

java初学者实践教程18 - 抽象类和接口 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_270980.htm

Java语言中允许有一种叫做抽象方法的东西，他只是一个名字没有具体的实现。像是这样：
`public abstract void abc () ;` 使用了abstract关键字，结尾用“
;”结束。与前几节我们用的方法都是具体方法，是有实现的。哪怕方法体中什么也不写
`public void abc () { }`也是具体方法。概念：包含一个或多个抽象方法的类称为抽象类。抽象类也必须声明abstract关键字。抽象类的使用有着一些限制，不能创建抽象类的实例。如果子类实现了抽象方法，则可以创建该子类的实例对象。要是子类也不实现的话，这个子类也是抽象类，也不能创建实例。接口是什么东西呢？接口是比抽象类更抽象的类。举例：
`public interface Name { }`接口里面的方法全都是抽象的，里面的变量全都是final的常量，而且实现接口的类必须将所有的抽象方法全部实现。抽象类里也可以有具体的方法。所以说，接口是最抽象的，其次是抽象类，而具体类本身就是对现实世界的抽象。软件开发本身就是将现实世界抽象成计算机世界。因为抽象类和接口比具体类抽象，所以使用时他们总是被继承而被实现的。不过继承他们的类不只是一个，有很多类实现他们的抽象方法。一个方法有多种实现方式，这里用到了OOP中的多态性。这使得设计变得非常清晰。因为基类是抽象类或是接口做一个描述，底下继承的类有若干个，我们只需要对接口或抽象类操作，也用不着管有多少个实现。如果是多人共同开发的项目的话，是非常有意义的。你自己写个东西，怎么实现的也

不用告诉别人，别人看个接口就够了。接口的实现用关键字implement而不是extends.如果用了extends的那就是继承这个接口。那么那个子类也是接口，是原来的子接口。举个接口的例子吧：实践：//声明一个接口 public interface Say { public void sayMessage().} //两个实现类 public class SayHello implements Say { public void sayMessage() { System.out.println("hello"). }} public class SayHi implements Say { public void sayMessage() { System.out.println("Hi"). }} //这是一个测试类 public class TestSay { public static void main(String[] args) { //同样都是用Say这个接口类型实例，却可以输出两个结果 Say say = new SayHello(). say.sayMessage(). Say say1 = new SayHi(). say1.sayMessage(). }} 接口还有一个重要的作用，我们在面向对象那节课里提过一个概念，java语言中只有单继承，就是说只能从一个父类继承。单继承的好处是，一旦继承的太多了，改了一个类子类就都变了。牵一发，而动全身。那么如果我们想继承多个父类的特性怎么办呢？就用接口吧，这个类可以先继承一个类，再去实现其它的接口，接口里面都是抽象方法，不会造成，牵一发，而动全身的效应。改变多继承的特性，也是对C语言的一项改进 业界有一种说法说，与其说java是面向对象编程，还不如说它是面向接口编程。强调的方面是接口的抽象描述性。它也是对C的一种改进，C里面没有接口。所以说java语言适合多人团队合作的大项目，看一个接口就可以了，后面怎么实现的可以不管。 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