Goals & Objectives

The goal of this program is to correctly calculate the total cost of a trip based on the inputs from the user. Users can select from Miles and Kilometers, and get the correct amount in USD.

Lessons Learned

The lessons I learned in this project were 1) getting the UI to interact with another class based on input from the user and 2) calculating the correct costs and converting if one of the units differed from the others. For example, if distance was in Miles, and Gas Price was in Dollars/gallon, but milage was in kms/liter, getting the program to identify, convert, and then calculate the price based on the update. It took me some thought, and a little use of Excel to see the math working so I knew what to expect versus what I was getting in each field.

Also, always include a reset button. The requirements for this project didn’t call for it, but a button to reset all fields without having to close and reopen is a time saver, but a lot.

Functional Requirements

1. Present a User Interface for input in the program.
2. Get the values of each input TextField on button click.
3. Create a TripCost object.
4. Calculate the total cost of the trip.
5. Output in TextField the cost of the proposed trip.

Pseudocode

Import JavaFX Application

Import JavaFX Geometry

Import JavaFX Scene

Import JavaFX Stage

Class Project3 extends Application {

Declare new private TextField tfDistance

Declare new private TextField tfGasolineCost

Declare new private TextField tfGasMileage

Declare new private TextField tfNumberOfDays

Declare new private TextField tfHotelCost

Declare new private TextField tfFoodCost

Declare new private TextField tfAttractions

Declare new private TextField tfTotalTripCost

Declare new private Button btnCalculate

Declare new private String ComboBox cbMiles

Declare new private String ComboBox cbGasCost

Declare new private String ComboBox cbMileage

Method start (Require Stage primaryStage) {

Add String “miles”, “kilometers” to cbMiles

Add String “dollars/gal”, “dollars/liter” to cbGasCost

Add String “miles/gallon”, “kms/liter” to cbMileage

Set btnCalculate Preferred Width to 150px

Declare new GridPane gridPane

Set gridPane Horizontal and Vertical gaps at 5px

Add new Label, tfDistance, and cbMiles to gridPane

Add new Label, tfGasolineCost, and cbGasCost to gridPane

Add new Label, tfGasMileage, and cbMileage to gridPane

Add new Label and tfNumberOfDays to gridPane

Add new Label and tfHotelCost to gridPane

Add new Label and tfFoodCost to gridPane

Add new Label and tfAttractions to gridPane

Add btnCalcualte to gridPane

Add new Label and tfTotalTripCost to gridPane

Set alignment properties for gridPane and all nodes

Set btnCalcualte Action Event to call Method calculateTripCost()

Set new Scene scene with gridPane, 425px width and 300px height

Set primaryStage title as “Trip Cost Estimator”

Add scene to primaryStage

Show primaryStage

End

Method void calculateTripCost {

Declare distance as double parsed from tfDistance

Declare distanceUnit as String parsed from cbMiles

Declare gasCost as double parsed from tfGasolineCost

Declare gasCostUnit as String parsed from cbGasCost

Declare gasMileage as double parsed from tfGasMileage

Declare gasMileageUnit as String parsed from cbMileage

Declare numberOfDays as int parsed from tfNumberOfDays

Declare hotelCost as double parsed from tfHotelCost

Declare foodCost as double parsed from tfFoodCost

Declare attractions as double parsed from tfAttractions

Declare TripCost trip as new TripCost with parsed inputs

Set tfTotalTripCost value as formatted String returned from TripCost method getTotalTripCost

End

End

Class TripCost {

Declare new private double distance

Declare new private String distanceUnit

Declare new private double gasPrice

Declare new private String gasPriceUnit

Declare new private double gasMileage

Declare new private String gasMileageUnit

Declare new private int numberOfDays

Declare new private double hotelCost

Declare new private double foodCost

Declare new private double attractions

Public TripCost {

This( distance = 1, distanceUnit = “miles”, gasPrice = 1.00, gasPriceUnit = “dollars/gal”, gasMileage = 12.5, gasMileageUnit = “miles/gallon”, numberOfDays = 1, hotelCost = 1.00, foodCost = 1.00, attractions = 0.00)

End

Public TripCost(distance as double, distanceUnit as String, gasPrice as double, gasPriceUnit as String, gasMileage as double, gasMileageUnit as String, numberOfDays as int, hotelCost as double, foodCost as double, attractions as double) {

Set this.distance as distance

Set this.distanceUnit as distanceUnit

Set this.gasPrice as gasPrice

Set this.gasPriceUnit as gasPriceUnit

Set this.gasMileage as gasMileage

Set this.gasMileageUnit as gasMileageUnit

Set this.numberOfDays as numberOfDays

Set this.hotelCost as hotelCost

Set this.foodCost as foodCost

Set this.attractions as attractions

End

Public String getDistanceUnit { Return this.distanceUnit

Public String getGasPriceUnit { Return this.gasPriceUnit

Public String getGasMileageUnit { Return this.gasMileageUnit

Public double getDistance { Return this.distance

Public double getGasPrice { Return this.gasPrice

Public double getGasMileage { Return this.gasMileage

Public int getNumberOfDays { Return this.numberOfDays

Public double getHotelCost { Return this.hotelCost

Public double getFoodCost { Return this.foodCost

Public double getAttractions { Return this.attractions

Method getGasolineCost() {

Declare double KILOMETERS\_PER\_HOUR as 1.609347

Dcelare double LITERS\_PER\_GALLON as 3.78541178

Declare double distance as getDistance()

