

# JAVA中值传递和引用传递的三种情况

原创

2016年12月04日 00:14:39

标签: java / 值传递 / 引用传递 / String

3208

## 前言

我们都知道，java中数据类型分为基本数据类型和引用数据类型。

- 基本数据类型
  - 整型：byte，short，int，long
  - 浮点型：float，double
  - 字符型：char
  - 布尔型：boolean
- 引用数据类型
  - 数组
  - 类
  - 接口

方法的参数分为实际参数，和形式参数。

- 形式参数：定义方法时写的参数。
- 实际参数：调用方法时写的具体数值。

一般情况下，在数据做为参数传递的时候，基本数据类型是值传递，引用数据类型是引用传递（地址传递）。

## 值传递

```
1 public static void main(String[] args) {
2     int num1 = 10;
3     int num2 = 20;
4
5     swap(num1, num2);
6
7     System.out.println("num1 = " + num1);
8     System.out.println("num2 = " + num2);
9 }
10
11 public static void swap(int a, int b) {
12     int temp = a;
13     a = b;
14     b = temp;
15
16     System.out.println("a = " + a);
17     System.out.println("b = " + b);
18 }
```

运行的结果是：

```
1 a = 20
2 b = 10
3 num1 = 10
4 num2 = 20
```

原因：

流程：

- 主函数进栈，num1、num2初始化。

### 联系我们



请扫描二维码联系客服  
webmaster@csdn.net  
400-660-0108  
QQ客服 客服论坛

关于 招聘 广告服务 百度  
©1999-2018 CSDN版权所有  
京ICP证09002463号

经营性网站备案信息  
网络110报警服务  
中国互联网举报中心  
北京互联网违法和不良信息举报中心

### 他的最新文章 更多文章

- 类的初始化过程
- 多态的总结
- DP与SP的区别
- Android启动页面留白

### 文章分类

Java基础	3篇
Android填坑	1篇
Android基础	1篇

### 文章存档

2016年12月	1篇
2016年11月	2篇
2016年9月	2篇

### 他的热门文章

- JAVA中值传递和引用传递的三种情况  
3080
- DP与SP的区别  
201
- 类的初始化过程  
114
- 多态的总结  
100
- Android启动页面留白  
81

2. 调用swap方法，swap( )进栈，将num1和num2的**值**，复制一份给a和b。
3. swap方法中对a、b的值进行交换。
4. swap方法运行完毕，a、b的值已经交换。
5. swap方法弹栈。
6. 主函数弹栈。

## 解析：

在swap方法中，a、b的值进行交换，并不会影响到num1、num2。因为，a、b中的值，只是从num1、num2的复制过来的。

也就是说，a、b相当于num1、num2的副本，副本的内容无论怎么修改，都不会影响到原件本身。

## 引用传递

```
1 public static void main(String[] args) {
2     int[] arr = {1,2,3,4,5};
3
4     change(arr);
5
6     System.out.println(arr[0]);
7 }
8
9 //将数组的第一个元素变为0
10 public static void change(int[] array) {
11     int len = array.length;
12     array[0] = 0;
13 }
```

运行的结果是：

```
1 0
```

## 原因：

## 流程：

1. 主函数进栈，int[] arr初始化。
2. 调用change方法，change( )进栈，将arr的**地址值**，复制一份给array。
3. change方法中，根据地址值，找到堆中的数组，并将第一个元素的值改为0。
4. change方法运行完毕，数组中第一个元素的值已经改变。
5. change方法弹栈。
6. 主函数弹栈。

## 解析：

调用change()的时候，形参array接收的是arr地址值的副本。并在change方法中，通过地址值，对数组进行操作。change方法弹栈以后，数组中的值已经改变。main方法中，打印出来的arr[0]也就从原来的1变成了0。

无论是主函数，还是change方法，操作的都是同一个地址值对应的数组。

就像你把自己家的钥匙给了另一个人，这个人拿着钥匙在你家一顿瞎折腾，然后走了。等你拿着钥匙回到家以后，家里已经变成了被折腾过后，惨不忍睹的样子。。

这里的钥匙就相当于地址值，家就相当于数组本身。

## String类型传递

```
1 public static void main(String[] args) {
2     String str = "AAA";
3
4     change(str);
5
6     System.out.println(str);
7 }
```

### 联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

👤 QQ客服 🗨 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

```
8 public static void change(String s) {
9     s = "abc";
10 }
```

