详细讲述Java中的克隆 PDF转换可能丢失图片或格式，建议阅读原文
https://www.100test.com/kao_ti2020/253/2021_2022__E8_AF_A6_E7_BB_86_E8_AE_B2_E8_c104_253553.htm 经常听到有人说java中没有指针。事实如此吗？no，java是有指针的，只不过换了个名字而已，也就是我们经常提到的引用。我们知道，在java中一切都是对象，那么我们如何操控对象？如何在成千上万的对象中找到我们所需的那个对象呢？又是如何让对象按照我们的意思来完成任务的呢？ Object o = new Object()；这是java中最常见的语句了，在这句话中做了三件事。首先声明一个Object类型的变量o，在内存中为对象划分一块地址new Object（），将声明的变量指向内存中的对象。如此一来，我们就可以通过o来操纵对象了。就好像孩子们玩的遥控飞机，在空中飞行的是飞机，而使它做出优美动作的却是孩子们手中的摇控器。 "克隆"是如今听到的较多的词汇，听说已经将某只羊克隆了好几份了。但愿这种技术不要在人身上实验。java中也有"克隆"，与现实世界的克隆一样，将一个实际存在的对象拷贝几份。如下：//倒霉的羊public class Sheep implements Cloneable{private String name.public void setName(String arg) {name = arg.}public String getName() {return name.}public Object clone() throws CloneNotSupportedException {return super.clone().}}//克隆public class Main {public static void main(String[] args) throws CloneNotSupportedException {Sheep sheep = new Sheep(). //先得到那只羊的实例sheep.setName("我是真的"). //给它做个记号System.out.println("sheep.getName() = " sheep.getName()).Sheep sheepClone = (Sheep)sheep.clone(). //开

始克隆System.out.println("sheepClone.getName() = " sheepClone.getName()).}} 运行程序结果为： sheep.getName() = 我是真的 sheepClone.getName() = 我是真的 两只羊是一模一样的（哪怕那只羊瘸腿）。让我们来看看代码。首先要注意的是Sheep类实现了Cloneable接口（该接口属于java.lang包，默认已经导入了），该接口中并没有定义要实现的方法，是个空接口，起标志作用。也就是说，实现了这个接口的羊就不再是只普通的羊，它是一只可以被克隆的羊。再往下看，有个clone方法，返回Object类型的对象，并抛出CloneNotSupportedException异常。该方法覆写了父类（Object）的clone方法，并在最后调用了super.clone()，这也意味着无论clone类继承结构是什么样的，super.clone ()都会直接或间接调用Object类的clone()方法。看看jdk帮助文档会发现，Object类的clone()是一个native方法，我们知道，native方法的效率一般来说都是远高于java中的非native方法。这也说明了new一个对象，然后将原对象中的数据导入到新创建的对象中去的做法是多么愚蠢。必须说明的是Object中的clone方法是protected的，所以要使用clone就必须继承Object类（默认）。并且为了可以使其它类调用该方法，必须将其作用域设置为public. 以上只是一个简单clone的实现。明天说说"影子clone"和"深度clone". 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