

Músicos - 2025 - Parcial Funcional

Necesitamos modelar requerimientos para una importante personalidad del ámbito musical, que nos contó que un músico tiene un nombre, el grado de experiencia (un número que va de 1 a 10), su instrumento favorito y por último el historial de actuaciones que consiste en una lista con la fecha (día, mes y año) y la cantidad de público que asistió.



Punto 1: Informer

En este punto no puede utilizar funciones auxiliares, solo composición y aplicación parcial. No puede utilizar recursividad.

- Queremos saber si un músico tuvo actuaciones donde hubo más de 5.000 asistentes
- Queremos saber cuántas actuaciones tuvo un músico en un año en particular

Punto 2: Radioactivity

Existen varias actividades que pueden hacer los músicos

- Tocar un instrumento: le cambia el instrumento favorito al músico y además le suma 1 a su experiencia si el instrumento es "oboe", "fagot" o "cello"
- Cantar, que no cambia el instrumento principal pero le agrega como prefijo "Lalala" al nombre porque lo pone contento
- Hacer una presentación, que le agrega una presentación al historial del músico para una fecha y un público, obviamente al final de esa lista. Eso le suma 1 a su experiencia y si el músico toca el "oboe" o el "fagot" o el "cello", suma 1 punto más de experiencia
- Pensar, que es una actividad muy importante, pero que no produce cambios en el músico

Indicar cómo podemos hacer que un músico

- toque el fagot
- cante
- haga una presentación para el día 2 de agosto de este año con 300 asistentes
- piense

Considere que el músico tenga inicialmente una experiencia de 5 y al piano como su instrumento favorito.

Punto 3: Mucha experiencia

Queremos saber cuánta experiencia le hace ganar a un músico una lista de actividades. Por ejemplo: tocar el oboe, cantar y hacer una presentación para 500 personas debería darle 3 puntos extra de experiencia a los que ya tenía el músico.

En este punto no puede utilizar recursividad.

Punto 4: Obsesionario en la mayor

Queremos saber si un músico tiene presentaciones correctas, esto significa que la cantidad de público en las presentaciones que están en la posición impar es impar, y la cantidad de público en las presentaciones que están en la posición par son pares.

Algunos ejemplos:

- sin presentaciones, el músico tiene presentaciones correctas
- si la secuencia de presentaciones es 501, 2000, 755 el músico tiene presentaciones correctas
- si la secuencia de presentaciones es 501, 2000, 755, 1000 el músico tiene presentaciones correctas
- si la secuencia de presentaciones es 501, 2001, 755 el músico no tiene presentaciones correctas

En este punto debe utilizar recursividad.

wor filgo whiteboard 058
 aca data Musico = Musico {
 nombre :: String,
 gradoXP :: Int,
 instrumento :: String,
 actuaciones :: [Actuacion]
 }
 Heja 1/2
 Maximino Torrado
 Docente: Raúl González
 9 (Nueve)

en data Actuación = Actuación {
 ev fecha :: (Int, Int, Int),
 ev cantidad :: Int
 }
 3) a) masDe5000 :: Musico → Bool
 masDe500 = any(> 5000). map cantidad. actuaciones
 ↗ Es poco declarativo
 b)
 cuentas ActuacionesEn :: Int → Musico → Int
 cuentas ActuacionesEn año musico = length. filter(== año). map año. map fecha. actuaciones \$ musico
 ↗ Es poco declarativo. Delegar
 año :: (a, a, a) → a
 año (-, -, añozo) = añozo

2) a)
 tocar Instrumento :: String → Actividad
 tocar Instrumento instrumentazo musico =
 ↗ Delegar
 | elem instrumentazo específicos = map GradoXP (+1). map Instrumento (const instrumentazo) \$ musico
 | otherwise = map Instrumento (const instrumentazo) \$ musico
 específicos :: [String]
 específicos = ["dso", "fagot", "cello"]
 ↗ Hace falta un más expresivo
 map Instrumento :: (String → String) → Musico → Musico
 map Instrumento f musico = musico { instrumento = f. instrumento \$ musico }
 map GradoXP :: (Int → Int) → Musico → Musico
 map GradoXP f musico = musico { gradoXP = f. gradoXP \$ musico }

tipo Actividad = Musico → Musico

Cantar :: Actividad
Cantar = map Instrumento (: "Lalala")

HacerPresentación :: Activación → Actividad

HacerPresentación activación musical =

Si Toca, map Grado XP (+1), map Actuaciones (f [.. actuación])

Si Toca :: TMusico → Musico

Si Toca musical

I flip elem específicos, instrumento & musical = map Grado XP (+1)
Otherwise = id TMusico

map Actuaciones :: ([Activación] → [Activación]) → TMusico → Musico

map Actuaciones f musical = musical { actuaciones = f. actuaciones & musical }

Pensar :: Actividad

Pensar = id

ActividadEjemplo :: Actividad

ActividadEjemplo =

Pensar, HacerPresentación (Actividad (2,8,2025) 300), cantar, tocar Instrumento "fagot"

MusicoEjemplo :: Musico

MusicoEjemplo = Musico "charly" 5 "Piano" []

> actividad Ejemplo musicoEjemplo

Musico "charly" 8 "lalala fagot" [Actividad (2,8,2025) 300]

wuentaExperiencia :: Musico \rightarrow [Actividad] \rightarrow Int 3)
wuentaExperiencia musico actividades =
gradoXP. foldL (\musico actividad \rightarrow actividad musico) musico actividades

TienePresentacionesCorrectas :: Musico \rightarrow Bool 4)
TienePresentacionesCorrectas musico = evaluar . map cantidad . actuaciones \$ musico

evaluar :: [Int] \rightarrow Bool

evaluar [] = True

evaluar [pos1] = False \rightarrow no siempre. depende si la posición es impar o par
evaluar (pos1 : pos2 : tail) =

~~((not even \$ pos1) && (even \$ pos2)) && evaluar (pos2 : tail)~~ bien

9 (Nueve)

Haga 2/2
Maximo Tarado
Docente: Raúl González