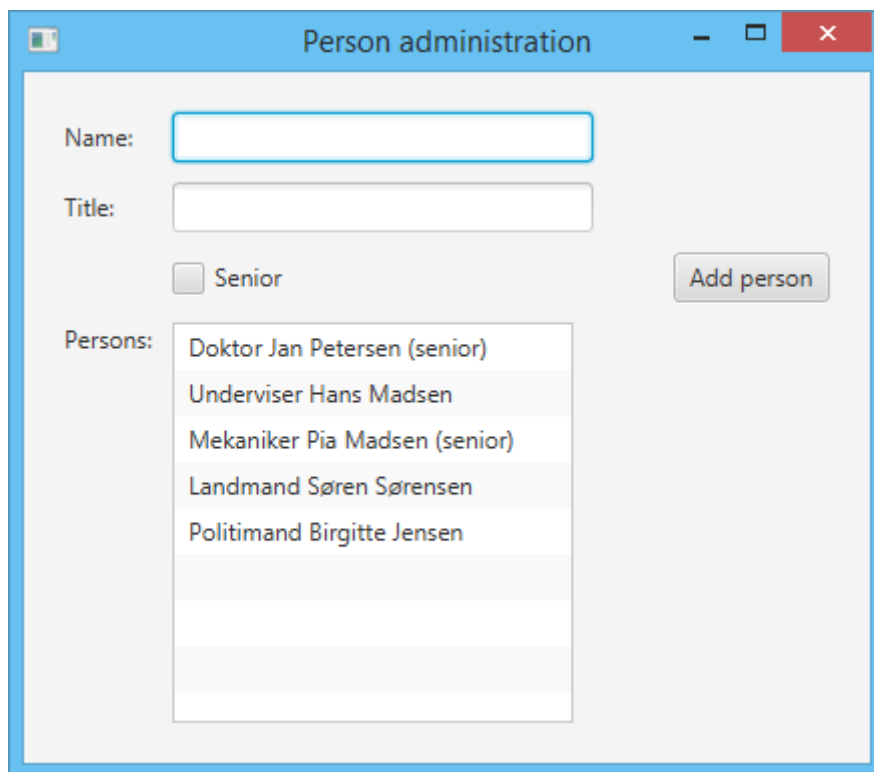


Tirsdag d. 11. april 2023

Exercise 1

Make a program with the GUI shown here:



The *Add Person* button adds a description of the typed in person to the *Persons* list view. If the *Senior* check box is checked, the description in the list view must show it. (List view and check box is mentioned in the paper on GUI and in the GUI examples.)

You can use the following Person class (or make your own Person class).

```
public class Person {  
    private String name;  
    private String title;  
    private boolean senior;  
  
    public Person(String name, String title, boolean senior) {  
        this.name = name;  
        this.title = title;  
        this.senior = senior;  
    }  
}
```

```
public String toString() {  
    String s = title + " " + name;  
    if (senior) {  
        s += " (senior)";  
    }  
    return s;  
}
```

