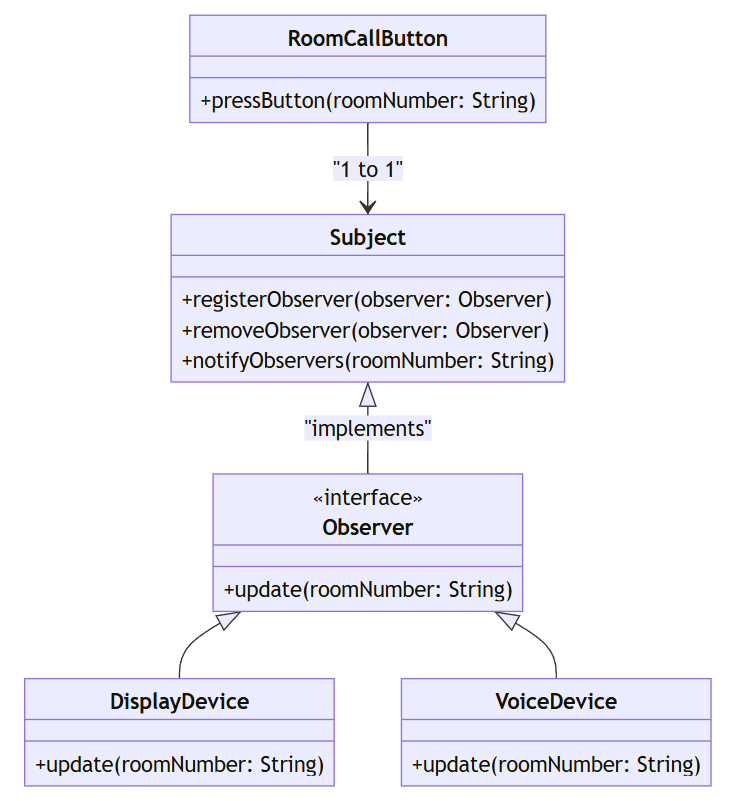
**题目：**

小明正在开发一个病房呼叫应答仿真系统。系统中每个病房配备一个呼叫按钮和一个呼叫显示应答器，疗区大厅配备一个显示应答器和一个语音播报器。假设，按下001号病房的呼叫按钮后，那么，所有的呼叫显示应答器都会显示发出呼叫的病房号001，大厅同时语音播报该病房号001。当医护人员按下任意一处的呼叫显示应答器的应答按钮后，所有呼叫显示应答器停止显示该房间号，大厅也停止语音播报。请用恰当的设计模式实现该系统，画出类图，给出核心代码。

**模式：**

观察者模式

**类图：**



|  |
| --- |
| **代码** |
| import java.util.ArrayList;  import java.util.List;  *// 抽象主题类，管理观察者*  abstract class Subject {      private List<Observer> observers = new ArrayList<>();      public void registerObserver(Observer observer) {          observers.add(observer);      }      public void removeObserver(Observer observer) {          observers.remove(observer);      }      public void notifyObservers(String roomNumber) {          for (Observer observer : observers) {              observer.update(roomNumber);          }      }  }  *// 观察者接口*  interface Observer {      void update(String roomNumber);  }  *// 病房呼叫按钮类*  class RoomCallButton extends Subject {      public void pressButton(String roomNumber) {          System.out.println("Room " + roomNumber + " called for assistance.");          notifyObservers(roomNumber);      }  }  *// 显示设备类*  class DisplayDevice implements Observer {      private String name;      public DisplayDevice(String name) {          this.name = name;      }      @Override      public void update(String roomNumber) {          if (roomNumber != null) {              System.out.println(name + " displaying call from room: " + roomNumber);          } else {              System.out.println(name + " cleared display.");          }      }  }  *// 语音播报设备类*  class VoiceDevice implements Observer {      @Override      public void update(String roomNumber) {          if (roomNumber != null) {              System.out.println("Broadcasting voice message: 'Room " + roomNumber + " is calling for assistance.'");          } else {              System.out.println("Voice broadcast stopped.");          }      }  }  *// 测试类*  public class WardCallSimulation {      public static void main(String[] args) {  *// 创建病房呼叫按钮*          RoomCallButton roomCallButton = new RoomCallButton();  *// 创建观察者设备*          DisplayDevice display1 = new DisplayDevice("Display Device 1");          DisplayDevice display2 = new DisplayDevice("Display Device 2");          VoiceDevice voiceDevice = new VoiceDevice();  *// 注册观察者*          roomCallButton.registerObserver(display1);          roomCallButton.registerObserver(display2);          roomCallButton.registerObserver(voiceDevice);  *// 模拟病房呼叫*          roomCallButton.pressButton("001");  *// 模拟医护人员应答*          System.out.println("Nurse responds to the call...");          roomCallButton.notifyObservers(null);  *// 传递 null 清除显示和语音播报*      }  } |