Class java.lang.StringBuffer

```
java.lang.Object
   |
   +----java.lang.StringBuffer
```

---

public class **StringBuffer**
extends Object

This Class is a growable buffer for characters. It is mainly used to create Strings. The compiler uses it to implement the "+" operator. For example:

```
"a" + 4 + "c"
```

is compiled to:

```
new StringBuffer().append("a").append(4).append("c").toString()
```

Note that the method toString() does not create a copy of the internal buffer. Instead the buffer is marked as shared. Any further changes to the buffer will cause a copy to be made.

See Also:
String, ByteArrayOutputStream
Version:
1.21, 08/11/95
Author:
Arthur van Hoff

Constructor Index

o **StringBuffer**()
Constructs an empty String buffer.
o **StringBuffer**(int)
Constructs an empty String buffer with the specified initial length.
o **StringBuffer**(String)
Constructs a String buffer with the specified initial value.

Method Index

o **append**(Object)
Appends an object to the end of this buffer.
o **append**(String)
Appends a String to the end of this buffer.
o **append**(char[])
Appends an array of characters to the end of this buffer.
o **append**(char[], int, int)
Appends a part of an array of characters to the end of this buffer.
o **append**(boolean)
Appends a boolean to the end of this buffer.
o **append**(char)
Appends a character to the end of this buffer.
o **append**(int)
Appends an integer to the end of this buffer.
o **append**(long)
Appends a long to the end of this buffer.
o **append**(float)
Appends a float to the end of this buffer.
o **append**(double)
Appends a double to the end of this buffer.
o **capacity**()
Returns the current capacity of the String buffer.
o **charAt**(int)
Returns the character at the specified index.
o **copyWhenShared**()
Copies the buffer value if it is shared.
o **ensureCapacity**(int)
Ensures that the capacity of the buffer is at least equal to the specified minimum.
o **getChars**(int, int, char[], int)
Copies the characters of the specified substring (determined by srcBegin and srcEnd) into the character array, starting at the array's dstBegin location.
o **insert**(int, Object)
Inserts an object into the String buffer.
o **insert**(int, String)
Inserts a String into the String buffer.
o **insert**(int, char[])
Inserts an array of characters into the String buffer.
o **insert**(int, boolean)
Inserts a boolean into the String buffer.
o **insert**(int, char)
Inserts a character into the String buffer.
o **insert**(int, int)
Inserts an integer into the String buffer.
o **insert**(int, long)
Inserts a long into the String buffer.
o **insert**(int, float)
Inserts a float into the String buffer.

o **insert**(int, double)
Inserts a double into the String buffer.
o **length**()
Returns the length (character count) of the buffer.
o **setCharAt**(int, char)
Changes the character at the specified index to be ch.
o **setLength**(int)
Sets the length of the String.
o **toString**()
Converts to a String representing the data in the buffer.

Constructors

o **StringBuffer**

```
public StringBuffer()
```

Constructs an empty String buffer.

o **StringBuffer**

```
public StringBuffer(int length)
```

Constructs an empty String buffer with the specified initial length.
Parameters:
length – the initial length

o **StringBuffer**

```
public StringBuffer(String str)
```

Constructs a String buffer with the specified initial value.
Parameters:
str – the initial value of the buffer

Methods

o **length**

```
public int length()
```

Returns the length (character count) of the buffer.

o **capacity**

```
public int capacity()
```

Returns the current capacity of the String buffer. The capacity is the amount of storage available for newly inserted characters; beyond which an allocation will occur.

o **copyWhenShared**

```
public void copyWhenShared()
```

Copies the buffer value if it is shared.

o **ensureCapacity**

```
public synchronized void ensureCapacity(int minimumCapacity)
```

Ensures that the capacity of the buffer is at least equal to the specified minimum.
Parameters:
minimumCapacity – the minimum desired capacity

o **setLength**

```
public synchronized void setLength(int newLength)
```

Sets the length of the String. If the length is reduced, characters are lost. If the length is extended, the values of the new characters are set to 0.
Parameters:
newLength – the new length of the buffer
Throws: StringIndexOutOfBoundsException
If the length is invalid.

o **charAt**

```
public synchronized char charAt(int index)
```

Returns the character at the specified index. An index ranges from 0..length()–1.
Parameters:
index – the index of the desired character
Throws: StringIndexOutOfBoundsException
If the index is invalid.

o **getChars**

```
public synchronized void getChars(int srcBegin,
                                  int srcEnd,
                                  char dst[],
                                  int dstBegin)
```

Copies the characters of the specified substring (determined by srcBegin and srcEnd) into the character array, starting at the array's dstBegin location. Both srcBegin and srcEnd must be legal indexes into the buffer.
Parameters:

srcBegin – begin copy at this offset in the String
srcEnd – stop copying at this offset in the String
dst – the array to copy the data into
dstBegin – offset into dst

Throws: StringIndexOutOfBoundsException
If there is an invalid index into the buffer.

o **setCharAt**

```
public synchronized void setCharAt(int index,
                                   char ch)
```

Changes the character at the specified index to be ch.

