

[面试题]String、StringBuffer、StringBuilder 有什么区别?

作者: [Mmmrkr](#)

原文链接: <https://ld246.com/article/1573630962998>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



[面试题]String、StringBuffer、StringBuilder有什么区别？

这是一道经典的面试题，我在一些面试题目的文章中都能看到，所以希望在此总结一下。

先说结论

线程安全性

- 线程安全：String、StringBuffer
- 非线程安全：StringBuilder

执行效率

- StringBuffer > String > StringBuilder

存储空间

- String的值不可变，每次对String的操作都会生成新的String对象，耗费大量内存空间，从而引起GC
- StringBuffer和StringBuilder都是可变的

使用场景

- 操作少量的数据用String
- 单线程操作大量数据用StringBuilder
- 多线程操作大量数据用StringBuffer

以上内容来源于[程序汪公众号]

如果只是针对面试题，那上面这些答案应该已经足够了，但是我想深入研究一这上面的细节，希望可以对你我都有所帮助。

String类

什么是String类？

- String 类是典型的 `Immutable`类，其自身声明为final class，代表它的不可继承。
- String类本质上是个char数组，而且用private和final关键字修饰。为了避免我们通过对数组的操作者更粗暴的反射直接修改数组，String的所有方法都没有去动数组里的元素。

由于它的不可变性，类似拼接、裁剪字符串等动作，都会产生新的String队形。String操作的效率往对应用性能有明显影响。

String类的特性是什么？

- 由于String类的不可变性，所以说String类是线程安全的
- 也由于String类的不可变性，每次对String类的操作都会产生新的String对象，耗费大量空间，影效率

StringBuffer类

什么是StringBuffer类？

- 为了解决String类拼接产生太多中间对象的问题，所以在Java 1.5版本中增加了StringBuffer类
- 可以通过StringBuffer类的append或者add方法，把字符创添加到已有序列的末尾或者指定位置

StringBuffer类的特性是什么？

- StringBuffer是线程安全的可修改字符序列，因为append方法都用synchronized关键字修饰
- 也由于它的线程安全，带来了额外的性能开销

除非有线程安全的需要，不然还是使用它的继承者，也就是StringBuilder

----在极客时间的Java核心技术36讲栏目中是这样定性的

StringBuilder类

什么是StringBuilder类？

- StringBuilder类在能力上和StringBuffer没有本质区别，但是它去掉了线程安全的部分，有效减少开销

StringBuilder类的特性是什么？

- StringBuilder是非线程安全的
- 因为减少了开销，所以它的执行效率也最高

StringBuilder类是绝大部分情况下进行字符串拼接的首选

----同样在极客时间的Java核心技术36讲栏目中这样提到

题外话

- StringBuffer类和StringBuilder类的初始数组大小是16，至于为什么是16在这里不作讨论
- 可能有人会说我一直都是String一路走到黑，也不见有什么问题啊，关于这个问题我们可以通过下的代码解释，若果我们通过JDK8编译再反编译的话，可以看到，字符串拼接操作会自动被 javac 转换StringBuilder 操作。而在

JDK 9 里面则是因为 Java 9 为了更加统一字符串操作优化，提供了 StringConcatFactory，作为一统一的入口。

```
String myStr = "aa" + "bb" + "cc" + "dd";
```