IPO charts

1. Orchestration Layer

a. **Module Name: Service Broker**

Module descriptions: The service broker's purpose is to help manage the workflow and coordinate activities by choosing which program to run.

Input:

- ServiceCode
- List of program referenced by the ServiceCode

Output:

- Output from program referenced by the ServiceCode

Service Broker Pseudocode:

1. Initialize an array of command line arguments from the user input
2. Determine the language to use for error messages based on the last argument of the input
3. If no language is specified or the specified language is not supported, default to English
4. Extract the name of the service to run from the first argument of the input
5. If the service is "Tax", run the "CalcFederalTaxes.py" module with the input parameters
6. If the service is "IRA", run the "CalcMonthlyPayoutIRA.py" module with the input parameters
7. If the service is "Payroll", run the "payrollpaychecks.py" module with the input parameters
8. If an error occurs during the module execution, output an error message in the specified language
9. If the module executes successfully, output the result of the module in the console

2. Business Layer

a. **Module Name: Federal Tax Services**

Module descriptions: The Federal Tax Services allows the user to calculate the amount of federal taxes that are due based upon gross annual income, the year they are filing, and their filing status. It will then return the amount of federal taxes due based upon these details.

Input:

- Tax year filing taxes for
- Status - (joint, single, head of household)
- Gross income

Output:

- Tax owed

Pseudocode:

1. If amount of command line arguments is not correct error
2. Else continue program
3. Open user input file
4. Parse through file and store user input
    a. User input is in this format
        i. Year, filing status first letter, gross annual income
        ii. Ex. 2020,S,100000
    b. Parse through based on “,”
5. Combine the year and filing status in a string and add .txt
6. Access .txt file that applies to customer
7. If gross annual income is less than number on left of comma apply percentage on right of comma in the .txt and add with previous sum
    a. Once the if is reached the program has reached the last tax bracket and breaks out of the loop
8. Else subtract the previous tax bracket amount from the current tax bracket amount
    a. First run through for loop previous amount is zero
9. Add that amount to the running total for federal tax
10. Subtract gross annual income from the amount for that tax bracket and store as leftover taxable income
11. Store the current item as the previous item for next loop
12. Return this final amount of taxes owed to the main function
13. Print the final amount of taxes to the command line

b. Module Name: Payroll Paycheck

Module descriptions: The payroll paycheck module allows users to input their gross annual salary and the amount of pay periods they will have. This will allow them to find out how much they will receive at each paycheck.

Input:

- Gross annual income
- Number of pay periods

Output:

- FICA, medicaid, State, Federal, and total tax

Pseudocode:

1. Accept a filename as input from the user.
2. Try to open the file with the given filename.
3. If the file cannot be opened, print an error message and exit the program.
4. Read the contents of the file and store each line as an element in a list.
5. If the list contains fewer than two elements, print an error message and exit the program.
6. If the list contains more than two elements, print an error message and exit the program.
7. Convert the first element in the list to an integer and store it as the gross annual income.
8. Convert the second element in the list to an integer and store it as the number of pay periods.
9. Try to open the "payrolltax.txt" file.
10. If the file cannot be opened, print an error message and exit the program.
11. Parse the file to get the tax rates and store them in a dictionary.
12. Calculate the FICA, Medicaid, state, federal, and total taxes based on the input and the tax rates.
13. Print out the calculated taxes.

c. Module Name: IRA Services

Module descriptions: The IRA Service allows for the gross amount in the IRA, the payout age, and the age for the last payout to be input. From this data the amount for the monthly IRA payout will be calculated.

Input:

- Amount in IRA
- Age for payments to start
- Age for payments to end

Output:

- Payout per month

Pseudocode:

1. If amount of command line arguments is not correct error
2. Else continue program
3. Open user input file
4. Parse through file and store user input
    a. user input is in this format
        i. Gross amount in IRA, age to start payout, age for last payment
        ii. Ex. 800000,65,89
    b. Parse through based on “,“
5. Subtract age to start payout from age for last payout
6. Multiply by 12 to find the overall amount of months
7. Divide gross amount in ira by the number of overall months
8. Return this to main function

9. Print the monthly payout amount to the command line

3. Utility Layer

a. Module Name: Messages

Module descriptions: The Messages module prints out the messages that have been sent to the module.

Input:

- Message code
- Language to translate the message

Output:

- Error message in the language specified in input parameter

Pseudocode:

1. Set current_dir variable to the current working directory
2. Define readJson function that takes in a language parameter
3. Change directory to the utility folder containing message codes file
4. Open the JSON file for the specified language and read its contents
5. Change directory back to the original directory
6. Return the JSON data
7. Define errorMessageDatas function that takes in a language parameter
8. Check if the language is Spanish or German, and if so, return the JSON data for that language
9. If the language is not Spanish or German, set the language to English and return the JSON data for English
10. Define getServiceBrokerErrorMessage function that takes in a service code and a language parameter
11. Retrieve the JSON data for the specified language
12. If the language is Spanish or German, return the error message for the specified service code in that language
13. If the language is not Spanish or German, return the error message for the specified service code in English
14. Define main function
15. Check if there are either less than 2 or more than 3 command line arguments and print "401" if so
16. If there are 2 command line arguments, print the error message for the specified service code in English
17. If there are 3 command line arguments, print the error message for the specified service code in the specified language, if it is valid
18. Call the main function if the script is being run as the main program