

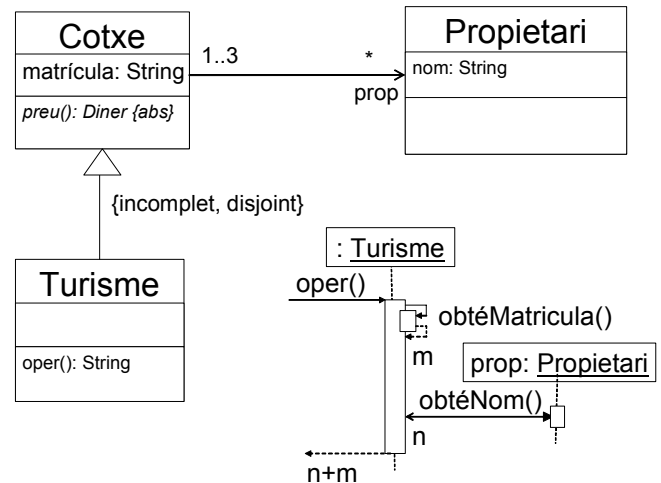
Cognoms i Nom: _____

DNI: _____

TEORIA – S’ha de contestar amb aquest mateix full.

[0,9 punts] Citeu els tres errors dels següents diagrames explicant BREUMENT que hi ha malament en cada un d’ells.

- 1) l’operació `Cotxe::preu()` no pot ser abstracte ja que la classe pot tenir instàncies.
- 2) La crida a l’operació `obtéNom()` des d’`oper` crea un acoblament innecessari.
- 3) El rol `prop` és de cardinalitat `*` per la qual cosa no pot definir una única instància al diagrama de seqüència.



[0,3 punts] Cita 2 factors de qualitat externs del catàleg de Meyer amb sinergia positiva i d’altres dos amb conflicte. No podeu fer servir ni la Facilitat d’us ni l’Economia ni repetir cap factor.

Cal contestar 4 de la llista de la transparència del tema 2: Fiabilitat (correctesa + robustesa), Funcionalitat, Modularitat (extensibilitat + reusabilitat), Puntualitat, Compatibilitat, Verificabilitat, Eficiència, Integritat, Portabilitat, Reparabilitat. **Noteu** que no hi és la canviabilitat.

[0,5 punts] En relació al principi Obert-Tancat.

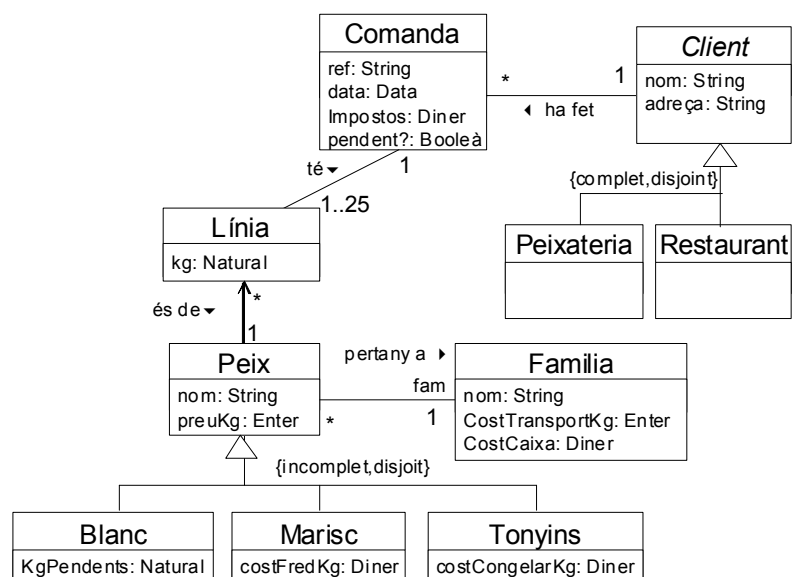
Els mòduls haurien de ser: oberts per a l’extensió i tancats per la modificació.

El comportament dels mòduls que satisfan aquest principi es canvia afegint nou codi, i no pas canviant codi existent

L’ús correcte del polimorfisme afavoreix aquest principi.

[0,3 punts] En el model “*empenta*” de comunicació entre dues capes adjacents, ¿cóm es comunica la informació?

Els paràmetres porten la informació que la classe responsable de l’operació necessita de la classe que realitza la crida



PROBLEMA.

