



链滴

为什么 java 中只有值传递

作者: [douniwan](#)

原文链接: <https://ld246.com/article/1572216855265>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



值传递||引用传递??

值传递 (pass by value) : 是指在调用函数时将实际参数复制一份传递到函数中, 这样在函数中如参数进行修改, 将不会影响到实际参数

引用传递 (pass by reference) : 是指在调用函数的时候将实际参数的地址直接传递到函数中, 那在函数中对参数的修改, 将影响到实际参数。

区别: 值传递会创建副本, 引用传递不会

到底是什么传递?

关于java中的参数传递是值传递还是引用传递的问题, 对于我这个新手来说的确是十分恼火的, 之前的不确定理解 (...反正不清楚) 如果是传递的参数是普通类型, 那就是值传递, 如果是对象, 那就是用传递。而在我看到Hollis的关于这个问题的讲解时, 我豁然开朗, 原来java中只有值传递呀。

为什么会存在参数是对象类型的时候是引用传递的理解呢? 因为有代码证明呀 ~ ~ ~ (事实是举例不)

例子 1

```
public class Test{
public static void main(String[] args){
    Test test=new Test();
    Employee employee=new Employee();
    employee.setName("employee-A");
    employee.pass(employee);
    System.out.println("print in main,user is "+employee);
}
    public void pass(Employee employee){
        employee.setName("employee-B")
    }
}
```

```
    System.out.println("print in pass,user is "+employee);
  }
}
```

输出结果如下:

```
print in pass,user is {name='employee-B'}
```

```
print in main,user is {name='employee-B'}
```

经过pass方法后, 实参的值竟然被改变了, 那么按照上面的引用传递的定义, 这不就是引用传递了么所以有人就得出了在java的方法中, 传递的是对象类型的时候是引用传递。

但是, 这种表述事实上是错误的, 我们接着看下面一个同样是对象作为参数的情况

例子 2

```
public class Test{
public static void main(String[] args){
  Test test=new Test();
  Employee employee=new Employee();
  employee.setName("employee-A");
  employee.pass(employee);
  System.out.println("print in main,user is "+employee);
}
  public void pass(Employee employee){
  employee=new Employee();
  employee.setName("employee-B")
  System.out.println("print in pass,user is "+employee);
  }
}
```

输出结果:

```
print in pass,user is {name='employee-B'}
```

```
print in main,user is {name='employee-A'}
```

在上面的pass中改变了employee对象, 如果是引用传递的话那么在输出应该跟例子1一样呀, 但是并不一样, 这是为什么呢?

事实上在上面的过程中, 当我们调用pass方法的时候, 并把employee作为实际参数传递给形式参数employee的时候, 会把实际参数的地址给形式参数, 这时形式参数也指向了实际参数的地址。所以在例子中, 我们对形式参数的属性进行修改的时候会影响到实际参数, 因为此时形式参数持有的地址就是实际参数的地址。而在例子2中, 形式参数得到实际参数的地址后, 我们又对形式参数重新分配一个对象例, 此时形式参数不在只有实际参数的地址, 而是持有新建立的对象的地址。

也就是说, 这里实际是把实际参数的地址复制了一份, 传递给了形式参数。所以这个过程其实是值传递, 把实际对象的地址当做了值传递给了形式参数。

更加详细的内容: <https://mp.weixin.qq.com/s/F7Niaa7nD1tLApCEGKAj4A>

原来是值传递呀!

在了解到Java中只有值传递后，脑海中又浮现出好像看到过相关知识的情节。于是再去翻 java核心技
卷I的时候，发现在关于方法参数的那一节讲过这个问题，其中给出了如下例子。

```
public static void swap(Employee x,Employee y){
    Employee temp=x;
    x=y;
    y=temp;
}
public static void main(String[] args){
Employee a=new Employee("Alice",...);
Employee b=new Employee("Bob",...);
swap(a,b);
}
```

执行swap(a,b)后，a仍然是Alice 而 b仍然是Bob。

由此也可以看出参数传递的方式为值传递。