TAREA 3: DGT

En esta tarea se codificar el diagrama UML que me muestra en la Figura1.

A continuación, se describirá cada una de clase y sus métodos:

1. Clase Propietario.

Clase que modela un objeto Propietario correspondiente a 3 atributos, el constructor junto los getters&setters y toString.

2. Clase Estado.

Clase enum con 3 constantes.

3. Clase Vehiculo

Clase abstracta que actúa como padre de clase Moto y clase Coche. Con 5 atributos (1 de ellos es un objeto de la clase Propietario y otro un estado dentro de las posibilidades de la clase Enum).

Esta clase únicamente proporciona el constructor, los getters y setters e introduce como abstracto el método calcularImpuesto.

4. Clase Moto

Proporciona a mayores 1 atributo y codifica el método calculoImpuesto()

- **calculoImpuesto():** Se aplica la siguiente tabla del impuesto aplicado según el cv de la moto.

cv	impuesto
0 - 50	100
50 - 150	150,5
150 - 250	202,1
> 250	255,4

5. Clase Coche

Proporciona a mayores 2 atributos y codifica el método calculoImpuesto()

- **calculoImpuesto()**: Se aplicará la siguiente formula : **50 + (numAsientos * 50).** Además, en caso de que el coche no tenga **airbags** se aplica un **incremento de 45 euros**.

6. Clase DGT.

Almacena en dos colecciones ArraysList el conjunto de Vehículos y Propietarios registrados.

Se llevará a cabo los siguientes métodos:

- **altaVehiculo(vehiculo: Vehiculo) : boolean**. Se ingresa en listadoVehiculo un vehículo que se le pasa como argumento y retorna true. Esto se realizará sólo y cuando no haya otro vehículo con la misma matrícula ya en el listadoVehiculo en ese caso no se ingresará el vehiculo y se retornará false.

- **altaPropietario(dni: String, nombre: String, apellidos: String) : boolean**. Se ingresa en listadoPropietarios un nuevo propietario a partir de todos sus datos y retorna true. Esto se realizará sólo y cuando no haya otro propietario con el mismo dni ya en el listadoPropietario en ese caso no se ingresará el propietario y se retornará false.

- **bajaVehiculo(matricula: String) : boolean**. Si hay un vehículo con esa matricula en listadoVehiculo se retirará y retornaremos true, sino retornaremos false.

- **bajaPropietario(dni: String) : boolean**. Si hay un propietario con ese dni en listadoPropietarios se retirará y retornaremos true, sino retornaremos false.
  En caso de retirar un propietario, buscaremos en listadoVehiculos los vehículos cuyo dni del propietario coincida con el propietario que hemos dado de baja y también retiraremos dichos vehículos.

- **listarVehiculos(): void**. Mostramos por terminal todos los vehículos y sus atributos que hay en listadoVehiculos.

- **listarVehiculos(): void.** Mostramos por terminal todos los propietarios y sus atributos que hay en listadoPropietarios.

- **obtenerVehiculo (matricula: String): Vehiculo**. Retorna un objeto de la clase Vehiculo en caso de que haya un elemento en listadoVehiculos cuya matrícula coincida. En caso de que no haya ninguna coincidencia retornaríamos null.

- **obtenerPropietario (dni: String): Propietario.** Retorna un objeto de la clase Propietario en caso de que haya un elemento en listadoPropietarios cuyo dni coincida. En caso de que no haya ninguna coincidencia retornaríamos null.

- **obtenerVehiculosFiltroPropietario(dni: String): Vehiculo[].** Retornar todos los vehículos cuyo dni del propietario del mismo coincida con el que se pasa por argumento. Retornará un vector de objetos de la clase Vehiculo con todos los objetos y en caso de que no haya ninguna coincidencia se retornará el vector, pero vacío.

- **obtenerVehiculosFiltroEstado(dni: String): Vehiculo[].** Retornar todos los vehículos cuyo estado coincida con el que se pasa por argumento. Retornará un vector de objetos de la clase Vehiculo con todos los objetos y en caso de que no haya ninguna coincidencia se retornará el vector, pero vacío.

7. Clase Main.

Crea una clase main donde pruebas cada uno de los métodos de la clase DGT. Para ello se realizará un menú compuesto por las siguientes opciones:

1. Listar propietarios.
2. Listar vehículos.
3. Dar de alta un propietario.
4. Dar de alta un vehículo.
5. Dar de baja un propietario.
6. Dar de baja un vehículo.
7. Obtener información de un vehículo filtrando por matricula.
8. Obtener información de un propietario filtrando por dni.
9. Obtener información de los vehículos filtrando por propietario.
10. Obtener información de los vehículos filtrando por estado.

-1. Salir

El objetivo del menú es facilitar la verificación de cada uno de los métodos, por lo que se mostrará todos los mensajes que el programador consideré oportuno a fin de ejemplificar el correcto funcionamiento de la plataforma programada.

ENTREGA:

- Codificar cada una de las clases anteriores y el main.
- Comprimirlo todo y subirlo al apartado de tareas dentro del plazo previsto.

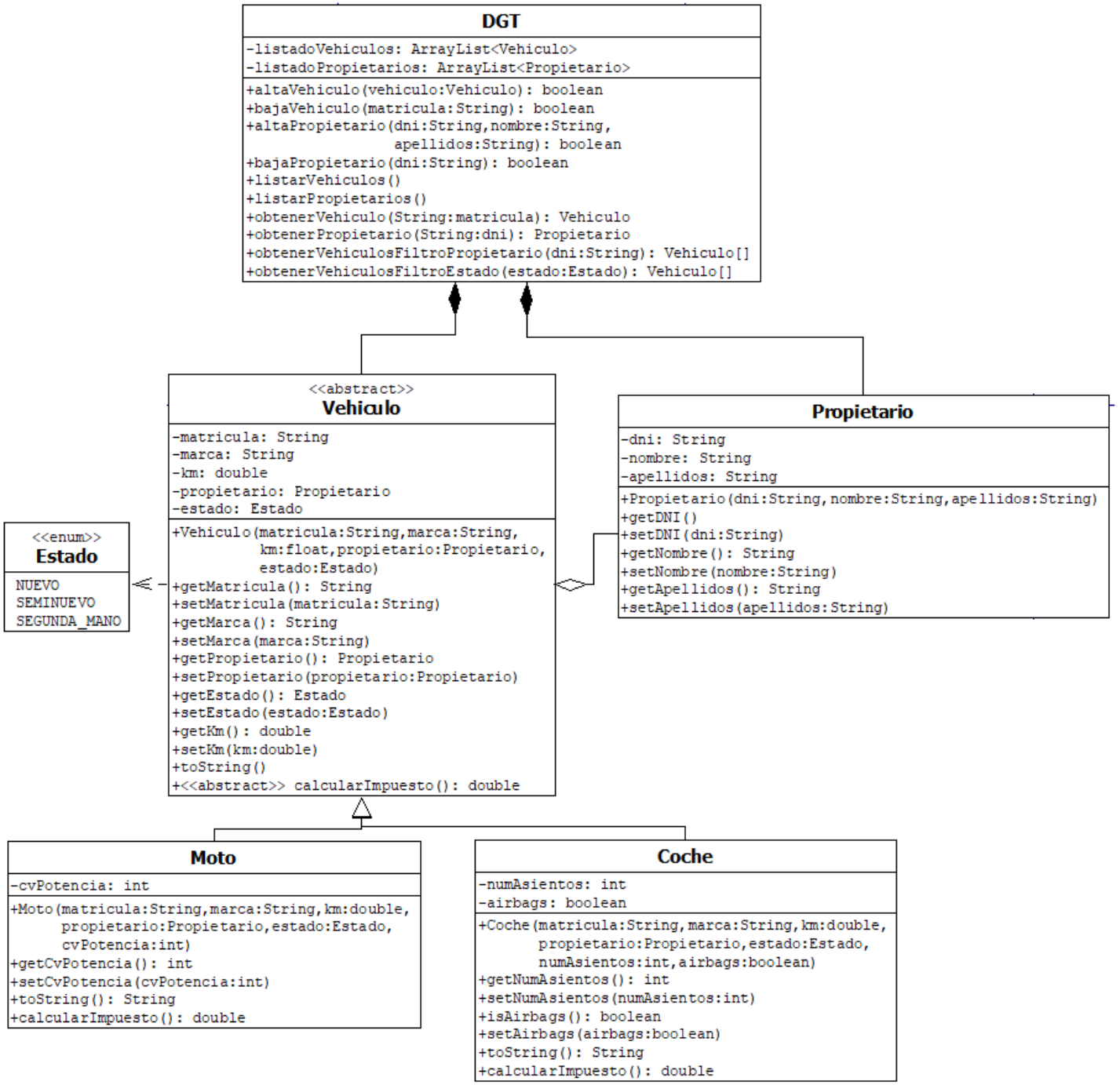

Figura1. Diagrama UML tarea DGT