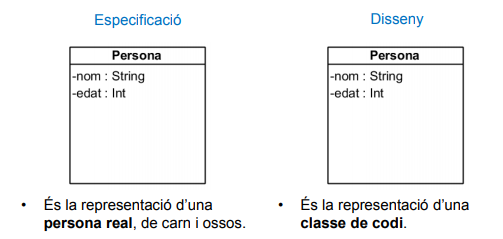
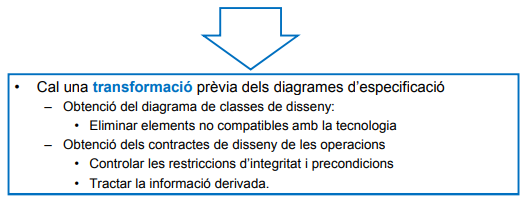
**Tema 7: De l’Especificació al Disseny: Diagrama de Classes i Contractes**

**Quina diferència hi ha?**



**De l’Especificació al Disseny**

* Al disseny hi tenim components de software i no conceptes de domini.
* Limitació tecnològica:
  + No podem implementar directament tots els conceptes que hem usat a l’especificació, alguns depenen de la tecnologia.



**Transformació del diagrama de classes**

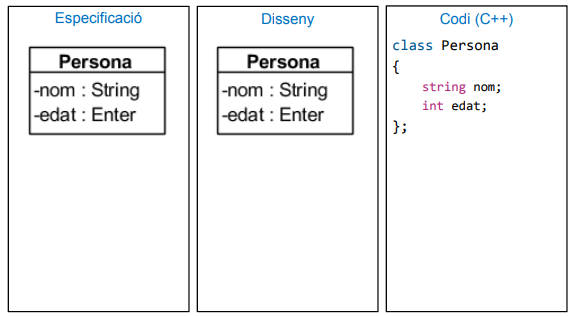
El diagrama de classes d’especificació és més expressiu que el diagrama de classes de disseny, per tant, per caldrà:

* Mantenir aquells elements que siguin compatibles:
  + Classes no Associatives.
  + Generalitzacions/Especialitzacions Disjoint.
  + Associacions binàries.
  + Atributs.
* Eliminar aquells elements que no siguin compatibles i reemplaçar-los per d’altres que sí que ho siguin:
  + Classes Associatives.
  + Associacions N-àries.
  + Altres casos de Generalització / Especialització.
  + Classes Especials (com Data, Hora, ...).
  + Atributs derivats.
* Afegir aquells conceptes de disseny que no són necessaris a especificació.
  + Visibilitat, Àmbit i Excepcions.

**Elements Compatibles:**

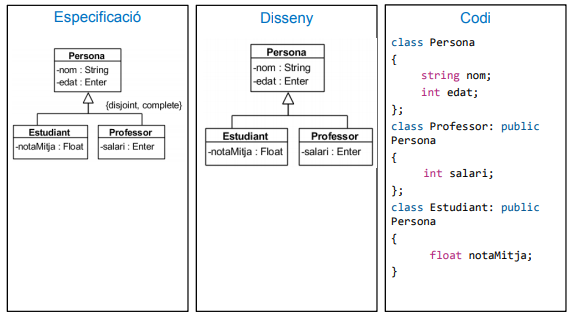
Classes no associatives

Les classes d’especificació són compatibles, per tant, es mantenen a disseny amb la mateixa forma.



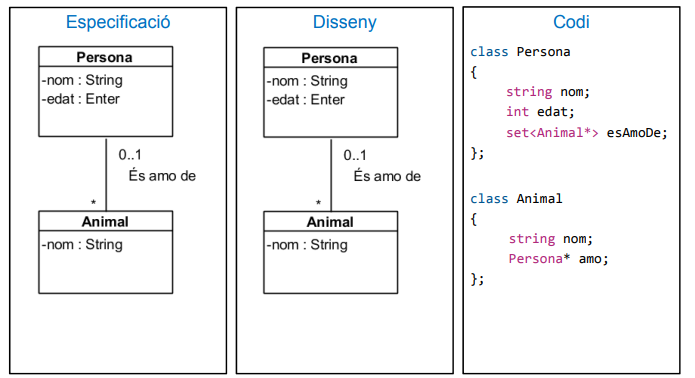
Generalitzacions/especialitzacions disjoint

Les herències disjoint són compatibles, per tant es mantenen igual.



