TransactionManager

Tests and documentatation

1. Overview of the project

A class for processing data from a file. Class provides methods for reading data from a file, modifying transaction records, and committing changes back to the file.

a. Attributes:

file : str

name of a file containing transaction data

b. Methods:

read_data(file: str)

Read data from the given file.

display_header()

Display the header of the file.

display_transactions (index: int = 0)

Display the transactions from the file.

add_transaction()

Add a new record to the data.

change_transaction(index: int)

Change a transaction record in the data.

change_header(index: int)

Change a header of a file.

validate_new_header_val(val, max_len, min_len = 0)

Validate user input for the header of a file.

take_and_validate_amount() -> str

Take and validate the amount for a transaction.

take_and_validate_currency() -> str

Take and validate the currency for a transaction.

commit_changes()

Commit the changes to the file.

format_counter(column: str) -> str

Format a counter column by aligning it to the right.

format_amount(column: str) -> str

Format an amount column by aligning it to the right and inserting a decimal point.

user_interface()

Launch a command-line interface for interacting with the data manager.

interface_of_show()

Launch a command-line interface for displaying transactions and header information.

interface_of_change()

Launch a command-line interface for changing transactions and header information.

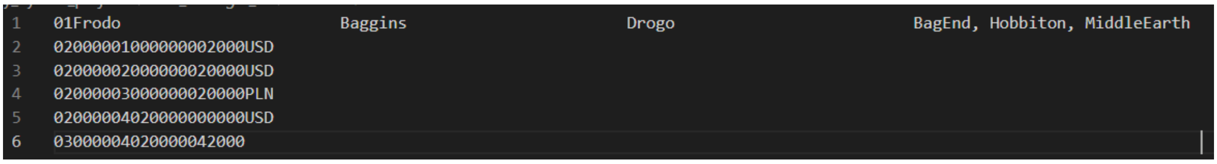
interface_for_close_field()

Launch a command-line interface for closing fields for changes.

2. Tests

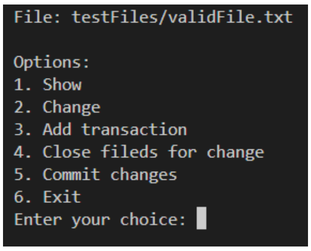
2.1 Valid file:

For a correctly structured file like one given below, all functions are working correctly.



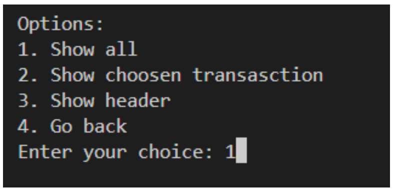
```
01Frodo                       Baggins                       Drogo                         BagEnd, Hobbiton, MiddleEarth
02000001000000002000USD
02000002000000020000USD
02000003000000020000PLN
02000004020000000000USD
03000004020000042000
```

2.1.1 Loading a file

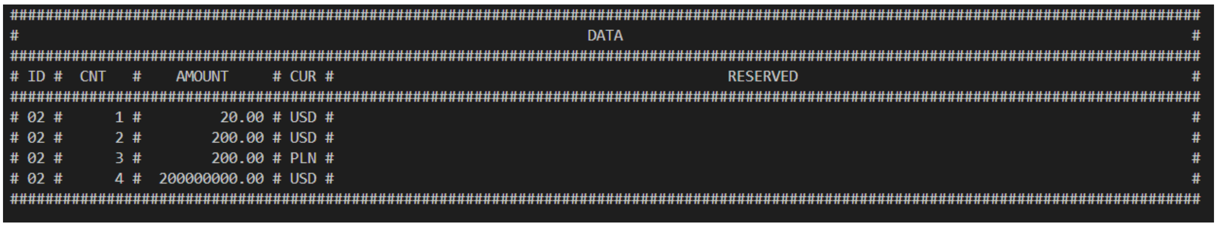


```
File: testFiles/validFile.txt

Options:
1. Show
2. Change
3. Add transaction
4. Close fileds for change
5. Commit changes
6. Exit
Enter your choice: 
```

2.1.2 Show all data:

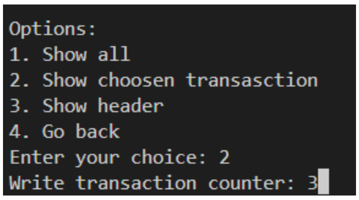


```
Options:
1. Show all
2. Show choosen transasction
3. Show header
4. Go back
Enter your choice: 1
```

