

Problem statement:

An *administrator of an apartment building* (client) needs an application for *himself* (user).
The program shall help the administrator to manage the monthly expenses for each apartment in the building, for a certain month.

Apartment Building Administrator - Feature list

F1. Add a new transaction into the list
F2. Modify expenses from the list
F3. Write the expenses having different properties
F4. Obtain the different characteristics of expenses
F5. Filter the expenses
F6. Undo the last operation
F7. Redo the last undo

Iteration plan

Iteration	Planned features
I1	F1. Add a new transaction into the list F3. Write the expenses having different properties
I2	F2. Modify expenses from the list F4. Obtain the different characteristics of expenses
I3	F5. Filter the expenses F6. Undo the last operation F7. Redo the last undo

Iteration 1

Running scenario

	User	Program	Description
a		# Pick one of the options below. # 1 - Add a new transaction into the list # 3 - Write the expenses having different properties # 0 - Exit	Program shows the main menu.
b	1		User inputs 1.
		# You are going to add a new transaction into the list.	Program prompts user that he picked option 1.
c	50		User inputs 50 as expense.
d	2		User inputs 2 as type of expense (heating).
e	23		User inputs 23 as apartment's number.
f		# Transaction has been successfully added. # Pick one of the options below. # 1 - Add a new transaction into the list # 3 - Write the expenses having different properties # 0 - Exit	Program prompts user that he successfully added the transaction and shows the main menu.
g	3		User inputs 3.
h		# You are going to be written the expenses having different properties. # 1 - Write all the apartments with an overall expense greater than a certain value # 2 - Write all the apartments from a certain subsequence with an overall expense less than a certain value # 3 - Write all the expenses having a certain type # 4 - Compute the balance for a certain apartment # 0 - Return to main menu	Program prompts user that he picked option 3 and shows the corresponding submenu.
i	4		User inputs 4, corresponding to computing the balance.
j	23		User inputs 23, corresponding to the apartment's number.
k		# Apartment 23 with balance 100. # You are going to be written the expenses having different properties. # 1 - Write all the apartments with an overall expense greater than a certain value	Program prints the balance of the apartment 23 and shows the corresponding submenu

		# 2 - Write all the apartments from a certain subsequence with an overall expense less than a certain value # 3 - Write all the expenses having a certain type # 4 - Compute the balance for a certain apartment # 0 - Return to main menu	of the option 3.
l	0		User inputs 0 to exit the submenu.
m		# Pick one of the options below. # 1 - Add a new transaction into the list # 3 - Write the expenses having different properties # 0 - Exit	Program shows the main menu.
n	0		User inputs 0 to exit the program.
o		# Have a nice day!	Program shows a nice message.

Work items/tasks

T1	Read a natural number
T2	Check if a transaction (expense, type, apartment) is valid
T3	Add a transaction (expense, type, apartment) to the list
T4	Compute balance for an apartment (the total amount of expenses of all types)
T5	Select apartments with a balance greater than a certain value
T6	Select apartments with a balance less than a certain value
T7	Select apartments before a certain number
T8	Select the expenses of a certain type
T9	Implement user interface

Iteration 2

Running scenario

	User	Program	Description
a		# Main Menu # Pick one of the options below. # 1 - Add a new transaction into the list # 2 - Modify expenses from the list # 3 - Write the expenses having different properties # 4 - Obtain different characteristics of expenses # 0 - Exit	Program shows the main menu.
b	1		User inputs 1.
c		# You are going to add a new transaction into the list.	Program prompts user that he picked option 1.
d	50		User inputs 50 as expense.
e	2		User inputs 2 as type of expense (heating).
f	23		User inputs 23 as apartment's number.
g		# Transaction 50 for heating has been successfully added at apartment 23. # Main Menu # Pick one of the options below. # 1 - Add a new transaction into the list # 2 - Modify expenses from the list # 3 - Write the expenses having different properties # 4 - Obtain different characteristics of expenses # 0 - Exit	Program prompts user that he successfully added the transaction and shows the main menu.
h	2		User inputs 2.
i		# You are going to modify expenses from the list. # 1 - Remove all the expenses from a certain apartment # 2 - Remove all the expenses from a certain range of apartments # 3 - Remove all the expenses of a certain type from all the apartments # 4 - Replace the value of an expense from the list # 0 - Return to main menu	Program prompts user that he picked option 2 and shows the corresponding submenu.
j	4		User inputs 4, corresponding to replacing an expense.
k	23		User inputs 23 as apartment's number.
l	2		User inputs 2 as type of

