UML Estàtic
Diagrames de Classes

UF3

INTRODUCCIÓ AL DISSENY ORIENTAT A OBJECTES

M05. Entorns de desenvolupament

Hèctor López
hector.lopez@fje.edu

JESUÏTES El Clot
Escola del Clot

Introducció al DOO

Classes

Què és una classe?

Podríem entendre una classe com una "**plantilla**" per a la **creació d'objectes**, que contenen una sèrie de dades, segons un model predefinit.

Les classes s'utilitzen per a **representar entitats** o **conceptes**, sovint del món real.

Cada classe és un model que defineix un conjunt de variables, l'estat, mètodes apropiats per a operar amb les dades, i el comportament.

Cada objecte creat a partir de la classe s'anomena **instància** de la classe.

Les classes són un pilar fonamental de la programació orientada a objectes (POO), ja que permeten fer abstracció de les dades i les operacions.

Introducció al DOO

Classes

Què és una classe?

Classe d'objectes: descriu un conjunt d'objectes amb:

- les mateixes propietats
- comportament comú
- idèntica relació amb altres objectes
- semàntica comuna

l'abstracció *consisteix* ***eliminar distincions entre objectes*** *del món real per a poder observar-ne els aspectes comuns.*

els objectes d'una classe tindran les ***mateixes propietats*** *i els* ***mateixos patrons de comportament****.*

Introducció al DOO

Classes

Components

Les classes es composen d'elements, anomenats "**membres**", de manera genèrica. A continuació veiem els diversos tipus de membres d'una classe:

- **Camps de dades**: emmagatzemen l'estat de la classe mitjançant variables, estructures de dades, i fins i tot altres classes.
- **Mètodes**: subrutines (o funcions) de manipulació de les dades.
- **Propietats** (no tots els llenguatges en permeten): podriem dir que és un component que es troba a mig camí entre els camps i els mètodes.

Utilitzant un símil amb la llengua, si les **classes** representen **noms** (substantius), els **camps** de dades serien **substantius** o **adjectius**, i els **mètodes** en serien els **verbs**.

Introducció al DOO

Classes

Ha quedat clar què és una instància?

En el paradigma de la orientació a objectes, una **instància** fa referència a una **realització específica** d'una **classe** o prototipus determinats.

En llenguatges que creen objectes a partir de classes, un objecte és una instància d'una classe, i una variable és un atribut d'una classe.

Exemple:

En un context del món real, podríem pensar en un ''Gos'' com una classe i en un gos concret, com una instància d'aquesta classe.

En aquest cas no ens importa pas la raça del gos. Si fos del nostre interès modelar-la i diferenciéssim entre un doberman i un xiuaua, no només cada instància seria diferent, sinó que provindrien de classes o prototipus diferents (concepte d'***herència***).

Introducció al DOO

Introducció al món del disseny i l'anàlisi orientat a objectes

Per fer l'anàlisi i el disseny OO farem servir conceptes i notacions UML.

Els diagrames de UML es poden classificar en tres grups:

- El model estàtic, que descriu l'estructura de classes i objectes.
- El model dinàmic o de comportament, que defineix les interrelacions entre objectes.
- El model de implementació, que descriu el programari en termes de components i la seva ubicació.

El model estàtic consta de classes i objectes i de les relacions entre aquests.

L'UML (http://www.uml.org/) està definit per l'OMG. http://www.omg.org/

UNIFIED MODELING LANGUAGE™

Model estàtic

Concepte de model estàtic i diagrama de classes

El model estàtic consta de classes i objectes, i de les relacions entre aquests. S'anomena estàtic perquè mostra les relacions possibles al llarg del temps i no únicament aquelles que son vàlides en un determinat moment.

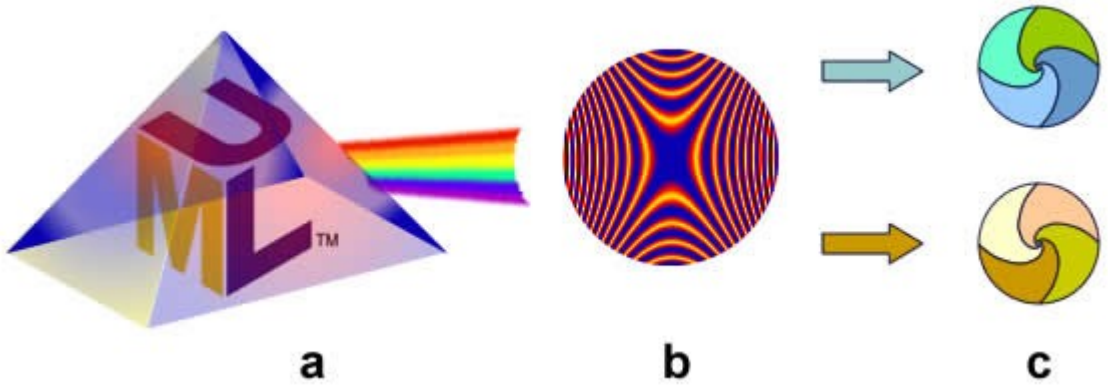
Consta de dos diagrames:

Diagrama de classes: pot contenir classes, objectes i relacions entre elles que sempre estan definides. Mostra la estructura estàtica de les classes en un domini (part del món considerada per l'aplicació) i mostra les relacions i les classes que existeixen. Les relacions poden ser *herència*, *associació*, *agregació, composició* o *ús*. És **independent del llenguatge de programació** i per tant, s'hauran d'evitar aquells aspectes específics del llenguatge. Si en algun moment hem de representar alguna qüestió i l'UML no ho permet podrem utilitzar les extensions d'UML.

Diagrama d'objectes: només té en compte objectes i les relacions entre ells, aquest diagrama és opcional fer-lo, i **es fa servir en poques ocasions**, especialment **per a fer exemples** del diagrama de classes.

Model estàtic

Diagrames de classes



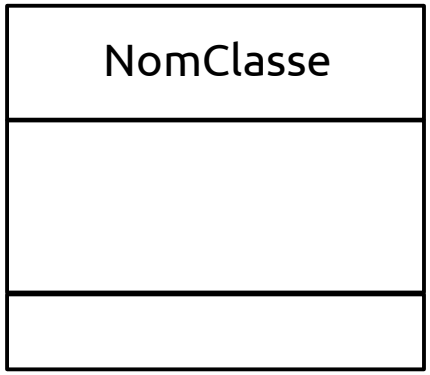
Com esmentàvem abans, un diagrama de classes mostra l'**estructura estàtica de les classes en un domini** (porció del món real considerada per una aplicació); s'hi mostren les **classes**, i també les **relacions** entre aquestes, que poden ser d'**herència**, **associació**, **agregació**, **composició**, o **ús**.

Els llenguatges de programació **suporten** les relacions d'**herència** però **no distingeixen** entre **els altres tres tipus de relacions** que hi ha; per tant, en passar del disseny a la programació, aquests tipus **s'hauran de transformar**.

Model estàtic

Concepte de model estàtic i diagrama de classes

Heus aquí com es representa una classe sense atributs ni mètodes amb UML:



Un diagrama de classes mostra l'**estructura estàtica de les classes en un domini** (porció del món real considerada per una aplicació); s'hi mostren les **classes**, i també les **relacions** entre aquestes, que poden ser d'**herència**, **associació**, **agregació**, **ús,** etc.

Anem a treballar tots aquests conceptes pas a pas...

Model estàtic

Classificadors

Un **classificador** és una entitat bàsica del model estàtic. És més general que el concepte de classe, i el conjunt d'elements que el formen s'anomenen instàncies. No tenen representació gràfica, i hi ha tres tipus d'estereotips:

Classe: el concepte de classe segons UML és el mateix que l'OO i les seves instàncies són els objectes. Cada instància té identitat.

