

## 4. SEW-Tests (JavaFX & JDBC)

Klasse: 3BHITM

Kat.-Nr:

Name:

4

Checken Sie das Ausgangsprojekt von Github-Classroom aus und erweitern Sie dieses um die unten beschriebene Funktionalität. Suchen Sie auch noch TODO's in den Source-Files. Führen Sie regelmäßige Commits durch und vergessen Sie nicht, das Repository für die Abgabe zu pushen!

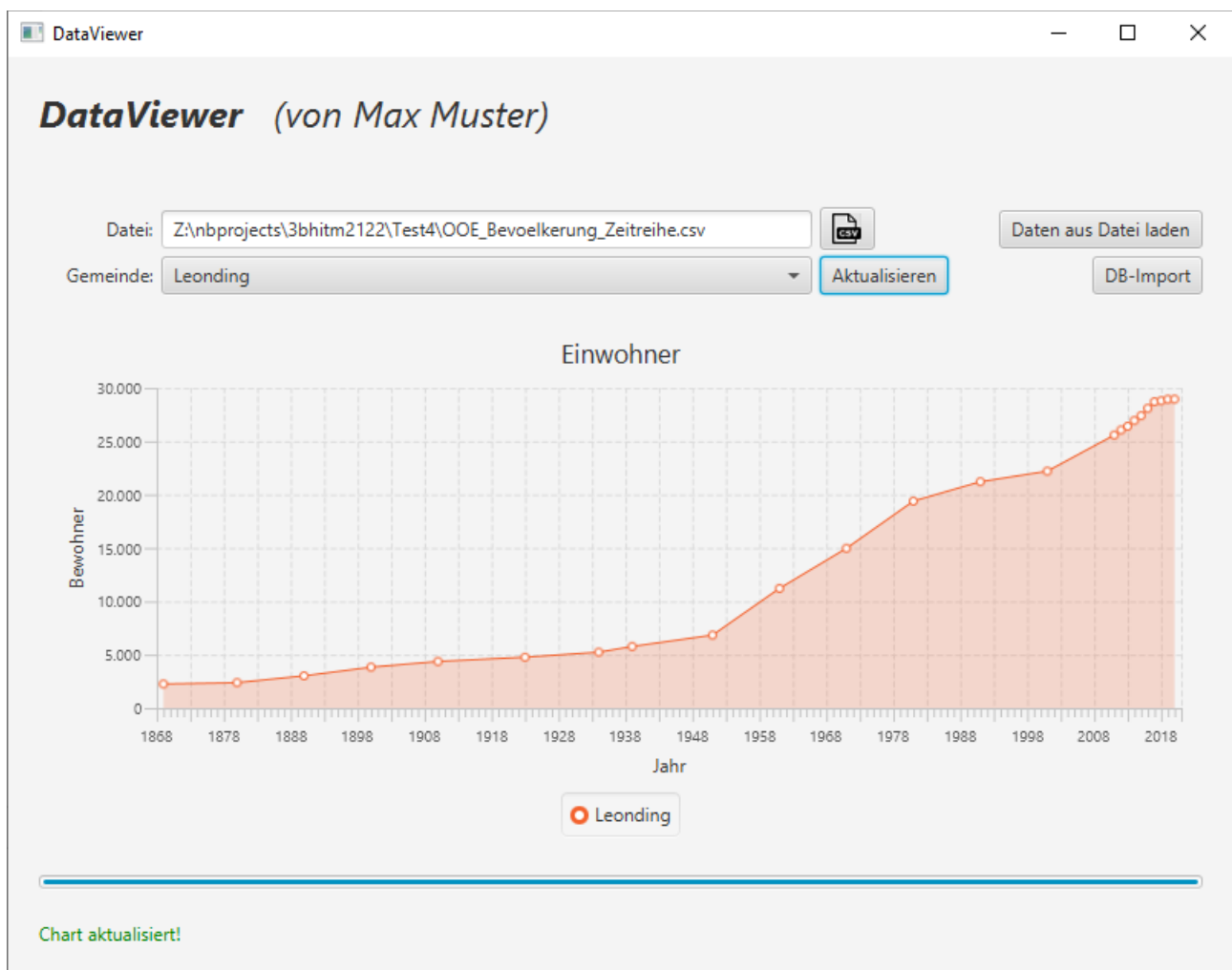
Als Datenbank ist Apache Derby zu verwenden, der JDBC-Treiber ist bereits als Dependency in der pom.xml konfiguriert. Die Laufzeitumgebung finden Sie am Angabelaufwerk!

### Allgemeine-Beschreibung

Es soll ein Tool zur grafischen Anzeige der Entwicklung der Einwohnerzahlen fertiggestellt werden. Als Datenquelle dient dazu ein vom Land OÖ bereitgestelltes CSV-File.

Der Focus des Tests liegt auf **Streams**. Versuchen Sie also möglichst **ohne Schleifen** auszukommen und die Lösungen mittels Java-Streaming-API umzusetzen.

