

Ejercicio Concesionario

(Abstract/Polimorfismo/Private/Protected/)

Se requiere desarrollar un sistema de gestión para un concesionario de vehículos que venda automóviles y motocicletas. Cada vehículo tiene atributos como modelo, año de fabricación y precio base. Los vehículos se dividen en dos categorías principales: automóviles y motocicletas.

Los automóviles se caracterizan por tener un número específico de puertas, mientras que las motocicletas se caracterizan por su cilindrada.

Se necesita implementar un programa en Java que modele esta situación utilizando clases y herencia con polimorfismo. Cada clase debe representar un tipo de vehículo, y debe haber una clase abstracta llamada "Vehiculo" que contenga los atributos comunes y métodos abstractos que todas las clases hijas deben implementar.

Además, se necesita un método para calcular el precio de venta de cada vehículo. Este método debe tener en cuenta un descuento por antigüedad, que será del 5% del precio base por cada año de antigüedad del vehículo. Para los automóviles, se debe sumar una cantidad fija adicional de 1000 euro al precio de venta.

El programa debe incluir un método main que cree instancias de automóviles y motocicletas, y muestre el precio de venta calculado para cada uno.

El sistema debe incluir las siguientes clases:

(0,5 ptos) -

a1) Clase abstracta "Vehiculo":

- Atributos:(todos protected)
 - modelo: el modelo del vehículo.
 - añoFabricacion: el año de fabricación del vehículo.
 - precioBase: el precio base del vehículo.
- Métodos:
 - Constructor que inicializa los atributos.
 - Método abstracto calcularPrecioVenta() que calcula y devuelve el precio de venta del vehículo.

(0,75 ptos) -

b1) Clase "Automovil" :

- Atributos adicionales:
 - numeroPuertas: el número de puertas del automóvil.
- Métodos:

- Constructor que inicializa los atributos.
- Implementación del método calcularPrecioVenta() que tiene en cuenta el descuento y la cantidad adicional para automóviles.

(0,75 ptos) -

c1) Clase "Motocicleta":

- Atributos adicionales:
 - cilindrada: la cilindrada de la motocicleta.
- Métodos:
 - Constructor que inicializa los atributos.
 - Implementación del método calcularPrecioVenta() que tiene en cuenta el descuento.

(1 ptos) -

d1) Añade otro método abstracto a la clase "Vehículo" e implementa los métodos en las clases que especializan de ella.

(1 ptos) -

e1) Además, incluye un método main en una clase separada para probar el funcionamiento del sistema, creando instancias de automóviles y motocicletas y mostrando el precio de venta calculado para cada uno. También debes probar el método añadido en el punto anterior.

Instancias a probar en el main:

Automóvil auto = new Automovil("Toyota Corolla", 2018, 20000, 4)

Motocicleta moto = new Motocicleta("Honda CB500", 2020, 10000, 500)

f) Añade en el código comentarios explicando brevemente donde aparezcan aplicados los siguientes conceptos:

- **Abstracción - (0.5 ptos)**
- **Polimorfismo - (0.5 ptos)**
- **Encapsulamiento - (0.5 ptos)**
- **Variables de clase - (0.5 ptos)**

(1 ptos) - g) Una vez hecho y comprobado su funcionamiento correcto:

1.- Adjuntar el proyecto exportado Concesionario.zip a la tarea

(3 ptos) - 2.- Modificar las clases como se indica a continuación y vuelve a enviar Concesionario2.zip:

- la visibilidad de los atributos de la Clase abstracta "Vehiculo" a **private**
- añadir y modificar los métodos necesarios para que todo funcione correctamente
- Todos los precios y descuentos deben mostrarse hasta la segunda cifra decimal.
- El resultado debe mostrar ordenadamente para las instancias auto y moto; el precio base, el precio con descuento y la frase "Usted ha ahorrado: \$" con el descuento total aplicado.
- Volver a comentar el en qué ha influido el cambio de tipo de Encapsulamiento donde corresponda.