# Lab 6B: Dinner & A Movie GUI

## Description

Add a GUI input screen to the Dinner & A Movie application. The screen will input the guest names for your dinner out.

At least one guest name is required to move on in the application, and at most 4 guest names may be entered.

Once a guest name has been input, it should display on the Guest List.

## Requirements

1. **Do NOT use NetBeans** or another Swing/GUI IDE for this. Please use a code editor to write the code manually.
2. Import classes from **javax.swing** and **java.awt** packages as needed. Only import the classes you use in your application.
3. Modify the **DoubleDate** class to extend the **JFrame** class.
4. Declare the following instance variables:
   1. JPanel inputPanel
   2. JPanel guestList
   3. JLabel addGuestPrompt
   4. JLabel guestListHeader
   5. JTextField newGuestName
   6. JButton addGuest
   7. JButton letsGo
5. Modify the **DoubleDate** constructor to configure the GUI components.
   1. Remove the parameters.
   2. Remove the addition of names to the **guests** ArrayList. Names will now be added through the GUI.
   3. Call the **JFrame** constructor by calling the **super** constructor and pass the title “Double Date” as a String argument.
   4. Set the layout of the **DoubleDate** by calling the **setLayout** method, which is inherited from the **JFrame** class.
      1. Set the layout to be a new **GridLayout** with 2 rows and 2 columns.
   5. Instantiate **addGuestPrompt** with the text “Enter a guest name:”. Using the **add** method which **DoubleDate** inherits from **JFrame**, add **addGuestPrompt** to the **JFrame** layout.
   6. Instantiate **guestListHeader** with the text “Guest List”. Using the **add** method which **DoubleDate** inherits from **JFrame**, add **guestListHeader** to the **JFrame** layout.
   7. Instantiate **inputPanel** as a new **JPanel**. Set the layout of this **JPanel** to a new **FlowLayout** using its **setLayout** method.
   8. Instantiate the **JTextField** object as a new **JTextField** with a size of 20.
   9. Add the **JTextField** object to the **inputPanel**, using the **add** method of the **inputPanel**.
   10. Instantiate the **addGuest** button as a new **JButton** with the text “Add Guest to List”. Add it to the **inputPanel**.
   11. Instantiate the **letsGo** button as a new **JButton** with the text “Let’s Go Out!”. Add it to the **inputPanel**.
   12. Set the visibility of the **letsGo** button to **false**. It should only become visible when there is at least one guest on the Guest List.
   13. Add the **inputPanel** to the **DoubleDate** **JFrame** layout.
   14. Instantiate the **guestList** as a new **JPanel**. Set its layout as a **new FlowLayout**, and add it to the **DoubleDate** **JFrame** layout.
   15. Set the size and default close operation of the **DoubleDate** **JFrame** using the following lines of code:

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(500, 250);

1. The layout of the GUI is complete, however it will not appear or do anything yet.
2. Add new method to the **DoubleDate** class: **public void goOnDate(DoubleDate date)**
   1. Copy everything from the existing **main** method into the **goOnDate** method, except for the declaration of the **DoubleDate** object (the declaration should stay in the **main** method).
3. Update the **main** method:
   1. In the main method, there should be a declaration of a new **DoubleDate** object, and we will now make the **DoubleDate** GUI visible.
   2. DoubleDate date = new DoubleDate();
   3. date.setVisible(true);
4. Add the **JFrame** **paint** method:

public void paint( Graphics g ) {

super.paint( g );

