

Examen Objectgericht Programmeren 2017-eerste zit.

1: Delegates

- lijst van namen inladen vanuit een file die is opgebouwd uit allemaal lijnen van de structuur:

LANDCODE, Familiennaam, Voornaam

dit moet in een methode die een lijst van Persoon-objecten returnt

(Persoon heeft 3 velden met getters en setters voor bovenstaande 3 eigenschappen)

de file staat in hetzelfde pad als de klasse(n) die worden gebruikt voor het inladen

werk met try-met-bronnen, wanneer het bestand leeg is, of wanneer er een fout optreedt tijdens het inlezen, return dan een lege lijst.

- lijst van namen inladen, deze keer vanuit een derby databank, deze staat op een server adres van server was ~"derby.gov.be", filenaam is "names", hierbij krijg je ook nog een username en password, waarmee je dan de connectie kan maken.

er zijn 4 kolommen op de databank met de namen van de 3 eigenschappen (Strings) en 1 kolom met een jaartal (int)

- geef de declaratie van de fx elementen met fx:id tabel, naamKolom en landKolom, het eerste is een TableView, laatste 2 zijn TableColumnns

- geef de argumenten die worden meegegeven bij het aanmaken van de CellValueFactory voor de naamKolom en landKolom

- maak een implementatie van een LandCel, waarin de volledige landnaam moet komen in hoofdletters (bv. bij landcode BE dus BELGIË)

landnaam kan verkregen worden met ISOCODES.getFullName(code)

in de cel moet ook een afbeelding van de vlag van het land komen, deze afbeeldingen staan in een folder met de naam "afbeeldingen"

de namen van de afbeeldingen zijn van het formaat landcode.png (bv. be.png)

- schrijf hoe je de tabel zou opvullen met z'n inhoud

- schrijf hoe, je de eerder gemaakte LandCel, zou implementeren in een CellFactory

2: Lights

- er wordt gevraagd om bepaalde delen van een MVC te schrijven. Wij kregen als oefening een verkeerslicht, met 3 circles (groen-rood-oranje) met een knop die ze doet verspringen.

Ons model was een IntegerProperty, onze views de lichten en de knop de controller. Je krijgt ook het fxml, maar er is een deel van weg gelaten, dit moet je aanvullen.

De lichten hadden ook een index, 0 was groen, 1 was oranje en 2 was rood.

- vul het fxml aan en implementeer zelf het model (met fx:define) :

```
<Items>
```

```
<Light index="0" model=*VUL IN*/>
```

```
<Light index="1" model=*VUL IN*/>
<Light index="2" model=*VUL IN*/>
</Items/>
```

- schrijf de klasse Light die Circle extends (view)
- schrijf TrafficLightsCompanion, die bij de initialize op rood start
- schrijf TesterCompanion

3: Tasks

- geef van volgende opdrachten waar ze worden uitgevoerd, de JavaFX-Thread of de nieuwe achtergrond Thread (neen Kris, het zijn Threads, geen draden):
 - cancel
 - updateProgress
 - call
 - succes

4: Void

wanneer gebruiken we Void en waarom, leg kort uit.

met dank aan thomas vankerckhoven, ruben wambacq en karel lievens, voor wie hoofdletters aan het begin van een zin des duivels zijn.