

UNIDAD 8:

HERENCIA

CLASE 4: EJERCICIOS



I Y 2: EJERCICIOS INTERFACES

Ejercicio 1: **libros**

Define la interfaz **Libro** que tendrá los métodos:

- **getNumPaginas()** devolverá un entero
- **getInformacion()** recibirá y devolverá nada
- **getEstatusLeido()** recibirá y devolverá nada

Define la clase **LibroFisico** que cuenta con:

- Los atributos **nombre**, **numPaginas**, **frasesDestacadas** (**array de strings**), **marcapaginas**, **dimensiones** (que será **array de 3**), **peso**.

Por otro lado, tendrá los métodos:

- **Constructor** con parámetros
- **getNumPaginas()** devolverá la cantidad de páginas
- **getInformacion()** mostrará por pantalla toda la información del libro.
- **getEstatusLeido()** mostrará por pantalla: “Actualmente leyendo la página **marcapaginas** de **numPaginas**”
- **setMarcapaginas()** cambiará el valor del marcapaginas si es posterior a su valor actual
- **destacarFrase()** recibirá la página y la frase a destacar y los guardará en el array con el formato “pagina:frase”

eliminarFrases() recibe una página del libro y recorre el array eliminando todas las frases que encuentre en esa página

- **calcularPrecio()** devuelve el double resultante de multiplicar las tres dimensiones y sumar el peso del libro

Define la clase **Ebook** que cuenta con:

- Los atributos **nombre**, **numPaginas**, **tamaño en MB**, **formato** (PDF, EPUB, MOBI, AZW) **y porcentajeLeido**.

Por otro lado, tendrá los métodos:

- **Constructor** con parámetros
- **getNumPaginas()** devolverá la cantidad de páginas

- **getInformacion()** mostrará por pantalla toda la información del libro.
- **getEstatusLeido()** mostrará por pantalla: “Actualmente has leído un **porcentajeLeido%** del total del libro”
- **setPorcentaje()** actualizará el porcentajeLeido teniendo en cuenta la página pasada por parámetro y el número de páginas del ebook.
- **transformar()** modificará el formato al pasado por parámetro teniendo en cuenta que el tamaño varía según formato:
EPUB +20% / MOBI -50% / AZW +100% con respecto al tamaño que tendría en PDF

Una vez las clases estén creadas, en **main** crea un **array de 6 Libros: 3 LibroFisico y 3 Ebook**. Después crea un **menú** que permita escoger un Libro o un Ebook y modificar o revisar sus valores (recomiendo crear 2 menús anidados, uno externo que pregunte si se quieren modificar ebooks o libros y, según se elija, mostrar un segundo menú con las opciones disponibles para ese formato) También recomendaría crear el **método toString para ambas clases** que en vez de mostrar toda la información, sólo muestre lo fundamental (titulo, numPaginas, formato (en caso de ebook) o dimensiones (en caso de fisico)).

GetInformacion para LibroFisico:

Nombrelibro de **numPaginas** de
dimensiones **d1xd2xd3** con peso **peso**:
Usted ha leído hasta la página
marcapaginas listando las frases
destacadas:

- Página **x**: **frase**

GetInformacion para Ebook:

Nombrelibro de **numPaginas** de
formato **formato** y tamaño **tamanyo**:
Usted ha leído un porcentaje de
porcentajeLeido del total.

Ejercicio 2: **productos**

Define la interfaz **Producto** que tendrá los métodos:

- **getNombre()** devolverá un string
- **getPrecio()** devolverá un double
- **getDescripcion()** devolverá un string
- **aplicarDescuento()** recibirá un double que será un porcentaje
- **reponer()** recibirá un int
- **comprobarDisponibilidad()** devolverá un boolean.
- **mostrarInformacion()** devolverá y recibirá nada.

Define la clase **Ropa** que cuenta con:

- Los atributos **nombre**, **precio**, **descripcion**, **tipoMaterial**, **color** y **talla**. Color y talla serán arrays. **Disponibilidad** será un array bidimensional cuyas colores serán las filas y tallas las columnas.

Por otro lado, tendrá los métodos:

- **Constructor** con parámetros
- **getNombre()**, **getPrecio()** y **getDescripcion()** devolverán las variables pedidas.
- **aplicarDescuento()** el porcentaje del descuento se restará teniendo en cuenta que el precio original es el 100.
- **reponer()** la cantidad de añadirá a todas las posiciones del array.