Tipus de dada: consisteix en un tipus bàsic ofert per un llenguatge o desenvolupat pel programador. També té operacions, però les seves instàncies no tenen identitat.

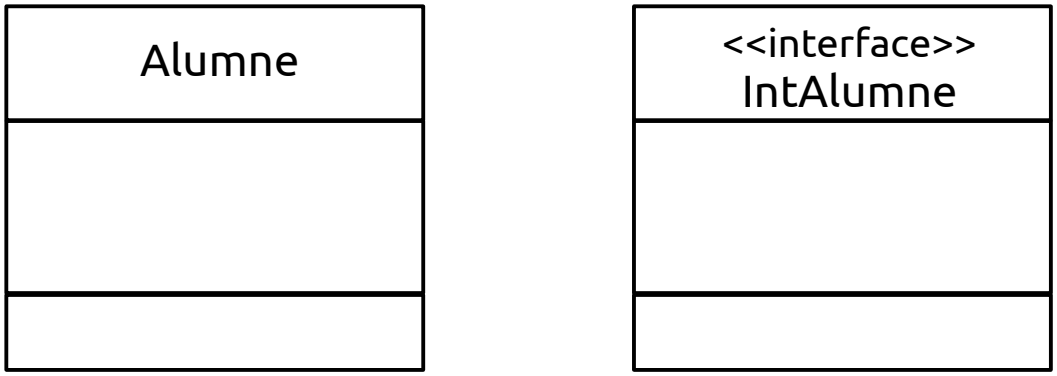
Interfície: descriu només aquelles operacions d'una classe que són visibles des d'altres classes.

Els tres classificadors es representen mitjançant un rectangle, han de tenir un nom que normalment serà un substantiu singular, amb la primera lletra en majúscula. Si es vol, es pot indicar l'estereotip (entre cometes llatines, «»), i si no s'indica, per defecte és una classe.

Model estàtic

Classificadors

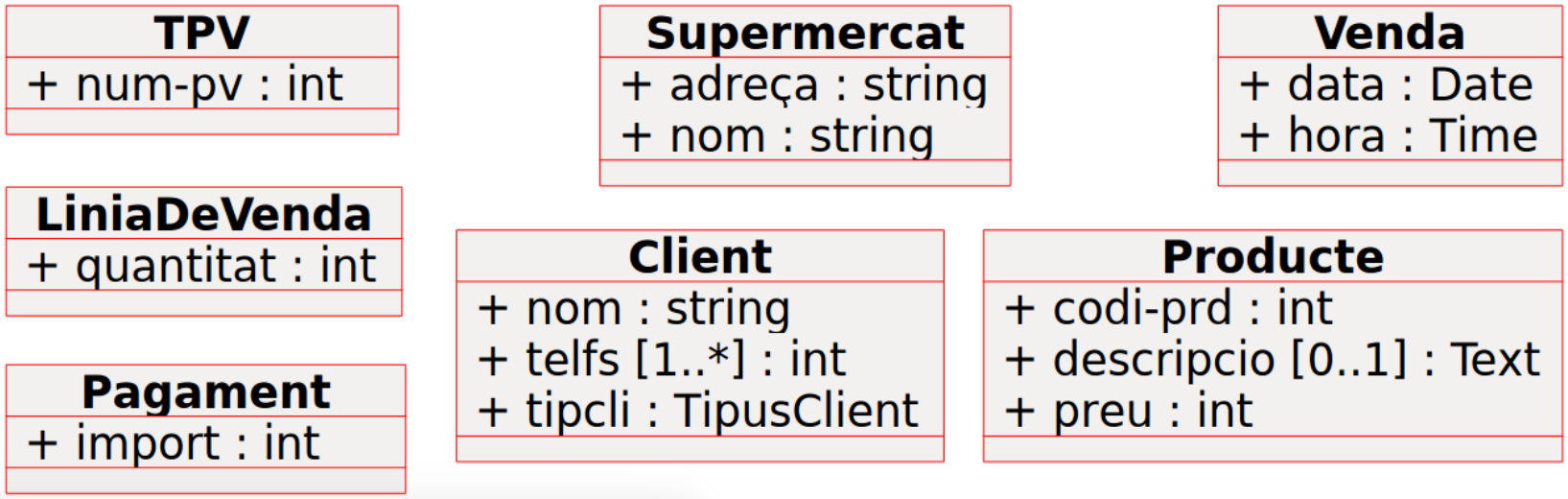
A continuació un exemple de classe i d'interfície:



Model estàtic

Atributs

A continuació un exemple sobre com especificar els atributs d'una classe en UML:



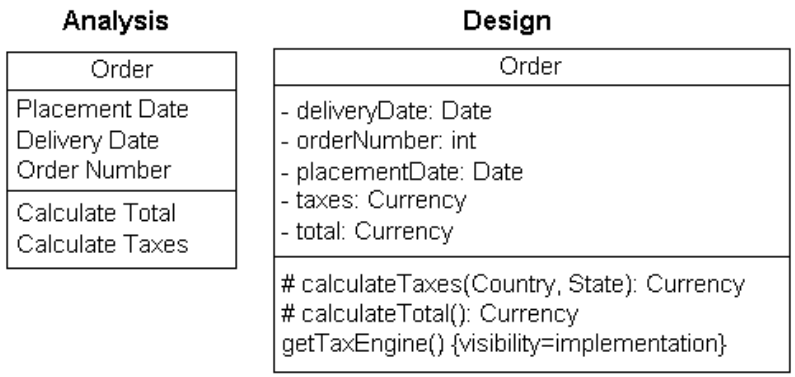
Recordeu que: un atribut és una propietat compartida per **TOTS** els objectes d'una classe.

Model estàtic

Visibilitat

Tant per als atributs (com podeu veure en l'exemple anterior) com per als mètodes, els següents símbols especifiquen els diferents tipus de visibilitat que tenim:

- **+** especifica que l'atribut/mètode és **públic** (*public*).
- **-** especifica que l'atribut/mètode és **privat** (*private*).
- # especifica que l'atribut/mètode és **protegit** (*protected*).



Model estàtic

Associacions

Una associació representa una **relació entre classes**, i dóna una semàntica comuna per a totes les futures "connexions" entre els objectes d'aquestes. Són el mecanisme que permet que els objectes es puguin **comunicar** uns amb altres.

Les associacions han de tenir un **nom** i una **direcció**, que especifiqui el propòsit d'aquestes. Entre dues classes hi poden haver diverses associacions amb significats diferents. A cadascuna de les classes de la relació també se li pot donar un **rol**.

Els **rols** poden ser o bé **unidireccionals** o bé **bidireccionals** (indica si els dos objectes que participaran en la relació podran enviar missatges a l'altre, o si només un podrà fer-ho).

En una associació cadascuna de les classes fa un paper, i cada paper té una **multiplicitat** (o cardinalitat). La multiplicitat d'un costat de l'associació indica **quantes instàncies de l'altra classe podem tenir per només una instància de la classe en qüestió**.

A banda de les **associacions binàries** (entre dues classes), en UML també podem especificar que les classes tinguin relacions amb sí mateixes. Quan una classe té una relació amb sí mateixa, aquesta associació s'anomena **associació reflexiva**.

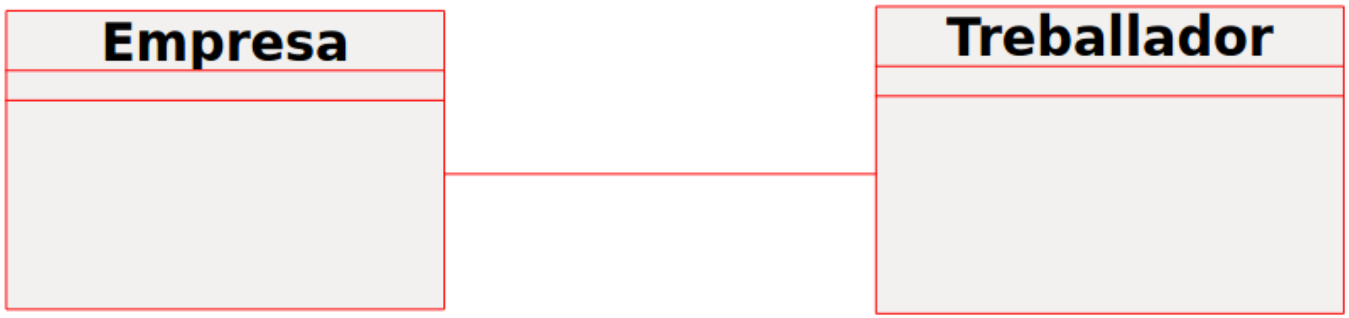
Model estàtic

Associacions

Anem a comentar-ho poc a poc, analitzant **cadascun dels paràgrafs de la diapositiva anterior**, i veient com es representa en UML cadascun d'aquests conceptes...

*«Una associació representa una **relació entre classes**, i dóna una semàntica comuna per a totes les futures "connexions" entre els objectes d'aquestes. Són el mecanisme que permet que els objectes es puguin **comunicar** uns amb altres.»*

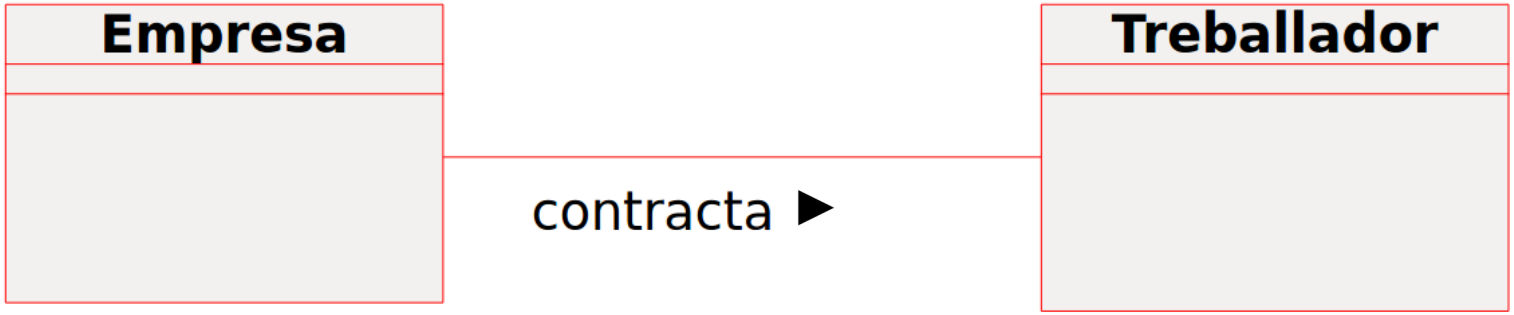
En UML les associacions es representen mitjançant línies, connectant les classes que formen part d'aquesta relació:



Model estàtic

Associacions

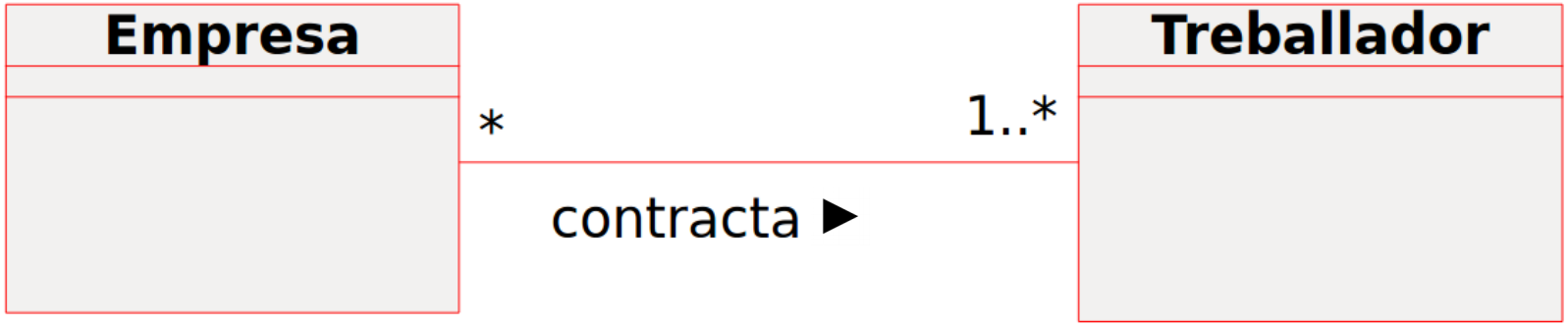
«*Les associacions poden tenir un **nom** i una **direcció**, que especifiqui el propòsit d'aquestes. Entre dues classes hi poden haver diverses associacions amb significats diferents. A cadascuna de les classes de la relació també se li pot donar un **rol**.*»



Model estàtic

Associacions

«*En una associació cadascuna de les classes fa un paper, i cada paper té una* ***multiplicitat*** *(o cardinalitat). La multiplicitat d'un costat de l'associació indica* ***quantes instàncies de l'altra classe podem tenir per només una instància de la classe en qüestió***.»



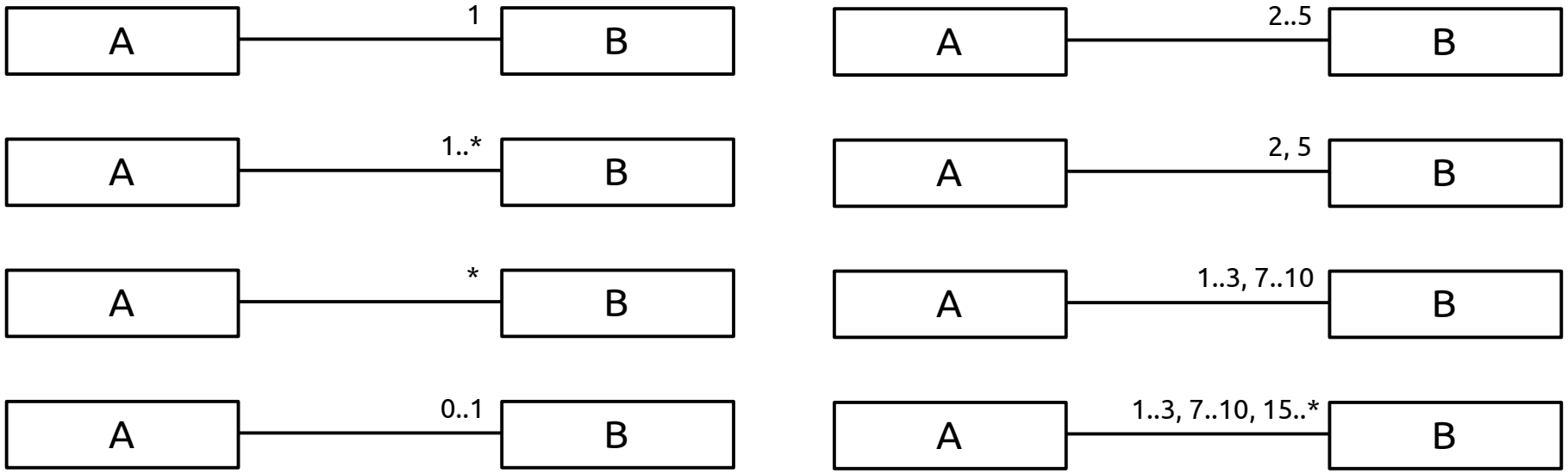
Interpretació (''en un moment donat''):

1. **Una** empresa contracta a **com a mínim un** treballador fins a **diversos** treballadors.
2. **Un** treballador pot ser contractat per **cap** o **diverses** empreses.

