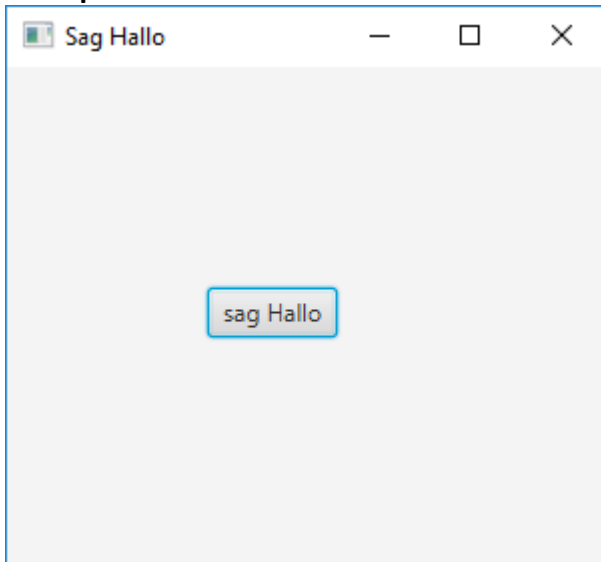


1. Aufgabe

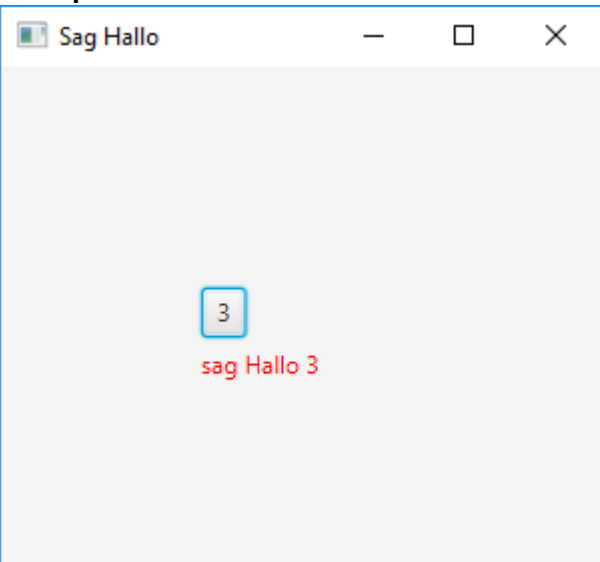
Üben Sie an diesem Beispiel das Erstellen einfacher JavaFX-Anwendungen ohne und mit FXML-File.

- a) Erstellen Sie eine JavaFX-Applikation mit main- und start-Methode. Innerhalb der start-Methode wird die GUI via Java-Code erstellt. Definieren Sie dazu eine Pane mit Button und Label für die Ausgabe. Den Zustand bei Start der Anwendung zeigt Prinzip-Skizze 1. Bei ein- oder mehrmaligen Drückens des Buttons verändert sich der Button und es erscheint eine Ausgabe entsprechend Prinzip-Skizze 2. Beim Bewegen der Maus über die GUI-Oberfläche erscheinen die entsprechenden Stage-Koordinaten (ebenfalls über das Ausgabelabel) - entsprechend Prinzip-Skizze 3.
- b) Erstellen Sie nun eine funktional identische Anwendung. Allerdings soll die GUI-Oberfläche nun via FXML beschrieben sein. Dazu benötigen Sie nun 3 Files: ein .java-File zum Starten der Anwendung, ein FXML-File und eine weitere .javaFile, welches als Controller für das FXML-File dient.

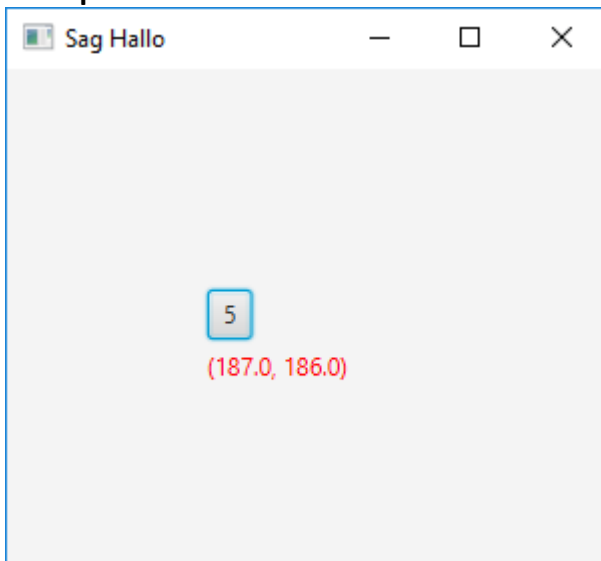
Prinzip-Skizze 1:



Prinzip-Skizze 2:

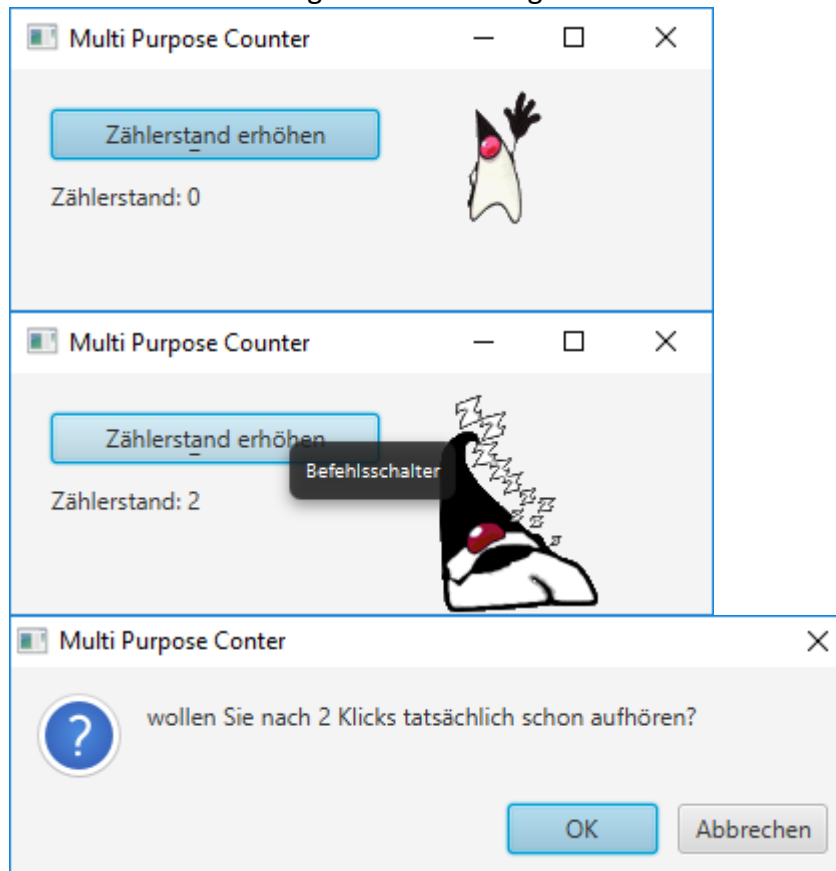


Prinzip-Skizze 3:



2. Aufgabe

Realisieren Sie nun Aufgabe 9 von Übung 5 auf Basis von JavaFX.



3. Aufgabe

Realisieren Sie nun Aufgabe 5 von Übung 6 auf Basis von JavaFX.

4. Aufgabe

Entwickeln Sie eine Java-Anwendung zur Verwaltung von Studenten. Es sollen Studenten erfasst und in einer Tabelle dargestellt werden. Dazu benötigen Sie die Klassen `Student` und `Studentendaten`. Ein Student hat Matrikelnummer (5-stellig), Name, Vorname und ECTS (zwischen 0 und 210), Semester. Die Studenten werden in der Klasse `Studentendaten` verwaltet. Die Klasse `Student` soll das Interface `Comparable` implementieren und die Methoden `equals` und `hashCode` bzgl. der Matrikelnr überschreiben (vergleiche Übung 8, Aufgabe 5).

Erstellen Sie die geeigneten Klassen zusammen mit FXML-Files und deren Controllern zur zentralen Studentenverwaltung. Die Studentenverwaltung soll folgende Funktionen bereitstellen:

- Aufruf eines eigenen Fensters zum Erfassen von Studenten. Beim Speichern wird auch das Fenster wieder geschlossen.

