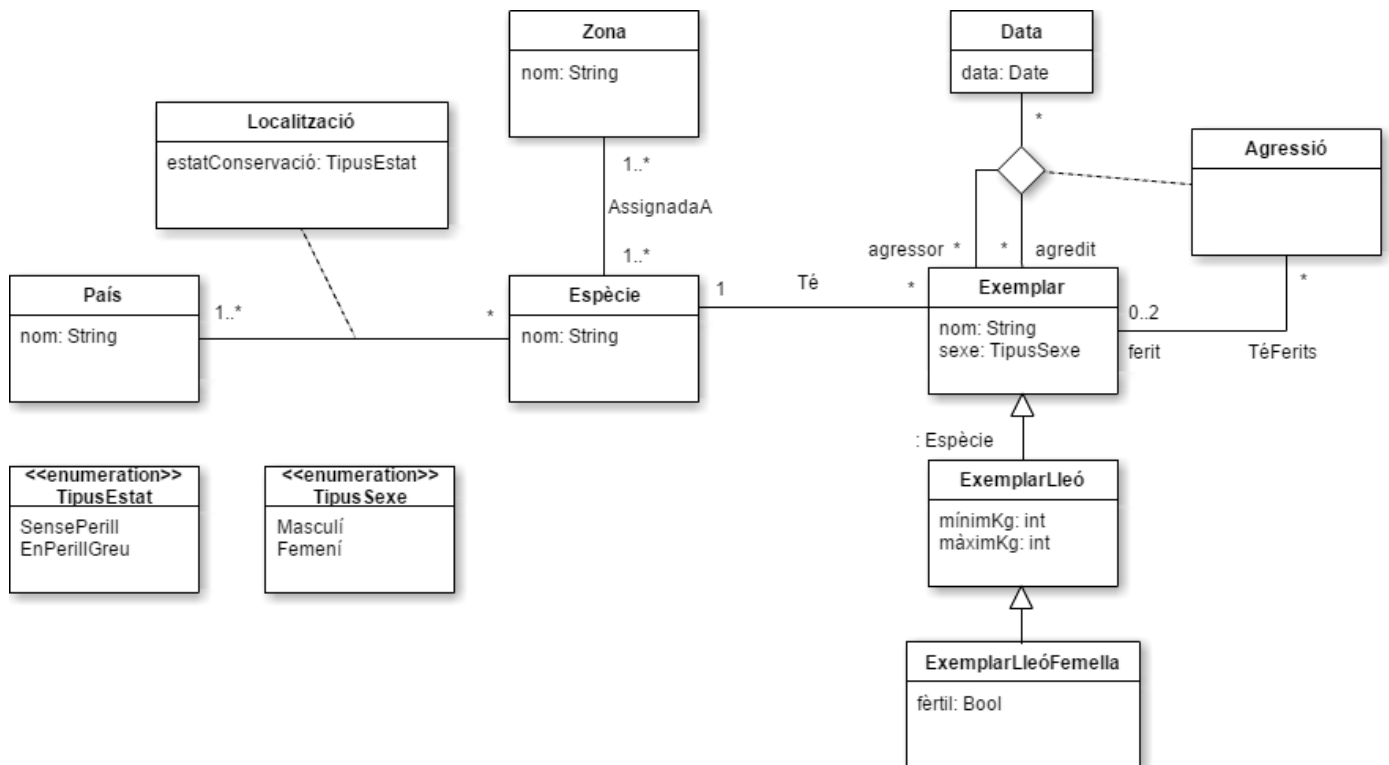


8a entrega d'IES, QP1718 – solució comportament (problema zoològic)

Un zoo està interessat en un sistema software per gestionar els seus animals. En concret, el zoo vol registrar diferents espècies animals. L'esquema conceptual d'aquest sistema és el següent:



Restricciones Textuales:

Claus Externes: (Espècie, nom), País(nom), (Exemplar, nom), (Zona, nom), (Data, data)

1. mínimKg <= màximKg.
2. Una instància de ExemplarLleóFemella és un exemplar de sexe Femella.
3. Els ferits d'una agressió són o bé l'agressor, o bé l'agredit.
4. Un exemplar no pot agredir-se a si mateix.

