

# Shopping day

Se apropie Black Friday si toata lumea este in goana dupa cumparaturi. Toti clientii au un set de **trasaturi comune**, precum **numele**, **varsta** si **bugetul** pe care il au disponibil. Clientul poate sa fie un copil, un parinte sau un bunic.

Clientii doresc sa cumpere **produse** caracterizate de **nume**, **pret**, **stoc** si **tip** (tip-ul produsului poate sa fie JUCARIE, ELECTROCASNIC, MANCARE-> Hint: Enum).

Folosind mostenirea si suprascrierea metodelor implementati metoda:

**cumparaProdus(Produs produs)** atat in superclasa cat si in subclase. Fiecare tip de client are unele constrangeri cu privire la cumparaturile pe care le poate face, dupa cum urmeaza:

- In cazul unui client generic, metoda afiseaza un mesaj. (Exemplu: Clientul **detalii\_client** doreste sa cumpere produsul **detalii\_produs**.)
- Un copil cu varsta sub 10 ani poate cumpara jucarii sau mancare daca bugetul copilului nu depaseste pretul produsului si daca produsul se afla in stoc. Copilul cu varsta sub 10 ani nu are voie sa cumpere electrocasnice! Afisati mesaje semnificative.
- Un parinte poate cumpara orice tip de produs atata timp cat acesta nu depaseste 500 RON si bineinteles daca produsul este in stoc.
- Un bunic poate cumpara orice produs atata timp cat bugetul bunicului dupa cumpararea produsului nu scade sub 100 RON.

Clientul poate folosi toate aceste produse pentru a crea o **Comanda** care se caracterizeaza prin: **id-ul comenzii**, **un client (cel care initiaza comanda)**, **un tablou de produse (sir/array)** si **un numar maxim de produse**. Definiti clasa Comanda si implementati urmatoarele metode:

1. **adaugareProdus(Produs produs)** -> adauga un produs in sirul de produse, daca este posibil. Afisati mesaje corespunzatoare. (Exemplu: Clientul **nume\_client** a adaugat produsul **nume\_produs** la comanda.)
2. **stergereProdus(Produs produs)** -> sterge un produs din sirul de produse, daca este posibil. Afisati mesaje corespunzatoare. (Exemplu: Clientul **nume\_client** a sters produsul **nume\_produs** din comanda.)
3. **afisareComanda()** -> afiseaza toate produsele din comanda
4. **totalComanda()** -> afiseaza pretul total al produselor din comanda

## CERINTE:

Cititi si intelegeti povestea.

Implementati o solutie in cod Java dupa cum urmeaza:

1. Implementarea corecta (declarare, attribute, setters + getters, constructor, toString) a claselor principale (Client, Produs, Magazin) (1.5p)
2. Folosirea mostenirii si implementarea corecta a tipurilor de clienti (1.5p)
3. Implementarea corecta a enum-ului (0.5p)
4. Folosirea conceptului de suprascriere a metodelor (1p)
5. Implementarea metodelor de adaugare si stergere a unui produs (in cadrul sirului de produse) (1p)
6. Implementarea metodei de afisare a produselor din comanda (0.25p)
7. Implementarea metodei de calculare al totalului comenzii (0.25p)
8. Simularea unei povesti (intr-o clasa **TestClass** care sa contina **metoda main**) prin: (2p)
  - a. Crearea a minim 3 clienti (minim 1 copil, 1 parinte, 1 bunic) (0.3p)
  - b. Crearea a minim 3 produse (1 din fiecare tip) (0.3p)
  - c. Testarea metodelor de cumparaProdus (0.2p)
  - d. Crearea unei Comenzi care poate contine 5 produse (0.2p)
  - e. Adaugarea a 3 produse in comanda (folosind metoda adaugaProdus) (0.3p)
  - f. Afisarea produselor din comanda (folosind metoda afisareProduse) (0.1p)
  - g. Stergerea unui produs din comanda (folosind metoda stergereProdus) (0.3p)
  - h. Afisarea produselor din comanda pentru a evidentia stergerea (folosind metoda afisareProduse) (0.1p)
  - i. Afisarea totalului comenzii (folosind metoda totalComanda) (0.2p)
9. CLEAN CODE (1p)
  - a. Denumirea sugestiva a variabilelor si a claselor
  - b. Folosirea denumirilor tip camelCase
  - c. Formatarea corespunzatoare a codului

Oficiu: 1p;

Note:

La final, arhivati folderul proiectului java in format : Nume\_Prenume\_Gr30225\_Colocviu1.zip/rar si urcati pe moodle.