Implementation Manual of the Money Converter App:

This is a JavaFX application that converts currencies between USD, JPY, AUD, EUR, CAD, and GBP. It includes a graphical user interface (GUI) that allows users to select the input currency, output currency, and input amount. The application will then convert the input amount into the selected output currency and display the result.

## System Requirements

* Java Development Kit (JDK) 11 or higher
* JavaFX library
* A compatible operating system (Windows, Linux, or macOS)

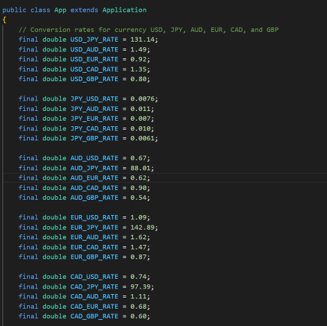
## Usage

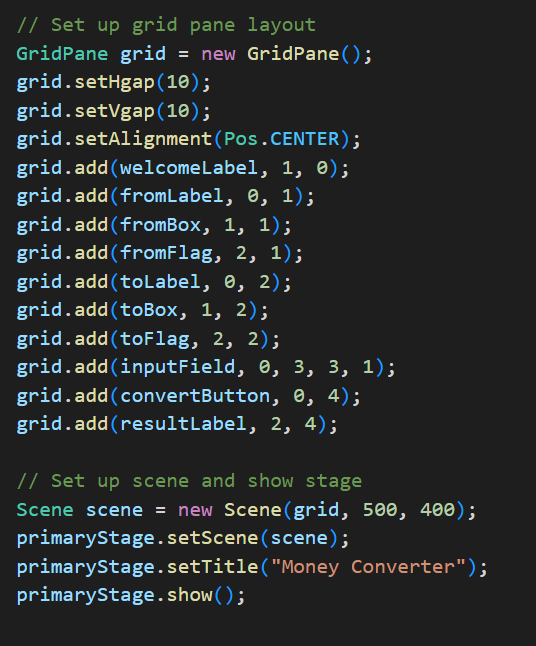
1. Launch the application by running the App class as described above. A window should appear with the title "Money Converter".
2. Use the two drop-down menus labeled "From" and "To" to select the input and output currencies, respectively. The options are "US Dollar", "Japanese Yen", "Australian Dollar", "Euro", "Canadian Dollar", and "British Pound".
3. Enter the amount of money to be converted in the text field labeled "Enter amount" using the keyboard. The input must be a positive number.
4. Click the "Convert" button to convert the input amount to the selected output currency. The result will be displayed in the label on the right side of the window labeled "Result".
5. To perform another conversion, simply repeat steps 2-4 with the new input and output currencies.

## Technical Details

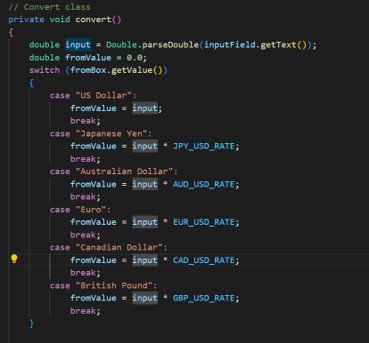
The application is written in Java and uses the JavaFX library for the GUI components.

The conversion rates are stored in final double variables that are used later in the program to perform currency conversions.



The GUI is constructed using a GridePane layout with various Label, ChoiceBox, TextField, ImageView, and Button components. The GUI is launched by calling the start method of the App class, which creates a new Stage object and sets its scene to the GridPane created by the constructor. 

The conversion calculation is performed by the convert method, which reads the input amount from the TextField, converts it to USD using the appropriate conversion rate, and then converts it to the output currency using the appropriate conversion rate. The result is then displayed in the Label component. The convert method initializes a variable convertedAmount to zero, which will store the converted amount. It uses a switch statement to determine the appropriate conversion rate to use based on the fromCurrency and toCurrency values. It calls the getConversionRate method, passing in a string representing the conversion rate to use. It multiplies the inputAmount by the conversion rate to get the convertedAmount. It updates the resultLabel with the converted amount, using the String.format method to format the text.

The flag images are retrieved by the getFlagUrl method. The flag images are from the [flagcdn.com](https://flagcdn.com) website using URLs constructed from the selected input and output currencies. It uses a switch statement to determine the appropriate flag for each currency type.

