String Buffer.

*) In case of String when we are creating a object and if we want to change the object then the object is not changed rather it will create a new object.

e.g. String s = "Kiit"

s.concat("Hello");

s → Kiit

Kiit Hello

*) This is called as immutability.

*) But in case of String Buffer we can change an object. So object of String buffer is mutable.

StringBuffer sb = new StringBuffer("Kiit");

sb.append("Hello");

sb → Kiit Hello

*) If data is frequently changing like calculator, notepad etc use StringBuffer.

Syntax

public final class StringBuffer extends AbstractStringBuilder implements java.io.Serializable, CharSequence

{

4 Constructors

StringBuffer() {--} No Arg.

StringBuffer (CharSequence cs) {--}

" " (String str) {--}

" " (int capacity) {--}

}

Methods

public synchronized int length() {--}
int capacity() {--}

insert();
reverse();
delete();
deleteCharAt();
replace();

charAt();
indexOf();
SubString();
SubSequence();
toString();

String Builder

*) In StringBuffer all methods are Synchronized that means no two threads can execute a Synchronized method Simultaneously which increase waiting time of a thread.

*) In StringBuilder class all methods are nonSynchronized and mutable.