

I.E.S. Pablo Picasso

Programación 1DAM - Curso 2023/2024

Actividad 7.04 – Prueba Práctica Bloque 7

Vamos a hacer un juego de combates. En un juego de combate, dos grupos de personajes compiten entre si hasta que uno elimina a todos los participantes del otro. En cada turno se elige un personaje y una acción (atacar o curar) y a qué personaje se dirige la acción (puede ser del mismo o del otro equipo). Nuestro juego es bastante simple y tiene las siguientes reglas:

- Todos los personajes tienen un nombre y una vida inicial. La vida es un valor numérico entero positivo que indica la vitalidad del personaje y debe ser mayor de 0. Un personaje puede ir perdiendo y ganando vida a lo largo del juego pero ésta nunca puede ser mayor que la vida que tenía inicialmente, ni menor que cero. Cuando la vida llega a cero el personaje está muerto y ya no puede participar más en el juego ni puede curarse. El nombre es el nombre por el que se conoce al personaje y no debe estar en blanco para que se pueda identificar facilmente.
- Todo personaje puede ser atacado con una letalidad determinada. La letalidad es un valor numérico entero positivo o cero que indica la potencia del ataque. Cada personaje, según sus capacidades reacciona de una manera u otra a un ataque. Es ilegal intentar usar una letalidad negativa en un ataque.
- Todo personaje, excepto un personaje muerto, puede ser intentado curar con un nivel de poder de curacion determinado. El poder de curación es un número entero positivo o cero que indica lo fuerte que es la curación. Cada personaje, según sus capacidades, reacciona de una manera u otra a una curación. **Ningún personaje puede llegar a tener más vida después de ser curado que la que tenía inicialmente cuando se creó.** Es ilegal intentar curar con poder negativo.
- Los tipos de personaje son:
 - **Guerrero:** Puede llevar una armadura y puede manejar una de dos armas: espada y maza.
 - Si es atacado y tiene armadura, pierde la armadura y tanta vida como la **mitad** de la letalidad. Si no tiene armadura pierde la totalidad de la letalidad.
 - Si ataca a otro personaje y maneja una espada lo hace con tanta letalidad como la mitad de la vida que tenga mientras que si maneja una maza lo hace con tanta letalidad como la vida que tenga.
 - Si es curado recupera el doble de vida que el poder de curación que recibe.
 - No puede curar a nadie.
 - Si se le proporciona una armadura y no tiene se la pone. Si ya tiene la descarta. No se pueden acumular armaduras.
 - Puede cambiar de arma en cualquier momento.
 - **Arquero:** Puede llevar también armadura y su única arma es un arco que puede manejar con un nivel de maestría que puede ser: novato, medio y experto.
 - Si es atacado sigue las mismas reglas que un guerrero
 - Si ataca a otro personaje la letalidad con que lo hace es la mitad de la vida si es experto, un tercio si es medio y un quinto si es novato.
 - Si es curado recupera la misma vida que el poder de curación que recibe.

- No puede curar a nadie
- Si se le proporciona una armadura y no tiene ya una se la pone. Si ya tiene una se descarta. No se pueden acumular armaduras.
- Puede subir (no bajar) de nivel en cualquier momento.
- **Sanador:** No interviene directamente en la lucha por lo que no tiene armas ni armaduras pero tienes amplios poderes de curación.
 - Si es atacado pierde tanta vida como letalidad reciba en el ataque.
 - No puede atacar a nadie (su poder de ataque es cero)
 - Si es curado recibe tanta vida como el poder de curación que recibe y un tercio mas.
 - Si cura lo hace con tanto poder como la mitad de la vida que tiene
- **Sacerdote:** No maneja armas ni armaduras, al igual que el sanador pero puede disponer de una reliquia sagrada que le ayuda en su labor.
 - Si es atacado y tiene una reliquia no pierde vida pero pierde la reliquia. Si no tiene reliquia pierde tanta vida como letalidad reciba en el ataque
 - No puede atacar a nadie
 - Si es curado y tiene la reliquia recibe tanta vida como el poder de curación que recibe y un 50% más. Si no tiene la reliquia recibe tanta vida como el poder de curación que recibe.
 - Si cura y tiene la reliquia lo hace con tanto poder como vida tiene. Si no tiene la reliquia lo hace con tanto poder como la mitad de la vida que tiene.
 - Puede obtener una reliquia en cualquier momento pero no puede tener más de una reliquia en un momento dado.
- Todos los personajes deben soportar las operaciones:
 - Ser creados con un nombre y la vida inicial.
 - Recibir un ataque de una letalidad determinada (no importa de quién lo reciba, esto es algo que se implementa en el sistema de juego por turnos que está fuera del ámbito de este ejercicio), según las características particulares de cada tipo de personaje, comentadas arriba
 - Recibir una curación con un poder determinado (al igual que el anterior, no importa de quién lo reciba)
 - Consultar su poder de curación actual
 - Consultar su poder de ataque actual
 - Consultar si está vivo o muerto

