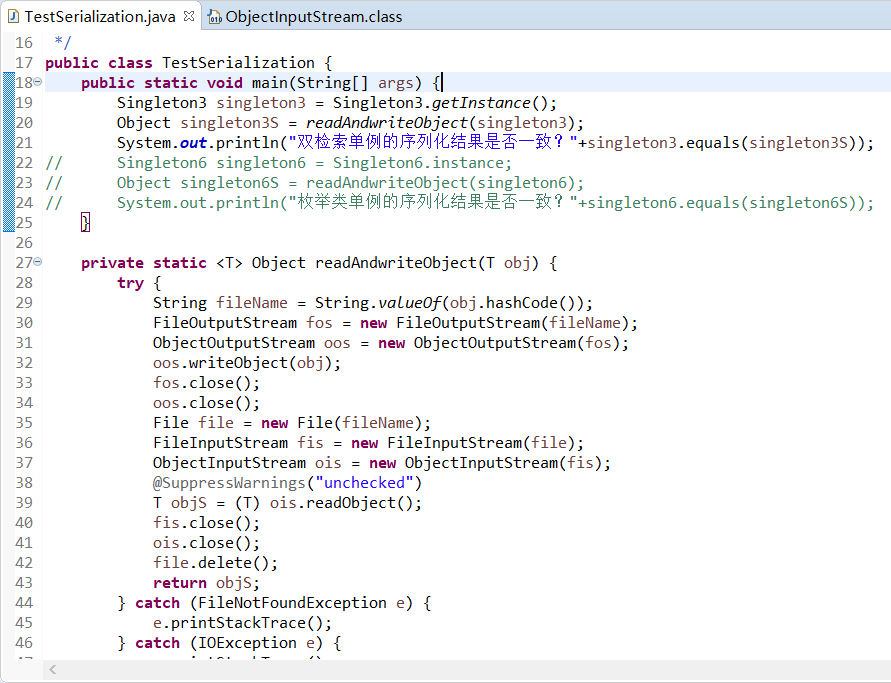
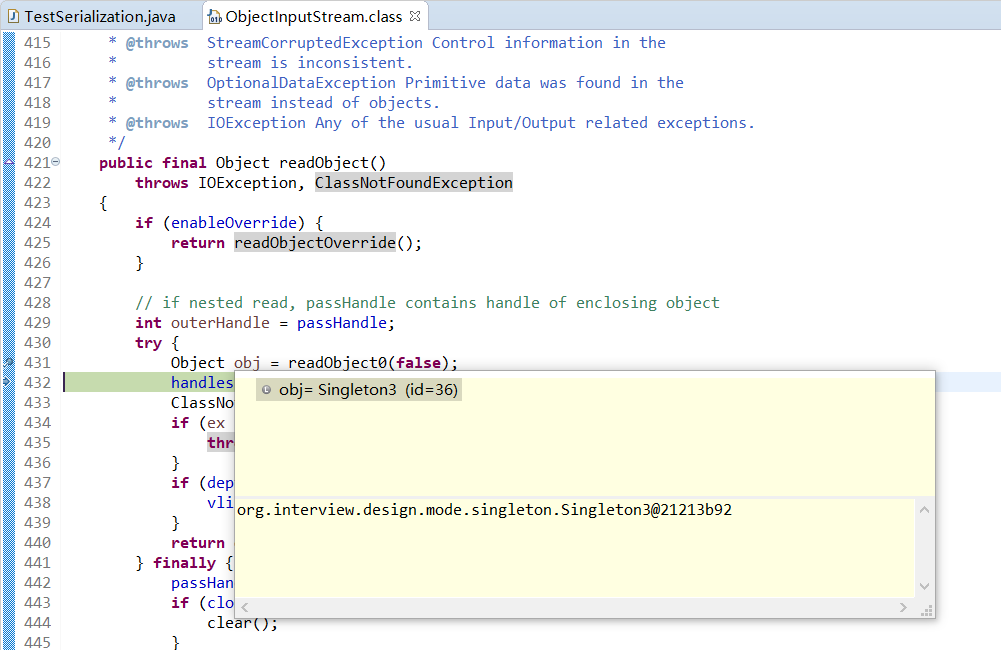
测试代码编写

