

STUDENTEN

We bouwen een eenvoudige app om door een lijst van studenten te gaan. Zowel het model als de view zijn al klaar, we moeten ze enkel nog koppelen door een presenter te implementeren!

De voornaamste JavaFX klassen die we voor deze oefening gebruikt hebben zijn:

- [javafx.geometry.Insets](#)
- [javafx.scene.control.Button](#)
- [javafx.scene.control.Label](#)
- [javafx.scene.control.TextField](#)
- [javafx.scene.layout.GridPane](#)



Figuur 1: eindresultaat

Raadpleeg in eerste instantie de [JavaFX documentatie](#) als je ergens vast zit!

1 WIREFRAME

Aangezien de View al geïmplementeerd is hoeven we dit niet meer te doen. Bouw als oefening de wireframe van deze app na in een tool naar keuze (bvb. mockflow.com).

2 HOOFDSCHERM AANMAKEN – MVP

De **Main** klasse, het model en de view zijn al klaar en worden al aangemaakt in de **start** methode. Voeg de presenter toe om ze te koppelen.

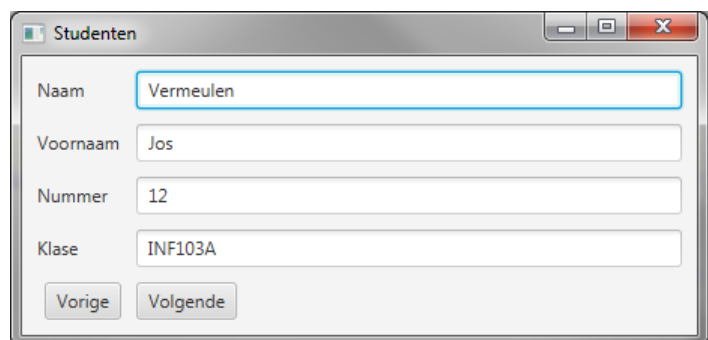
- De **Presenter** klasse:
 - Maak de **Presenter** klasse aan. Deze komt mee in het **view** package te staan.
 - Zorg voor twee attributen in de presenter: model en view
 - Schrijf een constructor die de attributen van een waarde voorziet
- De **start** methode van de klasse **Main**
 - Maak een instantie van de **Presenter** klasse

3 UI OPBOUWEN

De view is al uitgewerkt, maar kan nog beter:

- Probeer er voor te zorgen dat de tekstvelden mee groeien als het window verbreed wordt.

Tip: gebruik
`GridPane.setConstraints(...)`.



4 AFHANDELEN EVENTS

De **Presenter** klasse zorgt voor de afhandeling van de events. Als je op “volgende” klikt moet uit het model de volgende student gehaald worden en getoond worden in de view. Als je op “vorige” klikt de vorige.

4.1 De methode `handleEvents`

Voeg de `handleEvents` methode toe aan de klasse **Presenter**.

In de methode zelf moet er een eventhandler toegevoegd worden de knoppen die deel uitmaken van de view. Om dit te kunnen doen moet je eerst de nodige getters toevoegen aan de **StudentView** klasse.

Zorg er voor dat elke eventhandler de `updateView` methode aanroept met een parameter die aangeeft in welke richting de gebruiker navigeert.

4.2 De methode `updateView`

Voeg de `updateView` methode toe aan de klasse **Presenter**.

De methode zelf:

- Deze methode heeft een parameter die aangeeft in welke richting de gebruiker navigeert (m.a.w. heeft de gebruiker “vorige” of “volgende” geklikt?). Bepaal zelf een zinvol datatype voor deze parameter.
- Haal de juiste student op uit het model en toon deze student in de view. Je zal een tellertje moeten bijhouden in de Presenter om te weten welke student je op dit moment toont. Je moet ook de nodige getters toevoegen aan de **StudentView** klasse om de gegevens te kunnen tonen.
- Je doet **geen** wijzigingen in het model!