

Ejercicio de modelo conceptual - OCL

En una región se determinó realizar un concurso de bandas de música.

Las competencias están categorizadas por un género musical, y para algunos géneros preestablecidos, sus competencias asociadas están subcategorizadas por rango de edades de sus integrantes. No pueden existir categorías sin rangos edades de géneros musicales que deben aceptar rangos de edades.

Las bandas tienen un nombre y representan a una comuna de la región. No pueden existir dentro de una misma competencia dos bandas que representen a la misma comuna.

Las bandas están conformadas por músicos. Los músicos pueden conformar más de una banda. También tienen un nombre.

Una banda puede presentarse a más de una competencia, y en todos estos casos lo hará siempre con sus mismos músicos integrantes. Si una banda se presentara a una competencia subcategorizada por rangos de edades, entonces todos sus integrantes deben tener su edad válida en ese rango establecido. Una banda no se puede presentar en más de una competencia ligada al mismo género musical.

Los géneros musicales aceptan solamente el uso de algunos instrumentos musicales. Esto significa que las bandas que compitan en ese género, sus integrantes podrán utilizar solamente instrumentos permitidos en esa competencia. Cada integrante de la banda tocará uno o más instrumentos musicales válidos dentro del género de la competencia, pero podrá tocar otros instrumentos en otras competencias en que se presente la banda.

Los instrumentos están clasificados por un tipo (de viento, cuerdas, etc).

Se pide lo siguiente:

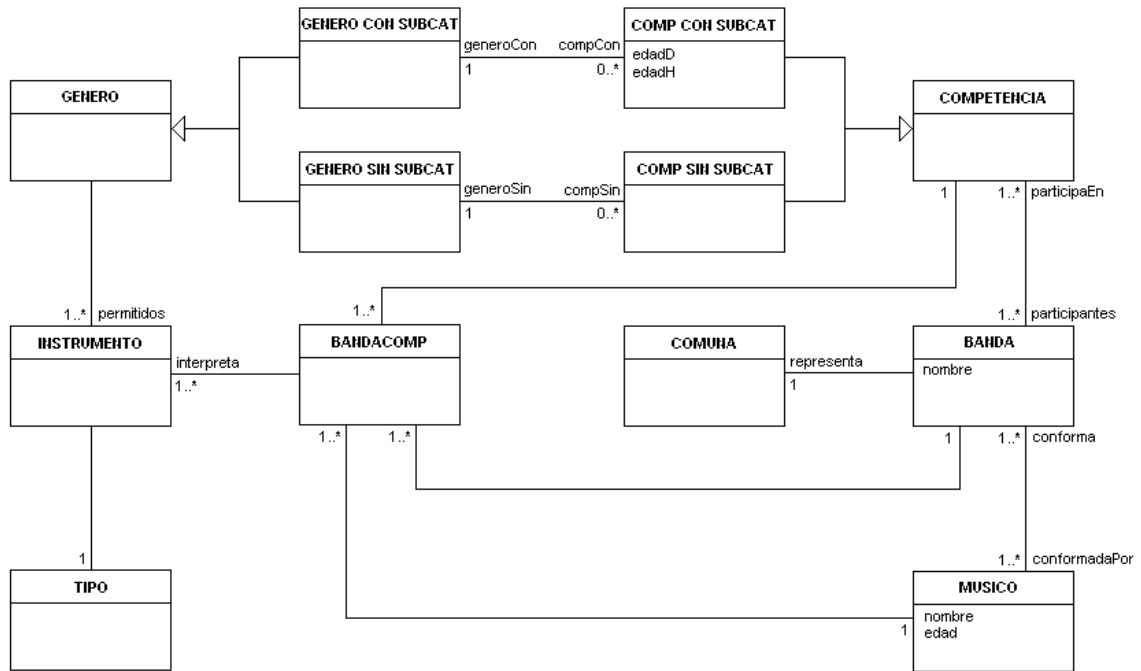
a) Realizar el diagrama de clases conceptuales que refleje el modelo conceptual de todo lo expresado anteriormente.

b) Expresar en OCL todas las restricciones expresadas anteriormente, y además lo siguiente:

- Un atributo derivado 'cantInstrumentos', que dado un músico, retorne la cantidad total de instrumentos distintos que éste toca en las distintas competencias de las bandas que pertenece.

- Un query 'instrumentosComunes', que dado un género musical, obtenga los instrumentos de un tipo dado que hayan sido usados por todas las bandas que compiten en ese género.

Resolución:



En este diagrama no se especifica el atributo derivado 'cantInstrumentos' ni el query 'instrumentosComunes'.

Auxiliar 1) Género de una competencia

```
context: Competencia
def: genero: Genero = if self.OCLisTypeOf(CompConSubcat) then
    self.OCLasType(CompConSubcat).generoCon
    else
    self.OCLasType(CompSinSubcat).generoSin
    endif
```

Auxiliar 2) Competencias de un género

```
context: Genero
def: competencias: Set(Competencia) =
    if self.OCLisTypeOf(GeneroConSubcat) then
        self.OCLasType(GeneroConSubcat).compCon
    else
        self.OCLasType(GeneroSinSubcat).compSin
    endif
```

1) Bandas en competencia son de distintas comunas

```
context: Competencia
inv: self.participantes->
    forAll(b1, b2 | b1 <> b2 implies b1.representa <> b2.representa)
```

2) No pueden existir dos instancias distintas de BandaComp con la misma combinación Competencia-Banda-Músico

```
context: BandaComp
inv: BandaComp.allInstances()->forAll(bc | bc <> self implies
    bc.Competencia <> self.Competencia or
    bc.Banda <> self.Banda or
    bc.Musico <> self.Musico)
```

3) Toda instancia de BandaComp debe corresponderse con una asociación real existente Competencia-Banda-Músico

```
context: BandaComp
inv: self.Competencia.participantes->includes(self.Banda) and
    self.Banda.conformadaPor->includes(self.Musico)
```

4) Toda asociación real existente Competencia-Banda-Músico debe corresponderse con una instancia de BandaComp

```
context: Competencia
inv: self.participantes->forAll(b | b.conformadaPor->forAll(m |
    BandaComp.allInstances->select(bc | bc.Competencia = self and bc.Banda =
    b and bc.Musico = m)->size() = 1))
```

5) Músicos en rangos de edades

```
context: CompConSubcat
inv: self.participantes.conformadaPor->
    forAll(m | self.edadD <= m.edad and m.edad <= self.edadH)
```

6) Bandas en sólo una competencia de género

```
context: Genero
inv: let ps: Bag(Banda) = self.competencias.participantes in
    ps->forAll(b | ps->count(b) = 1)
```

7) Instrumentos válidos

context: Genero

inv: self.permitidos->

includesAll(self.competencias.BandaComp.interpreta->asSet())

8) Atributo derivado 'cantInstrumentos'

context: Musico::cantInstrumentos: Integer

derive: self.BandaComp.intepreta->asSet()->size()

9) Query 'instrumentosComunes'

context: Genero::instrumentosComunes(t: Tipo): Set(Instrumento)

pre: Tipo.allInstances()->includes(t)

body: self.permitidos->

select(i | i.Tipo = t and self.competencias->forAll(c |
c.participantes->forAll(b | BandaComp.allInstances()->select(bc |
bc.Competencia = c and bc.Banda = b).interpreta->includes(i))))