**Java StringBuffer和StringBuilder类**

当对字符串进行修改的时候，需要使用StringBuffer和StringBuilder类。

和String类不同的是，StringBuffer和StringBuilder类的对象能够被多次的修改，并且不产生新的未使用对象。

StringBuilder类在Java 5中被提出，它和StringBuffer之间的最大不同在于StringBuilder的方法不是线程安全的（不能同步访问）。

由于StringBuilder相较于StringBuffer有速度优势，所以多数情况下建议使用StringBuilder类。然而在应用程序要求线程安全的情况下，则必须使用StringBuffer类。

**实例**

public class Test{

public static void main(String args[]){

StringBuffer sBuffer = new StringBuffer(" test");

sBuffer.append(" String Buffer");

System.out.println(sBuffer);

}

}

以上实例编译运行结果如下：

test String Buffer

**StringBuffer 方法**

以下是StringBuffer类支持的主要方法：

|  |  |
| --- | --- |
| **序号** | **方法描述** |
| 1 | public StringBuffer append(String s) 将指定的字符串追加到此字符序列。 |
| 2 | public StringBuffer reverse()  将此字符序列用其反转形式取代。 |
| 3 | public delete(int start, int end) 移除此序列的子字符串中的字符。 |
| 4 | public insert(int offset, int i) 将 int 参数的字符串表示形式插入此序列中。 |
| 5 | replace(int start, int end, String str) 使用给定 String 中的字符替换此序列的子字符串中的字符。 |

下面的列表里的方法和String类的方法类似：

|  |  |
| --- | --- |
| 序号 | 方法描述 |
| 1 | int capacity() 返回当前容量。 |
| 2 | char charAt(int index) 返回此序列中指定索引处的 char 值。 |
| 3 | void ensureCapacity(int minimumCapacity) 确保容量至少等于指定的最小值。 |
| 4 | void getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin) 将字符从此序列复制到目标字符数组 dst。 |
| 5 | int indexOf(String str) 返回第一次出现的指定子字符串在该字符串中的索引。 |
| 6 | int indexOf(String str, int fromIndex) 从指定的索引处开始，返回第一次出现的指定子字符串在该字符串中的索引。 |
| 7 | int lastIndexOf(String str) 返回最右边出现的指定子字符串在此字符串中的索引。 |
| 8 | int lastIndexOf(String str, int fromIndex) 返回最后一次出现的指定子字符串在此字符串中的索引。 |
| 9 | int length()  返回长度（字符数）。 |
| 10 | void setCharAt(int index, char ch) 将给定索引处的字符设置为 ch。 |
| 11 | void setLength(int newLength) 设置字符序列的长度。 |
| 12 | CharSequence subSequence(int start, int end) 返回一个新的字符序列，该字符序列是此序列的子序列。 |
| 13 | String substring(int start) 返回一个新的 String，它包含此字符序列当前所包含的字符子序列。 |
| 14 | String substring(int start, int end) 返回一个新的 String，它包含此序列当前所包含的字符子序列。 |
| 15 | String toString() 返回此序列中数据的字符串表示形式。 |