

String、StringBuilder、StringBuffer

1. 运行速度。运行速度方面，StringBuilder>StringBuffer>String

String 最慢的原因是：

String 为字符串常量，StringBuilder和StringBuffer为字符串变量，即String对象创建之后该对象是不可更改的，但后两者是变量，是可以更改的。

a. 三者都是在jdk中都被定义为final类，这意味着他们不可以继承。

b. String最常见，与StringBuffer相比，String的性能比较差，因为对String类型进行改变的时候都会生成一个新的String对象，这在字符串拼接操作时候很明显，因为内容经常改变的字符串不应该使用String，如果不考虑多线程，应该使用StringBuilder。

```
1 String str="abc";
2 System.out.println(str);
3 str=str+"de";
4 System.out.println(str);
```

如果运行这段代码会发现先输出"abc"，然后又输出"abcde"，好像是str这个对象被更改了，其实，这只是一一种错觉罢了，JVM对于这几行代码是这样处理的，首先创建一个String对象str，并把"abc"赋值给str，然后在第三行中，其实JVM又创建了一个新的对象也名为str，然后再把原来的str的值和"de"加起来再赋值给新的str，而原来的str就会被JVM的垃圾回收机制（GC）给回收掉了，所以，str实际上并没有被更改，也就是前面说的String对象一旦创建之后就不可更改了。所以，Java中对String对象进行的操作实际上是一个不断创建新的对象并且将旧的对象回收的一个过程，所以执行速度很慢。

c. StringBuffer生成一个对象后，在进行字符串拼接操作时，调用append方法即可，不会产生新的对象，仅仅是对本身进行操作，性能比String更高，另外StringBuffer是线程安全的，因此适合在多线程中使用，正因为如此，在单线程中，使用StringBuffer需要进行锁的控制，速度会比StringBuilder相比会比较慢。

在线程安全上，StringBuilder是线程不安全的，而StringBuffer是线程安全的

如果一个StringBuffer对象在字符串缓冲区被多个线程使用时，StringBuffer中很多方法可以带有synchronized关键字，但不能保证线程安全，有可能会出现一些错误的操作。所以如果要进行的操作是多线程的，那么就要使用StringBuffer，但是在单

3. 总结一下

String：适用于少量的字符串操作的情况

StringBuilder：适用于单线程下在字符串缓冲区进行大量操作的情况

StringBuffer：适用多线程下在字符串缓冲区进行大量操作的情况