

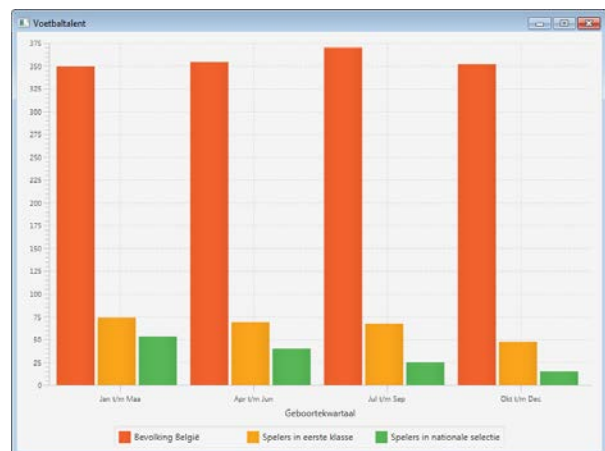
VOETBAL

In deze oefening gebruiken we JavaFX om een bar chart te tonen.

Deze opdracht is gebaseerd op [een artikel van VTM nieuws](#).

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

- [javafx.scene.chart.BarChart](#)
- [javafx.scene.chart.CategoryAxis](#)
- [javafx.scene.chart.NumberAxis](#)
- [javafx.scene.chart.XYChart](#)
- [javafx.scene.chart.XYChart.Data](#)
- [javafx.scene.chart.XYChart.Series](#)



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

1 WIREFRAME

De wireframe die bij deze oefening hoort is triviaal. Je mag deze als oefening bouwen met behulp van een tool naar keuze.

2 HOOFDSCHERM AANMAKEN – MVP

De *model* klasse is **VoetbalModel**. Je hoeft in deze klasse niets aan te passen.

De *view* klasse is **VoetbalView**. We werken deze klasse uit in punt 3.

De *presenter* klasse is **Presenter**. We werken deze klasse uit in punt 4.

De **Main** klasse is gegeven en hoeft je niet aan te passen.

3 UI OPBOUWEN

- Voeg een attribuut van het type **BarChart** toe aan de klasse **VoetbalView**.
- Schrijf een getter voor dit attribuut. Gebruik de juiste [access level](#)!

3.1 De methode initialiseNodes

- Gebruik de constructor van **BarChart** [die twee parameters heeft](#), een X-as en een Y-as. De parameters die je meegeeft aan de constructor zijn een nieuwe **CategoryAxis** (als X-as) en een nieuwe **NumberAxis** (als Y-as). Voor de assen kan je telkens de default constructor gebruiken.

3.2 De methode layoutNodes

- Plaats je attribuut centraal op je **BorderPane**.

4 AFHANDELEN EVENTS

4.1 De methode `addEventHandlers`

De methode `addEventHandlers` hebben we niet nodig voor deze opdracht.

4.2 De methode `updateView`

De methode `updateView` moet de nodige gegevens ophalen uit het model en ze tonen in de view:

- Vraag de X-as van je bar chart op uit de view en geef ze het label "*Geboortekwartaal*".
- Maak een nieuwe reeks van gegevens om te tonen in je bar chart. Zo'n reeks van gegevens noemen we een **serie**. We gaan als volgt tewerk:
 - Maak een nieuwe reeks aan om je gegevens in op te slaan.
Tip:
`XYChart.Series<String, Number> mijnSerie = new XYChart.Series<>();`
 - Zet de naam van je reeks op "*Bevolking België*".
 - **Voor elk kwartaal uit het model** voeg je een nieuw datapunt toe aan je serie:
 - Tip:
`mijnSerie.getData().add(new XYChart.Data<>(param1, param2));`
 - Neem als **eerste** parameter het **kwartaal**. (zie model)
 - Neem als **tweede** parameter het **aantal geboortes in dit kwartaal**. (zie model)
- Maak een nieuwe (tweede) serie aan en gebruik nu als **eerste** parameter het **kwartaal** en als **tweede** parameter het **aantal spelers eerste klasse die geboren zijn in dit kwartaal**.
- Maak een nieuwe (derde) serie aan en gebruik nu als **eerste** parameter het **kwartaal** en als **tweede** parameter het **aantal spelers van de nationale selectie die geboren zijn in dit kwartaal**.
- Voeg alle drie deze series toe aan je bar chart (`getData().add(...)`).