Associacions binàries

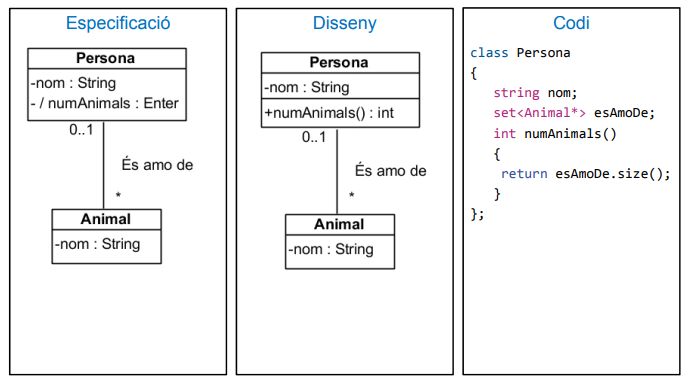
Les associacions binàries es mantenen igual, tot i que a disseny els haurem d’afegir navegabilitat.



**Elements No Compatibles:**

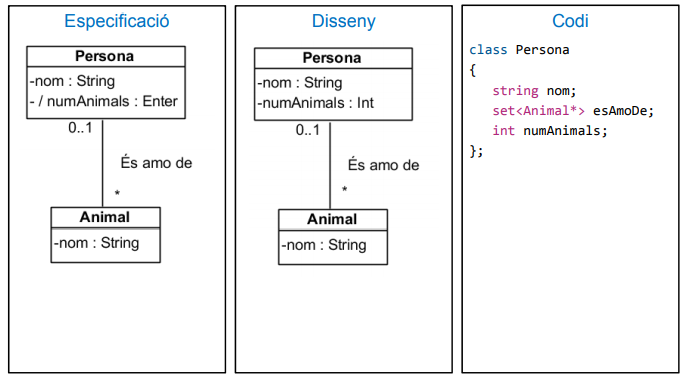
Atributs Derivats (I)

Els atributs derivats es poden convertir en una funció nova que calcula el valor. Si són així, direm que són atributs **Calculats**.



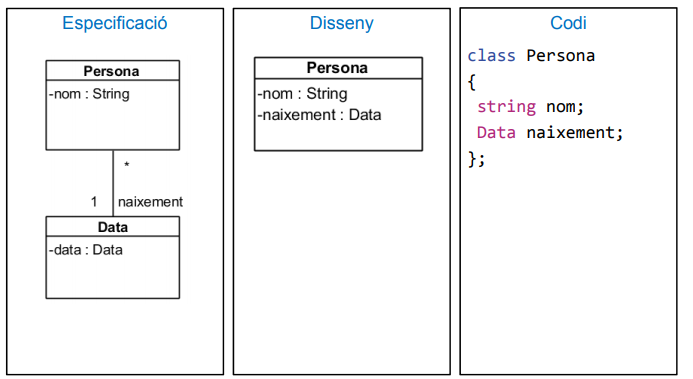
Atributs Derivats (II)

Els atributs derivats es poden convertir en un nou atribut “real” que s’haurà de mantenir actualitzat. Si són així, direm que s’han **Materialitzat**.



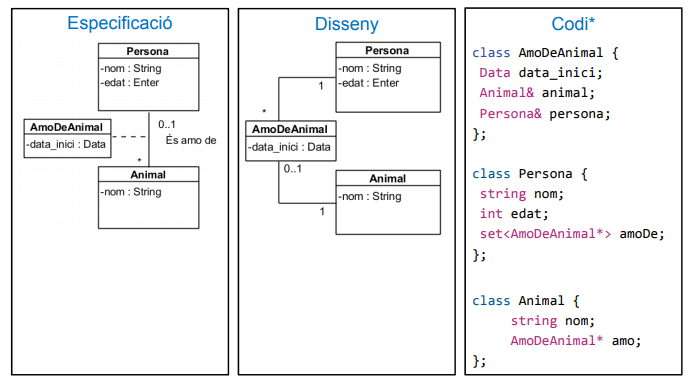
Classes Especials (Data, Hora, ...)

