当一个事务发生了指定的动作的时候,要通知另外一个事务做出相应的处理

GUI编程的时候,使用的事件监听器就使用了这个模式

**观察者设计模式的实现步骤:把要通知对象的行为抽取出来,定义在一个接口上,然后在本类中维护该接口的成员**

:Weather.java

|  |
| --- |
| package com.company;  //如果要订阅天气的群体,也要实现Weather接口  public interface Weather{  public void notifyWeather(String weather);  } |

:WeatherStation

|  |
| --- |
| package com.company;  import java.util.ArrayList;  import java.util.Random;  /\*  需求:编写一个气象站,气象站要不断更新天气,人要根据当前的天气,做出相应的处理  \*/  public class WeatherStation {  String weathers[] = {"暴雨", "雾霾", "冰雹", "台风", "霜冻", "晴天"};  String weather;//当前的天气  Random random = new Random();  ArrayList<Weather> list = new ArrayList();  public void addListener(Weather e){  this.list.add(e);  }  //工作  public void startWork(){  new Thread(){  @Override  public void run(){  while(true){  //每隔1~1.5秒更新一次天气  WeatherStation.this.updateWeather();  for(Weather e:WeatherStation.this.list){  e.notifyWeather(WeatherStation.this.weather);  }  int n = WeatherStation.this.random.nextInt(501) + 1000;  try {  Thread.sleep(n);  } catch (InterruptedException e) {  e.printStackTrace();  }  }  }  }.start();  }  //更新天气的方法  public void updateWeather(){  int index = this.random.nextInt(this.weathers.length);  this.weather = this.weathers[index];  System.out.println("当前的天气:" + this.weather);  }  } |

:Emp.java

|  |
| --- |
| package com.company;  public class Emp implements Weather{  String name;  public Emp(String name){  this.name = name;  }  //人要根据天气做出相应的处理方案  public void notifyWeather( String weather){  // "暴雨", "雾霾", "冰雹", "台风", "霜冻", "晴天"  if("暴雨".equals(weather)){  System.out.println(name + "带着雨伞上班");  }else if("雾霾".equals(weather)){  System.out.println(name + "带着防毒面具上班");  }else if("冰雹".equals(weather)){  System.out.println(name + "带着头盔上班");  }else if("台风".equals(weather)){  System.out.println(name + "带着大石头上班");  }else if("霜冻".equals(weather)){  System.out.println(name + "带着棉被上班");  }else if("晴天".equals(weather)){  System.out.println(name + "开开心心上班");  }  }  public static void main(String args[]){  }  } |

:Student.java

|  |
| --- |
| package com.company;  public class Student implements Weather{  String name;  public Student(String name){  this.name = name;  }  public void notifyWeather ( String weather){  // "暴雨", "雾霾", "冰雹", "台风", "霜冻", "晴天"  if("暴雨".equals(weather)){  System.out.println(name + "带着宿舍休息");  }else if("雾霾".equals(weather)){  System.out.println(name + "心情不好,休息");  }else if("冰雹".equals(weather)){  System.out.println(name + "在宿舍休息!!");  }else if("台风".equals(weather)){  System.out.println(name + "在宿舍休息");  }else if("霜冻".equals(weather)){  System.out.println(name + "盖着棉被休息");  }else if("晴天".equals(weather)){  System.out.println(name + "去教室休息");  }  }  } |

:WeatherMain

|  |
| --- |
| package com.company;  public class WeatherMain{  public static void main(String args[]){  Emp e= new Emp("小明");  Emp e1= new Emp("小话");  Student s = new Student("同学A");  WeatherStation station = new WeatherStation();  station.addListener(e);  station.addListener(e1);  station.addListener(s);  station.startWork();  }  } |