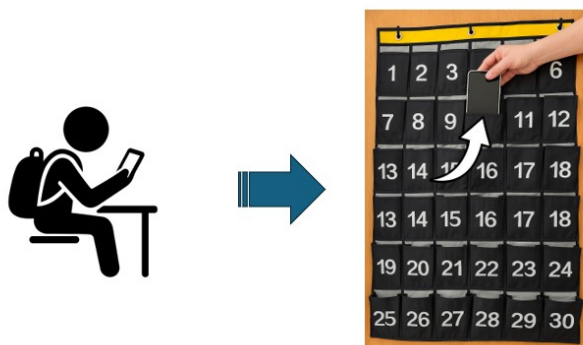


NORMAS

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 entregar un fichero .zip que incluya los ficheros indicados al final de este documento con el nombre:

AD-UD3-AT.02-Dragolandia-Apellidos_Nombre.zip

**Donde Apellidos y Nombre serán tus apellidos y tu nombre.

1. 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.

- o Métodos:

- ♣ constructor/es
- ♣ getter/setter
- ♣ lanzarHechizo: Recibe como parámetro un Monstruo y le resta puntos de vida en función del nivel de magia.

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.
- Métodos:
 - ♣ constructor/es
 - ♣ getter/setter
 - ♣ atacar: Recibe como parámetro un Mago y consume de la vida de este la fuerza que tenga el monstruo.

3. Clase Bosque

- 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.
- Métodos:
 - ♣ constructor/es
 - ♣ getter/setter
 - ♣ mostrarJefe: Muestra los datos del monstruo jefe.
 - ♣ cambiarJefe: Permite asignar un nuevo monstruo jefe que recibe como parámetro, al bosque.



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

5. **Clase principal**

- o Crea un mago y un monstruo leyendo sus valores por teclado.
- o Crea un bosque y asigna un monstruo jefe.
- o Simula una batalla por turnos entre el mago y el monstruo jefe del bosque:
 - ♣ El mago lanza un hechizo al monstruo.
 - ♣ El monstruo ataca al mago.
 - ♣ La batalla termina cuando uno de los dos pierde toda su vida.
- o Al final, muestra quién sobrevive y quién domina el bosque.