上面测试的是双检锁类的单例序列化和反序列化，下面的是枚举类的单例



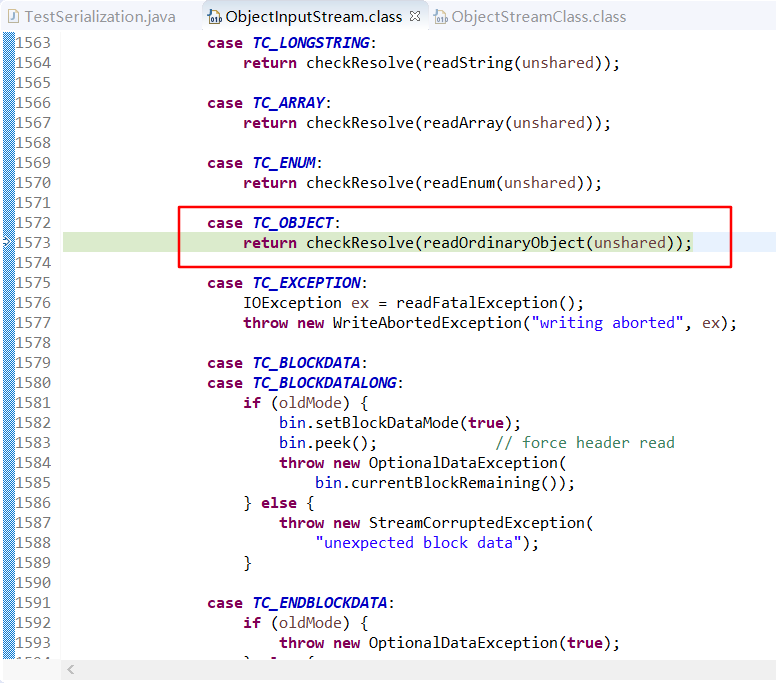
#### 双检锁单例序列化

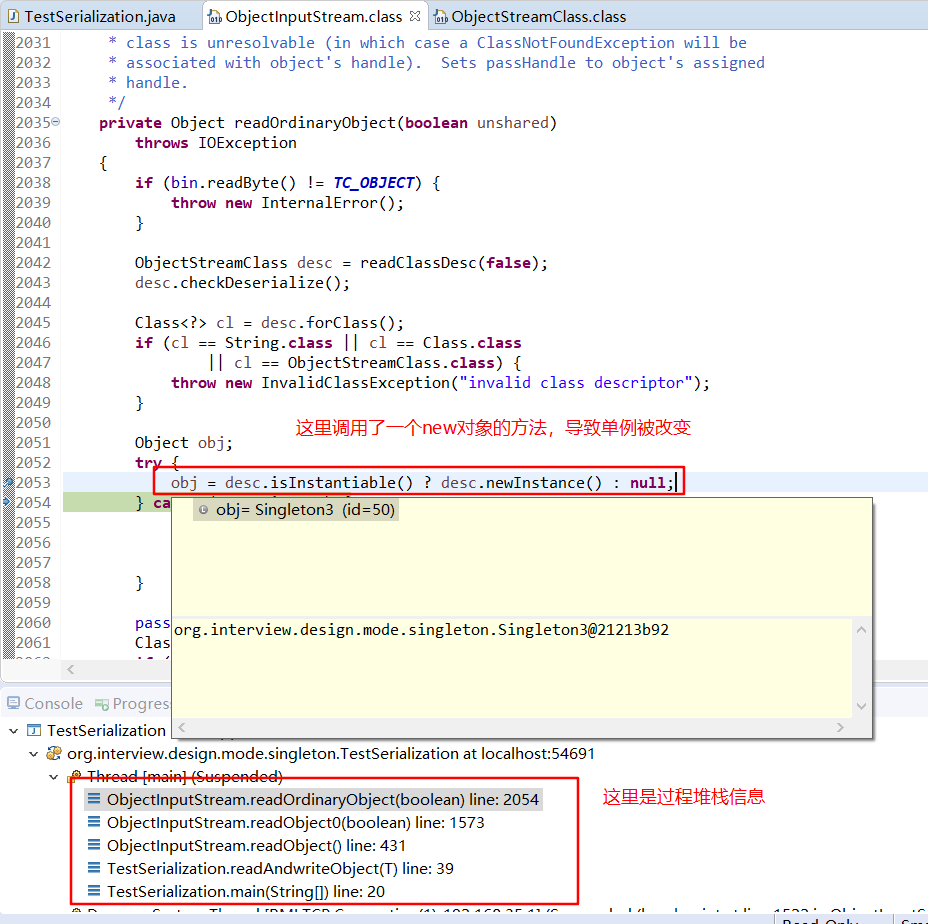
可以看到，普通单例会创建相关的对象，属性信息是无法直接看到，显示的是一串内存地址