Les classes especials esdevenen atributs amb el tipus corresponent, per tant, s’ha d’eliminar la classe original i afegir com a un atribut.



Classes Associatives

Les classes associatives es converteixen en classes no associatives i es relacionen amb les dues originals...

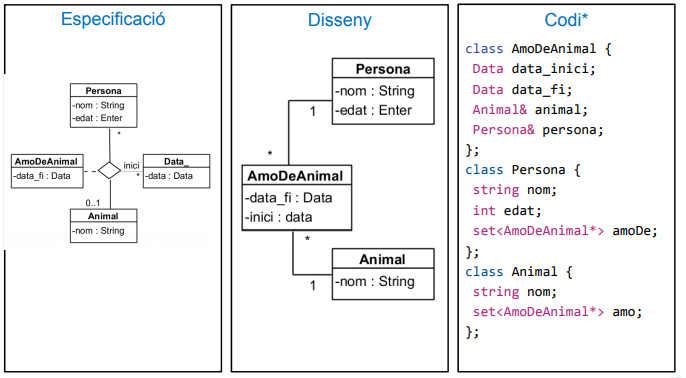


El diagrama obtingut ha de tenir la mateixa semàntica que l’original, per tant, haurem d’afegir restriccions textuals per a compensar la pèrdua d’informació.

* No hi pot haver dos “AmoDeAnimal” amb els mateixos Persona i Animal.

Associacions N-àries

Les associacions N-Àries s’han de canviar per una classe associativa relacionada amb les N originals...



El diagrama obtingut ha de tenir la mateixa semàntica que l’original, per tant, haurem d’afegir restriccions textuals per a compensar la pèrdua d’informació (igual que amb les classes associatives):

* RT1: No hi pot haver dos “AmoDeAnimal” amb els mateixos Persona, Animal i Inici.

També s’han de considerar les multiplicitats de cada un dels membres de la N-ària per a afegir noves restriccions:

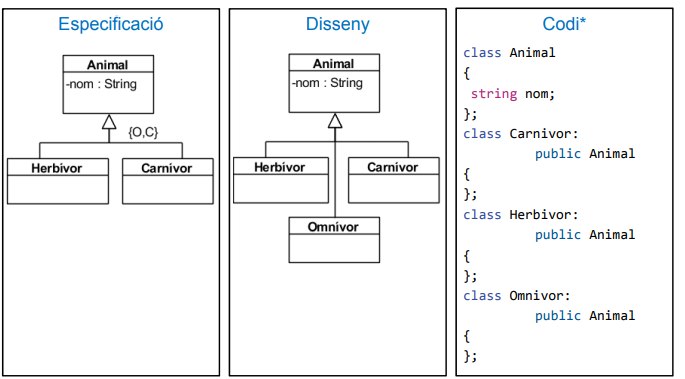
* RT2: Donada una persona i una Data, màxim pot esdevenir amo d’un Animal.

I això pot provocar que algunes restriccions textuals siguin redundants.

* RT1 és redundant amb RT2, per tant, no s’ha d’afegir RT1.

Generalització / Especialització No Disjoint

No hi ha una forma única de fer-ho depèn de cada situació. El producte cartesià és només una de les opcions, però n’hi ha més.



**Elements Exclusius de Disseny:**

Visibilitat

Defineix quins objectes tenen dret a consultar i modificar informació declarada en un diagrama de classes. Pot ser de tres tipus:

* Pública (+).
* Privada (-).
* Protegida (#).

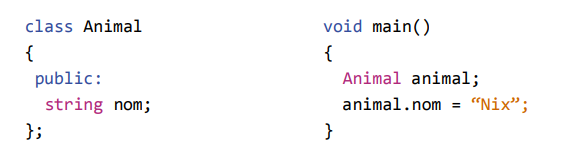
Aplica a:

* Atributs.
* Operacions.
* Rols.

A IES assumirem que, per defecte, els atributs i rols són privats, i les operacions són públiques.