Parameters:
index – the index of the character
ch – the new character

Throws: ArrayIndexOutOfBoundsException
If the index is invalid.

o **append**

```
public synchronized StringBuffer append(Object obj)
```

Appends an object to the end of this buffer.

Parameters:
obj – the object to be appended

Returns:
the StringBuffer itself, NOT a new one.

o **append**

```
public synchronized StringBuffer append(String str)
```

Appends a String to the end of this buffer.

Parameters:
str – the String to be appended

Returns:
the StringBuffer itself, NOT a new one.

o **append**

```
public synchronized StringBuffer append(char str[])
```

Appends an array of characters to the end of this buffer.

Parameters:
str – the characters to be appended

Returns:
the StringBuffer itself, NOT a new one.

o **append**

```
public synchronized StringBuffer append(char str[],
                                        int offset,
                                        int len)
```

Appends a part of an array of characters to the end of this buffer.

Parameters:

str – the characters to be appended

offset – where to start

len – the number of characters to add

Returns:

the StringBuffer itself, NOT a new one.

o **append**

```
public StringBuffer append(boolean b)
```

Appends a boolean to the end of this buffer.

Parameters:

b – the boolean to be appended

Returns:

the StringBuffer itself, NOT a new one.

o **append**

```
public synchronized StringBuffer append(char c)
```

Appends a character to the end of this buffer.

Parameters:

ch – the character to be appended

Returns:

the StringBuffer itself, NOT a new one.

o **append**

```
public StringBuffer append(int i)
```

Appends an integer to the end of this buffer.

Parameters:

i – the integer to be appended

Returns:

the StringBuffer itself, NOT a new one.

o **append**

```
public StringBuffer append(long l)
```

Appends a long to the end of this buffer.

Parameters:
l – the long to be appended
Returns:
the StringBuffer itself, NOT a new one.

o **append**

```
public StringBuffer append(float f)
```

Appends a float to the end of this buffer.
Parameters:
f – the float to be appended
Returns:
the StringBuffer itself, NOT a new one.

o **append**

```
public StringBuffer append(double d)
```

Appends a double to the end of this buffer.
Parameters:
d – the double to be appended
Returns:
the StringBuffer itself, NOT a new one.

o **insert**

```
public synchronized StringBuffer insert(int offset,
                                        Object obj)
```

Inserts an object into the String buffer.
Parameters:
offset – the offset at which to insert
obj – the object to insert
Returns:
the StringBuffer itself, NOT a new one.
Throws: ArrayIndexOutOfBoundsException
If the offset is invalid.

o **insert**

```
public synchronized StringBuffer insert(int offset,
                                        String str)
```

Inserts a String into the String buffer.
Parameters:
offset – the offset at which to insert
str – the String to insert
Returns:

the StringBuffer itself, NOT a new one.

Throws: ArrayIndexOutOfBoundsException

If the offset is invalid.

o **insert**

```
public synchronized StringBuffer insert(int offset,
                                        char str[])
```

Inserts an array of characters into the String buffer.

Parameters:

offset – the offset at which to insert

str – the characters to insert

Returns:

the StringBuffer itself, NOT a new one.

Throws: ArrayIndexOutOfBoundsException

If the offset is invalid.

o **insert**

```
public StringBuffer insert(int offset,
                           boolean b)
```

Inserts a boolean into the String buffer.

Parameters:

offset – the offset at which to insert

b – the boolean to insert

Returns:

the StringBuffer itself, NOT a new one.

Throws: ArrayIndexOutOfBoundsException

If the offset is invalid.

o **insert**

```
public synchronized StringBuffer insert(int offset,
                                        char c)
```

Inserts a character into the String buffer.

Parameters:

offset – the offset at which to insert

ch – the character to insert

Returns:

the StringBuffer itself, NOT a new one.

Throws: ArrayIndexOutOfBoundsException

If the offset invalid.

o **insert**

```
public StringBuffer insert(int offset,
                           int i)
```

Inserts an integer into the String buffer.

Parameters:

offset – the offset at which to insert

i – the integer to insert

Returns:

the StringBuffer itself, NOT a new one.

Throws: ArrayIndexOutOfBoundsException

If the offset is invalid.

o **insert**

```
public StringBuffer insert(int offset,
                           long l)
```

Inserts a long into the String buffer.

Parameters:

offset – the offset at which to insert

l – the long to insert

Returns:

the StringBuffer itself, NOT a new one.

Throws: ArrayIndexOutOfBoundsException

If the offset is invalid.

o **insert**

```
public StringBuffer insert(int offset,
                           float f)
```

Inserts a float into the String buffer.

Parameters:

offset – the offset at which to insert

f – the float to insert

Returns:

the StringBuffer itself, NOT a new one.

Throws: ArrayIndexOutOfBoundsException

If the offset is invalid.

o **insert**

```
public StringBuffer insert(int offset,
                           double d)
```

Inserts a double into the String buffer.

Parameters:

offset – the offset at which to insert

d – the double to insert

Returns:

the StringBuffer itself, NOT a new one.

Throws: ArrayIndexOutOfBoundsException

If the offset is invalid.

o **toString**

```
public synchronized String toString()
```

Converts to a String representing the data in the buffer.

Overrides:

toString in class Object