设计模式（观察者模式）

作者：Pleuvoir

原文链接：https://ld246.com/article/1522988618521

来源网站：链滴

观察者模式分离了观察者和被观察者自身的责任，让类各自护自己的功能，提高了系统的可重用性。

怎么做?

1. 通过使用JDK中提供的{@link java.util.Observable} 以及 {@link java.util.Observer} 这两个类即完成简单的观察者模式。

代码示例：

```
import java.util.Observable;
import java.util.Observer;

/**
 * 通过使用JDK中提供的{@link java.util.Observable} 以及 {@link java.util.Observer} 这两个类即完成简单的观察者模式。
 *
 * <p>
 * {@link Observable} 主要维护观察者（即被通知对象）， 实现了 {@link Observer} 该接口表明是个观察者，会被强制实现update方法来处理接收到通知后的行为，
 * 当发生改变后，<code>Observable.notifyObservers()</code> 会遍历所有的观察者update方法
 *
 * @author pleuvoir
 *
 */
public class Teacher extends Observable {

    public void homeworkIsComing(String workname) {
        System.out.println("当前有" + countObservers() + "个通知对象");
```

```
        setChanged();
        notifyObservers(workname + "来了");
    }


    public static class Student implements Observer{
        public void update(Observable o, Object arg) {
            Teacher teacher = (Teacher) o;
            System.out.println(this.getClass().getName()  + "收到" + teacher.getClass().getName()
 "通知" + arg.toString());
        }
    }


    public static void main(String[] args) {
        Teacher teacher = new Teacher();
        for (int i = 0; i < 9; i++) {
            teacher.addObserver(new Student());
        }
        teacher.homeworkIsComing("寒假作业");
    }

}
```

结果：

```
当前有9个通知对象
observer.Teacher$Student收到observer.Teacher通知寒假作业来了
observer.Teacher$Student收到observer.Teacher通知寒假作业来了
observer.Teacher$Student收到observer.Teacher通知寒假作业来了
observer.Teacher$Student收到observer.Teacher通知寒假作业来了
observer.Teacher$Student收到observer.Teacher通知寒假作业来了
observer.Teacher$Student收到observer.Teacher通知寒假作业来了
observer.Teacher$Student收到observer.Teacher通知寒假作业来了
observer.Teacher$Student收到observer.Teacher通知寒假作业来了
observer.Teacher$Student收到observer.Teacher通知寒假作业来了
```

结论

1. 观察看上去是一个主动的行为，但是其实观察者不是主动调用自己的业务代码的，相反，是被观察调用的。

2. 基于此的还有更为具体的发布订阅模式和事件机制。