StringBuffer Class

StringBuffer class is a mutable class unlike the String class which is immutable. Both the capacity and character string of a StringBuffer Class. StringBuffer can be changed dynamically. String buffers are preferred when heavy modification of character strings is involved (appending, inserting, deleting, modifying etc).

Strings can be obtained from string buffers. Since the StringBuffer class does not override the equals() method from the Object class, contents of string buffers should be converted to String objects for string comparison.

A StringIndexOutOfBoundsException is thrown if an index is not valid when using wrong index in String Buffer manipulations

Creation of StringBuffers

StringBuffer Constructors

```
public class StringBufferDemo {


        public static void main(String[] args) {

                //        Examples of Creation of Strings

                StringBuffer strBuf1 = new StringBuffer("Bob");

                StringBuffer strBuf2 = new StringBuffer(100); //With capacity 100

                StringBuffer strBuf3 = new StringBuffer(); //Default Capacity 16

                System.out.println("strBuf1 : " + strBuf1);

                System.out.println("strBuf2 capacity : " + strBuf2.capacity());

                System.out.println("strBuf3 capacity : " + strBuf3.capacity());

        }

}
```

Download StringBufferDemo.java

Output

strBuf1 : Bob
strBuf2 capacity : 100
strBuf3 capacity : 16

StringBuffer Functions

The following program explains the usage of the some of the basic StringBuffer methods like :

1.**capacity**()
Returns the current capacity of the String buffer.

2. **length**()
Returns the length (character count) of this string buffer.

3.**charAt**(int index)
The specified character of the sequence currently represented by the string buffer, as indicated by the index argument, is returned.

4.setCharAt(int index, char ch)
The character at the specified index of this string buffer is set to ch

5. **toString**()
Converts to a string representing the data in this string buffer

6.**insert**(int offset, char c)
Inserts the string representation of the char argument into this string buffer.
Note that the StringBuffer class has got many overloaded 'insert' methods which can be used based on the application need.

7. **delete**(int start, int end)
Removes the characters in a substring of this StringBuffer

8. **replace**(int start, int end, String str)
Replaces the characters in a substring of this StringBuffer with characters in the specified String.

9.**reverse**()
The character sequence contained in this string buffer is replaced by the reverse of the sequence.

10. **append**(String str)
Appends the string to this string buffer.
Note that the StringBuffer class has got many overloaded 'append' methods which can be used based on the application need.

11. **setLength**(intnewLength)
Sets the length of this String buffer.

```
public class StringBufferFunctionsDemo {

    public static void main(String[] args) {
        //      Examples of Creation of Strings
        StringBuffer strBuf1 = new StringBuffer("Bobby");
        StringBuffer strBuf2 = new StringBuffer(100); //With capacity 100
        StringBuffer strBuf3 = new StringBuffer(); //Default Capacity 16
        System.out.println("strBuf1 : " + strBuf1);
        System.out.println("strBuf1 capacity : " + strBuf1.capacity());
        System.out.println("strBuf2 capacity : " + strBuf2.capacity());
        System.out.println("strBuf3 capacity : " + strBuf3.capacity());
        System.out.println("strBuf1 length : " + strBuf1.length());
        System.out.println("strBuf1 charAt 2 : " + strBuf1.charAt(2));
        //      A StringIndexOutOfBoundsException is thrown if the index is not valid.
        strBuf1.setCharAt(1, 't');
        System.out.println("strBuf1 after setCharAt 1 to t is : "
                + strBuf1);
        System.out
                .println("strBuf1 toString() is : " + strBuf1.toString());
        strBuf3.append("beginner-java-tutorial");
        System.out.println("strBuf3 when appended with a String : "
                + strBuf3.toString());
        strBuf3.insert(1, 'c');
        System.out.println("strBuf3 when c is inserted at 1 : "
                + strBuf3.toString());
```

```
        strBuf3.delete(1, 'c');

        System.out.println("strBuf3 when c is deleted at 1 : "
                + strBuf3.toString());

        strBuf3.reverse();

        System.out.println("Reversed strBuf3 : " + strBuf3);

        strBuf2.setLength(5);

        strBuf2.append("jdbc-tutorial");

        System.out.println("strBuf2 : " + strBuf2);

        //      We can clear a StringBuffer using the following line

        strBuf2.setLength(0);

        System.out.println("strBuf2 when cleared using setLength(0): "
                + strBuf2);

    }

}
```

DownloadStringBufferFunctionsDemo.java

Output

```
strBuf1 : Bobby
strBuf1 capacity : 21
strBuf2 capacity : 100
strBuf3 capacity : 16
strBuf1 length : 5
strBuf1 charAt 2 : b
strBuf1 after setCharAt 1 to t is : Btbby
strBuf1 toString() is : Btbby
strBuf3 when appended with a String : beginner-java-tutorial
strBuf3 when c is inserted at 1 : bceginner-java-tutorial
strBuf3 when c is deleted at 1 : b
Reversed strBuf3 : b
strBuf2 :
```