

java初学者实践教程8jdk5的装箱与装箱 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/270/2021_2022_java_E5_88_9D_E5_AD_A6_c67_270395.htm 我们现在知道了，所有对象型的数据类型的基类是java.lang.Object。而写java程序的时候非常多的工作都是在写这些类，和实现里面的方法。而偏偏就有那么8种基本类型和他们不一样。以至于让你来回在这两种之间转换，这是很让人头疼的事情。Java中int，long，char这样的类型不是对象型。因此java里提供了一种叫做包装类（wrapper）的东西，使基本类型，有着相应的对象类型Integer，Long，Character等。这样就可以，先把基本类型的东西，转成对象来用，然后再转回去。来来回回，千锤百炼。到了jdk5.0的时候呢，就不用了。看下面的例子：实践

```
public class Test1 { public static void main(String[] args) { // 装箱
int i = 0. Integer integer = i.//i这么一个基本类型的数，可以赋值给Integer型的变量 // 简单的拆箱
int j = integer. //integer这种原始类型的数，也能赋值给j这个原始类型的变量
Integer counter = 1. // 装箱
int counter2 = counter. // 拆箱
while (counter < 10) {
System.out.println("计数 " + counter). //看啊，counter这个对象型的数，还能自动增加
} } }
```

在幕后JVM已经自动执行了转换，同理Boolean和boolean之间也可以，自动拆箱装箱。但是，Integer和int毕竟还是有着不同的。看下面例子：

```
public class Test2 { public static void main(String[] args) { Integer i1 = 256.
Integer i2 = 256. if (i1 == i2) System.out.println("相等!"). else
System.out.println("不相等!"). } }
```

结果输出的是“不相等！”，两个对象比较，它们在内存中开辟的是两个地址怎么能相等

呢？警告：你可千万不能依赖这个结果，请把i1和i2的值，改成100。（请看Test3.java）看看什么结果，令人惊讶的是改了个数，居然输出了“相等！”。这是因为JVM可以选择要尝试这段代码的最佳优化，并对两个Integer对象使用一个实例，这样的话“==”就会返回true了。在自动装箱时对于值从128到127之间的值，使用一个实例。这种装箱与拆箱机制对，程序流程控制语句，也有很大影响：

```
public class Test4 {
public static void main(String[] args) { Boolean arriving = true.
Integer peopleInRoom = 0. int maxCapacity = 100. while
(peopleInRoom if (arriving) { System.out.printf("很高兴见到
你.%d号先生\n",peopleInRoom). peopleInRoom .} else {
peopleInRoom--. } }}}
```

另外一个从unboxing获得好处的语句是switch.在jdk5.0之前的JVM，switch接受int、short、character或者byte值，而在unboxing的操作中，你现在也可以为它赋予新引入的enum之外的Integer，Short，Char以及Byte值。Enum的值，我们在后面的教程会详细讲述。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