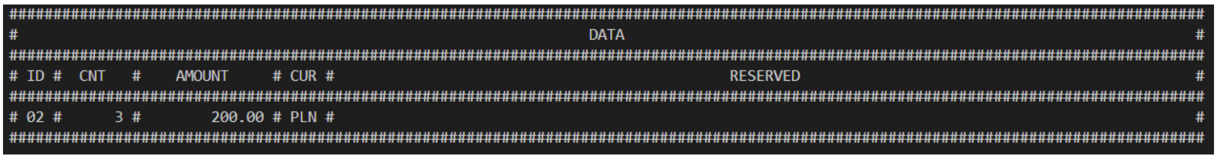


```
########################################################################################################################################################
#                                                                         DATA                                                                         #
########################################################################################################################################################
# ID #  CNT   #    AMOUNT     # CUR #                                                       RESERVED                                                   #
########################################################################################################################################################
# 02 #      1 #         20.00 # USD #                                                                                                                  #
# 02 #      2 #        200.00 # USD #                                                                                                                  #
# 02 #      3 #        200.00 # PLN #                                                                                                                  #
# 02 #      4 #  200000000.00 # USD #                                                                                                                  #
########################################################################################################################################################
```

2.1.2 Show data of given index

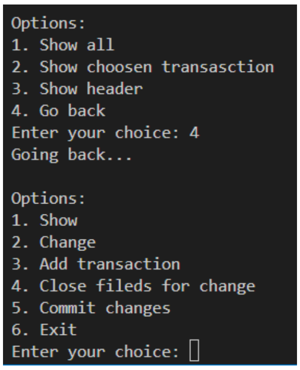


```
Options:
1. Show all
2. Show choosen transasction
3. Show header
4. Go back
Enter your choice: 2
Write transaction counter: 3
```



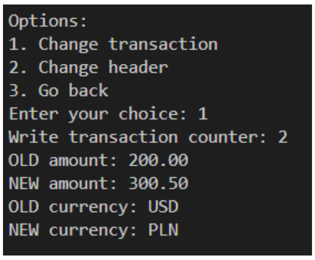
```
########################################################################################################################################################
#                                                                         DATA                                                                         #
########################################################################################################################################################
# ID #  CNT   #    AMOUNT     # CUR #                                                       RESERVED                                                   #
########################################################################################################################################################
# 02 #      3 #        200.00 # PLN #                                                                                                                  #
########################################################################################################################################################
```

2.1.3 Going back



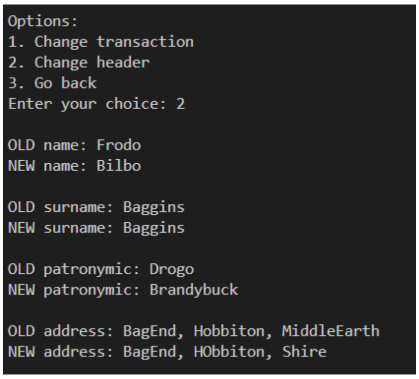
```
Options:
1. Show all
2. Show choosen transasction
3. Show header
4. Go back
Enter your choice: 4
Going back...

Options:
1. Show
2. Change
3. Add transaction
4. Close fileds for change
5. Commit changes
6. Exit
Enter your choice: 
```

2.1.4 Changing a transaction by index



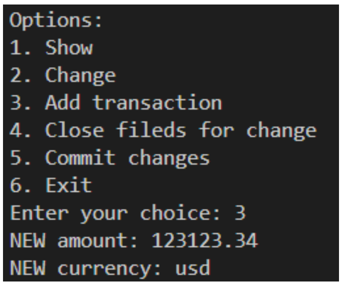
```
Options:
1. Change transaction
2. Change header
3. Go back
Enter your choice: 1
Write transaction counter: 2
OLD amount: 200.00
NEW amount: 300.50
OLD currency: USD
NEW currency: PLN
```

2.1.4 Changing a header



```
Options:
1. Change transaction
2. Change header
3. Go back
Enter your choice: 2

OLD name: Frodo
NEW name: Bilbo

OLD surname: Baggins
NEW surname: Baggins

OLD patronymic: Drogo
NEW patronymic: Brandybuck

OLD address: BagEnd, Hobbiton, MiddleEarth
NEW address: BagEnd, HObbiton, Shire
```

