TIRANA METROPOLITAN UNIVERSITY

FACULTY OF COMPUTER SCIENCE AND IT

Project in Object Oriented Programing

Title: Currency Converter

Student:

Kleart Adri

Mentor:

Msc. Evis Plaku

Msc.Megi Tartari

BSc, Software Engineering

Tirana, January 2024

I. Abstract

Currency converter is a simply project like a calculator-app developed using Java features.Different countries use different currency, and there is daily variation in these currencies relative to one another. Those who transfer money from one country to another must be updated with the latest currency exchange rates in the market.
Any user can use such an application, but it is especially helpful in business, share, and finance-related areas where daily currency exchange and money transfers occur.

In this currency converter app, users are provided with two option to select the type of currency they have and the second is the currency that they want to convert to. The program allows users to enter amount to be converted,and thenit is display the converted amount.The program also include the “exit” button to quit from the program and also the “reset” button for putting the values in the beginning.

II. Key Functionalities

Topics that are included for building this project are:

1. ***Object-oriented programming paradigm***: The code uses classes (MyCurrencyConverter), objects (like jPanel1, jLabel1, etc.), and encapsulation (data members are declared private like currencyUnits and exchangeRates).
2. ***Exception Handling:*** It catches a NumberFormatException that might occur when attempting to convert the text from the t1 field to a double using Double.parseDouble(t1.getText()). If the conversion fails because the text in t1 is not a valid number, a NumberFormatException is thrown, and the catch block is executed. In the catch block, a JOptionPane is displayed to inform the user to enter a valid number in the amount field.
3. ***Generic Collections:*** The code does not use generic collections like lists, sets, maps, queues, or stacks. However, it does use arrays to store currency units and exchange rates.
4. ***GUI and event-driven programming***: The code uses Swing components (JFrame, JPanel, JLabel, JComboBox, etc.) to create a graphical user interface. It also includes event-driven programming elements, as seen in the action listeners for the combo boxes (firstCountry and secondCountry).

   (not included topics in this project are:files and streams,and also the usage of generic collections like lists, sets, maps, queues, or stacks,also for this project it isn’t included JavaFX for GUI but JavaSwing)
   The project it is worked in Netbeans IDE 8.2 version.

III. Technical Part

- Package and Imports: The code is part of the mycurrencyconverter.project package and imports classes from the javax.swing package for GUI creation.
- Class Declaration: The MyCurrencyConverter class extends javax.swing.JFrame, making it a GUI window.
- Currency Data: The currencyUnits array stores currency symbols, and the exchangeRates 2D array stores conversion rates between currencies.
- Constructor: The MyCurrencyConverter() constructor initializes the GUI components by calling the initComponents() method.
- GUI Components: The GUI consists of various components:
  Labels: jLabel1, jLabel2, jLabel3 display static text on the GUI.
  Combo Boxes: firstCountry and secondCountry allow the user to select currencies.
  Text Fields: t1 and t2 are for inputting the amount to convert and displaying the result.
  Buttons: convertbutton, resetbutton, exit are for triggering conversion, resetting fields, and closing the application.
- Event Handlers: The itemStateChanged methods handle user interactions with the combo boxes. When the selected item in a combo box changes, the corresponding currency symbol is displayed next to the combo box.
- Conversion Logic: The convertbutton's actionPerformed method contains the logic for converting currencies. It first checks if the user has selected both currencies and entered an amount. If not, it shows an error message. Then, it tries to parse the amount entered by the user and calculate the converted amount using the exchange rates. If the parsing fails (i.e., the user didn't enter a valid number), it shows an error message
- Main Method: The main method sets the look and feel of the GUI to Nimbus (if available), then creates and displays an instance of MyCurrencyConverter.

IV. Summary

The currency converter application with a graphical user interface (GUI), uses the Swing library for the GUI. The user can select two currencies and input an amount to convert. The application then calculates and displays the converted amount. The conversion rates are stored in a 2D array. The application also handles invalid inputs and displays appropriate error messages. The GUI consists of labels, combo boxes, text fields, and buttons. The main method creates and displays the application window.

Next Steps:

Currency Updating:Implementing a mechanism to dynamically update exchange rates from an external source, such as an API, to ensure accuracy and reflect real-time currency values not putting them each time.

V. Screenshots