



ASSIGNMENT 11

GENERAL INFORMATION

This assignment is made public on Friday, April 24th, 2003. The assignment is due on

Friday, May 1st, 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*)

LEARNING OBJECTIVES: Creation of a simple GUI; modification of given code; object oriented design.

Implement a graphical user interface `TrafficLight` that models a traffic light. The user interface should consist of three vertical labels representing the three light bulbs, and a button that can be used to change the state of the traffic light. Initially the traffic light should show red, i.e., only the top label should be red, the others being black. When pressing the button the usual phases of the traffic light should be shown, i.e., red-orange, green, orange, red.

Note that the assignment is a minor modification of the user interface `StopGo.java` that was discussed in class.

Submit your code, the UML class diagrams, and write at least half a page explaining the design of your traffic light.

Hint: Use good object oriented design and re-use of code. You could, for example, create a class `Bulb` representing one of the light-bulbs, having a particular state (on or off) and a particular color attached. You might use a method `changeState` to change the state of the bulb, and have methods to return the state or the current color. Another class might join together the three light bulbs into one traffic light. Again, a method `changeState` might switch between the four different states of the traffic light. Finally, everything has to be put into a frame, a button attached, and the `ActionListener` has to be implemented.

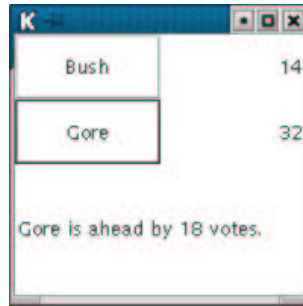
WHAT TO SUBMIT: Submit the copy of your code, a UML diagram showing the class relation ship, plus at least have a page describing your design and how your program works.

2. (*Vote Count*)

LEARNING OBJECTIVES: Design of a GUI with a small number of components.

(Compare the exam Fall 2000.)

Write a graphical user interface that allows to register votes in an election. The interface should look as follows:



It consists of two panels where the bottom one contains just a label, whereas the top one (using grid-layout) contains two buttons and two labels. When clicking a button an internal counter is incremented, the new value is displayed on the corresponding label, and depending on the difference the bottom label shows the text *Even.*, or that one of the two is ahead by so-and-so many votes.

Add a further `reset` button, setting both vote counts back to 0. The new interface should look as follows:



For a working version download the files `Election.class`, `window.class`, and `myWindowListener.class` from the course homepage.

WHAT TO SUBMIT: Submit the code of your program, plus a screen shot showing your program running.

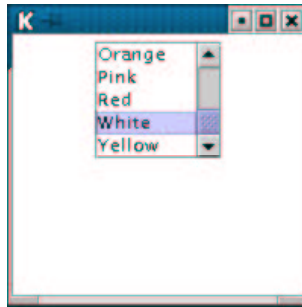
3. (*JList*)

LEARNING OBJECTIVES: Using a new component in a GUI; reading Java documentation and finding out functionality of Java classes.

(This questions is optional, but you should make a serious attempt to solve it and get used to finding information in the Java documentation.)

Read about the `java.swing` component `JList`. It is a menu with a ruler on the right, allowing to select a single item from a list.

Write a program that allows to pick from a list of colors and that will use the selected color as background color of a user interface. The user interface itself might look as follows:



For a working version download the files `ColorChoice.class` and `ColorWindow.class` and `myColorWindowListener.class` from the course homepage.

Hints: The easiest is to maintain two arrays. One array of strings should contain the color names, the other array should be an array of objects of class `Color`.

Of class `JList` I used the following methods:

- `JList(String[])`
- `setVisibleRowCount(int)`
- `setSelectionMode(int)`
- `getSelectedIndex()`

Note also that you have to implement the interface `ListSelectionListener`, for which you have to import `javax.swing.event.*`.

WHAT TO SUBMIT: Submit the code of your program, plus screen shots showing your program running.