**SA思考题11**

**1、查阅、学习并举例说明如何利用Java基础类库实现观察者模式的功能。**

**答：**

**观察者模式：**当一个对象的状态发生改变时，需要同时通知一个或多个对象更新自己的行为。

在 Java 程序中，类 Observable 和接口 Observer 的最大功能就是实现观察者模式。

**1）情景举例**

现在房地产调控比较严格，很多购房者都在关注着房子的价格变化，每当房子价格变化时，所有的购房者都可以观察得到。实际上以上的购房者都属于观察者，他们都在关注着房子的价格。这个观察变化的过程就可以成为观察者模式。

**2）方法介绍**

在 Java 中可以直接依靠类库中的Observable类和Observer接口实现以上观察者模式的功能。在Java 应用中，需要被观察的类必须继承于 Observable 类。每个观察者类都需要实现 Observer 接口。

**Observable类定义了观察者关注的主题行为**，我们只需要继承Observable类就可以达到复用的效果，不用自己定义抽象类或接口，Observable类中有三个方法需要我们了解：

1. 添加观察者的方法addObserver(Observer o)
2. 标记我们继承类Observable的主题类对象为已改变的对象的setChanged()方法
3. 通知我们的观察者的方法notifyObservers()

**我们的观察者需要实现一个Observer接口**，当观察者关注的主题状态发生改变的时候，就会通知我们的观察者，也就是调用观察者的update(Observable o, Object arg)方法。

**3）示例项目**

**第一步：创建通知者类**

House类具有房价属性，默认初始值为100000，继承类Observable，覆盖父类的notifyObservers()方法，在里面调用方法setChanged()，当调用notifyObservers()方法的时候，就会通知所有的观察者，即调用观察者的update()方法。



**第二步：创建观察者类**

实现Observer接口，实现其update()方法。

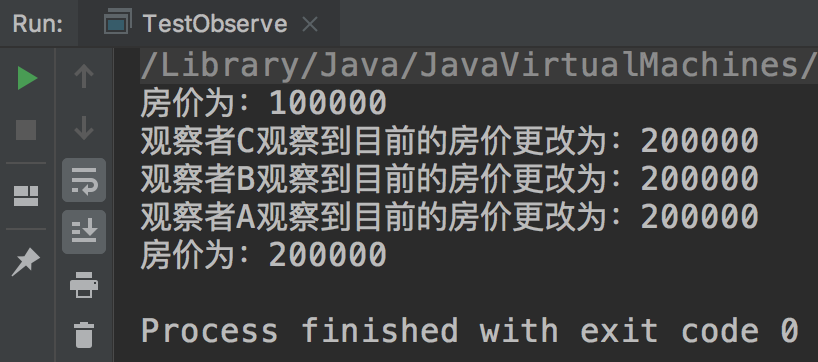


**第三步：编写测试类**

将房价改为200000，并通知观察者



**运行结果**



**补充**

在实际开发中，我们需要通知的观察者可能之前就已经定义好了更新自己状态的方法，方法名可能不是update()，我们的观察者也可能是各种各样的，所定义的更新自己行为的方法名各不一样，而我们的通知类中的方法notifyObservers（）只能调用观察者的update()。此时我们可以使用代理，观察者代码可以修改如下，进而实现功能，update部分代码如下：



**2、LoD原则强调“只和朋友通信，不和陌生人说话”。请举例说明“朋友圈”认定依据是什么？**

**答：**

**迪米特法则(Law of Demeter, LoD)：一个对象应该对其他对象保持最少的了解，又叫最少知道原则(Least Knowledge Principle, LKP)。**

如果两个类不必彼此直接通信，那么这两个类就不应当发生直接的相互作用。如果其中的一个类需要调用另外一个类的某一个方法的话，可以通过第三者转发这个调用。尽量降低类与类之间的耦合，高内聚低耦合的体现，强调只和朋友交流，不和陌生人说话。保持神秘。

**朋友圈的确定：**

**以下的条件是成为朋友的认定依据：**

1.当前对象本身(this)

