

子类覆盖测试

```
class SuperClass {
    int test() {
        System.out.println("test in SuperClass");
        return 0;
    }
}

class SubClass extends SuperClass {
    @Override
    int test() {
        System.out.println("test in SubClass");
        return 1;
    }
}

public class OverrideTest {
    public static void main(String[] args) {
        SubClass sub = new SubClass();
        sub.test(); //test in SubClass
        SuperClass super_00 = (SuperClass)sub;
        super_00.test(); //test in SubClass
        SuperClass super_01 = new SubClass();
        super_01.test(); //test in SubClass
        SuperClass super_02 = new SuperClass();
        super_02.test(); //test in SuperClass
    }
}
```

分析: 在 main 方法中, 我们测试了各种将子类转型为父类对象的情形, 发现只要新建对象时是 `new SubClass()`, 则无论是否转换类型, 始终输出 `test in SubClass`, 这意味着子类始终覆盖了父类方法, 无论指向它的引用声明的是父类类型还是子类类型