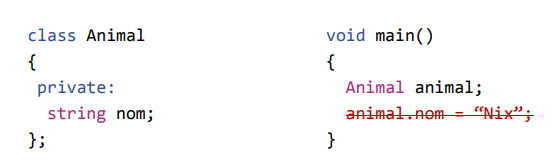
1. Visibilitat Públic:

* Donat un element X d’una classe C:
  + Si és públic, qualsevol que vegi C, veurà X.



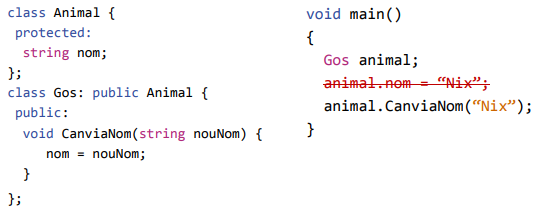
2. Visibilitat Privat:

* Donat un element X d’una classe C:
  + Si és privat, només C veurà X.



3. Visibilitat Protegit:

* Donat un element X d’una classe C:
  + Si és protegit, només C o els seus descendents veuran X.



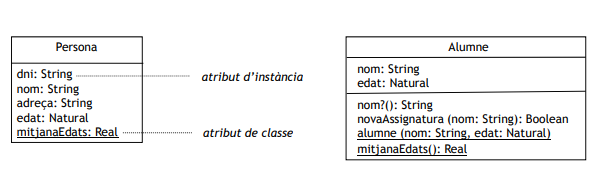
Àmbit

Determina si els atributs o operacions són aplicables a objectes individuals o a la classe que defineix els elements. Poden ser:

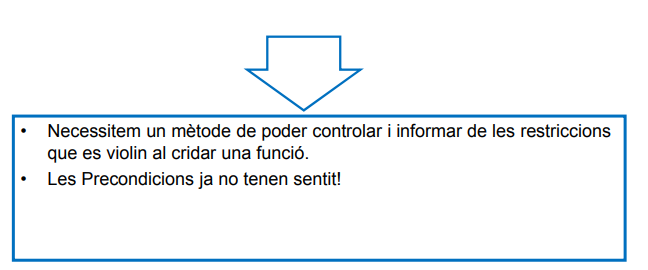
* De classe (estàtic).
  + X està associat al C.
* D’instància (no estàtic).
  + X està associat als objectes de C.

Els atributs o operacions de classe (estàtics) es marquen subratllant el nom de la operació / atribut.

Exemple:



Disseny sense precondicions

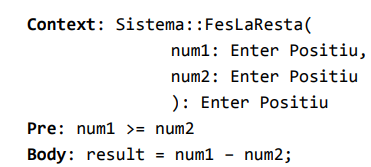
A diferència d’especificació, a disseny no hi ha cap sistema que ens protegeixi de violar restriccions d’integritat, multiplicitats, precondicions o qualsevol protecció que ens doni el diagrama de classes de forma natural.

Excepcions:

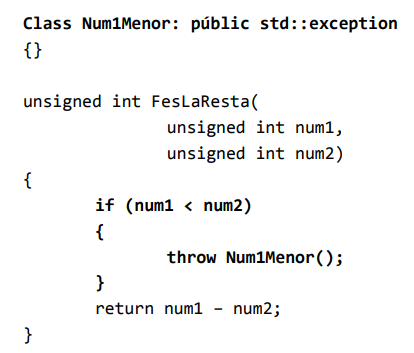
Els llenguatges de programació ens proporcionen diverses maneres de poder controlar els casos en els que la operació no s’ha pogut realitzar per causa d’algun problema. La més utilitzada és mitjançant l’ús d’Excepcions.

Gestió d’Excepcions:

* Segons Wikipedia: La gestió d’excepcions és una tècnica de programació que permet al programador controlar els errors ocasionats durant l’execució d’un programa informàtic. Quan es produeix algun tipus d’error, el sistema reacciona executant un fragment de codi que resolgui la situació, per exemple, retornant un missatge d’error o retornant un valor per defecte.
* Informal: És la manera de controlar i gestionar problemes. Quan es produeixi un problema, llencem una excepció per a notificar-ho, això atura la execució.

