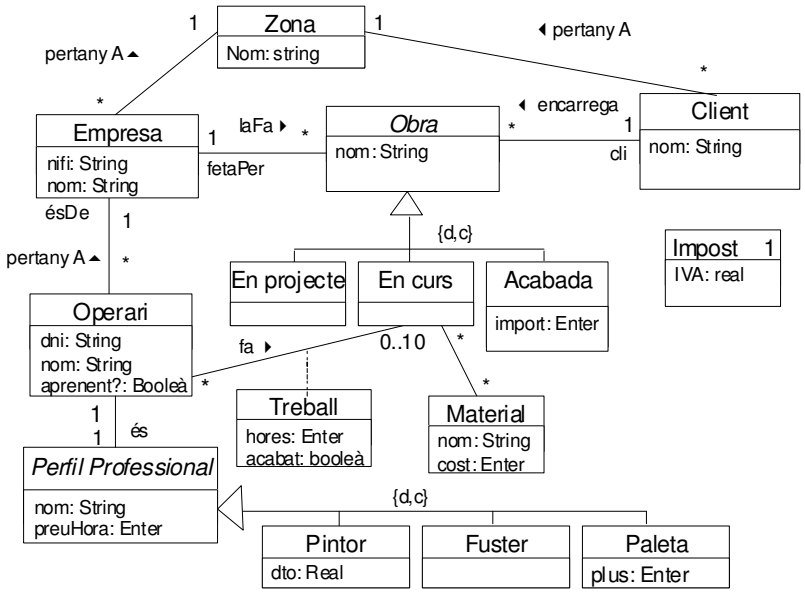
Curs 2008-09 - Tercer acte avaluatori ES2 - Problema **6-maig-09**

Entregueu en el munt del vostre professor
Resolgueu les dues preguntes de teoria en una única fulla

A la figura de la dreta es mostra el diagrama de classes, d'un sistema d'informació per gestionar un grup d'empreses que fan obres de construcció. Les empreses tenen operaris amb perfils professionals de tres tipus: pintor, fuster i paleta. En una obra poden intervenir operaris de diversos perfils i empreses. Per a cada una de les obres i els operaris que intervenen s'anoten les hores de cada tipus treballades.



RI textuals:

RT1: Claus de les classes: (Empresa, nif), (Obra, nom), (Operari, dni), (Client, nom), (Zona,nom), (Perfil Professional, nom), (Material, nom)

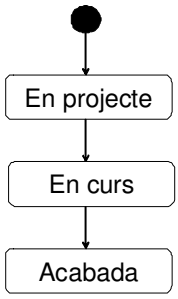
.... La resta de RI no tenen interès aquí.

El cost d'una obra En Curs es calcula sumant l'import de les hores treballades per cada operari assignat a l'obra amb el cost del material utilitzat en l'obra i multiplicant el resultat per (1+Impost.IVA)

L'import de les hores treballades per cada operari es calcula així:

- **Paleta**: (hores empleades per l'operari Paleta*preuHora) + plus
- **Pintor**: (hores empleades per l'operari Pintor*preuHora) – [dto] : el descompte dto només s'aplica si el pintor és aprenent.
- **Fuster**: hores empleades per l'operari Fuster*preuHora

L'atribut Import de la classe Acabada guarda el cost de l'obra en curs quan s'ha acabada i facturada. El cost d'una obra En Projecte és cero.

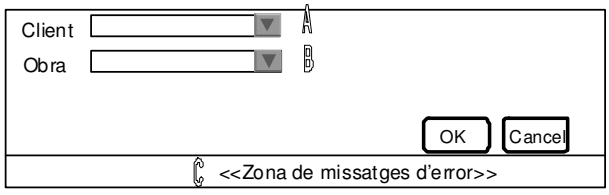


El sistema controla també l'estat de l'obra segons el diagrama d'estats que es mostra a la dreta

Sobre aquest model s'especifiquen dos casos d´ús:

- *Facturació*, per acabar una obra.
- *Llista de operaris*, per obtenir una relació d'operaris que intervenen en obres grans.

Després d'assignar les responsabilitats a les capes s'han trobat, entre d'altres, les següents operacions:



Cas d'us *Llista d'operaris*

context CapaDeDomini::operarisGransObres(): Set (String)
post resultat = Retorna el nom de tots els operaris que intervenen en obres En Curs de cost major que 12000€ i on l'empresa i el client de l'obra són de la mateixa zona

- El sistema mostra la llista de clients en A.
- L'usuari escull un client de la llista A.
- El sistema mostra les obres del client en la llista B
- L'usuari escull una obra de la llista B.
- L'usuari prem OK.
- El sistema passa l'obra d'en curs a acabada i la factura.
- Si hi ha error, el sistema el mostra en el quadre C
- El qualsevol moment, L'usuari pot cancel·lar el cas d'ús prement Cancel.

