

依靠简洁的设计和明确的逻辑来优化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\\_ti2020/252/2021\\_2022\\_\\_E4\\_BE\\_9D\\_E9\\_9D\\_A0\\_E7\\_AE\\_80\\_E6\\_c104\\_252462.htm](https://www.100test.com/kao_ti2020/252/2021_2022__E4_BE_9D_E9_9D_A0_E7_AE_80_E6_c104_252462.htm) 作为 Java 开发的重要的一部分就是性能优化。一个没有经过开发小组深入讨论优化问题的 Java 工程是不能称其为完成了。这样做的前提是认定 Java 的速度不够快。而且多数人都持有上述看法，并相信除非他们来“趟这浑水”（指：他们做一些附加的优化工作），否则 Java 是不能完成任务的。你是否看过任何一种宣称能够使你的计算机的性能得到很大提高的工具了？我经常在 Download.com 以及类似的网站上搜寻每一个我能得到的那种工具。我先向那些制作者表示我的敬意。但是，这些工具往往是没有什​​么效果的，只是锻炼了我的耐力。在几个月之后，我又开始去寻找那些工具。要使我形成这种观念“我的电脑跑得慢，无论我怎样做，我是不能使其跑得快些”，是相当困难的。唯一最好的解决方式就是格式化我的硬盘，去掉不需要的东西，不要随心所欲的安装软件。而那些只是在表面上解决问题的方式是没有用的。那也就是我在 Java 领域的经验。如果你计划在编码结束后和发布之前，来一次性能优化，那么，你也许就想错了。这种行动所带来那一点提升的性能是不值得你所投入的时间和精力。忽略性能？那么你是否应当忽略性能，并承认你的程序有合理的理由而不用达到很好的执行性能了？当然不是！好好的设计你的程序，不要使类有太多的层次，而且并不是每个层次都能很好的完成交给它们的任务。这也许只是老生常谈，但是，设计得太过头是个大问题。我猜想设计者倾向于展示他们的能力，或者

他们所知道的时新诀窍。我一直感到：那些没有负担编码责任的设计者，或者一旦完成了最初的设计就不再做什么的设计者，都是不可能做出好的设计。告诉他们，他们要对那些基于他们的设计而写的编码负责任，并尽可能的让设计简洁，而不是成为一种只是用来演示如何使用设计模式的行为。自认为不善于极限编程（extreme programming），但是根据我目前对它所了解，我喜欢这种方式。相比于那些创建能够处理任何事情的巨大系统来说，强调保持系统的灵活性和按需要重新构建（refactor）是一个能够令人欢心鼓舞的观点。

---

----- 关于重构（refactor）我的拼写检查工具不承认refactor这个单词，我也许有必要在此先说明一下。一本叫Extreme Programming Explored的书如下的说明了重构：“重构就是在不影响代码功能的情况下，改善代码设计的过程。为了让我们的代码尽可能的简单，应付任何一种将要遇到的变化，就需要重构。”

---

----- 我相信，使用灵活的和成熟的技术作出的设计，会比那些使用高、精技术的大型设计要能够更好的工作。通常，工程变得复杂的原因是由于多余的使用了像 Enterprise JavaBeans(EJB) 这样的东东。EJB 也可以很有用，但那只是在少数情况下。大多数和JSP、servlet 及 Java类紧密联系的 J2EE 工程是十分不错的，并且也许就是像 Strut 一样的框架。

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)