

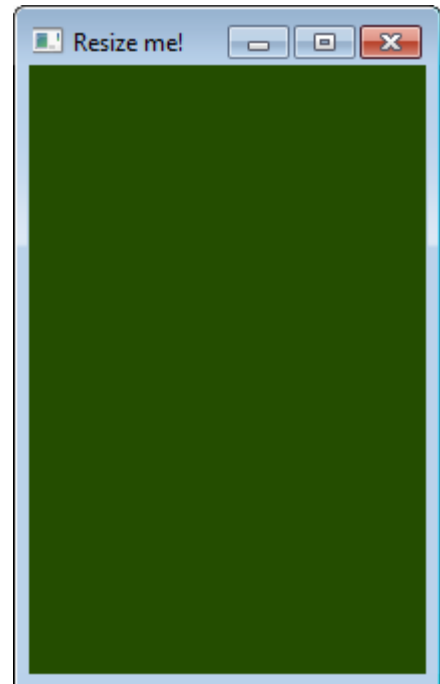
## RESIZE

We bouwen een applicatie waarvan de schermgrootte steeds wordt bewaard. Elke keer je de applicatie opstart zal het scherm dezelfde afmetingen hebben als toen je de applicatie de laatste keer hebt afgesloten.

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

- [java.io.File](#)
- [java.io.FileReader](#)
- [java.util.Formatter](#)
- [java.io.IOException](#)
- [javafx.geometry.Dimension2D](#)

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



## 1 WIREFRAME

De GUI van deze oefening is reeds uitgewerkt. Het wireframe mag je als oefening maken met behulp van een tool naar keuze.

## 2 HOOFDSCHERM AANMAKEN – MVP

Er is geen *model*. Waarom denk je dat de opslag van gegevens in deze oefening aan de view kant gebeurt?

De *view* klasse is **ResizeView**. Deze klasse is volledig uitgewerkt.

De *presenter* klasse is **ResizePresenter**. De presenter maakt gebruik van de klasse **Settings** die instaat voor het inladen en opslaan van GUI-instellingen. De **Settings** klasse werken we uit in punt 4.

De **Main** klasse is gegeven.

## 3 UI OPBOUWEN

De view is volledig uitgewerkt voor deze opdracht.

## 4 AFHANDELEN EVENTS

De klasse **ResizePresenter** is gegeven. We werken de **Settings** klasse verder uit.

## 4.1 De methode `saveWindowSize`

Deze methode heeft een parameter van het type `Dimension2D` die de breedte en de hoogte van het venster bevat. (zie `getWidth` en `getHeight`)

- Maak een **Formatter** aan en geef **SETTINGS\_FILE** (zie bovenaan de **Settings** klasse) mee als parameter aan de constructor.
- Gebruik de **Formatter** om de breedte en de hoogte van het venster op te slaan analoog aan dit voorbeeld:

214x342

Let op: we werken met **gehele** getallen!

- Vang eventuele **IOExceptions** op en druk de stacktrace af.
- Zorg er voor dat de **Formatter** in alle gevallen correct wordt gesloten.  
(Dit kan je doen met behulp van *try-with-resources* indien dit concept al behandeld is geweest in de les.)

## 4.2 De methode `loadWindowSize`

Deze methode heeft geen parameters, maar geeft wel een object terug van het type `Dimension2D`.

- Maak een **FileReader** aan en geef **SETTINGS\_FILE** (zie bovenaan de **Settings** klasse) mee als parameter aan de constructor.
- Maak een array van **char** aan (lengte 20) om te gebruiken als buffer.
- Lees, met behulp van de `read` methode, de volledige inhoud van het bestand.
- Maak een **String** aan op basis van je buffer, `trim` deze **String** en splits vervolgens inhoud van deze **String** op basis van de letter "x". Je hebt nu twee **Strings**: de breedte en de hoogte.
- Laat de `loadWindowSize` methode een nieuw `Dimension2D` object teruggeven. Gebruik `Integer.parseInt` om beide **Strings** om te zetten naar getallen zodat je ze kan meegeven aan de constructor van `Dimension2D`.
- Vang eventuele **IOExceptions** op. Indien er zich een **IOException** voordoet moet de `loadWindowSize` methode de dimensie 600 bij 400 teruggeven.
- Zorg er voor dat de **FileReader** in alle gevallen correct wordt gesloten.  
(Dit kan je doen met behulp van *try-with-resources* indien dit concept al behandeld is geweest in de les.)