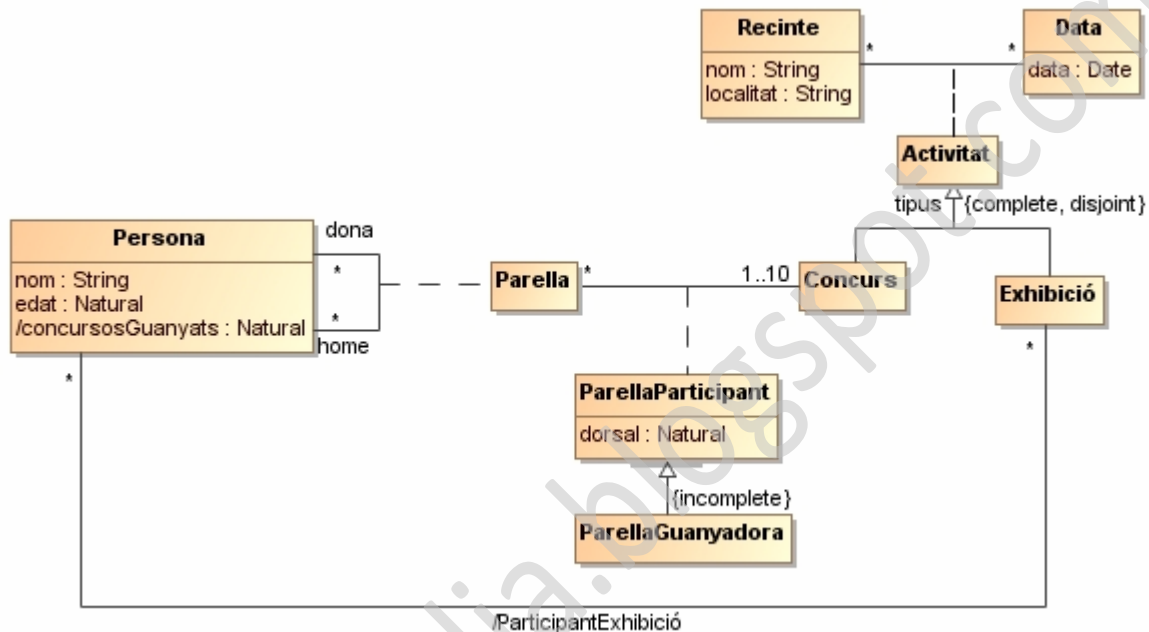


Model del Comportament

Una escola de ball organitza dos tipus d'activitat, concursos i exhibicions de ball. Les activitats s'organitzen en una data i un recinte determinat, del que en sabem el nom que l'identifica i la localitat on es troba. El sistema també emmagatzema el nom i l'edat de les persones que, per parelles, participen a un concurs, i se'ls hi assigna un número de dorsal al concurs. De cada concurs també s'enregistren les parelles guanyadores. A les exhibicions hi participen totes les persones que hagin estat guanyadores en algun concurs de l'any anterior a l'any de la celebració de l'exhibició.



Restriccions textuais

1. Claus externes: (Persona, nom), (Recinte, nom), (Data, data)
2. Els membres d'una parella han de ser diferents
3. Una persona no pot participar més d'una vegada al mateix concurs (com a membre de parelles diferents)
4. Una persona no pot participar a més d'un concurs i/o exhibició en la mateixa data
5. En un concurs no hi ha dorsals repetits
6. En un mateix dia no hi pot haver dues exhibicions

Informació derivada

1. /concursosGuanyats d'una Persona és el nombre de concursos en que la persona ha format part d'una parella guanyadora
2. Les persones que participen a una exhibició són tots els que han estat guanyadors d'algun concurs de l'any anterior a l'any de l'exhibició

El sistema a desenvolupar no ha de donar d'alta Recintes, Dates, ni Persones atès que hi ha un altre sistema encarregat de fer-ho. No obstant, ha de proporcionar les funcionalitats següents: crear una nova activitat, enregistrar els guanyadors d'un concurs i llistar les activitats organitzades amb més de 50 participants en un determinat recinte.

