

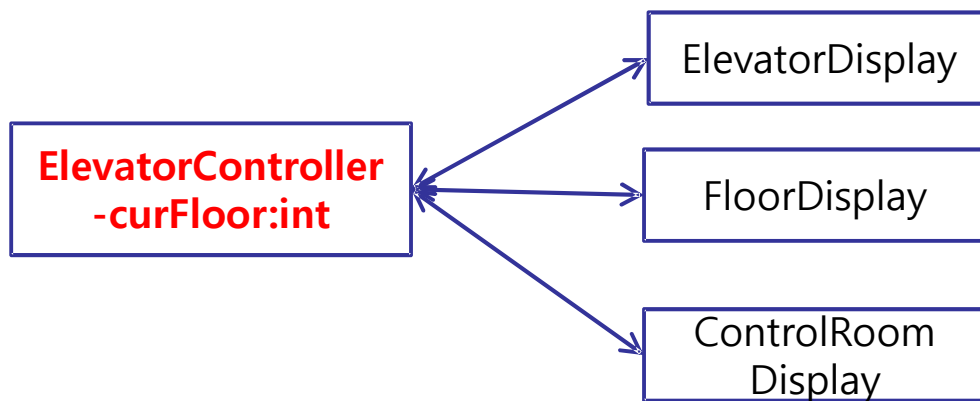


Observer pattern

**PRACTICE – ELEVATOR
LOCATION**

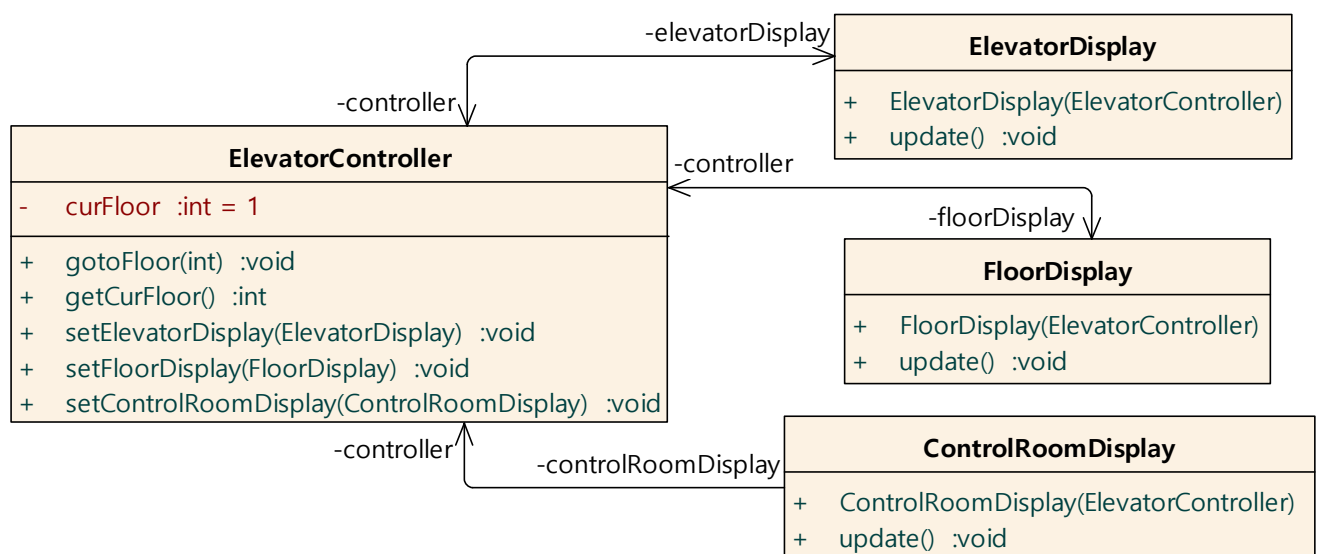
Elevator Location - Design

- ◆ When a Elevator Controller moves an elevator, the elevator's location changes.
- ◆ The location needs to be shown on Elevator Display, Floor Display, and ControlRoom Display



