Why we use StringBuffer Object?

.....

if the content is not fixed and keep on changing then string is not

recommoneded because for every changes a new object will be created

,which creates memory and performance Problems.

To overcome this Problem we should go for StringBuffer where all

changes will be performed in the existing object instead of creating new Object.

StringBuffer class

.....

In Java StringBuffer class is used to create

mutable (modifiable) string Object. The StringBuffer

class in java is same as String class except it is mutable i.e. it can

be changeble.

THE StringBuffer class is present in java.lang package.

>it is also final class like String.

>StringBuffer object is created by using new keyword only.

Constructor:-

.....

1- StringBuffer sb=new StringBuffer();

.....

Create an Eempty StringBuffer Object with default intial Capacity (16).

But Once Stringbuffrer reaches its max capacity then a new Stringbuffer Object is created with new

capacity.

newcapacity=(oldcapacity+1)*2

But in case of String there is no changes so length() is equal to the capacity.

Note:--

String concept is implemented by array Concept where StringBuffer is implemented

by Collections Concept.

ex>-

```
StringBuffer sb=new StringBuffer();
        sop(sb.capacity());  //16

        sb.append("a...p");
        sop(sb.capacity());  //16

        sb.appened("q");
        sop(sb.capacity);  //34
```

2-StringBuffer sb=new StringBuffer(int intialCapacity)

.....

creates an empty stringBuffer object with the specified capacity.

3-StringBuffer sb=new StringBuffer(String s)

.....

creates an equavalent StringBuffer for the given String object with capacity=s.length()+16

ex.

```
StringBuffer sb=new StringBuffer("btps");
        sop(sb.capacity()) ;  //4+16=20
```

important methods of StringBuffer class

.....

1-public int length()

2-public int capacity()

.....

3-public char charAt(int index)

4-public void setCharAt(int index,char ch)

.....

To replace the character locating at specified index with provided character.

(like replace)

5-public StringBuffer append(String s) Only add at LAST

public StringBuffer append(int i)

public StringBuffer append(float f)

public StringBuffer append(double d)

public StringBuffer append(boolean b)

6-public StringBuffer insert(int index,String s)

.....

By using this method we can insert a String at perticlular index.

we have some overloaded methods.

public StringBuffer insert(int index,int i)

public StringBuffer insert(int index,float f)

public StringBuffer insert(int index,double d)

public StringBuffer insert(int index,boolean b)

EX.

```
StringBuffer sb=new StringBuffer("abcdefgh");
        sb.insert(2,"xyz");
        sop(sb);
```

7-public StringBuffer delete(int begin,int end)

.....

To delete the characters from begin index to (end-1) index.

8-public StringBuffer deleteCharAt(int index)

9-public StringBuffer reverse()

10-public void setLength(int length)

.....

ex>-

```
    StringBuffer sb=new
StringBuffer("btpsindia");
            sb.setLength(4);   //btps
```

11-public void ensureCapacity(int minimumCapacity)

.....

TO increase capacity on run time based on our requirment with existing object.

ex.>- StringBuffer sb=new StringBuffer();

sop(sb.capacity()); //16

sb.ensureCapacity(500); 500

sop(sb.capacity());

12-public void trimToSize()

.....

ex>- StringBuffer sb=new StringBuffer(1000);

sb.append("xyz");

sop(sb.capacity()); //1000

sb.trimToSize();

sop(sb.capacity()); //3

Note:-All the methods present inside StringBuffer class are decared as Synchronized.

Hence At a a time only one thread is allowed to operate on StringBuffer object.

IT increase waiting time of thread and creates Performance Problem.

--To overcome this problem sun pepole introduced

StringBuilder Concept in java.

StringBuilder Object in java

.....

IT is exactly same as StringBufffer (including methods and constructors).

This class is Introduced in version 1.5

But there are some IMP differences..

1-All methods present in inside StringBuffer are Synchronized but in StringBuilder No methods is declared as Synchronized.

2-In StringBuffer At a time only one thread is allowed to operate on StringBuffer object

where in StringBuilder Object Multiple thread can operatre .

3-IN StringBuffer Object Performance is Low where as Performance is high in StringBuilder.

4-StringBuffer is Introduced in Version 1.0 v where as StringBuilder is Introduced in 1.5 v

Differnce Between String, StringBuffer, StringBuilder

.....

1-If the Conetent is fixed and won't change frequently then we can go for String object

if the content is not fixed and we change frquently but the thread safty is rquired, then we can

for StringBuffer.

if the content is not fixed and we change frquently but thread safty is not required then we

can go for Stringbuilder Object.

method Chaining

.....

For most of the methods in String,Stringbuffer,StringBuilder,the return type are same type only.

Hence after applying a method on the result we can call another method which fomrs method chaining

Ex.-

```
StringBuffer sb=new StringBuffer();

sb.append("btps").reverse().append("India").ins
ert(2,"kanpur");
```

In method chaining all the methods will be excecuted from left ro right..