

GUÍA DE EJERCICIOS

DESARROLLO WEB FULL
STACK CON JAVA

Nº 8

ING. LUISINA DE PAULA

#

DESARROLLO
WEB FULLSTACK
con JAVA

Guía de Ejercicios Nº 8: Programación Orientada a Objetos - Parte 2

1) Repaso Objetos

- a. Crear una clase llamada VideoJuego, que tenga los siguientes atributos: codigo, titulo, consola, cantidadJugadores, categoría (tener en cuenta todos sus atributos, constructores, métodos getters y setters).
 - ✓ Crear un vector de tipo VideoJuego que pueda almacenar 5 videojuegos. Crear 5 videojuegos y guardarlos en el vector.
 - ✓ Recorrer el vector creado y mostrar por pantalla el titulo, consola y cantidad de jugadores de los videojuegos almacenados.
 - ✓ Cambiar el nombre y la cantidad de jugadores de dos videojuegos. Mostrar por pantalla los datos de todos los videojuegos luego del cambio.
 - ✓ Recorrer el vector y mostrar por pantalla únicamente a los videojuegos que sean de la consola "Nintendo 64".

- b. Crear una clase llamada Fruta, que tenga los siguientes atributos: nombre, color, calorías, tipoCascara, esComestible (tener en cuenta todos sus atributos, constructores, métodos getters y setters).
 - ✓ Crear un vector de tipo Fruta que pueda almacenar 5 frutas. Crear 5 frutas y guardarlas en el vector.
 - ✓ Recorrer el vector creado y mostrar por pantalla el nombre y las calorías de las frutas almacenadas.
 - ✓ Cambiar todos los datos de dos frutas. Mostrar por pantalla los datos de todas las frutas luego del cambio.
 - ✓ Recorrer el vector y mostrar por pantalla únicamente a las frutas que sean de color verde.

2) Herencia, Polimorfismo y encapsulamiento

Ejercicio Nº 1

- Crear una clase **Planta** con los atributos: nombre, alto del tallo, tieneHojas, clima ideal. (con sus métodos y constructores correspondientes)
- Crear sus clases hijas que compartan sus atributos y métodos:
 - **Árbol**: variedad, tipo de tronco, radio de tronco, color, tipo de hojas.
 - **Flor**: color de pétalos, cantidad promedio de pétalos, color del pistilo, color de los pétalos, variedad de flor, estación que florece
 - **Arbusto**: Ancho arbusto, esDomestico, variedad arbusto, color de hojas, sePodaONo
- Una vez creada las clases crear los siguientes métodos:
 - **Árbol**: método para mostrar un mensaje en pantalla que diga “Hola soy un árbol”
 - **Flor**: método para mostrar un mensaje en pantalla que diga “Hola soy una flor”
 - **Arbusto**: método para mostrar un mensaje en pantalla que diga “Hola soy un arbusto”
- Crear en el Main un objeto de cada clase hija. Llamar a los 3 métodos desde cada objeto.
- Cambiar el modificador de acceso de los métodos de public a private. Intentar acceder desde el main a estos métodos.

Ejercicio Nº 2

- Crear una clase Vehiculo con los atributos: patente, num de chasis, motor, color, marca, modelo cantidad de asientos (con sus métodos y constructores correspondientes).
- Crear sus clases hijas que compartan sus atributos y métodos:
 - Auto: materialasientos, cantidad_caballos
 - Moto: cilindrada, material_manubrio
 - Colectivo: aptoDiscapacitados, poseeLectorSube, tipoColectivo
 - Camion: tieneAcoplado, cantidadEjes
- Una vez creada las clases, crear en el Main un vector de tipo Vehículo y almacenar 3 autos, 3 motos, 2 colectivos y 1 camión (crear un objeto para cada uno de ellos).
- Crear los siguientes métodos (en cada subclase correspondiente):
 - Auto: método para mostrar un mensaje en pantalla que diga “Hola soy un auto y mi marca es: ” (mostrar marca)
 - Moto: método para mostrar un mensaje en pantalla que diga “Hola soy una moto

y mi cilindrada es de: " (mostrar cilindrada)

- Colectivo: método para mostrar un mensaje en pantalla que diga "Hola soy un colectivo y mi cantidad de asientos es de: " (mostrar cantidad de asientos).
- Camion: método para mostrar un mensaje en pantalla que diga "Hola soy un camión y mi cantidad de Ejes es de: " (mostrar cantidad de ejes).
- Recorrer el vector y ejecutar el método que corresponde en cada posición del mismo. PISTA: *Todos tendrán el mismo método, con el mismo nombre pero únicamente cambia el mensaje que muestran...* ¿Cómo se puede hacer para que todos posean el mismo método pero solo cambien sus datos que muestran? IMPLEMENTAR HERENCIA y sobreescritura de métodos.