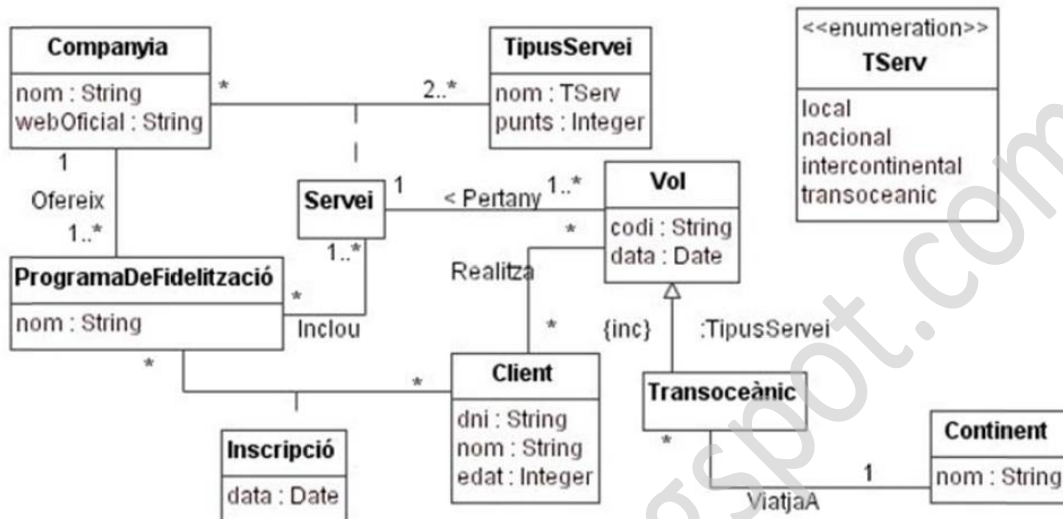


7è problema d'IES – 20 o 21 d'abril del 2015 – QP1415

Considereu un sistema per un consorci de companyies aèries. Les companyies associades disposen de diversos tipus de servei (local, nacional, intercontinental i transoceànic) i ofereixen programes de fidelització als seus clients per a què s'hi inscriguin. S'enregistren també els vols que pertanyen a un servei i els clients que realitzen els vols. La part rellevant de l'esquema conceptual de les dades en UML es mostra a la figura següent:



Restriccions textuais

1. Claus externes: (Companyia, nom), (TipusServei, nom), (Continent, nom), (Vol, codi+ data), (Client, dni)
2. Una companyia no pot oferir dos o més programes de fidelització amb el mateix nom
3. Els serveis inclosos a un programa de fidelització són de la companyia que ofereix el programa
4. Un client no es pot inscriure a més d'un programa de fidelització de la mateixa companyia

Suposarem que el sistema a desenvolupar no ha de donar d'alta instàncies de les classes Companyia, TipusServei, ProgramaDeFidelització, Continent ni Client atès que hi ha un altre sistema que ho fa. El sistema ha de permetre efectuar les funcionalitats següents: alta de servei i vols, alta d'inscripció i consulta de clients sènior actius.

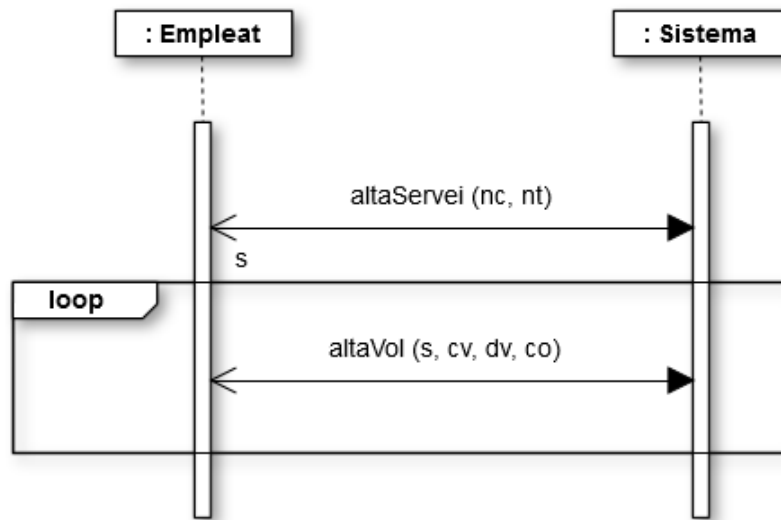
AltaServeiIvols: Quan un directiu vol donar d'alta un servei i els seus vols li demana a un empleat del consorci que ho faci. L'empleat indica al sistema la informació necessària per donar d'alta un servei. A continuació, per cada vol que ha de pertànyer al servei, l'empleat dóna la informació necessària per donar d'alta el vol. Si es tracta d'un vol transoceànic, indica també el continent on viatja. Cal considerar que aquesta funcionalitat només es pot realitzar si la companyia del servei té menys de 20 programes de fidelització. Feu que la interacció per portar a terme aquesta funcionalitat requereixi més d'un esdeveniment.

