Java StringBuilder class

StringBuilder is identical to StringBuffer except for one important difference that it is not synchronized, which means it is not thread safe.

StringBuilder also used for creating string object that is mutable and non synchronized. The StringBuilder class provides no guarantee of synchronization. StringBuffer and StringBuilder both are mutable but if synchronization is not required then it is recommend to use StringBuilder class.

This class is located into java.lang package

StringBuilder Constructors

1. StringBuilder (): creates an empty StringBuilder and reserves room for 16 characters.

2. StringBuilder (int size): create an empty string and takes an integer argument to set capacity of the buffer.

3. StringBuilder (String str): create a StringBuilder object and initialize it with string str.

4. StringBuilder (CharSequence seq): It creates stringbuilder object by using CharSequence interface.

- **Creating a StringBuilder Class**

Let's use StringBuilder class to create string object and check its mutability also.

```
public class Demo {
  public static void main(String[] args) {
    StringBuilder sb = new StringBuilder("study");
    System.out.println(sb);
    // modifying object
    sb.append("today");
    System.out.println(sb);
```

```
 }
}
```

O/P

study

studytoday

- Difference between StringBuffer and StringBuilder class

StringBuffer class	**StringBuilder class**
StringBuffer is synchronized.	StringBuilder is not synchronized.
Because of synchronisation, StringBuffer operation is slower than StringBuilder.	StringBuilder operates faster.
StringBuffer is thread-safe	StringBuilder is not thread-safe
StringBuffer is less efficient as compare to StringBuilder	StringBuilder is more efficient as compared to StringBuffer.
Its storage area is in the heap	Its storage area is the stack
It is mutable	It is mutable
Methods are synchronized	Methods are not synchronized

It is alternative of string class	It is more flexible as compared to the string class
Introduced in Java 1.0	Introduced in Java 1.5
Its performance is moderate	Its performance is very high

- **StringBuilder Methods**

StringBuilder class has various methods to handle string object, such as append, insert, replace, reverse etc. Lets see usage of these with the help of examples.

Example of StringBuilder append string

In this example, we are appending a new string using appen() method to the existing string object.

```
public class Demo {
  public static void main(String[] args) {
    StringBuilder sb = new StringBuilder("study");
    System.out.println(sb);
    //appending object
    sb.append("together");
    System.out.println(sb);
  }
}
```

Output

```
study
```

- **StringBuilder Replace Method**

It is used to replace a substring from the string object. This method takes three arguments, first is start index, second is last index and third is substring to be replaced.

```
public class Demo {
  public static void main(String[] args) {
    StringBuilder sb = new StringBuilder("Java is a programming language");
    System.out.println(sb);
    // replacing object
    sb.replace( 10, 21, "computer");
    System.out.println(sb);
  }
}
```

Output

Java is a programming language

Java is a computer language

- **StringBuilder Reverse Method**

It is used to reverse the string object. It completely reverses the string from start to end characters. See the below example.

```
public class Demo {
  public static void main(String[] args) {
    StringBuilder sb = new StringBuilder("Java stringbuilder");
    System.out.println(sb);
    // reverse object
```

```
    sb.reverse();
    System.out.println(sb);
  }
}
```

Output

Java stringbuilder

redliubgnirts avaJ