**StringBuffer in java**

The string created by String class always immutable in java and We have learned the methods on the immutable string. In this post, we will learn the StringBuffer in java, mutable string by StringBuffer, and StringBuffer methods in java.

Here is the table content of the article will we will cover this topic.

1. StringBuilder class in java

2. How to create a string by StringBuilder

3. Some important constructors

4. Important methods of StringBuilder

**StringBuffer class in java**

1.The StringBuffer class is another way to create a string in java. The StringBuffer class exists in the java.lang package. Serializable, Comparable, and CharSequence are three interfaces that are implemented by the StringBuffer class.

2.Like StringBuilder in java, StringBuffer in java is also used to create a mutable string. A mutable string can be modified without creating a new object. StringBuffer is the same as the StringBuilder class but the StringBuffer is thread-safe.

How to create a string by StringBuffer

The StringBuffer class provides some constructors that are used to create the object of StringBuffer. Each and every constructor creates an object of StringBuffer that holds the string value. Some constructor takes arguments of a different type but each one creates a mutable string. So we can modify the object after creation, Let’s take an example and create a string by use of the default constructor of StringBuffer.

public class StringBufferExample

{

public static void main(String args[])

{

StringBuffer name = new StringBuffer();

System.out.println("Is it blank string: "+ (name.length() == 0));

name.append("Hello");

System.out.println("Value after append: "+name);

}

}

Output: Is it blank string: true

Value after append: Hello

**Some important constructors**

**1. StringBuffer():**

This constructor creates an empty string with initial capacity 16. We can modify the object later.

**2. StringBuffer(String str):**

This constructor creates a string buffer object with some specified value.

**3. StringBuffer(int capacity):**

This constructor creates an empty buffer with a specified capacity.

**1. StringBuffer()**

* This constructor creates an empty string buffer.
* This constructor doesn’t have any parameters.
* Its initials capacity is 16 which is provided by JVM at the time of object creation of StringBuffer.

StringBuffer name = new StringBuffer();

class ExampleOfStringBuffer

{

public static void main(String args[])

{

// creating a blank string

StringBuffer name = new StringBuffer();

// If name is blank its length is equals to 0

System.out.println("Is name is blank = "+ (name.length() == 0));

// Display the capacity of name

System.out.println(name.capacity());

}

}

Output: Is name is blank = true

16

Memory representation:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

**StringBuffer(String str)**

* This constructor creates a string buffer with the specified string.
* This constructor takes one parameter of string type.
* Its initials capacity is 16 which is provided by JVM at the time of object creation of StringBuffer.

StringBuffer name = new StringBuffer(specifiedString);

class ExampleOfStringBuffer

{

public static void main(String args[])

{

// creating a StringBuffer with specified string

StringBuffer name = new StringBuffer("Ravi");

System.out.println("Name of Student = "+ name);

System.out.println(name.capacity()); // Display the capacity of name

}

}

Output: Name of Student = Ravi

20

Memory representation:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| R | a | v | i |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |

**3. StringBuffer(int capacity)**

* This constructor creates a blank string buffer with a specified capacity.
* This constructor takes one parameter of integer type.
* If you want to specify the capacity of string Buffer you should use this constructor.

StringBuffer name = new StringBuffer(int capacity);

class ExampleOfStringBuffer

{

public static void main(String args[])

{

// creating a StringBuffer with specified capacity

StringBuffer name = new StringBuffer(10);

name.append("Ravi"); // Appedning string in string buffer

System.out.println("Name Of Student = "+ name);

System.out.println(name.capacity()); // Display the capacity of name

}

}

Output: Is name is Student = Ravi

10

Memory representation:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| R | a | v | I |  |  |  |  |  |  |  |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

**Important methods of StringBuilder**

**1. append() method:**

This method is used to append the given text in a string. StringBuffer class has various forms of this method we will discuss it later.

class ExampleOfStringBuffer

{

public static void main(String args[])

{

// creating a StringBuffer with specified string

StringBuffer name = new StringBuffer("Ravi");

name.append("kant"); // Appending string in string buffer

System.out.println("Name of Student = "+ name);

}}

Output: Name of Student = Ravikant

**2. insert() method:**

This method is used to insert the given text in the string at a given position. StringBuffer class has various forms of this method we will discuss it later.

class ExampleOfStringBuffer

{

public static void main(String args[])

{// creating a StringBuffer with specified string

StringBuffer name = new StringBuffer("Ravi");// inserting string in string buffer

name.insert(4, "kant");

System.out.println("Name of Student = "+ name);}}

Output: Name of Student = Ravikan

**3. replace(startIndex, endIndex, string) method:**

This method is used to replace the string by given text. The replace() replaces the given string from the specified startIndex and endIndex.

class ExampleOfStringBuffer

{

public static void main(String args[])

{

// creating a StringBuffer with specified string

StringBuffer name = new StringBuffer("Ravi kant");

name.replace(5, 7, "OK");

System.out.println("Name of Student = "+ name);

}

}