彻底学习Java语言中的覆盖和重载的使用 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/257/2021_2022__E5_BD_BB_E5_BA_95_E5_AD_A6_E4_c67_257039.htm override可以翻译为覆盖，从字面就可以知道，它是覆盖了一个方法并且对其重写，以求达到不同的作用。对我们来说最熟悉的覆盖就是对接口方法的实现，在接口中一般只是对方法进行了声明，而我们在实现时，就需要实现接口声明的所有方法。除了这个典型的用法以外，我们在继承中也可能会在子类覆盖父类中的方法。在覆盖要注意以下的几点： 1、覆盖的方法的标志必须要和被覆盖的方法的标志完全匹配，才能达到覆盖的效果； 2、覆盖的方法的返回值必须和被覆盖的方法的返回一致； 3、覆盖的方法所抛出的异常必须和被覆盖方法的所抛出的异常一致，或者是其子类； 4、被覆盖的方法不能为private，否则在其子类中只是新定义了一个方法，并没有对其进行覆盖。 overload对我们来说可能比较熟悉，可以翻译为重载，它是指我们可以定义一些名称相同的方法，通过定义不同的输入参数来区分这些方法，然后再调用时，VM就会根据不同的参数样式，来选择合适的方法执行。在使用重载要注意以下的几点： 1、在使用重载时只能通过不同的参数样式。例如，不同的参数类型，不同的参数个数，不同的参数顺序（当然，同一方法内的几个参数类型必须不一样，例如可以是fun(int, float)， 但是不能为fun(int, int)）； 2、不能通过访问权限、返回类型、抛出的异常进行重载； 3、方法的异常类型和数目不会对重载造成影响； 4、对于继承来说，如果某一方法在父类中是访问权限是priavte，那么就不能

在子类对其进行重载，如果定义的话，也只是定义了一个新方法，而不会达到重载的效果。 下面是对override和overload的测试程序，其中注释中的内容都是会产生编译错误的代码，我们将注释去掉，看看在编译时会产生什么效果。 // 对overload测试的文件：OverloadTest.javapublic class OverloadTest {// 下面几个方法用来验证可以通过定义不同的参数类型和参数的数目进行方法重载。public void fun(){System.out.println("method fun in OverloadTest, no parameter").}public void fun(float f) {System.out.println("method fun in OverloadTest, parameter type: float").}public void fun(int i){System.out.println("method fun in OverloadTest, parameter type: int").}public void fun(int i1, int i2) {System.out.println("method fun in OverloadTest, parameter type: int, int").}// 下面的两个方法用来验证可以通过定义不同的参数顺序进行方法重载。// 需要注意：这里的参数肯定不是相同的类型，否则的顺序的先后就毫无意义。public void fun1(int i, float f) {System.out.println("method fun1 in OverloadTest, sequence of parameters is: int, float").}public void fun1(float f, int i) {System.out.println("method fun1 in OverloadTest, sequence of parameters is: float, int").}// 下面的两个方法用来验证方法抛出的异常对于重载的影响.// 无论是异常的类型还是异常的个数都不会对重载造成任何的影响。public void fun2() throws TestException {System.out.println("fun2 in OverloadTest, exception: TestException").}public void fun2(int i) throws TestException, TestException1 {System.out.println("fun2 in OverloadTest, exception: TestException, TestException1").}public

void fun2(float f) throws Exception {System.out.println("fun2 in OverloadTest, exception: Exception").}// 不能通过抛出的异常类型来重载fun方法。//public void fun(int i) throws Exception {// System.out.println("method fun in OverloadTest, parameter type: int, exception: Exception").//}// 不能通过返回值重载fun方法。//public boolean fun(int i) throws Exception {// System.out.println("method fun in OverloadTest, parameter type: int, exception: Exception, return: boolean").// return true.//}private void fun3() { }// 不能通过不同的访问权限进行重载public void fun3() { }public static void main(String[] args) {// 这里只是定义了OverloadTest的实例，所以test不会调用// OverloadTest1中的方法。OverloadTest test = new OverloadTest1().// 这里定义了OverloadTest1的实例，因为OverloadTest1是OverloadTest// 的子类，所以test1会调用OverloadTest中的方法。OverloadTest1 test1 = new OverloadTest1().try {int i = 1, j = 2, m = 3.// 这里不会调用OverloadTest1的fun方法// test.fun(i, m, j).test1.fun(i, j, m).test1.fun().// 这个调用不会执行，因为fun3()在OverloadTest中访问权限是priavte //test1.fun3().test1.fun3(i).} catch(Exception e) { }}} 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