

对Java程序多态性支持的改进 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_AF_B9Java_E7_A8_8B_c104_252505.htm 多态性(polymorphism)是面向对象编程的属性，它允许多个方法使用同一个接口。Java从多个方面支持多态性，其中两个方面最为突出。第一个是每个方法(标记为final的方法除外)都可以被子类重写.第二个是设立interface关键字。下面将给出这两方面的详细介绍。由于超类中的方法可以在派生类中重写，因此创建类的层次结构非常简单。在类的层次结构中，每个子类都将它的超类特化(specialization)。大家知道，超类的一个引用可以引用它的任何一个子类，而且通过超类的引用调用某子类对象的一个方法时，会自动执行由该子类重写后的版本。因此，可以用超类来定义对象的形式并提供对象的默认实现，而子类根据这种默认实现进行修改，以更好地适应具体情况的要求。因此，在超类中定义的一个接口可以作为多个不同实现的基础。当然，Java进一步采取了“一个接口，多个方法”的概念。它定义了interface关键字，这样就可以将类的方法和类的实现完全分离。尽管接口是抽象的，但是仍然可以声明接口类型的引用。这个概念非常重要，因为它可以改进多态性的应用。只要某个类实现一个接口，并且该接口提供了某种功能，那么任何需要这种功能的代码都可以使用这个类的对象。例如，假设某个接口的名称为MyIF，考虑下面的方法：`void myMeth(MyIF ob) { // ... }`任何实现了MyIF接口的对象都可以传递给myMeth()方法。该对象的其他功能无需考虑。myMeth()方法可以对任何实现了MyIF接口的对象进行操作

。 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