

Student-Budgeting-Tool

CS2020 - Budgeting Tool Coursework 2

About the project

In this Assignment, we were tasked with creating a simple Budget Tool for student which displays a GUI with various inputs such as Wages, Loans, Bills and Rent etc. These values are entered by the user (us) through JTextFields which is then calculated and presented back to the user with the Total Income when the calculate button is pressed.

Built with

This project was built in VScode using Java (JSwing) and the repository was cloned from GitHub for learning how to use git & version control. (Although using git wasn't necessary for this project I still used it to try and get used to using it.)

Getting started

Pre-requisites

1. The Program takes 3 or more income and spending fields and the system calculates the total income when the calculate button is pressed. The basic system:
 - a. Income fields: Wages, Loans, Cash Back
 - b. Expenses fields: Bills, Rent, Food Shop
 - c. Input validation: gives an error message if something other than numbers is entered. Also displays the surplus/ deficit to user using red for deficit and black for surplus
2. The Extensions of the basic system:
 - a. The time period that can be specified by the user using the JComboBox. The user can adjust their income/expenses field per week, month or year.
 - b. "Spreadsheet" like behaviour which updates the totals without the use of the calculate button
3. The Undo Buttons: -Single level undo button so that the user can undo their most recent action and saves the state of the numbers when a change is made

- a. Multi level undo which can undo multiple actions using a stack

Set Up

To set up the project I cloned the git repository to VSCode and opened it there. Then go to the src\main\java\budget and right-click on BudgetCalculator.java which gives us the option to Run Java which runs the code.

Usage

Once the program is running a graphical user interface is shown to the user with the income and expenses fields. The user is then prompted to enter their numerical values and the total income is calculated based on the fields: Wages, Loans, Cash Back and Bills, Rent, Food Shop. Where the former are added together and the latter are taken away. The program updates when the user changes any one of these fields or their time period. The user can also undo their previous changes using the undo buttons.

Classes Explanation

The two classes I coded in were the BudgetCalculator class (The main class where the code runs) and the MultiLevelUndo class.

- The MultiLevelUndo class saves and restores the state of the input fields and it does this through managing a stack of previous states and does this both for single level and multilevel undos. The current state is saved when a change is made and the user can undo using the buttons
- The BudgetCalculator class is the class that initialises the program and holds the input fields for the income and expenses where the user can interact. It uses the MultiLevelUndo class to handle the saving and restoring state and calls all the methods by creating a state manager which is an object of the class. Then the calculateTotalIncome() method is connected to the calculate button and used to calculate the income after each change.

JUnit Test

The tests make sure that both undo buttons and calculations do what they are required. The BCTest.java is where I created more tests for the computer to see if it would do what I

expected with some example data. This includes testing if the computer would undo the data inputted via the single level or multi level undo button depending which one is pressed. The only downside is that it has to be pressed more than once to work but in the tests it completes them fine

Road Map i.e plan to improve for next time

Improvements for next time:

- Designing the user interface to look appealing for the user
- Adapting the data by predicting spending based on previous entries
- More JUnit tests that make sure nothing goes uncovered

Name

Adefolami Olawuyi Student ID: 52317231

Resources Used

<https://stackoverflow.com/questions/5359955/simple-focus-listener-in-java>

https://docs.oracle.com/javase/tutorial/uiswing/events/focuslistener.html#:~:text=The%20Focus%20Listener%20API&text=Returns%20the%20true%20value%20if,focus%2Dgained%20event%20is%20temporary.&text=Returns%20the%20component%20that%20fired%20the%20focus%20event.&text=Returns%20the%20other%20component%20involved,component%20that%20lost%20the%20focus.*/

<https://examples.javacodegeeks.com/java-development/desktop-java/swing/jlabel/set-foreground-color-in-jlabel/>

<https://www.tutorialspoint.com/how-can-we-set-the-foreground-and-background-color-to-jcombobox-items-in-java#:~:text=A%20JComboBox%20can%20generate%20an,methods%20of%20a%20JComboBox%20class.>

<https://www.youtube.com/watch?v=3XkmZfBDfek>

for read me file: <https://github.com/othneildrew/Best-README-Template>

