

区分引用类型和原始类型 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/252/2021\\_2022\\_\\_E5\\_8C\\_BA\\_E5\\_88\\_86\\_E5\\_BC\\_95\\_E7\\_c104\\_252373.htm](https://www.100test.com/kao_ti2020/252/2021_2022__E5_8C_BA_E5_88_86_E5_BC_95_E7_c104_252373.htm)

Java 提供两种不同的类型：引用类型和原始类型（或内置类型）。另外，Java 还为每个原始类型提供了封装类(Wrapper)。如果需要有一个整型变量，是使用基本的 int 型呢，还是使用 Integer 类的一个对象呢？如果需要声明一个布尔类型，是使用基本的 boolean，还是使用 Boolean 类的一个对象呢？本文可帮助您作出决定。下面列出了原始类型以及它们的对象封装类。原始类型封装类 boolean Boolean char Character byte Byte short Short int Integer long Long float Float double Double

引用类型和原始类型的行为完全不同，并且它们具有不同的语义。例如，假定一个方法中有两个局部变量，一个变量为 int 原始类型，另一个变量是对一个 Integer 对象的对象引用：`int i = 5. // 原始类型 Integer j = new Integer(10). // 对象引用` 这两个变量都存储在局部变量表中，并且都是在 Java 操作数堆栈中操作的，但对它们的表示却完全不同。（本文中以下部分将用通用术语堆栈代替操作数堆栈或局部变量表。）原始类型 int 和对象引用各占堆栈的 32 位。（要表示一个 int 或一个对象引用，Java 虚拟机实现至少需要使用 32 位存储。）Integer 对象的堆栈项并不是对象本身，而是一个对象引用。Java 中的所有对象都要通过对象引用访问。对象引用是指向对象存储所在堆中的某个区域的指针。当声明一个原始类型时，就为类型本身声明了存储。前面的两行代码表示如下：引用类型和原始类型具有不同的特征和用法，它们包括：大小和速度问题，这种类型以哪种类型

的数据结构存储，当引用类型和原始类型用作某个类的实例数据时所指定的缺省值。对象引用实例变量的缺省值为 null，而原始类型实例变量的缺省值与它们的类型有关。许多程序的代码将同时包含原始类型以及它们的对象封装。当检查它们是否相等时，同时使用这两种类型并了解它们如何正确相互作用和共存将成为问题。程序员必须了解这两种类型是如何工作和相互作用的，以避免代码出错。例如，不能对原始类型调用方法，但可以对对象调用方法：

```
int j = 5.j.hashCode(). // 错误//...Integer i = new Integer(5).i.hashCode(). // 正确
```

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)