Cas d'us *Facturació*

Interfície externa: es mostra a la dreta

context CapaDeDomini::obtéObresClient(nomC: String): Set(String)
pre: *existeixClient*: Existeix el client de nom *nomC*.
exc: *clientSenseObres*: el client *nomC* no té obres enregistrades al sistema.
post: *obtéObres*: result = Noms de les obres del client *nomC*.

context CapaDeDomini::acabar(nomO: String)
pre: *existeixObra*: Existeix una obra de nom *nomO*
exc: *noEnCurs*: *nomO* no correspon a una obra en curs.
exc: *téTreballNoAcabats:* l'obra nomO té algun treball no acabat.
post: *eliminaOperaris:* s'elimina l'associació entre l'obra en curs i els operaris que han intervingut.
post: *eliminaMaterial*: s'elimina l'associació entre l'obra en curs i els materials utilitzats.
post: *acabaObra*: L'obra *nomO* es passa a acabada.
post *calculaImport***:** es calcula l'import de l'obra acabada en les condicions de l'enunciat**.**

context CapaDePresentació::canviaClient()
pre: *clientVàlid*: en la llista A hi ha seleccionat un client que existeix en el sistema.
post: o*btéObres*: es crida a l'operació CapaDeDomini::obtéObresClient() amb el client de la llista A obtenint el resultat en *llObres*.
post; *noTéObres*: si es produeix l'excepció *clientSenseObres* és mostra l'error a C
post: *mostraObres*: altrament es mostra *llObres* en B.

La resta de l'especificació de capes dels dos casos d'ús no té interès aquí.

Si us convé, podeu fer ús, sense dissenyar-la, de l'operació:

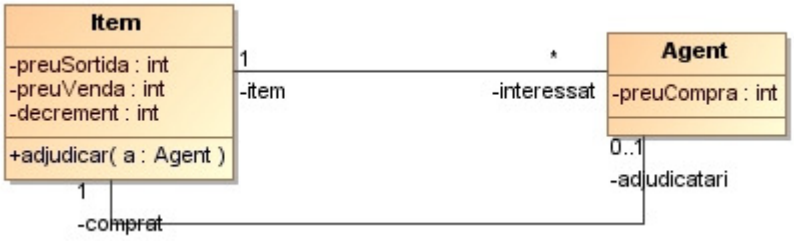
context En Curs::costMaterials(): enter
post: result = cost total dels materials que s'han fet servir a l'obra *self*.

Escollint per a totes les operacions de la capa de domini **controlador transaccional**, es demana:

a) [1 punt] Dibuixeu el diagrama de classes de disseny amb l'aplicació dels patrons que hagueu aplicat. Dissenyeu els controladors transaccionals de les tres operacions de capa de domini incloent atributs i operacions, però no cal que doneu els contractes. Les classes que no canviïn respecte de l'enunciat, amb que poseu el nom n'hi ha prou. Les classes abstractes, indiqueu-les explícitament. Inclogueu la informació sobre navegabilitat i acoblament resultats del vostre disseny.
b) [4 punts] Construïu els diagrames de seqüència de l'operació operarisGransObres() de la capa de domini i de totes aquelles auxiliars que hagueu menester.
c) [2 punts] Construïu els diagrames de seqüència de l'operació acabar de la capa de domini i de totes aquelles auxiliars que hagueu menester.
d) [1 punt] Construïu el diagrama de seqüència de l'operació canviaClient() de la capa de presentació.

1) [1 punt] Feu el mapa navegacional de la interfície gràfica del cas d'ús Facturació del problema.

2) [1 punt] Suposem que volem implementar un sistema de subhastes inverses on tenim les següents classes:



Context Item::adjudicar(a:Agent)
exc: ja-adjudicat: adjudicatari != null
post: adjudicat: adjudicatari = a

Volem afegir la següent operació
context Item::subhastar()
pre: Hi ha com a mínim un Agent interessat
post: Es notifica als agents interessats el preu de venda (*preuVenda*) de l'item. Si hi ha algun tal que el seu preu de compra sigui menor o igual que el preu de venda actual de l'item, se li adjudica l'item (fent servir l'operació adjudicar). En cas contrari, es baixa el preu i es torna a notificar fins que es trobi un adjudicatari.

Es demana: Fer el disseny de l'operació anterior aplicant els patrons que sigui oportú.