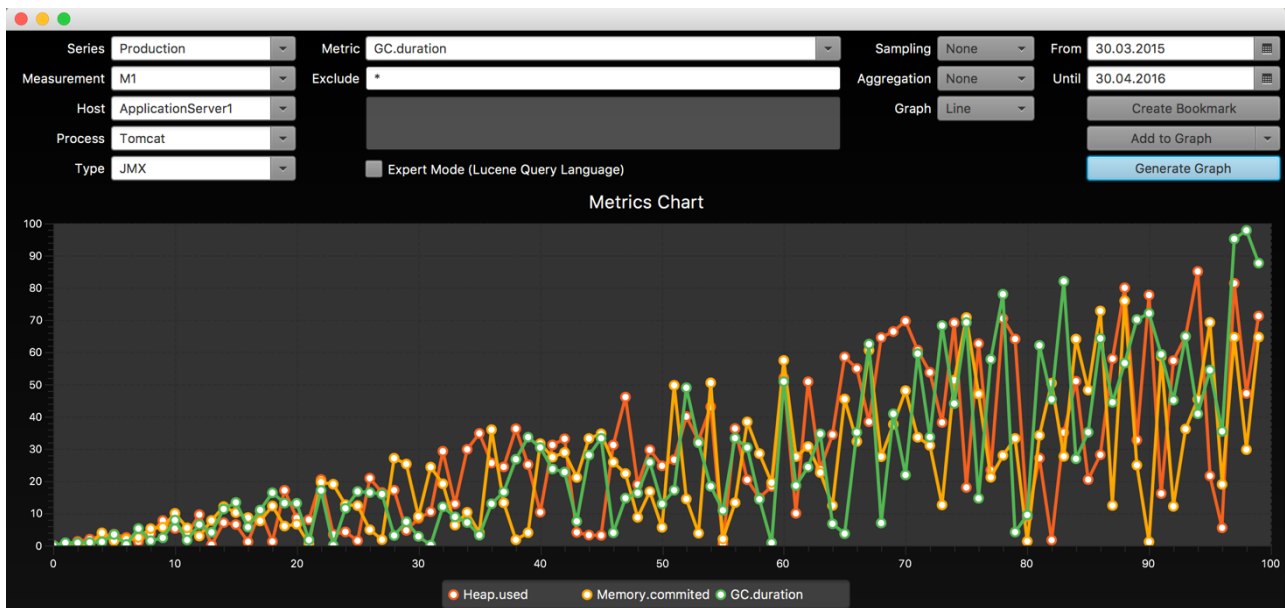


# GUI Vorlesung 2017

## Übung 5 - GUI Komponenten



### Beschreibung

Ziel der Übung ist es, eine Oberfläche bestehend aus Eingabeformular und Diagramm 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 den Header und eine Diagramm-Ansicht enthält.

Gehen Sie dabei wie folgt vor:

- 1) Verwenden Sie das Ergebnis aus Übung 4. Erstellen Sie dazu ein neues Unterpaket und verschieben Sie die Inhalte dorthin. Passen Sie ggf. die Pfade an.
- 2) Erstellen Sie eine neue Maske (*Chart.fxml*) mit einem JavaFX Line Chart.
- 3) Verwenden Sie in der neuen Chart-Maske den Header (*Header.fxml*) mittels *fx:include*. ACHTUNG: Der SceneBuilder kann anscheinend nur in AnchorPanes inkludieren. Die

---

Chart-Maske muss also vorübergehend eine AnchorPane als Root-Knoten besitzen.

Die eingefügte Komponente wird im SceneBuilder anschließend korrekt dargestellt.

**Hinweis:** Die Gesamtansicht ist jetzt hierarchisch geschachtelt (Vorlesung 2). Verwenden Sie einen geeigneten Container für das Layout.

## Aufgabe 2

Erstellen Sie für die Chart-Komponente einen Controller, der eine Methode „generateGraph“ implementiert, die beim Druck auf den „generateGraph“-Button des Headers aufgerufen wird.

Die Methode gibt den Aufruf an das ChartModel weiter, welches Zufallsdaten erzeugt (siehe unten).

Gehen Sie dabei wie folgt vor:

- 1) Ändern Sie die CategoryAxis des LineCharts in der FXML-Datei auf NumberAxis. Editieren Sie dazu direkt das XML in Netbeans.
- 2) Holen Sie sich im ChartController den Zugriff auf den Button.  
Hinweis: Dazu ist der Zugriff auf den HeaderController notwendig. Dieser kann, wie in der Vorlesung besprochen, in den ChartController injiziert werden.
- 3) Registrieren Sie einen EventHandler und führen Sie eine entsprechende Methode (generateGraph ) im ChartController aus. Die Methode soll den Aufruf an das Model weitergeben (siehe Schritt 4).
- 4) 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.

### Hinweise:

- Sie können die Daten für das LineChart per Data Binding anbinden oder mit setData eine ObservableList verwenden.
- Sie können sich mit FXCollections.observableArrayList() eine ObservableList erzeugen.
- Beispiel für die Generierung von Testdaten

```
public final void generateData() {
    seriesList.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));
    }
    seriesList.addAll(series);
}
```

---

### **Aufgabe 3**

Verwenden Sie beim Erzeugen der Testdaten als Name für die „Series“ den aktuellen Wert aus dem Eingabefeld „Series“.

### **Aufgabe 4**

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

Implementieren Sie eine einfache Lösungsalternative.