- **getNombre()**, **getPrecio()** y **getDescripcion()** devolverán las variables pedidas.
- **aplicarDescuento()** el porcentaje del descuento se restará teniendo en cuenta que el precio original es el 100.
- **reponer()** la cantidad de añadirá a todas las posiciones del array.
- **comprobarDisponibilidad()** comprobará si queda alguna prenda en alguna combinación de la tabla, devolverá el booleano y si es true mostrará por pantalla las combinaciones disponibles.
- **mostrarInformacion()** mostrará toda la información de la prenda.

- **comprar()** recibirá la talla y el color en dos strings, buscará los índices en el array y disminuirá en 1 la celda de la disponibilidad.
- **probar()** calculará un random 0 o 1 y devolverá un booleano correspondiente.

Define la clase **Juguete** que cuenta con:

- Los atributos **nombre**, **precio**, **descripción**, **edadRecomendada**, **cantidad** y **esInteractivo**.

Por otro lado, tendrá los métodos:

- **Constructor** con parámetros
- **getNombre()**, **getPrecio()** y **getDescripcion()** devolverán las

variables pedidas.

- **aplicarDescuento()** se le pasará un entero que se disminuirá al precio si es menor.
- **reponer()** el valor pasado por parámetro se añadirá a la cantidad.
- **comprobarDisponibilidad()** comprobará si cantidad es mayor a 0 y si es true mostrará por pantalla las combinaciones disponibles.
- **mostrarInformacion()** mostrará toda la información del juguete.
- **comprar()** disminuirá en uno la cantidad disponible.
- **esSeguroParaEdad()** recibirá una edad y la comparará con la edadRecomendada

y devolverá un booleano en función.

Define la clase **Medicamento** que cuenta con:

- Los atributos **nombre**, **precio**, **descripción**, **cantidad**, **fechaCad**, **requiereReceta** y un array de string **componentes**.

Por otro lado, tendrá los métodos:

- **Constructor** con parámetros
- **getNombre()**, **getPrecio()** y **getDescripcion()** devolverán las variables pedidas.
- **reponer()** el valor pasado por parámetro se añadirá a la cantidad.

variables pedidas.

- **aplicarDescuento()** se le pasará un char que corresponderá con un grupo: ‘a’ será para ancianos (80%), ‘b’ para desempleados (50%) y ‘c’ para jóvenes (30%). Este porcentaje se restará del precio.
- **comprobarDisponibilidad()** comprobará si cantidad es mayor a 0 y si es true mostrará por pantalla las combinaciones disponibles.
- **mostrarInformacion()** mostrará toda la información del medicamento.

- **contiene()** se le pasará un string y comprueba si el medicamento contiene el componente pasado.

- **comprobarReceta()** comprueba si se requiere una receta para adquirir ese medicamento.

Después en main crea un array de Productos y almacena varios de cada tipo. Crea un menú que permita comprar o revisar la información de todos los productos eligiendo la sección que se va a visitar y posteriormente las acciones posibles.

3 Y 4: EJERCICIOS

INTERFACES Y

HERENCIA

Ejercicio 3: **centro_de_recuperacion**

Crea la Interfaz **Animal** con métodos:

- **getNombre()** que devolverá un string
- **alimentar()** que recibirá y devolverá nada
- **revisarSalud()** que recibirá y devolverá nada

Crea la clase abstracta **AnimalBase** que implementa Animal:

- Tendrá los atributos generales: **nombre, edad, estadoSalud, dieta, nivelEnergia** que inicia a 50

Como método abstracto tendrá el método **mostrarInformacion()** y **necesitaDescanso()**

- Implementa **getNombre()** recuperándolo.
- **alimentar()** aumenta en 10 el **nivelEnergia**.
- **revisarSalud()** si el estado es **Herido, Grave o Riesgoso**, **estadoSalud** se modificará a **Recuperándose**, si es **Recuperándose** entonces cambiará a **Bien**.

Crea la clase **Ave** que hereda de **AnimalBase**:

- Tendrá los atributos **puedeVolar**(booleano) y **habitat**

Además los métodos:

- **volar()** si puede volar y la energía es mayor a 10 entonces resta 10.
- **migrar()** mostrará por pantalla “*El ave está migrando a otra zona en el habitat **habitat***”
- **mostrarInformacion()** muestra toda la información del ave.
- **necesitaDescanso()** si la energía es menor al 20 devuelve true.

Crea la clase **Mamífero** que hereda de **AnimalBase**:

- Tendrá el atributo **peso**

Además los métodos:

- **mostrarInformacion()** muestra toda la información del mamífero.
- **necesitaDescanso()** si la energía es menor al 10 devuelve true.
- **correr()** si la energía es superior a 10, resta el peso a la energía.

Crea la clase **Reptil** que hereda de **AnimalBase**:

- Tendrá los atributos **temperaturaCorporal**

Además los métodos:

- **tomarSol()** su temperaturaCorporal aumenta 1.5 grados.
- **estarALaSombra()** su temperaturaCorporal disminuye 2 grados.
- **mostrarInformacion()** muestra toda la información del reptil.
- **necesitaDescanso()** si la energía es menor al 15 y su temperatura es menor a 30 grados devuelve true.

Crea una clase main que contenga un array de 6 animales (2 de cada tipo) después:

- Recorre el array para mostrar su información
- Utiliza los métodos para modificar los valores internos de las clases
- Comprueba si necesitan descanso
- Recorre el array de nuevo para mostrar su información

Ejercicio 4: **dispositivo**

Crear una interfaz llamada **Dispositivo** con los siguientes métodos:

- **encender()**
- **apagar()**
- **mostrarInformacion()**

Crear una interfaz llamada **Electronico** con los siguientes métodos:

- **cargarBateria()**
- **reiniciar()**
- **conectarWiFi()** recibirá dos strings, red y contraseña, obtendrá un número aleatorio (0 o 1) para decidir si la contraseña es correcta. Devolverá un booleano en consecuencia.
- **verificarBateria()** devolverá un entero
- **modoAhorroEnergia()**

Crea la clase **Telefono** que implementa **Dispositivo** y **Electronico** y tendrá:

- Los atributos **bateria**, **ahorroEnergia** y **estado**.

Además, los métodos:

- **encender()** pondrá estado a true.
- **apagar()** pondrá estado a false.
- **mostrarInformacion()** mostrará toda la información del telefono.
- **cargarBateria()** obtendrá un número aleatorio entre 0 y 50 que añadirá a la bateria actual, si se alcanza un valor mayor a 100, se colocará un 100.
- **reiniciar()** pondrá estado a false, ahorroEnergia a false y después volverá a poner estado a true.
- **verificarBateria()** devolverá la batería restante.
- **modoAhorroEnergia()** si la bateria es inferior a 20 se activará.
- **utilizar()** se le pasará una cadena con una aplicación: si es “Whatsapp” disminuirá 5 la bateria, si es “Instagram”, 10 y si es TikTok, 20. Si es cualquier otra cosa disminuirá 1.

Crea la clase **Ordenador** que implementa **Dispositivo** y **Electronico** y tendrá:

- Los atributos **bateria**, **ahorroEnergia** y **estado**.

Además, los métodos:

- **encender()** pondrá estado a true.
- **apagar()** pondrá estado a false.
- **mostrarInformacion()** mostrará toda la información del ordenador.
- **cargarBateria()** realiza un bucle que va desde 0 a un número aleatorio entre 50 y 100. En cada iteración del bucle aumenta en 1 la batería y muestra ese avance por pantalla.

- **reiniciar()** pondrá estado y ahorroEnergía a false. Después volverá a poner estado a true.
- **verificarBateria()** devolverá el valor de la bateria.
- **modoAhorroEnergia()** modifica la variable ahorroEnergia.
- **utilizar()** recibirá un entero que serán las pestañas abiertas, este entero se restará 2 veces a la batería cada vez que se llame al método. Si ahorroEnergia está encendido entonces restará sólo multiplicado por 1.

Crea la clase **Lavadora** que implementa **Dispositivo** y tendrá:

- Los atributos modo que almacenará cadenas, potencia que será entero, estado será booleano y capacidad y capacidad máxima (final) que serán enteros.

Además, los métodos:

- **encender()** pondrá estado a true.
- **apagar()** pondrá estado a false.
- **mostrarInformacion()** mostrará toda la información de la lavadora.
- **restarCapacidad()** eliminará una cantidad pasada por parámetro de la capacidad.

- **aumentarCapacidad()** mientras que la suma entre lo pasado por parámetro y la capacidad actual será menor a la capacidad máxima lo añadirá, sino mostrará un mensaje de error.
- **seleccionarModo()** modo podrá ser tres valores “Corta duración, potencia máxima o sostenible” si el valor pasado coincide con uno de ellos, lo modificará.
- **seleccionarPotencia()** el valor pasado por parámetro se igualará a la variable potencia.

Finalmente crea un main que cree 3 objetos (uno de cada tipo) y prueba todos los métodos asociados a cada uno con diferentes valores (utiliza 4 veces el teléfono con diferentes aplicaciones, apagalo, reinicialo, carga la batería y muestra la información), (utiliza 2 o 3 veces el ordenador, apaga, enciende, reinicia, comprueba la batería, cárgala y muestra la información) y lo mismo con la lavadora.

Puedes añadir los System.out que estimes para que todo el proceso pueda ser seguido.