
Programming Assignment 10: Property Tax

Total Points (40 points) - Due Monday, December 4th at 11:59 PM

Overview

This programming assignment is intended to demonstrate your knowledge of the following:

- Create a basic JavaFX application with a closeable window.
- Create **Label** controls, **TextField** controls, and **Button** controls.
- Arrange controls in a layout container.
- Construct a scene graph and set it to the stage.
- Write an event handler that executes when a **Button** is clicked.

A JavaFX **TextField** control enables users of a JavaFX application to enter text which can then be read by the application.

The JavaFX **TextField** control is represented by the class `javafx.scene.control.TextField`.

The JavaFX **Label** control can display a text or image label inside a JavaFX GUI. The label control must be added to the scene graph to be visible.

The JavaFX **Label** control is represented by the class `javafx.scene.control.Label`.

A JavaFX **Button** control enables a JavaFX application to have some action executed when the application user clicks the button.

The JavaFX **Button** control is represented by the class `javafx.scene.control.Button`. A JavaFX **Button** can have a text and an icon on it which indicate to the user what clicking the button will do.

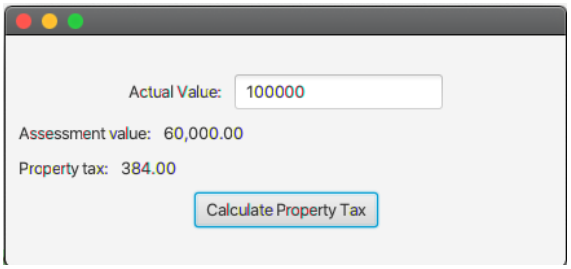
Introduction

In this assignment, you will create a GUI application that displays the assessment value and property tax when a user enters the actual value of a property with a **JavaFX** user interface.

A county collects property taxes on the assessment value of property, which is 60 percent of the property's actual value. If an acre of land is valued at $10 thousand, its assessment value is $6 thousand.

The property tax is then $0.64 for each $100 of the assessment value. The tax for the acre assessed at $6,000 will be $38.40.

The completed user interface will look like this:



In your code, you will create a **Scene** containing the following controls:

- various **Label** controls to display text

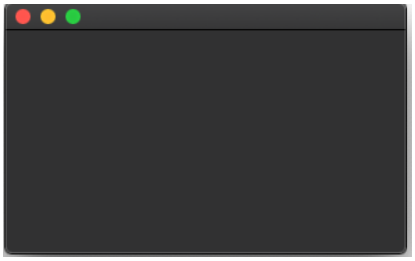
- a **TextField** control to read the user's input
- a **Button** control to calculate and display the property tax

For a JavaFX **TextField** to be visible, the **TextField** object must be added to the scene graph. This means adding it to a **Scene** object, or as child of a layout which is attached to a **Scene** object.

All of these controls will be placed into a **VBox** layout container to achieve a vertical arrangement. The **VBox** will then become the root node in the **Scene**. You will also write an event handler that responds when the user clicks the **Button** control.

Step #1: Write a Program

1. Start a new Java source code file.
2. Write import statements for the following JavaFX classes:
   - `javafx.application.Application`
   - `javafx.stage.Stage`
3. Create a class named **PropertyTax** that inherits from the **Application** class.
4. Inside the **PropertyTax** class, write a main method that calls the **launch** method.
5. Inside the **PropertyTax** class, write a **start** method that accepts a **Stage** object as an argument. The method should call the **Stage** object's **show()** method.
6. Save the file, and then compile and execute it. You should see an empty window. Close the window.



Step #2: Create the Controls

1. Write import statements for the following **JavaFX** classes:
   - `javafx.scene.Scene`
   - `javafx.scene.control.Label`
   - `javafx.scene.control.TextField`
   - `javafx.scene.control.Button`
   - `javafx.scene.layout.VBox;`
   - `javafx.scene.layout.HBox;`
   - `javafx.geometry.Pos;`
   - `javafx.geometry.Insets;`
2. Inside the **Start** method:
   - Write code that instantiates the control for the *actual value*. Then, write code that creates an HBox layout container, and place the control that you created inside the HBox. You should have 10 pixels of spacing between the controls in the HBox. Set the HBox container's alignment to *center*.
   - Write code that instantiates the controls for the *assessment value output* label. Then,

write code that creates an HBox layout container, and place the control that you created inside the HBox. Set the HBox container's alignment to *center left.*

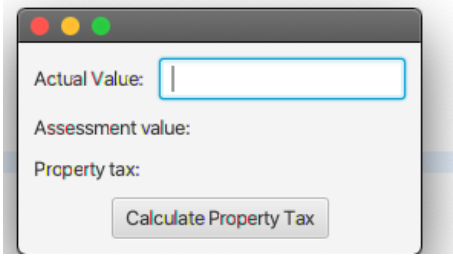
- Write code that instantiates the controls for the *property tax output* label. Then, write code that creates an HBox layout container, and place the control that you created inside the HBox. Set the HBox container's alignment to *center left.*
- Write code that instantiates the controls for the **calcButton** control. Then, write code that creates an HBox layout container, and place the controls that you created inside the HBox. Set the HBox container's alignment to *center*.

Step #3: Place the Controls in a VBox Layout Container and Complete the Scene Graph

1. Inside the **Start** method, write code that creates a VBox layout container, and place the controls that you created in Step #2 inside the VBox. You should have 10 pixels of spacing between the controls in the VBox.
2. Set the VBox container's alignment to center.
3. Call the VBox container's **setPadding()** method to set 10 pixels of padding around the VBox.
4. Instantiate a **Scene** and set the VBox as the root node.
5. Set the scene to the stage, and call the stage's **show()** method to display it.
6. Save the file, and then compile and execute it. The user interface should look like this:
7. Close the window to stop the application.
8. Create the assessment value output labels.

Step #4: Write an Event Handler for the Button Control

1. Write an event handler that executes when the user clicks the **Button** control. The event handler should calculate the assessment value and the property tax. Then, display that value in the bottom **Label** control.
2. Save the file. Then, compile and execute it.



CLASS NAME: Your program class should be called **PropertyTax.java**

Submission Instructions

- Execute the program and copy/paste the output that is produced by your program into the bottom of the source code file, making it into a comment. I will run the programs myself to verify the output.
- Make sure the run "matches" your source. If the run you submit could not have come from the source you submit, it will be graded as if you did not submit a run at all.
- README.doc (You must edit this and insert your own screenshot, or a sample run of your program.)
- Use the assignment submission link to submit the source code file.
- Submit the following files:
    - **PropertyTax.java**
    - **README.doc**
- Zip your Java files together to create one compressed file. Example: **hibrahim_assignment10.zip**
- Upload the compressed file into Canvas.