



## ASSIGNMENT 11 — SOLUTION

### GENERAL INFORMATION

This assignment is made public on Friday, April 4th, 2002. The assignment is due on  
Friday, May 2nd, 1 PM.

Hand in your assignment to the teaching assistant running your lab session.  
The first page of your (written) assignment has to contain at least the following information:

- the course name (Introduction to Programming - Concepts and Tools)
- your name and your student number
- name and student number of the fellow student if you submit in pairs
- assignment number

*Please staple your assignment!*

You will get back the graded assignment one week after submission deadline.

### QUESTIONS

1. (*Traffic light*)

### SOLUTION

Below you will find the code for classes

- BulbState.java
- Bulb.java
- States.java
- TrafficLight.java
- TrafficLightGUI.java

which make up my solution. Note the different ways the states are implemented for Bulb and for TrafficLight. Also note that we create an array of labels in class TrafficLightGUI, which are easy to set up. Note also that there are still comments missing, information about the author, etc., and other things to polish the code.

```
class BulbState{

    public static boolean ON = true;
    public static boolean OFF = false;

    private boolean state;

    BulbState(){state = BulbState.OFF;}
    BulbState(boolean state){this.state = state;}

    public void switchState(){state = ! state;}

    public boolean getState(){return state;}
    public void setState(boolean state){this.state = state;}

}
```

```
import java.awt.Color;

class Bulb{

    public static final Color OFF_COLOR = Color.black;
    private BulbState state;
    private Color color;

    public Bulb(Color c){
        color = c;
        state = new BulbState(BulbState.OFF);
    }

    public void switchState(){state.switchState();}
    public boolean getState(){return state.getState();}
    public void setState(boolean state){this.state.setState(state);}

    public Color getColor(){
        if(state.getState()==BulbState.ON)
            return color;
        else
            return Bulb.OFF_COLOR;
    }
}
```

```

class States {

    private int state;
    private int howMany;

    States(int h){
        state = 1;
        try{
            if(h<=0)
                throw new Exception("\nThere must be more than 0 states!");
            howMany = h;
        }
        catch(Exception e)
            {System.exit(0);}
    }

    public int next(){
        this.state = (state-1+1)%howMany+1;
        return state;
    }

    public int getState(){return state;}
    public int getHowMany(){return howMany;}

    public boolean setState(int state){
        boolean correct = false;
        try{
            if(1<=state && state <= howMany){
                this.state = state;
                correct = true;
            }else if(state > howMany)
                this.state = (state-1)%howMany+1;
            else
                throw new Exception("\nA state has to be positive!");
        }
        catch(Exception e)
            {System.exit(0);}
        return correct;
    }
}

```

```

import java.awt.Color;

class TrafficLight{

    private Bulb red = new Bulb(Color.red);
    private Bulb orange = new Bulb(Color.orange);
    private Bulb green = new Bulb(Color.green);
    States state = new States(4);

    TrafficLight(){red.setState(BulbState.ON);}

    public void next(){
        switch(state.getState()){
            case 1:
                orange.switchState();
                break;
            case 2:
                red.switchState();
                orange.switchState();
                green.switchState();
                break;
            case 3:
                orange.switchState();
                green.switchState();
                break;
            case 4:
                red.switchState();
                orange.switchState();
                break;
            default:
                }
            state.next();
        }

    public Color[] getColors(){
        Color[] colors = new Color[3];

        colors[0] = red.getColor();
        colors[1] = orange.getColor();
        colors[2] = green.getColor();

        return colors;
    }

    public int getState(){return state.getState();}

}

```

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class TrafficLightGUI{

    static window w;

    public static void main(String[] args){

        w = new window();
        w.addWindowListener(new myWindowListener());
        w.show();

    }
}

class myWindowListener extends WindowAdapter{

    public void windowClosing(WindowEvent e){System.exit(0);}

}

class window extends Frame implements ActionListener{

    Button b;
    Label[] lights = new Label[3];
    Label currentState = new Label();
    TrafficLight tl = new TrafficLight();

    public window(){
        setLayout(new GridLayout(5,1));

        for(int i = 0;i<3;i++)
        {
            lights[i] = new Label("");
            lights[i].setBackground(tl.getColors()[i]);
            add(lights[i]);
        }

        currentState = new Label("Current state: " + tl.getState(),Label.LEFT);
        currentState.setBackground(Color.white);
        add(currentState);

        b = new Button("Next");
        b.setBackground(Color.white);
        b.addActionListener(this);
        add(b);

        setSize(150,200);
    }

    public void actionPerformed(ActionEvent e){
        if(e.getSource()==b){
            tl.next();
            for(int i = 0;i<3;i++)
                lights[i].setBackground(tl.getColors()[i]);
            currentState.setText("Current state: " + tl.getState());
        }
    }
}

```