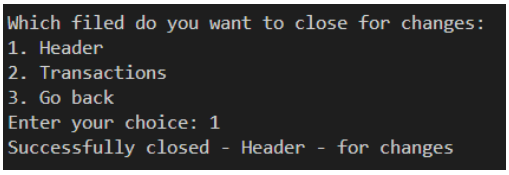
2.1.5 Adding New record



```
Options:
1. Show
2. Change
3. Add transaction
4. Close fileds for change
5. Commit changes
6. Exit
Enter your choice: 3
NEW amount: 123123.34
NEW currency: usd
```

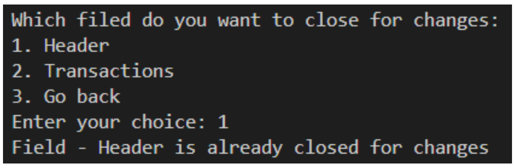
2.1.6 Closing fields for changes

a) Closing header:



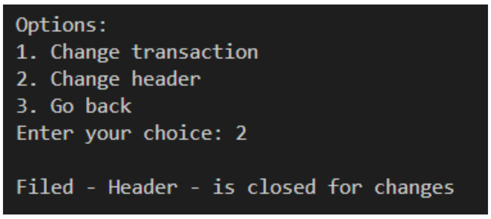
```
Which filed do you want to close for changes:
1. Header
2. Transactions
3. Go back
Enter your choice: 1
Successfully closed - Header - for changes
```

If user wants to close already closed filed message will be shown



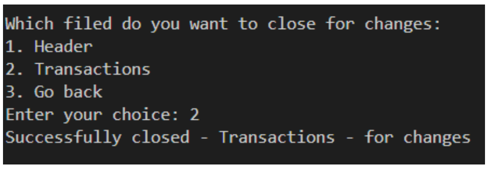
```
Which filed do you want to close for changes:
1. Header
2. Transactions
3. Go back
Enter your choice: 1
Field - Header is already closed for changes
```

After this operation you can’t change header field in *change* option



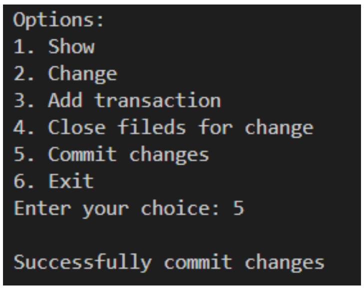
```
Options:
1. Change transaction
2. Change header
3. Go back
Enter your choice: 2

Filed - Header - is closed for changes
```

b) Closing transactions



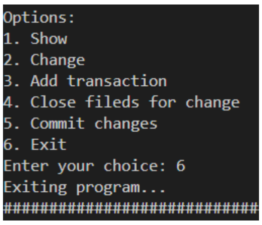
```
Which filed do you want to close for changes:
1. Header
2. Transactions
3. Go back
Enter your choice: 2
Successfully closed - Transactions - for changes
```

2.1.6 Commit changes



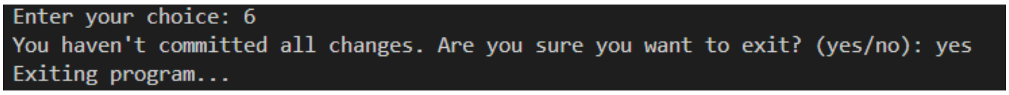
```
Options:
1. Show
2. Change
3. Add transaction
4. Close fileds for change
5. Commit changes
6. Exit
Enter your choice: 5

Successfully commit changes
```