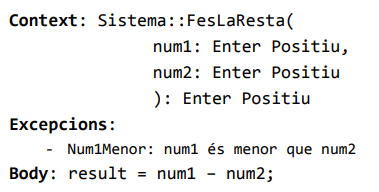


Per traduir això es faria:



Contractes de les Operacions a Disseny:

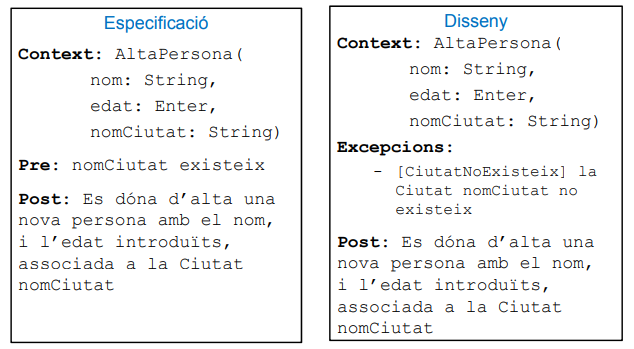
* S’elimina la secció de precondicions.
* S’afegeix una secció d’excepcions:
  + Per cada possible problema que la operació pugui provocar crearem una nova excepció. La definiríem amb un nom explicatiu i una descripció.



Excepcions possibles:

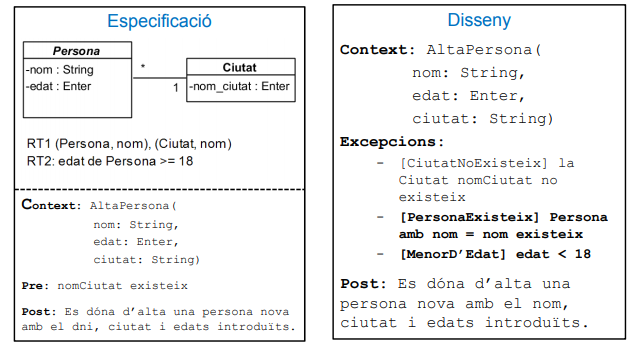
* Les excepcions que es poden produir en una operació poden ser dels següents tipus:
  + Violació d’una precondició.
  + Violació d’una restricció textual.
  + Violació d’una restricció gràfica.
  + Violació d’una restricció del model (implícita).
* Per cada operació haurem de fer una anàlisi i veure quines es poden produir.

Violació d’una precondició:



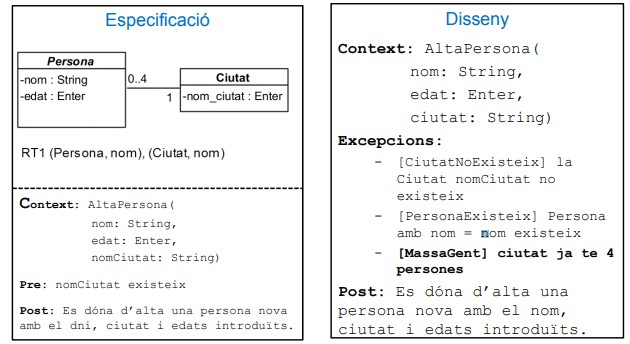
Per cada Precondició que hi hagi a la nostra operació, haurem d’afegir una excepció nova.

Violació d’una restricció textual:



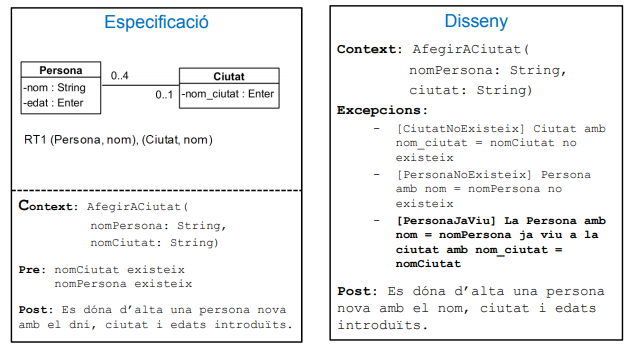
Per cada restricció d’integritat que es pugui violar, haurem d’afegir una nova excepció.