Model estàtic

Associacions

Multiplicitat de les associacions binàries



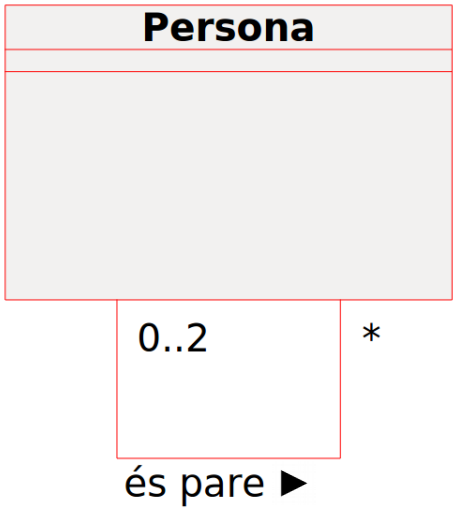
Per tota instància ***a*** de la classe A, la multiplicitat del costat B defineix el nombre mínim i màxim d'instàncies de B associades amb ***a***.

Model estàtic

Associacions

Reflexives

*"A banda de les **associacions binàries** (entre dues classes), en UML també podem especificar que les classes tinguin relacions amb sí mateixes. Quan una classe té una relació amb sí mateixa, aquesta associació s'anomena **associació reflexiva**".*



Interpretació ("en un moment donat"):

1. **Una** persona pot ser pare de **cap** o **diverses** persones.
2. **Una** persona pot tenir de **cap** a **dos** pares.

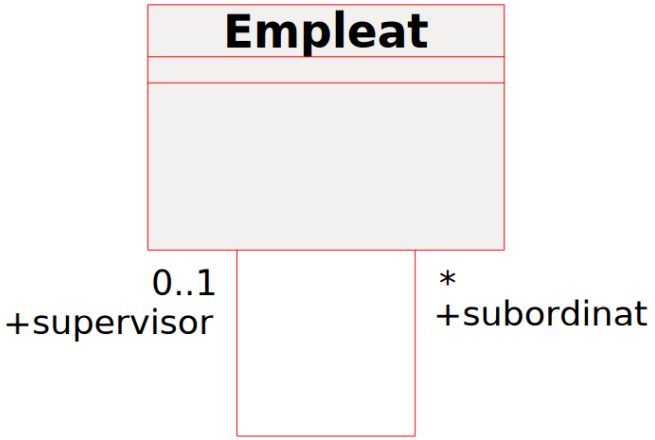
Model estàtic

Associacions

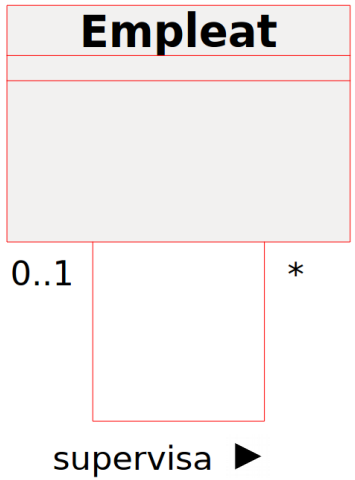
Reflexives

Exemple amb la situació
de supervisió/subordinació
de treballadors en una empresa.

Fent ús de rols



Fent ús de l'associació



Interpretació (''en un moment donat''):

1. **Un** empleat pot supervisar de **cap** a **diversos** empleats.
2. **Un** empleat pot ser supervisat per **cap** o, **com a màxim un** altre empleat.

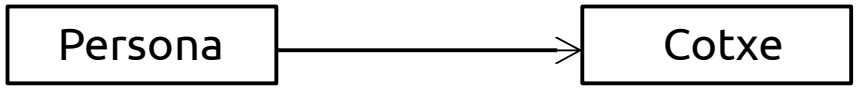
Model estàtic

Associacions

Unidireccionals

*«Els **rols** poden ser o bé **unidireccionals** o bé **bidireccionals** (indica si els dos objectes que participaran en la relació podran enviar missatges a l'altre, o si només un podrà fer-ho)»*

Representació:



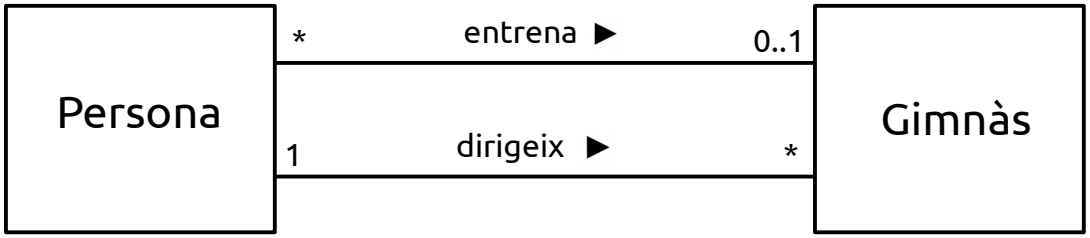
Nota: Recordeu que quan no posem fletxa en cap de les dues bandes de l'associació estem representant una associació bidireccional.

Model estàtic

Associacions

Podem especificar més d'una associació entre dues classes?

En UML **no tenim** cap **limitació** pel que fa al **número d'associacions** que podem establir entre dues classes:

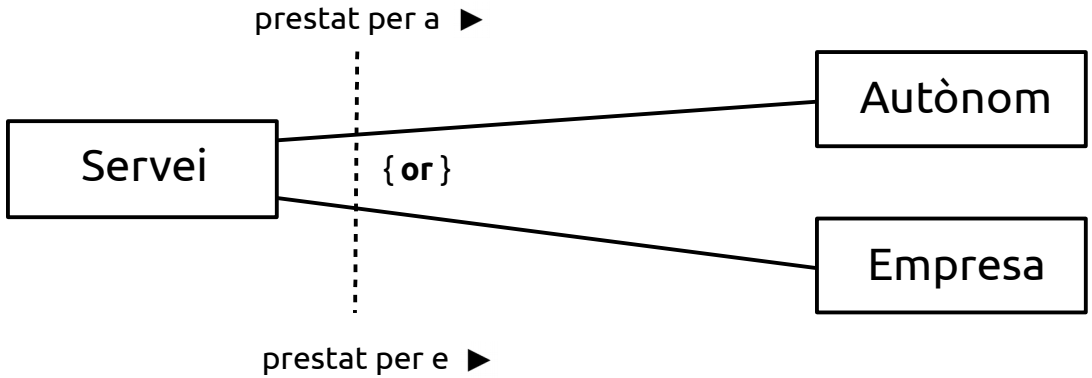


Model estàtic

Associacions

Associacions alternatives (I)

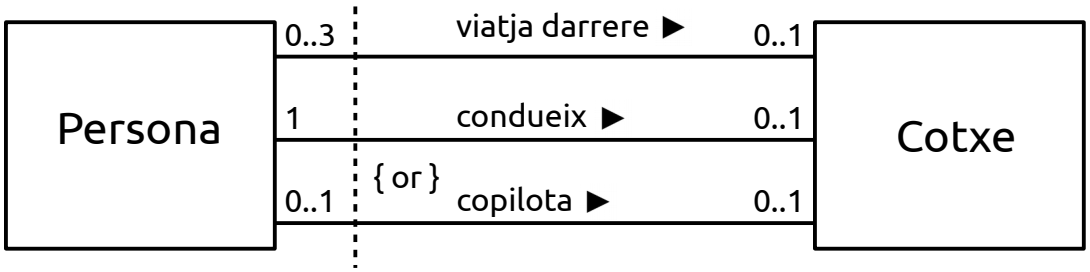
A vegades pot passar que una classe participi en dues associacions, però que els objectes hagin de participar, o bé en una, o bé en l'altra associació, però no pas a totes dues a la vegada. Per representar aquest cas ho faríem de la següent manera:



Associacions

Associacions alternatives (II)

Sí podem especificar **condicions entre** aquestes **associacions**, en el cas que hi hagi més d'una, ja siguin incloents o excloents:



Amb la paraula reservada **{ or }** d'UML estem especificant que, per una instància de persona i una instància de cotxe, o bé es dóna la relació viatja darrere, o bé la relació condueix, o bé la condició copilota.

