

# CSE3OAD/CSE4OAD – Lab 3

## Building JavaFX Interfaces for Applications (draft)

The following files are provided for this lab:

HasKey.java (an interface)

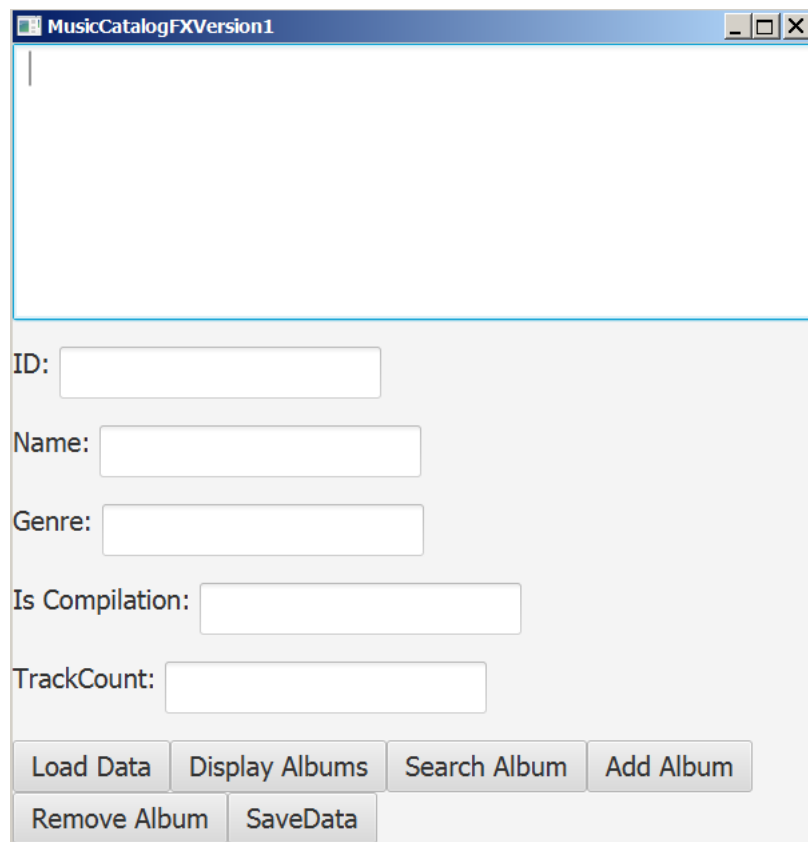
MusicAlbum.java (represents a music album)

MusicCatalogDS.java (the control class that provides the data source)

MusicCatalogFXVersion1.class (this class provides the FX interface to interact with the data source controller. This is the class to be implemented in Question 1. Only the class file is provided, which can be run to see what is to be expected)

### Question 1

Write the class MusicCatalogFXVersion1.java, which when launched presents the stage:



The screenshot shows a JavaFX application window titled "MusicCatalogFXVersion1". The window has a light gray background and a blue title bar. At the top, there is a large, empty rectangular area, likely for displaying a list of albums. Below this area, there are five text input fields, each preceded by a label: "ID:", "Name:", "Genre:", "Is Compilation:", and "TrackCount:". At the bottom of the window, there are six buttons arranged in two rows. The top row contains "Load Data", "Display Albums", "Search Album", and "Add Album". The bottom row contains "Remove Album" and "SaveData".

The screen is divided into 3 sections:

- The first section has a text area where we can display a collection of music albums, or just an single music album.
- The second section has 5 text fields, which can be used to perform various operations on the data source.
- The third section has buttons to perform operations: load data, display all music albums, search for an album, add an album, delete an album, and save data.

Include the code to perform the operations.

To see how the application works, run the provided `MusicCatalogFXVersion1.class`

(See the example `CatalogFXVersion2` given in the lecture notes and the demo programs.)

## Question 2.

- a) Make a copy of the demo program `CreateTableView.java` provided for the lecture, and modify it to present a table view for the music albums.
- b) Modify the program to provide a button to sort the rows in the table view by genre (in ascending order), and within genre by name (in ascending order).

See the demo program `CreateCustomizedSorter.java` provided for the lecture.

- c) Modify the program to add a text field that can be used to filter the rows. The table view should show only music albums whose name contains the text in the text field as a substring.

See the demo program `CreateFilter.java` provided for the lecture.

