

Java技巧:提高J2SE性能的代码技巧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\\_ti2020/643/2021\\_2022\\_Java\\_E6\\_8A\\_80\\_E5\\_B7\\_A7\\_c104\\_643107.htm](https://www.100test.com/kao_ti2020/643/2021_2022_Java_E6_8A_80_E5_B7_A7_c104_643107.htm) 1.格式化 文本格式化类：

— java.text.DateFaormat — java.text.NumberFaormat —

java.text.MessageFaormat 可能非常慢 例如：第一次调

用DateFormat.getInstance()创建超过12000个对象 每次调

用DateFormat.format创建13个对象，加上第一次调用时的那次

— 重用同一个DateFormat实例而不是重复调

用DateFormat.getInstance是很值得做的 其它的格式化类有

相似的特点 2.XML处理：SAX 处理XML文档的低级，事件

驱动库 最好的选择，如果：— 你需要以线性的方式遍历

整个文档 — 你只需要查看几项 — 内存效率是一个问题

3.XML处理 试着在你的应用程序里消除处理特定XML对象

通过一个RMI调用传递DOM树，结果会使整个java对象树

序列化。。。通常会4倍甚至更多的空间 4.Buffered I/O I/O

类默认不缓冲 为你的I/O流使用缓冲的包装器

BufferedOutputStream BufferedInputStream BufferedReader

BufferedWriter 尝试在构造方法中使用缓冲大小参数 —

BufferedInputStream:默认2048个字节(JDK1.4.1) —

BufferedReader:默认8192个字符(JDK1.4.1) 5.流与Readers

在java里有两个并行的I/O接口 Reader和Writer —

BufferedInputStream:字符序列(16-bit) InputStream

和OutputStream — 字节序列(8-bit) 处理基于文本的文件

，Readers& Writers更快 更快的幅度达50% 参考OReilly

的"java Performance Tuning"第八章 6.新I/O 在JDK1.4或更高

版本，java.nio包给出了更多I/O选项 — 正则表达式匹配 — 内存映射缓存 — 非中断I/O 这些能提供更高的I/O性能

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)