

Java传址或传值计算机等级考试 PDF转换可能丢失图片或格式
， 建议阅读原文

https://www.100test.com/kao_ti2020/638/2021_2022_Java_E4_BC_A0_E5_9D_80_c97_638820.htm 编辑特别推荐: 全国计算机等级考试 (等考) 指定教材 全国计算机等级考试学习视频 全国计算机等级考试网上辅导招生 全国计算机等级考试时间及科目预告 百考试题教育全国计算机等级考试在线测试平台 全国计算机等级考试资料下载 全国计算机等级考试论坛

java 是传值还是传址呢,这个问题已经要被讨论烂了.在我的认识中也是几经反复,结果都没有认识到本质. 有些人说是传值的,有些人说要看情况,有些情况是传值,有些情况是传址 java 实际只是传值的,为什么我们会对它的认识有误区呢,来看个例子吧.

```
view plaincopy to clipboardprint? package com.yecg.java.passvalue.  
public class PassReferenceOrValue { /** * @param args */ public  
static void main(String[] args) { // TODO Auto-generated method  
stub int a =1. passValue(a). System.out.println(a). } public static  
void passValue(int a){ a = 5. } } package com.yecg.java.passvalue.  
public class PassReferenceOrValue { /** * @param args */ public  
static void main(String[] args) { // TODO Auto-generated method  
stub int a =1. passValue(a). System.out.println(a). } public static  
void passValue(int a){ a = 5. } }
```

结果很显然是1,那么这里就证明了java实际上是传值的,它在调用函数的时候传入的参数实际上是原数的拷贝,对拷贝进行了修改是不影响原值的 那么为什么会认为它有时候是传址的呢,在于参数是一个对象的时候

```
view plaincopy to clipboardprint? package com.yecg.java.passvalue.  
public class PassReferenceOrValue { /** * @param args */ public
```

```

static void main(String[] args) { // TODO Auto-generated method
stub //parameter is object Value value = new Value().
value.setParam("int"). passValue(value).
System.out.println(value.getParam()). } public static void
passValue(Value v){ v.setParam("changed"). } } class Value{ String
param. public String getParam() { return param. } public void
setParam(String param) { this.param = param. } } package
com.yecg.java.passvalue. public class PassReferenceOrValue { /** *
@param args */ public static void main(String[] args) { // TODO
Auto-generated method stub //parameter is object Value value =
new Value(). value.setParam("int"). passValue(value).
System.out.println(value.getParam()). } public static void
passValue(Value v){ v.setParam("changed"). } } class Value{ String
param. public String getParam() { return param. } public void
setParam(String param) { this.param = param. } }

```

这里我传入的是一个对象.通过调用函数把对象里面的值改变了.结果是changed, 为什么会这样呢,照理说如果它是传值的话结果应该是int才对阿,因为它传的是个对象的拷贝阿. 这里的确是传的值,只是这个值是value对象地址的拷贝.所以现在我对它做改变原先的对象是会改变的. 可能这里你还是觉得它是传址,那么我现在就来证明他为什么不是传址,passvalue方法改成这样:

```

view plaincopy to clipboardprint? public static void passValue(Value
v){ Value value = new Value(). value.setParam("changed"). v =
value. } public static void passValue(Value v){ Value value = new
Value(). value.setParam("changed"). v = value. }

```

现在执行的结果是int了.所以你现在应该明白传的是对象地址的拷贝这句话的

含义了,因为它如果不是拷贝而就是传的地址的话,那么我新new了一个value对象,并对参数进行赋值,(v=value) 那么这个时候两个对象的地址就该一样,我对新的value进行修改,也会对原来的value对象产生影响.但是现在并没有.所以在java中最好不要对参数进行赋值,无论是对象还是基本类型,因为这时没有意义的,只会把程序搞的更加的不可读. 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