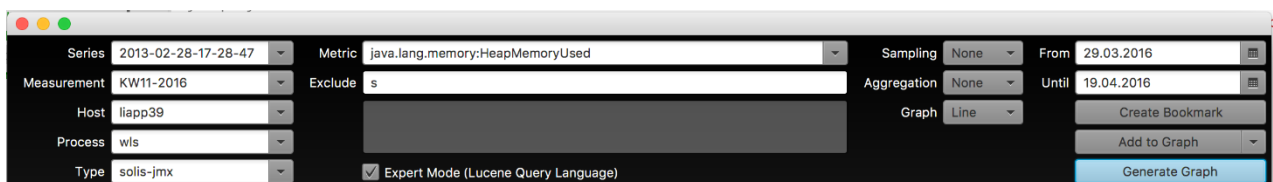


GUI Vorlesung 2018

Übung 3 - Eventverarbeitung

Beschreibung

Ziel der Übung ist es die Oberfläche aus Übung 2 mit Funktionalität zu versehen (die Musterlösung für Übung 2 ist bei Github eingeecheckt).



Beim Druck auf den Button „Generate Graph“ sollen alle Inhalte strukturiert ausgegeben werden.

Aufgabe 1

- Erstellen Sie dazu, falls noch nicht vorhanden, eine Klasse `HeaderController`. Diese Klasse wird im Scene Builder als Controller registriert.
- Erstellen Sie im Controller die Methode `void onGenerateGraph(ActionEvent e)` und registrieren Sie diese im Scene Builder.
- Testen Sie, ob beim Druck auf dem Knopf die Methode aufgerufen wird.

Aufgabe 2

- Schreiben Sie eine Klasse „`HeaderSettings`“ die alle Werte des Dialogs in einem Objekt speichern kann. Implementieren Sie eine `toSettings()` Methode im Controller um ein `HeaderSettings`-Objekt zu erzeugen.
- Beim Druck auf den Knopf soll die neue Klasse mit all ihren Attributen in der Konsole ausgegeben werden.

```
Generate Graph: HeaderSettings{
  series=2013-02-28-17-28-47
  measurement=KW11-2016
  host=liapp39
  process=wls
  type=solis-jmx
  metric=java.lang.memory:HeapMemoryUsed
  exclude=s
  sampling=None
  aggregation=None
  graph=Line
  from=29.03.2016
  until=19.04.2016
}
```

Aufgabe 3

- Verlagern Sie die Registrierung des Eventhandler in die Methode `inititalize()`. Der Controller muss dazu die Schnittstelle `Initializable` implementieren. Verwenden Sie die **JavaFX Low-Level-API** zur Registrierung des Handlers. Der Handler soll Sie die bereits erstellte Methode `onGenerateGraph` aufrufen (Siehe Aufgabe 1).

Aufgabe 4 (optional)

- Registrieren Sie Event-Filter auf mindestens zwei Ebenen.
- Verhindern Sie die Eingabe eines kleinen „a“ für alle Text-Felder.
Tipps:
 - Key-Event „`KEY_TYPED`“
 - Feld `Character` im Event