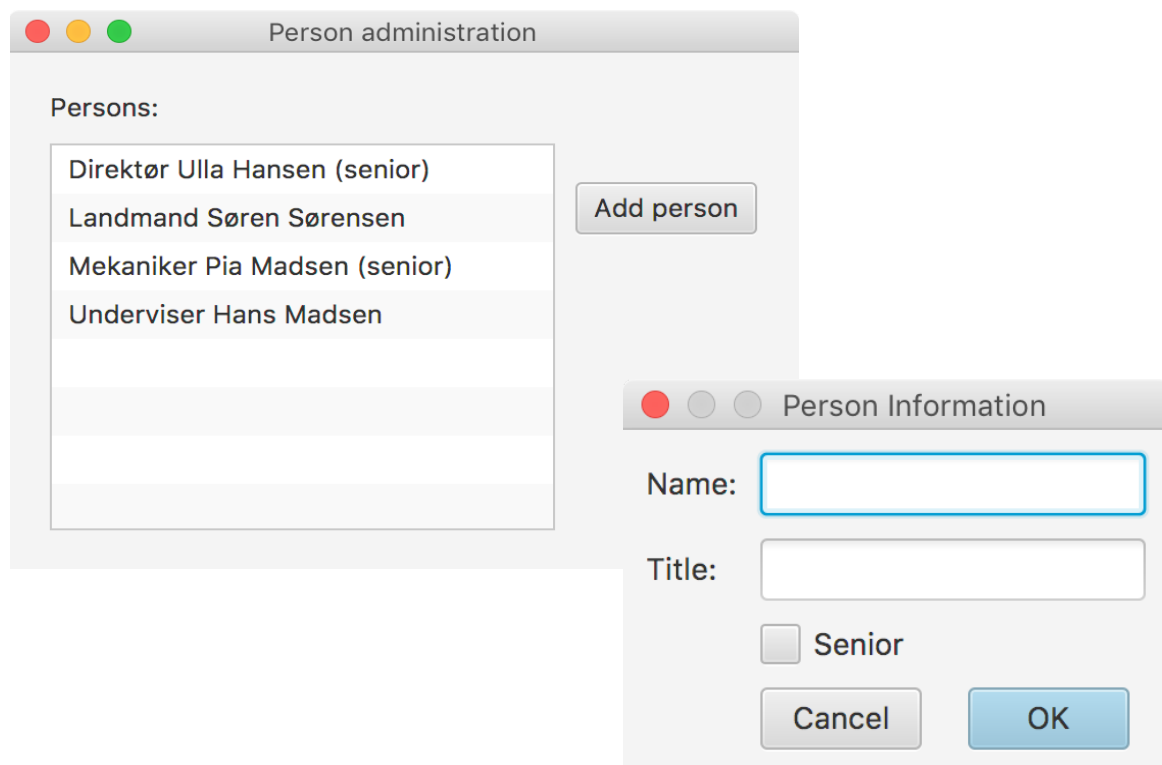
Exercise 2

Extend exercise 1 to show a dialog window with a suitable message, if the *Add Person* button is pressed, when the *Name* text field or the *Title* text field is empty.

Exercise 3

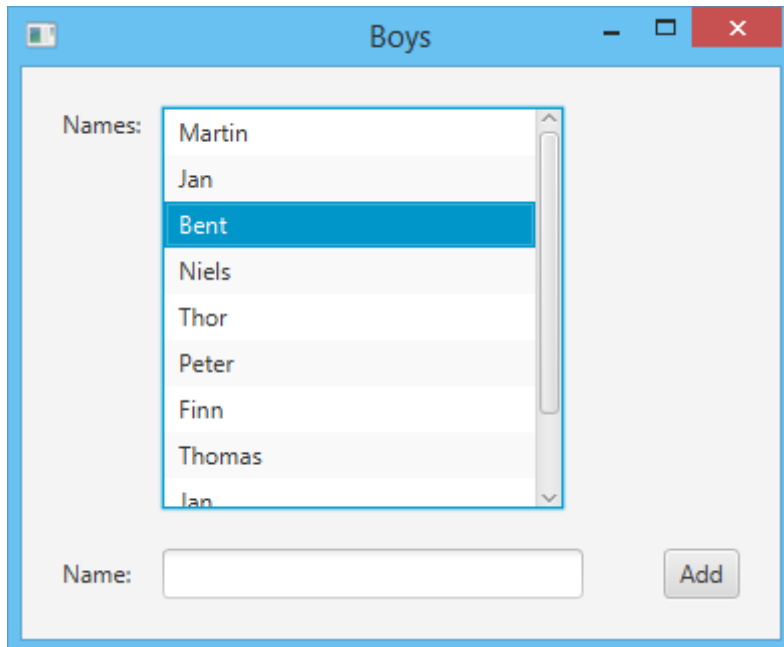
The program in this exercise does the same as the program in exercise 1. But this time the information about a person is gathered in a new window (see the windows below here). When the *Add Person* button is clicked, the *Person Information* window is opened as a modal dialog.

The GUI examples contains a dialog window used for input. Use this example as basis for your input window.



