



ASSIGNMENT (AFLEVERINGSOPGAVE) 8

GENERAL INFORMATION

This assignment is made public on Friday, March 26th, 1 PM. The assignment is due on
Friday, April 2nd, 1PM.

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 (Grundlæggende Programmering)
- name and student number of the fellow student(s) in your group (max two)
- assignment number

Please staple your assignment!

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

WARM-UP AND SUGGESTIONS FOR FURTHER EXERCISES

As warm-up I suggest that you have a look at the review questions of Chapter 8 of the book (p. 304 of my copy).

As to further exercises I suggest the following from Chapter 8 of the textbook (pp. 304–306 of my copy):

- Exercises 1–3 are elementary questions about displaying a window.
- Exercises 4 and 5 ask to design a simple window for displaying the results of a calculation.
- Exercises 6 and 7 are about parsing a string and displaying the individual components (tokens) in individual boxes.
- Exercises 8–11 are about a simple calculator that was introduced in Section 8.4 of the book.
- Exercises 12–16 are about different layout managers. Exercises 15 and 16 ask to implement an interface for playing Tic-Tac-Toe on a three by three grid.

It is up to you whether or not you want to work on these exercises. Those exercises will not be marked.

ØVELSER

Implement a graphical user interface `TrafficLight` that models a traffic light used to control the flow of cars. 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.

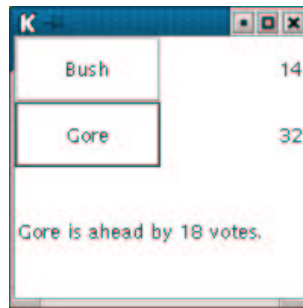
Hint: Use good object oriented design and re-use of code. You could, for example, use the class `Bulb` (Assignment 6) representing one of the light-bulbs, having a particular state (on or off) and a particular color attached. You might use a method `change()` to change the state of the bulb, and have 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 `change()` 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.

ASSIGNMENT: GUIs

1. *Vote Count*

Opgaven er fra examen E2000.

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:



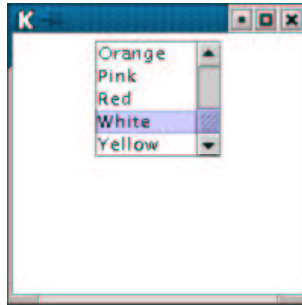
For a working version download the files `Election.class`, `ElectionWindow.class`, and `ElectionWindowListener.class` from the course homepage.

HVAD SKAL AFLEVERES: Aflever koden, plus billeder (screen shots) som viser jeres GUI. Supplerer jeres program med en beskrivelse hvordan det er bygget op, og med et UML klasse diagram.

2. *JList*

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 `ColorWindowListener.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.*`.

HVAD SKAL AFLEVERES: Aflever koden, plus billeder (screen shots) som viser jeres GUI. Supplerer jeres program med en beskrivelse hvordan det er bygget op, og med et UML klasse diagram.