

oPunto 1

Se tiene la siguiente porción de código Haskell que funciona correctamente

```
puedenPasar chicos = filter superaAlturaMaximaParaEntrarAlPelotero chicos
superaAlturaMaximaParaEntrarAlPelotero unChico = altura unChico > 1.1
```

(Se asume que la función `altura` ya está hecha correctamente y se la puede utilizar)

- Hacer una solución alternativa utilizando una expresión lambda en vez de la función auxiliar descripta.
- Hacer otra solución alternativa sin dicha función auxiliar ni expresión lambda, con composición. ¿Qué otro concepto es necesario utilizar?
- De las tres soluciones ¿cuál te parece más expresiva? ¿Por qué?
- Hacer la solución en Smalltalk. ¿A cuál de las tres se asemeja más?
- Establecer al menos dos relaciones conceptuales entre ambos paradigmas, que se apliquen en la solución.
- Señalar los errores de la siguiente solución en Prolog, explicarlos y corregirlos:

```
puedenPasar(Chicos, ChicosQuePasan):-
    findall(Chico, altura(UnChico) > 1.1, ChicosQuePasan).
```

Punto 2

Hay una ampliación del requerimiento, por la cual las alturas máximas son diferentes según cada pelotero. Esto implica que se debe pasar como parámetro el pelotero en cuestión.

- En funcional, elegir una de las tres soluciones del punto 1 y modificarla para el nuevo requerimiento, asumiendo que existe una función que aplicada a un pelotero devuelve la altura máxima.
- En objetos, se hizo una solución equivalente con objetos que representan a los peloteros, con una variable de instancia para la altura máxima y su getter, pero se dieron cuenta que los peloteros no tienen simplemente valores diferentes para las alturas máximas, sino que se calculan de distinta manera.
 - ¿Qué impacto tiene este nuevo cambio sobre la solución descripta? Justificar conceptualmente y desarrollar una solución ejemplificando con al menos dos tipos de peloteros.
 - ¿Fue necesaria o conveniente la herencia para desarrollar lo pedido? Justificar.
- En lógico, al igual que en objetos, se quiere hacer una solución representando a los diferentes tipos de peloteros con distintos funtores. Codificar la solución, con los mismos ejemplos de objetos y explicar conceptualmente relacionando ambos paradigmas.

Punto 3

Hay una última modificación del requerimiento que plantea que el permiso para poder pasar no siempre se limita a comparar la altura del chico contra la correspondiente al pelotero, sino que para algunos chicos en particular se requiere verificar además que el propio chico cumpla cierta condición adicional.

Explicar los cambios necesarios para soportar este nuevo requerimiento y explicar los nuevos conceptos que aparecen en la solución. No es necesario implementar la condición adicional, sólo el envío de mensaje (sin el método).