

java认证:高可移植性J2ME软件开发笔记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_ti2020/571/2021_2022_java_E8_AE_A4_E8_AF_81_c104_571350.htm 1 内容介绍 随着现在的手机品牌越来越多，本来J2ME的手机性能相对于智能机来说就差，加上现在流行的MTK，这样导致手机的CPU，内存就更少的了。这篇文章就是怎么样解决这些小内存，CPU，RMS都严重受限的参考文档。 预期读者 本文主要适合那些有经验的J2ME程序员在优化软件，或者是需要考虑软件兼容性时的参考文档。 2 优化笔记 J2me程序由于其非凡的运行环境限制，所以优化就显得比较重要，以下是我在学习j2me编程所收集的一些技巧和自己的心得。 1.显示图象时确定好你的fps，最好先做几次小实验，这样能让你在显示效果和运行速度上有比较好的平衡。 2.GamaCanvas.getGraphics()每次都会产生一个全新的对象，但是对这些对象的操作都是对同一个graphics，所以还是只取一次供后面使用。 3.让多个对象使用同样的监听器，比如让主MIDlet类实现CommandListener和ItemStateListener接口。 4.考虑使用手机开发商提供的一些sdk，没人会比他们更了解他们手机，所以有些时候能显著提高速度，特别是是图片，视频使用方面。这里推荐使用的模拟器是Nokia S40，Moto SDK 6.11 这些模拟器在模拟的过程中几乎与真机一样。相当的准确。 5.使用监视工具分析MIDlet的瓶颈，wtk和各个公司提供的开发包里都会有，可以找到程序的弱点。假如是在手机上，用timer测试你认为有可以的地方。在这里，开发Http，Socket网络的时候，打开相关的监视器，这样可以知道你的程序到底发送了些什么数

据 6.使用System.gc()，在无线程阻塞的情况下可以有效的缓解内存压力，但是有些公司不是太推荐使用(如nokia).sun的说法也是越低端的片子执行的越慢，总之，慎用吧。 7.用固定的数组代替使用Vector。使用System.arraycopy等native方法，这样比你自己的效率要高 8.图片的优化。考虑使用设备的规格，可能高分辨率的图片不一定显示出来。一般建议使用128色就可以了。 9.不用的对象赋值为null，为更快的回收。特别是字符串，Vector，Hashtable等类变量。如果你的class，写了很多的Hashtable，Vector，Object[]的变量的话，建议你构建一个clear的方法，并在方法里面把那些容器变量先干掉，然后再把class赋值为null。 10.用混淆器处理你的类文件，防止反编译，还有一个好的副作用就是它减小class文件的大小，因为混淆器往往用较短的字符串代替原来的变量或方法名。 11.若数据太大，可以将数据编码为字符串，运行时解码，或把数据存为二进制文件并与程序打包，用类装载器的getResourceAsStream方法在运行时存取。 12.使用现有的类，比如你使用了GameCanvas，就不用自己生成off-screen，另外像CLDC的profile没有构造集合类，所以我们可以用内建的Hashtable和Vector类来实现。 13.用优化软件jPresto(<http://www.s5systems.com/jPresto.htm>)---没用过，不过暂且写上吧。假如对文件大小，内存限制非常严格，这时候就只能用一些非常规的方法了，可能这些方法背离了OO思想，但是多数情况下，它是起到了非常积极的作用的(但是也可能)，假如你更在意于程序的维护和扩展方面，我建议你还是跳过吧 -_- 14.把所有资源文件做成一个数据文件。然后在程序中写一个算法，进行解码 15.把所有的小图片文件做成一

个文件，在运行时再"切割"开。很多张小icon的时候，可以通过合并工具把这些icon合并成一种大的png图片，然后在程序中对这些图片进行切割。用Image.createImage(src, x,y,w,h)进行切割。

16.使用有限的面向对象，因为纯粹的OO往往意味着更多的虚方法，更多的层次关系，更多的class

17.生成尽可能少的class，class都有一定的系统开销。

18.class中生成尽可能少的方法。速度比较：同步方法lt.实例方法lt.静态方法。

19.用final static限定方法可以获得小幅的速度提高。

20.对数据成员用public限定，来代替get和set方法，不过要注重安全性。

其他归类

- *只优化需要的代码
- *只在有价值的地方优化
- *用profiler来找要优化的地方
- *在具体的设备上profiler无能为力，在硬件上使用System timer
- *在于用低级技术之前，总是先研究你的代码并且试着改进算法
- *绘图是慢的，所以尽量节俭地使用图形调用
- *在可以减少绘制区域的地方使用setClip()
- *尽可能的把东西放到循环之外
- *拼命地预先计算和暂存
- *字符串带来垃圾，垃圾不好，所以使用StringBuffers来代替
- *什么都不假设
- *可能就使用static final方法，避免synchronized修饰符
- *传递尽可能少的参数到经常调用的方法
- *如果可能，完全地去掉函数调用
- *解开循环
- *对2的幂的乘除运算用位移运算代替
- *你可以使用位运算符代替取模运算来实现循环
- *试着用零来代替和其他数的比较
- *数组访问比C要慢，所以暂存数组元素
- *消去公共的子表达式
- *局部变量要比引用变量快
- *如果可以callSerially()就不要wait()
- *在switch()中使用小的变量作选项
- *检查定点数学库并且优化它
- *拆开嵌套的FP调用来减少类型转换
- *除法比乘法慢，所以用乘于倒数来代替除法
- *用使用过和测试过的算法
- *为了保护可移植性，小心地使用私有高性

能API。 J2ME优化可能使你的程序在不同的模拟器，不同的设备下有不同的运行效果，所以 优化一定要建立在开发设备的规格上。 以上所列举的方法不一定在所有midp设备上都起作用，也不一定都适合每一个程序，总之，应该根据自身的情况。

3 注意事项 1.内存的管理 2.图片资源的管理 3.图片资源的管理 4.字符串的管理 5.Vector， Hashtabe等管理 4 参考资料 ITPUT的移动开发技术 频道

<http://publish.itpub.net/lists/7826/0/7826.shtml> Nokia 开发者论坛
<http://forum.nokia.com.cn/sch/index.html> 更多优质资料尽在百考试题论坛 百考试题在线题库 java认证更多详细资料 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