StringBuilder in java

**As we already know String class is used to create the immutable string in java. But we can mutate the String object by using StringBuilder class**

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

**StringBuilder class in java**

1.We can create a mutable string by use of the StringBuilder class. The StringBuilder class is placed in java. lang package. Serializable, Comparable, and CharSequence are three interfaces that are implemented by the StringBuilder class.

2.StringBuilder class is used to create a mutable string, by means of a mutable string, we can modify the string without creating a new object. StringBuilder class is the same as the StringBuffer class except for thread-safe. Actually, the StringBuilder is not synchronized, so it is not threaded safe.

How to create a string by StringBuilder

* To create a string by StringBuilder class we need to understand the Constructor of StringBuilder.
* Each constructor creates an object of StringBuilder that holds the string.
* After the creation of the string, we can modify it because it is a mutable string.

Let’s take an example and create a string by use of the default constructor of StringBuilder.

public class StringBuilderExample

{

public static void main(String args[])

{

StringBuilder s1= new StringBuilder();

System.out.println(“is it blank string:+(S1.length() = = 0)”)

S1.append("Hello");

System.out.println("Value after append: "+S1);

}

}

Output: Is it blank string: true

Value after append: Hello

In the above example, we are creating an object of StringBuilder by default constructor that is S1. After that, we are appending the string in S1 and printing the string.

**Some important constructors**

**1. StringBuilder():**

It is used to declare an empty string builder. Its initial capacity is 16.

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

It is used to declare a string builder with some specified value.

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

It is used to declare an empty string builder with a specified capacity.

**1. StringBuilder ()**

* It is a constructor that is used to create an empty string builder object.
* This constructor doesn’t have any parameters.
* Its initials capacity is 16 which is provided by JVM at the time of object creation of StringBuilder.

**StringBuilder name = new StringBuilder();**

class Example Of StringBuilder

{

public static void main(String args[])

{

// creating a blank string

StringBuilder name = new StringBuilder();

// 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 |

**2. StringBuilder(String str)**

* It is a constructor that is used to create a string builder object with a **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 StringBuilder.

StringBuilder name = new StringBuilder(specifiedString);

class Example Of StringBuilder

{

public static void main(String args[])

{

// creating a StringBuilder with specified string

StringBuilder name = new StringBuilder ("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. StringBuilder(int capacity)**

* It is a constructor that is used to create a blank string builder object with a specified capacity.
* This constructor takes one parameter of integer type.
* If you want to specify the capacity of string Builder you should use this constructor.

**StringBuilder name=new StringBuilder(int capacity);**

class ExampleOfStringBuilder

{

public static void main(String args[])

{

// creating a StringBuilder with specified capacity

StringBuilder name = new StringBuilder (10);

name.append("Ravi"); // Appending 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/join the given text in a string. StringBuilder class has various forms of this method we will discuss it later.

class ExampleOfStringBuilder

{

public static void main(String args[])

{// creating a StringBuilder with specified string

StringBuilder name = new StringBuilder("Ravi");

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

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. StringBuilder class has various forms of this method we will discuss it later.

class ExampleOfStringBuilder

{

public static void main(String args[])

{

// creating a StringBuilder with specified string

StringBuilder name = new StringBuilder("Ravi");

// inserting string in string buffer

name.insert(4, "kant");

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

}

}

Output: Name of Student = Ravikant

**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 ExampleOfStringBuilder

{

public static void main(String args[])

{

// creating a StringBuilder with specified string

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

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

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

}

}