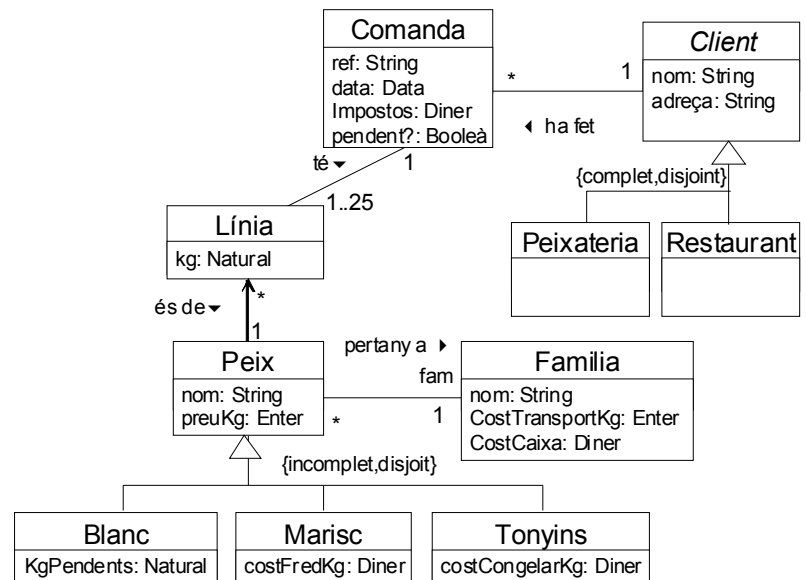
A la figura de la dreta es mostra el diagrama de classes, ja normalitzat, d'un sistema d'informació per gestionar comandes d'un magatzem de peix.

RI textuais:

RT1: Claus de les classes: (Client, nom), (Comanda, ref), (Familia, nom), (Peix, nom).

RT2: No poden haver dues línies de la mateixa Comanda i Peix

La resta de RI no tenen interès aquí.



El **preu total de un Kg de Peix** es calcula com:

$$ptKg = preuKg + (\text{CostTransportKg de la família a la qual pertany el Peix}) + \text{costEspecial}.$$

El cost especial només s'aplica a les especialitats de peix segons el següent criteri:

- **Blanc:** costEspecial = costCaixa de la Família a la qual pertany el Peix.
- **Marisc:** costEspecial = (costCaixa de la Família a la qual pertany el Peix) + costFredKg
- **Tonyins:** costEspecial = costCongelarKg.

Sobre aquest model es necessiten tres operacions:

context Comanda::cost(): Diner

post *costComanda*: el cost de la comanda calculat com la suma dels costos de totes les línies de la comanda més l'atribut *impostos* de la comanda *self*. El cost de cada línia es calcula multiplicant l'atribut *kg* de la Línia per el *ptKg* del Peix associat a la línia.

context Restaurant::costPendent(): Diner

post *cost Pendent*: suma del cost de totes les comandes pendents del Restaurant *self*.

context Comanda::afegirLiniea(nouPeix: Peix, pKg: natural)

pre: *existeixPeix*: *nouPeix* correspon a un Peix del sistema.

exc *senseKgs*: el paràmetre *pKg* és menor que 1.

exc *misteri*: excepció que heu de descobrir analitzant els invariants derivats del diagrama de classes.

post *altaLinia*: Es crea la instància de Línia amb *kg=pKg*.

post *creaEsDe*: Es crea l'associació *ésDe* entre la nova Línia i el Mòbil *nouPeix*.

post *creaTé*: Es crea l'associació *té* entre la nova Línia i la comanda *self*.

post *incrementaKg*: si el peix es Blanc es suma *pKg* a l'atribut *kgPendants*.

Suposant que hi ha navegabilitat entre Peix i Línia, es demana:

- [4 punts] Dissenyar completament l'operació *cost()*.
- [1 punt] Dissenyar completament l'operació *costPendent()*.
- [0,5 punts] Feu una proposta per a la declaració de l'excepció *misteri*.
- [2,0 punts] Dissenyar completament l'operació *afegirLiniea()*.
- [0,5 punts] Justifiqueu les navegabilitats i acoblaments resultats de l'enunciat i del vostre disseny i inclogueu-les dins del diagrama repetit al full de teoria.

No cal que assigneu cap operació a les classes en el diagrama de classes.

Solució.

Navegabilitats:

Operació cost():

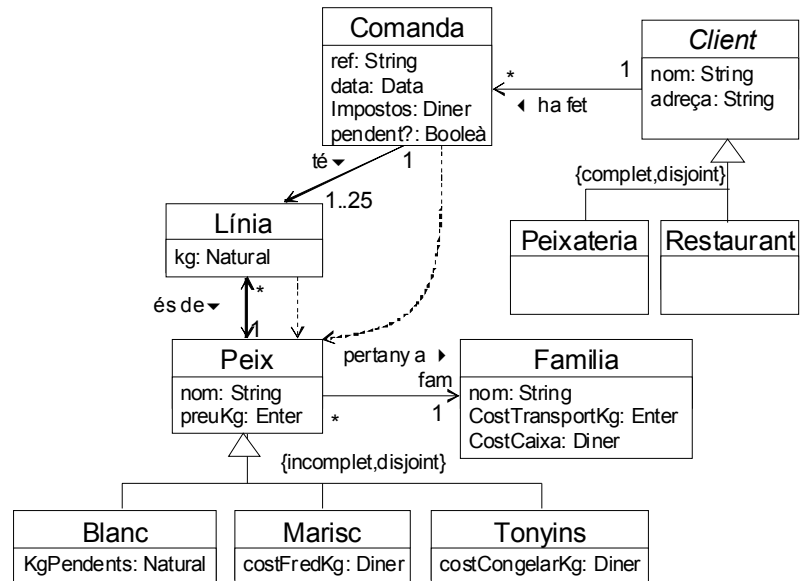
- Comanda → Línia
- Línia → Peix
- Peix → Família (per obtenir el costos).

