



Reto 2 – Concesionario

Objetivo:

El objetivo de este reto es que el estudiante reconozca y aplique los elementos básicos del paradigma de la programación orientada a objetos en un escenario abstraído de la cotidianidad.

Contexto:

CARMAX S.A.S adquirió un nuevo lote de autos que podrá en exhibición y dará un bono de comisión a aquellos empleados que logre vender los siguientes tipos de autos:

- Bono por \$750.000 para aquellos que vendan autos deportivos o camionetas
- Bono por \$500.000 para aquellos que vendan autos eléctricos
- Bono por \$350.000 para aquellos que vendan autos tradicionales

La representación de un auto en el sistema del concesionario **CARMAX S.A.S** se puede visualizar de la siguiente manera:

Auto
- marca: String - tipo: int
+ Auto(marca:String, tipo:int) + getters + setters

Los diferentes tipos de automóviles estarán representados de forma numérica, **(1)** para deportivos y camionetas, **(2)** para autos eléctricos y **(3)** para autos tradicionales.

Los empleados del concesionario tendrán asociado ahora un número de autos vendidos, y según la cantidad y tipo de auto vendido, se conocerá el valor ganado por concepto de comisión de cada uno de los empleados durante un mes de trabajo. Con este nuevo atributo los empleados se representarían de la siguiente manera:



Empleado
- id: int - nombre: String - apellido: String - autosVendidos: ArrayList<Auto> - salario: int
+ Empleado(nombre:String, apellido:String, salario:int) + calcularMiNomina(Empleado empleado): double + getters + setters

Reto:

Nota: Continúe estructurando las clases y métodos del proyecto **Concesionario** implementadas en el *Reto1*, y realice los cambios pertinentes(refactorización) según los requerimientos del reto actual.

A partir de la nueva información brindada sobre los empleados del concesionario **CARMAX S.A.S.** Desarrolle el método **calcularMiNomina**, recuerde que para llevar a cabo el cálculo de la nómina se deben de deducir los aportes a salud y pensión del total devengado de cada empleado.

Un ejemplo de la estructura el código de las clases a implementar se muestra a continuación:

```
public class Empleado {  
    private int id;  
    private String nombre;  
    private String apellido;  
    private ArrayList<Auto> autos = new ArrayList<>();  
    private int salario;  
  
    public Empleado(String nombre, String apellido, int salario){  
    }  
  
    public static double calcularMiNomina(Empleado empleado){  
    }  
}
```



```
public class Auto {  
    private String marca;  
    private int tipo;  
  
    public Auto(String marca, int tipo) {  
  
    }  
  
}
```

Casos de prueba:

Para verificar el funcionamiento de su programa se sugiere que tenga en consideración los siguientes casos de prueba

# CASO DE PRUEBA	DATOS DE ENTRADA	SALIDA ESPERADA								
calcularMiNomina		Valor: 2.875.00000								
	<table><tr><th>Empleado</th><th>Autos Venidos</th></tr><tr><td>Nombre: Diego</td><td>Auto Eléctrico</td></tr><tr><td>Apellido: Moreno</td><td>Camioneta</td></tr><tr><td>Salario: \$1'875.000</td><td></td></tr></table>		Empleado	Autos Venidos	Nombre: Diego	Auto Eléctrico	Apellido: Moreno	Camioneta	Salario: \$1'875.000	
	Empleado		Autos Venidos							
	Nombre: Diego		Auto Eléctrico							
	Apellido: Moreno		Camioneta							
Salario: \$1'875.000										

Entrega:

1. Suba a la plataforma un archivo con el nombre de **Empleado.java**, este nombre debe de respetarse, dado que, si no se nombre de dicha manera no se tendrá en cuenta para la calificación del reto.
2. De igual manera, llame clase que representa a los carros del concesionario **Auto.java** y suba este archivo de igual manera a la plataforma.
3. **Importante:** Los métodos deben de llamarse **exactamente igual** a como se muestra en el ejemplo de la estructura del código.