



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 - Programación Avanzada
1° semestre 2016

Actividad 02

Herencia, Polimorfismo, ABC

Instrucciones

El Plan Deportivo de Ingeniería le pide ayuda para organizar la información de sus competidores (en adelante, “atletas”). Cada atleta posee un nombre determinado, un nivel de energía entre 0 y 100, una velocidad, un registro ordenado de sus marcas (tiempos de llegada en segundos) y pertenece a uno de los siguientes tipos: **Ciclista, Nadador o Corredor**. A su vez, los ciclistas se clasifican en: **Ciclista de montaña, y Ciclista de pistas**. Finalmente, existen **Triatletas** que son tanto Corredores como Nadadores y Ciclistas de pista a la vez.

La empresa agrupa a sus atletas en diversos **Equipos**. Cada Equipo consta de un **Director Técnico** y un conjunto de atletas.

Se pide que usted modele el problema utilizando clases abstractas y herencia cuando corresponda.

Requerimientos

- Todos los atletas pueden **descansar**, lo que sumará 1 a su nivel de energía (siempre que el resultado no supere 100), independientemente del tipo de atleta en cuestión.
- Se debe poder comparar atletas de una misma categoría con los operadores ($>$, $<$, $==$) de acuerdo a la **mejor marca** (el menor de todos los tiempos registrados) de cada uno. Es posible comparar a un Ciclista de pista con un Triatleta de acuerdo a la marca respectiva, pero nunca atletas de distintas categorías (Ciclista de montaña y Ciclista de pistas se consideran categorías **diferentes**). Usted debe controlar, por ejemplo, que si se intenta comparar a un nadador con un triatleta, pero el último no tiene ninguna marca registrada para la categoría nadador, se imprima error y se retorne None al comparar.
- Cada Equipo es de un tipo específico de atletas, y solo pueden haber atletas de ese tipo en el equipo. Un triatleta puede pertenecer a cualquier equipo cuyo tipo corresponda a una categoría en la que el este puede competir.
- Es posible sumar equipos con el operador $+$. Si los equipos son de distinta categoría, el resultado es None. En caso contrario, se debe retornar un **nuevo equipo** cuyos integrantes sean la unión de los conjuntos de atletas de los equipos originales.
- El director técnico puede **alentar** a su equipo. Esto sumará 3 unidades al nivel de energía de los atletas.
- Todos los atletas pueden **entrenar**. Si su nivel de energía no es cero, se restará 1 a la energía, y se incrementará la velocidad del atleta. Si no hay suficiente energía para entrenar, se debe imprimir un

mensaje en consola. La cantidad de unidades de velocidad que se suman al atleta dependen de su tipo, y es 1 para los ciclistas, 2 para los nadadores, y 3 para los corredores.

- Finalmente, todos los atletas pueden **competir**, pero la forma en que compiten es distinta para cada tipo de atleta. Cuando un atleta compite, su energía disminuye. Además, se imprime un mensaje en consola. Se calcula el tiempo que tardó en completar el circuito y se agrega a su registro de marcas. Para el cálculo, utilice `random.gauss(1000/velocidad, 1)`. El mensaje que se imprime y las unidades de energía que se restan al atleta se muestran en la siguiente tabla:

Tipo de atleta	Decremento de energía	Mensaje
Nadador	1	El nadador <i>nombre</i> está nadando
Corredor	1	El corredor <i>nombre</i> está corriendo
Ciclista de montaña	1	El ciclista <i>nombre</i> está pedaleando una MountainBike
Ciclista de pista	1	El ciclista <i>nombre</i> está yendo a su máxima velocidad por la pista
Triatleta	3	El triatleta <i>nombre</i> está compitiendo en un triatlón

Notas

- Existen algunos puntos que no están completamente especificados en el enunciado. Por ejemplo: al sumar equipos, ¿quién es el DT del equipo resultante? ¿El del primer equipo, el del segundo equipo, None, Karim, ...? O cuando entrenan los triatletas: ¿en cuánto se incrementa su velocidad? Usted tiene libertad para decidir en casos como estos, siempre que lo que escriba no contradiga el resto del enunciado.
- Cuando un triatleta compite, deben conseguirse tres marcas al azar: una para nadador, una para ciclista de pista, y una para corredor. Utilice `random.gauss(1000/velocidad, 1)` para conseguir las tres.

To - DO

- (2.00 pts) Correcta modelación de OOP con ABC
- (2.00 pts) Main correcto: Se espera que cree un Main de test cumpliendo los siguientes pasos:
 - Crear un Nadador
 - Crear un Triatlete
 - Para el Triatleta creado, agregar una marcas para Corredor y Ciclista.
 - Imprimir las marcas del Triatleta creado
 - Crear un Team de Nadadores
 - Agregar al Triatleta creado a este Team
 - Comparar las marcas de los Nadadores con las del Triatleta del mismo equipo e imprimir al de menor marca (debería tirar error ya que el Triatleta no tiene marca de Nadador).
 - Agregar marca de Nadador al Triatleta y repetir el paso anterior.
- (2.00 pts) Resto de las funcionalidades no evaluadas en el Main

Tips

- Usted debe preocuparse de que, de alguna forma, un Triatleta guarde en forma separada sus marcas como Corredor, Nadador y Ciclista de pista.
- Planifique bien el modelo en su cabeza antes de empezar a escribir código. Algunos de los ítems pedidos pueden volverse sustancialmente más sencillos.