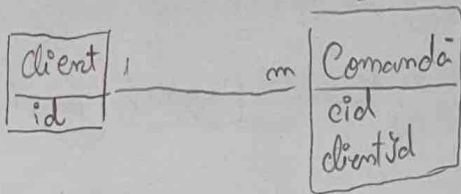


1. Explicați principiul solid Open Closed.

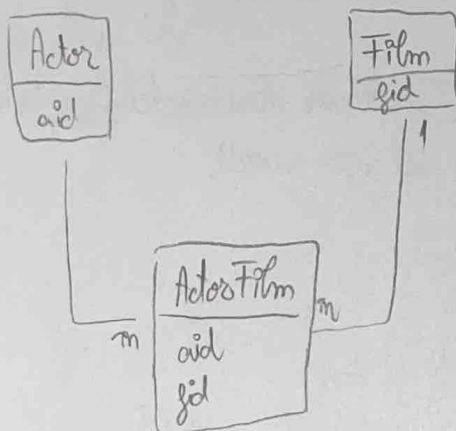
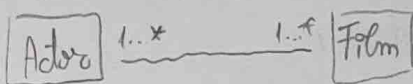
Este deschis p. extindere și închis p. modificare.

2. Ilustrați modalitatea de repr. a asocierilor din modelul obiectual la niv. BD relațional.

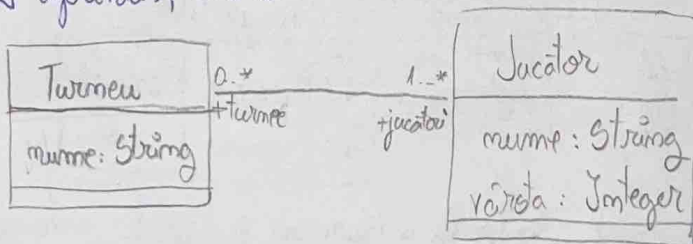
* one-to-one sau one-to-many



* many-to-many



3. Dați un ex. de invariant la nivelul unei diagrame UML de clase (îmbaj natural și expresie och. apruntă), care să utilizeze unul dintre iteratorii och. pe colecții.



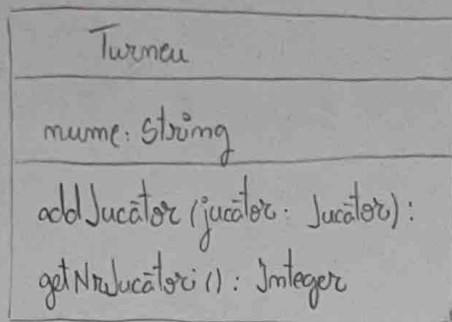
context Turneu:

inv. varstaMajora:

self.jucatori -> forAll (jucator : jucator | jucator.varsta >= 18)

2

3. Folosind UML și OCL, dați un exemplu de precondiție care să utilizeze unul dintre iteratorii pe colecții. Fac și postcondiție

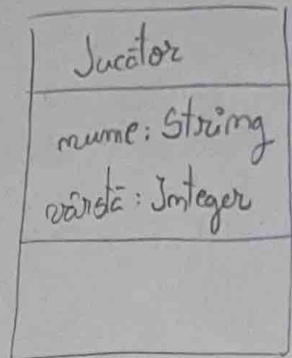


0..*

+turnee

1..*

+jucatori



precondiția: să nu existe deja acel jucator în turneu

postcondiția: să crească nr. de jucatori din turneu cu 1

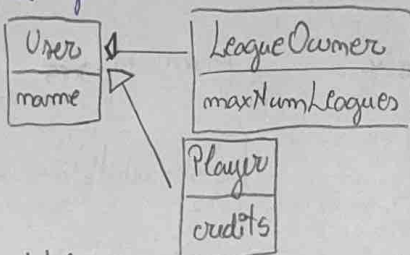
context Turneu :: addJucator (jucator: Jucator)

pre: not self.jucatori → exists (j: Jucator | j = jucator)

post: self@pre.getNumJucatori() = self.getNumJucatori() + 1

2. Ilustrați modalitatea de reprezentarea ierarhiilor de menținere din modelul direct al la nivelul BD relaționale (folosind un model diferit de cel din curs).