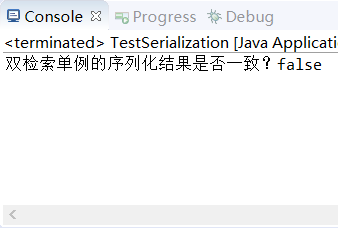
重新debug，找到具体调用的地方，可以看到新建了一个对象，导致单例被改变

对应的调用栈：readObject--->readObject0--->case TC\_OBJECT--->readOrdinaryObject



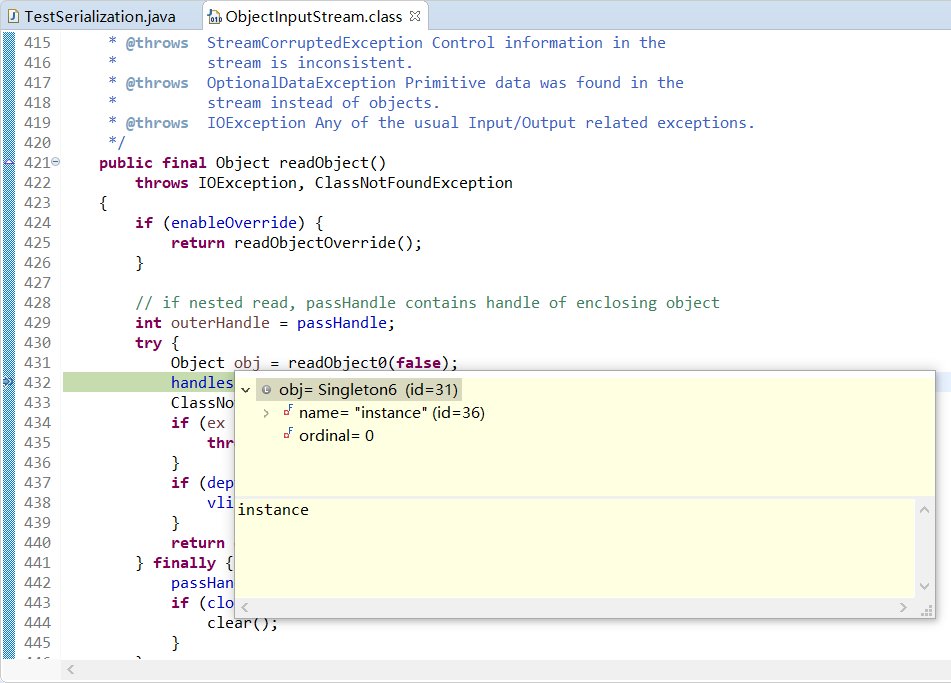


双检锁单例序列化结果



#### 枚举类序列化

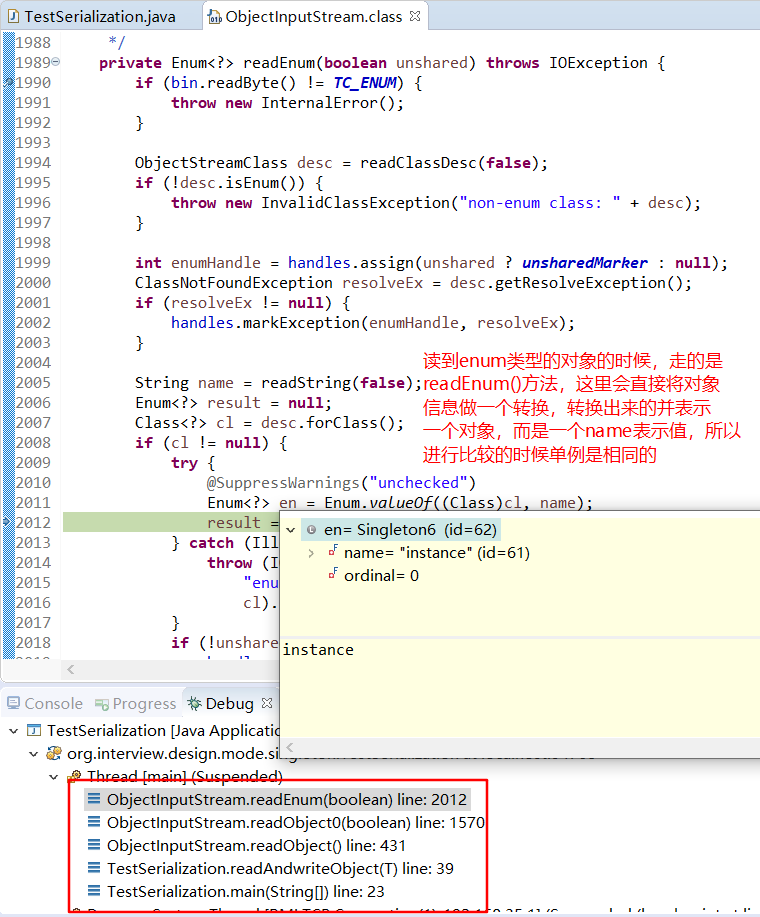
可以看到，普通单例会创建相关的对象，属性信息是直接可以看到



重新debug，进入的方法是readEnum

对应的调用栈：readObject--->readObject0--->case TC\_ENUM--->readEnum





枚举类单例序列化结果