3

Elevator Controller - Design



4

Source Code – ElevatorDisplay

```
public class ElevatorDisplay {  
    private ElevatorController controller ;  
    public ElevatorDisplay(ElevatorController controller) {  
        this.controller = controller ;  
    }  
    public void update() {  
        int curFloor = controller.getCurFloor() ;  
        System.out.println("Elevator Display: " + curFloor) ;  
    }  
}
```

5

Source Code – FloorDisplay

```
public class FloorDisplay {  
    private ElevatorController controller ;  
    public FloorDisplay(ElevatorController controller) {  
        this.controller = controller ;  
    }  
    public void update() {  
        int curFloor = controller.getCurFloor() ;  
        System.out.println("Floor Display: " + curFloor) ;  
    }  
}
```

6

Source Code – ControlRoomDisplay

```
public class ControlRoomDisplay {
    private ElevatorController controller ;
    public ControlRoomDisplay(ElevatorController controller) {
        this.controller = controller ;
    }
    public void update() {
        int curFloor = controller.getCurFloor() ;
        System.out.println("Control Room: " + curFloor) ;
    }
}
```

7

Source Code - Elevator Controller

```
public class ElevatorController {
    private int curFloor = 1;
    private ElevatorDisplay elevatorDisplay ;
    private FloorDisplay floorDisplay ;
    private ControlRoomDisplay controlRoomDisplay ;
    public void gotoFloor(int destination) {
        curFloor = destination ;
        elevatorDisplay.update();
        floorDisplay.update() ;
        controlRoomDisplay.update() ;
    }
    public int getCurFloor() { return curFloor ; }
    public void setElevatorDisplay(ElevatorDisplay elevatorDisplay) {
        this.elevatorDisplay = elevatorDisplay;
    }
    public void setFloorDisplay(FloorDisplay floorDisplay) {
        this.floorDisplay = floorDisplay;
    }
    public void setControlRoomDisplay(ControlRoomDisplay display) {
        this.controlRoomDisplay = display;
    }
}
```

8

Client – Source Code

```
public class Client {  
    public static void main(String[] args) {  
        ElevatorController controller = new ElevatorController() ;  
  
        ElevatorDisplay elevatorDisplay = new ElevatorDisplay(controller) ;  
        FloorDisplay floorDisplay = new FloorDisplay(controller) ;  
        ControlRoomDisplay controlRoomDisplay =  
            new ControlRoomDisplay(controller) ;  
  
        controller.setElevatorDisplay(elevatorDisplay) ;  
        controller.setFloorDisplay(floorDisplay) ;  
        controller.setControlRoomDisplay(controlRoomDisplay) ;  
  
        controller.gotoFloor(5) ;  
        controller.gotoFloor(10) ;  
    }  
}
```

```
Elevator Display: 5  
Floor Display: 5  
Control Room: 5  
Elevator Display: 10  
Floor Display: 10  
Control Room: 10
```

9

Violation of OOD Principle

Principle	Codes(class/method) that violate the principle
SRP	
OCP	
LSP	
ISP	
DIP	

Source Code - Elevator Controller

```
public class ElevatorController {  
    private int curFloor = 1;  
    private ElevatorDisplay elevatorDisplay ;  
    private FloorDisplay floorDisplay ;  
    private ControlRoomDisplay controlRoomDisplay ;  
    public void gotoFloor(int destination) {  
        curFloor = destination ;  
        elevatorDisplay.update();  
        floorDisplay.update() ;  
        controlRoomDisplay.update() ;  
    }  
    public int getCurFloor() { return curFloor ; }  
    public void setElevatorDisplay(ElevatorDisplay elevatorDisplay) {  
        this.elevatorDisplay = elevatorDisplay;  
    }  
    public void setFloorDisplay(FloorDisplay floorDisplay) {  
        this.floorDisplay = floorDisplay;  
    }  
    public void setControlRoomDisplay(ControlRoomDisplay display) {  
        this.controlRoomDisplay = display;  
    }  
}
```

11

Solution – Observer Design

12

Source Code – Observer and Subject

13

Source Code – Concrete Observers

14

Source Code – Concrete Subject

15

Source Code - Client

16