

LAB 4 - 6

Cerințe

Fiecare student primește o problemă din lista de mai jos.

Cerințe generale:

- 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 (sa **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.

- Adaugă număr complex la sfârșitul listei
- Inserare număr complex pe o poziție dată.

2. Modifică elemente din listă.

- Șterge element de pe o poziție dată.
- Șterge elementele de pe un interval de poziții.
- Înlocuiește toate aparițiile unui număr complex cu un alt număr complex.

3. Căutare numere.

- Tipărește partea imaginara pentru numerele din listă. Se dă intervalul de poziții (sub secvența).
- Tipărește toate numerele complexe care au modulul mai mic decât 10
- Tipărește toate numerele complexe care au modulul egal cu 10

4. Operații cu numerele din listă

- suma numerelor dintr-o subsecvență dată (se da poziția de început și sfârșit).
- Produsul numerelor dintr-o subsecvență dată (se da poziția de început și sfârșit).
- Tipărește lista sortată descrescător după partea imaginara

5. Filtrare.

- Filtrare parte reala prim – elimină din listă numerele complexe la care partea reala este prim.
- Filtrare modul – elimina din lista numerele complexe la care modulul este \leq sau $>$ decât un număr dat.

6. Undo

- Reface ultima operație (lista de numere revine la numerele ce existau înainte de ultima operație care a modificat lista) – **Nu** folosiți funcția **deepCopy**
-

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

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

- adaugă o nouă cheltuială (se specifică ziua, suma, tipul)
- actualizează cheltuială (se specifică ziua, suma, tipul)

2. Ștergere.

- Șterge toate cheltuielile pentru ziua dată
- Șterge cheltuielile pentru un interval de timp (se dă ziua de început și sfârșit, se șterg toate cheltuielile pentru perioada dată)
- Șterge toate cheltuielile de un anumit tip

3. Căutări.

- Tipărește toate cheltuielile mai mari decât o sumă dată
- 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)
- tipărește toate cheltuielile de un anumit tip.

4. Rapoarte.

- Tipărește suma totală pentru un anumit tip de cheltuială
- Găsește ziua în care suma cheltuită e maximă
- Tipărește toate cheltuielile ce au o anumită sumă
- Tipărește cheltuielile sortate după tip

5. Filtrare.

- Elimină toate cheltuielile de un anumit tip
- Elimină toate cheltuielile mai mici decât o sumă dată

6. Undo

- Reface ultima operație (lista de cheltuieli revine ce exista înainte de ultima operație care a modificat lista). – **Nu** folosiți funcția **deepCopy**

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

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**

- Reface ultima operație (lista de cheltuieli revine la ce exista înainte de ultima operație care a modificat lista). – **Nu** folosiți funcția **deepCopy**

P6. AGENȚIE DE TURISM

Creați o aplicație pentru gestiunea pachetelor de călătorie oferite de o agenție de turism. Aplicația stochează pachete de călătorie oferite clienților după cum urmează: data de început și cea de sfârșit a călătoriei, destinația și prețul.

Aplicația va permite:

1. Adăugare

- Adaugă pachet de călătorie
- Modifică pachet de călătorie

2. Ștergere

- Ștergerea tuturor pachetelor de călătorie disponibile pentru o destinație dată
- Ștergerea tuturor pachetelor de călătorie care au o durată mai scurtă decât un număr de zile dat
- Ștergerea tuturor pachetelor de călătorie care au prețul mai mare decât o sumă dată

3. Căutare

- Tipărirea pachetelor de călătorie care presupun un sejur într-un interval dat (e.g. 10/08/2018 - 24/08/2018 - un pachet valid este cel a cărui dată de început este aceeași sau după de data de început citită și data de sfârșit este înainte sau aceeași cu data de sfârșit introdusă de la tastatură)
- Tipărirea pachetelor de călătorie cu o destinație dată și cu preț mai mic decât o sumă dată.
- Tipărirea pachetelor de călătorie cu o anumită dată de sfârșit

4. Rapoarte

- Tipărirea numărului de oferte pentru o destinație dată.
- Tipărirea tuturor pachetelor disponibile într-o anumită perioadă citită de la tastatură (vezi 3.i.) în ordinea crescătoare a prețului.
- Tipărirea mediei de preț pentru o destinație dată

5. Filtrare

- Eliminarea ofertelor care au un preț mai mare decât cel dat și o destinație diferită de cea citită de la tastatură
- Eliminarea ofertelor în care sejurul presupune zile dintr-o anumită lună (citită de la tastatură sub forma unui număr natural între 1 și 12)

6. Undo

- Refacerea ultimei operații (lista de oferte revine la ce exista înainte de ultima operație care a modificat lista) – **Nu** folosiți funcția deepCopy