2.1.7 Exit program



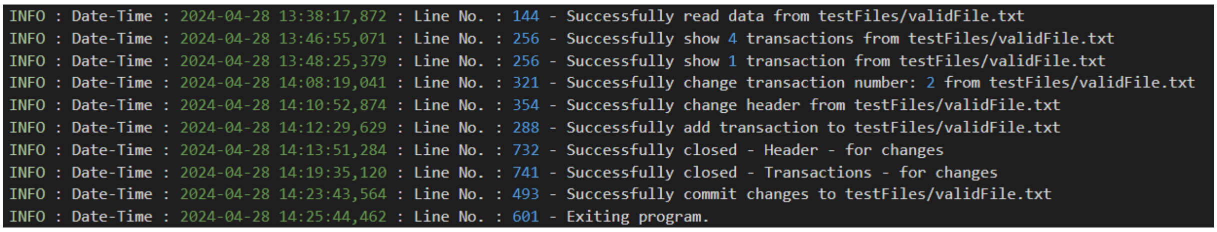
```
Options:
1. Show
2. Change
3. Add transaction
4. Close fileds for change
5. Commit changes
6. Exit
Enter your choice: 6
Exiting program...
############################
```

If changes haven’t been committed program will ask if user is sure he wants to exit



```
Enter your choice: 6
You haven't committed all changes. Are you sure you want to exit? (yes/no): yes
Exiting program...
```

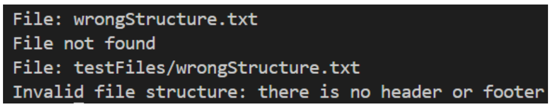
2.1.8 Log file after all operations



```
INFO : Date-Time : 2024-04-28 13:38:17,872 : Line No. : 144 - Successfully read data from testFiles/validFile.txt
INFO : Date-Time : 2024-04-28 13:46:55,071 : Line No. : 256 - Successfully show 4 transactions from testFiles/validFile.txt
INFO : Date-Time : 2024-04-28 13:48:25,379 : Line No. : 256 - Successfully show 1 transaction from testFiles/validFile.txt
INFO : Date-Time : 2024-04-28 14:08:19,041 : Line No. : 321 - Successfully change transaction number: 2 from testFiles/validFile.txt
INFO : Date-Time : 2024-04-28 14:10:52,874 : Line No. : 354 - Successfully change header from testFiles/validFile.txt
INFO : Date-Time : 2024-04-28 14:12:29,629 : Line No. : 288 - Successfully add transaction to testFiles/validFile.txt
INFO : Date-Time : 2024-04-28 14:13:51,284 : Line No. : 732 - Successfully closed - Header - for changes
INFO : Date-Time : 2024-04-28 14:19:35,120 : Line No. : 741 - Successfully closed - Transactions - for changes
INFO : Date-Time : 2024-04-28 14:23:43,564 : Line No. : 493 - Successfully commit changes to testFiles/validFile.txt
INFO : Date-Time : 2024-04-28 14:25:44,462 : Line No. : 601 - Exiting program.
```

2.2 Dealing with errors and invalid inputs

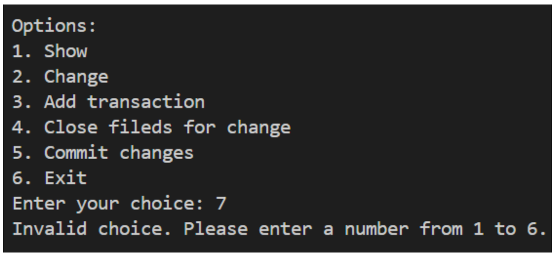
2.2.1 Check if file exists and if structure is correct



```
File: wrongStructure.txt
File not found
File: testFiles/wrongStructure.txt
Invalid file structure: there is no header or footer
```

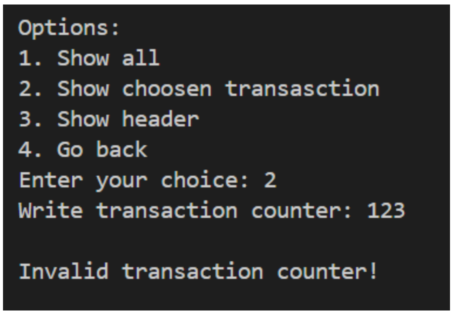
2.2.2 Invalid Input test

Invalid input during choosing options:



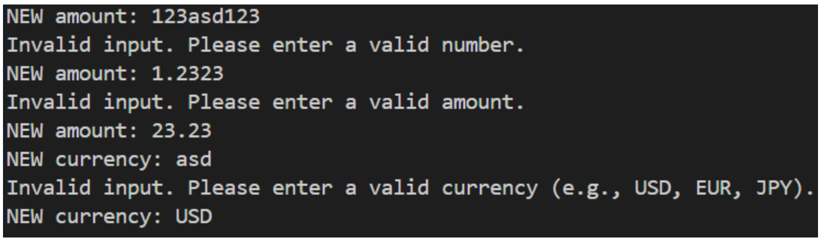
```
Options:
1. Show
2. Change
3. Add transaction
4. Close fileds for change
5. Commit changes
6. Exit
Enter your choice: 7
Invalid choice. Please enter a number from 1 to 6.
```

