TDT4180 MMI - Exercise 2, T2: PersonPanel form implementation

The goal of this exercise is to learn how to make a form for editing the properties of a data object. The data object has the role of model, i.e. the underlying data that the Graphical User Interface (GUI) components let the user view and control. In this exercise we will use a simple person class, with information about a person's name, date of birth, gender, email address and height.

This exercise is intentionally not a true Model-View-Controller(MVC) implementation. That follows in exercise T3, building on this exercise.

1. Create the data class with set and get functions

Create a class named Person with the following (private) fields:

name

> name is a String field for holding the person's name.

dateOfBirth

> dateOfBirth is a field of the type String.

gender

> The gender is either female or male. Define an enum (class) named Gender that is this field's type.

email

> email is a String field for holding the person's email address.

height

> height is an int field.

For each of the five fields, define corresponding public getters and setters, e.g. void setEmail(String), String getEmail().

(hint: use Eclipse's built-in support for this, in the Source menu).

2. Create the SWING panel

Create a class named PersonPanel that inherits from JPanel.

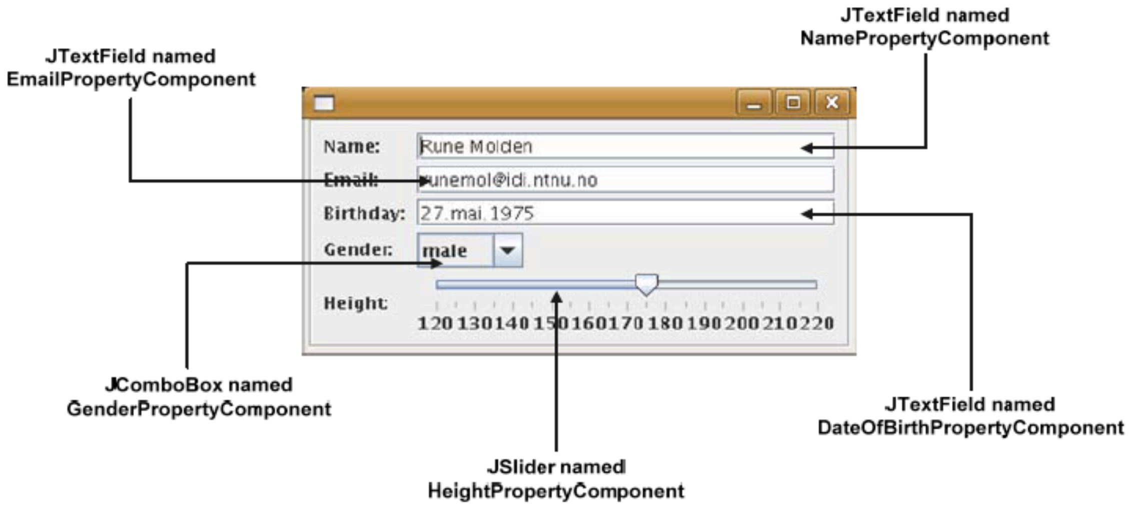
Write a main method that creates a JFrame with a PersonPanel instance as its content pane (for testing purposes).

For each of the fields in Person, the PersonPanel will contain two components:

1) a JLabel for the name of the field and

2) a field-specific component for viewing and controlling the field's value.

See the GUI below:



Name: Rune Molden

Email: runemol@idi.ntnu.no

Birthday: 27.mai.1975

Gender: male

Height:

JTextField named NamePropertyComponent
JTextField named EmailPropertyComponent
120 130 140 150 160 170 180 190 200 210 220
JComboBox named GenderPropertyComponent
JTextField named DateOfBirthPropertyComponent
JSlider named HeightPropertyComponent

The PersonPanel's constructor must create these 10 components and add them as children (of itself) (Use an appropriate Layout Manager class, e.g. GridBagLayout).

Use the following SWING components (see figure):

JTextField

> JTextField is used for viewing and controlling String values without any defined syntax.

JComboBox

> JComboBox support selection of specific set of value, typically a limited set of String or enum values.

JSlider

> JSlider support selection of an integer within an interval or range.See illustration of GUI below.

To make the unit test code work you must make sure the setName method of active components are called with the the appropriate names, "NamePropertyComponent", "EmailPropertyComponent", "DateOfBirthPropertyComponent", "GenderPropertyComponent", "HeightPropertyComponent", respectively.

3. Connect the PersonPanel object to the Person object

a) Create a private field of class Person in PersonPanel called model. Let it initially be null.

b) Create a function in PersonPanel named setModel(Person) that when called from outside sets the model of that PersonPanel to the Person argument and then updates all the SWING elements in the PersonPanel GUI by reading the corresponding fields of the person object.

c) In addition, implement a corresponding getModel function in the PersonPanel class returning the model.

d) For each of the SWING elements (name, email,,) in PersonPanel, make sure that when the user changes a value in the GUI, the corresponding new value is set in the model (a Person object). If the model is null, do nothing.

By calling PersonPanel's setModel method, other parts of the application (that are not part of this exercise) may change which Person instance the user may view and control. For testing purposes, you may call setModel(..) in the main code, with different instances of Person.