Das fxm1-File ist bereits im Projekt vorhanden und vollständig befüllt!



**Aufgabe 1 – Gemeindeliste befüllen (Handler: importAction())****6 Punkte**

Wenn der Benutzer die Datei auswählt und den Button "Daten aus Datei laden" anklickt sollen alle enthaltenen Gemeinden in der entsprechenden ChoiceBox (**cbGemeinde**) eingetragen werden.

- Die Gemeinden müssen **alphabetisch sortiert** aufgelistet werden!
- Stellen Sie sicher, dass diese Aktion **nebenläufig** ausgeführt wird und beliebig oft wiederholt werden kann.
- Stellen Sie den Fortschritt in dem Progressbar (**pbProgress**) dar!
- Zu Beginn soll in der Statuszeile der Text "Importiere..." ausgegeben werden. Nutzen Sie dazu die vorbereitete Methode **statusAction()**.
- Sobald die Choice-Box vollständig ist, soll als Status-Text "Gemeindeliste wurde aktualisiert!" ausgegeben werden. Nutzen Sie dazu die vorbereitete Methode **statusOk()**. Im Fehlerfall kann mit **statusError()** eine entsprechende Meldung in rot dargestellt werden.

**Aufgabe 2 – Diagrammdaten aufbereiten (Handler: redrawAction())****6 Punkte**

Wenn eine Gemeinde aus der ChoiceBox ausgewählt wurde und der Button "Aktualisieren" gedrückt wird, soll das Chart aktualisiert werden.

Lesen Sie dazu wieder die CSV-Datei, filtern Sie die gewählte Gemeinde raus und erstellen Sie für jeden übrig gebliebenen Datensatz eine Instanz von `XYChart.Data(Integer,Integer)`. Diese Data-Instanzen werden dann zu einer neuen `XYChart.Series` zusammengefasst.

Einen Beispielcode finden Sie im Template!  
Arbeiten Sie auch hier wieder mit Streams!

- Auch diese Aktion ist **nebenläufig** auszuführen und muss beliebig oft ausführbar sein!
- Setzen Sie wieder entsprechende **Status-Meldungen** ("Aufbereitung der Daten...", "Chart aktualisiert!").

**Aufgabe 3 – Chart-ToolTips (Methode: addChartToolTips())****5 Punkte**

Wir möchten für jeden Datenpunkt einen ToolTip bereitstellen, indem nochmals die Jahreszahl und die exakte Anzahl an Einwohnern zu diesem Zeitpunkt angezeigt wird.

Die Logik dazu finden Sie in der Methode `addChartToolTips()`.

- Schreiben Sie diese Methode so um, dass **keine Schleifen** verwendet werden!
- Sorgen Sie dafür, dass die Methode aufgerufen wird, sobald die Diagrammdaten korrekt gesetzt wurden (also am Ende von Aufgabe 2).

#### Aufgabe 4 – Importieren der Daten in die Datenbank (Handler: dbImportAction()) 8 Punkte

Wird der Button "DB-Import" angeklickt sollen alle Datensätze des CSV-Files in eine Datenbank importiert werden. Nutzen Sie dazu eine Derby-Datenbank.

- Für jeden Datensatz sind die **Gemeinde-Nummer, Gemeinde-Name, Jahr** und **Einwohnerzahl** in die Datenbank zu übertragen.
- Nutzen Sie die Datei **init.sql** um die Datenbanktabelle anzulegen.
- Setzen Sie wieder entsprechende Status-Meldungen ("Datenbank-Import läuft...", "DB-Import abgeschlossen!")
- Auch diese Operation soll **nebenläufig** und **beliebig oft ausführbar** sein
- Lagern Sie diesen Programmteil in eine **eigene Datei / Klasse** aus!
- Die **Datenbank-Parameter** können Sie zur Vereinfachung **fix** im Sourcecode hinterlegen!
- **Deaktivieren Sie das automatische Commit** und führen Sie zur Performance-Steigerung nur **ein Commit am Ende** des Import-Vorgangs aus!
- Zeigen Sie den Fortschritt in dem **Progressbar** (pbProgress) an!

#### Aufgabe 5 – Gesamtdaten anzeigen

5 Punkte

- Erweitern Sie die Gemeinde-Auswahl dahingehend, dass als **erster Eintrag** die Option **"Gesamt"** wählbar ist.
- Wird "Gesamt" als Gemeinde gewählt, so sollen die **Einwohnerzahlen** über alle Gemeinden hinweg **zusammengezählt** werden und somit die zu dem Jahr gültige Gesamteinwohnerzahl von Oberösterreich dargestellt werden.
- Passen Sie dazu die Aufbereitung der XYChart.Data-Datensätze entsprechend an, wobei auch hier wieder Streams zu verwenden sind.  
Hinweis: **Collectors.groupingBy**, **Collectors.summingLong**, ... könnten hier hilfreich sein!

**Viel Erfolg !!**

Note		Sgt	gut	bef	gnd	ngd
Punkte	>=	27	23	19	15	0