

参与者：

Subject(目标，Subject)：

目标知道它的观察者。可以有任意多个观察者观察同一个目标。

提供注册和删除观察者对象的接口。

Observer(观察者，Observer)：为那些在目标发生改变时需要获得通知的对象定义一个更新接口。

ConcreteSubject(具体目标，Teacher)

将有关状态存入各ConcreteObserve对象。

当他的状态发生改变时，向他的各个观察者发出通知。

ConcreteObserver(具体观察者, Student)：

维护一个指向ConcreteSubject对象的引用。

存储有关状态，这些状态应与目标的状态保持一致。

实现Observer的更新接口以使自身状态与目标的状态保持一致。

协作关系：

当ConcreteSubject发生任何可能导致其观察者与其本身状态不一致的改变时，他将通知他的各个观察者。

在得到一个具体目标的改变通知后，ConcreteObserver对象可向目标对象查询信息。ConcreteObserver使用这些信息以使他的状态与目标对象的状态一致。

Subject代码：

**package** observer;

**public** **interface** Subject{

**public** **void** attach(Observer o);

**public** **void** detach(Observer o);

**public** **void** notice();

}

Observer代码：

**package** observer;

**public** **interface** Observer{

**public** **void** update();

}

Teacher代码；

**package** observer;

**import** java.util.Vector;

**public** **class** Teacher **implements** Subject{

**private** String phone;

**private** Vector students;

**public** Teacher(){

       phone = "";

       students = **new** Vector();

    }

**public**  **void** attach(Observer o){

       students.add(o);

    }

**public** **void** detach(Observer o){

       students.remove(o);

    }

**public** **void** notice(){

**for**(**int** i=0;i<students.size();i++)

           ((Observer)students.get(i)).update();

    }

**public** **void** setPhone(String phone){

**this**.phone = phone;

       notice();

    }

**public** String getPhone(){

**return** phone;

    }

}

Student代码：

**package** observer;

**public** **class** Student **implements** Observer{

**private** String name;

**private** String phone;

**private** Teacher teacher;

**public** Student(String name,Teacher t){

**this**.name = name;

       teacher = t;

    }

**public** **void** show(){

       System.*out*.println("Name:"+name+"\nTeacher'sphone:"+phone);

    }

**public** **void** update(){

       phone = teacher.getPhone();

    }

}

Client代码：

**package** observer;

**import** java.util.Vector;

**public** **class** Client{

**public** **static** **void** main(String[] args){

       Vector students = **new** Vector();

       Teacher t = **new** Teacher();

**for**(**int** i= 0 ;i<10;i++){

           Student st = **new** Student("lili"+i,t);

           students.add(st);

           t.attach(st);

       }

       t.setPhone("88803807");

**for**(**int** i=0;i<10;i++)

              ((Student)students.get(i)).show();

       t.setPhone("88808880");

**for**(**int** i=0;i<10;i++)

              ((Student)students.get(i)).show();

    }

}