CIS 279 HW 05 JavaFX Basics

A graphical interface to an application that calculates loan payments presents a good option for a basic JavaFX project. It will present the user with TextFields for collecting loan data – annual interest rate, term in years and amount loaned. The application will pass this data into an object of the Loan class that you created in a previous assignment. In addition to loan data, the application will collect income data – salary and wages, interest income, investment income and other income. The purpose of collecting income data is to use it for determining a potential borrower's eligibility for a loan of a given amount. The example below shows how this interface should look. Clicking "Calc Payment" will do the following:

- Calculate the potential borrower's total income as the sum of salaries and wages, interest income, investment income and other income

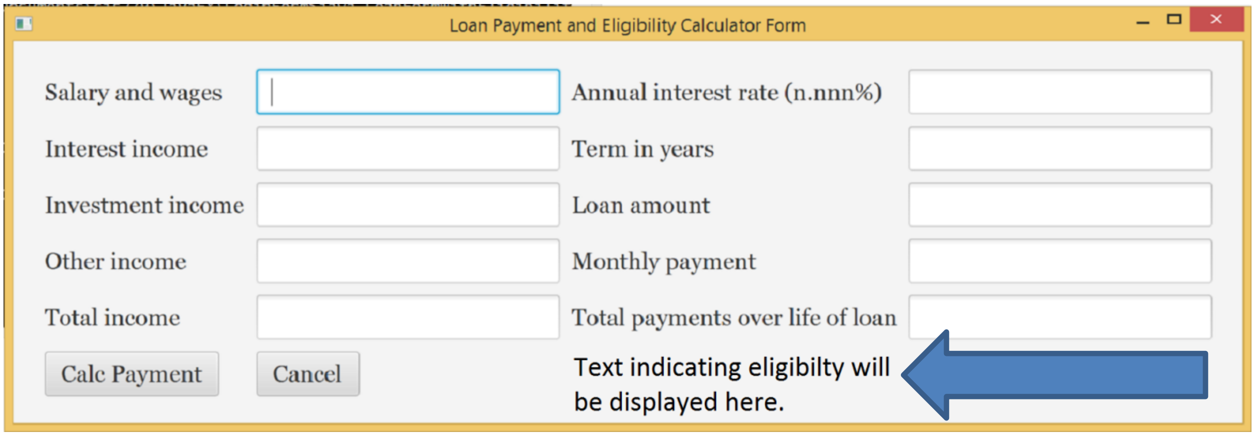
- Use an object of the Loan class to store loan data in its attributes and calculate the periodic payment and sum of payments over the life of a loan which has the specified interest rate, term and amount

- Indicate, by displaying a message in the position shown in the sample, whether the potential borrower is eligible for the loan based on whether the periodic payment divided by one-twelfth of total income (the monthly income) is less than or equal to 25%. If the loan payment is greater than 25% of the monthly income, the potential borrower is not eligible for the loan. Incorporate the actual calculated percent in the message.

Because the percent used for determining eligibility may change, use a named constant placed close to the beginning to represent it.

Clicking "Cancel" will close the application.

Sample interface:



Text indicating eligibilty will be displayed here.

Use whichever style of implementing EvetHandlers you prefer:

the lamba style of EventHandler shown here and demonstrated in LoginFormGridPaneV3.java:

```
btn.setOnAction((ActionEvent e) ->
{
        setNotifications();
});
```

or an object of a named EvenHandler class shown here and demonstrated in LoginFormGridPaneV3:

```
ClickSignonButton signonBtnHandler = new ClickSignonButton( this );

btn.setOnAction(signonBtnHandler);

class ClickSignonButton implements EventHandler<ActionEvent>
{
        LoginFormGridPaneV3 formObj = null;

        public ClickSignonButton(LoginFormGridPaneV3 formObj)
        {
                this.formObj = formObj;
        }

        public void handle(ActionEvent e)
        {
                formObj.setNotifications();
        }
}
```