

Course Name: A Level (2nd Sem)

Subject: JAVA

Topic: StringBuilder class

Date: 03-04-20

StringBuilder class

Java StringBuilder class is also used to create mutable (modifiable) string object. StringBuilder is same as StringBuffer except for one important difference. StringBuilder is not synchronized, which means it is not thread safe.

Constructors of StringBuilder class

- **StringBuilder ()** : Constructs a string builder with no characters in it and an initial capacity of 16 characters.
- **StringBuilder (int capacity)** : Constructs a string builder with no characters in it and an initial capacity specified by the capacity argument.
- **StringBuilder (String str)** : Constructs a string builder initialized to the contents of the specified string. The initial capacity of the string builder is 16 plus the length of the string argument.

methods of StringBuilder class

1)append()

The append() method concatenates the given argument(string representation) to the end of the invoking StringBuilder object.

```
class StringBuilder1
{
    public static void main(String args[])
    {
        StringBuilder s1=new StringBuilder("quarantine ");
        S1.append("isolation");
        System.out.println(s1);
    }
}
```

2)insert()

The insert() method inserts the given argument(string representation) into the invoking StringBuilder object at the given position.

```
class StringBuilder2
{
    public static void main(String args[])
    {
        StringBuilder s1=new StringBuilder("breathing difficulties ");
        S1.insert(1,"fever");
        System.out.println(s1);
    }
}
```

3)replace()

The replace() method replaces the string from specified start index to the end index.

```
class StringBuilder3
{
    public static void main(String args[])
    {
        StringBuilder s1=new StringBuilder("corona");
        S1.replace(1,2,"SARS");
        System.out.println(s1);
    }
}
```

4)reverse()

This method reverses the characters within a StringBuilder object.

```
class StringBuilder4
{
    public static void main(String args[])
    {
        StringBuilder s1=new StringBuilder("CORONA");
        s1.reverse();
        System.out.println(s1);
    }
}
```

5)delete()

The delete() method of StringBuilder class deletes the string from the specified beginIndex to endIndex.

```
class StringBuilder5
{
    public static void main(String args[])
    {
        StringBuilder s1=new StringBuilder("covid19");
        s1.delete(1,3);
        System.out.println(s1);
    }
}
```

6)capacity()

The capacity() method returns the current capacity of StringBuilder object. The capacity is the amount of storage available for newly inserted characters, beyond which an allocation will occur.

```
class StringBuilder6
{
    public static void main(String args[])
    {
        StringBuilder strBuilder=new StringBuilder();
        System.out.println(strBuilder.capacity());
        strBuilder.append("corona");
        System.out.println(strBuilder.capacity());
        strBuilder.append("coronaviruscovid19");
        System.out.println(strBuilder.capacity());
        strBuilder.append("1");
        System.out.println(strBuilder.capacity()); //(oldcapacity*2)+2
        StringBuilder strBuilder2=new StringBuilder("cccccccc");
        System.out.println(strBuilder2.capacity());
    }
}
```

Difference between StringBuffer and StringBuilder

| StringBuffer | StringBuilder |
|---|---|
| StringBuffer is synchronized i.e. thread safe. | StringBuilder is non-synchronized i.e. not thread safe. |
| StringBuffer is less efficient and slower than StringBuilder as StringBuffer is synchronized. | StringBuilder is more efficient and faster than StringBuffer. |
| StringBuffer is old, its there in JDK from very first release. | StringBuilder is introduced much later in release of JDK 1.5 |