```
    return;  
  }  
}
```

I leave it to you to draw the UML class diagram.

## 2. (*Vote Count*)

### SOLUTION

My solution is the following, again non-polished code:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class Election{

    static window w;

    public static void main(String[] args){

        w = new window();
        w.addWindowListener(new myWindowListener());
        w.show();

    }
}

class myWindowListener extends WindowAdapter{

    public void windowClosing(WindowEvent e){System.exit(0);}

}

class window extends Frame implements ActionListener{

    Label bush;
    Label gore;
    Button bush_button;
    Button gore_button;
    Panel p;
    Label message;
    Button reset;

    int bush_counter = 0;
    int gore_counter = 0;

    public window(){
        setLayout(new GridLayout(3,1));
        p = new Panel();
        p.setLayout(new GridLayout(2,2));

        bush_button = new Button();
        bush_button.setLabel("Bush");
        bush_button.setBackground(Color.white);
        bush_button.addActionListener(this);
        p.add(bush_button);

        bush = new Label("0",Label.RIGHT);
        bush.setBackground(Color.white);
        p.add(bush);

        gore_button = new Button();
```

```

gore_button.setLabel("Gore");
gore_button.setBackground(Color.white);
gore_button.addActionListener(this);
p.add(gore_button);

gore = new Label("0",Label.RIGHT);
gore.setBackground(Color.white);
p.add(gore);

add(p);

message = new Label("Even.");
message.setBackground(Color.white);
add(message);

reset = new Button();
reset.setLabel("Reset");
reset.setBackground(Color.white);
reset.addActionListener(this);
add(reset);

setSize(200,200);
}

public void actionPerformed(ActionEvent e){
    if(e.getSource()==reset){
        bush_counter=0;
        bush.setText(""+bush_counter);
        gore_counter=0;
        gore.setText(""+bush_counter);
        message.setText("Even.");
    }
    else if(e.getSource()==bush_button){
        bush_counter++;
        bush.setText(""+bush_counter);
    }
    else{
        gore_counter++;
        gore.setText(""+gore_counter);
    }

    if(bush_counter==gore_counter){
        message.setText("Even.");
    }else if(bush_counter > gore_counter){
        message.setText("Bush is ahead by " + (bush_counter-gore_counter) + " votes.");
    }else{
        message.setText("Gore is ahead by " + (gore_counter-bush_counter) + " votes.");
    }
    return;
}
}

```

### 3. (*JList*)

#### SOLUTION

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import java.awt.Color;

class ColorChoice{
    static ColorWindow w;

    public static void main(String[] args){
        w = new ColorWindow();
        w.addWindowListener(new myColorWindowListener());
        w.setSize(200,200);
        w.show();
    }
}

class myColorWindowListener extends WindowAdapter{
    public void windowClosing(WindowEvent e){System.exit(0);}
}

class ColorWindow extends JFrame implements ListSelectionListener{
    Container c;
    JList colorList;

    private static String colorNames[] = {
        "Black", "Blue", "Cyan", "Dark Gray", "Gray", "Green", "Light Gray",
        "Magenta", "Orange", "Pink", "Red", "White", "Yellow"};

    private static Color colors[] = {
        Color.black, Color.blue, Color.cyan, Color.darkGray, Color.gray,
        Color.green, Color.lightGray, Color.magenta, Color.orange,
        Color.pink,
        Color.red, Color.white, Color.yellow
    };

    ColorWindow(){
        super();
        c = getContentPane();
        c.setLayout(new FlowLayout());

        colorList = new JList(colorNames);

        colorList.setVisibleRowCount(5);
        colorList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        c.add(new JScrollPane(colorList));
        colorList.addListSelectionListener(this);
    }

    public void valueChanged(ListSelectionEvent e){
        c.setBackground(colors[colorList.getSelectedIndex()]);
    }
}
```