2.以参量形式传入到当前对象方法中的对象

3.当前对象的实例变量直接引用的对象

4.当前对象的实例变量如果是，那么聚集中的元素也就是朋友

5.当前对象所创建的对象

任何一个对象，如果满足上面的条件之一，就是当前对象的“朋友”；否则就是“陌生人”。

**出现在成员变量、方法输入输出参数中的类称为朋友关系类，而出现在方法体内部的类不属于朋友类**，所以这类实体类也是我们要避免的。

从类上面来说也需要进行自我的保护，也即类里面尽量减少对外公开的非 public 的方法、和非静态的 public 变量、成员变量 private 化，利用包权限和 protected 保护自己。

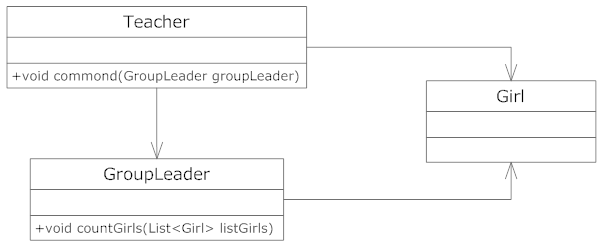
假如有这样一个场景，有一个方法放在A类也行，放在B类也行，那么如何去做？原则是：如果该方法放在本类中既不增加类间关系，也对本类不产生负面影响，那就可以放在本类中。对外部的类的引用越少越好。

