

## Barem POO 02.06.2025

**1p** - oficiu (sursa compilează)

**2p** - ierarhie (identificarea claselor + atributelor)

De exemplu:

Produs (nume, gramaj)

| Bautura (pet)

| Desert (tip)

| Burger (ingrediente)

Comanda (id, vector<pair<Produs, int>>)

| se accepta și doar vector<Produs>

Angajat (energie)

| Casier

| Bucatar

| Livrator

Se acceptă și alte soluții cât timp îndeplinesc principiile de bază ale POO.

**0.25p** - const (utilizarea const pentru attribute, funcții sau argumente la funcții)

Ideal, minim într-un loc unde nu este cod generat.

**0.25p** - static (folosirea de funcții și attribute statice)

**0.25p** - destructor virtual

Nu se punctează dacă nu se folosește în toate locurile în care este necesar.

**0.25p** - moștenire (reutilizare de cod)

De exemplu, apelarea unei funcții a clasei de bază în momentul rescrierii ei în clasa derivată.

**0.25p** - abstract (existența a unei clasei abstracte)

**0.25p** - existența unei funcții ce calculează energia necesară pentru un produs.

Se punctează complet dacă este funcție virtuală, rescrisă corespunzător în derivate

**0.25p** - existența unei funcții care calculează energia unei necesare pentru o comandă.

Se punctează doar dacă există o clasă Comanda implementată corespunzător

**0.25p** - folosirea STL containers

De exemplu, vector, list, map, etc.

**0.25p** - folosirea din STL a altor elemente diferite de containers

De exemplu, sort, transform, swap.

**0.25p** - aruncarea unei excepții

Se punctează doar dacă excepția se propagă (nu poate fi tratată în același loc de locul unde a fost aruncată)

**0.25p** - prinderea unei excepții

Tratarea cu sens a unei excepții aruncate

**0.25p** - validarea tipului unui desert

În constructor se validează că tipul unui desert este doar cel prestabilit.

Se punctează și dacă este folosit un enum.

**0.25p** - RTTI

Utilizarea dynamic\_cast sau type\_info

**0.25p** - Existența unei clase Singleton

**0.25p** - Existența unei clase Factory

**1p** - identificarea comportamentului de bază al clasei angajat

Minim, funcții care întorc costul de bază pentru preparare, livrare, comandare, recuperare.

Alte funcții precum cele care verifică dacă o acțiune poate fi executată.

**0.75p** - identificarea comportamentului claselor derivate ale clasei angajat

Rescrierea funcțiilor în clasele derivate ca să fie evidențiat comportamentul specific al derivatelor.

De exemplu, funcția de recuperare trebuie rescrisă în clasa Bucătar.

**0.5p** - asamblarea comportamentului clasei Angajat

Funcții care îmbină comportamentul de bază + verificările necesare ca să creeze clasa Angajat

**0.25p** - ID comandă incrementat automat în constructor

**0.5p** - existența unui meniu cu produse predefinite

**0.5p** - afișarea numărului de angajați de fiecare tip.

Se punctează complet doar dacă este folosit `dynamic_cast` sau `type_info`

**1p** - implementarea ciclului dat ca exemplu

**1p** - prioritizarea unei comenzi

Se presupune că implementarea naivă a ciclului va lucra prima comandă disponibilă dintr-un vector.

Prioritizarea înseamnă existența unei opțiuni în meniu în care pot introduce un ID al unei comenzi care va prioritiza ca o anumită comandă să fie preparată / livrată prima.

**0.5p** - optimizarea ciclului

Se presupune că ciclul iterează naiv prin angajați.

Varianta optimizată va accesa angajații astfel încât să aloce acțiunile specifice către angajații cu o anumită funcție

**1p** - strategia de alegere a comenzilor care sunt preparate

Se presupune că implementarea naivă a ciclului va lucra prima comandă disponibilă dintr-un vector.

Varianta pentru acest subpunct ar trebui să nu mai aleagă prima comandă, ci una conform criteriului ales.

Se punctează maxim dacă se implementează conform Strategy Pattern.

**0.5p** - bonus (pentru folosirea conceptelor POO care nu sunt în barem, precum template)

**-0.5p** - depunctări (de exemplu memory leaks)

**Total: 14p**

**ATENȚIE!** Nu se punctează lucrurile care nu sunt folosite cu sens conform enunțului.  
(de exemplu, aruncarea unei excepții în momentul citirii opțiunilor din meniu)