Exercise 4

Make a program with the GUI shown here:



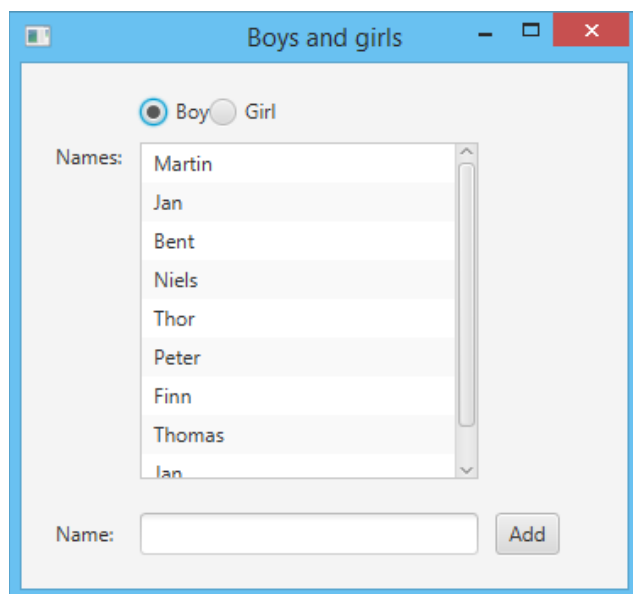
The window has a list view showing names of boys (remember the names in a field of type `ArrayList<String>`). When the *Add* button is pressed, the boy name in the text field must be added to the list view.

Exercise 5

Make a program with the GUI shown here:

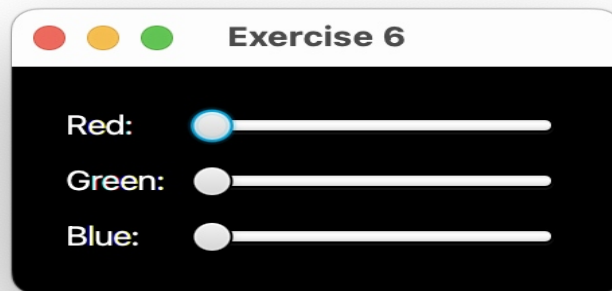
The list view shows either names of boys or names of girls, depending on the selected radio button at the window's top. The names are remembered in 2 fields of type `ArrayList<String>`. When the *Add* button is clicked, the name in the text field have to be added to the list shown in the list view.

Hint: Use only one `ListView`, and replace the content of the list view, when a new radio button is selected.



Exercise 6 – extra

Make a program with sliders as shown in the window here:



When the sliders are moved to the right, the color of the background is changed towards white, because more color is added (the color is determined by the amount of red, green and blue).

A slider, that can have values between 0 and 255, is coded like this:

```
Slider slider = new Slider(0, 255, 0);  
slider.valueProperty().addListener(  
    event -> this.sliderValueChanged(pane));
```

The last line adds a listener that calls the method
private void sliderValueChanged(GridPane pane)
every time the slider is moved.

Setting the background color of the pane with only blue is done with the following code:

```
String color = "#0000"+String.format("%02X", blueValue);  
pane.setStyle("-fx-background: " + color);
```

The color is determined by a hexadecimal string with two digits for red, 2 digits for green and 2 digits for blue. The code above uses the value of the blue slider to determine the last two digits of the color.

(Hexadecimal numbers uses digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. The number FF in hexadecimal is $15 \cdot 16 + 15 = 255$ in decimal.)

Opgave 7 (ekstra)

I denne opgave kan du eksperimentere med tabs. I JavaFX findes der noget der hedder en TabPane, som kan vise indhold i flere forskellige tabs. Lav 2 list views (f.eks. med Boys og Girls, se opgave 5). og vis dem i hvert sin tab i vinduet. Du kan læse JavaDoc til TabView i JavaFX 8 her: <http://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/TabPane.html>

Læg især mærke til eksemplet i ovenstående link, hvor der vises hvordan man sætter indhold på en tab, og tilføjer tab'en til et TabPane. Dit TabPane kan du efterfølgende sætte i en GridPane i din Scene. I eksemplet sætter de en Rectangle i den Tab de laver, men man kan sætte stort set hvad som helst i sin Tab, også et ListView.

Resultatet kunne se sådan ud:

