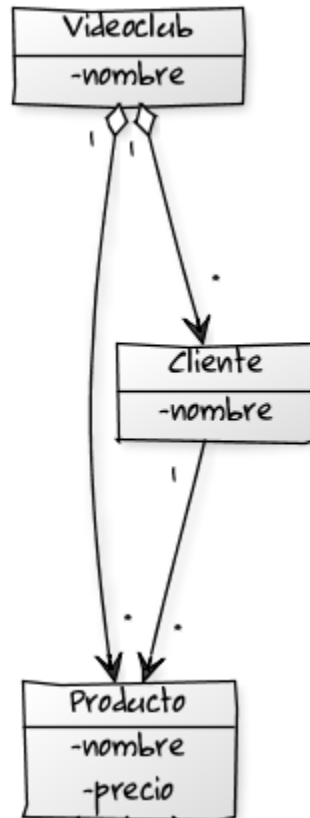


EJERCICIO DE VIDEO CLUB -VERSIÓN 1-

Crear objetos para gestionar un videoclub. El diagrama de clases en UML para representar gráficamente las clases que vamos a utilizar para gestionar los diferentes objetos es el siguiente.

Paquete: paqueteClasesProductoCliente (contendrá las clases Producto y Cliente)

Paquete: paquetePpal (contendrá las clases VideoClub y Ppal)



- La clase **Producto**: representa a un producto.

Cuenta con las propiedades: id, nombre, precio, duracion e idioma, de carácter privado y las propiedades estáticas numProductos y contProductos.

La propiedad estática numProductos lleva la cuenta de los productos activos (sería `SELECT COUNT(*) FROM Producto;` en SQL)

La propiedad estática contProductos indica el número de productos creados en total, incluyendo los que se eliminen (el método `finalize()` lo hará).

El valor de id será el mismo que el de contProductos, que se incrementará conforme se instancien productos nuevos.

El precio y la duración nunca pueden ser negativos o cero.

El idioma ha de ser uno de un conjunto, que será un array de carácter estático con valor fijo: "ESPAÑOL", "INGLES", "FRANCES", "CHINO" y "ALEMAN".

Se ha de gestionar con excepciones para los campos: precio, duración e idioma de tipo `IllegalArgumentException` y llevará asociado un mensaje.

CONSTRUCTORES

1) Constructor cuyos parámetros de entrada sean: nombre, precio, duración e idioma. Actuará como patrón. -Recordad incrementar contProductos, numProductos e id que tomará el valor de contProductos)

2) Constructor cuyo parámetro de entrada es: otroProducto de tipo Producto.

Además añadid los métodos: getId, getNombre, getPrecio, getDuracion, getIdioma, setNombre, setPrecio, setDuracion, setIdioma y toString.

También añadid los métodos estáticos: getNumProductos y getContProductos.

-La clase **Cliente**: representa a un cliente.

Todas las propiedades son privadas.

Cuenta con las propiedades id, nombre y una colección (un ArrayList de Productos) donde se va a guardar los productos alquilados de ese cliente y se denomina productosAlquilados.

Además estarán las propiedades estáticas: numClientes y contClientes.

La propiedad estática numClientes lleva la cuenta de los clientes activos (sería SELECT COUNT(*) FROM Cliente; en SQL)

La propiedad estática contClientes indica el número de clientes creados en total, incluyendo los que se eliminen (el método finalize() lo hará).

El valor de id será el mismo que el de contClientes, que se incrementará conforme se instancien clientes nuevos.

CONSTRUCTOR

Constructor cuyo único parámetro de entrada es el nombre del nuevo cliente.

---Recordad que el ArrayList de Productos se puede inicializar al declarar las propiedades o en el constructor.

MÉTODOS

Añadir además de los métodos: getId, getNombre, setNombre, getProductosAlquilados.

Método: alquilarProducto(objeto Producto), tal que añada al array productosAlquilados ese objetoProducto y no devuelve nada.

Método: devolverProducto(objeto Producto), tal que borre del array productosAlquilados ese objeto y devuelve true si se ha podido borrar o no.

Método: contieneProducto(objetoProducto), tal que compruebe si ese producto ya está alquilado por ese cliente o no

Método: toString, devuelve en una cadena el nombre del cliente y un listado de todos los productos que tiene alquilados (si esa lista no está vacía).

La clase **Videoclub**: es la clase principal de nuestra aplicación. Representa a un videoclub en el dominio del problema y cuenta con el nombre del VideoClub, una colección (ArrayList) de Clientes y otra de Productos registrados, cuya denominación es: clientes y productos, respectivamente, todas ellas son privadas.

CONSTRUCTOR

Constructor cuyo único parámetro de entrada es el nombre del nuevo videoClub.

MÉTODOS:

getNombre, setNombre, getProductos, getClientes.

Método: addProducto(objeto Producto), tal que añada al array productos ese objeto Producto y no

devuelve nada.

Método: addCliente(objeto Cliente), tal que añada al array clientes ese objeto Cliente y no devuelve nada.

Método: getNumProductos, tal que devuelve el número de productos registrados en el videoclub.

Método: getNumClientes, tal que devuelve el número de clientes registrados en el videoclub.

Método: toString, devuelve en una cadena con el nombre del video club más la lista de productos registrados en el video club y la de clientes (si no están vacías).

Método: alquilar, tal que dándole un objeto Producto y otro Cliente, llame al método alquilarProducto de ese objeto de tipo cliente.

Crea la clase Ppal y desde el método main ejecuta:

1. Crear un videoclub que se denomine 'ViedoMax'.
2. Crear 4 clientes: 'Francisco','Javier','Paco','Antonio'.
3. Crear 5 productos: 'El Señor de los Anillos','The Hobbit','Star Wars III','El discurso del Rey','Shrek'.
4. Registrar los 4 clientes en el videoclub (addCliente), y los 5 productos (addProductos).
5. El cliente: 'Francisco' alquila la película: 'El Señor de los Anillos' y 'The Hobbit'./desde el objeto videoclub
6. Mostar el número de películas.
7. Mostrar el número de clientes.
8. El cliente: 'Javier' alquila la película: 'Star Wars III'. //desde el objeto videoclub
9. Obtener la lista de clientes registrados.
10. Obtener la lista de productos registrados.
11. Obtener la lista de productos alquilados por el cliente: 'Francisco'.
12. Crea un producto que posea las mismas características que 'Star Wars III', evidentemente con otro id.
13. Mostrar el número de productos existentes.