Nota: Les opcions que podem posar dins els { } són: **`or`**, **`and`** i **`xor`**.

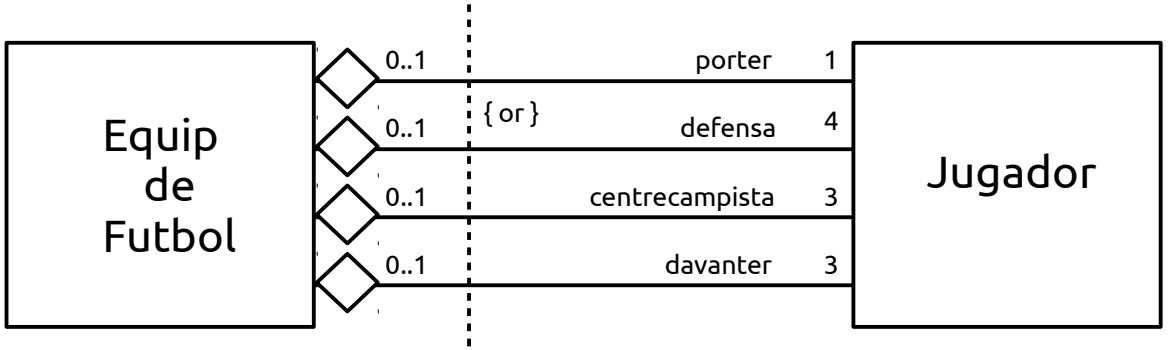
Model estàtic

Agregacions

Les agregacions són un tipus especial d'**associació** en les quals les classes participants no tenen un estatus igualitari, sinó que formen una relació de tipus '***part-tot***'.

Una agregació descriu una classe pren el rol del **tot**. És a dir, és composada (té) per altres classes, que prenen el rol de **part**. S'anomena **component** cadascun dels objectes del primer paper, i **agregats**, els del segon.

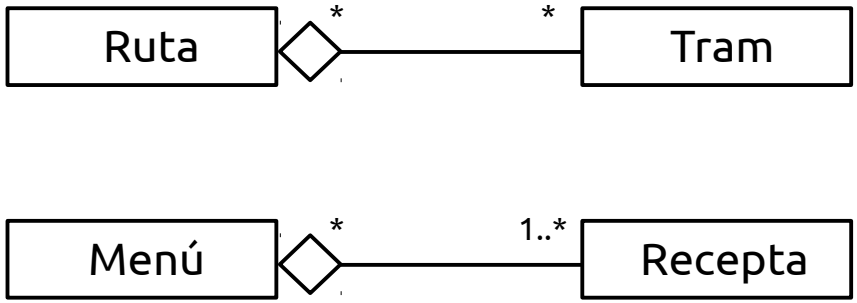
En UML, les agregacions es representen mitjançant una **associació amb un romb** a la banda on es troba la classe que pren el rol del *tot*:



Model estàtic

Agregacions

Exemples d'agregacions:



Model estàtic

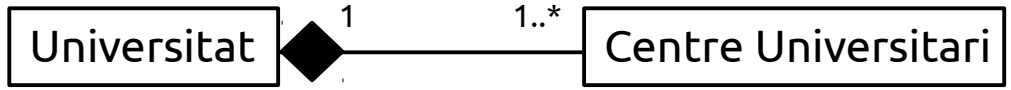
Composicions

Les composicions són un **cas particular d'una agregació**, on els objectes components no tenen vida pròpia. És a dir, que han d'estar relacionats sempre amb el tot, de tal manera que **si es destrueix el tot**, **es destrueixen els components** (no tenen sentit per sí sols).

Un objecte component només pot formar part d'un objecte compost, i no pot passar d'un objecte compost a un altre. Això no passa en el cas de les agregacions.

La multiplicitat del cap **compost** doncs, pot ser com a **màxim 1**.

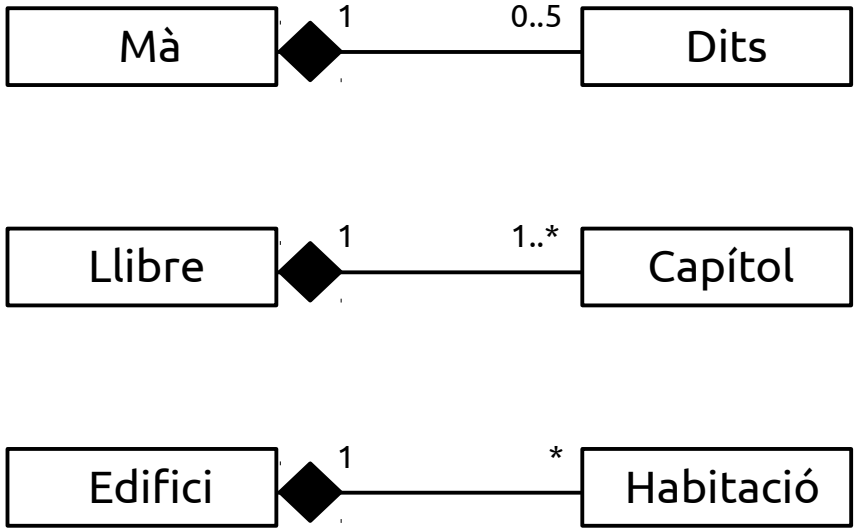
En les composicions el *tot* s'anomena *classe composta* i les *parts* s'anomenen *components*.



Model estàtic

Composicions

Exemples de composicions:

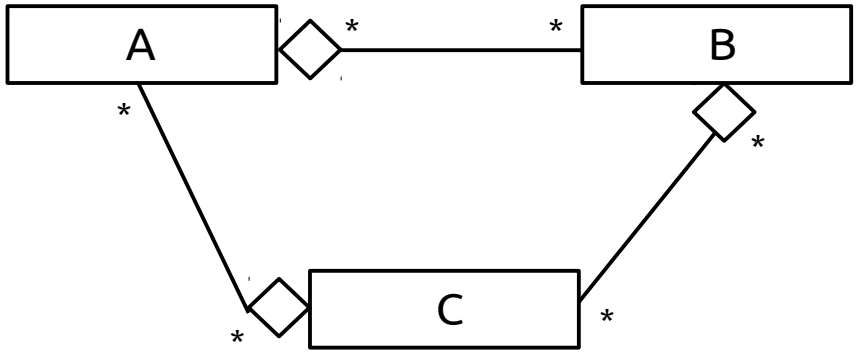


Model estàtic

Agregacions ()

Sovint les agregacions són difícils de diferenciar de les associacions normals. Tant, que la distinció entre elles és sovint subjectiva.

