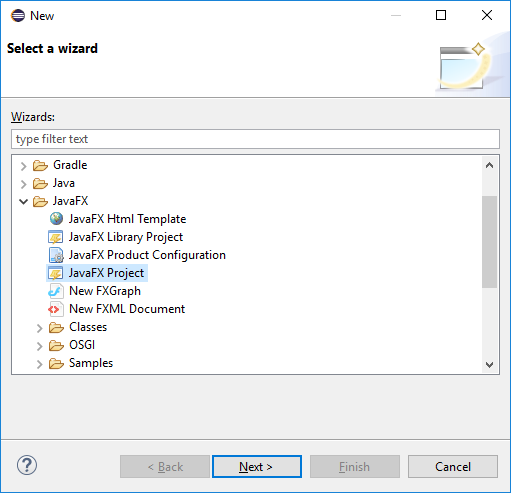
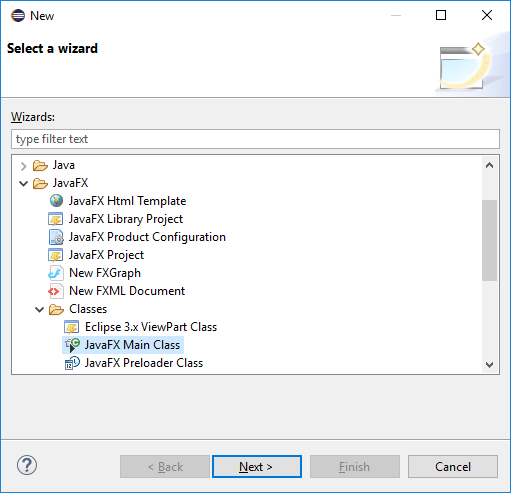
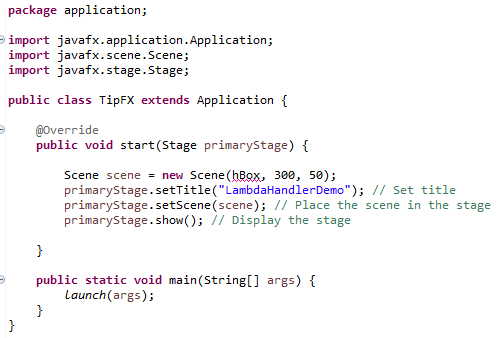
**Creating a JavaFX GUI Application**

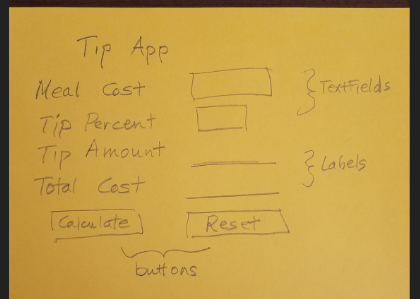
1. From the menu in Eclipse, select **File > New > Other** and expand JavaFX.
2. Select JavaFX project and click Next.



1. Enter a project name and click Finish.
2. Expand your project all the way to the **application** package in the Package Explorer. You will see a file named Main.java that has a skeletal JavaFX program that you can complete.
3. If you want to add another JavaFX program to the project…
   1. Click on the **application** package in your project
   2. Menu select **File > New > Other > JavaFX > Classes** and select JavaFX Main Class and click Next
4. Enter a legal class name and Finish.
5. The easiest way to continue is to **copy & paste the final few lines of any example into the start()** method. You will likely have to change the lines that create the Scene and set the title.



1. Next, step away from the computer. Get a pencil and paper and **make a rough sketch** of the GUI that you want to create. This will help you choose the best **layout and controls** needed for the GUI. I got this for a **Tip program**.



1. This sketch suggests that a **GridPane with 5 rows and 2 columns** should work nicely.
2. That being done, return to the computer and make a gridpane and start adding controls.
3. Use **Source > Organize Imports** and select the **javafx** option where imports are needed.
4. Before coding any event handers, run your program to see if the layout is as desired. If not, fix it.
5. Use **lambdas** (page 605 – 609) to handle events like button clicks.
6. Test your GUI, and check the math with a calculator.

**public** **class** TipFX **extends** Application {

**private** NumberFormat fmt; // so it's visible everywhere in class

@Override

**public** **void** start(Stage primaryStage) {

fmt = NumberFormat.*getCurrencyInstance*();

GridPane grid = **new** GridPane();

grid.setPadding(**new** Insets(12,12,12,12));

grid.setHgap(6);

grid.setVgap(6);

grid.setStyle("-fx-background-color:papayawhip;");

// add the controls, first column

grid.add(**new** Label("Meal Cost"),0,0);

grid.add(**new** Label("Tip Percent"),0,1);

grid.add(**new** Label("Tip Amount"),0,2);

grid.add(**new** Label("Total Cost"),0,3);

Button btCalc = **new** Button("Calculate"); // button cannot be anonymous

grid.add(btCalc,0,4);

// add the controls, second column. None of these can be anonymous

TextField tfMealCost = **new** TextField();

grid.add(tfMealCost,1,0);

TextField tfTipPercent = **new** TextField();

grid.add(tfTipPercent,1,1);

Label lbTip = **new** Label();

grid.add(lbTip,1,2);

Label lbTotal = **new** Label();

grid.add(lbTotal,1,3);

Button btReset = **new** Button("Reset");

grid.add(btReset,1,4);

// reduce textfield widths

tfMealCost.setMaxWidth(60);

tfTipPercent.setMaxWidth(60);

// styling the buttons for fun

btCalc.setStyle("-fx-background-color:tomato;-fx-text-fill:yellow;");

btReset.setStyle("-fx-background-color:tomato;-fx-text-fill:yellow;");

// create and code the lambdas

btCalc.setOnAction(e -> {

// do the math

**double** meal = Double.*parseDouble*(tfMealCost.getText());

**double** tipPct = Double.*parseDouble*(tfTipPercent.getText());

**double** tip = meal \* tipPct / 100;

**double** total = meal + tip;

// complete GUI, but use currency format

// NumberFormat fmt = NumberFormat.getCurrencyInstance();

lbTip.setText(fmt.format(tip));

lbTotal.setText(fmt.format(total));

});

btReset.setOnAction(e -> {

tfMealCost.clear();

tfTipPercent.clear();

lbTip.setText(fmt.format(0));

lbTotal.setText(fmt.format(0));

});

Scene = **new** Scene(grid, 170, 150);

primaryStage.setTitle("My Tip App"); // Set title

primaryStage.setScene(scene); // Place the scene in the stage

primaryStage.show(); // Display the stage

}

**public** **static** **void** main(String[] args) {

*launch*(args);

}

}