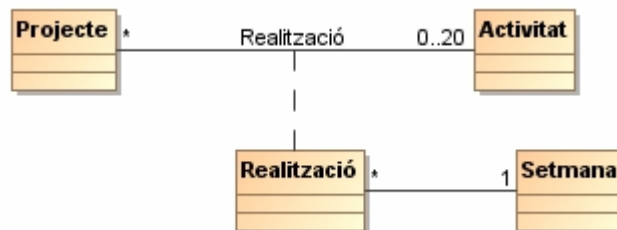


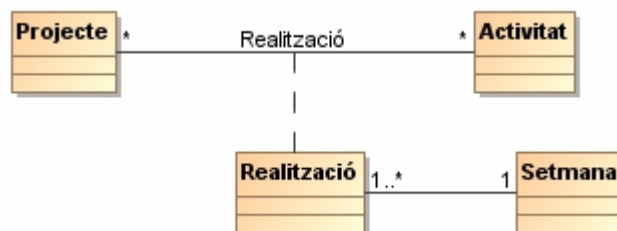
Teoria (3 punts)

1.

- a) Es vol expressar la restricció: “Un projecte inclou com a màxim 20 activitats”. Digueu com modificaríeu el model proposat per a expressar, de la millor manera possible, la restricció mantenint *Realització* com a classe associativa.



- b) Es vol expressar la restricció: “Per tota setmana, hi ha d’haver com a mínim una activitat proposada” i feu el mateix que a l’apartat a).



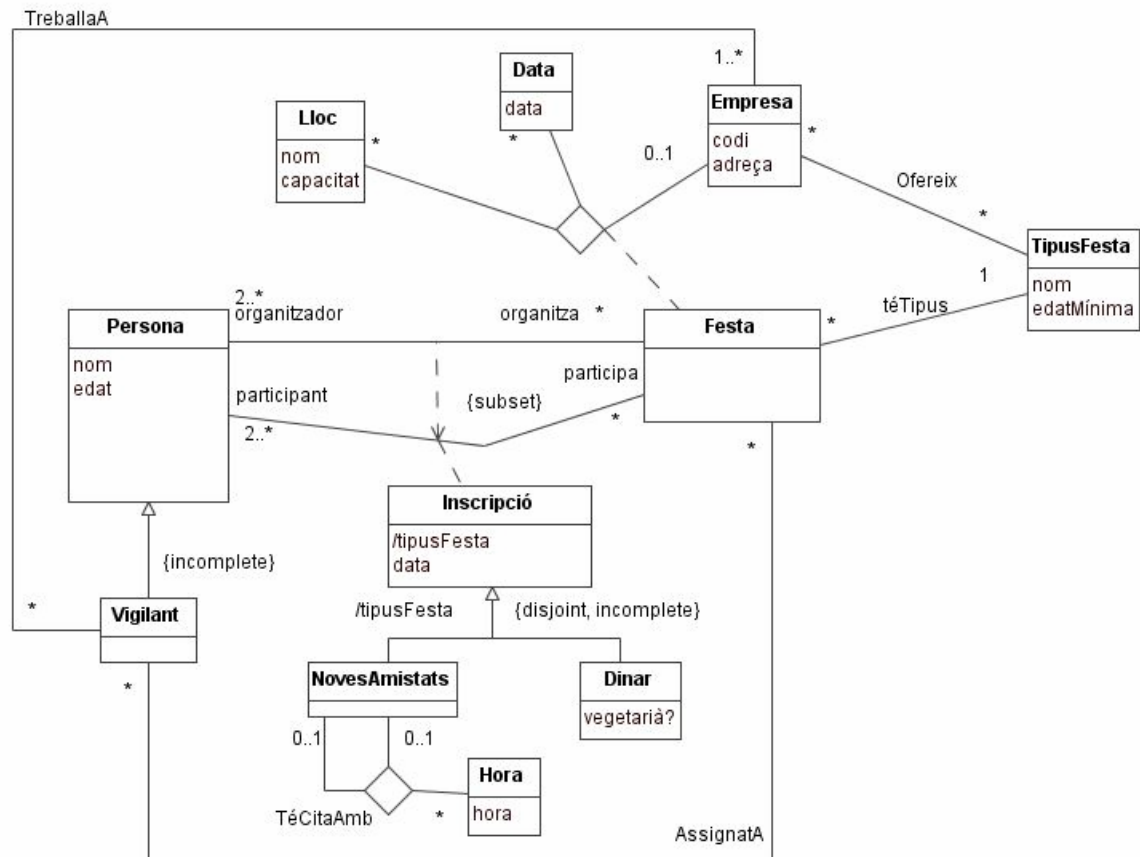
2. Errors:

- La instància de l’associació Té: (#d1,#p1,#p2) haurien de ser dues instàncies (#d1,#p1) i (#d1,#p2)
- Falten dues instància de l’associació Té: (#d2,...) degut a la cardinalitat 2..*
- Falta una instància de l’associació Impartició: (... ,#a3,...) degut a la cardinalitat 1..*
- La instància Pre-requisit no ha de tenir identificador propi, és (#r1) ja que és un subtipus de Requisit
- La instància #r2 també hauria de ser instància de Pre-requisit o Co-requisit (pel fet de ser complete)

3. Errors:

- self.allInstances() → self
- self.allInstances().producte → self.ofereix.producte
- if prod.oclAsType(Plat) → if prod.oclIsTypeOf(Plat)
- self.treballaEn → self.cuiner
- prod.plat → prod.oclAsType(Plat)
- prod.plat.sapCuinar → prod.oclAsType(Plat).cuiner
- isEmpty() → notEmpty()
- prod.oclIsTypeOf(Menu) → prod.oclAsType(Menu)

1r Problema – Model Conceptual de les Dades (3,5 punts)



Informació derivada

1. /tipusFesta d'una Inscripció és el nom del tipus de la festa de la inscripció.

Restriccions textuais

1. Claus: (Lloc, nom), (Tipus de Festa, nom), (Empresa, codi), (Data, data), (Persona, nom), (Hora, hora).
2. Una festa ha de tenir un tipus dels que ofereix l'empresa que l'organitza.
3. La data d'una inscripció ha de ser prèvia a la data de la festa.
4. A un lloc no s'hi poden inscriure més persones que la seva capacitat.
5. Una persona no es pot inscriure a una festa si la seva edat és menor que l'edat mínima del tipus de la festa.
6. Un vigilant no es pot inscriure a una festa organitzada per l'empresa a la que treballa.
7. Una persona inscrita a una festa no es pot citar amb ella mateixa.
8. Una persona inscrita a una festa no es pot citar amb dues persones diferents a la mateixa hora.
9. Una inscripció només pot tenir cites amb inscripcions de la mateixa festa.
10. Un vigilant només pot estar assignat a festes organitzades per una de les empreses on treballa.

2on Problema – Model del Comportament (3,5 punts)

Model del Comportament

Introduir exemplar

Operació: `introduirExemplar(nom: String, sexe: TSexe, espècie: String, cuidadors: Set(String), pares: Set(String), entrenador: String)`

Precondicions:

`Espècie.allInstances()->exists(e | e.nom = espècie)` and
`Cuidador.allInstances().dni->includesAll(cuidadors)` and
`Exemplar.allInstances().nom->includesAll(pares)`
`espècie = 'Dofí'` implies `Entrenador.allInstances()->exists(e | e.dni = entrenador)`

Postcondicions:

`e.oclIsNew()` and
if `espècie = 'Dofí'` then `e.oclIsTypeOf(Dofí)` and `e.entrenador.dni = entrenador`
else `e.oclIsTypeOf(Exemplar)`
endif and
`e.nom = nom` and `e.sexe = sexe` and `e.espècie.nom = espècie` and
`e.cuidador.dni-> includesAll(cuidadors)` and
`e.progenitor.nom-> includesAll(pares)`

Valorar dofins

Operació: `obtenirEspectacle(data: Date, hora: Time): Espectacle`

Precondicions:

`Espectacle.allInstances()->exists(e | e.data.data = data and e.hora.hora = hora)`

Postcondicions: -

Sortida:

`result = Espectacle.allInstances()->select(e | e.data.data = data and e.hora.hora = hora)`

Operació: `valorarDofí(e: Espectacle, nomDofí: String, valoració: TVal)`

Precondicions:

`Dofí.allInstances()->exists(d | d.nom = nomDofí)` and
`not Actuació.allInstances()->exists(a | a.dofí.nom = nomDofí and a.espectacle = e)`

Postcondicions:

`a.oclIsNew()` and `a.oclIsTypeOf(Actuació)` and `a.espectacle = e` and `a.dofí.nom = nomDofí` and `a.valoració = valoració`

Llistar dofins

Operació:

`llistarDofins():`
`Set(TupleType(nom: String,`
`espectacles: Set(TupleType(entrenador: String, valoració: TVal))))`

Precondicions:

`Dofí.allInstances()->forall(d | d.espectacle.entrenador->asSet()-> size()>=2)`

Postcondicions: -

Sortida:

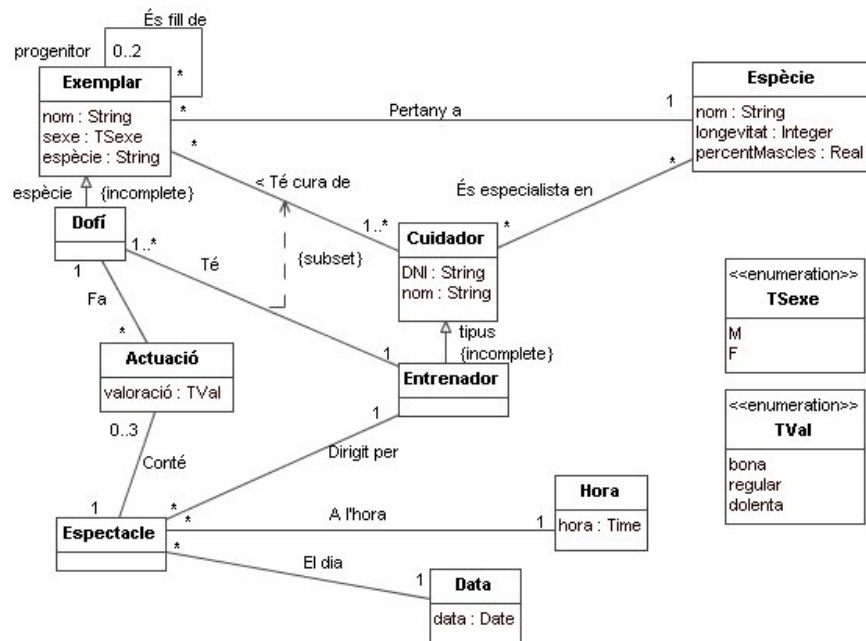
`let dofins: Set(Dofí) = Dofí.allInstances()->select(d | d.actuació-> select(a |`
`a.valoració = 'bona')->size() >3)`
`in`

```

result = dofins-> collect(d |
Tuple {nom = d.nom,
      espectacles = d.actuació->select(a | a.espectacle.entrenador == d.entrenador)-> collect(a |
      Tuple {entrenador = a.espectacle.entrenador.dni,
            valoració = a.valoració}}))

```

Normalització



Restriccions textuais afegides:

- No hi pot haver dues actuacions del mateix dofi en el mateix espectacle
- No hi pot haver dos espectacles en les mateixes data i hora
- Un entrenador no pot dirigir més de dos espectacles en una data

Normalització contractes

```

introduirExemplar(nom: String, sexe: TSexe, especie: String, cuidadors:
Set(String), pares: Set(String), entrenador: String)

```

Precondicions afegides:

- No existeix cap exemplar amb el nom 'nom'
- Els exemplars 'pares' no provoquen un cicle
- Els exemplars 'pares' són de sexe diferent
- Els exemplars 'pares' són de l'especie 'especie'
- Hi ha com a màxim 2 elements a 'pares' (cardinalitat 0..2)
- Tots els 'cuidadors' són especialistes en l'especie 'especie'
- Hi ha com a mínim un element a la llista 'cuidadors' (cardinalitat 1..*)
- Si 'especie' és 'Dofí', l'entrenador és un dels 'cuidadors' (subset)

Postcondicions afegides:

- L'atribut 'especie' del nou exemplar pren el valor del paràmetre 'especie'
- S'actualitza el valor de l'atribut 'percentMasclles' de l'especie 'especie'