer deleteintstart, intend</u>	This method removes the characters in a substring of this sequence.
21	<u>StringBuffer deleteCharAtintindex</u>	This method removes the char at the specified position in this sequence
22	<u>void ensureCapacityintminimumCapacity</u>	This method ensures that the capacity is at least equal to the specified minimum.
23	<u>void getCharsintsrcBegin, intsrcEnd, char[]dst, intdstBegin</u>	This method characters are copied from this sequence into the destination character array dst.
24		

[int indexOfStringstr](#)

This method returns the index within this string of the first occurrence of the specified substring.

25

[int indexOfStringstr, intfromIndex](#)

This method returns the index within this string of the first occurrence of the specified substring, starting at the specified index.

26

[StringBuffer insertintoffset, booleanb](#)

This method inserts the string representation of the boolean argument into this sequence.

27

[StringBuffer insertintoffset, charc](#)

This method inserts the string representation of the char argument into this sequence.

28

[StringBuffer insertintoffset, char\[\]str](#)

This method inserts the string representation of the char array argument into this sequence.

29

[StringBuffer insertintindex, char\[\]str, intoffset, intlen](#)

This method inserts the string representation of a subarray of the str array argument into this sequence.

30

[StringBuffer insertintdstOffset, CharSequences](#)

This method inserts the specified CharSequence into this sequence.

31

[StringBuffer insertintdstOffset, CharSequences, intstart, intend](#)

This method inserts a subsequence of the specified CharSequence into this sequence.

32

[StringBuffer insertintoffset, doubled](#)

This method inserts the string representation of the double argument into this sequence.

33

[StringBuffer insertintoffset, floatf](#)

This method inserts the string representation of the float argument into this sequence.

34

[StringBuffer insert\(int offset, int i](#)

This method inserts the string representation of the second int argument into this sequence.

35	<u>StringBuffer insertintoffset, longl</u>	This method inserts the string representation of the long argument into this sequence.
36	<u>StringBuffer insertintoffset, Objectobj</u>	This method inserts the string representation of the Object argument into this character sequence.
37	<u>StringBuffer insertintoffset, Stringstr</u>	This method inserts the string into this character sequence.
38	<u>int lastIndexOfStringstr</u>	This method returns the index within this string of the rightmost occurrence of the specified substring.
39	<u>int lastIndexOfStringstr, intfromIndex</u>	This method returns the index within this string of the last occurrence of the specified substring.
40	<u>int length</u>	This method returns the length <i>charactercount</i> .
41	<u>int offsetByCodePointsintindex, intcodePointOffset</u>	This method returns the index within this sequence that is offset from the given index by codePointOffset code points.
42	<u>StringBuffer replaceintstart, intend, Stringstr</u>	This method replaces the characters in a substring of this sequence with characters in the specified String.
43	<u>StringBuffer reverse</u>	This method causes this character sequence to be replaced by the reverse of the sequence.
44	<u>void setCharAtintindex, charch</u>	The character at the specified index is set to ch.
45	<u>void setLengthintnewLength</u>	This method sets the length of the character sequence.

- 46 [CharSequence subSequence](#)*intstart, intend*
This method returns a new character sequence that is a subsequence of this sequence.
- 47 [String substring](#)*intstart*
This method returns a new String that contains a subsequence of characters currently contained in this character sequence
- 48 [String substring](#)*intstart, intend*
This method returns a new String that contains a subsequence of characters currently contained in this sequence.
- 49 [String toString](#)
This method returns a string representing the data in this sequence.
- 50 [void trimToSize](#)
This method attempts to reduce storage used for the character sequence.

Methods inherited

This class inherits methods from the following classes:

• [java.lang.Object](#)

Processing math: 100%