			expense (heating).
m	100		User inputs 100 as new expense.
n		# Expense 50 for heating has been replaced to 100 at apartment 23. # You are going to modify expenses from the list. # 1 - Remove all the expenses from a certain apartment # 2 - Remove all the expenses from a certain range of apartments # 3 - Remove all the expenses of a certain type from all the apartments # 4 - Replace the value of an expense from the list # 0 - Return to main menu	Program prompts user that he successfully replaced the expense and shows the submenu again.
p	0		User inputs 0 to exit the submenu.
p		# Pick one of the options below. # 1 - Add a new transaction into the list # 3 - Write the expenses having different properties # 0 - Exit	Program shows the main menu.
q	0		User inputs 0 to exit the program.
r		# Have a nice day!	Program shows a nice message.

Work items/tasks

T1	Remove a sublist of expenses
T2	Select the expenses of a certain apartment
T3	Select the expenses of a certain amount
T4	Replace an expense with a certain index
T5	Sum all the expenses of a certain type
T6	Find the maximum expense from a list
T7	Sort expenses in ascending order
T8	Sort expenses in descending order
T9	Implement user interface

Iteration 3

Running scenario

	User	Program	Description
a		# Main Menu # Pick one of the options below. # 1 - Add a new transaction into the list # 2 - Modify expenses from the list # 3 - Write the expenses having different properties # 4 - Obtain different characteristics of expenses # 5 - Filter expenses # 6 - Undo last action that modified the list of expenses # 7 - Redo last action that modified the list of expenses # 0 - Exit	Program shows the main menu.
b	1		User inputs 1.
c		# You are going to add a new transaction into the list.	Program prompts user that he picked option 1.
d	50		User inputs 50 as expense.
e	2		User inputs 2 as type of expense (heating).
f	23		User inputs 23 as apartment's number.
g		# Transaction 50 for heating has been successfully added at apartment 23. # Main Menu # Pick one of the options below. # 1 - Add a new transaction into the list # 2 - Modify expenses from the list # 3 - Write the expenses having different properties # 4 - Obtain different characteristics of expenses # 5 - Filter expenses # 6 - Undo last action that modified the list of expenses # 7 - Redo last action that modified the list of expenses # 0 - Exit	Program prompts user that he successfully added the transaction and shows the main menu.
h	6		User inputs 2.
i		# Last modification has been undone. # Main Menu # Pick one of the options below. # 1 - Add a new transaction into the list # 2 - Modify expenses from the list # 3 - Write the expenses having different properties # 4 - Obtain different characteristics of expenses # 5 - Filter expenses # 6 - Undo last action that modified the list of expenses # 7 - Redo last action that modified the list of expenses	Program prompts user that he picked option 6 and that he undid the last modification.

		# 0 - Exit	
j	3		User inputs 3.
k		# You are going to be written the expenses having different properties. # 1 - Write all the apartments with an overall expense greater than a certain value # 2 - Write all the apartments from a certain subsequence with an overall expense less than a certain value # 3 - Write all the expenses having a certain type # 4 - Compute the balance for a certain apartment # 0 - Return to main menu	Program prompts user that he picked option 3 and shows the corresponding submenu.
l	4		User inputs 4, corresponding to computing the balance.
m	23		User inputs 23, corresponding to the apartment's number.
n		# No expenses for apartment 23. # You are going to be written the expenses having different properties. # 1 - Write all the apartments with an overall expense greater than a certain value # 2 - Write all the apartments from a certain subsequence with an overall expense less than a certain value # 3 - Write all the expenses having a certain type # 4 - Compute the balance for a certain apartment # 0 - Return to main menu	Program prints the balance of the apartment 23 and shows the corresponding submenu of the option 3.
p	0		User inputs 0 to exit the submenu.
p		# Main Menu # Pick one of the options below. # 1 - Add a new transaction into the list # 2 - Modify expenses from the list # 3 - Write the expenses having different properties # 4 - Obtain different characteristics of expenses # 5 - Filter expenses # 6 - Undo last action that modified the list of expenses # 7 - Redo last action that modified the list of expenses # 0 - Exit	
q	0		User inputs 0 to exit the program.
r		# Have a nice day!	Program shows a nice message.

Work items/tasks

T1	Group expenses of the same apartment and type into one expense
T2	Get the set of all apartments
T3	Implement backup versions of the list of transactions
T4	Undo last modification using the backup version
T5	Redo last undo using the backup version
T6	Implement user interface