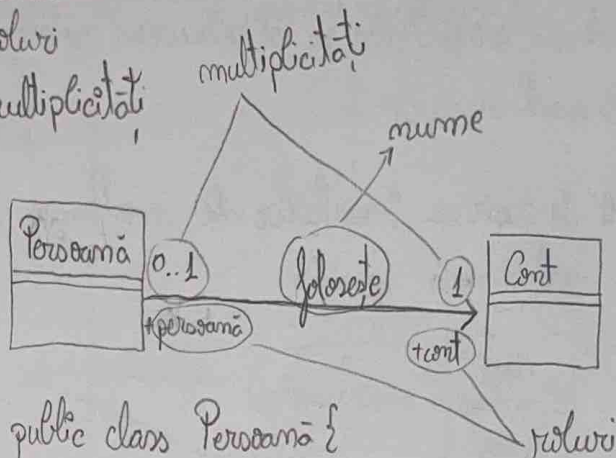


A1. Care sunt prop. unei relații de asociere între clase și cum se reflectă acestea în codul sursă scris pe baza unei diagrame de clase? Dați exemple (model vs. cod)

- nume
- roluri
- multiplicități



unidirecțională, one to one

```
public class Persoană {
    private Cont cont;
    public Persoană()
    {
        cont = new Cont();
    }
    public Cont getCont()
    {
        return cont;
    }
}
```

- Numele de roluri trebuie să fie ca în diagramă
- multiplicități:

1: avem o referință simplă la cealaltă clasă (clasa asociată)
 0...*: avem un set de referințe la cealaltă clasă (clasa asociată)

A2: Descrie șablonul structural Composite

Se folosește pentru a itera peste toate componentele:

component: interfața pt. frunze și composite

frunze: cu implementare specifică

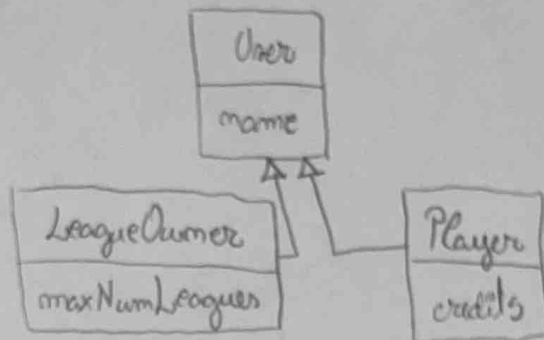
composite: pe metoda suprascrisă din interfață se iterează peste toate componentele

A3: Care este esența abordării Design by Contract referitor la asigurarea fiabilității sistemului?
 fiabilitate = îndeplinirea funcțiilor în mod consistent, fără erori sau defecte

Există:

- precondiții } referitor la funcții
- postcondiții }
- invarianta de clasă : cum ci treb. să se păstreze pe tot parcursul vieții unei clase
- asocieri: pt. a verifica că cele 3 sunt îndeplinite

A4: Ilustrare, folosind exemple, modalitățile de repr. a ierarhiilor de mentenire din modelul obiectual la nivelul bazelor de date relaționale.



