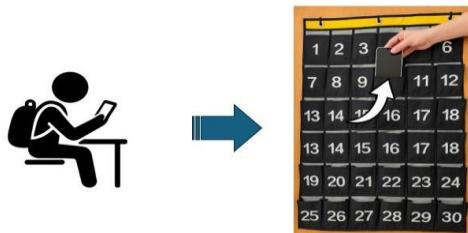




CICLO DAM2: Acceso a Datos

NORMAS

Si consultas el móvil --> sigue las instrucciones del pictograma 1.



Pictograma 1 1

Todo el código tiene que estar comentado.

PROHIBIDO usar bucles infinitos y de más de 1 return salvo las excepciones comentadas en clase: for-each.

Está prohibida la copia o plagio de código, así como el uso de herramientas no autorizadas.

Deben ser verificadas las entradas por teclado y por lo tanto asegurar la no ruptura del programa en caso de que los valores introducidos no sean los esperados.

El usuario debe estar informado con mensajes claros de la operación que se va a ejecutar, si se ha ejecutado satisfactoriamente o no, etc.

Deben ser capturadas las excepciones que puedan ocurrir en el programa.

FORMATO DE ENTREGA:

Debes actualizar el proyecto en GitHub y subir la versión actual completa sobre una nueva rama (todo lo anterior debería estar en la rama master) que se llamará enero26.

Dragolandia

Desarrolla una aplicación de aventura medieval que usando Hibernate y MySQL gestiona una base de datos denominada dragolandia. La aplicación debe construirse respetando el **patrón de diseño MVC (Modelo–Vista–Controlador)**.

A través del código se crearán las correspondientes tablas en la base de datos y también se persistirá la información manejada en la clase Principal.

Deberás crear las siguientes clases en Java:

1. Clase Mago

o Atributos:

- id (numérico): identificador del mago
- nombre (texto): Nombre del mago.
- vida (valor numérico entero): Puntos de vida del mago que nunca puede ser un valor negativo.
- nivelMagia (valor numérico entero): Nivel de poder mágico del mago que nunca puede ser un valor negativo.
- Conjuros: lista de hechizos que son conocidos para el mago

o Métodos:

- lanzarHechizo: Recibe como parámetro un Monstruo y le resta puntos de vida en función del nivel de magia.
- LanzarHechizo: Recibe como parámetro un monstruo y un hechizo y le resta al monstruo la vida que consuma el hechizo lanzado, siempre y cuando sea un hechizo de los conjuros de este mago. Si el hechizo no es de los conjuros, se le restará la vida al mago 1 punto.

2. Clase Monstruo

o Atributos:

- id (numérico): identificador del monstruo
- nombre (texto): Nombre del monstruo.

- vida (valor numérico entero): Puntos de vida del monstruo que nunca puede ser un valor negativo.
 - tipo (enumerado): Tipo de monstruo cuyos posibles valores son: ogro, troll, espectro.
 - fuerza (valor numérico entero): impacto que provoca el monstruo cuando ataca.
 - o Métodos:
 - atacar: Recibe como parámetro un Mago y consume de la vida de este la fuerza que tenga el monstruo en función del tipo de monstruo que sea.
- Puedes establecer los valores que deseas de vida para cada tipo de monstruo.

3. Clase Dragón o Atributos:

- Nombre (texto): Nombre del dragón
- intensidadFuego (valor numérico entero): valor numérico que implica la intensidad de la llama que lanza el dragón.
- resistencia (valor numérico entero): vida del dragón
- o Métodos:
 - exhalar: Recibe como parámetro un monstruo y consume de la vida de este el valor que tenga asignado como intensidad de fuego.
- o Relaciones:
 - Un Dragón habita en un Bosque.

4. Clase Bosque

o Atributos:

- id (numérico): identificador del bosque
- nombre (texto): Nombre del bosque.
- nivelPeligro (valor numérico entero): Nivel de peligrosidad del bosque.
- monstruoJefe: Objeto de tipo Monstruo que representa al jefe del bosque.
- Lista de monstruos asignados a este bosque

o Métodos:

- mostrarJefe: Muestra los datos del monstruo jefe.

- cambiarJefe: Permite asignar un nuevo monstruo jefe que recibe como parámetro, al bosque.
- addMonstruo: asigna un monstruo al bosque. O Relaciones:
- Un bosque tiene 0 o varios monstruos

5. Hechizo:

- o Bola de fuego:

- Efecto: cantidad de vida a reducir de todos los elementos afectados (0 o varios) o Rayo:
- Efecto: cantidad de vida a reducir del elemento afectado (1).

- o Bola de nieve:

- Efecto: el personaje oponente se queda congelado y por lo tanto se queda sin vida.

- o Añade uno o más hechizos.

6. Crea las clases que necesites a mayores de las anteriormente indicadas.

Roles de los integrantes del juego:

- En nuestro juego el Mago tendrá el rol de combatir a los monstruos, proteger el bosque y colaborar con el dragón.
- El Monstruo amenazará a los magos dentro del bosque.
- El Monstruo jefe, el cual es un monstruo especial del bosque en el que estamos jugando, será el obstáculo principal. Será el líder de los monstruos y objetivo de los magos y del dragón.
- El bosque es el escenario principal del juego y contiene monstruos y el monstruo jefe.
- El Dragón es el protector del bosque y combate a los monstruos.

Recuerda, que como comentamos, deben existir las operaciones CRUD (creación, actualización, eliminación, búsqueda/consulta) para cada entidad definida en nuestro proyecto. Como sabemos, estas operaciones deben estar encapsuladas en una clase por entidad.

Dinámica del juego

Inicio del juego:

- Crear un bosque.
- Crear un mínimo de 3 monstruos
- Asignar un monstruo jefe al bosque creado.
- Crear un dragón
- Crear un mínimo de 2 magos y asignarle un mínimo de 2 conjuros a cada uno.

Mientras los componentes del juego tengan vida:

- a) Cada Mago lanza unconjuro de todos los que hay disponibles en el juego. Si el conjuro es conocido por el Mago, se actualizarán los datos de los monstruos del bosque. En caso de que el conjuro no sea conocido por el mago, la vida del mago será reducida 1 punto.
 - b) Cada monstruo ataca a un mago, aplicando los cambios que corresponda en sus datos.
 - c) El Dragón ataca al monstruo jefe
-
1. Si la vida de un mago, monstruo o dragón llega a 0, será eliminado del juego. En caso de que el monstruo jefe muera, será asignado uno de los monstruos vivos que tengamos en el bosque como monstruo jefe.
 2. Después de cada ronda de juego, será mostrado el resultado del estado de todos los personajes (vivos y muertos, junto con la vida o resistencia según el tipo). También se indicará el nombre del monstruo jefe del bosque.
 3. La partida se terminará cuando ya no haya monstruos en el bosque o no tengamos magos.

Ampliación

Propón las mejoras o cambios que consideres sobre el juego planteado. No es necesario que los implementes, descríbelos y si es necesario, adjunta el nuevo diagrama de clases.