Course Name - Object Oriented Programming using Java

Lecture 17– Methods of StringBuffer Class (Contd.)

Presented By
Dr. Sudipta Sahana
Asso. Prof.
Dept. of CSE
UEM - Kolkata

Topic of Interest

- ensureCapacity()
- getChars()
- indexOf()
- insert()
- length()
- setCharAt()
- setLength()
- substring()
- toString()

ensureCapacity()

This method ensures that the capacity is at least equal to the specified minimum. If the current capacity is less than the argument, then a new internal array is allocated with greater capacity. The new capacity is the larger of −

1. The minimumCapacity argument.
2. Twice the old capacity, plus 2.

If the minimumCapacity argument is nonpositive, this method takes no action and simply returns.

Example of ensureCapacity()

```
public class StringBufferDemo {
   public static void main(String[] args) {
      StringBuffer buff1 = new StringBuffer("tuts point");
      System.out.println("buffer1 = " + buff1);
      System.out.println("Old Capacity = " + buff1.capacity());
      buff1.ensureCapacity(32);
      System.out.println("New Capacity = " + buff1.capacity());
      StringBuffer buff2 = new StringBuffer("compile online");
      System.out.println("buffer2 = " + buff2);
      System.out.println("Old Capacity = " + buff2.capacity());
      buff2.ensureCapacity(28);
      System.out.println("New Capacity = " + buff2.capacity());
   }
}
```

Output

```
buffer1 = tuts point
Old Capacity = 26
New Capacity = 54
buffer2 = compile online
Old Capacity = 30
New Capacity = 30
```

getChars()

The java string getChars() method copies the content of this string into specified char array. There are 4 arguments passed in getChars() method. The signature of getChars() method is given below:

public void getChars(int srcBeginIndex, int srcEndIndex, char[] destination, int dstBeginIndex)

```
public class StringBufferDemo {
   public static void main(String[] args) {
      StringBuffer buff = new StringBuffer("java
programming");
      System.out.println("buffer = " + buff);
 char[] chArr = new char[]{'t','u','t','o','r','i','a','l','s'};
      buff.getChars(5, 9, chArr, 3);
      System.out.println(chArr);
   }
}
```

Output

buffer = java programming
tutprogls

indexOf()

The java.lang.StringBuffer.indexOf(String str, int fromIndex) method returns the index within this string of the first occurrence of the specified substring, starting at the specified index. The **fromIndex** argument is the index from which to start the search

```
public class StringBufferDemo {
   public static void main(String[] args) {
      StringBuffer buff = new StringBuffer("programming
language");
      System.out.println("buffer = " + buff);
      System.out.println("Index of substring = " +
buff.indexOf("age"));
 System.out.println("Index of substring = " +
buff.indexOf("am",2));
 System.out.println("Index of substring = " +
buff.indexOf("am",10));
   }
}
```

Output

```
buffer = programming language
Index of substring = 17
Index of substring = 5
Index of substring = -1
```

insert()

This method is used to insert either string or character or integer or real constant or boolean value at a specific index value of the given string.

```
class StringHandling
{
public static void main(String arg[])
{
StringBuffer sb=new StringBuffer("this
is my java code");
System.out.println(sb.insert(11, "first "));
}
}
```

Output

this is my first java code

length()

This method returns the length (character count) of the sequence of characters currently represented by this object.

```
public class StringBufferDemo {

   public static void main(String[] args) {

      StringBuffer buff = new StringBuffer("Tutorials");
 System.out.println("length = " + buff.length());

      buff = new StringBuffer("");

      System.out.println("length = " + buff.length());
   }
}
```

Output

```
length = 9
length = 0
```

setCharAt()

This method sets the character at the specified **index** to **ch**.This sequence is altered to represent a new character sequence that is identical to the old character sequence, except that it contains the character **ch** at position **index**.

```
public class StringBufferDemo {

   public static void main(String[] args) {

StringBuffer buff = new StringBuffer("AMIT");
System.out.println("buffer = " + buff);
System.out.println("character at index 3 = " +
buff.charAt(3));
buff.setCharAt(3, 'L');
System.out.println("After Set, buffer = " + buff);
System.out.println("character at index 3 = " +
buff.charAt(3));
   }
}
```

Output

```
buffer = AMIT
character at index 3 = T
After Set, buffer = AMIL
character at index 3 = L
```

setLength()

This method sets the length of the character sequence. The sequence is changed to a new character sequence whose length is specified by the argument.

```
public class StringBufferDemo {

   public static void main(String[] args) {

      StringBuffer buff = new StringBuffer("tutorials");
      System.out.println("buffer1 = " + buff);
      System.out.println("length = " + buff.length());
      buff.setLength(5);
      System.out.println("buffer2 = " + buff);
      System.out.println("length = " + buff.length());
   }
}
```

Output

```
buffer1 = tutorials
length = 9
buffer2 = tutor
length = 5
```

substring()

The java.lang.StringBuffer.substring(int start) method returns a new String that contains a subsequence of characters currently contained in this character sequence. The substring begins at the specified index, **start** and extends to the end of this sequence.

```
public class StringBufferDemo {

   public static void main(String[] args) {

      StringBuffer buff = new
StringBuffer("tutorials");
      System.out.println("buffer = " + buff);

      // prints substring from index 2
      System.out.println("substring = "+
buff.substring(2));
   }
}
```

Output

```
buffer = tutorials
substring = torials
```

toString()

This method is used to convert mutable string values into immutable string.

```
class StringHandling
{
public static void main(String arg[])
{
StringBuffer sb=new StringBuffer("java");
String s=sb.toString();
System.out.println(s);
s.concat("code");
System.out.println(s);
}
}
```

Output

```
java
java
```

Thank You