

Desafío - Tipos de métodos

En este desafío validaremos nuestros conocimientos de codificación de clases con métodos constructores, accesadores, mutadores y sobrecarga.

Lee todo el documento antes de comenzar el desarrollo **individual o grupal**, para asegurarte de tener el máximo de puntaje y enfocar bien los esfuerzos.

Descripción

Has sido contratado por la startup “**Juegos por comida**”, y se te ha solicitado desarrollar el algoritmo de la primera escena de su próximo juego “**Gran Fantasía**”.

El prototipo debe desarrollarse utilizando una aplicación de consola escrita en Python, que conste de un script principal que ejecute el juego, y una clase que permita crear personajes, que debe ser importada en el script principal. Las opciones de juego del usuario, así como el nombre de su personaje, se deben solicitar mediante `input`.

La clase que permite crear personajes debe considerar lo siguiente:

- Cada personaje tiene un nombre, el cual debe ser especificado al momento de crear un personaje.
- Cada personaje recién creado tiene nivel 1 y experiencia 0 (estos son los únicos valores posibles al momento de crear un personaje).
- A cada personaje es posible consultarle o asignarle un estado. Al solicitar el estado de un personaje, se debe mostrar en pantalla su nombre, su nivel y su experiencia. Al asignar un valor al estado, este valor corresponderá a la experiencia recibida por el personaje. Según la experiencia recibida, se debe modificar la experiencia actual del personaje y su nivel de acuerdo a lo siguiente:
 - Los valores posibles de experiencia son entre 0 y 99 inclusive.
 - El nivel mínimo es 1 y no hay máximo.
 - Cada 100 puntos de experiencia recibidos, el personaje sube 1 nivel (su nivel aumenta en 1). Ejemplo: El personaje tiene actualmente nivel 1 y experiencia 0. Si se asigna 250 a su estado, pasará a tener nivel 3 y experiencia 50.
 - Si la experiencia recibida es negativa, se debe restar a la experiencia actual del personaje. Cada vez que la experiencia sea menor a 0, el personaje baja de nivel (su nivel disminuye en 1). Ejemplo: El personaje tiene actualmente nivel 3 y experiencia 50. Si se asigna -150 a su estado, pasará a tener nivel 2 y experiencia 0. Si el personaje ya tiene nivel 1 y experiencia 0, no se altera su nivel ni su experiencia al recibir experiencia negativa.



Tip: Puede usar una variable temporal que corresponda a la suma entre la experiencia actual del personaje y la asignada al estado. Luego, mediante ciclos `while` puede ir actualizando este valor y el nivel del personaje y, terminados los ciclos, se asigna a la experiencia actual el valor de suma temporal actualizado. Otra posible solución es utilizando división redondeada (`//`) y módulo (`%`).

- Un personaje se considera mayor a otro si tiene mayor nivel, y se considera menor a otro si tiene menor nivel.



Tip: Investiga sobre los métodos especiales `__gt__` y `__lt__`. Puedes revisar la documentación oficial en el siguiente [link](#).

El script principal que ejecuta la escena del juego debe considerar el siguiente flujo:

- Dar la bienvenida al jugador y solicitar el nombre para su personaje
- Crear el personaje del jugador y mostrar su estado en pantalla
- Crear un personaje "Orco" y calcular la probabilidad de ganar que tiene el personaje del jugador versus el orco. Para esta probabilidad, debe considerar lo siguiente:
 - Si el jugador es menor al orco, tiene un 33% de probabilidades de ganar.
 - Si el jugador es mayor al orco, tiene un 66% de probabilidades de ganar.
 - Si el jugador es igual al orco, tiene un 50% de probabilidades de ganar.



Tip: Puede agregar en la clase un método de instancia que retorne la probabilidad de la instancia actual de ganar versus otra instancia.

- Informar por pantalla al jugador que ha aparecido un orco y la probabilidad que tiene de ganarle. Informarle también que en caso de ganarle, recibirá 50 puntos de experiencia y el orco perderá 30 y que, en caso de perder, perderá 30 puntos de experiencia mientras que el orco ganará 50. Consultar finalmente al jugador si desea atacar o huir. El jugador debe ingresar 1 para "Atacar" y 2 para "Huir".



Tip: Puedes agregar en la clase un método estático que reciba por parámetro la probabilidad, muestre en pantalla todo lo requerido y retorne la opción de juego del usuario.