El sistema a desenvolupar no ha de donar d'alta País ni Agressió ja que hi ha un altre sistema encarregat de fer-ho. En canvi, ha de proporcionar les funcionalitats següents:

Alta Exemplar: Quan el director del zoo vol donar d'alta a un exemplar d'una espècie, ell mateix introdueix la informació necessària per fer-ho. És a dir, tota la informació requerida per a qualsevol exemplar. Aquesta funcionalitat només pot realitzar-se si l'espècie de l'exemplar ja existeix, i aquesta espècie està assignada a almenys una zona que té, com a mínim, dues espècies assignades.

Alta Zona: Quan l'encarregat d'infraestructures del zoo vol donar d'alta una zona, introdueix la informació de la zona, i la informació de les espècies que hi haurà assignades. S'ha de considerar que si l'espècie no existeix en el sistema, s'ha de crear. Feu que la interacció per dur a terme aquesta funcionalitat requereixi més d'un esdeveniment.

ConsultaAgressionsRellevants: Quan el veterinari principal del zoo vol consultar les agressions rellevants d'un exemplar, ell mateix introdueix el nom de l'exemplar agressor. El sistema mostra informació de cada agressió en la què participa aquest exemplar com a agressor, i on l'exemplar agredit és d'una espècie que està “EnPerillGreu” en almenys un país. En concret, el sistema mostra, per cadascuna d'aquestes agressions, la data de l'agressió, i el nom dels exemplars que en varen resultar ferits. Aquesta funcionalitat només es pot realitzar si l'exemplar agressor és d'una espècie que està “SensePerill” en tots els països en què es localitza.

Es demana que feu mitjançant la notació UML:

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

Solució:

Cas d'Ús AltaExemplar

Context altaExemplar(nom:String, sexe:TipusSexe, minimKg:Int, maximKg:Int, fétil:Bool, nomEsp:String)

Pre: Espècie.allInstances()->exists (e | e.nom=nomEsp and e.zona->exist(z| z.especie->size())>=2))

Post: Exemplar.allInstances()->exists (ex|ex.ocllsNew() and ex.nom=nom and ex.sexe=sexe

and ex.especie.nom=nomEsp and

If ex.especie.nom="Lleó" then

ex.ocllsTypeOf(ExemplarLleó) and ex.ocllsType(ExemplarLleó).minimKg=minimKg and

ex.OclAsType(ExemplarLleó).maximKg=maximKg and

If ex.sexe:TipusSexe::Femella then

ex.ocllsTypeOf(ExemplarFemella) and ex.ocllsType(ExemplarFemella).fétil=fétil

Endif

Endif)

Caso d'Ús Alta Zona

Context altaZona(nomZ:String): z:Zona

Post: Zona.allInstances()->exists(z|z.ocllsNew() and z.nom=nomZ and result= z)

Context registraEspecie(z:Zona, nomE:String): e:Espècie

Post: if not(Especie.allInstances()@pre->exists (e|e.nom=nomE)) then

especie.allInstances()->(exists e|e.ocllsnew() and e.nom=nomE and result=e)

endif and

z.especie.nom->includes(nomE)

Context AltaLocalització(e:Espècie, nomP:String, estatCons:TipusEstat)

Pre: País.allInstances()->exists(p|p.nom=nomP)

Post: Localització.allInstances()->exists (l| l.ocllsNew() and l.estatConservacio=estatCons and
l.pais.nom=nomP and l.especie=e))

Caso d'Ús Consulta AgressionsRellevants

Context agressionsRellevants (nomEx:String.): Set(TupleType(dateAgressio:Date, nomFerits:Set(String)))

Pre: Exemplar.allInstances()->exist (e|e.nom=nomEx and

not (e.especie.localitzacio.estatConservació->includes ("EnPerillGreu"))

Body:

Let agressioRellevant:Set(Agressió)= Agressió.allInstances()->select

(a|a.agressor.nom=nomExemplar and a.agredit.especie.localització.estatConservació-

>

includes("EnPerillGreu"))

in

agressioRellevant->collect(r|r Tuple{ dateAgressio=r.data.data, nomFerits= r.ferit.nom })