Java代码的优化策略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_ti2020/629/2021_2022_Java_E4_BB_A3_E7_A0_81_c104_629774.htm 整理了一些影响性能的代码和优化方法，以后希望能陆续补充和优化 1. 如何使用Exception Exception降低性能。一个异常抛出首先需要创建一个新的对象。Throwable接口中的构造器调用名为 fillInStackTrace()的本地方法。这个方法负责巡检栈的整个框架来收集跟踪信息。这样无论何时有异常抛出，它要求虚拟机装载调用栈，因为一个新的对象在中部被创建。 异常应当仅用于有错误发生时，而不要控制流。 2. 不要两次初始化变量 Java通过调用独特的类构造器默认地初始化变量为一个已知的值。所有的对象被设置成null，integers (byte, short, int, long)被设置成0，float和double设置成0.0，Boolean变量设置成false。这对那些扩展自其它类的类尤其重要，这跟使用一个新的关键词创建一个对象时所有一连串的构造器被自动调用一样。 3. 在任何可能的地方让类为Final 标记为final的类不能被扩展。在《核心Java API》中有大量这个技术java认证更多详细资料的例子，诸如java.lang.String。将String类标记为final阻止了开发者创建他们自己实现的长度方法。 更深入点说，如果类是final的，所有类的方法也是final的。Java编译器可能会内联所有的方法(这依赖于编译器的实现)。在我的测试里，我已经看到性能平均增加了50%。 4. 在任何可能的地方使用局部变量 属于方法调用部分的自变量和声明为此调用一部分的临时变量存储在栈中，这比较快。诸如static，实例(instance)变量和新的对象创建在堆中，这比较慢。局部变量的更深入优化依赖于你正

在使用的编译器或虚拟机。 5. 停止小聪明 很多开发人员在脑子中编写可复用和灵活的代码，而有时候在他们的程序中就产生额外的开销。曾经或者另外的时候他们编写了类似这样的代码： public void doSomething(File file) { FileInputStream fileIn = new FileInputStream(file). // do something 他够灵活，但是同时他们也产生了更多的开销。这个主意背后做的事情是操纵一个InputStream，而不是一个文件，因此它应该重写如下： public void doSomething(InputStream inputStream){ // do something 6. 乘法和除法 我有太多的东东适用于摩尔法则它声明CPU功率每年成倍增长。“ 摩尔法则 ”表明每年由开发者所写的差劲的代码数量三倍增加，划去了摩尔法则的任何好处。 考虑下面的代码： for (val = 0. val lt. 100000. val = 5) { shiftX = val lt. 3. myRaise = val lt. 1. } 代替了乘以8，我们使用同等效果的左移3位。每一个移动相当于乘以2，变量myRaise对此做了证明。同样向右移位相当于除以2，当然这会使执行速度加快，但可能会使你的东东以后难于理解.所以这只是个建议 7. 用代码有效处理内存溢出 OutOfMemoryError是由于内存不够后普遍会遇到的问题，下面一段代码能有效判断内存溢出错误，并在内存溢出发生时有效回收内存 通过该方法可以联想到有效管理连接池溢出，道理等同。 import Java.util.*. public class DataServer { private Hashtable data = new Hashtable(). public Object get (String key) { Object obj = data.get (key). if (obj == null) { System.out.print (key “ ”). try { // simulate getting lots of data obj = new Double[1000000]. data.put (key, obj). } catch (OutOfMemoryError e) { System.out.print (“ No Memory! ”). flushCache(). obj = get (key).// try again } } return (obj). } public

void flushCache() { System.out.println (“ Clearing cache ”). data.clear(). } public static void main (String[] args) { DataServer ds = new DataServer(). int count = 0. while (true) // infinite loop for test ds.get (“ ” count). } } 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