

ACTIVIDAD 1

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** autogenerado, 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 (por defecto).

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. Además de un **vector** de videojuegos, esta clase contendrá un **índice** que tendrá una doble función: marcará la tanto la primera posición libre del vector de videojuegos, como el número de elementos que éste contiene instanciados.

Para esta clase **Almacen**:

Codifica un **constructor** que reciba como parámetro el número de videojuegos que puede contener (es decir, la capacidad del vector). Al instanciar un objeto de esta clase, el índice del almacén será 0.

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 no hay videojuegos en el vector, este método devolverá el mensaje:

El almacén está vacío.

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]

Codifica un método de objeto **consultar** que tenga como parámetro la posición del videojuego a consultar y que devuelva el videojuego que se encuentra en dicha posición en el vector o null si no hay ningún videojuego en esa posición.

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 **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.

Codifica una clase **Actividad01** que incluya un programa principal **main** que gestione un almacén. Inserta al menos 3 videojuegos y muestra el contenido del almacén.

ENUNCIADOS DE EJEMPLOS DE HERENCIA Y POLIMORFISMO

Ejemplo 1 – Ejemplo extends

Escribe un programa en Java que pida los datos de un Empleado. De dicho empleado nos interesa: **nombre** y **suelo**. A continuación pedirá los datos de un **Jefe**, del que nos interesan los mismos datos que de un empleado y además su **título** universitario y el nombre del **departamento** del que es responsable.

Ejemplo 2 – Actividad Jerarquía

Escribe un programa en Java que permita:

- Pedir datos de seres vivos
- Visualizar dichos datos y alimentarlos

El programa tendrá que utilizar 4 clases:

- **La clase Principal.** Esta clase contiene el método main, que es por donde va a arrancar y cerrar el programa. Instancia un objeto de cada tipo y aliméntalo.
- **La clase Animal.** De los animales nos interesa conocer el número de crías que tienen. Esta clase tendrá métodos para:
 - Pedir sus datos.
 - Visualizar sus datos.
 - Alimentar al animal.
- **La clase Mamífero.** Contiene datos específicos como el número de huesos que tiene y el número de extremidades. Hay que escribir los métodos necesarios para manejar sus datos exclusivos (como mamífero).
- **La clase Persona.** Contiene datos exclusivos del ser humano como, por ejemplo, su profesión. Hay que escribir los métodos necesarios para manejar sus datos exclusivos (como persona).

En este ejemplo la clase Persona hereda de la clase Mamífero y la clase Mamífero hereda de la clase Animal.

Ejemplo 3 - Herencia de Constructor

Ejemplo 4 – Actividad herencia

Ejemplo 5 - Polimorfismo

Clase Persona:

- dni
- nombre
- edad
- + imprimeDatos()

Clase Alumno extendida de Persona:

- matricula
- + imprimeDatos()

Clase Profe extendida de Persona:

- asignatura
- + imprimeDatos()

Clase Principal:

1. Instancia un vector para 6 Personas (2 objetos de cada tipo)
2. Recorre el vector imprimiendo los datos de cada tipo. "método imprimeDatos()"
3. Recorre el vector reconociendo la instancia de cada tipo. (instanceof)
4. Saca los datos de las personas, luego los profes y por último los alumnos.

Ejemplo 6 – Abstracto

Clase abstracta Figura:

- color
- + area();

Clase Cuadrado:

- base
- + area()

Clase Circulo:

- radio
- + area()

Clase Triangulo:

- base
- altura
- + area()

Clase Principal:

1. Instancia un vector para 6 Figuras (2 objetos de cada tipo)
2. Recorre el vector imprimiendo el area de cada uno.
3. Recorre el vector reconociendo la instancia de cada tipo. (instanceof)

ENUNCIADOS DE EJEMPLOS DE HERENCIA Y POLIMORFISMO

Ejemplo 7

Clase Abstracta Vehiculo:

- matricula
- numruedas
- + ponerEnMarcha();

Clase Moto:

- cilindrada
- + ponerEnMarcha "la moto con matricula XXX se pone en marcha"

Clase Coche:

- numAsientos
- + ponerEnMarcha "El coche con matricula XXX se pone en marcha"

Clase Principal

- Instanciar un vector con 4 vehiculos y
- poner en marcha de forma aleatoria

Ejemplo 8 – interface

Implementa la interface Obligatorio para pedir los datos por teclado y visualizar los datos:

```
public interface Obligatorio {  
    public void pedirTodosLosDatos();  
    public void visualTodosLosDatos();  
}
```

Implementa esta interface en el ejercicio 5 en la clase **Persona**.

ENUNCIADOS DE EJEMPLOS DE HERENCIA Y POLIMORFISMO

Ejemplo – vehículos

Clase Abstracta Vehiculo:

- matricula
- numruedas
- + abstract ponerEnMarcha

Clase Moto:

- cilindrada
- + ponerEnMarcha "la moto con matricula XXX se pone en marcha"

Clase Coche:

- numAsientos
- + ponerEnMarcha "El coche con matricula XXX se pone en marcha"

Clase Principal

Instanciar un vector con 4 vehiculos y
poner en marcha de forma aleatoria