Steven Guikal August 15, 2019

**Final Project – Part B - Journal**

7:46 AM - Project Start

Created Main class with main method.

Created Transaction class with properties name, category, amount, and date.

7:50 AM - Menu and Printing Functions

Added the getChoice, printMenu, and padText functions from the Java Vision assignment.

8:01 AM - Menu Printing

Almost all menu printing options have been added.

Only exceptions are dynamic ones for dates.

8:37 AM - Adding Expenses

A user can create expenses and choose a category from a list of categories.

The array that stores transactions is re-initialized with a new size to

allow for dynamic sizes without having to store an extra length variable

or using excess memory.

A getCategory function is implemented which displays all categories and

asks the user to choose one.

9:12 AM - Modify Categories Menu option

Added a listCategories function which lists categories.

Modified getCategory function to use listCategories.

Adding, removing, modifying, and viewing categories is now possible.

Error messages are provided if a category does not exist.

9:24 AM - Add Income + Helper Function + Food Break

Created getAmount and getDate helper function to get the amount and date of a transaction.

These are currently used in the addExpense and addIncome functions.

Adding income is now an option that works and adds an income transaction.

11:37 AM - Searching

Create search method.

Change transactions to store LocalDate instead of Date to be able to get

months and days using built-in methods. This also makes date comparisons

significantly easier.

A helper function, getDateFromString has been temporarily added to allow

for easier testing.

All searching is currently done sequentially

12:00 - Saving & Loading + Lunch

Added a getDateString method to Transaction class to get the date as a

saveable string. Transactions and Categories are now successfully saved

and loaded.

6:53 PM - Expense Plan

Added the plan array to store planned costs per category.

Updated saving and loading methods to save and load the plan.

Modified category removal to remove the category's amount from the plan.

Added automatic loading (or saving) of files on program start, along with

messages.

8:35 PM - Goals and Saving

Adding goals is now possible with their own class and array.

Saving and loading have been updated to include goals.

Saving and loading functions are in their final form.

9:39 PM - Goals

Goal can now be modified, completed or removed.

No check is currently implemented to prevent completing goals

when there are not enough savings.

9:48 PM - Sorting

Transactions are now sorted by date before saving.

NEXY DAY

1:24 PM - Starting on Output

The first output screen (Current Month) has been completed.

Methods for the other outputs have been started but not finished.

A main view method has been created to call the individual outputs

2:01 PM - Month Output + Break

Past month output has been finished.

Current and Past month output have been combined into one function.

A DecimalFormat for currency has been added to unify all output.

Converting LocalDate to string has all been moved to the Transaction class.

6:41 PM - Total Output

Output for totals has been completed.

9:04 PM - Goals and Plan output

Goals and plan output has been added.

Fixed a bug not showing all months during month selection.

Fixed a bug not showing all months during totals.

Plan is updated when a new category is added.

9:20 PM - Savings check + FINAL EDIT

Added a check to make sure you don't accidentally complete goals you

don't have the balance for.

[-] While I stated that searching for a date would be done with

a binary search, I didn't do this. It would make the could

less clean and doesn't increase efficiency drastically.

No visible effect was modified.

[-] I also decided to go against a split screen for the Plan

and Goals page because either one might be omitted if no

goals or plans exist and they both already take up a lot of

width. Working with text is also quite difficult horizontally

since you can only print top to bottom (without array trickery).

[=] Time Spent: ~12 hours