

Introducción a la programación OO

Tamagotchi

Introducción

Modelar e implementar en java la clase "Mascota" a partir de la cual se puedan instanciar objetos que representan una mascota virtual en una aplicación que emula este tipo de juegos.

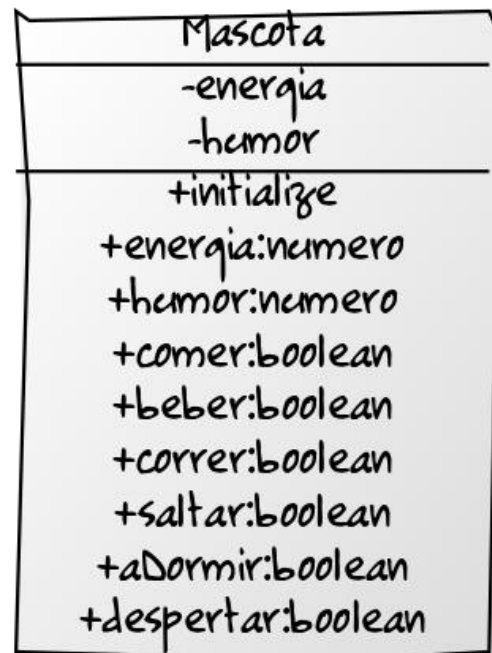
Esencialmente la mascota tiene energía un número entero entre 0 y 100 y humor un valor entero entre 1 y 5 que significa: muy enojado, enojado, neutral, contento y chocho respectivamente desde el 1 hasta el 5.

Dicha mascota debe responder a los siguientes mensajes implementados en comportamientos que están divididos en comportamientos de ingesta, comportamientos de actividad y otros.

Comportamientos de ingesta:

comer => incrementa la energía (que es un número entre 0 y 100 unidades) en 10% de la energía que tiene la mascota e incrementa el humor en 1 nivel.

beber => incrementa la energía (que es un número entre 0 y 100 unidades) en 5% de la energía que tiene la mascota e incrementa el humor en 1 nivel.



Comportamientos de actividades:

`correr` => decrementa la energía en un 35% de la energía que tiene la mascota. Y decrementa el humor en 2 niveles.

`saltar` => decrementa la energía en un 15% de la energía que tiene la mascota. Y decrementa el humor en 2 niveles.

Otros comportamientos:

`dormir` => la mascota pasa a estado durmiendo y no responde a ningún otro comportamiento excepto despertar. Además, la energía se incrementa en 25 unidades y el humor en 2 niveles.

`despertar` => la mascota pasa a estado despierta y comienza a responder a los demás comportamientos. Además, el humor se decrementa en un nivel.

Por otro lado, se deben respetar las siguientes reglas adicionales para todos los comportamientos en general:

- 1 - A partir de la 3 ingesta consecutiva el nivel de humor comienza a decrementar en 1 por cada ingesta.
- 2 - Cuando la energía llega a 0 la mascota muere de cansada.
- 3 - Si la mascota realiza 5 ingestas consecutivas muere de empacho.
- 4 - Si la mascota realiza 3 actividades consecutivas se empaca y se duerme.

Agregar además el comportamiento `toString` que devuelva una representación de cadena de la mascota incluyendo su nombre, unidades de energía, nivel de alegría, si duerme y si vive.

Notas:

1. la energía no puede superar 100 unidades, es decir si está en 100 no aumenta y si baja de cero la mascota muere y ya no responde más a ningún comportamiento.
2. el nivel de alegría fluctúa entre 1 muuuuy enojado y 5 chocho y si la alegría llega a 0 la mascota se va a dormir por propia iniciativa.
3. cuando la mascota está en un estado que no responde a cierto comportamiento el mismo debe retornar `false` y si el comportamiento se pudo realizar correctamente retorna `true`.

Ejercicio

- a) Escribir un programa en un main, que cree una mascota y luego envíe los mensajes a la mascota para cubrir un ciclo completo de la vida de esta mascota: nacimiento, actividad, muerte.

El programa debe mostrar el estado global de la mascota luego de cada envío de mensaje (ejecución de método).
- b) **Opcional, solo para alumnos avanzados:** Escribir un programa guiado por menús de opciones en consola, que permita jugar con una mascota como si fuera con un tamagotchi:
 - a) Todo aspecto no cubierto por el enunciado (por ejemplo, las proporciones de incremento y decremento de energía de las distintas acciones), deben ser resueltos por el programador de alguna manera coherente.