

java认证:Java究竟是传值还是传引用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_ti2020/582/2021_2022_java_E8_AE_A4_E8_AF_81_c104_582671.htm

1. 简单类型是按值传递的 Java 方法的参数是简单类型的时候，是按值传递的 (pass by value)

。这一点我们可以通过一个简单的例子来说明：`public class Test {public static void test(boolean test) {test = !`

`test.System.out.println("In test(boolean) : test = " test);}public static void main(String[] args) {boolean test =`

`true.System.out.println("Before test(boolean) : test = "`

`test).test(test).System.out.println("After test(boolean) : test = "`

`test).}}` 运行结果：`Before test(boolean) : test = true`

`In test(boolean) : test = false`

`After test(boolean) : test = true` 不难看出，虽然在 `test(boolean)` 方法中改变了传进来的参数的值，但对

这个参数源变量本身并没有影响，即对 `main(String[])` 方法里的

`test` 变量没有影响。那说明，参数类型是简单类型的时候

，是按值传递的。以参数形式传递简单类型的变量时，实际上

是将参数的值作了一个拷贝传进方法函数的，那么在方法

函数里再怎么改变其值，其结果都是只改变了拷贝的值，而

不是源值。2. 什么是引用 Java 是传值还是传引用，问题主要

出在对象的传递上，因为 Java 中简单类型没有引用。既然争

论中提到了引用这个东西，为了搞清楚这个问题，我们必须

要知道引用是什么。简单的说，引用其实就像是一个对象的

名字或者别名 (alias)，一个对象在内存中会请求一块空间来保

存数据，根据对象的大小，它可能需要占用的空间大小也不

等。访问对象的时候，我们不会直接是访问对象在内存中的

数据，而是通过引用去访问。引用也是一种数据类型，我们可以把它想象为类似 C 语言中指针的东西，它指示了对象在内存中的地址只不过我们不能观察到这个地址究竟是什么。如果我们定义了不止一个引用指向同一个对象，那么这些引用是不相同的，因为引用也是一种数据类型，需要一定的内存空间来保存。但是它们的值是相同的，都指示同一个对象在内存中的位置。比如 `String a = "Hello".String b = a`。这里，`a` 和 `b` 是不同的两个引用，我们使用了两个定义语句来定义它们。但它们的值是一样的，都指向同一个对象 "Hello"。也许你还觉得不够直观，因为 `String` 对象的值本身是不可更改的（像 `b = "World". b = a`。这种情况不是改变了 "World" 这一对象的值，而是改变了它的引用 `b` 的值使之指向了另一个 `String` 对象 `a`）。那么我们用 `StringBuffer` 来举一个例子：`public class Test {public static void main(String[] args) {StringBuffer a = new StringBuffer("Hello").StringBuffer b = a.b.append("World").System.out.println("a is " a).}}`运行结果：`a is Hello, World` 这个例子中 `a` 和 `b` 都是引用，当改变了 `b` 指示的对象的值的时候，从输出结果来看，`a` 所指示的对象的值也改变了。所以，`a` 和 `b` 都指向同一个对象即包含 "Hello" 的一个 `StringBuffer` 对象。这里我描述了两个要点：1. 引用是一种数据类型，保存了对象在内存中的地址，这种类型即不是我们平时所说的简单数据类型也不是类实例(对象)；2. 不同的引用可能指向同一个对象，换句话说，一个对象可以有多个引用，即该类类型的变量。3. 对象是如何传递的呢 关于对象的传递，有两种说法，即“它是按值传递的”和“它是按引用传递的”。这两种说法各有各的道理，但是它们都没有从

本质上去分析，即致于产生了争论。既然现在我们已经知道了引用是什么东西，那么现在不妨来分析一下对象作为参数是如何传递的。还是先以一个程序为例：`public class Test {public static void test(StringBuffer str) {str.append("World!");}public static void main(String[] args) {StringBuffer string = new`

`StringBuffer("Hello").test(string).System.out.println(string).}}`运行结果：Hello, World! `test(string)` 调用了 `test(StringBuffer)` 方法，并将 `string` 作为参数传递了进去。这里 `string` 是一个引用，这一点是毋庸置疑的。前面提到，引用是一种数据类型，而且不是对象，所以它不可能按引用传递，所以它是按值传递的，那么它的值究竟是什么？是对象的地址。由此可见，对象作为参数的时候是按值传递的，对吗？错！为什么错，让我们看另一个例子：

`public class Test {public static void test(String str) {str = "World";}public static void main(String[] args) {String string = "Hello".test(string).System.out.println(string).}}`运行结果：Hello 为什么会这样呢？因为参数 `str` 是一个引用，而且它与 `string` 是不同的引用，虽然它们都是同一个对象的引用。`str = "World"` 则改变了 `str` 的值，使之指向了另一个对象，然而 `str` 指向的对象改变了，但它并没有对 "Hello" 造成任何影响，而且由于 `string` 和 `str` 是不同的引用，`str` 的改变也没有对 `string` 造成任何影响，结果就如例中所示。其结果是推翻了参数按值传递的说法。那么，对象作为参数的时候是按引用传递的了？也错！因为上一个例子的确能够说明它是按值传递的。结果，就像光到底是波还是粒子的问题一样，Java 方法的参数是按什么传递的问题，其答案就只能是：即是按

值传递也是按引用传递，只是参照物不同，结果也就不同。

单纯考虑参数str存的也是一种数据类型，可以看成是值传递。考虑参数str它是对象string的一个引用，此时就可看做是引用传递。

4. 正确看待传值还是传引用的问题 要正确的看待这个问题必须要搞清楚为什么会有这样一个问题。实际上，问题来源于C，而不是Java。C语言中有一种数据类型叫做指针，于是将一个数据作为参数传递给某个函数的时候，就有两种方式：传值，或是传指针，它们的区别，可以用一个简单的例子说明：

```
void SwapValue(int a, int b) {int t = a.a = b.b = t;}void SwapPointer(int * a, int * b) {int t = * a.* a = * b.* b = t;}void main() {int a = 0, b = 1.printf("1 : a = %d, b = %d\n", a, b).SwapValue(a, b).printf("2 : a = %d, b = %d\n", a, b).SwapPointer(amp.b).printf("3 : a = %d, b = %d\n", a, b).}
```

运行结果：1 : a = 0, b = 1 2 : a = 0, b = 1 3 : a = 1, b = 0 大家可以明显的看到，按指针传递参数可以方便的修改通过参数传递进来的值，而按值传递就不行。当Java成长起来的时候，许多的C程序员开始转向学习Java，他们发现，使用类似SwapValue的方法仍然不能改变通过参数传递进来的简单数据类型的值，但是如果是一个对象，则可能将其成员随意更改。于是他们觉得这很像是C语言中传值/传指针的问题。但是Java中没有指针，那么这个问题就演变成了传值/传引用的问题。可惜将这个问题放在Java中进行讨论并不恰当。讨论这样一个问题的最终目的只是为了搞清楚何种情况才能在方法函数中方便的更改参数的值并使之长期有效。Java中，改变参数的值有两种情况，第一种，使用赋值号“=”直接进行赋值使其改变，如例1和例4；第二种，对于某些对象的引用，通过一定途径

对其成员数据进行改变，如例 3.对于第一种情况，其改变不会影响到方法该方法以外的数据，或者说源数据。而第二种方法，则相反，会影响到源数据因为引用指示的对象没有变，对其成员数据进行改变则实质上是改变的该对象。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