

StringBuffer 始于 JDK 1.0, StringBuilder 始于 JDK 5.0; 但都继承自 AbstractStringBuilder。 (AbstractStringBuilder的实现原理为: AbstractStringBuilder中采用一个 char数组 来保存需要append的字符串, char数组有一个初始大小, 当append的字符串长度超过当前char数组容量时, 则对char数组进行动态扩展, 即重新申请一段更大的内存空间, 然后将当前char数组拷贝到新的位置, 因为重新分配内存并拷贝的开销比较大, 所以每次重新申请内存空间都是采用申请大于当前需要的内存空间的方式, 这里是 2 倍。)

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有速度优势, 但其方法不是线程安全的 (线程安全就是多线程访问时, 采用了加锁机制, 当一个线程访问该类的某个数据时, 进行保护, 其他线程不能进行访问直到该线程读取完, 其他线程才可使用。不会出现数据不一致或者数据污染。线程不安全就是不提供数据访问保护, 有可能出现多个线程先后更改数据造成所得到的数据是脏数据)。

但在某些特别情况下, String 对象的字符串拼接可以直接被JVM 在编译期确定下来, 这时, StringBuilder 在速度上就不占任何优势了。

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 <code>int</code> 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 <code>String</code> 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 <code>char</code> 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 <code>dst</code> 。
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 <code>ch</code> 。
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 <code>String</code> ，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 <code>String</code> ，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。

