

Evaluare – Test 2

Timp de lucru: 1 oră și 50 minute

Până la finalul timpului de lucru încărcăți 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 Produs:**
 - **Id:** unic și nenul
 - **Nume:** trebuie să nu conțină cifre și să fie nenul
 - **Pret:** trebuie să fie un număr pozitiv

Important:

După fiecare adăugare se vor afișa toate valorile din entitatea **Produs**.

Exemplu:

Id	Nume	Pret
1	Burger	20
2	Cartofi prajiti	10
3	Hot dog	15
4	Cola	5

2. **[2p] Adăugare Comanda:**
 - **Id:** unic și nenul
 - **Data:** formatul trebuie să fie zi/luna/an. Exemplu: 12/09/2021
 - **Produse:** o listă de id-uri valide din entitatea Măncare. Minim 1 valoare
 - **ModDePlata:** suporta 2 valori - card sau cash. Nu poate fi nul

Important:

După fiecare adăugare se vor afișa toate valorile din entitatea **Comanda**.

Exemplu:

Id	Data	Produse	ModDePlata
1	12/12/2021	1,1,2	cash
2	10/11/2021	3,1	card

3. **[2p] Afișați prețul unei comenzi în funcție de un id dat.**

Exemplu:

Pentru Id: 1
Se va afisa:
Pretul comenzii este 50

Mod de calcul: pret burger + pret burger + pret cartofi prajiti = 20 + 20 + 10 = 50

4. **[2p]** Afișați toate produsele care conțin un text dat de la tastatura. Căutarea trebuie sa functioneze si cu litere mari și cu litere mici.

Exemplu:

Textul cautat: BUR

Se va returna:

Id: 1, Nume: Burger, Pret: 20

Textul cautat: brr

Se va returna:

Nu exista produse pentru textul cautat.

5. **[3p]** Export JSON: se creează un fișier JSON cu un obiect în care cheile sunt numele produselor, iar valoarea unei chei X este o listă cu datele în care a fost comandat acel produs.

Exemplu:

```
{  
  "Burger": ["12/12/2021"],  
  "Cartofi prajiti": ["12/12/2021", "10/11/2021"],  
  "Hot dog": ["10/11/2021"],  
  "Cola": []  
}
```

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.