AltaInscripció: Quan un client es vol inscriure a un programa de fidelització, ell mateix indica al sistema la informació necessària per identificar el programa de fidelització, el seu dni i la data de la inscripció. Només es pot fer aquesta funcionalitat si el client té com a mínim un vol que pertanyi a un servei inclòs al programa de fidelització on es vol inscriure i si la data de la inscripció és posterior o igual a l'actual. Feu que la interacció requereixi un sol esdeveniment.

ConsultaClientsSèniorActius: Quan un empleat vol fer una consulta de clients sènior actius, ell mateix introdueix al sistema un nom de companyia. El sistema mostra el llistat dels clients que tenen més de 40 anys i que han fet més de 3 vols transoceànics que pertanyen a serveis de la companyia donada. El llistat resultant mostra per cada client el dni i les dates de tots els vols que té enregistrats al sistema. Aquesta funcionalitat només s'executa si la companyia existeix.

Es demana que feu mitjançant la notació UML:

- Diagrames de seqüència i contractes en OCL de les operacions corresponents a les funcionalitats anteriors.



context: Sistema::altaServei(nc:String, nt:TServ):Servei

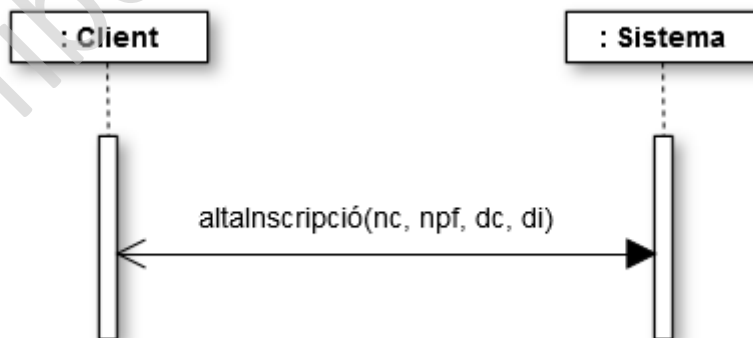
pre: Companyia.allInstances()->exists(c|c.nom=nc and c.programaDeFidelització->size()<20) and TipusServei.allInstances()->exists(t|t.nom=nt)

post: Servei.allInstances()->exists(s|s.oclIsNew() and s.companyia.nom=nc and s.tipusServei.nom=nt and result=s)

context: Sistema::altaVol(s:Servei, cv:String, dv:Date, co:String)

pre: s.tipusServei.nom=TServ::transoceànic implies Continent.allInstances()->exists(c|c.nom=co)

post: Vol.allInstances()->exists(v|v.oclIsNew() and v.servei=s and v.codi=cv and v.data=dv and if s.tipusServei.nom=TServ::transoceànic then v.oclAsType(Transoceànic).continent.nom=co endif)



context: Sistema::altaInscripció(nc:String, npf:String, dc:String, di:Date)

pre: di>=today() and Client.allInstances->select(cl|cl.dni=dc).vol.servei.programaDeFidelització->includes(Companyia.allInstances()->select(c|c.nom=nc).programaDeFidelització->select(pf|pf.nom=npf)))

post: `Inscripció.allInstances()->exists(i|i.oclIsNew() and
i.programaDeFidelització.nom=npf and
i.programaDeFidelització.companyia.nom=nc and
i.client.dni=dc and i.data=di)`



context: `Sistema::consultarClients(nc:String) :
Set(TupleType(dnic: String, datesc: Set(Date)))`

pre: `Companyia.allInstances()->exists(c|c.nom=nc)`

body: `let clients:Set(Client)=Client.allInstances()->
select(c|c.edat>40 and c.vol->select
(v|v.servei.companyia.nom=nc and
v.oclIsTypeOf(Transoceanic))->size(>3) in
clients->collect(c|Tuple{dnic=c.dni,
datesc=c.vol.data})`