- Mientras la opción de juego del usuario sea “Atacar” (1), se debe realizar lo siguiente:
 - Obtener el resultado del ataque del jugador al orco, donde los resultados posibles son “Gana” o “Pierde”. Para ello, genere un número al azar entre 0 y 1. Si el número obtenido es menor o igual que la probabilidad calculada, entonces el resultado del ataque es “Gana”. En caso contrario, el resultado es “Pierde”. **Tip:** Puede usar `uniform()` del módulo `random`.
 - Informar al jugador el resultado del ataque, los puntos de experiencia ganados o perdidos según corresponda, actualizar el estado del orco, actualizar el estado de su personaje y mostrar sus estados posteriores a la actualización. Recordar que si el jugador gana, el jugador gana 50 puntos de experiencia y el orco pierde 30, y en caso contrario, el jugador pierde 30 puntos de experiencia y el orco gana 50.
 - Con el estado modificado, actualizar el valor de probabilidad de ganar al orco, y volver a consultar al jugador su opción de juego.
- Si el jugador ha decidido huir, mostrar mensaje en pantalla informando que el orco ha quedado atrás.

Requerimientos

1. En un archivo llamado `personaje.py`, crea la clase que permita crear personajes. Considerando la información entregada en la descripción del problema, la clase debe tener los siguientes métodos:
(6 Puntos)
 - a. Constructor (considera parámetros y valores asignados a atributos de instancia)
 - b. Getter de estado.
 - c. Setter de estado.
 - d. Sobrecarga para comparar “menor que” .
 - e. Sobrecarga para comparar “mayor que”.
 - f. Sobrecarga para comparar “igual que” (opcional).
 - g. Método de instancia que retorna la probabilidad de la instancia actual de ganar respecto de otra instancia (opcional).
 - h. Método que muestra diálogo de enfrentamiento al orco (incluyendo probabilidad de ganar) y retorna opción escogida por el jugador (opcional).

2. En archivo llamado `juego.py`, crea la clase que permita jugar, considerando la información entregada en la descripción del problema:

(4 Puntos)

- Importar módulos necesarios para el desarrollo del juego.
- Creación de personajes y almacenamiento de opción de juego del jugador (calculando y mostrando probabilidad de ganar)
- Para la opción de ataque, según el resultado obtenido, actualización de estados de los personajes.
- A continuación del punto anterior, dentro del ataque, mostrar estados actualizados de los personajes, y nueva consulta al usuario considerando la probabilidad actualizada de ganar.



¡Mucho éxito!

Consideraciones y recomendaciones

Ejemplo de salida

```
¡Bienvenido a Gran Fantasía!  
Por favor indique nombre de su personaje:  
Evil666  
NOMBRE: Evil666          NIVEL: 1          EXP: 0  
  
¡Oh no!, ¡Ha aparecido un Orco!  
  
Con tu nivel actual, tienes 50.0% de probabilidades de ganarle al Orco.  
  
Si ganas, ganarás 50 puntos de experiencia y el orco perderá 30.  
Si pierdes, perderás 30 puntos de experiencia y el orco ganará 50.  
  
¿Qué deseas hacer?  
1. Atacar  
2. Huir  
1  
  
¡Le has ganado al orco, felicidades!  
¡Recibirás 50 puntos de experiencia!  
  
NOMBRE: Evil666          NIVEL: 1          EXP: 50  
NOMBRE: Orco             NIVEL: 1          EXP: 0
```

Con tu nivel actual, tienes 50.0% de probabilidades de ganarle al Orco.

Si ganas, ganarás 50 puntos de experiencia y el orco perderá 30.
Si pierdes, perderás 30 puntos de experiencia y el orco ganará 50.

¿Qué deseas hacer?

1. Atacar

2. Huir

1

¡Oh no! ¡El orco te ha ganado!
¡Has perdido 30 puntos de experiencia!

NOMBRE: Evil666	NIVEL: 1	EXP: 20
NOMBRE: Orco	NIVEL: 1	EXP: 50

Con tu nivel actual, tienes 50.0% de probabilidades de ganarle al Orco.

Si ganas, ganarás 50 puntos de experiencia y el orco perderá 30.
Si pierdes, perderás 30 puntos de experiencia y el orco ganará 50.

¿Qué deseas hacer?

1. Atacar

2. Huir

1

¡Oh no! ¡El orco te ha ganado!
¡Has perdido 30 puntos de experiencia!

NOMBRE: Evil666	NIVEL: 1	EXP: 0
NOMBRE: Orco	NIVEL: 2	EXP: 0

Con tu nivel actual, tienes 33.0% de probabilidades de ganarle al Orco.

Si ganas, ganarás 50 puntos de experiencia y el orco perderá 30.
Si pierdes, perderás 30 puntos de experiencia y el orco ganará 50.

¿Qué deseas hacer?

1. Atacar

2. Huir

1

¡Oh no! ¡El orco te ha ganado!
¡Has perdido 30 puntos de experiencia!

NOMBRE: Evil666	NIVEL: 1	EXP: 0
NOMBRE: Orco	NIVEL: 2	EXP: 50

Con tu nivel actual, tienes 33.0% de probabilidades de ganarle al Orco.

Si ganas, ganarás 50 puntos de experiencia y el orco perderá 30.
Si pierdes, perderás 30 puntos de experiencia y el orco ganará 50.

¿Qué deseas hacer?

1. Atacar
 2. Huir
- 2

¡Has huido! El orco ha quedado atrás.