Quan un usuari vol crear una nova activitat, indica al sistema el nom del recinte, la data en que s'organitza i si es tracta d'un concurs o d'una exhibició. A més caldrà, en cas de ser un concurs, indicar les dades de les parelles que s'inscriuen al concurs i el número de dorsal que se'ls hi assigna. Només es poden donar d'alta activitats si la seva data pertany a l'any actual. Feu que la interacció necessària per portar a terme aquesta funcionalitat requereixi més d'un esdeveniment.

Al finalitzar un concurs, qualsevol membre del jurat enregistra les parelles guanyadores. En aquest cas, indica al sistema les dades necessàries per identificar el concurs i les parelles guanyadores i el sistema ho enregistra. Feu que la interacció necessària per a portar a terme aquesta funcionalitat requereixi únicament un esdeveniment. Cal tenir en compte que, en qualsevol cas, aquesta operació només es du a terme si la data del concurs és igual a la data actual.

Quan un usuari demana el llistat de les activitats amb més de 50 persones en un determinat recinte, indica al sistema el nom del recinte. El sistema retorna, per a cada activitat, amb més de 50 persones, realitzada en el recinte, la data en que s'ha fet i el nombre total de participants. En cas de que l'activitat sigui una exhibició també es mostra, per cada participant en l'exhibició, el seu nom, l'edat i el nombre de concursos guanyats.

Es demana:

- *Model del Comportament del Sistema*: Diagrames de seqüència de totes les funcionalitats especificades i contractes en OCL de totes les operacions que hi apareixen.

Creació de nova activitat

Operació: novaActivitat (nom:String, data:Data, tipus:String): Activitat

Semàntica: Crea una nova instància d'Activitat amb les dades indicades

Precondició:

- Recinte.allInstances()->exists(r|r.nom=nom)
- Data.allInstances()->exists(d|d.data=data)
- not Activitat.allInstances()->exists(ac|ac.recinte.nom=nom and ac.data.data=data)
- data.year()==today().year()

Postcondició: ac.oclIsNew() and
if tipus='Concurs' then ac.oclIsTypeOf(Concurs)
else ac.oclIsTypeOf(Exhibició) endif and
ac.recinte.nom=nom and ac.data.data=data

Sortida: result = ac

Operació: inscriureParella(activitat:Activitat, nomHome:String, nomDona:String, numDorsal:Natural)

Semàntica: Crea una instància de parella (si no existia) i una instància de l'associació entre parella i concurs

Precondició:

- activitat.oclIsTypeOf(Concurs)
- Persona.allInstances()->exists(p|p.nom=nomHome)
- Persona.allInstances()->exists(p|p.nom=nomDona)

(nota: no es comprova que no existeix ja la parella participant ja que és redundant amb la Restricció Textual 3)

Postcondició: not (Parella.allInstances()->exists(par| (par.dona.nom=nomDona and par.home.nom=nomHome))) implies
(par.oclIsNew() and par.oclIsTypeOf(Parella) and
par.dona.nom=nomDona and par.home.nom=nomHome) and
part.oclIsNew() and part.oclIsTypeOf(ParellaParticipant) and
part.dorsal=numDorsal and
part.concurs=activitat.oclAsType(Concurs) and
part.parella.dona.nom=nomDona and part.parella.home.nom=nomHome

Enregistrar guanyadors

Operació: enregistrarGuanyadors(nomRecinte:String, data:Date, dorsals:Set(Natural))

Semàntica: Enregistra els guanyadors d'un concurs.

Precondició:

- Concurs.allInstances()->exists(co|co.recinte.nom=nom and co.data.data=data and co.parellaParticipant.dorsals->includesAll(dorsals))
- data=today()

Postcondició: - ParellaGuanyadora.allInstances()->select(pg|
pg.concurs.recinte.nom = nomRecinte and
pg.data.data=data).dorsal->includesAll(dorsals))

Llistar activitats amb molta participació

Operació: `l·listarActivitatsMoltParticipades(recinte:String):
Set(TupleType(data:Date, nombre:Natural,
participants:Set(TupleType(nom:String, edat:Natural,
concursos: Natural))))`

