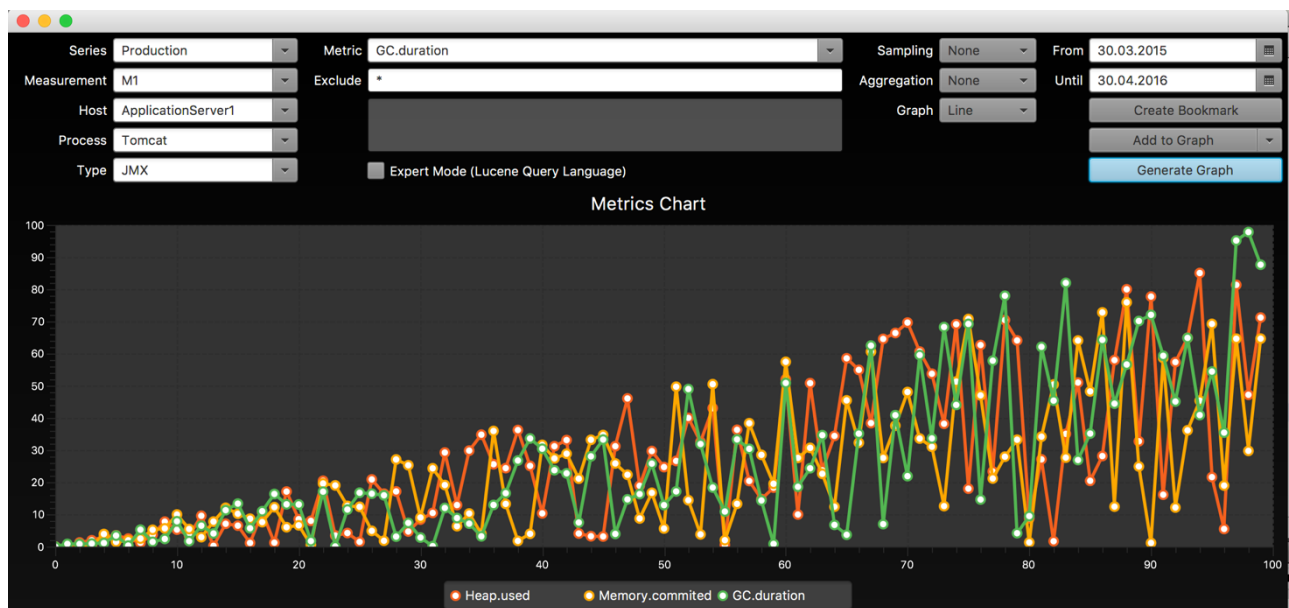


# GUI Vorlesung 2019

## Übung 5 - GUI Komponenten



### Beschreibung

Ziel der Übung ist es, eine Oberfläche bestehend aus Eingabeformular (Header) und Diagramm (Chart) zu erzeugen. Die Oberfläche aus Übung 4 soll dazu als Komponente verwendet und eingebettet werden.

### Aufgabe 1

Erstellen Sie die oben abgebildete grafische Oberfläche nach dem MVP-Pattern. Verwenden Sie dazu die Komponente „Header“ aus der letzten Übung und erstellen Sie eine neue Komponente „Chart“ die das Eingabeformular und ein Diagramm enthält.

Gehen Sie dabei wie folgt vor:

- 1) Verwenden Sie das Ergebnis aus Übung 4. Erstellen Sie ein neues Unterpaket und verschieben Sie die Inhalte dorthin. Passen Sie ggf. die Pfade an.
- 2) Erstellen Sie eine neue FXML Datei (*Chart.fxml*) mit Controller (*ChartPresenter*).
- 3) Fügen Sie (mit Scene Builder) ein vorerst leeres Diagramm (*LineChart*) ein.

- 
- 4) Ändern Sie die „CategoryAxis“ des LineCharts in der FXML-Datei auf „NumberAxis“. Editieren Sie dazu direkt das FXML in der Entwicklungsumgebung.
  - 5) Verwenden Sie im Chart den Header mittels `fx:include`.  
**Hinweis:** Falls es beim Inkludieren mit Scene Builder zu einem Fehler kommt, probieren Sie es mit einer anderen Root-Pane im Chart (Bug?!).
  - 6) Die Gesamtansicht ist jetzt hierarchisch geschachtelt (Vorlesung 2). Verwenden Sie einen geeigneten Container für das Layout.

## Aufgabe 2

Reagieren sie in der Elternkomponenten *Chart* auf Aktionen in der Kindkomponente *Header*.

Gehen Sie dabei wie folgt vor:

- 1) Erweitern Sie den Chart-Controller/Presenter um eine Methode „generateChartData“.
- 2) Holen Sie sich im *ChartPresenter* den Zugriff auf den Button des HeaderPresenter.  
**Hinweis:** Dazu ist im ChartPresenter der Zugriff auf den HeaderPresenter notwendig. Dieser kann, wie in der Vorlesung besprochen, in den ChartPresenter injiziert werden. Die `fx:id` nicht vergessen!
- 3) Registrieren Sie einen Event Handler und führen Sie eine entsprechende Methode (*generateChartData*) im Chart-Controller/Presenter aus.

## Aufgabe 3

Diskutieren Sie, warum es nicht ideal ist in der Komponente „Chart“ den Event Handler für den Button „generateGraph“ des Headers zu registrieren.

Implementieren Sie eine einfache Lösungsalternative.

## Aufgabe 4

Erzeugen Sie bei einem Klick auf „Generate Graph“ Zufallsdaten, die das LineChart befüllen.

Gehen Sie dabei wie folgt vor:

- 1) Erstellen Sie ein Model für die Chart-Ansicht, welches die Daten der Ansicht kapselt. Erzeugen Sie in dieser Klasse eine Methode „generateData“, die die Zufallsdaten erzeugt.  
**Hinweis:** Die passende Datenhaltung für ein LineChart und eine Methode um Zufallsdaten zu generieren finden Sie im Listing unten.
- 2) Verwenden Sie das neue Model im Presenter/Controller des Charts.
- 3) Binden Sie die Daten an das LineChart.
- 4) Rufen Sie in der Methode *generateChartData* die Methode des Models auf.

```
...

private final SimpleObjectProperty<ObservableList<XYChart.Series<Number,
Number>>> chartData = new
SimpleObjectProperty<>(FXCollections.observableArrayList());

public final void generateData() {
    chartData.get().clear();
    final XYChart.Series<Number, Number> series = new XYChart.Series<>();
    series.setName("Random Series");
    for(int i = 0; i < 100; i++) {
        series.getData().add(new XYChart.Data<>(i, Math.random() * i));
    }
    chartData.get().addAll(series);
}

...
```