

面向对象编程：Java复杂数据类型用法 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/145/2021\\_2022\\_\\_E9\\_9D\\_A2\\_E5\\_90\\_91\\_E5\\_AF\\_B9\\_E8\\_c104\\_145423.htm](https://www.100test.com/kao_ti2020/145/2021_2022__E9_9D_A2_E5_90_91_E5_AF_B9_E8_c104_145423.htm) 在这篇文章中，我们将讨论Java语言中的复杂数据类型。其内容涉及到初始化过程、默认初值和某些针对复杂数据类型的操作。索引同简单数据类型的定义一样，Java虚拟机（JVM）还定义了索引

（reference）这种数据类型。索引类型可以“引用”变量，由于Java没有明确地定义指针类型，所以索引类型可以被认为就是指向实际值或者指向变量所代表的实际值的指针。一个对象可以被多于一个以上的索引所“指”。JVM从不直接对对象寻址而是操作对象的索引。索引类型分成三种，它们是：类（class）、接口（interface）和数组（array）。索引类型可以引用动态创建的类实例、普通实例和数组。索引还可以包含特殊的值，这就是null索引。null索引在运行时上并没有对应的类型，但它可以被转换为任何类型。索引类型的默认值就是null。类（Class）指的是定义方法和数据的数据类型。从内部来看，JVM通常把class类型对象实现为指向方法和数据的一套指针。定义class类型的变量只能引用类的实例或者null，如以下代码所示：  

```
MyObject anObject = new MyObject(). // 合法
MyObject anotherObject = null. // 合法
MyObject stillAnotherObject = 0. // 非法
```

接口（interface）好比一种模版，这种模版定义了对象必须实现的方法，其目的就是让这些方法可以作为接口实例被引用。接口不能被实例化。类可以实现多个接口并且通过这些实现的接口被索引。接口变量只能索引实现该接口的类的实例。比方说，假

设我们定义了一个接口，名字是Comparable，同时还定义了一个类SortItem，这个类实现了接口Comparable，那么我们可以编写以下的定义代码：`Comparable c = new SortItem()`。假如Comparable接口定义了一个方法：`public void compare(Comparable item)`，那么SortItem类就必须提供compare方法的实现，如以下代码所示：`public class SortItem implements Comparable{ public void compare(Comparable item) { ...method implementation here }}`

数组 Java数组（array）是动态创建的索引对象，这一点和类非常相似，此外，同类一样，数组只能索引数组的实例或者null，如以下代码所示：`int[] myIntArray = new int[5].int[] anotherIntArray = null`。数组是Object类的继承，这样，Object类的所有方法都可以被数组调用。数组对象由元素组成。元素的数目也可以为0，在这种情况下称做数组为空。所有的数组都是从0开始对元素编号的，这意味着数组内的第1个元素的索引编号是数字0。所有对数组元素的访问都会在运行时上接受检查，如果试图使用编号小于0或者大于数组长度来索引元素就会产生ArrayIndexOutOfBoundsException异常并被抛出。

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)