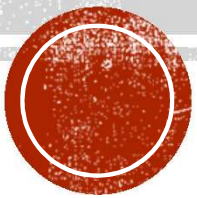
STRINGBUFFER

A peer class of String



STRING CLASS IN JAVA

- StringBuffer is a peer class of String that provides much of the functionality of strings. String represents fixed-length, immutable character sequences while StringBuffer represents growable and writable character sequences.

- StringBuffer may have characters and substrings inserted in the middle or appended to the end. It will automatically grow to make room for such additions and often has more characters preallocated than are actually needed, to allow room for growth.

STRINGBUFFER CONSTRUCTORS

- StringBuffer( ): It reserves room for 16 characters without reallocation.
- StringBuffer(int size): It accepts an integer argument that explicitly sets the size of the buffer.
- StringBuffer(String str): It accepts a String argument that sets the initial contents of the StringBuffer object and reserves room for 16 more characters without reallocation.

length( ) and capacity( ): The length of a StringBuffer can be found by the length( ) method, while the total allocated capacity can be found by the capacity( ) method.

METHODS

```
public class Driver{
    public static void main(String args[]) {
        StringBuffer s = new StringBuffer("CSEstudents");
        int p = s.length();
        int q = s.capacity();
        System.out.println("Length of string CSEstudents=" + p);
        System.out.println("Capacity of string CSEstudents=" + q);
    }
}
```

```
Length of string CSEstudents=11
Capacity of string CSEstudents=27
```

append( ): It is used to add text at the end of the existence text. Here are a few of its forms:

- StringBuffer append(String str)
- StringBuffer append(int num)

METHODS

```
public class Driver{
    public static void main(String args[]) {
        StringBuffer s = new StringBuffer("CSE");
        s.append("students");
        System.out.println(s); // returns CSEstudents
        s.append(1);
        System.out.println(s); // returns CSEstudents1
    }
}
```

```
CSEstudents
CSEstudents1
```

METHODS

insert( ): It is used to insert text at the specified index position. These are a few of its forms:

- StringBuffer insert(int index, String str)
- StringBuffer insert(int index, char ch)

```
public class Driver{
    public static void main(String args[]) {
        StringBuffer s = new StringBuffer("CSEStudents");

        s.insert(3, "Amrita");
        System.out.println(s);  //CSEAmritaStudents
        s.insert(0, 3);
        System.out.println(s);  //3CSEAmritaStudents
        s.insert(4, true);
        System.out.println(s);  //3CSEtrueAmritaStudents
        s.insert(8, 31.25d);
        System.out.println(s);  //3CSEtrue31.25AmritaStudents
        s.insert(4, 21.45f);
        System.out.println(s);  //3CSE21.45true31.25AmritaStudents

        char am_ar[] = { 'c', 'l', 'a', 's', 's' };
        s.insert(1, am_ar);
        System.out.println(s); //3classCSE21.45true31.25AmritaStudents
    }
}
```

```
CSEAmritaStudents
3CSEAmritaStudents
3CSEtrueAmritaStudents
3CSEtrue31.25AmritaStudents
3CSE21.45true31.25AmritaStudents
3classCSE21.45true31.25AmritaStudents
```

reverse( ): It can reverse the characters within a StringBuffer object using reverse( ).This method returns the reversed object on which it was called.

```
public class Driver{
    public static void main(String args[]) {
        StringBuffer s = new StringBuffer("CSEStudents");
        s.reverse();
        System.out.println(s);
    }
}
```

stnedutSESC

METHODS

delete( ) and deleteCharAt( ): The delete( ) method deletes a sequence of characters from the invoking object. The substring deleted runs from start Index to endIndex–1. The deleteCharAt( ) method deletes the character at the index specified by loc.

- StringBuffer delete(int startIndex, int endIndex)
- StringBuffer deleteCharAt(int loc)

```
public class Driver{
    public static void main(String args[]) {
        StringBuffer s = new StringBuffer("CSEStudents");
        s.delete(0, 3);
        System.out.println(s); //Students
        s.deleteCharAt(7);
        System.out.println(s); //Student
    }
}
```

```
Students
Student
```

METHODS

replace( ): The substring being replaced is specified by the indexes start Index and endIndex. Thus, the substring at start Index through endIndex–1 is replaced.

StringBuffer replace(int startIndex, int endIndex, String str)

```
public class Driver{
    public static void main(String args[]) {
        StringBuffer s = new StringBuffer("CSEAmritaStudents");
        s.replace(3, 9, " Class");
        System.out.println(s);
    }
}
```

```
CSE ClassStudents
```

METHODS

char charAt(int index): This method returns the char value in this sequence at the specified index.

public char charAt(int index)

void getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin): In this method, the characters are copied from this sequence into the destination character array dst.

public void getChars(int srcBegin,

int srcEnd,

char[] dst,

int dstBegin)

METHODS

int indexOf(String str): This method returns the index within this string of the first occurrence of the specified substring.

- public int indexOf(String str)
- public int indexOf(String str, int fromIndex)

int lastIndexOf(String str): This method returns the index within this string of the last occurrence of the specified substring.

- public int lastIndexOf(String str)
- public int lastIndexOf(String str, int fromIndex)

METHODS

METHODS

void setCharAt(int index, char ch): In this method, the character at the specified index is set to ch.

public void setCharAt(int index, char ch)

void setLength(int newLength): This method sets the length of the character sequence.

public void setLength(int newLength)

String substring(int start): This method returns a new String that contains a subsequence of characters currently contained in this character sequence.

- public String substring(int start)
- public String substring(int start, int end)