

SEMINAR 3

TO COVER:

- Funcții
- Specificațiile funcțiilor
- Scriere de teste și procesul TDD

1. Creați o aplicație cu interfață de tip consolă care gestionează o listă de task-uri.

Fiecare task are trei atribute:

descriere: descrierea task-ului (str, cel puțin 2 caractere)

data deadline: data deadline a task-ului (format dd-mm)

status: statusul task-ului (*pending*, *in-progress*, *done*)

Aplicația permite efectuarea repetată a următoarelor acțiuni:

1. [F1] Adăugare task (se citesc informații pentru câmpuri de la tastatură).
2. [F2] Căutare task cu deadline între două date citite de la tastatură.
3. [F3] Ștergere task cu descriere dată.
4. [F4] Eliminarea din listă a tuturor task-urilor cu un anumit status.
5. [F5] Afișarea task-urilor care conțin în descriere un string dat de la tastatură.
6. [F6] Afișare raport pe lună (luna - numărul de task-uri).
7. [F7] Undo pentru operațiile care modifică lista.

Adițional, se vor implementa și acțiuni de **printare** a listei de task-uri, și de **adăugare a unor task-uri** în mod default [F8]. Datele de intrare cu privire la task-uri se vor valida [F9].

Q:

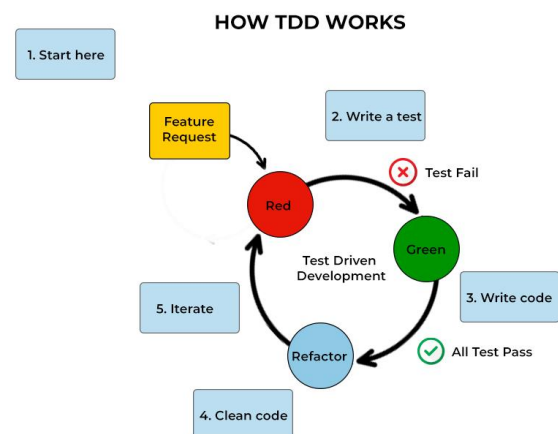
1) Cum reprezentăm datele din problemă?

2) Cum implementăm soluția pentru această problemă urmând un proces TDD? La ce ajută TDD?

3) Cum implementăm operația de **undo**?

4) Cum efectuăm validarea datelor?

- a) Validare input
- b) Validare în cadrul listei (e.g. task-uri duplicat)



PLAN

SEMINAR 3	F1, F8, F2, F4	Funcții, funcții de test, TDD
SEMINAR 4	F3, F5, F6	More TDD, Modularizare
SEMINAR 5	F7, F9	Aspecte legate de scrierea codului ușor testabil, ușor modificabil Excepții

F1-F9. Modelare → Activități → Cazuri de test (funcții de test)

Modelare: scenarii de rulare

Activități: tabel de task-uri

Teste: input-expected output, funcții cu assert

APPLIED TDD

