

Java语言的接口与类型安全 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/138/2021_2022_Java_E8_AF_AD_E8_A8_80_c97_138802.htm 什么是接口？Java中的接口是一系列方法的声明，是一些方法特征的集合，一个接口只有方法的特征没有方法的实现，因此这些方法可以在不同的地方被不同的类实现，而这些实现可以具有不同的行为（功能）。接口的两种含义：一，Java接口，Java语言中存在的结构，有特定的语法和结构；二，一个类所具有的方法的特征集合，是一种逻辑上的抽象。前者叫做“Java接口”，后者叫做“接口”。在Java语言规范中，一个方法的特征仅包括方法的名字，参数的数目和种类，而不包括方法的返回类型，参数的名字以及所抛出来的异常。在Java编译器检查方法的重载时，会根据这些条件判断两个方法是否是重载方法。但在Java编译器检查方法的置换时，则会进一步检查两个方法（分处超类型和子类型）的返还类型和抛出的异常是否相同。接口继承和实现继承的规则不同，一个类只有一个直接父类，但可以实现多个接口。Java接口本身没有任何实现，因为Java接口不涉及表象，而只描述public行为，所以Java接口比Java抽象类更抽象化。Java接口的方法只能是抽象的和公开的，Java接口不能有构造器，Java接口可以有public，静态的和final属性。接口把方法的特征和方法的实现分割开来。这种分割体现在接口常常代表一个角色，它包装与该角色相关的操作和属性，而实现这个接口的类便是扮演这个角色的演员。一个角色由不同的演员来演，而不同的演员之间除了扮演一个共同的角色之外，并不要求其它的共同之处。为什么使用接口？

两个类中的两个类似的功能，调用他们的类动态的决定一种实现，那他们提供一个抽象父类，子类分别实现父类所定义的方法。问题的出现：Java是一种单继承的语言，一般情况下，哪个具体类可能已经有了一个超类，解决是给它的父类加父类，或者给它父类的父类加父类，只到移动到类等级结构的最顶端。这样一来，对一个具体类的可插入性的设计，就变成了对整个等级结构中所有类的修改。接口是可插入性的保证。在一个等级结构中的任何一个类都可以实现一个接口，这个接口会影响到此类的所有子类，但不会影响到此类的任何超类。此类将不得不实现这个接口所规定的方法，而其子类可以从此类自动继承这些方法，当然也可以选择置换掉所有的这些方法，或者其中的某一些方法，这时候，这些子类具有了可插入性（并且可以用这个接口类型装载，传递实现了他的所有子类）。我们关心的不是那一个具体的类，而是这个类是否实现了我们需要的接口。接口提供了关联以及方法调用上的可插入性，软件系统的规模越大，生命周期越长，接口使得软件系统的灵活性和可扩展性，可插入性方面得到保证。类型使用Java接口将软件单位与内部和外部耦合起来。使用Java接口不是具体的类进行变量的类型声明，方法的返还类型声明，参量的类型声明，以及数据类型的转换。在理想的情况下，一个具体的Java类应当只实现Java接口和抽象Java类中声明的方法，而不应当给多余方法。类型等级结构Java接口（以及抽象类）一般用来作为一个类型的等级结构的起点。如果一个类已经有了一个主要的超类型，那么通过实现一个接口，这个类可以拥有另一个次要的超类型，这种次要的超类型叫做混合类型。Java接口常用方法单方法接

□public interface ActionListener() { public abstract void
actionPerformed(ActionEvent event).}仅且只有一个方法，只有
实现了这个接口（重写这个接口中的唯一一个方法），你才
有资格去事件监听器列表里注册（参数为ActionListener类型）
，当事件源变动时，自动调用这个唯一的actionPerformed方
法.标识接口 是没有任何方法和属性的接口。标识接口不对实
现它的类有任何语意上的要求，它仅仅表明了实现它的类属
于一个特定的类型（传递）。不推荐过多的使用标识接口。
常量接口 用Java接口来声明一些常量，然后由实现这个接
口的类使用这些常量（以前在做画板的时候这么干过）。建议
不要模仿这种常量接口的做法。Java语言类型安全问题 Java是
强类型的语言。这意味着Java编译器会对代码进行检查，以确
定没一次赋值，每一次方法的调用是符合类型的。如果有任
何不相符合的情况，Java编译器就会给出错误。类型检查是基
于这样一个简单的事实：每一变量的声明都给这个变量一个
类型；每一个方法包括构造器的声明都给这个方法的特征。
这样一来，Java编译器可以对任何的表达式推断出一个明显类
型，Java编译器可以基于明显类型对类型进行检查。Java语言
是类型安全的。这就是说，任何被Java编译器接受的合法
的Java类保证是类型安全的。换言之，在程序运行期间，不
会有任何类型的错误。一个Java程序根本不可能将一个本来属
于一个类型的变量当作另一个类型处理，因此也就不会产生由
此而引起的错误。简单的说，Java语言依靠三种机制做到了
类型安全：编译期间的类型检查，自动的存储管理，数组的
边界检查。 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