String Buffer

and

String Builder

String Buffer

1. Java StringBuffer class is used to create mutable (modifiable) String objects.
2. The StringBuffer class in Java is the same as String class except it is mutable i.e. it can be changed.

String vs String Buffer

The string represents **fixed-length, immutable character** sequences
StringBuffer represents **growable and writable character** sequences.

```
public class Main
{
    public static void main(String[] args)
         {
        String str="study";
        str.concat("night");
        System.out.println(str); //Output: study

        StringBuffer str1 = new StringBuffer("study");
        str1.append("night");
        System.out.println(str1); // Output: studynight
         }
}
```

Constructors of StringBuffer Class

Constructor	Description
StringBuffer()	It creates an empty String buffer with the initial capacity of 16.
StringBuffer(String str)	It creates a String buffer with the specified string..
StringBuffer(int capacity)	It creates an empty String buffer with the specified capacity as length.

Constructors of StringBuilder Class

Constructor	Description
StringBuilder()	It creates an empty String Builder with the initial capacity of 16.
StringBuilder(String str)	It creates a String Builder with the specified string.
StringBuilder(int length)	It creates an empty String Builder with the specified capacity as length.

Methods of StringBuffer & StringBuilder Class

Method	Description
append(String s)	It is used to append the specified string with this string. The append() method is overloaded like append(char), append(boolean), append(int), append(float), append(double) etc.
insert(int offset, String s)	It is used to insert the specified string with this string at the specified position. The insert() method is overloaded like insert(int, char), insert(int, boolean), insert(int, int), insert(int, float), insert(int, double) etc.
replace(int startIndex, int endIndex, String str)	It is used to replace the string from specified startIndex and endIndex.
delete(int startIndex, int endIndex)	It is used to delete the string from specified startIndex and endIndex.
reverse()	is used to reverse the string.
capacity()	It is used to return the current capacity.
ensureCapacity(int minimumCapacity)	It is used to ensure the capacity at least equal to the given minimum.
charAt(int index)	It is used to return the character at the specified position.
length()	It is used to return the length of the string i.e. total number of characters.
substring(int beginIndex)	It is used to return the substring from the specified beginIndex.
substring(int beginIndex, int endIndex)	It is used to return the substring from the specified beginIndex and endIndex.

append()

Concatenates the given argument with this String.

```
public class Main
{
    public static void main(String[] args)
        {
            StringBuffer sb=new StringBuffer("Hello ");
            sb.append("Java");//now original string is changed
            System.out.println(sb);//prints Hello Java
        }
}
```

insert()

Inserts the given String with this string at the given position.

```
public class Main
{
    public static void main(String[] args)
        {
            StringBuffer sb=new StringBuffer("Hello ");
            sb.insert(1,"Java");//now original string is changed
            System.out.println(sb);//prints HJavaello
        }
}
```

replace()

Replaces the given String from the specified beginIndex and endIndex.

```
public class Main
{
    public static void main(String[] args)
        {
            StringBuffer sb=new StringBuffer("Hello ");
            sb.replace(1,3,"Java");
            System.out.println(sb);//prints HJavalo
        }
}
```

delete()

Deletes the String from the specified beginIndex to endIndex.

```
public class Main
{
    public static void main(String[] args)
        {
            StringBuffer sb=new StringBuffer("Hello ");
            sb.delete(1,3);
            System.out.println(sb);//prints Hlo
        }
}
```

reverse()

Reverses the current String.

```
public class Main
{
    public static void main(String[] args)
        {
            StringBuffer sb=new StringBuffer("Hello ");
            sb.reverse();
            System.out.println(sb);//prints olleH
        }
}
```

capacity()

Returns the current capacity of the buffer. The default capacity of the buffer is 16.
If the number of character increases from its current capacity, it increases the capacity by (oldcapacity*2)+2.

```
public class Main
{
    public static void main(String[] args)
         {
        StringBuffer sb=new StringBuffer();
        System.out.println(sb.capacity());//default 16
        sb.append("Hello");
        System.out.println(sb.capacity());//now 16
        sb.append("java is my favourite language");
    System.out.println(sb.capacity());//now (16*2)+2=34 i.e (oldcapacity*2)+2
         }
}
```

ensureCapacity()

Ensures that the given capacity is the minimum to the current capacity.
If it is greater than the current capacity, it increases the capacity by (oldcapacity*2)+2

```
public class Main
{
    public static void main(String[] args)
         {
        StringBuffer sb=new StringBuffer();
        System.out.println(sb.capacity());//default 16
        sb.append("Hello");
        System.out.println(sb.capacity());//now 16
        sb.append("java is my favourite language");
    System.out.println(sb.capacity());//now (16*2)+2=34 i.e(oldcapacity*2)+2
        sb.ensureCapacity(10);//now no change
        System.out.println(sb.capacity());//now 34
        sb.ensureCapacity(50);//now (34*2)+2
        System.out.println(sb.capacity());//now 70
         }
}
```