

一个有意思的Java重载与多态问题 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/645/2021\\_2022\\_\\_E4\\_B8\\_80\\_E4\\_B8\\_AA\\_E6\\_9C\\_89\\_E6\\_c104\\_645115.htm](https://www.100test.com/kao_ti2020/645/2021_2022__E4_B8_80_E4_B8_AA_E6_9C_89_E6_c104_645115.htm) 在论坛上看到的一个例子，代码如下

```
public class SuperClass { private int value. public void method(SuperClass p){ this.value =p.value. } public void addValue(int v){ value =v. } public int getValue(){ return value. } }
```

```
public class SubClass extends SuperClass{ public void method(SubClass p){ super.method(p). this.addValue(4). } public void method(SuperClass p){ this.addValue(10). } }
```

```
public class Example { public static void main(String[] args){ SuperClass o2. SubClass o3. o2=new SubClass(). o3=new SubClass().
```

```
o2.method(o3). System.out.println(o2.getValue()). o3.method(o3). System.out.println(o3.getValue()). } }
```

输出结果为 10 4，为什么？这道题的关键在于方法的重载与多态。多态，简单来说就是，定义为父类的对象被赋予了子类的实例，那么执行该对象的方法的时候，会优先调用子类的方法。这样一来，看一下这段逻辑混乱的代码，o2.method(o3).o2是父类的类型所以即使赋给他子类的实体也是无法调用父类没有定义过的函数，也就是说superclass.method(subclass)这个方法不存在，所以编译器认为这个method是那个参数为父类的函数，即为superclass.method(superclass)；同时实体是子类的实体，所以结果就是调用子类override过的参数为父类的函数。最后，论坛上的高手说的比较好，关于设计的一些基本原则，尽可能不要override已经有实体的函数，如果一个函数打算在子类被override那就不要在父类做具体实现而是声明成抽象函数。

否则很容易违背里氏代换原则以及无法实现面向对象设计最根本的目的开闭原则的。面向对象的overlode和override不是被创造出来给人们讨论这类的繁琐的问题，而是用来让设计更清晰更人性更容易被人理解以及变更修改的。这样的代码的存在就已经违背了多态重载的初衷了。相关推荐：  
#0000ff>Jjava读取文件内容的函数 #0000ff>Java和PHP在Web开发方面的比较 #0000ff>Spring使用入门示例 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](http://www.100test.com)