简而言之，观察者模式=发布者+注册者。

观察者模式用来对GUI中的动作做侦听。Swing GUI的例子就表明了动作侦听是怎样实现观察者模式的。

下面是一个猎头的典型例子。这个图中有2个角色-猎头和求职者。求职者先在猎头处注册，当有新的工作机会时猎头就会通知求职者。

**观察者模式类图**

**Java代码**

**Subject接口**

|  |  |
| --- | --- |
| 1  2  3  4  5 | public interface Subject {      public void registerObserver(Observer o);      public void removeObserver(Observer o);      public void notifyAllObservers();  } |

**Observer接口**

|  |  |
| --- | --- |
| 1  2  3 | public interface Observer {      public void update(Subject s);  } |

**Hunter类实现了Subject接口**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | import java.util.ArrayList;    public class HeadHunter implements Subject{        //define a list of users, such as Mike, Bill, etc.      private ArrayList<Observer> userList;      private ArrayList<String> jobs;        public HeadHunter(){          userList = new ArrayList<Observer>();          jobs = new ArrayList<String>();      }        @Override      public void registerObserver(Observer o) {          userList.add(o);      }        @Override      public void removeObserver(Observer o) {}        @Override      public void notifyAllObservers() {          for(Observer o: userList){              o.update(this);          }      }        public void addJob(String job) {          this.jobs.add(job);          notifyAllObservers();      }        public ArrayList<String> getJobs() {          return jobs;      }        public String toString(){          return jobs.toString();      }  } |

JobSeeker是一个观察者：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | public class JobSeeker implements Observer {        private String name;        public JobSeeker(String name){          this.name = name;      }      @Override      public void update(Subject s) {          System.out.println(this.name + " got notified!");          //print job list          System.out.println(s);      }  } |

开始使用：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | public class Main {      public static void main(String[] args) {          HeadHunter hh = new HeadHunter();          hh.registerObserver(new JobSeeker("Mike"));          hh.registerObserver(new JobSeeker("Chris"));          hh.registerObserver(new JobSeeker("Jeff"));            //每次添加一个个job，所有找工作人都可以得到通知。          hh.addJob("Google Job");          hh.addJob("Yahoo Job");      }  } |