

JAVA的字符串拼接与性能Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/644/2021\\_2022\\_JAVA\\_E7\\_9A\\_84\\_E5\\_AD\\_97\\_c104\\_644953.htm](https://www.100test.com/kao_ti2020/644/2021_2022_JAVA_E7_9A_84_E5_AD_97_c104_644953.htm) 在JAVA中拼接两个字符串的最简便的方式就是使用操作符“+”了。如果你用“+”来连接固定长度的字符串，可能性能上会稍受影响，但是如果你是在循环中来“+”多个串的话，性能将指数倍的下降。假设有一个字符串，我们将对这个字符串做大量循环拼接操作，使用“+”的话将得到最低的性能。但是究竟这个性能有多差？如果我们同时也把StringBuffer,StringBuilder或String.concat()放入性能测试中，结果又会如何呢？本文将就这些问题给出一个答案！我们将使用Per4j来计算性能，因为这个工具可以给我们一个完整的性能指标集合，比如最小，最大耗时，统计时间段的标准偏差等。在测试代码中，为了得到一个准确的标准偏差值，我们将执行20个拼接“\*”50,000次的测试。下面是我们将使用到的拼接字符串的方法：Concatenation Operator (+) String concat method concat(String str) StringBuffer append method append(String str) StringBuilder append method append(String str) 最后，我们将看看字节码，来研究这些方法到底是如何执行的。现在，让我们先开始来创建我们的类。注意为了计算每个循环的性能，代码中的每段测试代码都需要用Per4J库进行封装。首先我们先定义迭代次数 1 private static 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](http://www.100test.com)