

User Documentation

1. How to build and run the project:

Inside the main project directory run the following command:

mvn clean compile exec:java@money-manager

This builds the project and immediately executes it in the command line.

2. Interaction with the program:

After running the project the main event loop starts. The program will ask “What do you want to do?”. User has a few options (case insensitive) and can only interact with the following commands:

help - provides information about the commands

end - terminates the program

add account - adds account with chosen parameters

remove account - removes account of the chosen name

add transaction - adds transaction of chosen parameters to chosen account

remove transaction - removes transaction with chosen Id from chosen account

add transfer – adds transfer between two accounts

display accounts - displays account/s in chosen way

save - saves current data

load - loads previously saved data

3. Detailed description of each command:

a) help:

Provides description of the commands.

b) end:

immediately terminates the program.

c) add account:

Program will ask the user to provide the name of the account which has to be unique. In case it's not program will return to the “What do you want to do?” stage. It will then ask for the currency of the

account and its initial balance (which has to be an Integer). After all the information is provided the account is created.

d) remove account:

Program will ask the user to provide the name of the account the user wants to remove. If there is an account with this name it will be removed along with all the transactions associated with it.

f) add transaction:

Program will ask the user to provide the name of the account to which the transaction is meant to be added. If such an account exists, user will be asked to provide title of the transaction, category, date in yyyy-mm-dd format, and value of the transaction (which has to be an integer). When all the data is provided correctly the transaction is added to the chosen account.

g) remove transaction:

Program will ask the user to provide the name of the account from which the transaction is meant to be removed. If such an account exists, user will be asked to provide id of the transaction (which can be seen when displaying transactions with the "display accounts" command). After that all of the transactions with the chosen id will be removed from the chosen account.

h) add transfer:

Program will ask the user to provide the name of the two accounts between which the user wishes to create a transfer. If such two accounts exist, user will be asked to provide title of the transfer, category, date in yyyy-mm-dd format, and value of the transaction (which has to be an integer). If the currencies of the two accounts are different the program will ask the user to provide an exchange rate between the two. After that, if all the data was provided correctly the transfer is added as two corresponding transactions, each to one of the chosen accounts.

i) display accounts:

Program will ask the user to choose whether to display all accounts or just one. In case the "all" option is chosen ("all", "all accounts", "all of them" are accepted) all accounts and their transactions are displayed. If "one" option is chosen ("one", "one account", "one of them" are accepted) user will be asked to provide the name of an account they want to display. If the chosen account exists user will be then asked to choose whether the transactions are meant to be displayed normally ("basic", "basic display" are accepted) or by category ("by category", "category", "category display" are accepted). Afterward the account/s are displayed in a chosen way.

j) save:

saves the currently stored data in "Money_Manager" file.

k) load:

loads the saved data from "Money_Manager" file. In case the file is unavailable or data is corrupted all changes are reverted.