L'única restricció que afegeix l'agregació respecte l'associació és que les cadenes d'agregacions entre instàncies d'objectes **no poden formar cicles**.

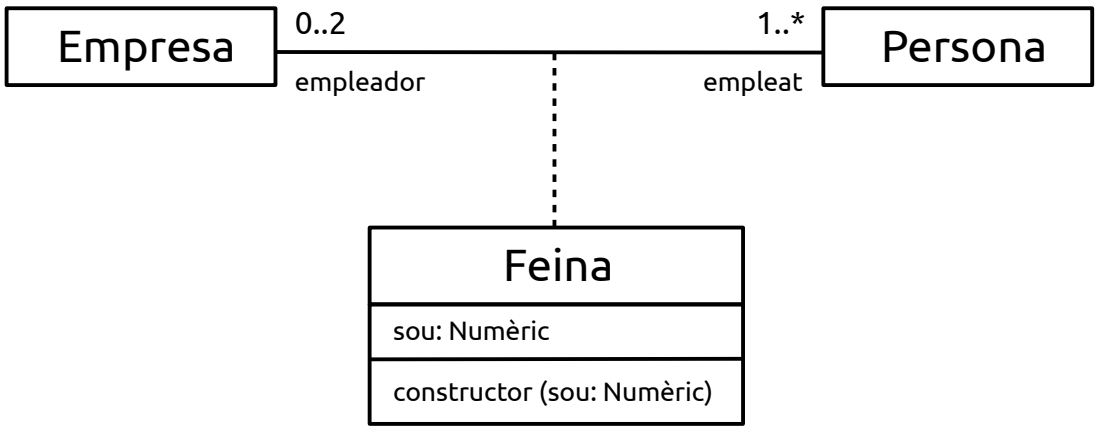


Model estàtic

Classes associatives

Les podríem definir com "*aquelles classes que neixen de la relació entre dues altres*".

Les associacions no tenen operacions ni atributs (no són classes). Així doncs, si en algun moment ens convingués emmagatzemar més informació relacionada amb una relació entre dues classes, podrem definir una **classe associativa**, tal i com es mostra a continuació (la classe associativa és la classe *Feina*):

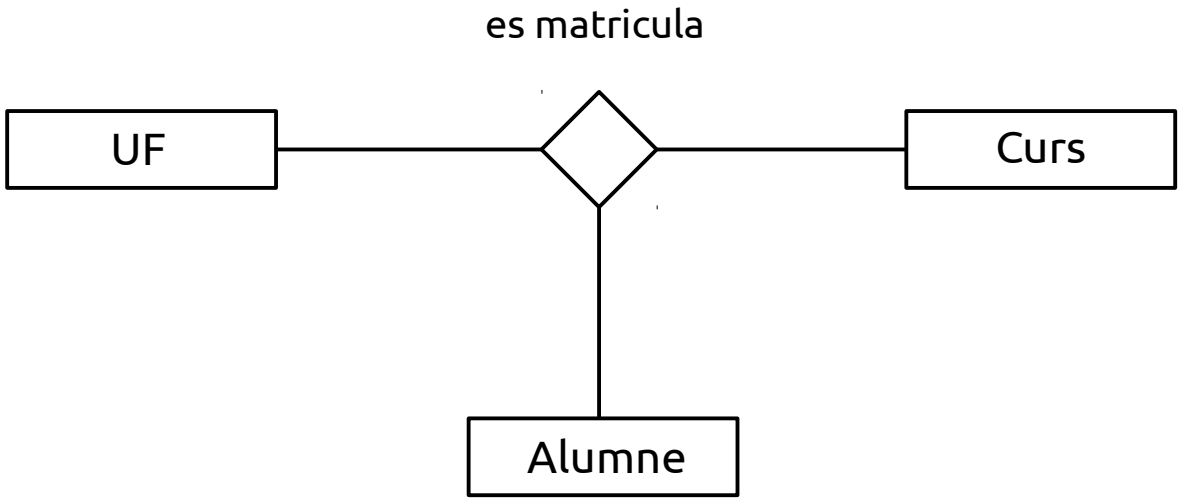


Model estàtic

Associacions

Ternàries

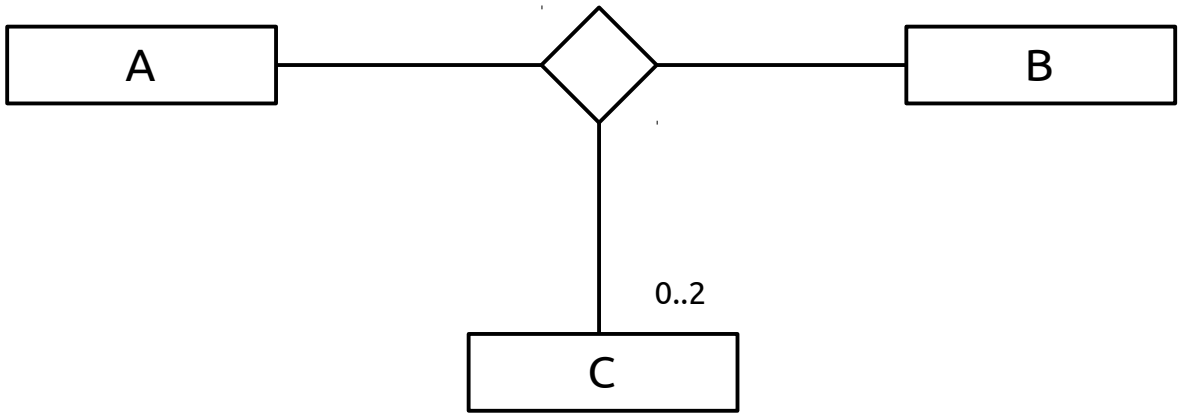
Les associacions **ternàries** ens permeten establir relacions d'ordre superior a dos entre classes.



Model estàtic

Associacions

Multiplicitat de les associacions ternàries



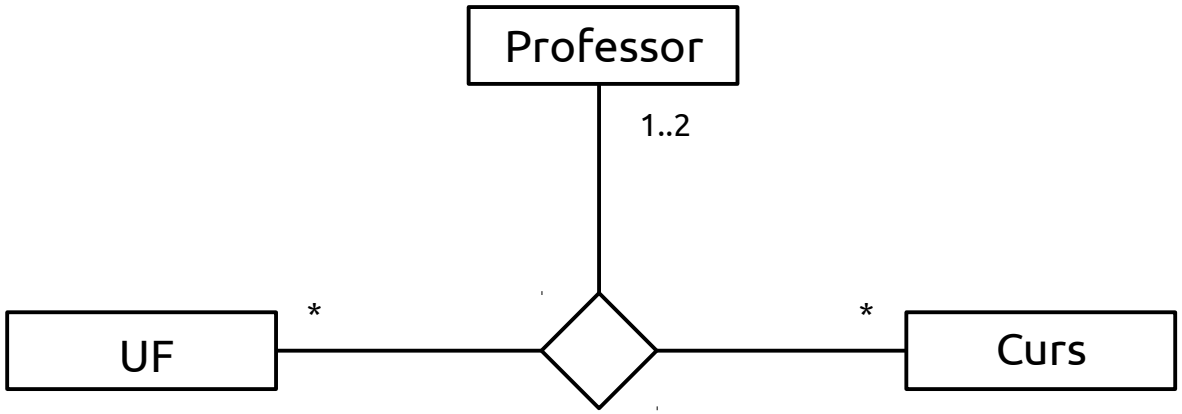
Per tota parella d'instàncies (***a,b***), on ***a*** és instància de la classe A i ***b*** és instància de la classe B, la multiplicitat al costat C defineix el nombre mínim i màxim d'instàncies de C associades amb la parella (***a,b***).

