

EJERCICIO VECTORES OBJETOS

ACTIVIDAD

Codifica una clase **Videojuego** para tratar la información de los diferentes videojuegos de una tienda. De cada uno de ellos se desea guardar: un código autogenerated, el título, el precio en euros y la cantidad que se tiene de ese videojuego en *stock*.

El código de cada videojuego será un número entero correlativo. Cuando la tienda saca un nuevo videojuego a la venta, se le asignará el código siguiente al código del último videojuego que se creó. Si la tienda deja de vender un videojuego, el código de ese videojuego no se reutilizará para otro videojuego.

Para esta clase **Videojuego**:

- Codifica un constructor que reciba como parámetros el título y el precio. Al instanciar un objeto de esta clase, la cantidad de unidades del videojuego será 10.
- Codifica un método de objeto que devuelva una cadena de texto con el resumen de los atributos del objeto. Para ello, sobrescribe el método **toString** que procede de la clase **Object**. El precio se deberá indicar con 2 dígitos decimales.

Codifica una clase **Almacen** que permita gestionar un contenedor de videojuegos. Esta clase almacena un vector de videojuegos.

Para esta clase **Almacen**:

- Codifica un constructor que reciba como parámetro el número de videojuegos que puede contener (es decir, la capacidad del vector).
- Codifica un método de objeto que devuelva una cadena de texto con el resumen de cada uno de los videojuegos, precedido por su posición dentro del vector, que será el valor que el usuario indique cuando quiera acceder a un videojuego. Para ello, sobrescribe el método **toString** que procede de la clase **Object**.

Si el vector contiene videojuegos, este método devolverá un mensaje como el siguiente:

```
(0) Videojuego[Código=1, Título=BofW, Precio=49.95, Cantidad=10]
(1) Videojuego[Código=2, Título=GTA5, Precio=19.95, Cantidad=10]
(2) Videojuego[Código=3, Título=FH4, Precio=29.95, Cantidad=10]
```

Si el vector no contiene videojuegos, este método devolverá una cadena vacía.

- Codifica un método de objeto **insertar** que tenga como parámetro un videojuego a insertar, que realice la operación cuando el vector no esté lleno, y que devuelva un booleano indicando si la inserción se ha realizado con éxito o no.
- Codifica un método de objeto **consultar** que tenga como parámetro el código de un videojuego a consultar y que devuelva el videojuego cuyo código coincida con el parámetro o **null** si no hay ningún videojuego con ese código.
- Codifica un método de objeto **eliminar** que tenga como parámetro la posición del videojuego a eliminar, que realice la operación cuando la posición referencie a un videojuego en el vector, y que devuelva un booleano indicando si la eliminación se ha realizado con éxito o no. Recuerda que no puede haber nulos por medio del vector, sino que las posiciones vacías se agrupan al final.

- Codifica un método de objeto **modificaCantidad** que tenga como parámetro un precio y modifique la cantidad de todos los videojuegos cuyo precio sea inferior al precio pasado como parámetro sumando 1 a la cantidad. Deberá devolver el número de videojuegos modificado.
- Codifica un método de objeto **buscaVideojuegos** que tenga como parámetro una cantidad y devuelva todos los videojuegos cuya cantidad coincida con el parámetro de entrada.

Codifica una clase **Principal** que incluya un programa principal **main** que gestione un almacén de videojuegos (con capacidad para 20 videojuegos) con el menú de opciones siguiente:

0) **Salir del programa.**

1) **Insertar un videojuego en el almacén.**

Leerá por teclado el título y el precio del videojuego a insertar.

Intentará realizar la inserción del videojuego, creado a partir de los datos leídos desde teclado, en el almacén.

Si el almacén está lleno (con 20 videojuegos), visualizará en consola el mensaje:

Error al insertar: almacén lleno.

Si la inserción se ha realizado con éxito, visualizará en consola el mensaje:

Se ha insertado el videojuego en el almacén con éxito.

2) **Eliminar un videojuego, por posición, del almacén.**

Leerá por teclado la posición del videojuego a eliminar.

Intentará realizar la eliminación del videojuego, indicado mediante la posición leída desde teclado, del almacén.

Si el almacén está vacío (sin videojuegos) o si no hay ningún videojuego en dicha posición dentro del almacén, visualizará en consola el mensaje:

Error al eliminar

Si la eliminación se ha realizado con éxito, visualizará en consola el mensaje:

Se ha eliminado el videojuego del almacén con éxito.

3) **Consultar un videojuego, por código, del almacén.**

Leerá por teclado el código del videojuego a consultar y si el almacén contiene un videojuego con dicho código, visualizará en consola el videojuego encontrado, indicando un resumen con los valores de todos los atributos de dicho videojuego.

En caso contrario, visualizará en consola el mensaje:

No se ha encontrado ningún videojuego con ese código.

4) **Consultar todos los videojuegos del almacén.**

Realizará la consulta de todos los videojuegos del almacén.

Si no hay videojuegos en el almacén, visualizará en consola el mensaje:

El almacén está vacío.

En caso contrario, visualizará en consola el listado de todos los videojuegos contenidos en el almacén, indicando para cada videojuego, su posición dentro del almacén y el resumen con los valores de todos los atributos de dicho videojuego. Por ejemplo:

(0) Videojuego[Código=1, Título=BotW, Precio=49.95, Cantidad=10]

(1) Videojuego[Código=2, Título=GTA5, Precio=19.95, Cantidad=10]

(2) Videojuego[Código=3, Título=FH4, Precio=29.95, Cantidad=10]

5) **Modificar las cantidades dado un precio.**

Leerá por teclado un precio e incrementará las cantidades de los videojuegos cuyo precio sea inferior al leído.

Saldrá por pantalla el número de videojuegos modificados.

6) **Busca videojuegos**

Leerá por teclado una cantidad y mostrará todos los videojuegos de los que haya esa cantidad de stock en el almacén.

Este programa principal deberá validar que la opción de menú elegida sea válida (comprendida entre 0 y 6). Si no lo es, visualizará en consola el mensaje:

La opción de menú debe estar comprendida entre 0 y 6.