Operació costPendent():

- Client → Comanda
- Reutilitza Comanda::cost()

Operació afegirLínia():

- No aporta noves navegabilitats



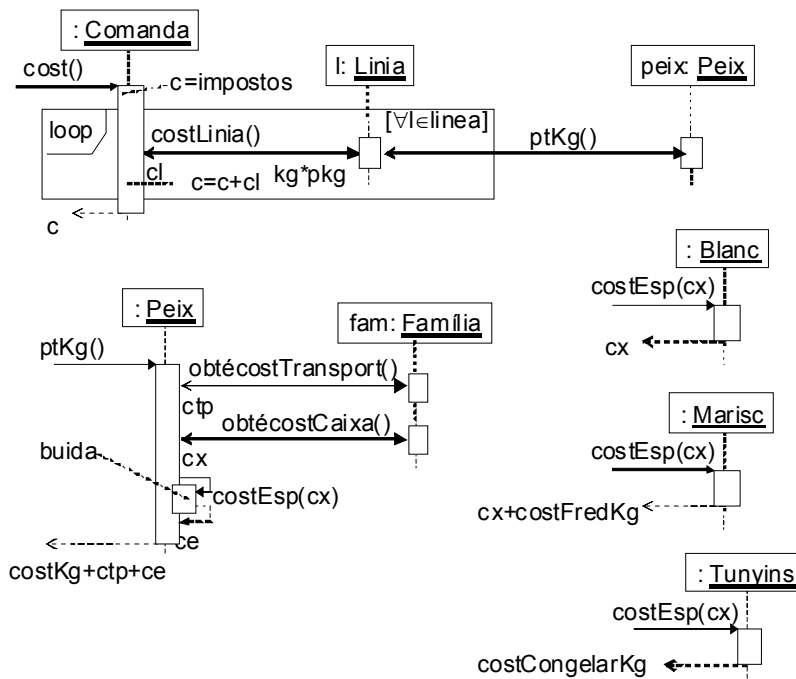
Acoblaments:

Operació afegirComanda():

- Comanda → Peix (Comanda rep un paràmetre de tipus Peix).
- Línia → Peix (Línia rep un paràmetre de tipus Peix).

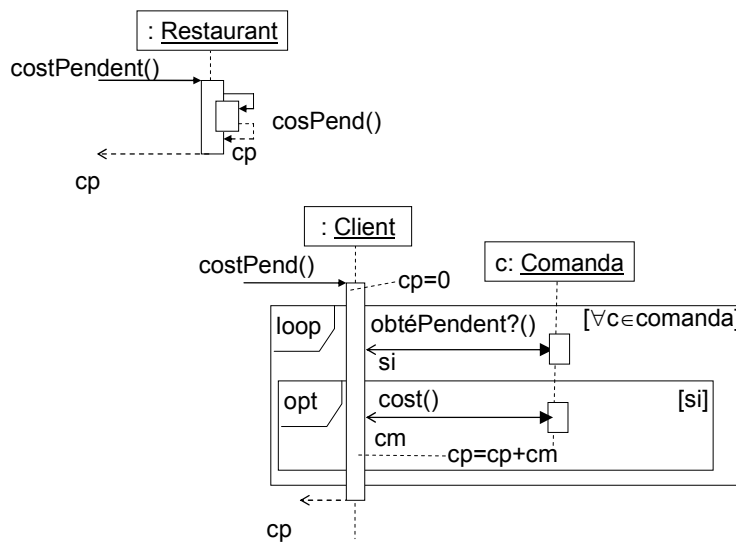
a) Disseny de l'operació Comanda::cost():

Els diagrames de seqüència són:



b) Disseny de l'operació Restaurant::costPendent():

Els diagrames de seqüència són:



- c) **exc misteri1**: La comanda ja té 25 línies (restricció gràfica 1..25).
exc misteri2: Ja existeix una línia de la mateixa comanda i Peix (RT2)

d) Disseny de l'operació afegirLínia().

Els diagrames de seqüència són:

