Reto 2

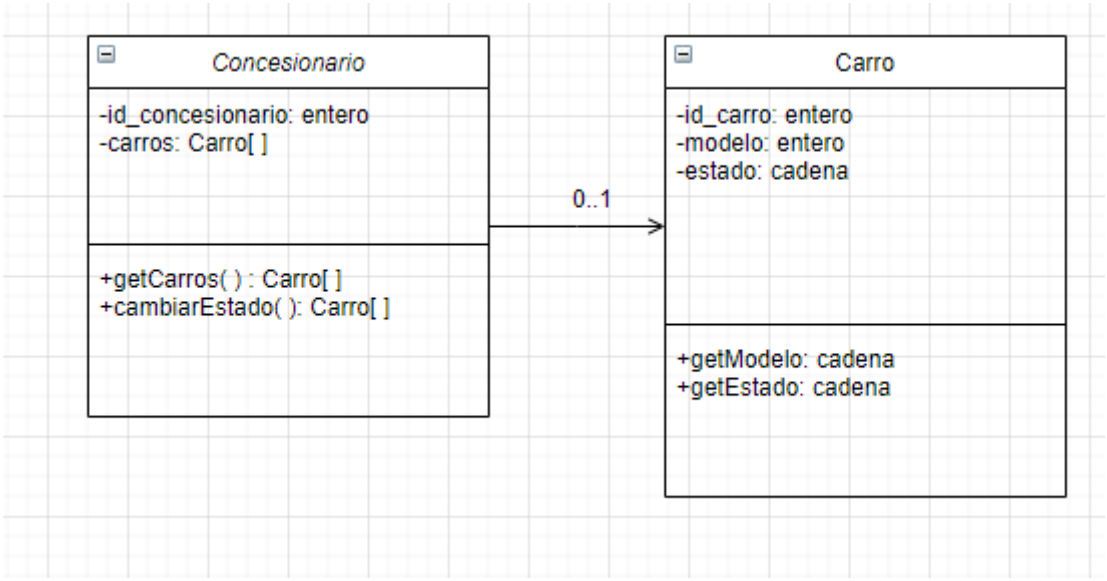
Objetivo

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

El concesionario de carros “AutoMax” está en búsqueda de renovar su catálogo, de tal forma que puedan darle un nuevo aire a su sistema de facturación en línea. Por lo tanto, se requiere un programa el cual verifique el modelo de los autos y en caso de que esté obsoleto, actualice su estado para luego ser retirado del catálogo del concesionario siguiendo ciertas especificaciones.

Considere el siguiente diagrama de clases para la implementación de las clases **Concesionario.java y Auto.java**



NOTA: Las clases deben llamarse tal cual se especifica.

Reto

Implemente una función llamada cambiarEstado() ubicada en la clase Concesionario, la cual debe recorrer un arreglo de tipo Carro, verificando qué tan obsoleto son los vehículos en el concesionario. La función debe retornar el arreglo después de editado el estado de cada vehículo, dependiendo la antigüedad del modelo. Este estado se debe modificar teniendo en cuenta los siguientes casos:

a) Si el modelo del vehículo es posterior o igual al 1997 pero anterior o igual al 2005, el vehículo es considerado “obsoleto” y se debe cambiar el estado por este valor.
b) Por el contrario, si el modelo del vehículo es posterior al 2005 pero anterior o igual al 2012, el vehículo es considerado “vigente” y se debe actualizar el estado por este valor.
c) Finalmente, si el modelo del vehículo es posterior al 2012, el vehículo es considerado “nuevo” y se debe actualizar el estado por este valor.

Adicionalmente, use las siguientes imágenes como referencia para la construcción de la clase Concesionario y Carro con sus atributos y métodos necesarios.

```java
public class Concesionario {

    private int id_concesionario;
    private Carro[] carros;

    public Concesionario(int id_concesionario) {

    }

    public Carro[] cambiarEstado() {


        return carros;
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub

    }

    int getId_concesionario() {
        return id_concesionario;
    }

    void setId_concesionario(int id_concesionario) {
        this.id_concesionario = id_concesionario;
    }

    Carro[] getCarros() {
        return carros;
    }

    void setCarros(Carro[] carros) {
        this.carros = carros;
    }

}
```

```java
public class Carro {

    private int id_carro;
    private int modelo;
    private String estado;


    public Carro(int id_carro, int modelo) {


    }

    int getId_carro() {
        return id_carro;
    }
    void setId_carro(int id_carro) {
        this.id_carro = id_carro;
    }
    int getModelo() {
        return modelo;
    }
    void setModelo(int modelo) {
        this.modelo = modelo;
    }
    String getEstado() {
        return estado;
    }
    void setEstado(String estado) {
        this.estado = estado;
    }


}
```

CASOS DE PRUEBA

Finalmente, para verificar el funcionamiento del programa se sugiere considerar los siguientes casos de prueba:

<table>
<tr><th># CASO DE PRUEBA</th><th>DATO DE ENTRADA</th><th>SALIDA ESPERADA</th></tr>
<tr><td>1</td><td>
<table>
<tr><th>Carro 1</th><th>Carro 2</th><th>Carro 3</th></tr>
<tr><td>Id_carro: 0
modelo: 1998
estado:”TBD”</td><td>Id_carro: 1
modelo: 2003
estado:”TBD”</td><td>Id_carro: 2
modelo: 2008
estado:”TBD”</td></tr>
</table>
</td><td>-El estado del carro 1 es obsoleto

-El estado del carro 2 es vigente

-El estado del carro 3 es vigente</td></tr>
<tr><td>2</td><td>
<table>
<tr><th>Carro 1</th><th>Carro 2</th><th>Carro 3</th></tr>
<tr><td>Id_carro: 0
modelo: 2009
estado:”TBD”</td><td>Id_carro: 1
modelo: 2001
estado:”TBD”</td><td>Id_carro: 2
modelo: 2017
estado:”TBD”</td></tr>
</table>
</td><td>-El estado del carro 1 es vigente

-El estado del carro 2 es obsoleto

-El estado del carro 3 es nuevo</td></tr>
</table>

3	<table><tr><th>Carro 1</th><th>Carro 2</th><th>Carro 3</th></tr><tr><td>Id_carro: 0 modelo: 2002 estado:”TBD”</td><td>Id_carro: 1 modelo: 1997 estado:”TBD”</td><td>Id_carro: 2 modelo: 2021 estado:”TBD”</td></tr></table>	-El estado del carro 1 es obsoleto -El estado del carro 2 es obsoleto -El estado del carro 3 es nuevo

Entrega:

1. Suba a la plataforma un archivo con el nombre de **Concesionario.java y Carro.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. **Importante:** Los métodos deben de llamarse **exactamente igual** a como se muestra en el ejemplo de la estructura del código.

3. **Importante:** Las salidas deben ser tal cual se muestran en los casos de pruebas. De lo contrario, el sistema no lo reconocerá.