

LABS 4 - 6

Requirements

Fiecare student primește o problemă din lista de mai jos.

Cerințe:

- Folosiți procesul de dezvoltare: Incrementală bazată pe funcționalități (vezi curs 1) și Dezvoltare dirijată de teste (curs 2)
- Planificați iterații pentru 3 laboratoare succesive. În fiecare săptămână primiți o notă pentru ce s-a realizat pentru iterația din săptămâna curentă.
- Prima iterație trebuie să conțină cel puțin 3 cerințe (din funcționalitățile 3-5)
- Documentația trebuie să conțină: enunțul, lista de funcționalități, planul de iterații, scenarii de rulare, lista de taskuri (activități)
- Toate funcțiile trebuie să includă specificații, toate funcțiile trebuie să fie testate (funcții de test cu assert) în afară de partea cu interacțiunea utilizator.
- Separați partea de interfață utilizator de restul aplicației (să **nu** aveți funcții care fac 2 lucruri: un calcul + tipărire/citire)
- La prima iterație se cere o soluție procedurală (mai multe funcții toate în același modul), varianta finală trebuie să fie modulară (curs 3)
- Datele de intrare trebuie validate, programul semnalează erorile către utilizator.

P1. Numere

Creați un program care lucrează cu numere complexe ($a + bi$). Programul gestionează o listă de numere complexe și permite efectuarea repetată a următoarelor acțiuni:

1. **Adaugă număr în listă.**
 - a. Adaugă număr complex la sfârșitul listei
 - b. Inserare număr complex pe o poziție dată.
2. **Modifică elemente din listă.**
 - a. Șterge element de pe o poziție dată.
 - b. Șterge elementele de pe un interval de poziții.

- c. Înlocuiește toate aparițiile unui număr complex cu un alt număr complex.
- 3. **Căutare numere.**
 - a. Tipărește partea imaginara pentru numerele din listă. Se dă intervalul de poziții (sub secvența).
 - b. Tipărește toate numerele complexe care au modulul mai mic decât 10
 - c. Tipărește toate numerele complexe care au modulul egal cu 10
- 4. **Operații cu numerele din listă**
 - a. suma numerelor dintr-o subsecvență dată (se da poziția de început și sfârșit).
 - b. Produsul numerelor dintr-o subsecvență dată (se da poziția de început și sfârșit).
 - c. Tipărește lista sortată descrescător după partea imaginara
- 5. **Filtrare.**
 - a. Filtrare parte reala prim – elimină din listă numerele complexe la care partea reala este prim.
 - b. Filtrare modul – elimina din lista numerele complexe la care modulul este \leq sau $>$ decât un număr dat.
- 6. **Undo**
 - a. Reface ultima operație (lista de numere revine la numerele ce existau înainte de ultima operație care a modificat lista)

P2. Concurs

Creați o aplicație pentru gestiunea concurenților de la un concurs de programare. Programul înregistrează scorul obținut de fiecare concurent la 10 probe diferite, fiecare probă este notat cu un scor de la 1 la 10. Fiecare participant este identificat printr-un număr de concurs, scorul este ținut într-o listă unde concurentul 3 are scorul pe poziția 3 în listă . Programul trebuie sa ofere următoarele funcționalități::

- 1. **Adaugă un scor la un participant.**
 - a. Adaugă scor pentru un nou participant (ultimul participant)
 - b. Inserare scor pentru un participant.
- 2. **Modificare scor.**
 - a. Șterge scorul pentru un participant dat.
 - b. Șterge scorul pentru un interval de participanți.
 - c. Înlocuiește scorul de la un participant.
- 3. **Tipărește lista de participanți.**
 - a. Tipărește participanții care au scor mai mic decât un scor dat.
 - b. Tipărește participanții ordonat după scor
 - c. Tipărește participanții cu scor mai mare decât un scor dat, ordonat după scor.
- 4. **Operații pe un subset de participanți.**
 - a. Calculează media scorurilor pentru un interval dat (ex. Se da 1 și 5 se tipărește media scorurilor participanților 1,2,3,4 și 5
 - b. Calculează scorul minim pentru un interval de participanți dat.
 - c. Tipărește participanții dintr-un interval dat care au scorul multiplu de 10.
- 5. **Filtrare.**
 - a. Filtrare participanți care au scorul multiplu unui număr dat. Ex. Se da numărul 10, se elimină scorul de la toți participanții care nu au scorul multiplu de 10.
 - b. Filtrare participanți care au scorul mai mic decât un scor dat.
- 6. **Undo**
 - a. Reface ultima operație (lista de scoruri revine la numerele ce existau înainte de ultima operație care a modificat lista)