运行的结果是：

```
1 AAA
```

这就神奇了！！！

String是一个类，类是引用数据类型，做为参数传递的时候，应该是引用传递。但是从结果看起来却是值传递。

## 原因：

String的API中有这么一句话：“their values cannot be changed after they are created”，意思是：String的值在创建之后不能被更改。

API中还有一段：

```
String str = "abc";
```

等效于：

```
char data[] = {'a', 'b', 'c'};
```

```
String str = new String(data);
```

也就是说：对String对象str的任何修改 等同于 重新创建一个对象，并将新的地址值赋值给str。

这样的话，上面的代码就可以写成：

```
1 public static void main(String[] args) {
2     String str1 = "AAA";
3
4     change(str1);
5
6     System.out.println(str1);
7 }
8 public static void change(String s) {
9     char data[] = {'a', 'b', 'c'}
10    String str = new String(data);
11    s = str;
12 }
```

## 流程：

1. 主函数进栈，str1初始化。
2. 调用change方法，change( )进栈，将str1的**地址值**，复制一份给s。
3. change方法中，重现创建了一个String对象"abc"，并将s指向了新的地址值。
4. change方法运行完毕，s所指向的地址值已经改变。
5. change方法弹栈。
6. 主函数弹栈。

## 解析：

String对象做为参数传递时，走的依然是引用传递，只不过String这个类比较特殊。

**String对象一旦创建，内容不可更改。每一次内容的更改都是重现创建出来的新对象。**

当change方法执行完毕时，s所指向的地址值已经改变。而s本来的地址值就是copy过来的副本，所以并不能改变str1的值。

## String类型类似情况：

```
1 class Person {
2     String name;
3
4     public Person(String name) {
5         this.name = name;
6     }
7 }
8 public class Test {
9     public static void main(String[] args) {
```

### 联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

👤 QQ客服 🗨 客服论坛

关于 招聘 广告服务 🐼 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

```
10         Person p = new Person("张三");
11
12         change(p);
13
14         System.out.println(p.name);
15     }
16
17     public static void change(Person p) {
18         Person person = new Person("李四");
19         p = person;
20     }
21 }
```

运行的结果是：

```
1 张三
```

## 总结

- 值传递的时候，将实参的**值**，copy一份给形参。
- 引用传递的时候，将实参的**地址值**，copy一份给形参。

也就是说，不管是值传递还是引用传递，形参拿到的仅仅是实参的副本，而不是实参本身。

版权声明：本文为博主原创文章，未经允许请随意转载。 <http://blog.csdn.net/zhzhao999/article/details/53449504>



严禁讨论涉及中国之军/政相关话题，违者会被禁言、封号！



a2572371 2018-02-23 14:11

1条回复 回复 1楼

```
public static void change(Person p) {
    p.setName("李四");
}
```

试试看，引用传递是引用的本身，好吧。

## java中按值传递和引用传递区别



morgerton 2017年02月08日 12:54 1693

java中传参方式

## Java：按值传递还是按引用传递详细解说



zzp\_403184692 2012年11月14日 23:02 135783

前天在做系统的时候被Java中参数传递问题卡了一下，回头查阅了相关的资料，对参数传递问题有了新的了解和掌握，但是有个问题感觉还是很模糊，就是Java中到底是否只存在值传递，因为在查阅资料时，经常看到有...

## java中的值传递和引用传递



WYZSC 2011年04月22日 12:51 22994

java中的值传递和引用传递，

## 一个简单的例子让你理解值传递和引用传递



java\_zys 2016年04月20日 16:36 3132

碰到一些做了好几年java开发的程序员还弄不清楚值传递和引用传递的区别，下面通过一个简单的例子来说明：public class Person { private String name; publi...

## 值传递和引用传递的区别



liqianglai 2014年05月09日 16:23 4646

所谓值传递，就是说仅将对象的值传递给目标对象，就相当于copy；系统将为目标对象重新开辟一个完全相同的内存空间。所谓引用，就是说将对象在内存中的地址传递给目标对象，就相当于使目标对象和原始对象对应...

## 联系我们



请扫描二维码联系客服

webmaster@csdn.net

400-660-0108

QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

几句话弄清楚Java参数传值还是传引用

yangliuy

2015年01月28日 14:58

17024

最近刷题用递归解决关于树的问题的时候，在递归函数调用的时候什么时候传入值，什么时候传入引用有疑问，因为这关系到什么时候我们需要恢复现场，什么时候不需要。在网上搜索了一下，感觉这篇总结的非常简单明了，简...