}

1. Add Action Listeners
   1. Each **JButton** needs an **ActionListener** to process button click events.
   2. To learn both methods, one action listener will be implemented as a **nested private class** and the other action listener will be defined as an **anonymous class**.
   3. **private class AddGuestHandler implements ActionListener**
      1. **public void actionPerformed(ActionEvent e)**
         1. Get the guest name using the **getText()** method of the **newGuestName** **JTextField** object.
         2. Add the new guest name to the **guests** ArrayList.
         3. Reset the value of the **newGuestName** **JTextField** to an empty String using the **setText(“”)** method.
         4. Add a **new JLabel** to the **guestList** **JPanel** containing the new guest name.
         5. Set the visibility of the **letsGo** **JButton** to true, as we now have at least one guest.
         6. By passing a condition to the **setVisible(boolean b)** method of the **addGuest** **JButton**, set the visibility to true if there are less than 4 guests or false if there are 4 or more guests.
            1. Use the **ArrayList size()** method to determine the number of guests.
            2. **Do NOT declare a boolean variable.**
         7. Because we have added new components to the **guestList** **JPanel**, we need to validate it.

guestList.validate();

* + - 1. Because the components of our **JFrame** have changed, it must be repainted. However, we can **repaint** the **guestList** **JPanel** alone, as it is the only piece that has changed. This is more optimal than repainting the entire **JFrame**.

guestList.repaint();

* 1. Add a new instance of the **AddGuestHandler** class to the **addGuest** **JButton** using the **addActionListener(ActionListener l)** method.
  2. For the **letsGo JButton**, we will define an **anonymous** class inside of the **addActionListener(ActionListener l)** method.
     1. In the **DoubleDate** constructor, after the **letsGo** **JButton** is instantiated, add the following lines of code:

letsGo.addActionListener( new ActionListener() {

public void actionPerformed( ActionEvent e ) {

goOnDate((DoubleDate) SwingUtilities.getRoot(

(Component) e.getSource()));

setVisible(false);

dispose();

}

} );

* + 1. In the **actionPerformed** method:
       1. The first line gets the root **JFrame Component** of the **Component** which was the **source** of the **ActionEvent**. In this case, that root **Component** happens to be the **DoubleDate** object, so we can cast it as such.
       2. We then pass the root **DoubleDate** **JFrame** to the **goOnDate** method, which takes a **DoubleDate** object as a parameter.
       3. Then we set the visibility of the **JFrame** to false, so it disappears from the screen.
       4. Finally, we **dispose** of the **JFrame** object, removing it from memory and terminating the program.

## References & Study Material

* [Using Top Level Containers](https://docs.oracle.com/javase/tutorial/uiswing/components/toplevel.html)
* [How to Use GridLayout](https://docs.oracle.com/javase/tutorial/uiswing/layout/grid.html)
* [How to Use Labels](https://docs.oracle.com/javase/tutorial/uiswing/components/label.html)
* [How to Use Text Fields](https://docs.oracle.com/javase/tutorial/uiswing/components/textfield.html)
* [How to Use Buttons, Check Boxes, and Radio Buttons](https://docs.oracle.com/javase/tutorial/uiswing/components/button.html)
* [javax.swing.Component setVisible method](https://docs.oracle.com/javase/7/docs/api/javax/swing/JComponent.html#setVisible(boolean))
* [SwingUtilities.getRoot(Component c)](https://docs.oracle.com/javase/7/docs/api/javax/swing/SwingUtilities.html#getRoot(java.awt.Component))
* [Nested Classes](https://docs.oracle.com/javase/tutorial/java/javaOO/nested.html)
* [Anonymous Classes](https://docs.oracle.com/javase/tutorial/java/javaOO/anonymousclasses.html)
* [Painting in AWT and Swing](http://www.oracle.com/technetwork/java/painting-140037.html)

## Enhancement / Study Ideas

* Change the font of the JLabel components.
* Change the background color of the JFrame or JPanels.
* Add borders to the JPanel components.
* Change margins and alignments of components.
* Add another JFrame (or reconfigure the existing one) to add menu items to the restaurant, or to place orders.
* Create a GUI to display the final bill and/or movie tickets, instead of outputting to the console.
* Find images of both sides of a coin, and implement a GUI coin toss to select the movie time.
* Validate that guest names are unique.
* Display error messages if guest names are unique or too short.