Pawel Stroka

R00198912

Object Oriented Programming

The View

The View folder contains the code for everything that the user sees.

It consists of the code of the following:

* The app window, its dimensions, font.
* Buttons
* Text input fields
* Output area field
* All 3 tabs and its resources overall.

The View file also contains the code for the methods that allow the Controller to retrieve/use/notice any input that has been given into the Text input fields as well as methods to access the all the output fields, or the drop-down list on tab 3.  
Lastly, at the bottom it has all the methods for button listeners for every button that is built in. These methods add button listeners, but they don’t contain the logic/function that happen when the button is pressed. This part is in the Controller.

The Model

The Model folder contains two Models necessary for this project, the Student model and the Module model. Both models have their relative schema and tables created in the data.db file.

The Controller

The Controller file contains the code responsible for everything that happens behind the scenes:

* Connection to the database by using the SQLite API (which is stored in the lib folder)
* Handling what happens when a button is pressed and its exceptions
* Add/Remove/ListAll Students in the database
* Add/Remove/ListAll Modules in the database
* Refresh and insert all student ids of students in the database into the drop-down list on tab 3
* Print the information of the selected student id when the “Select” button is pressed into the output field
* Demonstrate the Memory leak for the memory leak button

App.java

The main file, run it to execute the program.

It also has a connect() method that connects to the DB/data.db file and informs the user if the connection has been successfully made, in the terminal.

Memory Leak Button

This button is to simulate a memory leak. A memory leak occurs when the heap memory has been filled to the brim, making it unable to create any more objects (or any other data types that are stored in the heap as opposed to stored in the stack). In this case, as we can see by the logic located at the bottom of the controller file, the button creates an infinite while loop that creates the student object and stores it in the Arraylist, which is a datatype stored in the heap, eventually, because there is no freeing of memory, the application will run out of memory to store any more students, leading to a memory leak aka Out of memory error.

Missing Content

Unfortunately, I couldn’t manage to find a way to finish some points but I am okay to admit that.

* Unit tests
* Manage grades in DB
* Grade Integration/Implementation
* Potentially more but I am aware that I am missing the ones mentioned above.

Note: When I Tried and Run the program with “Run Code” in VScode it failed to recognize any files, (Model/Controller/View), but it worked just fine when I Ran it with “Run java”. Please let me know if there is an issue with the execution of it from your side.