→ verticală

User table

id	nume	role
56	Joe	League Owner
79	John	Player

LeagueOwner table

id	maxNumLeagues
56	12

Player table

id	credits
79	126

→ orizontală

LeagueOwner table

id	nume	maxNumLeagues
56	Joe	12

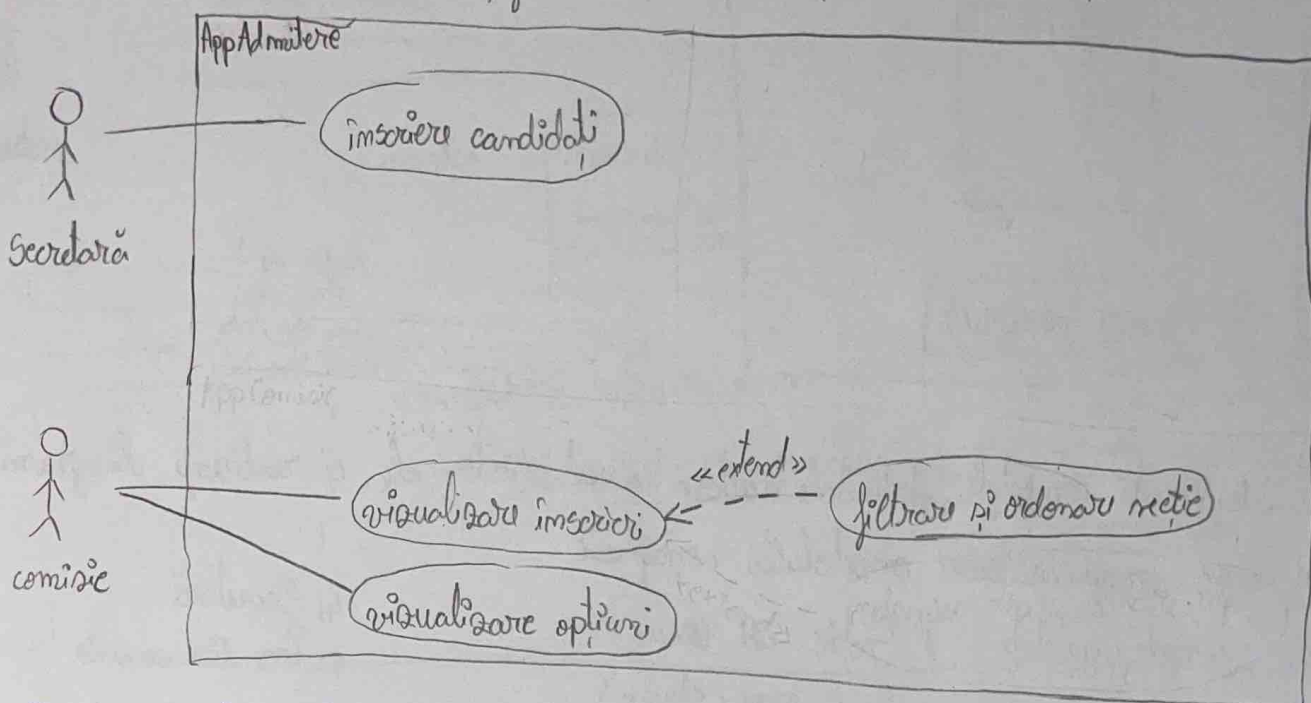
Player table

id	nume	credits
79	John	126

1: Identificați actorii, CU aflate aceluiași și trasați diagrama CU

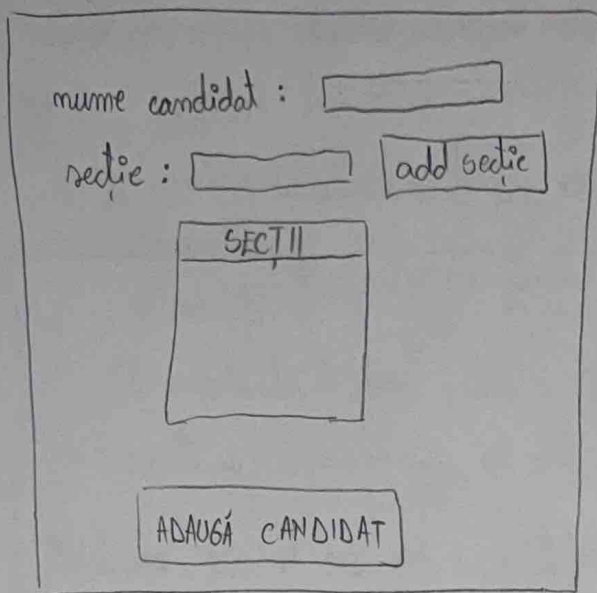
- secretară: înscrierea candidați

- comisie: vizualizare înscrieri, filtrare și ordonare reție, vizualizare opțiuni



B2: Descrie cazul de utilizare înscrierea candidat și realizați un prototip al interfeței grafice asociate.