Java到底是值传递？还是引用传递？

xiaosha009

2016年02月17日 14:23

1770

前言 最近和同事讨论算法效率问题无意中涉及到一个问题，java中调用方法的时候是值传递呢？还是引用传递？网上搜索一下相关问题，众说纷纭，有人说值传递，有人说引用传递，还有人说“基础类型是值传递，对象...

C++引用传递和值传递

zzuqqiu

2010年05月23日 00:39

10911

从概念上讲。指针从本质上讲就是存放变量地址的一个变量，在逻辑上是独立的，它可以被改变，包括其所指向的地址的改变和其指向的地址中所存放的数据的改变。而引用是一个别名，它在逻辑上不是独立的，它的存在具有依...

你真的理解Java的按引用传递吗？

cauchyweierstrass

2015年11月02日 10:42

8199

首先我们来看下面这段代码：public class Test1 { String a = "123"; public static void change(Test1 test) ...

Java到底有没有引用传递？

qq598535550

2016年10月29日 22:27

4803

（1）“在Java里面参数传递都是按值传递”这句话的意思是：按值传递是传递的值的拷贝，按引用传递其实传递的是引用的地址值，所以统称按值传递。（2）在Java里面只有基本类型和String，其它的都是...

C++入门篇之值传递，指针传递和引用传递的那些事

Ivan\_zgj

2016年02月26日 13:49

540

最近在自学C++，一边学习一边在心里将C++与C语言和Java进行比较。毫无疑问，作为以C语言为基础，和C语言同一个爹的一门编程语言，C++自然与C原因有很多相似之处，例如C++的变量传递就沿用了C语...

帮助你彻底理解值Java中的传递和引用传递

androidstarjack

2017年04月29日 22:01

814

最近在学习ndk开发时，对指针的概念了解的有点含糊。在这里简单的定义为一块内存地址。申请到一块内存后，比如 i=3；如果将其传入到一个方法体内，任意改变i，则最终i的支部会发生任何变化。接下来我们就以...

Java中只有值传递，没有引用传递

chenjiatao0909

2016年08月09日 16:25

404

在Java编程语言中，只有值传递，没有引用传递首先，要明白什么是值传递，什么是引用传递。①值传递的定义就是传递变量的值。②引用传递的定义就是传递变量的地址。现在，举例举例解释下值传递和引用传递，这里先...

java 引用传递，还是值传递 之坑

liangxw1

2016年03月30日 22:21

419

对非原生类型，即对象类型值传递的坑，工作中遇到的实际问题，记录下来，以备回忆和参考，说的不准确的地方务请大神留言，注，此处对传递的是String的情况暂不讨论，这个相对比较特殊，网上也有文章另...

JAVA 方法的参数是按值传递还是引用传递？

xinnian468153

2016年08月21日 09:33

527

JAVA 方法的参数是按值传递还是引用传递？

关于Java的值传递和引用传递

scau\_syd

2016年09月19日 19:55

424

结论：1. 对于值传递，如int, long等基本类型，不管传入方法后怎么修改，都改变不了方法外面的值。2. 对于引用传递，如果传入方法后指向了一个新的对象，那么无论怎么修改也改变不了方法外面...

联系我们



请扫描二维码联系客服

webmaster@csdn.net

400-660-0108

QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有  
京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

## java基本数据类型传递与引用传递区别详解



javazejian

2016年04月19日 17:12

📖 14304

java的值传递和引用传递在面试中一般都会都被涉及到，今天我们就来聊聊这个问题，首先我们必须认识到这个问题一般是相对函数而言的，也就是java中的方法参数，那么我们先来回顾一下在程序设计语言中有关参数...

## Java的参数传递到底是引用传递还是值传递



panyongcsd

2016年08月30日 17:14

📖 2186

首先看一段代码：`public static void main(String[] args) { Person a = new Person(23, "a"); Person b = ...`

## 什么是 值传递，地址传递和引用传递 有什么区别

什么是值传递，地址传递和引用传递？它们有什么不同？这在学校考试，找工作面试中都是常见的问题。它们之间的区别容易搞错，长时间不用也容易忘记。最近又被问到这个问题，想想还是整理一下记下来，也方便以后参考。...



The\_dying\_man

2017年06月14日 23:33

📖 951

## 按值传递还是按引用传递详细解说



xx123698

2016年12月12日 15:14

📖 1025

1：按值传递是什么 指的是在方法调用时，传递的参数是按值的拷贝传递。示例如下：`[java] view plain copy public class TempTest { ...`

### 联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

👤 QQ客服 🗨 客服论坛

关于 招聘 广告服务 🐾 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心