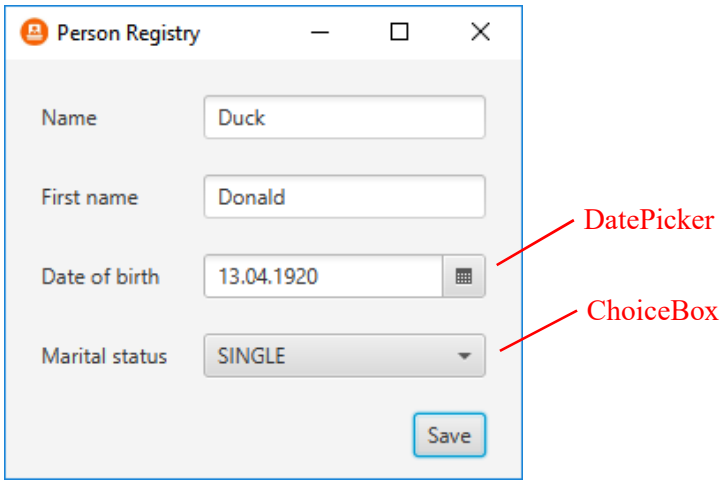
Object Oriented Programming 2

Rolf Haenni & Andres Scheidegger

Exercises 1

1. JavaFX GUI for Person Data

Write a JavaFX Application with the following GUI:



Use an appropriate pane as root node in the scene. Write a separate builder class for the construction of the scene object. Tweak the layout by experimenting with setting of gaps, padding, maximum, width minimum width, etc. on diverse nodes.

Choices for marital status are SINGLE, MARRIED, CIVIL_PARTNERSHIP, DIVORCED, WIDOWED. Use an enum (see OOP1).

Set an appropriate icon and title in the stage.

Clicking the Save button has no effect (see next task).

2. Implement Action Handler for Save button

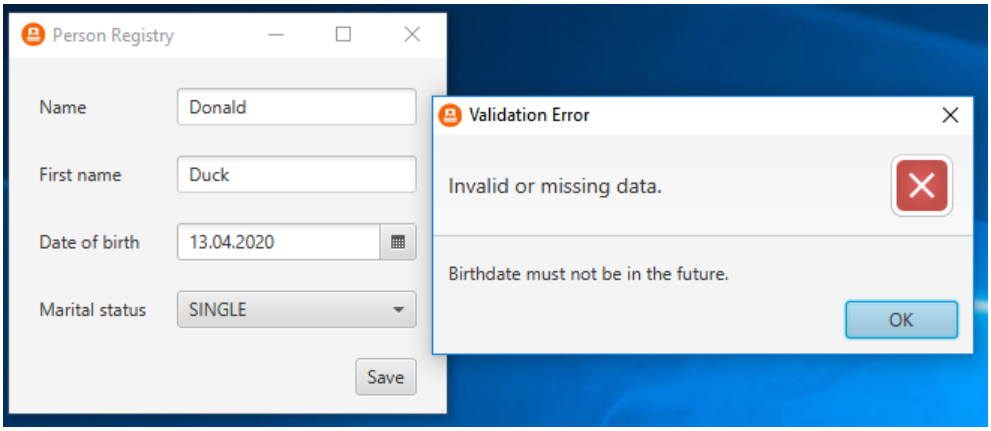
Implement an action handler class for the Save button. When clicked, the person data shall be collected from the controls and saved in a newly created Person object (see OOP1). The Person object shall then be printed to the console:

```
Duck Donald, 1920-04-13, SINGLE
```

Use ids (setId) to identify the nodes of the scene graph, which hold the person data.

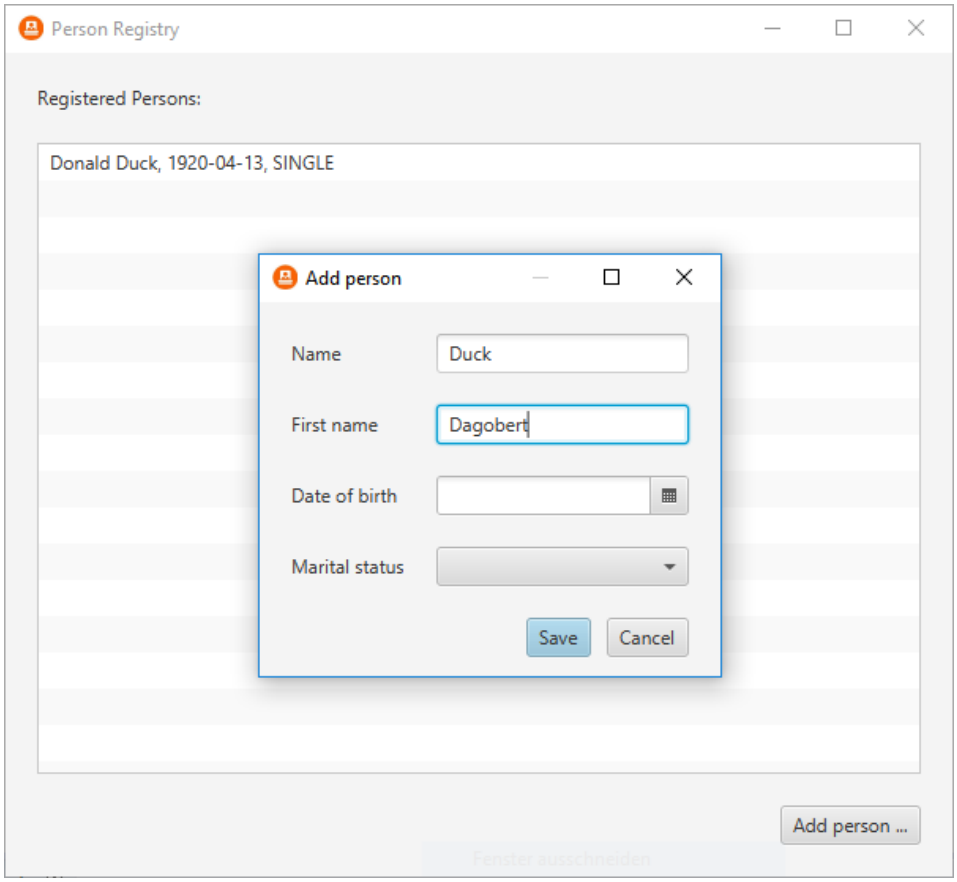
3. Validation of Input

When the save button is clicked, the input shall be validated first. All inputs must not be null and contain reasonable values (e.g. birthdate must not be in the future). If one of the inputs is invalid, an Alert dialog with an appropriate error message shall be displayed:



4. List of Persons

Refactor your application, so it shows a list of persons on its main screen. By clicking the "Add person …" button a dialog is opened, which allows to enter the person data. This dialog is mainly what you already programmed in Tasks 1 to 3. The refactored application looks like this:



Use an ObservableList<String> object, to hold the person data (one string per person, as returned by toString() of class Person).