Nume	Înscrierea Candidat
Actori	Secretară
Precondiții	Aplicația rulează.
Postcondiții	Candidatul este introdus în sistem (BD).
Flux de even.	<ol style="list-style-type: none"> 1. Secretara deschide aplicația. 2. Secretara introduce numele candidatului și seculă p. care a optat. 3. Secretara optează p. salvarea datelor.
Trigger	Secretara primește un dosar de înscriere.
Exceptions	<p>E1: date insuficiente</p> <ol style="list-style-type: none"> 1. Nu au fost introduse suficiente date și s-a optat p. salvarea lor 2. Se afișează un mesaj de alertă și secretara are posibilitatea să introducă date noi. <p>E2: date invalide</p>



B3: Identificați conceptele, atributele acestora și rel. dintre ele și realizați diagrama de clase corespunzătoare modelului conceptual.

concepte = entități :

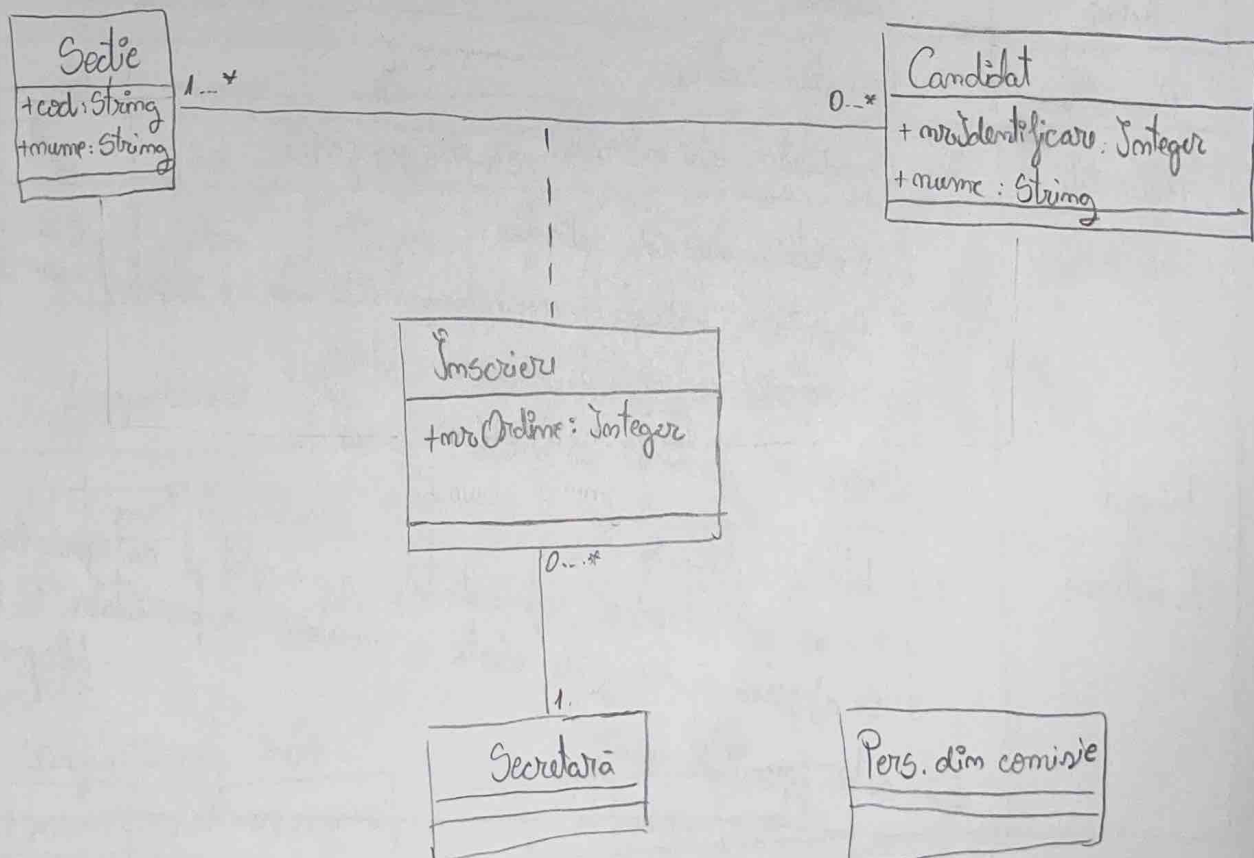
1. Secție: $\text{cod}(\text{String})$
 $\text{nume}(\text{String})$