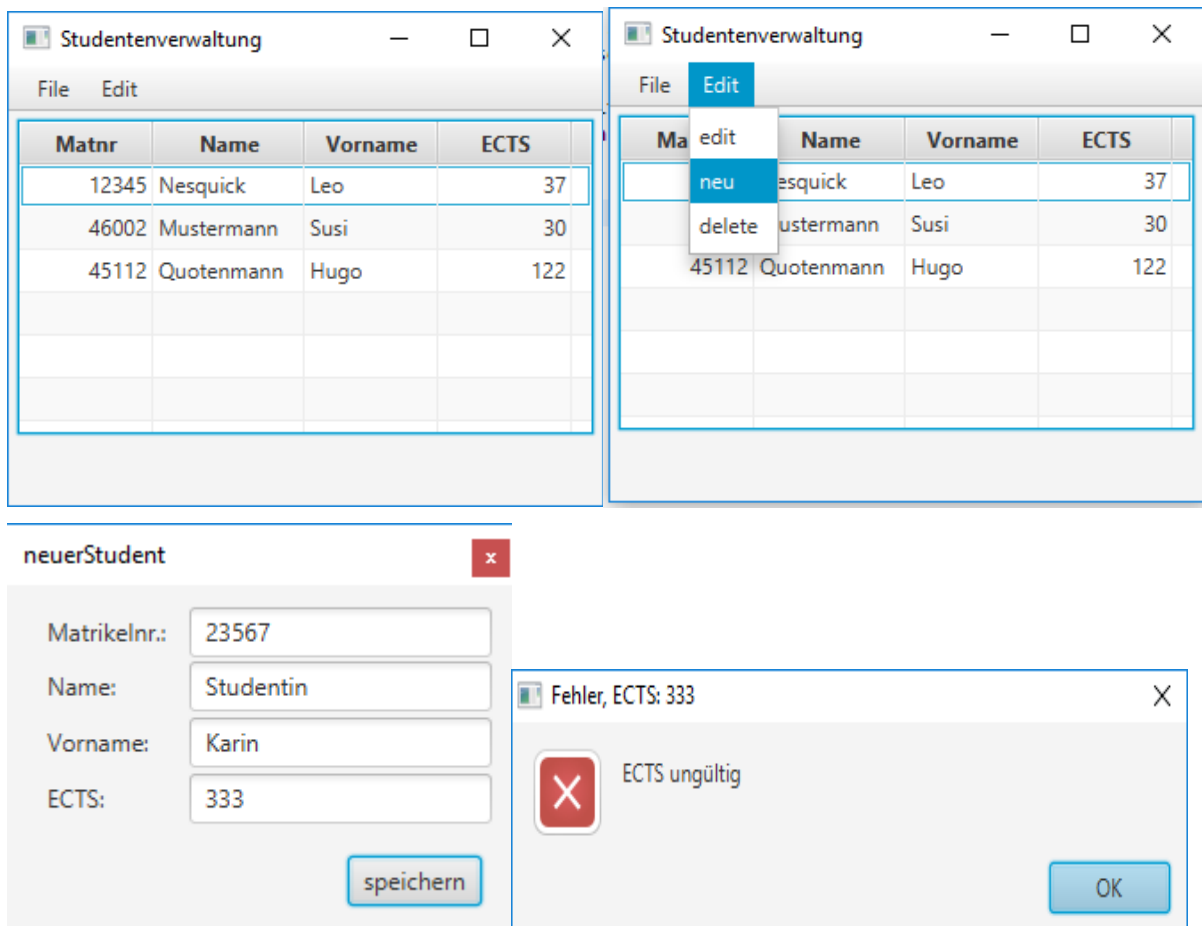
- Anzeigen der aktuellen Studentenliste via Tabelle. Selektierte Zeilen/Studenten sollen – über Menü-Item - gelöscht werden können.
- Editieren vorhandener Studentendaten durch Doppelklick in der Tabelle – alternativ kann das Menü-Item `edit` verwendet werden. Zum Ändern geht dann ein Fenster mit den jeweiligen Studentendaten auf, welches sich beim Speichern ebenfalls wieder schließt. Die Matrikelnummer kann nicht geändert werden.
- Testen Sie die Sortierfunktion der JavaFX-Tabelle, die implizit dabei ist.

Erstellen Sie Ihre Lösungsvariante so, dass es nur ein Studentendaten-Objekt geben kann (*Singleton-Objekt*).

Hinweise:

- Die Klasse Studentendaten besteht im Wesentlichen aus einer `ObservableList` vom Typ `Student` - und Methoden zum Löschen und Anfügen von Studenten sowie zum Abruf der Studentenliste.
- Die GUI für die Studentenverwaltung verwendet sinnvollerweise ein Border-Layout und stellt wesentliche Funktionen (auch) via Menü bereit.
- Bei der Erfassung von neuen Studenten muss überprüft werden, ob die Matrikelnummer schon einmal erfasst wurde.
- Warnungen/Fehlermeldungen sollen über `Alerts` kundgetan werden.

Siehe auch die folgenden Prinzip-Skizzen:



MN:45112
x

Matrikel...

45112

Name:

Quotenmann

Vorname:

Hugo

ECTS:

122

speichern

Studentenverwaltung
— □ ×

File

Edit

Matnr	Name	Vorname	ECTS
12345	Nesquick	Leo	37
46002	Mustermann	Susi	30
45112	Quotenmann	Hugo	122
23567	Studentin	Karin	92

Studentenverwaltung
— □ ×

File

Edit

Close

Matnr	Name	Vorname	ECTS
12345	Nesquick	Leo	37
46002	Mustermann	Susi	30
45112	Quotenmann	Hugo	137
23567	Studentin	Karin	92