

```

package com.observer.improve;

import java.util.ArrayList;
import java.util.Observable;

/**
 * 类是核心
 * 1. 包含最新的天气情况信息
 * 2. 含有 观察者集合，使用ArrayList管理
 * 3. 当数据有更新时，就主动的调用 ArrayList, 通知所有的（接入方）就看到最新的信息
 * @author Administrator
 *
 */
public class WeatherData implements Subject {
    private float temperature;
    private float pressure;
    private float humidity;
    //观察者集合
    private ArrayList<Observer> observers;

    //加入新的第三方

    public WeatherData() {
        observers = new ArrayList<Observer>();
    }

    public float getTemperature() {
        return temperature;
    }

    public float getPressure() {
        return pressure;
    }

    public float getHumidity() {
        return humidity;
    }

    public void dataChange() {
        //调用 接入方的 update

        notifyObservers();
    }

    //当数据有更新时，就调用 setData
    public void setData(float temperature, float pressure, float humidity) {

```

```

    this.temperatrue = temperature;
    this.pressure = pressure;
    this.humidity = humidity;
    //调用dataChange, 将最新的信息 推送给 接入方 currentConditions
    dataChange();
}

//注册一个观察者
@Override
public void registerObserver(Observer o) {
    // TODO Auto-generated method stub
    observers.add(o);
}

//移除一个观察者
@Override
public void removeObserver(Observer o) {
    // TODO Auto-generated method stub
    if(observers.contains(o)) {
        observers.remove(o);
    }
}

//遍历所有的观察者, 并通知
@Override
public void notifyObservers() {
    // TODO Auto-generated method stub
    for(int i = 0; i < observers.size(); i++) {
        observers.get(i).update(this.temperatrue, this.pressure, this.humidity);
    }
}
}

```