Karel de Grote Hogeschool

BOODSCHAP

We bouwen een applicatie waarin je een boodschap kan ingeven met een passende voorgrond- en achtergrondkleur. Dit geheel kan je opslaan in een binair bestand dat je kan doorgeven aan anderen. Wie de opdracht tot een goed einde brengt kan alle boodschappen lezen.

In de folder van het IDEA project vind je alvast enkele bestanden die je kan proberen inlezen.

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

- `java.io.File`
- `java.io.FileInputStream`
- `java.io.FileOutputStream`
- `java.io.DataInputStream`
- `java.io.DataOutputStream`
- `java.io.IOException`
- `java.util.Base64`
- `javafx.stage.FileChooser`
- `javafx.scene.paint.Color`

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

Voor deze oefening werken we niet met een *model*.

De *view* klasse is **`BoodschapView`**. Deze klasse is volledig uitgewerkt.

De *presenter* klasse is **`BoodschapPresenter`**. Het inlezen en de opslag van gegevens gebeurt in de event handlers. We werken de presenter uit in punt 4.

De **`Main`** klasse is gegeven.

3 UI OPBOUWEN

De view is volledig uitgewerkt voor deze opdracht.

4 AFHANDELEN EVENTS

In de klasse **BoodschapPresenter** vullen we twee event handlers aan.

4.1 De event handler van het MenuItem "*Opslaan*"

- Maak een nieuwe **FileChooser** aan en geef deze de titel "*Opslaan*". Toon de aangemaakte **FileChooser** met behulp van de **showSaveDialog** methode. Als parameter geef je het venster mee van de Scene van de view:
  this.view.getScene().getWindow()
  Deze methode geeft een **File** object terug dat het gekozen bestand voorstelt.
- Kijk na of het verkregen **File** object niet **null** is. Indien niet, dan kan je aan de slag gaan, anders doe je verder niets.
- Maak een **FileOutputStream** aan op basis van het eerder verkregen **File** object (je geeft dit mee via de constructor).
- Maak een **DataOutputStream** aan op basis van de aangemaakte **FileOutputStream**.
- Via de view kan je de gegevens van het scherm opvragen:
  Bericht: **this.view.getMessageField().getText()**
  Voorgrond: **this.view.getForegroundPicker().getValue()**
  Achtergrond: **this.view.getBackgroundPicker().getValue()**
- De twee kleuren (voorgrond en achtergrond) splitsen we elks op in **drie** variabelen van het type **short**: een **short** voor rood, een **short** voor groen en een **short** voor blauw. In totaal bekomen we **zes** variabelen van het type **short**.
  De methodes **getRed**, **getGreen** en **getBlue** van de klasse **Color** geven een kommagetal terug van 0.0 tot 1.0. Dit kommagetal moeten we eerst nog vermenigvuldigen met 255 vooraleer we de waarde voor de **short**s bekomen!
- Gebruik **zes** maal de methode **writeShort** van je **DataOutputStream** om de RGB-waarden weg te schrijven.
  Eerst: voorgrond rood, groen en blauw
  Dan: achtergrond rood, groen en blauw
- Met behulp van de **write** methode van de klasse **DataOutputStream** kan je een array van bytes wegschrijven:
  mijnStream.write(Base64.getEncoder().encode(mijnBoodschap.getBytes());
  We gebruiken een base64 codering om de tekst minder vlot leesbaar te maken.
- Vang eventuele **IOException**s op en druk de stacktrace af.
- Zorg er voor dat de **DataOutputStream** 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 event handler van het MenuItem "*Openen*"

- Maak een nieuwe **FileChooser** aan en geef deze de titel "*Openen*". Toon de aangemaakte **FileChooser** met behulp van de **showOpenDialog** methode. Als parameter geef je het venster mee van de Scene van de view:
  this.view.getScene().getWindow()
  Deze methode geeft een **File** object terug dat het gekozen bestand voorstelt.

- Kijk na of het verkregen **File** object niet **null** is. Indien niet, dan kan je aan de slag gaan, anders doe je verder niets.
- Maak een **FileInputStream** aan op basis van het eerder verkregen **File** object (je geeft dit mee via de constructor).
- Maak een **DataInputStream** aan op basis van de aangemaakte **FileInputStream**.
- Gebruik **zes** maal de methode **readShort** van je **DataInputStream** om de RGB-waarden in te lezen.
  Eerst: voorgrond rood, groen en blauw
  Dan: achtergrond rood, groen en blauw
- Maak een array van **char** aan (lengte 50) om te gebruiken als buffer.
- Lees, met behulp van de **read** methode, de resterende inhoud van het bestand. Deze methode geeft ook het aantal ingelezen bytes terug. Dit aantal heb je nodig in de volgende stap.
- Maak een String aan op basis van je buffer:
  ```
  mijnBoodschap  = new String(Base64.getDecoder().decode(new String(buffer, 0, lengte)));
  ```
- Zorg er voor dat de ingelezen gegevens zichtbaar worden in de view:
  Bericht: **this.view.getMessageField().setText(???)**
  Voorgrond: **this.view.getForegroundPicker().setValue(???)**
  Achtergrond: **this.view.getBackgroundPicker().setValue(???)**
- Vang eventuele **IOException**s op en druk de stacktrace af.
- Zorg er voor dat de **DataInputStream** 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.)