链滴

# String 是不可变的原因

作者：2457081614
原文链接：https://ld246.com/article/1578311547986
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)

# 概述

在Java中，String是不可变的，由final进行修饰，但是String为什么是不可变的呢？本文从内存、同和数据结构的角度进行阐述。

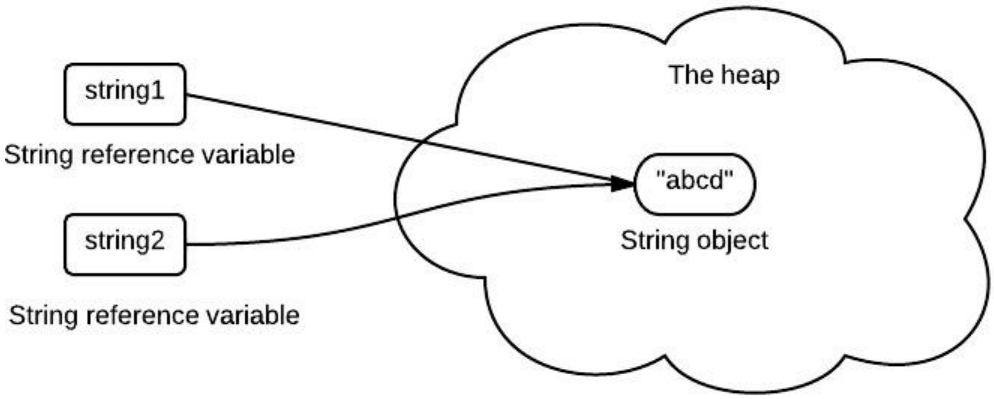
# 原因

## 1.字符串池的需要

字符串池是方法区一个重要的数据区域，下面这个代码只会在字符串池创建一个'abcd'对象。

```
String string1 = "abcd";
String string2 = "abcd";
```

具体引用如下图：




原文链接：String 是不可变的原因

如果string是可变的，那么一个引用更改字符串值就会导致另一个引用的值错误。

## 2.缓存 Hashcode

在 Java 中经常使用字符串的 hashcode。 例如，在 HashMap 或 HashSet 中。 作为不可变代码可保证散列代码总是相同的，这样就不担心更改的情况下改变它。 这意味着，不需要每次使用 hashcod 时都计算它。 这样更有效率。

## 3.促进其他对象的使用

```
HashSet<String> set = new HashSet<String>();
set.add(new String("a"));
set.add(new String("b"));
set.add(new String("c"));

for(String a: set)
    a.value = "a";
```

在这个例子中，如果 String 是可变的，那么它的值可以被修改，这将违反 set 的设计(set 包含不重的元素)。 对于诅咒，上面的例子只是为了演示，在一个真正的字符串类中没有value字段。

## 4.安全

字符串被广泛用作许多 java 类的参数，例如网络连接、打开文件等。 如果 String 不是不可变的，则接或文件将被更改，这可能会导致严重的安全威胁。 该方法认为它是连接到一台机器，但没有。 可字符串也可能在反射中导致安全问题，因为参数是字符串。

## 5.线程安全

String是不可变的，所以它是线程安全的。

原文链接：String 是不可变的原因