

## Etapa II

1. Extindeți proiectul din prima etapa prin realizarea persistenței utilizând fișiere. Se vor realiza fișiere cu extensia .csv (comma separated values) pentru cel puțin 4 dintre clasele definite în prima etapa.
  - se vor realiza servicii **singleton**(check end of pdf) pentru scrierea și citirea din fișier.
  - la pornirea programului se vor încărca datele din fișiere utilizând serviciile create.
  - folosiți pachete pentru a vă structura clasele (pachet pentru entități, pachet pentru servicii, pachet pentru excepții - pentru cei care la care am văzut că au folosit și excepții )
2. Realizarea unui serviciu de audit
  - se va realiza o clasă de serviciu care să scrie într-un fișier de tip CSV de fiecare dată când este executată una dintre acțiunile/interogările descrise în prima etapă. Structura fișierului în care se scrie:  
nume\_actiune, timestamp

Exemplu:

Dacă în main o să apălați o acțiune de `showAvailableTransactions()`, în fișierul de audit o să fie afișat:

**showAvailableTransactions, 2011-10-02 18:48:05**

**Deadline: 27.04.2020 23:59**

## Singleton pattern:

- Presupune crearea unei singure instanțe pentru o clasă prin care să fie gestionată o resursă/un eveniment în mod centralizat;
- Asigură un singur punct de acces, vizibil global, la unica instanță;
- Exemple: gestiune conexiune baze de date sau alte resurse.

```
public class SingletonService {  
  
    // un atribut static si privat de tipul clasei (reprezinta instanta unica care o sa fie folosita)  
    private static SingletonService instance = null;  
  
    // un constructor privat (apelabil doar in clasa respectiva)  
    private SingletonService() {  
    }  
  
    // o metoda publica ce da access la instanta unica  
    // instanta este create la primul apel al metodei  
    public static SingletonService getInstance() {  
        if (instance == null) {  
            instance = new SingletonService();  
        }  
        return instance;  
    }  
}
```