

PENSIOEN

We bouwen een super eenvoudige app om het jaar te berekenen wanneer we met pensioen zouden kunnen uitgaande van een pensioenleeftijd van 65 jaar. Hiernaast staat de wireframe van de applicatie. Deze is gemaakt met de online tool mockflow.com.



Figuur 1: wireframe (mockflow.com)

De model-klasse is al volledig klaar, de **Main** klasse gedeeltelijk. We moeten enkel de *view* en de *presenter* nog uitwerken.

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

- [javafx.scene.layout.BorderPane](#)
- [BorderPane.setLeft](#) en andere
- [javafx.geometry.Insets](#)

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

1 WIREFRAME

Bouw de gegeven wireframe na in een tool naar keuze. Hij bevat van links naar rechts een tekstveld (in te geven geboortjaar), een knop en een tekstveld waar het resultaat moet komen.

2 HOOFDSCHERM AANMAKEN – MVP

De **Main** klasse is al aangemaakt. Bestudeer de code. In de **start**-methode worden het model en de view via een presenter gekoppeld.

- Voeg de view toe in een **Scene** op de **primaryStage**.
- Zorg ervoor dat de app als titel "*Pensioenjaar*" krijgt
- Toon de **primaryStage**

3 UI OPBOUWEN

Hier werken we de klasse **Pensioenview** uit.

We hebben drie onderdelen nodig: Een **Button** en 2 **TextField**'s. Maak ze aan als attributen.

3.1 De methode **initialiseNodes**

Schrijf een nieuwe methode genaamd **initialiseNodes**. Roep deze methode op in de constructor.

In deze methode:

- Voorzie de drie attributen van een waarde.
- Het 2^{de} tekstveld mag niet editeerbaar zijn.
- Beide velden hebben bij voorkeur 4 kolommen.

3.2 De methode `layoutNodes`

We plaatsen de controls op de `BorderPane` (dit is de klasse `PensioenView` zelf).

Schrijf een nieuwe methode genaamd `layoutNodes`. Roep deze methode op in de constructor.

In de methode `layoutNodes`:

- Maak gebruik van de juiste `set...` methode van de klasse `BorderPane` om de componenten op de juiste plaats te krijgen.
- Zorg voor een horizontale en verticale "marge" van 10 pixels tussen de cellen van de `BorderPane`. Zorg ook voor een "padding" van 10 pixels.

Zoek in de documentatie van [BorderPane](#) naar de juiste methodes!

4 AFHANDELEN EVENTS

We zijn geïnteresseerd in het action event op de knop. Implementatie van de eventhandlers gebeurt in de `Presenter` klasse.

4.1 De methode `handleEvents`

De private methode `handleEvents` is aangemaakt in de klasse `Presenter` en wordt aangeroepen via de constructor. Je hoeft deze methode enkel nog in te vullen.

- Hang een eventhandler aan de knop. Om dit te kunnen doen moet je ervoor zorgen dat de button bereikbaar is via een getter die je aan de klasse `PensioenView` toevoegt (package-private).
- Zorg er voor dat de eventhandler de methode `updateView` (zie punt 4.2) oproept.

4.2 De methode `updateView`

Deze methode moet de juiste informatie uit het model ophalen en vervolgens de view up-to-date brengen met deze informatie.

- Bij het klikken op de "Bereken" `Button` moet je de view aanpassen door uit het model het pensioenjaar op te halen. Zorg ervoor dat beide `TextField` via een getter in de klasse `PensioenView` bereikbaar zijn.

5 UITBREIDING

Pas je programma aan zodat je een gebruikersvriendelijker interface hebt (op z'n minst labels toevoegen). Gebruik in `GridPane` plaats van `BorderPane`. Zijn er nog andere alternatieven?