Violació d’una restricció gràfica:



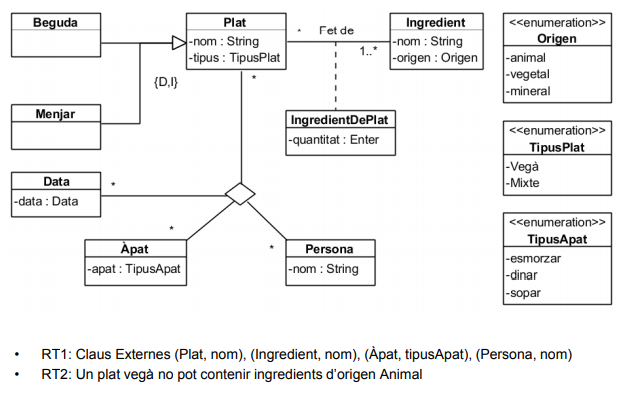
Per cada multiplicitat que es pugui violar, cal una nova excepció.

Violació d’una restricció del model (implícita):

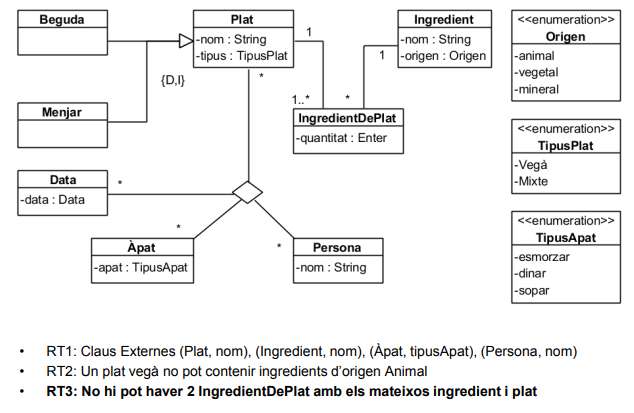


Per cada multiplicitat que es pugui violar, cal una nova excepció.

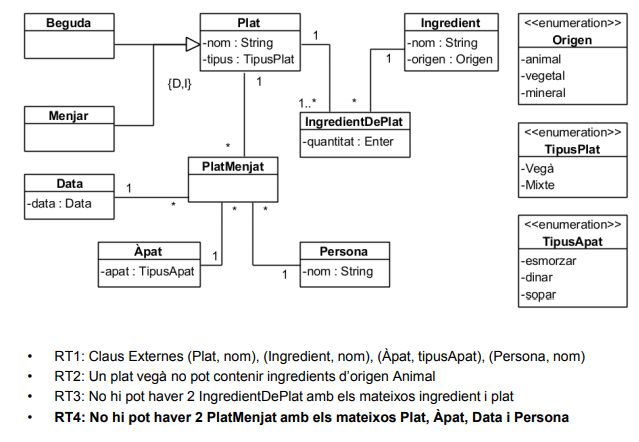
**Exemple de traducció:**



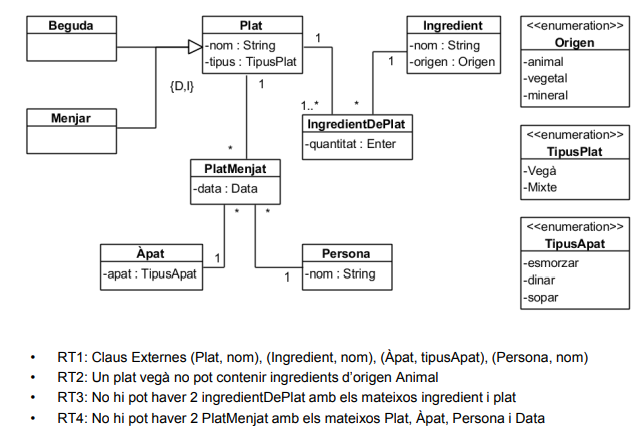
Pas 1: Treure la associativa IngredientDePlat (també s’afegeix una RT nova).



Pas 2: Resoldre la quaternària (s’afegeix una nova RT).



Pas 3: Treure la classe Data.



Pas 4: Resoldre la generalització disjoint.

