

```
// DisplayController.java - DISPLAY CONTROLLER
//
// MODULE INDEX
// NAME                CONTENTS
// DisplayController   Class declaration
// DisplayInit         Initialize the controller
// LiftCalled          Process when button is pressed
// LiftStopped         Process when lift stops at floor
//
// MAINTENANCE HISTORY
// DATE      PROGRAMMER AND DEGUILS
// 29-09-16 MPF Original
//-----

public class DisplayController
{
    //-----

    // DEFINITIONS

    private static final int MAX_FLOORS = 11;
    private static final int MAX_LIFTS = 3;
    private static final int LED_BLUE = 2;
    private static final int LED_RED = 1;
    private static final int LED_DARK = 0;
    private static final int LED_UNKNOWN = -1;

    private boolean[][] passengers = new boolean[MAX_FLOORS][MAX_FLOORS];
    private int[][] LED = new int[MAX_LIFTS][MAX_FLOORS];
    private LEDUnit light;

    // Called to initialize the controller.

    public void DisplayInit(LEDUnit light)
    {
        this.light = light;
        for (int lift = 0; lift < MAX_LIFTS; lift++)
            for (int floor = 0; floor < MAX_FLOORS; floor++)
                light.LEDSwitch(lift, floor, LED_DARK);
    }

    // Called when a call button is pressed.

    public void LiftCalled(int fromFloor, int toFloor)
    {
        // Keep track of destination floors

        passengers[fromFloor][toFloor] = true;

        // Light half when not fully lit

        for (int lift = 0; lift < MAX_LIFTS; lift++) {
            if (LED[lift][fromFloor] != LED_BLUE) {
                LED[lift][fromFloor] = LED_RED;
                light.LEDSwitch(lift, fromFloor, LED_RED);
            }
        }
    }
}
```

```
}

// Called when the lift stops at floor.

public void LiftStopped(int liftNo, int stopFloor, int direction)
{
    int newColour = LED_DARK;

    // Moving down

    if (direction < 0) {

        // If any passengers moving up, use RED. Otherwise go DARK

        for (int toFloor = stopFloor + 1; toFloor < MAX_FLOORS; toFloor++) {
            if (passengers[stopFloor][toFloor]) {
                newColour = LED_RED;
                break;
            }
        }

        LED[liftNo][stopFloor] = newColour;
        light.LEDSwitch(liftNo, stopFloor, newColour);

        // If no more passenger, go DARK for all lifts

        if (newColour == LED_DARK) {
            for (int lift = 0; lift < MAX_LIFTS; lift++) {
                if (LED[lift][stopFloor] == LED_RED){
                    LED[lift][stopFloor] = LED_DARK;
                    light.LEDSwitch(lift, stopFloor, LED_DARK);
                }
            }
        }

        // Light destination floors BLUE

        for (int toFloor = 0; toFloor < stopFloor; toFloor++) {
            if (passengers[stopFloor][toFloor]) {
                LED[liftNo][toFloor] = LED_BLUE;
                light.LEDSwitch(liftNo, toFloor, LED_BLUE);
                passengers[stopFloor][toFloor] = false;
            }
        }
    }

    // Moving up

    else if (direction > 0) {

        // If any passengers going down, use RED. Otherwise go DARK

        for (int toFloor = 0; toFloor < stopFloor; toFloor++) {
            if (passengers[stopFloor][toFloor]) {
                newColour = LED_RED;
                break;
            }
        }
    }
}
```

```
    }

    LED[liftNo][stopFloor] = newColour;
    light.LEDSwitch(liftNo, stopFloor, newColour);

    // If no more passenger, go DARK for all lifts

    if (newColour == LED_DARK) {
        for (int lift = 0; lift < MAX_LIFTS; lift++) {
            if (LED[lift][stopFloor] == LED_RED) {
                LED[lift][stopFloor] = LED_DARK;
                light.LEDSwitch(lift, stopFloor, LED_DARK);
            }
        }
    }

    // Light destination floors BLUE

    for (int toFloor = stopFloor + 1; toFloor < MAX_FLOORS; toFloor++) {
        if (passengers[stopFloor][toFloor]) {
            LED[liftNo][toFloor] = LED_BLUE;
            light.LEDSwitch(liftNo, toFloor, LED_BLUE);
            passengers[stopFloor][toFloor] = false;
        }
    }
}

// Not moving anymore

else {

    // If any passengers going down/up, use RED. Otherwise go DARK

    for (int toFloor = 0; toFloor < MAX_FLOORS; toFloor++) {
        if (passengers[stopFloor][toFloor]) {
            newColour = LED_RED;
            break;
        }
    }

    LED[liftNo][stopFloor] = newColour;
    light.LEDSwitch(liftNo, stopFloor, newColour);

    // If no more passenger, go DARK for all lifts

    if (newColour == LED_DARK) {
        for (int lift = 0; lift < MAX_LIFTS; lift++) {
            if (LED[lift][stopFloor] == LED_RED) {
                LED[lift][stopFloor] = LED_DARK;
                light.LEDSwitch(lift, stopFloor, LED_DARK);
            }
        }
    }
}
}
}
}
}
```