

## Laborator 6. Relatii intre clase

### Problema 1. Mini-aplicatia Agenda telefonica

Pas 1. Scrieți o clasă ModelNumereTelefon care să conțină informații despre modul în care sunt specificate numere de telefon. Informațiile sunt implementate în următoarele variabile: codTara, codJudet, nrTelefon sub formă de șiruri de caractere. Clasa conține un constructor fara parametri și un constructor cu parametri ce initializează cu valori explicite variabilele private definite în clasă, precum și metode set/get pentru variabilele instanță. Scrieți metoda convertesteNumarTelefon() care permite conversia unui număr de telefon într-un șir de caractere corespunzător (ex. +0040-241-516532)

Pas 2. Scrieți clasele Abonat care să conțină informațiile despre un abonat: cod abonat, numărul de telefon și Persoana care conține nume și prenume. Implementați un constructor care initializează variabilelor instanță cu valorile transmise prin parametri actuali și metodele accesori corespunzătoare pentru variabilele instanță. Fiecare număr de telefon verifică șablonul descris de obiectele clasei ModelNumereTelefon. Scrieți metoda getInformatii() astfel încât să returneze un șir de caractere cu informații complete despre un abonat. Codul abonatului este un număr întreg unic pentru fiecare abonat în parte.

Pas 3. Scrieți o clasă AgendaTelefonica care conține un tablou de obiecte de tip Abonat și metode ce implementează următoarele operații:

- adăugarea unui abonat în agenda,
- ștergerea unui abonat din agenda (în cazul în care este găsit),
- căutarea numărului de telefon al unui abonat cunoscând codul acestuia (afișați un mesaj corespunzător în cazul în care abonatul nu există în agendă),
- modificarea datelor unui abonat cunoscând codul acestuia (folosind metodele accesori definite în clasa Abonat),
- afișarea datelor despre toți abonații din agenda.

Pas 4. Scrieți o clasă de test TestAgenda care să folosească clasele create la pașii anteriori și în care să creați o agendă telefonică cu un număr maxim n de abonați, n primit ca parametru în linia de comandă. Datele despre fiecare abonat sunt primite de la utilizator prin intermediul unei ferestre ce apare la apelarea metodei statice showInputDialog() a clasei JOptionPane din pachetul javax.swing. Indicație: Metoda se va apela astfel:

```
JOptionPane.showInputDialog(null, "Introduceți date despre un abonat", "Introducere date", JOptionPane.INFORMATION_MESSAGE)
```

iar apelarea ei va returna un șir ce conține mesajul introdus de utilizator.

Afișați apoi într-o fereastră de dialog de același tip cu cea anterioară, un meniu cu opțiuni de tipul:

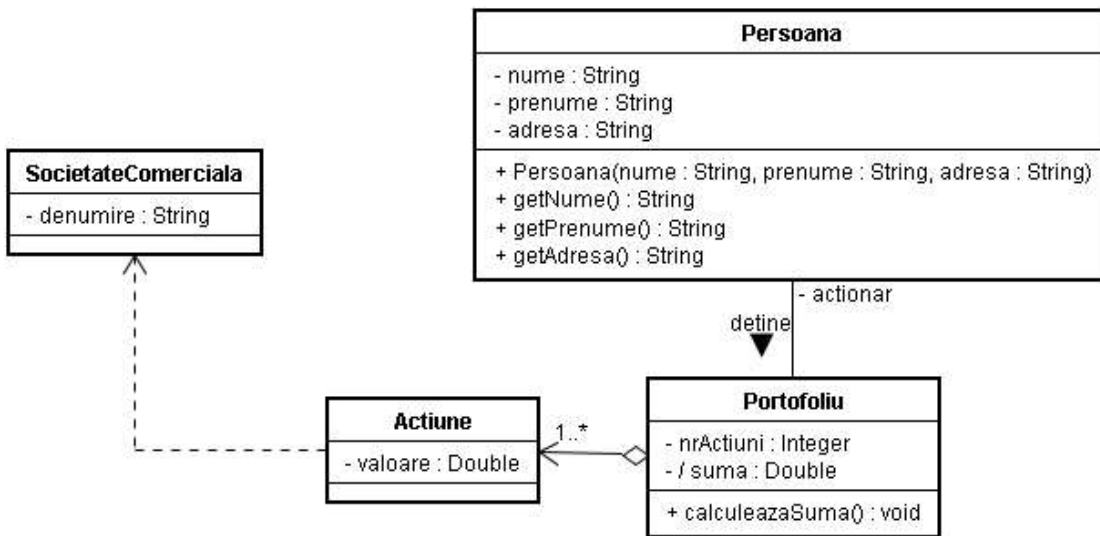
```
Alegeți o opțiune:  
1 - Adaugare abonat  
2 - Stergere abonat  
3 - Modificare abonat  
4 - Afișare agenda
```

În funcție de opțiunea introdusă de utilizator (1, 2, sau 3) veți cere datele necesare pentru:

- 1 – datele necesare pentru crearea unui abonat,
- 2 – codul abonatului
- 3 – codul abonatului, apoi noile date ale abonatului (în cazul în care nu se găsește abonatul respectiv în agendă se va semnaliza cu un mesaj de eroare).

La opțiunea 4 se va afișa conținutul curent al agendei.

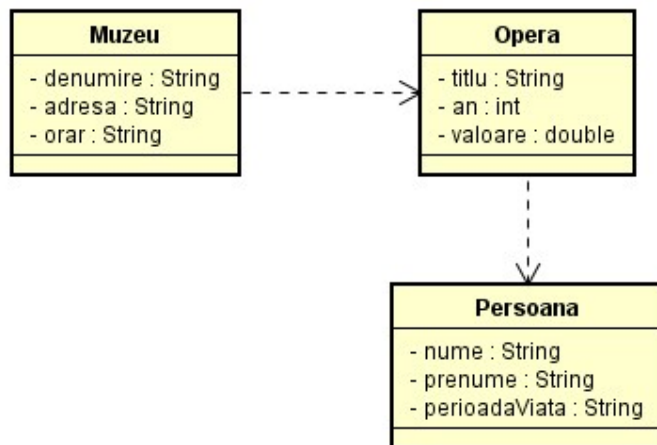
**Problema 2.** O societate comercială pe acțiuni vinde un număr oarecare de acțiuni. Fiecare acțiune are o valoare. Un acționar deține un portofoliu format dintr-un număr de acțiuni. Diagrama UML de clasă a acestei probleme este prezentată în continuare.



Să se scrie un program care calculează și afișează suma pe care o deține fiecare acționar, împreună cu acțiunile sale.

**Problema 3.** Intr-un muzeu sunt prezentate o serie de opere de artă. Despre muzeu știm următoarele caracteristici: denumire, adresă și orarul în care este deschis în sezon și în extrasezon. Fiecare operă este descrisă prin următoarele atribute: titlu, anul terminării operei și valoare. În plus, fiecare operă este realizată de un autor despre care cunoaștem numele, prenumele și perioada în care a trăit.

Diagrama UML incompletă a acestui program este prezentată în următoarea figură:



Să se scrie un program care are următoarele funcționalități:

- adaugă o operă nouă,
- afișează date despre toate operele aflate în muzeu,
- calculează și afișează valoarea medie a tuturor operelor dintr-un anumit an,
- afișează date (titlu și autor) despre cea mai veche operă aflată în muzeu la ora actuală.