如何优化JAVA程序开发，提高JAVA性能? PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022__E5_A6_82_E4_BD_95_E4_BC_98_E5_c104_144603.htm 通过使用一些辅助性工具来找到程序中的瓶颈，然后就可以对瓶颈部分的代码进行优化。一般有两种方案：即优化代码或更改设计方法。我们一般会选择后者，因为不去调用以下代码要比调用一些优化的代码更能提高程序的性能。而一个设计良好的程序能够精简代码，从而提高性能。下面将提供一些在JAVA程序的设计和编码中，为了能够提高JAVA程序的性能，而经常采用的一些方法和技巧。1．对象的生成和大小的调整。JAVA程序设计中一个普遍的问题就是没有好好的利用JAVA语言本身提供的函数，从而常常会生成大量的对象（或实例）。由于系统不仅要花时间生成对象，以后可能还需花时间对这些对象进行垃圾回收和处理。因此，生成过多的对象将会给程序的性能带来很大的影响。例1：关于String ,StringBuffer,和appendJAVA语言提供了对于String类型变量的操作。但如果使用不当，会给程序的性能带来影响。如下面的语句：String name=new String("HuangWeiFeng").System.out.println(name "is my name").看似已经很精简了，其实并非如此。为了生成二进制的代码，要进行如下的步骤和操作：(1) 生成新的字符串new String（STR_1).(2) 复制该字符串.(3) 加载字符串常量"HuangWeiFeng"（STR_2).(4) 调用字符串的构架器（Constructor）.(5) 保存该字符串到数组中（从位置0开始）.(6) 从java.io.PrintStream类中得到静态的out变量.(7) 生成新的字符串缓冲变量new StringBuffer(STR_BUF_1).(8) 复制该字

符串缓冲变量.(9) 调用字符串缓冲的构架器（Constructor）.(10) 保存该字符串缓冲到数组中（从位置1开始）.(11) 以STR_1为参数，调用字符串缓冲(StringBuffer)类中的append方法.(12) 加载字符串常量"is my name"(STR_3).(13) 以STR_3为参数，调用字符串缓冲(StringBuffer)类中的append方法.(14) 对于STR_BUF_1执行toString命令.(15) 调用out变量中的println方法，输出结果。由此可以看出，这两行简单的代码，就生成了STR_1,STR_2,STR_3,STR_4和STR_BUF_1五个对象变量。这些生成的类的实例一般都存放在堆中。堆要对所有类的超类，类的实例进行初始化，同时还要调用类极其每个超类的构架器。而这些操作都是非常消耗系统资源的。因此，对对象的生成进行限制，是完全有必要的。经修改，上面的代码可以用如下的代码来替换。StringBuffer name=new StringBuffer("HuangWeiFeng").System.out.println(name.append("is my name.").toString()).系统将进行如下的操作：(1) 生成新的字符串缓冲变量new StringBuffer(STR_BUF_1).(2) 复制该字符串缓冲变量.(3) 加载字符串常量"HuangWeiFeng"(STR_1).(4) 调用字符串缓冲的构架器（Constructor）.(5) 保存该字符串缓冲到数组中（从位置1开始）.(6) 从java.io.PrintStream类中得到静态的out变量.(7) 加载STR_BUF_1.(8) 加载字符串常量"is my name"(STR_2).(9) 以STR_2为参数，调用字符串缓冲(StringBuffer)实例中的append方法.(10) 对于STR_BUF_1执行toString命令(STR_3).(11)调用out变量中的println方法，输出结果。由此可以看出，经过改进后的代码只生成了四个对象变量：STR_1,STR_2,STR_3和STR_BUF_1.你可能觉得少生成一个对象不会对程序的性能有很大的提高。但下面的代码段2

的执行速度将是代码段1的2倍。因为代码段1生成了八个对象，而代码段2只生成了四个对象。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com