Se pide que crees la jerarquía de clases que modele estos personajes empleando herencia de forma que:

- Se reutilice al máximo el código y se eviten duplicaciones.
- Se restrinja la creación de instancias en clases que no deben tenerlas.
- Exista una clase que represente a cada uno de los personajes y que el comportamiento de las mismas sea exactamente el especificado
- Se te proporciona libertad para definir los métodos y constructores que consideres necesarios, siempre que el comportamiento sea el especificado.

Se proporciona un documento (**pruebas . pdf**) con distintas situaciones de combate que se pueden producir junto con los estados finales de los personajes. Crea las pruebas JUnit correspondientes para probar estas situaciones con tu librería y ver que se ejecutan correctamente.

Se proporciona:

- Archivo **pruebas . pdf** con situaciones de combate que deben poder representarse con nuestra librería, así como el resultado esperado de su evaluación (redondeos aparte)

Se deberá entregar:

- La carpeta **COMPLETA** del proyecto comprimida en el formato ZIP, incluyendo las pruebas JUnit. Si éstas estuvieran aparte, carpeta comprimida con las pruebas.

Criterios de evaluación y calificación:

Se proporcionará una calificación por cada uno de los siguientes criterios de evaluación:

- Criterio 7.b) Se han utilizado modificadores para bloquear y forzar la herencia de clases y métodos. El uso de **abstract** para forzar la herencia y crear clases **NO** instanciables.
- Criterio 7.c) Se ha reconocido la incidencia de los constructores en la herencia. Se valorará la creación de los constructores de las clases.
- Criterio 7.d) Se han creado clases heredadas que sobrescriban la implementación de métodos de la superclase. Se valorará la sobrescritura de métodos para redefinir el comportamiento de la superclase.
- Criterio 7.e) Se han diseñado y aplicado jerarquías de clases. El diseño de la jerarquía de clases que sea lo más cercana posible a la naturaleza del problema a modelar.
- Criterio 7.f) Se han probado y depurado las jerarquías de clases. Se valorarán las pruebas y su corrección.
- Criterio 7.g) Se han realizado programas que implementen y utilicen jerarquías de clases. se valorará tanto la corrección de las clases como la de las pruebas.
- Criterio 7.h) Se ha comentado y documentado el código. Se valorará el comentado del código, incluyendo clases, atributos, constantes, métodos, parámetros, tanto de los métodos privados como de los públicos o protegidos. Se deberá utilizar obligatoriamente el formato Javadoc para todo lo que sea público o protegido.
- Criterio 3.h) Se han seguido unas reglas de estilo de código. Se valorará el seguimiento más o menos estricto de las reglas de estilo de código

En caso de que la aplicación no compile o termine de forma prematura debido a errores de programación la puntuación de **TODOS** los criterios anteriores será de 0 puntos.

Si no se colocan las clases en el paquete especificado la puntuación para el criterio 7.e será de 0 puntos, independientemente de lo correctas o incorrectas que sean dichas clases.

Si no se proporcionan las pruebas la puntuación de los criterios 7.f y 7.g será de 0 puntos

Cada caso de prueba en JUnit debe indicar claramente el número del caso de prueba que está probando. En caso de que esto no se haga la puntuación del criterio 7.f será de 0 puntos, independientemente de la corrección y pertinencia de las pruebas.