Declare double gasPrice as getGasPRice()

Declare double gasMileage as getGasMileage()

Switch getDistanceUnit

Case “miles”:

If getGasMileageUnit is “kms/liter” {

Set gasMileage as gasMileage / (KILOMETERS\_PER\_HOUR \* LITERS\_PER\_GALLON)

If getGasPRiceUnit is “dollars/liter” {

Set gasPrice as gasPrice \* (1 / LITERS\_PER\_GALLON)

Break

Case “kilometers”:

If getGasMileageUnit is “miles/gallon” {

Set gasMileage as gasMileage \* KILOMETERS\_PER\_HOUR

If getGasPRiceUnit is “dollars/gal” {

Set gasPrice as gasPrice / LITERS\_PER\_GALLON

Break

End

Return (distance / gasMileage) \* gasPrice

End

Method getTotalTripCost {

Return (getGasolineCost() + ((getHotelCost() + getFoodCost() \* getNumberOfDays()) + getAttractions())

End

End

UML Diagram

|  |
| --- |
| **Project3** |
| -tfDistance : TextField |
| -tfGasolineCost : TextField |
| -tfGasMileage : TextField |
| -tfNumberOfDays : TextField |
| -tfHotelCost : TextField |
| -tfFoodCost : TextField |
| -tfAttractions : TextField |
| -tfTotalTripCost : TextField |
| -btnCalculate : Button |
| -cbMiles : ComboBox<String> |
| -cbGasCost : ComboBox<String> |
| -cbMileage : ComboBox<String> |
| +start(primaryStage : Stage)  +calculateTripCost() |

|  |
| --- |
| **TripCost** |
| -distance : double |
| -distanceUnit : String |
| -gasPrice : double |
| -gasPriceUnit : String |
| -gasMileage : double |
| -gasMileageUnit : String |
| -numberOfDays : int |
| -hotelCost : double |
| -foodCost : double |
| -attractions : double |
| +TripCost() |
| +TripCost(distance : double, distanceUnit : String, gasPrice : double, gasPriceUnit : String, gasMileage: double, gasMileageUnit: String, numberOfDays: int, hotelCost: double, foodCost: double, attractions: double) |
| +getDistanceUnit() : String |
| +getGasPriceUnit() : String |
| +getGasMileageUnit() : String |
| +getDistance() : double |
| +getGasPrice() : double |
| +getGasMileage(): double |
| +getNumberOfDays() : int |
| +getHotelCost() : double |
| +getFoodCost() : double |
| +getAttractions(): double |
| +getGasolineCost(): double |
| +getTotalTripCost(): double |

Test Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input/Output** | **Expected Result** | **Actual Result** | **Outcome (Pass/Fail)** |
| 1a | Present a User Interface for input in the program. | GUI launches with empty textfields | GUI launches with empty textfields | Pass |
| 2a | Get the values of each input TextField on button click. | TextField.getText() gets string of each textfield value | Program assigns value to each instance variable from TextField.getText() method. | Pass |
| 3a | Create a TripCost object. | A TripCost object is created with values from user | Method calculateTripCost creates a TripCost object with all required inputs | Pass |
| 4a | Calculate the total cost of the trip. | TripCost.getTotalTripCost() calculates trip cost based on inputs | TripCost.getTotalTripCost() calculates cost from stored inputs and returns double to program | Pass |
| 5a | Output in TextField the cost of the proposed trip. | Program updates TextField for the estimated trip cost | TextField tfTotalTripCost is updated with double from TripCost.getTotalTripCost() | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input/Output** | **Expected Result** | **Actual Result** | **Outcome (Pass/Fail)** |
| 1 | Distance: 1000 mi  Gasoline Cost: 3.95/gal  Gas Mileage: 22.3 mpg  Number of Days: 3  Hotel Cost: 150  Food Cost: 75  Attractions: 100 | $952.13 | $952.13 | Pass |
| 2 | Distance: 1200 mi  Gasoline Cost: 4.25/gal  Gas Mileage: 18.45 mpg  Number of Days: 5  Hotel Cost: 205  Food Cost: 100  Attractions: 200 | $2,001.42 | $2,001.42 | Pass |
| 3 | Distance: 1600 km  Gasoline Cost: 1.95/liter  Gas Mileage: 20 mpg  Number of Days: 6  Hotel Cost: 85  Food Cost: 110  Attractions: 65 | $1,331.93 | $1,331.93 | Pass |
| 4 | Distance: 2500 km  Gasoline Cost: 2.12/liter  Gas Mileage: 12.5 km/l  Number of Days: 6  Hotel Cost: 115  Food Cost: 80  Attractions: 200 | $1,794.00 | $1,794.00 | Pass |

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated