BILET 1

A. 1. Explicați diferența dintre domeniul problemei și domeniul soluției.

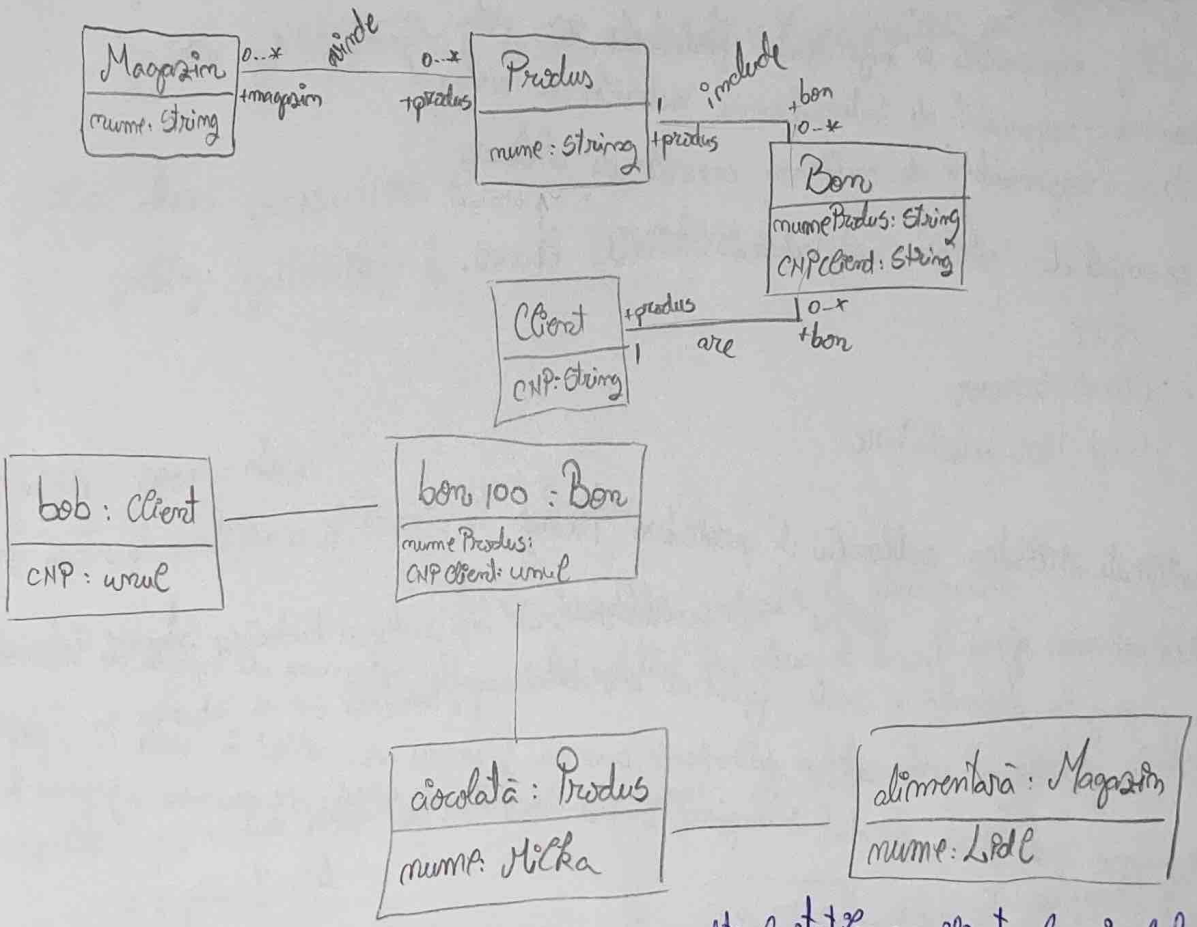
domeniul prob. = repr. sistemului pt. utilizatori; răsp. la întrebarea ce?
nu conține detalii de implementare; cerințe utilizator

domeniul soluție = repr. sistemul pt. programatori; răsp. la întrebarea cum?
introduc concepte neapărate clientului
arhitectură, modularitate, algoritmi

2. Definiți conceptul de scenariu.

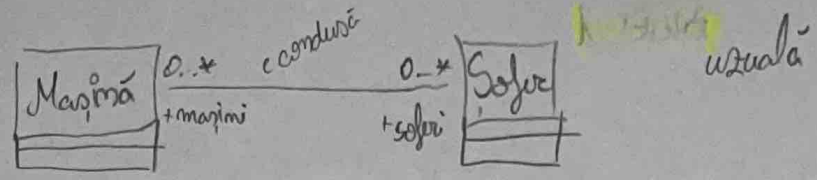
Un scenariu este o instanță a unui caz de utilizare, ce descrie o secvență concretă de acțiuni/evenimente. Sunt focusate pe inteligibilitate, concret.

3. Dați un exemplu de diagramă obiect corespunzătoare unei diagrame de clase.

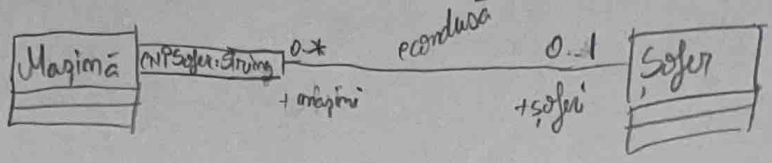


4. Explicați modul în care numele de roluri și multiplicitățile specificate la nivelul diagramei de clase influențează codul sursă aferent acelei diagrame.

numele de roluri corespund exact numelor de atribute din clase
multiplicități: one: referință simplă spre o altă clasă
many: avem o colecție/dicționar spre o altă clasă
↓ asocieri uzuale
→ asocieri calificate



uzuală




```
public class Mașină {
    public IDictionary<String, Șofer> șoferi {get; private set;} = new
                                        Dictionary<String, Șofer>(); }

public class Șofer {
    public ICollection<Mașină> mașini {get; private set;} = new HashSet<Mașină>(); }
```

5. Explicați semnificația stereotipurilor <<entity>>, <<boundary>>, <<control>>

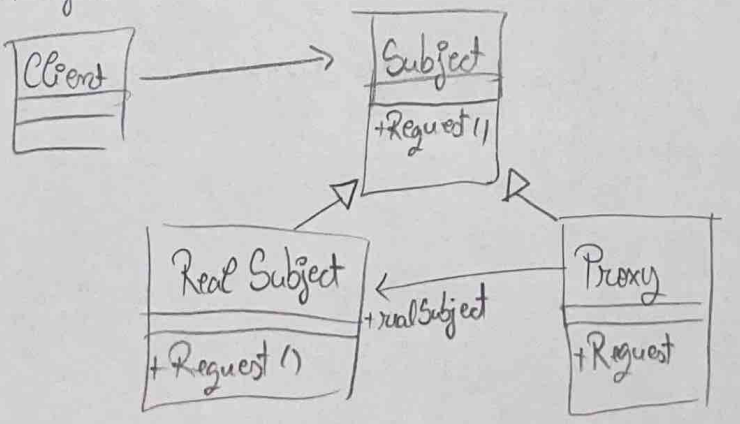
entity: responsabil de informația persistentă din sistem
boundary: responsabil de interacțiunea actorilor cu sistemul
control: responsabil de realizarea cazurilor de utilizare

6. 3 exemple de stiluri arhitecturale.

MVC
Client-Server
three-tier architecture

C. 1. Justificați utilitatea șablonului de proiectare Proxy

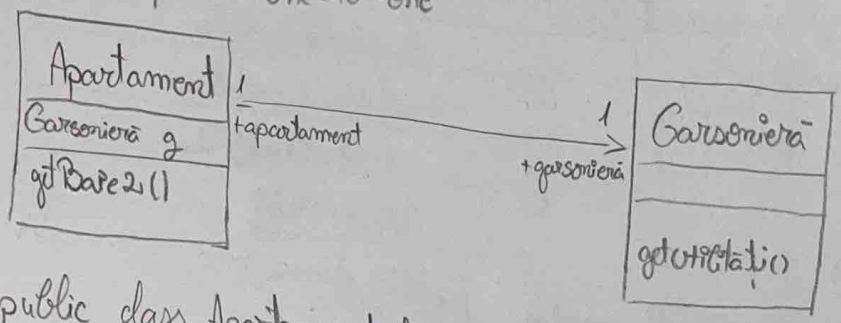
- introduce un grad de "ocolire" la accesul unui obiect
- un proxy la distanță ascunde faptul că un obiect se află într-un alt spațiu de adresă
- un proxy virtual poate efectua optimizări, cum ar fi crearea unui obiect la cerere (un proxy pt. o imagine instanțiază obiectul imagine concret doar atunci când este necesară afișarea imaginii)



2. Definiți conceptul de delegare

- este o alternativă la moștenirea clasică
- implementarea unei funcționalități prin reutilizare
- doar trimite mesajul la clasa pe care o deleagă
- legătură unidirecțională (referință spre obiectul delegat)

ex: unidirecțional one-to-one




```
public class Apartament {
    public Garsonieră garsonieră { get; private set; }
    public getUtilitățiExtins { Console.WriteLine(garsonieră.getUtilități + " -- "); }
}
public class Garsonieră {
    public getUtilități { Console.WriteLine(" -- "); }
}
```

3. SOLID open-close

entitățile sunt închise pt. modificare și deschise pt. extindere

4. Enunțați regula privind moștenirea contractelor referitor la precondiții?

Clientul se așteaptă ca un contract formulat relativ la clasa de bază să fie respectat și de clasele derivate.

precondiții: unei metode dintr-o subclasă îi e permis să slăbească precondiția metodei pe care o supradefinește

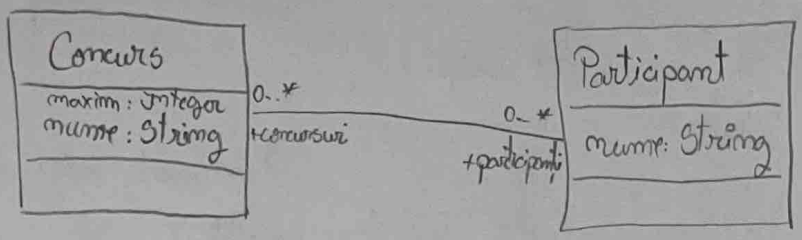
postcondiții: o metodă care supradefinește trebuie să asigure o postcondiție cel puțin la fel de puternică precum cea supradefinită

invarianți: o subclasă trb. să respecte toți invarianții superclasei; eventual poate introduce alții noi.

5. Design by Contract - principiul non-redundanței

În corpul funcției nu ar trebui să fie verificate precondiții (deja să fim siguri că s-a)

6. invariant



context Concurs:

inv maxParticipanți:

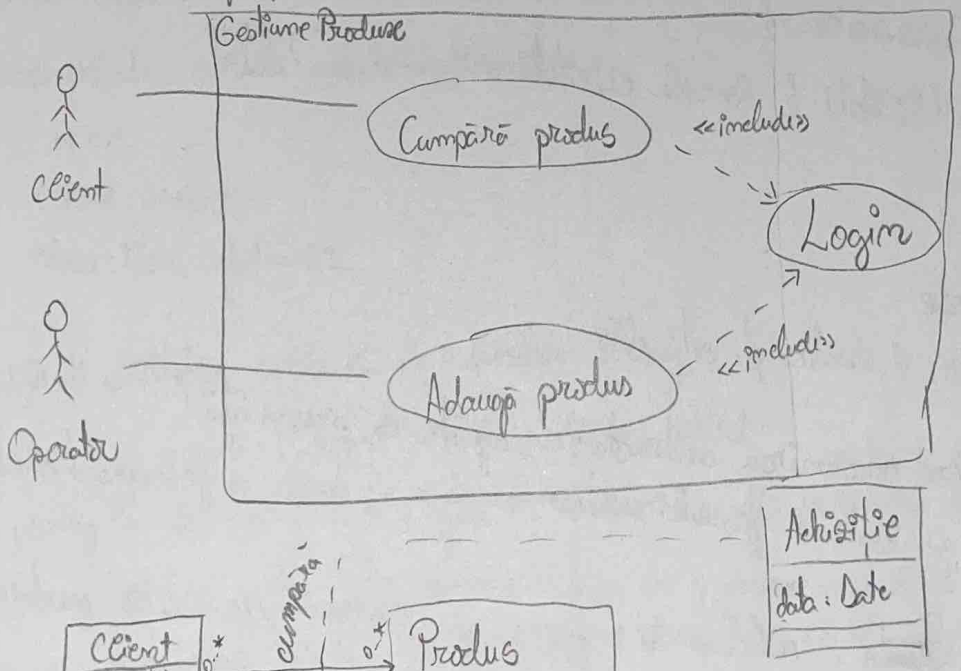
self.participanți → size <= self.maxim

B. 1. actori: Client, Operator

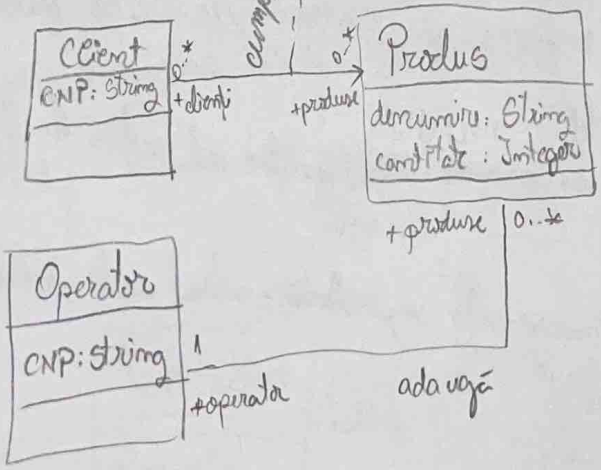
CU: - cumpărare produs

- adaugă produs

2.

