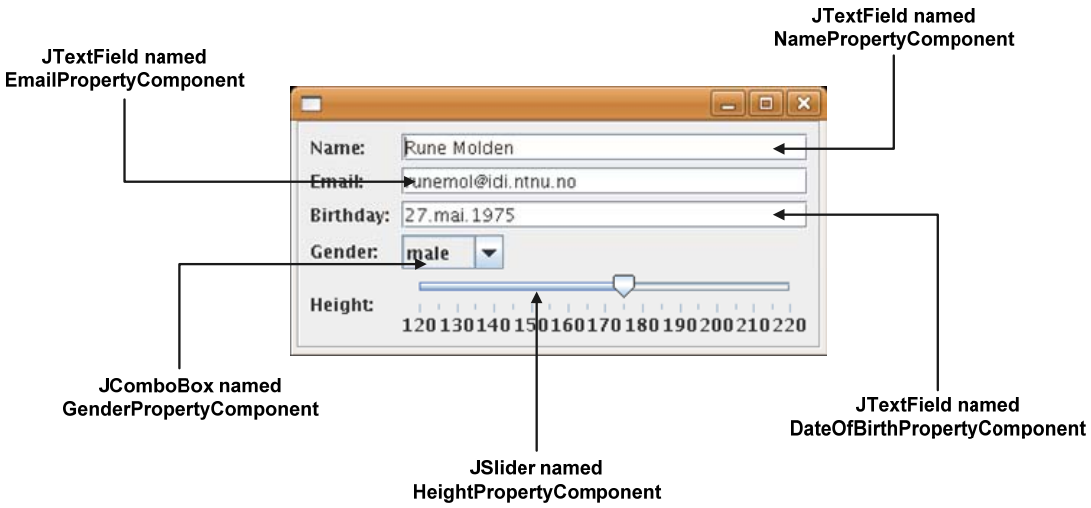
TDT4180 MMI - Exercise 3, T3: Complete PersonPanel form implementation

The goal of this exercise is to learn how to make the form in exercise T2 react to changes in the model through a full MVC implementation. The GUI is the same as in exercise T2.



Start by downloading the example code SimplePersonPanel.zip from It´s Learning. (You find it in the folder "Java/SWING eksempler"). Look at the code and understand how it works.

Extend your Person class from T2 so it generates property change events when the fields are changed by means of the setter methods. This is necessary to make your Person class completely fill the model role of the model-view-controller architecture. The property name for each field is the same as the field name.

Use the java.beans.PropertyChangeSupport class for handling the details, i.e. delegate to an instance of this class both for managing the listener list and firing the PropertyChangeEvents.

Extend the PersonPanel class from T2, so it listens to changes to the model, i.e. current Person instance, and reacts to PropertyChangeEvents by updating the field specific components. Take care to avoid updating unaffected components and to avoid an infinite loop of updating the components, changing the model and reacting to model changes.

After having tested that the code works for one PersonPanel, make a subclass of PersonPanel with all JTextFields non-editable. The method setEditable(boolean b) allows you to change this. The new subclass should be calles PassivePersonPanel. Change the gender and height fields to non-editable JTextFields, and use them to display the gender and height as text.

Make an instance of PassivePersonPanel with the same Person model as model, and add it to the window (JPanel). Changes in the PersonPanel should now automatically be updated in the PassivePersonPanel.