Sortida: `let activitats:Set(Activitat) = Activitat.allInstances()->
select (ac| ac.recinte.nom=recinte and
(ac.oclAsType(Concurs).parella-> size())>25 or
ac.oclAsType(Exhibició).persona->size())>50)) in
result = activitats->collect(act| Tuple{data = act.data.data,
nombre = if act.oclIsTypeOf(Concurs) then
act.oclAsType(Concurs).parella->size()*2 else
act.oclAsType(Exhibició).persona->size() endif, participants =
if act.oclIsTypeOf(Exhibició) then
act.oclAsType(Exhibició).persona->collect(pe |
Tuple{nom=pe.nom,edat=pe.edat,concursos=pe.concursosGuanyats})
else Set{} endif })`

El sistema a desenvolupar no ha de donar d'alta Recintes, Dates, ni Persones atès que hi ha un altre sistema que ho fa. No obstant, ha de ser capaç de crear una nova activitat, enregistrar els guanyadors d'un concurs i llistar els concursos massius fets a un recinte.

Quan un usuari vol crear una nova activitat, indica al sistema tota la informació necessària per fer-ho. Si l'activitat és un concurs, caldrà indicar també les parelles que participen al concurs i el seu dorsal. En aquest darrer cas, caldrà que es creïn les parelles que no existien prèviament. Feu que la interacció per portar a terme aquest cas d'ús requereixi més d'un esdeveniment.

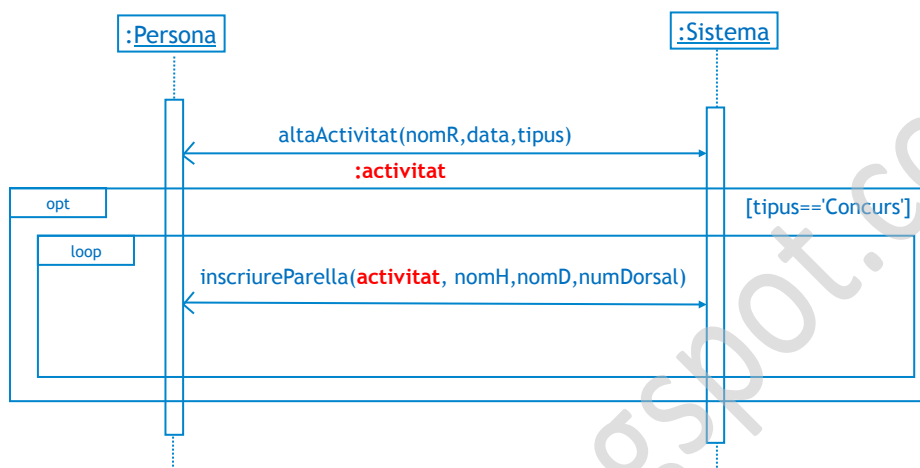
Un membre del jurat pot enregistrar les parelles guanyadores. En aquest cas, indica al sistema les dades necessàries per identificar el concurs i els dorsals de les parelles guanyadores. Feu que la interacció per portar a terme aquest cas d'ús requereixi més d'un esdeveniment.

Quan un usuari demana el llistat dels concursos massius d'un recinte, indica al sistema el nom del recinte. El sistema retorna, per a cada concurs realitzat al recinte amb més de 50 persones participants i tal que a la data del concurs no es va realitzar cap exhibició, la data en que s'ha fet i el nombre total de persones que hi ha participat.

Us demanem que feu mitjançant la notació UML:

- *Model del Comportament*: diagrames de seqüència de totes les funcionalitats especificades i contractes en OCL de totes les operacions que apareixen a aquests diagrames.