Invalid counter warning:



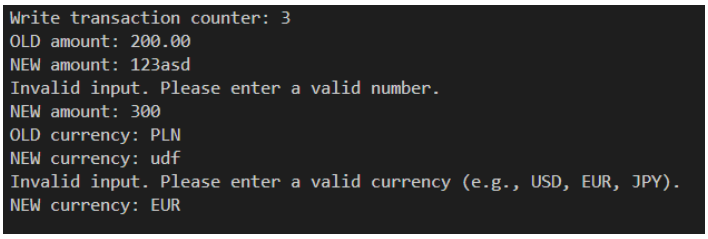
```
Options:
1. Show all
2. Show choosen transasction
3. Show header
4. Go back
Enter your choice: 2
Write transaction counter: 123

Invalid transaction counter!
```

Invalid new transaction information:



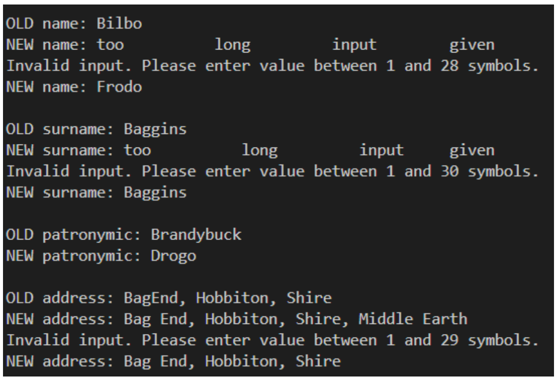
```
NEW amount: 123asd123
Invalid input. Please enter a valid number.
NEW amount: 1.2323
Invalid input. Please enter a valid amount.
NEW amount: 23.23
NEW currency: asd
Invalid input. Please enter a valid currency (e.g., USD, EUR, JPY).
NEW currency: USD
```

Invalid information during changing transaction:



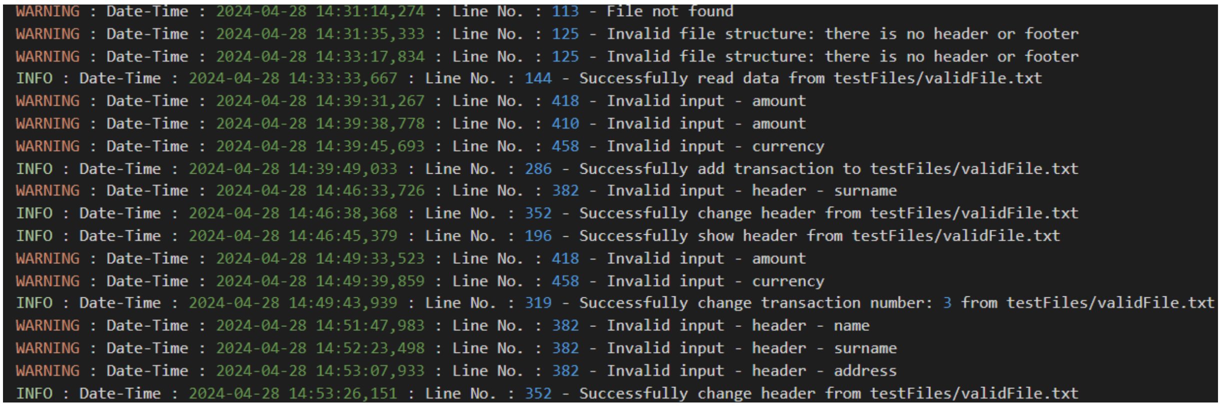
```
Write transaction counter: 3
OLD amount: 200.00
NEW amount: 123asd
Invalid input. Please enter a valid number.
NEW amount: 300
OLD currency: PLN
NEW currency: udf
Invalid input. Please enter a valid currency (e.g., USD, EUR, JPY).
NEW currency: EUR
```

