

# Enunciado Práctica Juego

Se deben implementar 3 clases que estructuren nuestro ejemplo:

1. Clase Personaje, va a ser el padre de nuestras clases y tendrá como atributo:

int vida;

la vida se inicializará a 100 en el constructor de Personaje.

Y como métodos:

Método getVida() que nos devuelva el valor de vida del personaje.

Método setVida() que establece el nuevo valor de vida del personaje.

2. Clase Guerrero, que hereda de la clase Personaje y tendrá como atributos:

int ataque

Y como métodos:

Constructor de Guerrero que inicialice nuestro atributo:

- ataque=10

Método movimientoLucha() que nos devuelva el valor de ataque y muestre por consola la sentencia "\*\*\*\*\*Espadazo\*\*\*\*\*".

Extra: añadir aleatoriedad a nuestra lucha con la librería math random

int numero = (int) (Math.random() \* ataque);

3. Clase Mago, que hereda de la clase Personaje y tendrá como atributos:

int magia

Y como métodos:

Constructor de Mago que inicialice nuestro atributo:

- magia=10

Método movimientoLucha() que nos devuelva el valor de la magia y muestre por consola la sentencia "\*\*\*\*\*Hechizo\*\*\*\*\*".

Extra: añadir aleatoriedad a nuestra lucha con la librería math random,

```
int numero = (int) (Math.random() * magia);
```

4. Construir una clase llamada “Arena” que contendrá el método Main, que va a ejecutar la siguiente secuencia:

- Crear un Guerrero.
- Mostrar su nivel de vida.
- Ejecutar un movimiento de lucha.

- Crear un Mago
- Mostrar su nivel de vida.
- Ejecutar un movimiento de lucha.

5. Añadir a nuestra clase “Arena” un método, llamado “Lucha” donde:

- Le pasaremos por parámetro el objeto Mago y el objeto Guerrero.
- Dentro del método el Guerrero ejecutará movimientoLucha y el valor que nos devuelva la función, se le restará a la vida del Mago.
- Dentro del método el Mago ejecutará movimientoLucha y el valor que nos devuelva la función, se le restará a la vida del Guerrero.

Desde nuestro Main, llamaremos a la función Lucha, y después mostraremos por consola la vida que le resta a nuestro Guerrero y nuestro Mago.

6. En nuestra clase arena. Se deben crear dos variables que serán inicializadas a 0 al comienzo del Main:

- danoValor: será un entero de tipo primitivo.
- danoReferencia: será un wrapper de tipo entero.

Ambas se pasarán como parámetro de entrada al método Lucha, y ambas almacenarán el daño acumulado en el combate (Lucha) sumando los puntos de vida restados al mago y al guerrero.

- Si al guerrero le restan 7 y al mago le restan 9 de vida, el daño acumulado será 16.

A nuestro método Lucha se le deben pasar por parámetro estas variables y cuando termine el combate (método lucha), en el main comprobaremos qué valor tienen guardadas ambas variables.

**¿Qué ocurre? ¿Qué problemas nos ocasiona?**

7. Añadir a todos constructores de nuestras clases:

- Que saque por pantalla el mensaje relativo a la clase que se está construyendo.
- Que escriba la fecha y hora actual de creación.

\*\*\*\* Construyendo un Guerrero a las Mon Dec 12 19:46:02 \*\*\*\*

8. Nuestra clase personaje pasa a ser abstracta y crearemos un método abstracto llamado:

- movimientoLucha()

Éste obliga a implementar el método movimientoLucha en cualquier subclase que extienda al personaje.

9. Convertir nuestros objetos Guerrero y Mago en final.

10. Crear una clase que se llame GuerreroFuerte que herede que la clase Guerrero. Ésta debe sobrescribir el método movimientoLucha sumando 5 al resultado de math.Random.

11. Se debe sobrecargar el método lucha de nuestra clase Arena. Aceptando que luchen:

- Mago vs Guerrero.
- Mago vs GuerreroFuerte.

Opcional: Crear un pequeño menú que nos permita elegir quien queremos que luche, y poder hacer varios combates hasta que uno de ellos agote su vida.

12. En nuestra clase Arena vamos a crear un nuevo Guerrero a partir de nuestro GuerreroFuerte, y éste debe luchar contra el mago.

- Observamos que habiendo hecho casting, **nuestro nuevo Guerrero es el Guerrero Fuerte encubierto** (ahora el Guerrero Fuerte tiene dos nombres, está asignado al nuevo Guerrero, y ambos son el mismo luchador).

13. Modificar el constructor del Mago para que su vida sea un 10% superior a la de cualquier personaje, utilizando la función super().

- Pista: tendremos que hacer uso de las funciones de Personaje getVida y setVida.

# Ejercicio juego completo

## Sistema de Juego.

Ampliar el programa de juego (Arena, Personaje, Guerrero, GuerreroFuerte, Mago) con las siguientes mejoras:

- 1) Al construir el guerrero, se muestre por pantalla la hora, minuto y segundos de creación. Esta hora de creación se guarda en un atributo tipo Date.
- 2) Al construir el mago, se muestre por pantalla tu fecha de nacimiento (la del alumno) en formato día, mes y año. Esta fecha se guarda en la clase en un atributo tipo Date.
- 3) Modificar el constructor del Mago para que su vida sea un 10% superior a la de cualquier personaje, utilizando la función `super()`. Pista: tendremos que hacer uso de las funciones de Personaje `getVida` y `setVida`.

Entregar en un zip todas las clases.