Cas d'us: Nova Activitat (2 events)



Cas d'us: Nova Activitat (2 events)

context: Sistema::altaActivitat(nomR:String, data:Data, tipus:String):
Activitat

Pre:

Recinte.allInstances()->exists(r|r.nom=nomR)

Post:

```

Activitat.allInstances()->exists(ac| ac.oclIsNew()
    and ac.recinte.nom=nomR
    and ac.data.data=data
    and
        if tipus='Concurs' then ac.oclIsTypeOf(Concurs)
        else ac.oclIsTypeOf(Exhibició)
        endif
    and result=ac)
  
```

Cas d'us: Nova Activitat (2 events)

context: Sistema::inscriureParella(a:Activitat, nomH:String,
nomD:String, numDorsal:Natural)

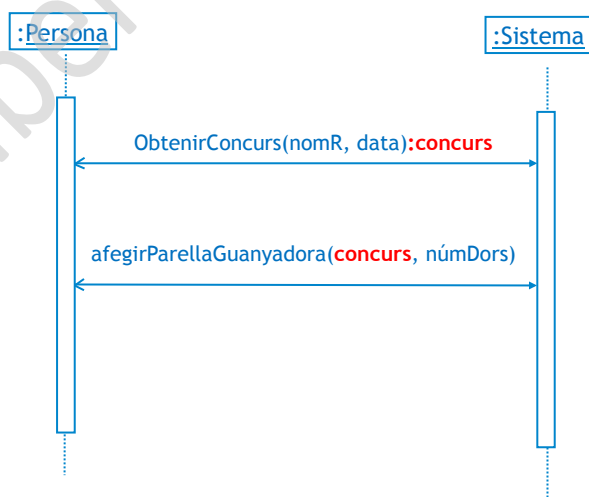
pre:

~~a.ocIsTypeOf(Concurs)~~ (redundant amb el DSS)
and Persona.allInstances()->exists(p|p.nom=nomH)
and Persona.allInstances()->exists(p|p.nom=nomD)

post:

not(Parella.allInstances()@pre->exists(par| par.dona.nom=nomD
and par.home.nom=nomH))
implies(Parella.allInstances()->exists(par| par.ocIsNew()
and par.dona.nom=nomD
and par.home.nom=nomH)
and
ParellaParticipant.allInstances()->exists(pp| pp.ocIsNew()
and pp.dorsal=numDorsal
and pp.concurs=a
and pp.parella.dona.nom=nomD
and pp.parella.home.nom=nomH)

Cas d'us: Enregistrar guanyadors (2 events)



Cas d'us: Enregistrar guanyadors (2 events)

context: Sistema::obtenirConcurs(nomR:String, data:Date): Concurs

pre:

```
Concurs.allInstances()->exists(co| co.recinte.nom=nomR
                                and co.data.data=data)
```

body:

```
Concurs.allInstances()->select(co| co.recinte.nom=nomR
                                and co.data.data=data)
```

Cas d'us: Enregistrar guanyadors (2 events)

context: Sistema::afegirParellaGuanyadora(c:Concurs, númDors:Natural)

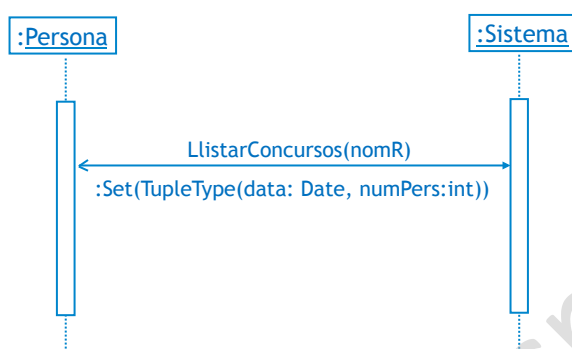
pre:

```
c.parellaParticipant.dorsal->includes(númDors)
```

post:

```
c.parellaParticipant->exists(pp|pp.dorsal=númDors
                              and pp.oclIsTypeOf(ParellaGuanyadora))
```


Cas d'us: Llistar concursos massius (1 event)



Cas d'us: Llistar concursos massius (1 event)

```

context: Sistema::l1listarConcursos(nomR:String) :
    Set(TupleType(data: Date, númPers: Integer))

pre:
    Recinte.allInstances()->exists(r|r.nom=nomR)

body:
    let concs:Set(Concursos) =
        Concurs.allInstances()->select(c|c.parella->size()->25
            and c.data.activitat.oclaAsType(Exhibició)->isEmpty()
            and c.recinte.nom=nomR)
    in
        concs -> collect(c |
            Tuple{data = c.data.data, númPers=c.parella->size()*2})

```