

UNIVERSIDAD POLITÉCNICA DE TLAXCALA

Ingeniería en tecnologías de la información

5° “H”

Fundamentos de la Programación

-Practica 4 Clase Carro-

Eric Jhonathan Anaya Márquez

Jazmín Portillo Michicol

Eder López Villarreal

Karen García Leticia Vásquez

18/01/2024

Profesor Saúl Olaf Loaiza

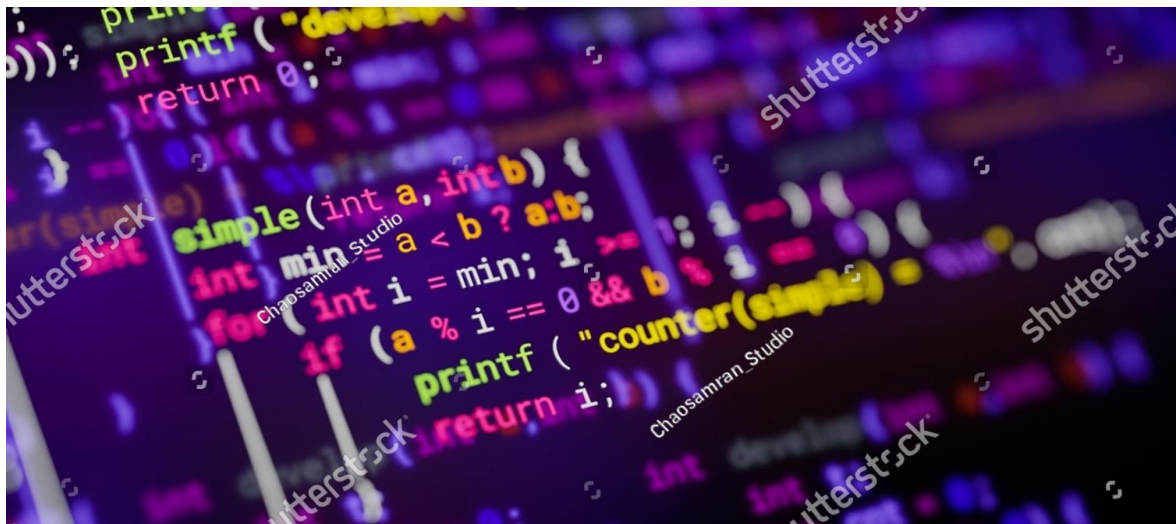
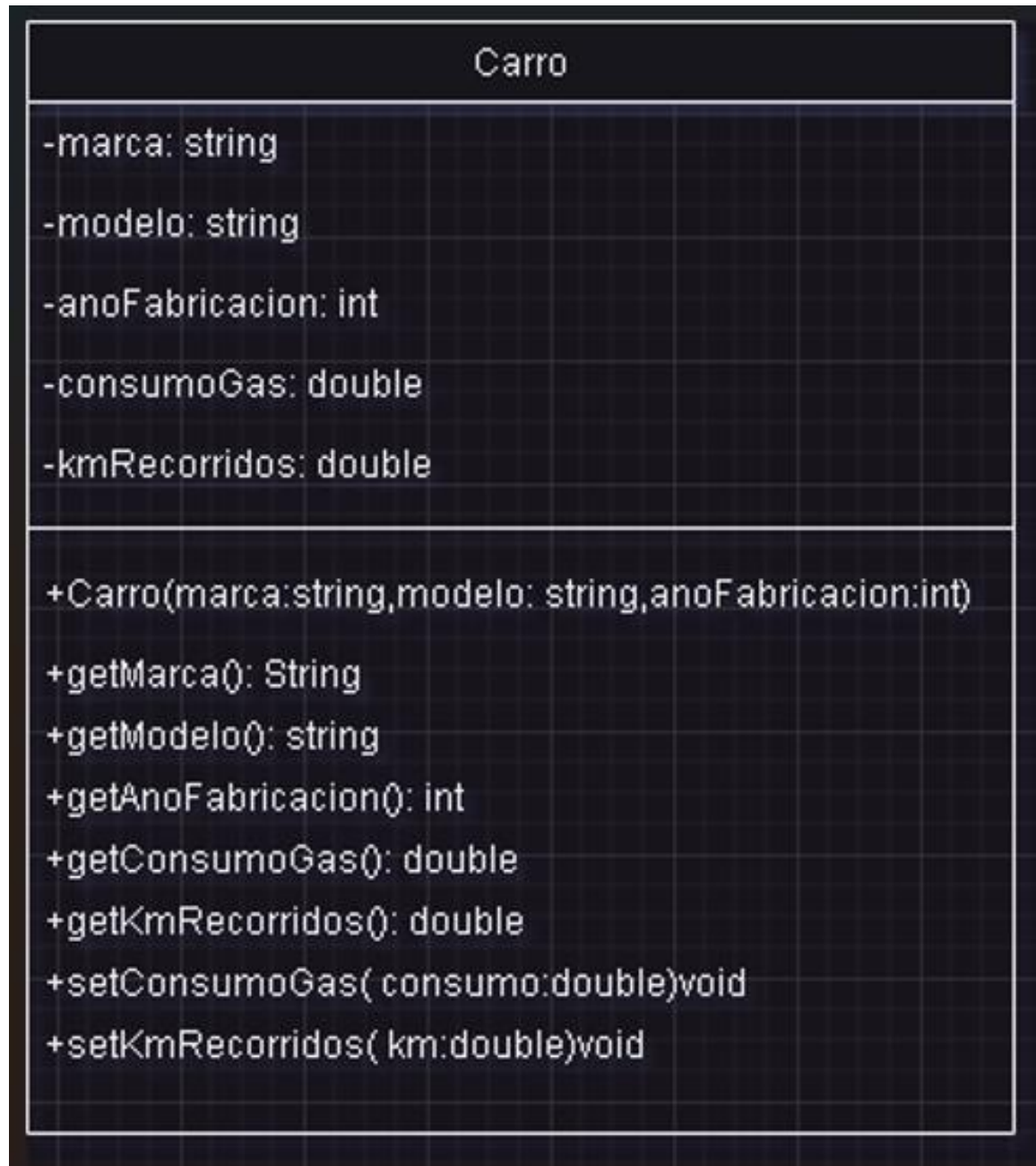


