Date: 19.10.2016 Name: Csoka Ervin

# Problem statement:

Jane is the administrator of an apartment building and she wants to manage the monthly expenses for  
each apartment. Jane needs an application to store, for a given month, the expenses for each  
apartment. Each expense is stored in the application using the following elements:  
-apartment(number of apartment, positive integer),  
-amount (positive integer),  
-type (from one of the predefined categories: water, heating, electricity, gas, other).  
Help Jane by creating an application that provides the following functionalities:  
1.Add <apartment> <type> <amount>  
2.Remove <apartment  
 Remove <start apartment> to <end apartment>  
 Remove <type>  
 Replace <apartment> <type> with <amount>  
3.List  
 List <apartment>  
 List [< | = | >] <amount>  
4.Sum <type>  
 Max <apartment>  
 Sort apartment  
 Sort type  
5.Filter <type>  
 Filter <value>  
6.Undo  
 The user will be able to undo all operations performed since program start.

# Feature list

|  |
| --- |
| F1. Add a new expense |
| F2. Remove all expenses for an apartment |
| F3. Remove expenses from an interval of apartments |
| F4. Remove all expenses of a type |
| F5. Replace the amount of expenses for a given apartment of a given type with a new amount. |
| F6. List all expenses |
| F7. List all expenses for an apartment |
| F8. List all expenses with the amount satisfying a given condition (<, <=, =, >=, >) |
| F9. Sum up all expenses of a given type |
| F10. List the maximum amount for each expense type for a given apartment |
| F11. Sort the apartments ascendingly by total amount of expenses |
| F12. Sort ascendingly the total amount of expenses for each type |
| F13. Remove all the expenses not having the given type |
| F14. Remove all expenses having the mount greater or equal to a given amount |
| F15. Undo the last operation that modified the program state |

# Iteration plan

|  |  |
| --- | --- |
| **Iteration** | **Planned features** |
| I1 | F1-F8 |
| I2 | F9-F15 |

# Iteration 1

## Running scenario

|  |  |  |  |
| --- | --- | --- | --- |
|  | User | Program | Description |
| a |  | initList() | Initialize the list with 10 entries |
| b | Add 55 gas 10 |  | Add the expense 55: gas, 10 to the list |
| c | Remove 7 |  | Removes all expenses for apartment 7 |
| d | list | 11: gas, 100  1: water, 200  5: other, 10  5: heating, 12  11: other, 55  11: other, 1055  12: gas, 99  13: heating, 55  12: other, 77  1: electricity, 66  55: gas, 100 | Lists all expenses |
| e | Remove 1 to 5 |  | Removes all expenses for the apartments in the interval [1, 5] |
| f | Remove gas |  | Removes all expenses with type gas |
| g | list | 11: other, 55  11: other, 1055  13: heating, 55  12: other, 77 | Lists all expenses |
| h | List 11 | 11: other, 55  11: other, 1055 | Lists all expenses for apartment 11 |
| i | List <= 55 | 11: other, 55  13: heating, 55 | Lists all expenses with the amount <= 55 |

**Work items/tasks**

|  |  |
| --- | --- |
| T1 | Create a new dictionary for an expense |
| T2 | Validate the format of an expense |
| T3 | Add an expense to the list |
| T4 | Remove all expenses for an apartment |
| T5 | Remove all expenses from an interval of apartments |
| T6 | Remove all expenses for a type |
| T7 | Replace the amount of expenses for a given apartment of a given type with a new amount. |
| T8 | Print an expense |
| T9 | Print all expenses |
| T10 | Print all expenses for an apartment |
| T11 | Print all expenses that satisfy a given condition |
| T12 | UI command for adding to the list |
| T13 | UI command for removing from the list |
| T14 | UI command for printing |
| T15 | List available commands |
| T16 | Read a command from the console |
| T17 | Start program loop |

# Iteration 2

## Running scenario

|  |  |  |  |
| --- | --- | --- | --- |
|  | User | Program | Description |
| a |  | initList() | Initialize the list with 10 entries |
| b | Sum gas | Sum is: 449 |  |
| c | Max 11 | {'water': 0, 'heating': 0, 'electricity': 0, 'gas': 100, 'other': 1055} |  |
| d | Sort apartment | 5 , 22  13 , 55  12 , 176  1 , 266  11 , 1210  7 , 9250 |  |
| e | Sort type | heating, 67  water, 200  gas, 449  other, 1197  electricity, 9066 |  |
| f | Filter gas  List | 11: gas, 100  12: gas, 99  7: gas, 250 | Keeps only gas type expenses in the list |

**Work items/tasks**

|  |  |
| --- | --- |
| T1 | Return the sum of all expenses of a type |
| T2 | Return a dictionary with the maximum amount for each expense type for an apartment |
| T3 | Sort the apartments ascendingly by total amount of expenses |
| T4 | Sort ascendingly the total amount of expenses for each type |
| T5 | Remove all the expenses not having a given type |
| T6 | Remove all expenses having the mount greater or equal to a given amount |
| T7 | UI command for the sum |
| T8 | UI command for the maximum per expense |
| T9 | UI command for sorting |
| T10 | UI command for filtering |
| T11 | Return a list with 10 elements |
| T12 | Undo the last operation that modified the program state |