în modelul direct:



→ verticală:

User table

id	nume	role
1	nume1	League Owner
2	nume2	Player

League Owner table

id	max Num Owner
1	100

Player table

id	credits
2	200

→ orizontală

League Owner table

id	nume	max Num Owner
1	nume1	100

Player table

id	nume	credits
2	nume2	200

1. Care este esența Design by Contract referitor la asigurarea fiabilității sistemelor soft?

Există: - precondiții referitor la funcții

- postcondiții

- invarianta de clasă: condiții care trebuie să se păstreze pe tot parcursul vieții unei clase

- aserțiuni: pt. a ne asigura că cele 3 sunt mereu îndeplinite

Așa este asigurată fiabilitatea: îndeplinirea funcțiilor în mod consistent, fără erori sau defecte

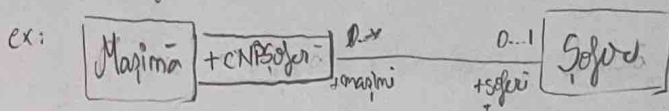
1. Explicati, folosind un exemplu, principiul SOLID al separării interfețelor.

Acest principiu se referă la faptul că interfețele ar trebui să fie mici și specializate pentru a nu forța clientii să depindă de interfețe cu metode pe care nu le folosesc.

ex: Procesarea Plății. În loc să fie o interfață mare cu toate modalitățile de plată, putem grupa în două: Plăți Online și Plăți Offline.

2. Ilustrați, folosind un exemplu (diferit de cel din curs), modalitatea de traducere în cod a unei asocieri calificate.

asocieri calificate: reduce multiplicitatea

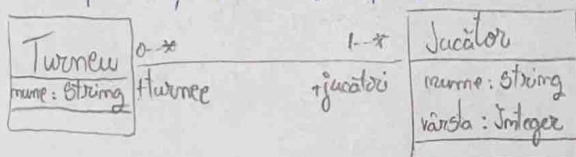


~~public class Marimă {
public IDictionary<String, Sofer> soferi {get; private set;} = new Dictionary<String, Sofer>();~~

~~public class Sofer {
public ICollection<Marimă> marimi {get; private set;} = new HashSet<Marimă>();~~

NU

3. Definiți un invariant la nivelul unei diagrame UML de clase, care să utilizeze unul dintre iteratorii pe colecții și să specifice corectitudinea acesteia.



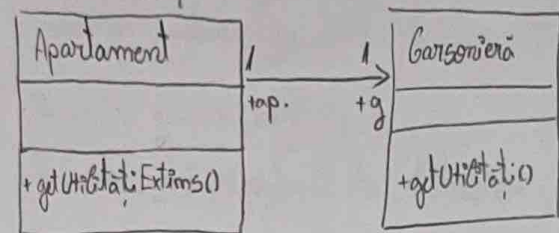
context Turneu

inv varsta Majoră:

self.jucatori → for All (jucator: Jucator / jucator.varsta >= 18)

1. Explicați conceptul de delegare și exemplificați.

• alternativă la moștenire, implementarea unei funcționalități prin reutilizare, trimite mesaj la obiect pe care o delegă, păstrând unidirecțională (reprezintă spre obiecte delegate)
ex: unidirecțional one-to-one



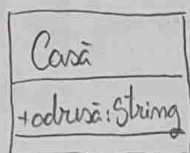
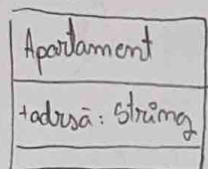
```

public class Apartment {
    public Garsoniera g { get; private set; }
    public getUtilitatiExtins() {
        Console.WriteLine(g.getUtilitati() + "...");
    }
}

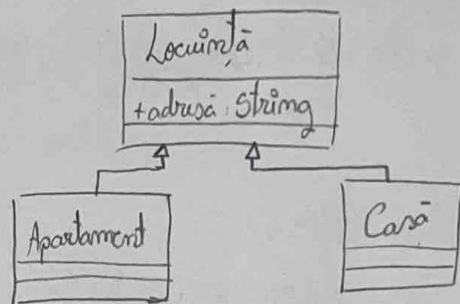
public class Garsoniera {
    public getUtilitati() { Console.WriteLine("..."); }
}
  
```

2. Dați exemple de transformare la nivelul modelului (model-to-model)

transformare clase în atribute:



se transf. în



4

1. Justificați necesitatea utilizării limbajului oc în modelare.

pentru definirea constrângerilor de integritate, complementare diagramelor UML

2. Care este semnificația dreptunghiurilor de activare din SD și ce regulă trebuie respectată privind dimensiunea acestora?

- activarea = execuția unei metode

rup. prin dreptunghiuri verticale

- dimensiunea corespunde intervalului de timp în care se execută metoda apelată mesajului

- revenirea dintr-un apel ar trebui să fie mai devreme de încheierea apelului care l-a apelat

