

# LABS 7 - 9

## Cerințe

---

- Folosiți dezvoltarea iterativă bazat pe funcționalități
- Identificați și planificați funcționalități pe 3 iterații
- Folosiți dezvoltare dirijată de teste. Toate funcțiile trebuie testate și specificate
- Folosiți arhitectură stratificată (UI, Controller, Domain, Repository)
- Validați datele – pentru intrări invalide, aplicația sa tipărească mesaje de eroare corespunzătoare – folosiți excepții.
- Documentația conține: Enunț, lista de funcționalități, planul de iterații. Pentru fiecare funcționalitate: scenariu de rulare

## P1. Catalog studenți

---

Facultatea stochează informații despre:

- studenți: <IDStudent>, <nume>
- discipline: <idDisciplină>, <nume>, <profesor>

Creați o aplicație care permite:

- gestionarea listei de studenți și listei de discipline.
- adaugă, șterge, modifică, lista de studenți, listă de discipline
- căutare student, căutarea de disciplină.
- Asignare de note la un student și o disciplină
- Creare de statistici:
  - lista de studenți și notele lor la o disciplină dată, ordonat: alfabetic după nume, după notă.
  - Primi 20% din studenți ordonat după media notelor la toate disciplinele (nume și notă)

## P2. Gestiune laboratoare studenți

Scrieți o aplicație pentru gestiunea notelor și a problemelor de laborator pentru o disciplină.

Aplicația gestionează:

- studenți: <studentID>, <nume>, <grup>
- problemă laborator: <număr laborator\_număr problemă>, <descriere>, <deadline>

Creați o aplicație care permite:

- gestiunea listei de studenți și probleme de laborator.
- adaugă, șterge, modifică, lista de studenți, listă de probleme
- căutare student, căutare problemă
- Asignare laborator/Notare laborator
- Creare statistici:
  - lista de studenți și notele lor la o problema de laborator dat, ordonat: alfabetic după nume, după notă.
  - Toți studenții cu media notelor de laborator mai mic decât 5. (nume student și notă)

### P3. Închiriere filme

Scrieți o aplicație pentru o firmă de închiriere de filme.

Aplicația stochează:

- filme: <id>, <titlu>, <descriere>, <gen>, etc
- clienți: <id>, <nume>, <CNP>, etc

Creați o aplicație care permite:

- gestiunea listei de filme și clienți.
- adaugă, șterge, modifică, lista de filme, lista de clienți
- căutare film, căutare clienți.
- Închiriere/returnare filme
- Rapoarte:
  - Clienți cu filme închiriate ordonat după: nume, după numărul de filme închiriate
  - Cele mai închiriate filme.
  - Primi 30% clienți cu cele mai multe filme (nume client și numărul de filme închiriate)

### P4. Bibliotecă

Scrieți o aplicație pentru o bibliotecă.

Aplicația stochează:

- cărți: <id>, <titlu>, <descriere>, <autor>, etc
- clienți: <id>, <nume>, <CNP>, etc

Creați o aplicație care permite:

- gestiunea listei de cărți și clienți.
- adaugă, șterge, modifică, lista de cărți, lista de clienți
- căutare carte, căutare clienți.
- Închiriere/returnare carte
- Rapoarte:
  - Cele mai închiriate cărți.
  - Clienți cu cărți închiriate ordonat după: nume, după numărul de cărți închiriate
  - Primi 20% dintre cei mai activi clienți (nume client și numărul de cărți închiriate)

### P5. Organizare evenimente

Scrieți o aplicație pentru organizarea de evenimente

Aplicația stochează:

- > persoane: <personID>, <nume>, <adresă>
- > evenimente: <ID>, <dată>, <timp>, <descriere>

Creați o aplicație care permite:

- gestiunea listei de persoane și evenimente.
- adaugă, șterge, modifică, lista de persoane, lista de evenimente
- căutare persoane, căutare evenimente
- Înscriere persoană la eveniment.
- Rapoarte:
  - Lista de evenimente la care participă o persoană ordonat alfabetic după descriere, după dată
  - Persoane participante la cele mai multe evenimente
  - Primele 20% evenimente cu cei mai mulți participanți (descriere, număr participanți)