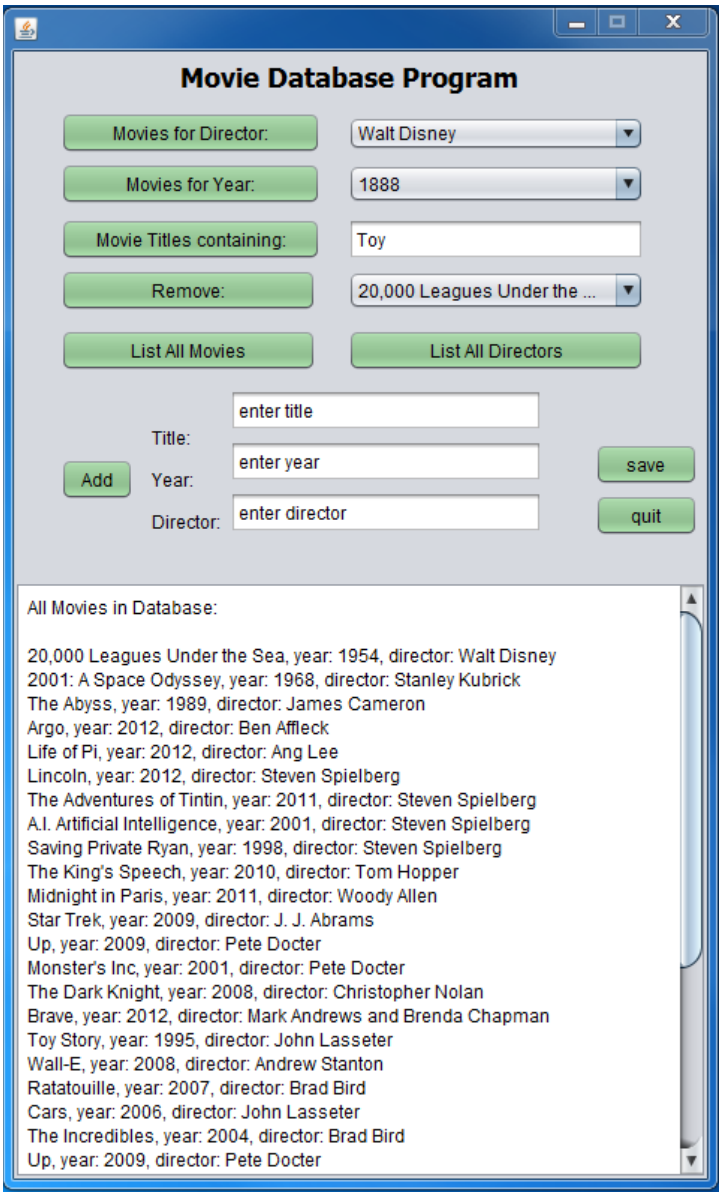
CS-141 Introduction to Programming
Creating a Simple Database – Part 3

In the final part of this lab, you will add a few additional queries and create a graphical user interface for your database. See the example below of the GUI for the movie database.

This lab will count 45 points (previous labs were worth 20 points each).

Sample Graphical User Interface for a Movie Database



Instructions:

1. **Design:** Before beginning to code, do a rough sketch of what you want your interface to look like. To do the actual coding, you may use the Netbeans interface builder, or you may write the code yourself from scratch. It is recommended that you use the Netbeans interface builder.
2. **Project:** Create a new Netbeans project. Do not select "Create Main Class".
3. **Main Frame:** If you plan on using the Netbeans interface builder, add a new `JFrame` *Form* to your project (right click on the project name and select `New -> JFrame form …`)
4. **Your Database:** Add a database object as an *instance member variable* to your extended JFrame.
5. **GUI Components:** Add the following items in your GUI (see picture above as example):
    a. **Title Label:** Use a `JLabel` to create a title bar.
    b. **Quit Button:** Quit the program. The `ActionListener` for this button just calls `System.exit(0)`.
    c. **Add Button:** Add a movie. You will need to get the new data from `JTextFields.`
    d. **Remove Button:** Remove a movie. The user can use a `JComboBox` to select which movie to remove. After `initComponents()` is called in your `JFrame` constructor, you will need to set the items in the `JComboBox` list. The easiest way to achieve this is to write a method in your database class that returns an `ArrayList` containing the items to place in the `JComboBox` list. We will go over this in class.
    e. **List All Button:** List all items in database.
    f. **Save Button:** Save all movies to a file which has the same format as the input file. You may need to either modify your `toString()` or create a new method altogether, e.g. `movieDBtoString()` in your database class. The format should be such that you can use this output file as an input file for your program.
    g. **List Button for Item :** Miscellaneous queries related to your database, e.g.
        i. Movies for Director
        ii. Movies for Year

        In order for the user to identify which director or which year, you will again need to use a `JComboBox`**.** Note, you when you create the ArrayList, you will need to avoid duplicates (see practice problem from class).

    h. Output Window: The output window is created by placing a `JScrollPane` in the frame, followed by placing a `JTextArea` inside the `JScrollPane`.

6. **Good Coding Practice:** *It is important that you name your components so they are easy to identify in the code (I don't want to see variable names such as `JTextField1` !!)*

7. **Testing**:
    a. You should test your code thoroughly.

b. In class & lab, on the day the program is due, each person will test at least one other person's program (similar to what we did for the expert system). An evaluation form will be provided in class (see page below to see what it will look like).
c. To do the testing, you will need to
    i. place your Netbeans program on our shared network drive called CS141. Directions are on the *Links* page. This will also be explained in class. In any case, your Netbeans project needs to be available!
    ii. Print a snapshot of your interface and give it to the person evaluating your code (see instructions for taking a snapshot given on the lab webpage)
d. If for any reason, you must miss class on day your program is due, it is expected that 1) your code is completed on time and 2) your code is available for someone in the class to test (e.g. email the zipped project to the professor prior to the class or place it on the shared drive).

8. **Submission**: Zip together your entire Netbeans project and submit it to WISE.

CS-141 Introduction to Programming
Database System Evaluation

Your name: _____ The name of the person whose code you are evaluating: _____

The content of their database (e.g. Movies): _____

1. Were you provided a snapshot of their graphical user interface? _____
   (If not, you will need to print it.)

2. Are the following components present?
   a. Title……Yes / No
   b. Save Button……Yes / No
   c. Quit Button……Yes / No
   d. List all Button……Yes / No
   e. Add Button……Yes / No
   f. Remove Button……Yes / No
   g. List all X (where X is a category, e.g. Director) ……Yes / No
   h. Other buttons? If so, what are they?
   i. An output area with scroll bars……Yes / No

3. Is the interface easy to understand? For example, are the buttons labeled and placed well.

4. Test each button. Does each do what it is supposed to do? If yes, put a check mark next to the button on the snapshot. If not, circle the button on the snapshot and indicate below what is wrong or confusing.

5. Do you have other comments, good or bad?

6. When done, staple this evaluation (first page) to the snapshot (second page) and turn it in.