

EL SEÑOR DE LOS ANILLOS

Realiza un programa en java que simule una batalla entre ambos bandos de la saga de *El Señor de los Anillos*.

Al comienzo del programa, el jugador deberá poder escoger el bando con el que desea jugar (luz u oscuridad). Realiza un pequeño menú inicial para que el jugador pueda escoger en cada partida.

El programa contendrá 4 personajes de cada bando de los cuales podremos distinguir entre dos tipos de personajes: **guerrero** y **magos**.

Cuando el usuario seleccione un bando tendrá a su disposición 4 personajes con las siguientes características:

LUZ: Gandalf, Aragorn, Legolas y Gimli

OSCURIDAD: Saruman, Balrog, Lurtz y Orco

Ambos tipos de personajes tienen:

- Nivel de ataque (entero)
- Nivel de defensa (entero)
- Nivel de vida (entero)
- Nombre (String)

El guerrero tiene una función especial, llamada ataque crítico. Lo que hace este ataque es darle un 15% de posibilidades de que ocurra y que su efecto sea que duplique los puntos de ataque que tiene el guerrero como atributo.

El mago tiene una función especial de curar sus heridas y lanzar hechizos.

El mago en cada turno puede elevar su nivel de vida. Podrá recuperar 10 puntos de vida en cada turno. No siempre será factible. Para ello, podemos hacer que esto tenga un 15% de posibilidades de que ocurra.

Así que podemos escribirle a Mago un método privado que se encargue de decidir si recupera puntos de vida o no.

Lanzar hechizos funciona exactamente igual que el ataque crítico. Es decir, si desea atacar tendrá un 15% de probabilidad de lanzar un hechizo, lo que hará que los puntos totales de ataque se dupliquen.

Crea las clases correspondientes, utilizando **herencia**. No se permitirá crear otro tipo de *Personaje* que no sean los indicados.

Cuando se crea una instancia de la clase de guerrero o mago, tienes que pasar sus parámetros de ataque, defensa y nivel de vida para crearlo, los parámetros serán los siguientes:

Gandalf (mago):

- Ataque: 25
- Defensa 10
- Vida: 100

Aragorn (guerrero):

- Ataque: 20
- Defensa 20
- Vida: 100

Legolas (guerrero):

- Ataque: 15
- Defensa 25
- Vida: 100

Gimli (guerrero):

- Ataque: 10
- Defensa 20
- Vida: 100

Saruman (mago):

- Ataque: 25
- Defensa 10
- Vida: 100

Balrog (mago):

- Ataque: 35
- Defensa 10
- Vida: 100

Lurtz (guerrero):

- Ataque: 25
- Defensa 15
- Vida: 100

Orco (guerrero):

- Ataque: 5
- Defensa 5
- Vida: 100

Utiliza un array de objetos para que el usuario pueda escoger el personaje a enfrentar junto con el seleccionado al azar del oponente.

El juego pregunta si quieren defenderse o atacar. Si atacan, invocan el poder con un número aleatorio hasta el atributo ataque que tengan y con este ataque bajan puntos de vida a su oponente.

Por ejemplo, si se recibe un ataque de 25, pero en nuestro turno hemos elegido defender y en el atributo defensa del personaje tenemos, por ejemplo, un 10, quedaría que los daños recibidos se reducen a 15 ($25 - 10 = 15$)

Incluso si la defensa es mayor o igual al ataque, este habría quedado bloqueado por completo y no se recibiría daño alguno.

Decidiremos qué puede ocurrir en cada turno, tanto si elige atacar como si elige defender.

A continuación, al azar su oponente pide atacar o defender, si ataca, el ataque hará un número aleatorio entre 0 y el atributo de ataque que contenga el personaje y se repetirá hasta que uno de los personajes alcance su nivel de vida a cero.

```
public int atacar() {
    Random azar = new Random();
    //Ataque mínimo: 1, Máximo: según nivel de ataque
    int puntosAtaque = azar.nextInt(ataque) + 1;

    return puntosAtaque;
}

public int defender() {
    Random azar = new Random();
    //Defensa mínima: 1, Máxima: según nivel de defensa
    int puntosDefensa = azar.nextInt(defensa) + 1;

    return puntosDefensa;
}
```

Crea los constructores que creas conveniente para crear los objetos.

Al final de la batalla deberás indicar quién es el vencedor (si luz u oscuridad).

Utiliza comentarios pertinentes.

Emplea tratamiento de errores.

Formatea o indenta el código fuente.

Utiliza nombres coherentes de variables, constantes, métodos, atributos...