

# 52从零开始学Java52之包装类的装箱与拆箱问题

前言

配套开源项目资料

一. 装箱

1. 概念

2. 案例

二. 拆箱

1. 概念

2. 案例

3. 小结

三. 结语

作者：孙玉昌，昵称【**一一哥**】，另外【**壹壹哥**】也是我哦

千锋教育高级教研员、CSDN博客专家、万粉博主、阿里云专家博主、掘金优质作者

## 前言

在上一篇文章中，**壹哥**给大家介绍了Java中包装类及其用法，但是有些内容在上一篇文章中还没有讲清楚，接下来我会在这篇文章中给大家继续讲解包装类的装箱和拆箱问题。你可能会很好奇，做java开发，怎么还装起箱子来了？那么就请大家带着疑惑往下看吧。

-----前戏已做完，精彩即开始-----

全文大约【**1200**】字，不说废话，只讲可以让你学到技术、明白原理的纯干货！本文带有丰富的案例及配图视频，让你更好地理解 and 运用文中的技术概念，并可以给你带来具有足够启迪的思考.....

# 配套开源项目资料

Github:

d/LearnJava

GitHub – SunLtd/LearnJava

Contribute to SunLtd/LearnJava development by creating an account on GitHub.

GitHub

Gitee:



一一哥/从零开始学Java

从零开始学Java系列 稀土掘金专栏地址: <https://juejin.cn/column/7175082165548351546> CSDN专...

Gitee

## 一. 装箱

### 1. 概念

所谓的**装箱(Boxing)**，**其实就是将基本类型转换成包装类**，例如把double变成Double。装箱分为手动装箱和自动装箱(Auto Boxing)，如果我们手动实例化一个包装类，这就是手动装箱。在JDK 1.5之前，必须进行手动装箱；JDK 1.5之后，可以进行自动装箱，不用再进行手工操作，这就简化了开发，提供了方便。

### 2. 案例

下面的案例是手动装箱和自动装箱的实现过程。在JDK 1.5前后，装箱有手动装箱和自动装箱之分，大家要注意。

```
1  /**
2   * @author 一一哥Sun
3   * V我领资料: syc_2312119590
4   * 各平台都有壹哥的同名博客哦
5   */
6  public class BoxTest {
7
8      public static void main(String[] args) {
9          //JDK 1.5之前的拆装箱过程--手动拆装箱
10         byte b = 10;
11         //手动装箱
12         Byte b1 = Byte.valueOf(b);
13         System.out.println("手动装箱后的结果, b1="+b1);
14
15         //手动装箱的另一种形式
16         int m = 100;
17         Integer obj = new Integer(m);
18
19         //手动拆箱的另一种形式
20         int n = obj.intValue();
21         System.out.println("n = " + n);
22
23         //JDK 1.5之后的拆装箱过程--自动拆装箱
24         int i = 10;
25         //自动装箱
26         Integer i001 = i;
27         System.out.println("自动装箱后的结果, i001="+i001);
28     }
29 }
```

## 二. 拆箱

### 1. 概念

所谓的**拆箱(Unboxing)**，**其实就是将包装类型转换成基本类型**，例如把Double变成double。同样的，拆箱也分为手动拆箱和自动拆箱(Auto Boxing)，手动实例化操作包装类，就是手动拆箱。在JDK 1.5之前必须手动进行拆箱，JDK 1.5之后就可以自动拆箱了。

### 2. 案例

下面的案例是手动拆箱和自动拆箱的实现过程。在JDK 1.5前后，拆箱有手动拆箱和自动拆箱之分，大家要注意。

```
Java | 复制代码

1  /**
2   * @author 一一哥Sun
3   * V我领资料: syc_2312119590
4   * 各平台都有壹哥的同名博客哦
5   */
6  public class BoxTest {
7
8      public static void main(String[] args) {
9          //JDK 1.5之前的拆装箱过程--手动拆装箱
10         byte b = 10;
11         //手动装箱
12         Byte b1 = Byte.valueOf(b);
13         System.out.println("手动装箱后的结果, b1="+b1);
14         //手动拆箱
15         byte b2 = b1.byteValue();
16         System.out.println("手动拆箱后的结果="+b2);
17
18         //JDK 1.5之后的拆装箱过程--自动拆装箱
19         int i = 10;
20         //自动装箱
21         Integer i001 = i;
22         System.out.println("自动装箱后的结果, i001="+i001);
23         //自动拆箱
24         int i002 = i001;
25         System.out.println("自动拆箱后的结果, i002="+i002);
26     }
27 }
```

### 3. 小结

我们要注意，自动装箱和自动拆箱只发生在编译阶段，其目的是为了减少代码的编写。从性能上来看，装箱和拆箱的过程会影响代码的执行效率，因为编译后的class代码是严格区分基本类型和引用类型的，所以编译阶段的装箱和拆箱，在编译后还是要回归原始。另外在自动拆箱时，如果赋值不当，有可能会产生NullPointerException异常，如下所示：

```
1 Integer n = null;
2 //这里就会产生NPE空指针异常
3 int m = n;
```

-----正片已结束，来根事后烟-----

### 三. 结语

至此，壹哥就把Java中的常用包装类给大家介绍完毕了，现在你知道包装类和基本类型有什么联系和区别了吗？本文重点总结如下：

- 将基本类型变成包装类就是装箱，把包装类变成基本类型就是拆箱；
- 装箱和拆箱都分为手动和自动两种情况；
- 自动装箱和自动拆箱都是在编译期完成的(JDK 1.5之后出现自动拆装箱)；
- 装箱和拆箱会影响执行效率，且拆箱时可能发生NullPointerException异常；

另外如果你独自学习觉得有很多困难，可以加入壹哥的学习互助群，大家一起交流学习。