

VEHICULO – LAVADERO 2016

Realizar la clase **Vehiculo** que posea como atributos protegidos:

- `_patente` : string (propiedad sólo lectura)
- `_cantRuedas` : Byte
- `_marca` : enum EMarcas (con los siguientes enumerados: Honda, Ford, Zanella, Scania, Iveco y Fiat)

Y los siguientes métodos:

- `(~) Mostrar()` : string
- `(+) AcelerarHasta(Byte)` : void
- `(+) Vehiculo (string, Byte, EMarcas)` (sin sobrecargas)

Sobrecarga de operadores:

- `(+) == (Vehiculo, Vehiculo)` : bool. Si las patentes y marcas son iguales, retorna TRUE.

Además se pide:

Crear tres clases (Auto, Camion y Moto) que hereden de Vehiculo y que posean: cantidadAsientos (int), tara (float) y cilindrada (float) como atributos respectivamente. Cada una de estas clases deberá implementar el método `MostrarAuto`, `MostrarCamion` y `MostrarMoto` (reutilizando código de la clase base) para poder retornar un **string** con todos sus atributos.

Por último se desea construir la clase Lavadero que tendrá como atributos:

- `(-) _vehiculos` : List<Vehiculo>
- `(-) _precioAuto` : float
- `(-) _precioCamion` :float
- `(-) _precioMoto` :float

Todos los atributos se inicializaran desde su constructor con parámetros. El constructor por default, que será privado, será el único encargado de inicializar la lista genérica.

Los métodos que tendrá Lavadero son:

- `MostrarTotalFacturado`: devolverá la ganancia del lavadero (Double), dicho método tendrá una sobrecarga que reciba como parámetro la enumeración EVehiculos (con Auto, Camion y Moto como enumerados) y retornará la ganancia del Lavadero por tipo de vehículo.
- Sobrecarga `==` entre un lavadero y un vehiculo, retornara TRUE, si el vehiculo se encuentra en el lavadero.
- Sobrecarga del operador `+`, que agregara un vehiculo siempre y cuando el vehiculo no se encuentre en el lavadero.
- Sobrecarga del operador `-`, que quitara al vehiculo del lavadero, siempre y cuando este dicho vehiculo.