

LABS 2 - 4

Requirements

You will be assigned with a problem from the list below (P1,P2,...) and you are required to fulfil the following:

- Create a C application
- Use simple feature driven development
- All the needed functions will be specified and tested, use test driven development
- Apply the layered architecture pattern
- Create a module for a list container (Dynamic Array)
- Create at least 3 modules :

Ex for the contest problem: module for UI, module for ContestManager, module for ADT List (or Dynamic array, ..) , optional: a module with utility functions.

P1. Contest

At a **programming contest**, after evaluating the existing solutions, the evaluation committee has recorded into a list the scores obtained by the participants (given name, family name, score). Write a program in order to help the committee to repeatedly execute the following functionalities:

1. **Add new participant.**
2. **Modify a participant (name, score).**
3. **Remove a participant**
4. **Filter participants.**
 - a. By score
 - b. By name
5. **Undo the last operation.**
 - a. *undo* – the last operation that has modified the list of scores is cancelled.

P2. Family expenses management

A **family** wants to manage its monthly expenses. In order to complete this task, the family needs an application to store, for a certain month, all the family's expenses. Each expense will be stored in the

application through the following elements: day (of the month in which it was made), amount of money and the type of the expense (the family wants to group its expenses in the following categories: house keeping, food, transport, clothing, telephone&internet, others – books, films, sports, etc). The family needs an application in order to repeatedly execute the following functionalities:

1. **Add a new expense**
2. **Modify expenses**
3. **Remove expense.**
4. **Obtain different characteristics of sublists.**
5. **Filter.**
 - a. By type
 - b. By amount
6. **Undo the last operation.**
 - a. *undo* – the last operation that has modified the list of expenses is cancelled.

P3. Bank account management

John wants to manage its bank account. In order to complete this task, John needs an application to store, for a certain month, all the banking transactions which were performed on his account. Each transaction will be stored in the application through the following elements: day (of the month in which the transaction was made), amount of money transferred into/from the account, the type of the transaction (into the account – **in** or from the account - **out**), and description of the transaction. Please help **John** to create an application in order to repeatedly execute the following functionalities:

1. **Add a new transaction.**
 2. **Modify transactions**
 3. **Remove transaction.**
 4. **Filter.**
 - By type
 - By amount
 5. **Undo the last operation.**
 - *undo* – the last operation that has modified the list of transactions is cancelled.
-

P4. Bloc administrator

Michael is the administrator of a bloc and wants to manage the monthly expenses for each apartment in the bloc. In order to complete this task, Michael needs an application to store, for a certain month, the expenses for each apartment. Each expense will be stored in the application through the following elements: amount, type of the expense (the administrator wants to group the expenses in several predefined categories: such as water, heating, illuminating, gas, others). Michael needs an application in order to repeatedly execute the following functionalities :

1. **Add a new expense.**
2. **Modify expenses.**
3. **Remove expense**
4. **Filter.**
 - By type
 - By amount
5. **Undo the last operation.**
 - *undo* – the last operation that has modified the list of expenses is cancelled.

P5. Pharmacy management

Create an application to manage the stock of medicines (id, name, active substance, quantity in stock) in a pharmacy.

1. **Add a new medicine.**
2. **Modify medicine.**
3. **Remove medicine**
4. **Filter.**
 1. By name
 2. By quantity
5. **Undo the last operation.**
 - *undo* – the last operation that has modified the list of medicines is cancelled.

P6. Car rental

Create an application to manage cars (nr, model, category – mini, sport, suv) for a car rental office.

1. **Add a new car.**
2. **Modify car.**
3. **Remove car**
4. **Filter.**
 1. By model
 2. By category
5. **Undo the last operation.**
 - *undo* – the last operation that has modified the list of cars is cancelled.

P7. Bakery

Create an application to manage the stock of ingredients (name, provider ,quantity) in a bakery.

1. **Add a new ingredient.**
2. **Modify ingredient.**
3. **Remove ingredient**
4. **Filter.**
 1. By quantity
 2. By name
5. **Undo the last operation.**
 - *undo* – the last operation that has modified the list of ingredients is cancelled.

P8. Shop

Create an application to manage the stock of products (name, type – laptop, tv,... - , quantity) in a shop.

1. **Add a new product.**
2. **Modify product.**
3. **Remove product**
4. **Filter.**
 1. By quantity
 2. By name
5. **Undo the last operation.**
 - *undo* – the last operation that has modified the list of products is cancelled.

P9. Travel agency

Create an application to manage the holiday offerings (destination, type – city break,..., price) in a travel agency.

1. **Add a new holiday offer.**
2. **Modify holiday offer.**
3. **Remove offer**
4. **Filter.**
 1. By type
 2. By price
5. **Undo the last operation.**
 - *undo* – the last operation that has modified the list of holidays is cancelled.

P10. Real estate agency

Create an application to manage real estate (type – land, deposit,..., address, price) offered by a real estate agency.

1. **Add a new real estate.**
2. **Modify real estate.**
3. **Remove real estate**
4. **Filter.**
 1. By type
 2. By price
5. **Undo the last operation.**
 - *undo* – the last operation that has modified the list of real estates is cancelled.