Model estàtic

Associacions

Ternàries

Exemple 1:



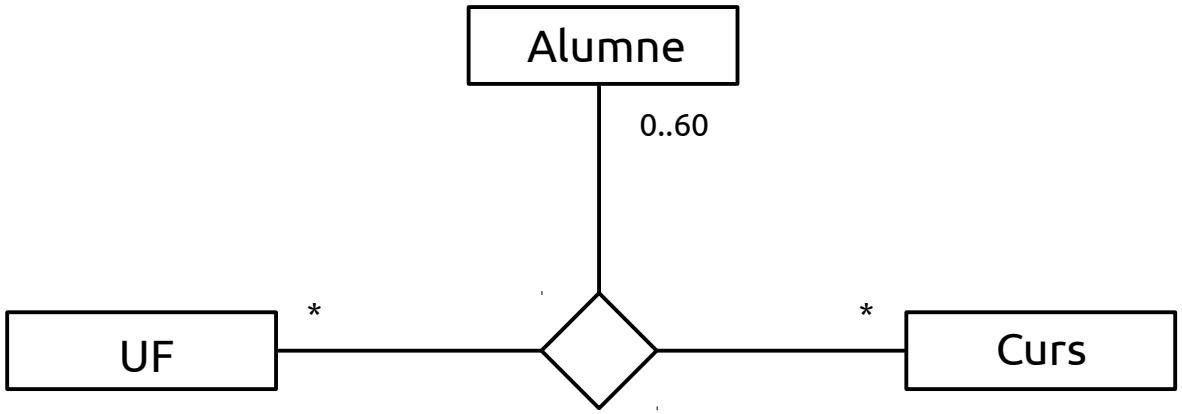
Segons aquest diagrama, per tota parella d'uf i curs, hi ha d'haver com a mínim un professor responsable, i com a màxim dos.

Model estàtic

Associacions

Ternàries

Exemple 2:



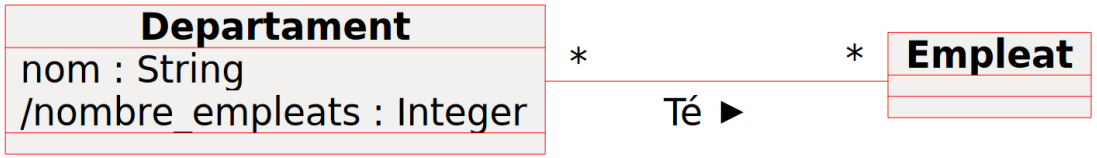
Aquest diagrama, per exemple, permet que hi hagi alguna parella d'uf i curs, per la qual no hi ha cap alumne. És a dir, estaríem dient que permetem que en una unitat formativa d'un curs, podríem estar sense alumnes.

Model estàtic

Informació derivada

- Un element (atribut o associació) és **derivat** si es pot calcular a partir d'altres elements.
- S'inclou quan millora la **claredat** del model conceptual.
- Una "*constraint*" (regla de derivació) ha d'especificar com es deriva.

Atribut derivat →



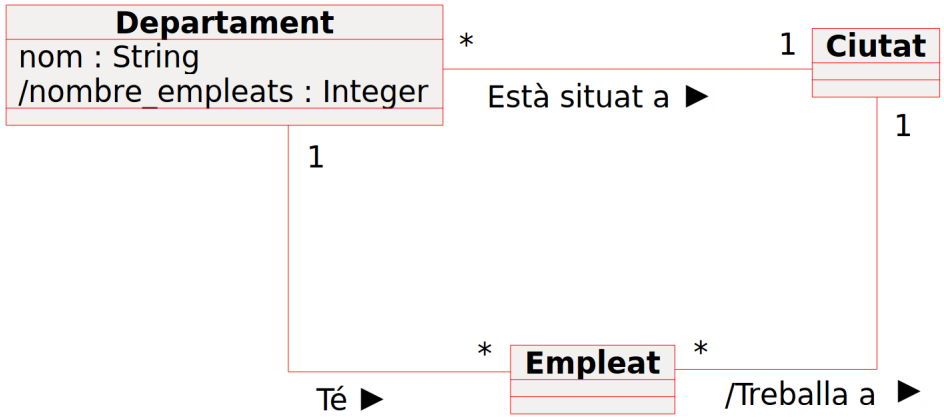
el nombre d'empleats d'un departament d és igual al nombre d'ocurrències de Té on apareix d

Model estàtic

Informació derivada

- Un element (atribut o associació) és **derivat** si es pot calcular a partir d'altres elements.
- S'inclou quan millora la **claredat** del model conceptual.
- Una "*constraint*" (regla de derivació) ha d'especificar com es deriva.

Associació derivada →

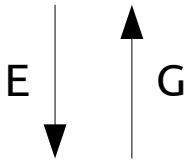


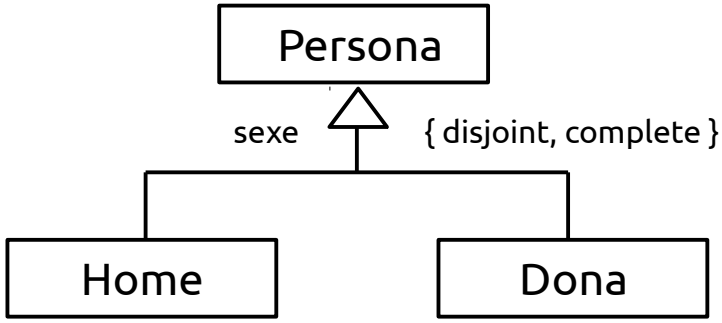
la ciutat on treballa un empleat és la ciutat on està situat el departament de l'empleat

Model estàtic

Generalització / Especialització

Identificar **elements comuns** entre els objectes definint relacions de superclasse (objecte **general**) i subclasse (objecte **especialitzat**).





discriminador → És el nom de la partició.
→ Ha de ser únic entre els atributs i rols de la superclasse.

Opcions:

1. **disjoint** → Un descendent no ho pot ser en més d'una subclasse.
   overlapping → Un descendent pot ser de més d'una subclasse.
2. **complete** → S'han especificat totes les subclasses.
   incomplete → La llista de subclasses és incompleta.

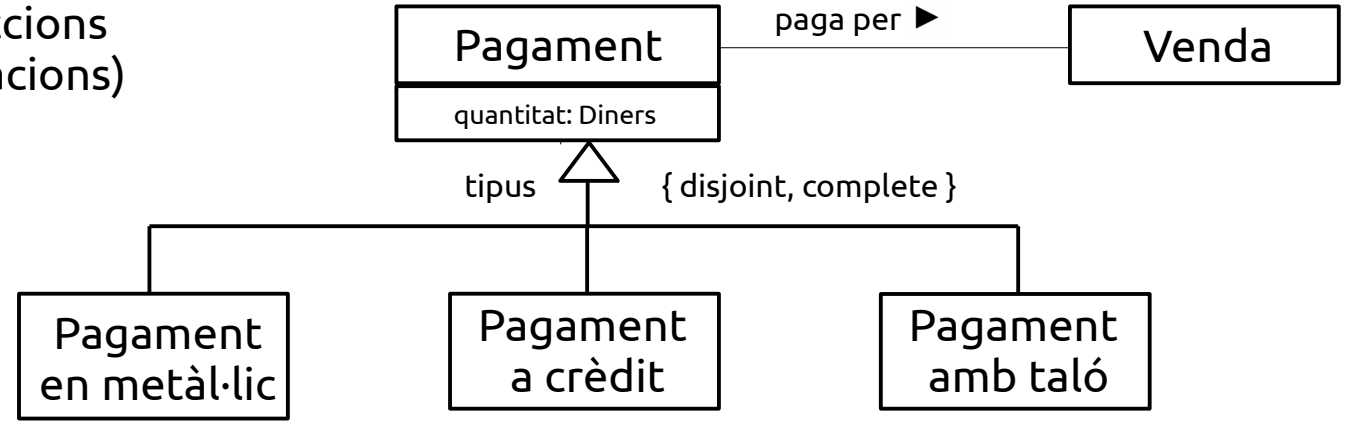
Model estàtic

Generalització / Especialització

Permet entendre els conceptes en termes més generals, refinats i abstractes, així com també fa els diagrames més expressius.