2. Candidat: $\text{nr_Identificare}(\text{Integer})$
 $\text{nume}(\text{String})$

3. Înscriere: $\text{nr_Ordine}(\text{Integer})$

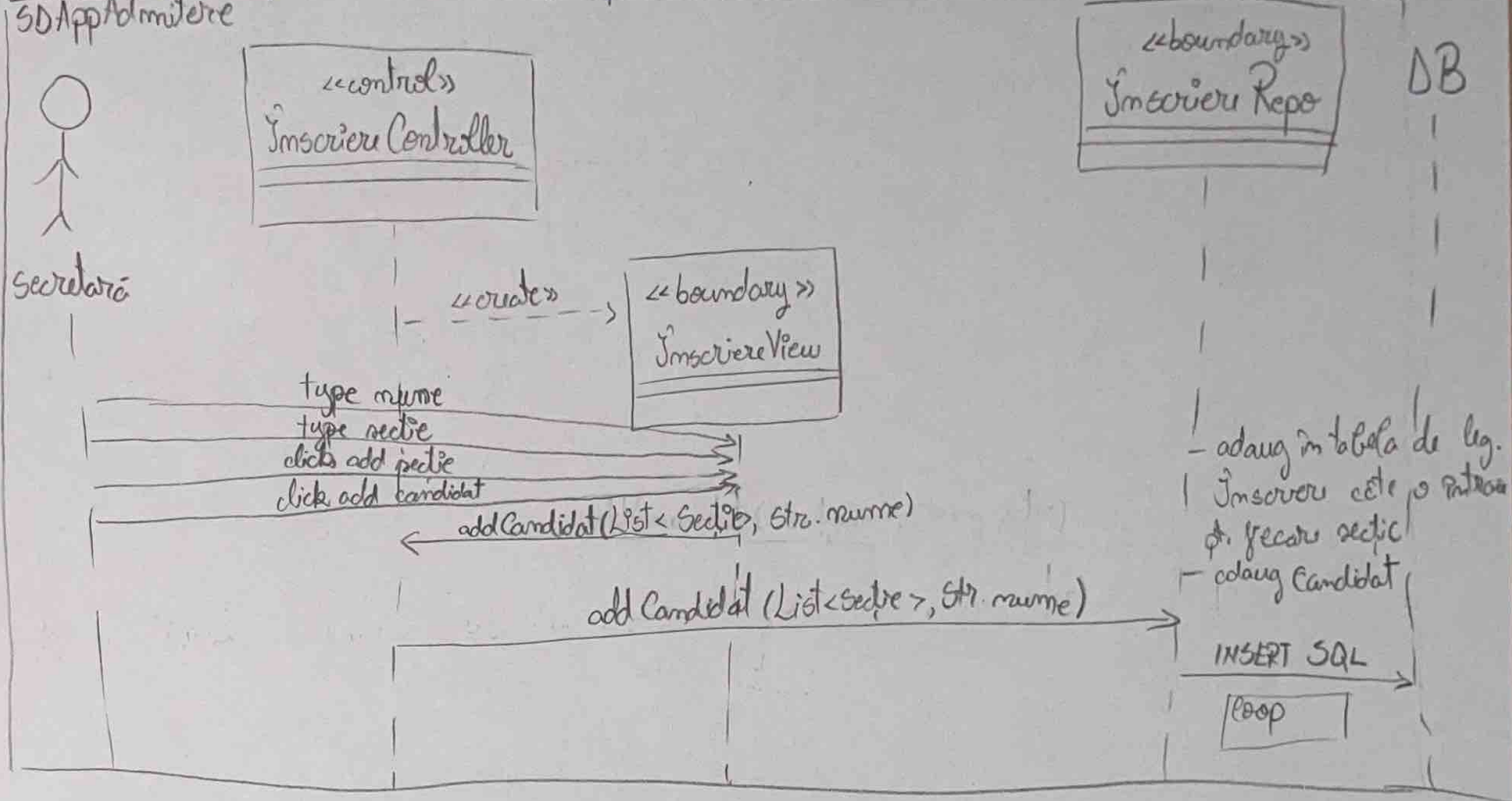
4. Secretară

5. Pers. din comisie



4: Trasati diagrama de secvență pt. un scenariu normal al CU înscrieri Candidat

SDAppAdminter



B5: Trasati diagrama detaliată de clase, care să includă și toate clasele identificate la punctul B4. Aceasta diagramă va trebui să includă atât atributele și metodele claselor, cât și relațiile dintre aceste clase.