**举例**

**1）不符合LoD的反例**

体育老师让体委清点全班女生数目。

**类图**



**代码**

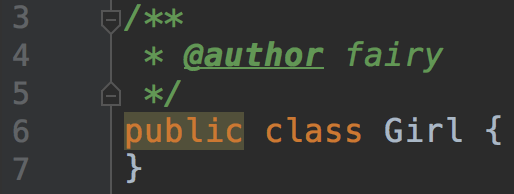
老师-Teacher类



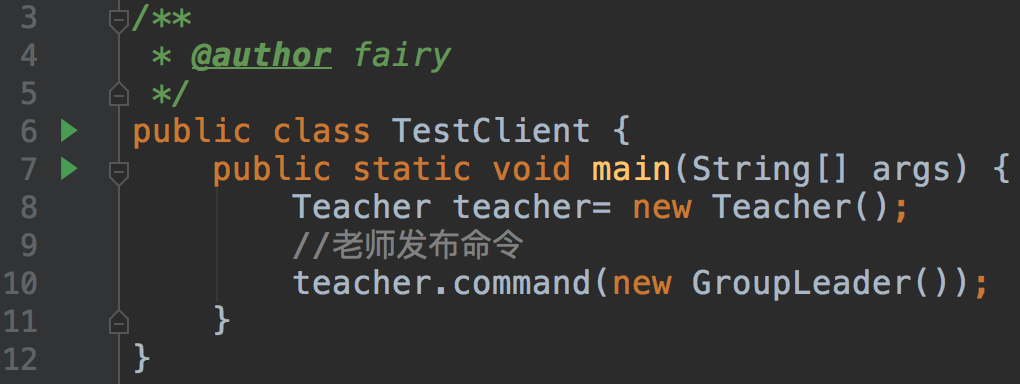
体委-GroupLeader类



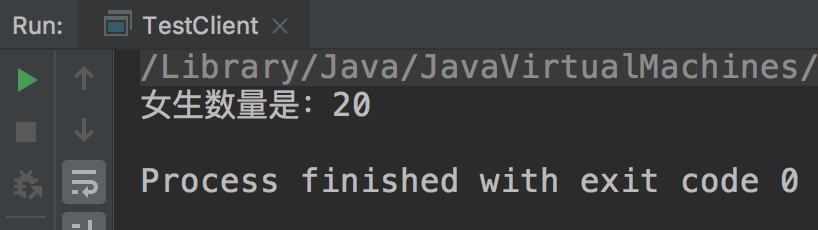
女生-Girl类



编写测试类



运行结果



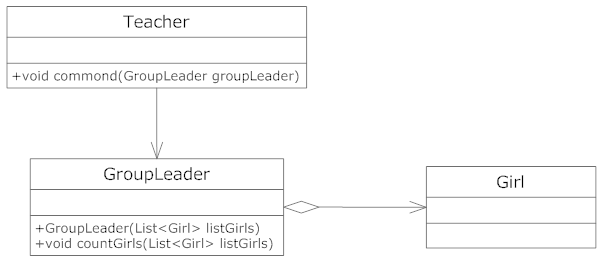
**分析**

根据上面的要求：老师让体委清点女生数目。老师与女生是陌生关系（老师不需要女生执行任何动作）。显然，上述做法违背LoD法则。

**2）依据LoD法则解耦**

与真实意义上的陌生类解耦，这里应为上述类图中未正确体会语义的虚假朋友类Girl

**类图**



**代码**

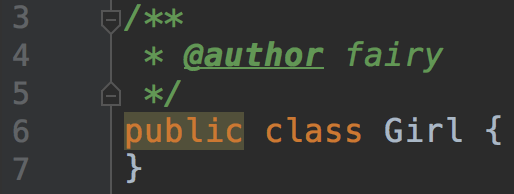
老师-Teacher类



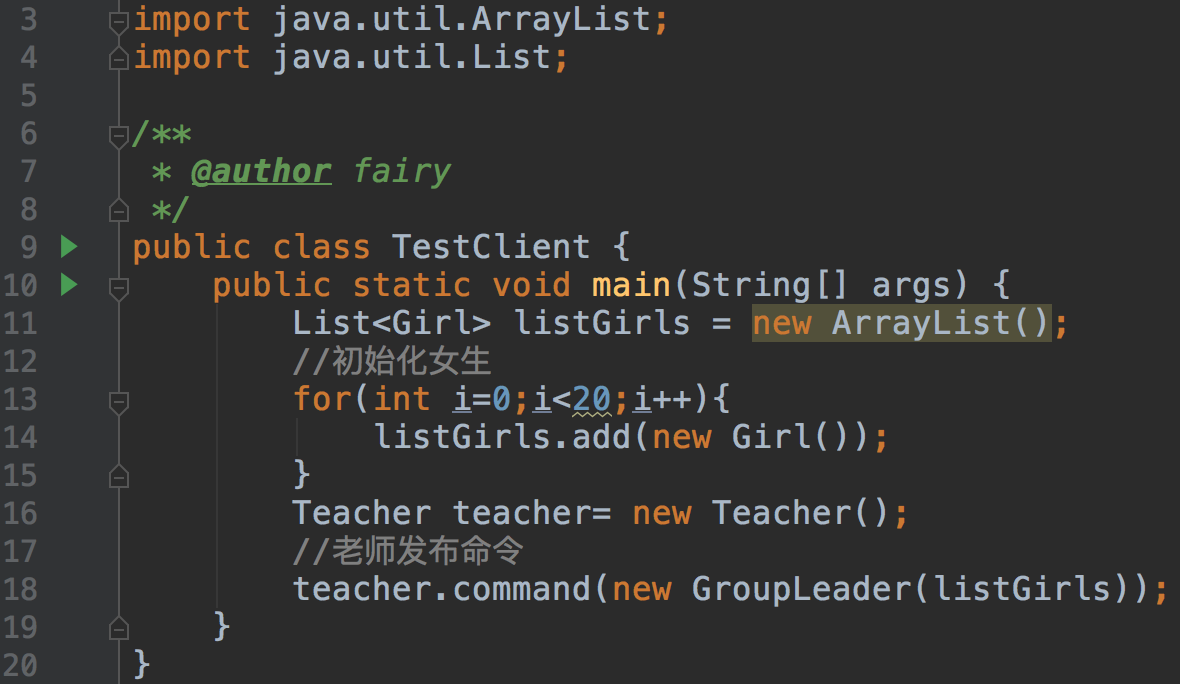
体委-GroupLeader类



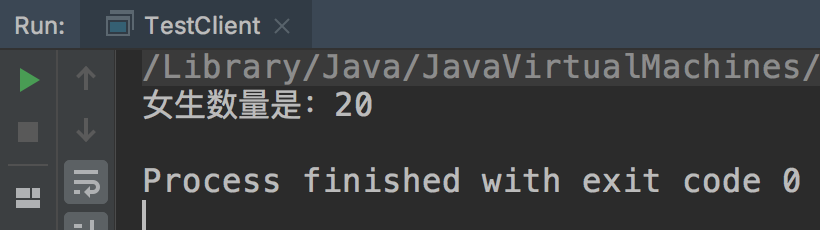
女生-Girl类



编写测试类



运行结果



**分析**

在整个过程中，体育老师只需给体委发布命令，而不必跟每个女生发生联系，符合LoD法则。