P3. Cheltuieli de familie

Scrieți o aplicație care gestionează cheltuielile de familie efectuate într-o lună. Pentru o cheltuială se vor retine ziua (din luna), suma, tipul cheltuielii (5 categorii: mâncare, întreținere, îmbrăcăminte, telefon, altele). Aplicația să permită efectuarea repetată a următoarelor operații:

1. **Adaugă cheltuială**
 - a. adaugă o nouă cheltuială (se specifică ziua, suma, tipul)
 - b. actualizează cheltuială (se specifică ziua, suma, tipul)
2. **Ștergere.**
 - a. Șterge toate cheltuielile pentru ziua dată
 - b. Șterge cheltuielile pentru un interval de timp (se dă ziua de început și sfârșit, se șterg toate cheltuielile pentru perioada dată)
 - c. Șterge toate cheltuielile de un anumit tip
3. **Căutări.**
 - a. Tipărește toate cheltuielile mai mari decât o sumă dată
 - b. Tipărește toate cheltuielile efectuate înainte de o zi dată și mai mici decât o sumă (se dă suma și ziua, se tipăresc toate cheltuielile mai mici ca suma dată și efectuate înainte de ziua specificată)
 - c. tipărește toate cheltuielile de un anumit tip.
4. **Rapoarte.**
 - a. Tipărește suma totală pentru un anumit tip de cheltuială
 - b. Găsește ziua în care suma cheltuită e maximă
 - c. Tipărește toate cheltuielile ce au o anumită sumă
 - d. Tipărește cheltuielile sortate după tip
5. **Filtrare.**
 - a. Elimină toate cheltuielile de un anumit tip
 - b. Elimină toate cheltuielile mai mici decât o sumă dată
6. **Undo**
 - a. Reface ultima operație (lista de cheltuieli revine ce exista înainte de ultima operație care a modificat lista).

P4. Cont bancar

Scrieți o aplicație pentru gestiunea unui cont bancar. Fiecare tranzacție are ziua, suma, și tipul (intrare/ieșire). Aplicația permite:

1. **Adăugare de noi tranzacții**
 - adaugă tranzacție (se dă ziua, suma, tipul)
 - actualizare tranzacție (se dă ziua, suma, tipul)
2. **Ștergere.**
 - Șterge toate tranzacțiile de la ziua specificată
 - Șterge tranzacțiile dintr-o perioadă dată (se dă ziua de început și sfârșit)
 - Șterge toate tranzacțiile de un anumit tip

3. **Căutări.**
 - Tipărește tranzațiile cu sume mai mari decât o sumă dată
 - Tipărește toate tranzațiile efectuate înainte de o zi și mai mari decât o sumă (se dă suma și ziua)
 - Tipărește tranzațiile de un anumit tip
4. **Rapoarte**
 - Suma totală a tranzațiilor de un anumit tip
 - Soldul contului la o dată specificată
 - Tipărește toate tranzațiile de un anumit tip ordonat după sumă
5. **Filtrare.**
 - Elimină toate tranzațiile de un anumit tip
 - Elimină toate tranzațiile mai mici decât o sumă dată care au tipul specificat
6. **Undo.**
 - Reface ultima operație (lista de tranzații revine la ce exista înainte de ultima operație care a modificat lista).

P5. Administrator de bloc

Creați o aplicație pentru gestiunea cheltuielilor lunare de la apartamentele unui bloc de locuințe. Aplicația stochează cheltuielile pentru fiecare apartament: suma și tipul cheltuielii (tip poate fi una dintre: apa, canal, încălzire, gaz, altele). Aplicația permite:

1. **Adăugare**
 - adaugă cheltuială pentru un apartament
 - modifică cheltuială
2. **Ștergere**
 - Șterge toate cheltuielile de la un apartament
 - Șterge cheltuielile de la apartamente consecutive (Ex. se dau două numere de apartament 2 și 5 și se șterg toate cheltuielile de la apartamentele 1,2,3,4 și 5)
 - Șterge cheltuielile de un anumit tip de la toate apartamentele
3. **Căutări**
 - Tipărește toate apartamentele care au cheltuieli mai mari decât o sumă dată
 - Tipărește cheltuielile de un anumit tip de la toate apartamentele
 - Tipărește toate cheltuielile efectuate înainte de o zi și mai mari decât o sumă (se dă suma și ziua)
4. **Rapoarte**
 - Tipărește suma totală pentru un tip de cheltuială
 - Tipărește toate apartamentele sortate după un tip de cheltuială
 - Tipărește totalul de cheltuieli pentru un apartament dat
5. **Filtru**
 - Elimină toate cheltuielile de un anumit tip
 - Elimină toate cheltuielile mai mici decât o sumă dată
6. **Undo the last operation.**
 - Reface ultima operație (lista de cheltuieli revine la ce exista înainte de ultima operație care a modificat lista).