- Tots els objectes de la subclasse ho són també de la superclasse.
- La definició de la superclasse ha de ser aplicable a la subclasse:
  - atributs
  - associacions
  - restriccions

  (- operacions)



Model estàtic

Generalització / Especialització

Motivacions per particionar una classe en subclasses

- La subclasse té atributs o associacions addicionals.
- La subclasse és manipulada o té comportament diferent de la superclasse o d'altres subclasses.

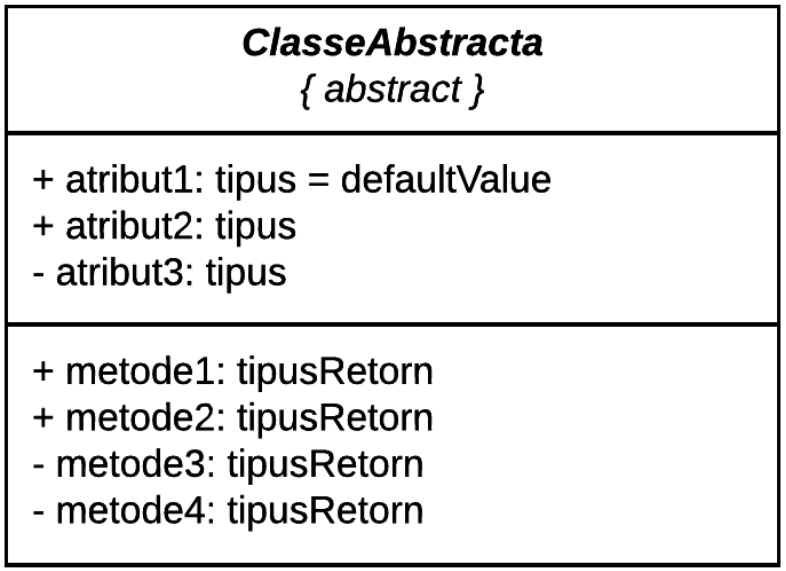
Motivacions per definir una superclasse

- Les subclasses potencials representen variacions d'un mateix concepte.
- Les subclasses tenen atributs o associacions que poden ser factoritzats i expressats a les superclasses.

Model estàtic

Classes abstractes

Representació:



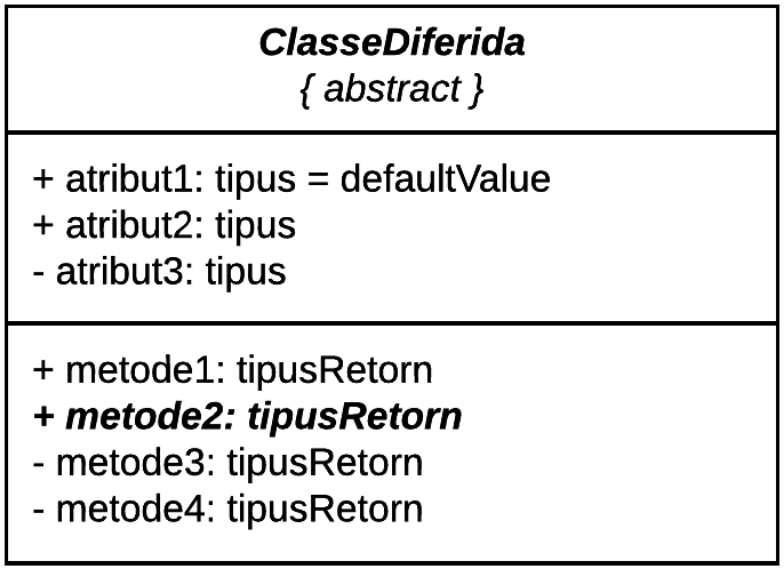
Recordatori:

→ Una classe **abstracta** és una classe de la qual no se'n poden crear instàncies, i utilitzem per descriure aspectes que després implementen altres subclasses.

Model estàtic

Classes diferides

Representació:



Recordatori:

→ Una classe **diferida** és un subtipus de classe abstracta que concretament té un o més serveis declarats, però no implementats (mètodes abstractes).

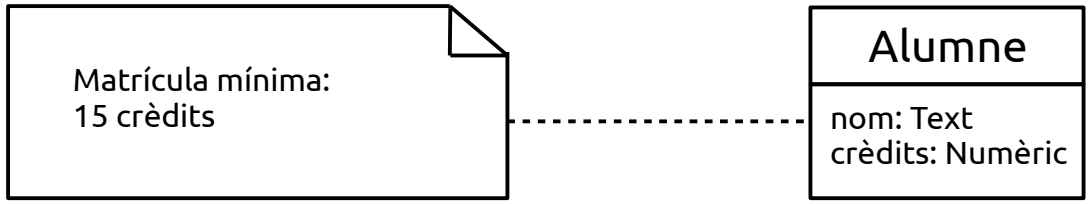
LES CLASSES DIFERIDES NO TENEN REPRESENTACIÓ ESPECÍFICA EN UML

Model estàtic

Comentaris i restriccions

Comentaris

Els comentaris es posen amb una línia discontínua sense fletxa, i serveixen per aportar informació textual sobre la classe en qüestió:



Model estàtic

Comentaris i restriccions

Restriccions

Les restriccions expressen condicions que ha de complir l'element del model al qual estan associats. Es representen igual que els comentaris, però es posen entre claudàtors.

Per definir les restriccions en UML, s'utilitza un llenguatge propi anomenat **OCL** (**O**bject **C**onstraint **L**anguage), que necessita especificar les precondicions, postcondicions i invariants.

Per què un llenguatge addicional?

→ Els models gràfics no són suficients per a una especificació precisa i no ambigua.

L'OCL s'utilitza per acabar de cobrir tot allò que no podem especificar amb l'UML (sovint no serà necessari), també sense possibles ambigüitats. Representa un suplement del llenguatge, dotant-lo d'expressions lliures de l'ambigüitat característica del llenguatge natural, basant-se en complexos matemàtics.

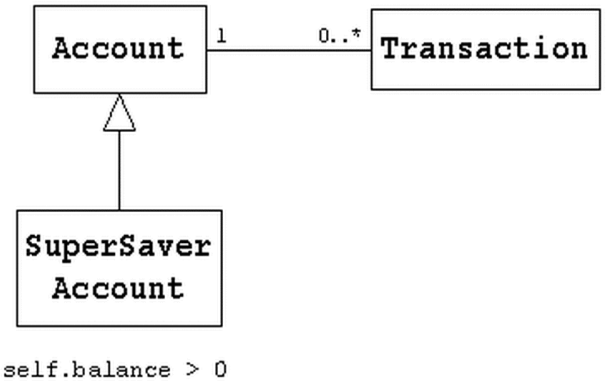
Model estàtic

Comentaris i restriccions

Restriccions

Exemple de codi **OCL**:

```
self.transaction -> forAll(t:Transaction | t.value > 100)
```




```
self.balance > 0
```