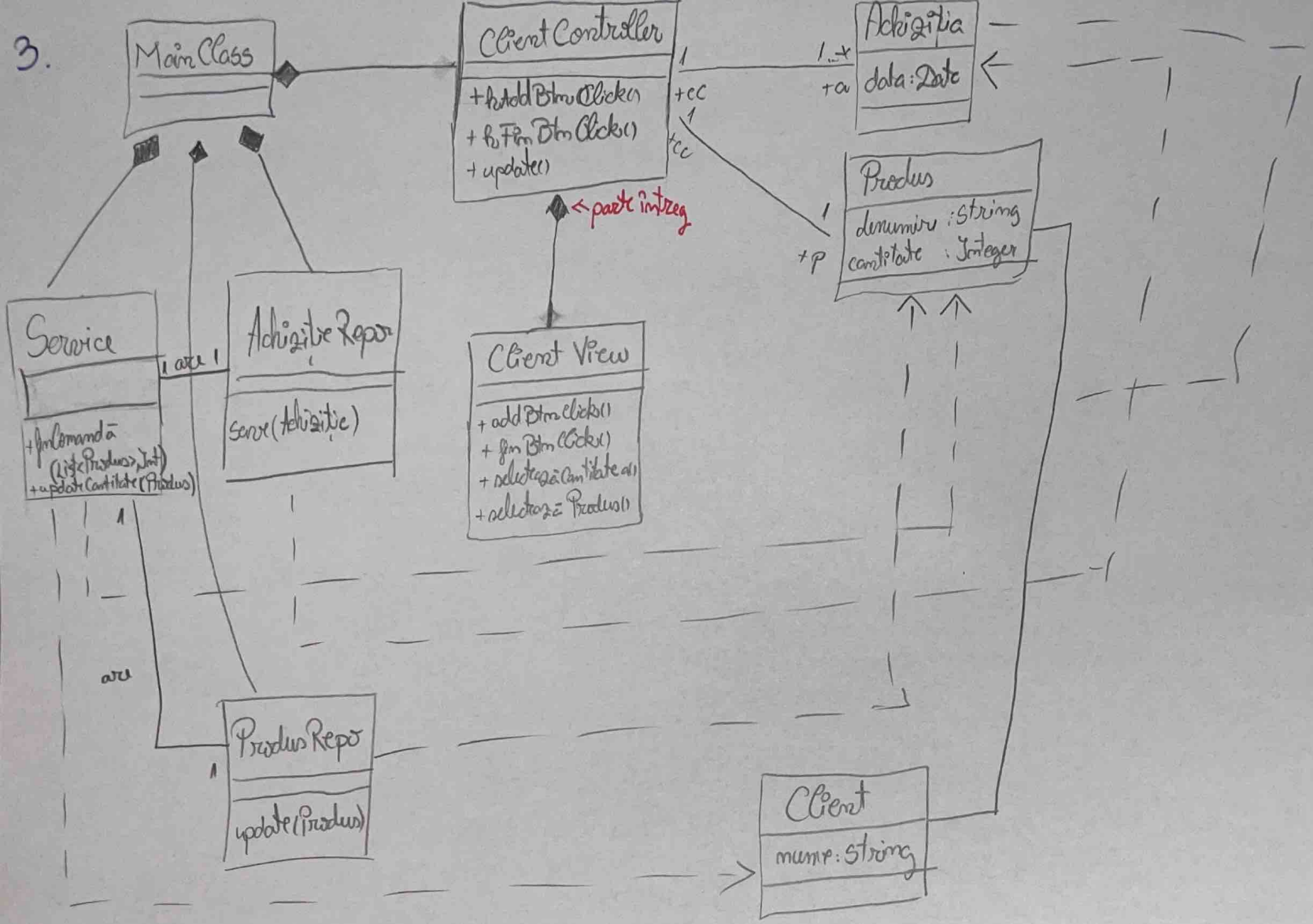


3, 4



Client (CNP)
Operator (CNP)
Produs (--)
} Achiziție (data)

3.


MainClass
ClientController
+handleAddBtnClicks()
+handleFinBtnClicks()
+update()
Achizitia
data:Date
1..*
+a
1
+cc
Produs
denumire:String
cantitate:Integer
+p
<parte intreg
Service
+implementare(List<Produs>,int)
+updateCantitate(Produs)
are
AchizitieRepo
save(Achizitie)
ClientView
+addBtnClicks()
+finBtnClicks()
+selectareCantitate()
+selectareProdus()
are
ProdusRepo
update(Produs)
Client
nume:String