Invalid information during changing header:



```
OLD name: Bilbo
NEW name: too          long         input        given
Invalid input. Please enter value between 1 and 28 symbols.
NEW name: Frodo

OLD surname: Baggins
NEW surname: too           long         input     given
Invalid input. Please enter value between 1 and 30 symbols.
NEW surname: Baggins

OLD patronymic: Brandybuck
NEW patronymic: Drogo

OLD address: BagEnd, Hobbiton, Shire
NEW address: Bag End, Hobbiton, Shire, Middle Earth
Invalid input. Please enter value between 1 and 29 symbols.
NEW address: Bag End, Hobbiton, Shire
```

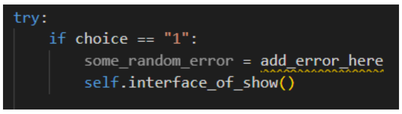
2.1.3 Log file after all operations



```
WARNING : Date-Time : 2024-04-28 14:31:14,274 : Line No. : 113 - File not found
WARNING : Date-Time : 2024-04-28 14:31:35,333 : Line No. : 125 - Invalid file structure: there is no header or footer
WARNING : Date-Time : 2024-04-28 14:33:17,834 : Line No. : 125 - Invalid file structure: there is no header or footer
INFO : Date-Time : 2024-04-28 14:33:33,667 : Line No. : 144 - Successfully read data from testFiles/validFile.txt
WARNING : Date-Time : 2024-04-28 14:39:31,267 : Line No. : 418 - Invalid input - amount
WARNING : Date-Time : 2024-04-28 14:39:38,778 : Line No. : 410 - Invalid input - amount
WARNING : Date-Time : 2024-04-28 14:39:45,693 : Line No. : 458 - Invalid input - currency
INFO : Date-Time : 2024-04-28 14:39:49,033 : Line No. : 286 - Successfully add transaction to testFiles/validFile.txt
WARNING : Date-Time : 2024-04-28 14:46:33,726 : Line No. : 382 - Invalid input - header - surname
INFO : Date-Time : 2024-04-28 14:46:38,368 : Line No. : 352 - Successfully change header from testFiles/validFile.txt
INFO : Date-Time : 2024-04-28 14:46:45,379 : Line No. : 196 - Successfully show header from testFiles/validFile.txt
WARNING : Date-Time : 2024-04-28 14:49:33,523 : Line No. : 418 - Invalid input - amount
WARNING : Date-Time : 2024-04-28 14:49:39,859 : Line No. : 458 - Invalid input - currency
INFO : Date-Time : 2024-04-28 14:49:43,939 : Line No. : 319 - Successfully change transaction number: 3 from testFiles/validFile.txt
WARNING : Date-Time : 2024-04-28 14:51:47,983 : Line No. : 382 - Invalid input - header - name
WARNING : Date-Time : 2024-04-28 14:52:23,498 : Line No. : 382 - Invalid input - header - surname
WARNING : Date-Time : 2024-04-28 14:53:07,933 : Line No. : 382 - Invalid input - header - address
INFO : Date-Time : 2024-04-28 14:53:26,151 : Line No. : 352 - Successfully change header from testFiles/validFile.txt
```

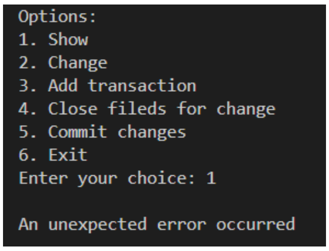
2.1.3 Handling Errors

Let’s add some random errors to the program. For example add random command to choice 1 of user interface in line 577:



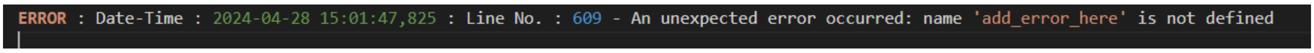
```
try:
    if choice == "1":
        some_random_error = add_error_here
        self.interface_of_show()
```

After this operation we can see information that there is a bug but our program did not crashed.



```
Options:
1. Show
2. Change
3. Add transaction
4. Close fileds for change
5. Commit changes
6. Exit
Enter your choice: 1

An unexpected error occurred
```

In log file we can see description of this error.



```
ERROR : Date-Time : 2024-04-28 15:01:47,825 : Line No. : 609 - An unexpected error occurred: name 'add_error_here' is not defined
```