

Evaluare – Test 2

Timp de lucru: 1 oră și 50 minute

Până la finalul timpului de lucru încărcați rezolvările pe repository-ul creat prin acceptarea assignment-ului **Lab 11**. Programul va fi evaluat strict în forma disponibilă pe acest repository la expirarea timpului de lucru.

Scrieți un program cu meniu de tip consolă. Programul va reține datele într-un fișier. Vor fi suportate următoarele funcționalități:

1. [1p] Adăugare client: id, nume (string nenul), **CNP (13 caractere)**, sold initial (pozitiv).
Important:
După fiecare adăugare se vor afișa toate obiectele pentru entitatea **Client**.
2. [2p] Adăugare tranzacție: id, id_client_sursa, id_client_destinatia (trebuie să existe și să fie distincte), valoare. **La adaugarea unei tranzacții trebuie verificat dacă soldul clientului sursa are destule fonduri, iar soldurile clienților trebuie actualizate în cazul în care tranzacția este permisă.**
Important:
După fiecare adăugare se vor afișa toate obiectele pentru entitatea **Tranzactie**.
3. [2p] Afișarea clienților ordonați descrescător după numărul de vocale (a, e, i, o, u) din nume. Se va afișa și acest număr.
4. [2p] Afișarea tuturor tranzacțiilor care sunt considerate periculoase. O tranzacție este **considerată periculoasă dacă are o valoare mai mare de 15000. Se vor afișa și numele și CNP-urile clienților implicați în astfel de tranzacții.**
5. [3p] Export istoric tranzactii in format JSON: se citește un CNP de la tastatura si se construiește istoricul clientului căutat astfel: se creează un fișier JSON cu un obiect în care cheile sunt id-urile tranzacțiilor, iar valoarea unei chei este un obiect de tip tuplu care contine următoarele informații: “trimitere” dacă clientul căutat este sursa / “primire” dacă clientul căutat este destinatar, numele celui alt client implicat, valoare.

Exemplu de export valid:

```
{
  "1" : ("trimitere", "Mihai", 25.0),
  "7" : ("primire", "Florin", 45.0),
  "25" : ("trimitere", "Andreea", 25.0)
}
```

Studentii de la matematică pot implementa această cerință cu afișare pe ecran în loc de fișier JSON.

Punctajul pe fiecare cerință se acordă astfel:

- 25% corectitudinea implementării.
- 25% arhitectură stratificată și interfață utilizator user friendly.
- 25% specificații scrise corect (unde se aplică) și denumiri sugestive.
- 25% teste relevante și scrise corect (unde se poate).
- studenții de la Matematică pot obține punctaj maxim fără să scrie teste (procentajul aferent se redistribuie la corectitudine), fără să rețină datele în fișiere și fără să folosească repository, dar dacă fac aceste lucruri pot obține 3 puncte bonus. Aceste bonusuri se acordă proporțional cu funcționalitățile rezolvate.
- studenții de la Mate-info și Fizică-info pot obține 1 punct bonus dacă au maxim 3 warning-uri PEP 8 și 2 puncte bonus dacă nu au niciun warning PEP 8. Aceste bonusuri se acordă doar pentru punctaje inițiale ≥ 7 .
- o cerință nefuncțională se notează cu 0.
- nereținerea datelor în fișiere de către studenții de la mate-info și fizică-info duce la o penalizare de 50% din punctajul obținut conform baremului.
- o implementare non-OOP duce la o penalizare de 50% din punctajul obținut conform baremului.