Diagrama UML clasificador del carro



Código de la clase Carro

```
1 package P00;
2
3 /**
4  * La clase Carro representa un vehículo con atributos como marca, modelo, año de fabricación,
5  * consumo de gas y kilómetros recorridos.
6  */
7 public class Carro {
8     private final int anoFabricacion; // Año de fabricación del automóvil (atributo final que no puede ser modificado una vez establecido)
9     private double consumoGas; // Consumo de gas del automóvil en km/l
10    private double kmRecorridos; // Kilómetros recorridos por el automóvil
11    private final String marca; // Marca del automóvil
12    private final String modelo; // Modelo del automóvil
13
14    /**
15     * Constructor de la clase Carro que inicializa los atributos básicos.
16     *
17     * @param marca La marca del automóvil.
18     * @param modelo El modelo del automóvil.
19     * @param anoFabricacion El año de fabricación del automóvil.
20     */
21 }
```

```
21 public Carro(String marca, String modelo, int anoFabricacion) {
22     this.marca = marca;
23     this.modelo = modelo;
24     this.anoFabricacion = anoFabricacion;
25     this.consumoGas = 0.0; // Se inicializa en 0.0 por defecto
26     this.kmRecorridos = 0.0; // Se inicializa en 0.0 por defecto
27 }
28
29 /**
30  * Obtiene la marca del automóvil.
31  *
32  * @return La marca del automóvil.
33  */
34 public String getMarca() {
35     return marca;
36 }
37
38 /**
39  * Obtiene el modelo del automóvil.
40  *
41  * @return El modelo del automóvil.
42  */
43 public String getModelo() {
44     return modelo;
45 }
```

```
45     }
46
47     /**
48      * Obtiene el año de fabricación del automóvil.
49      *
50      * @return El año de fabricación del automóvil.
51      */
52     3 usages
53     public int getAñoFabricacion() {
54         return añoFabricacion;
55     }
56
57     /**
58      * Obtiene el consumo de gas del automóvil.
59      *
60      * @return El consumo de gas del automóvil en km/l.
61      */
62     3 usages
63     public double getConsumoGas() {
64         return consumoGas;
65     }
66
67     /**
68      * Obtiene los kilómetros recorridos por el automóvil.
69      *
70      * @return Los kilómetros recorridos por el automóvil.
```

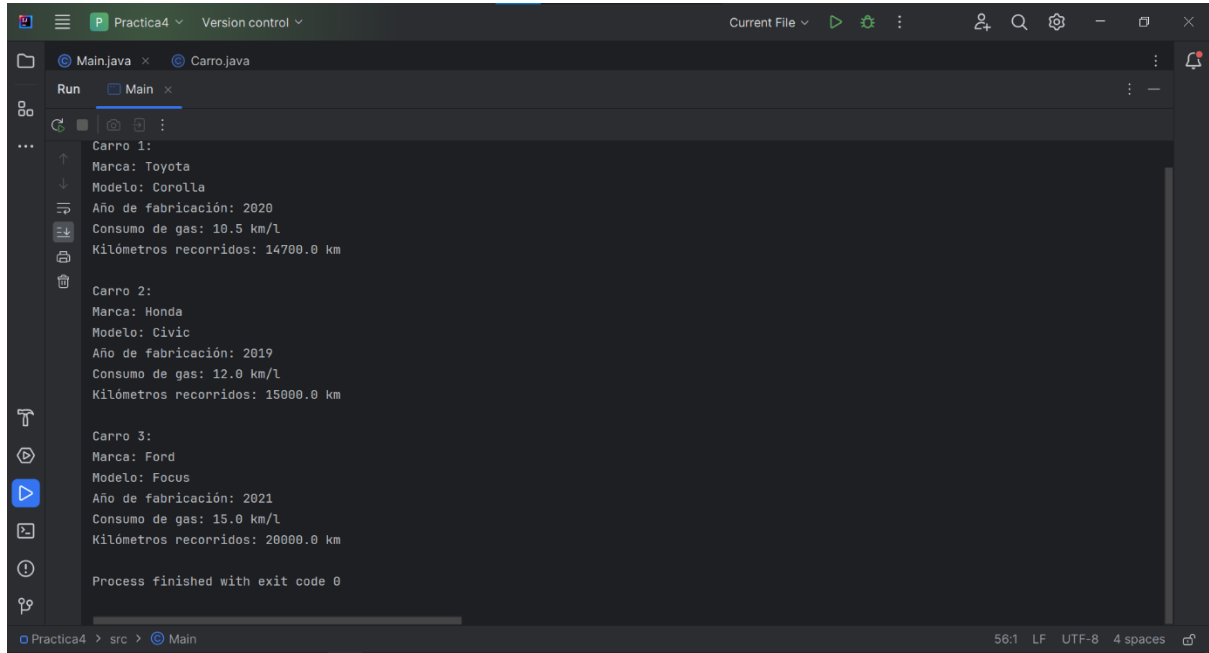
```
69     */
70     3 usages
71     public double getKmRecorridos() {
72         return kmRecorridos;
73     }
74
75     /**
76      * Establece el consumo de gas del automóvil.
77      *
78      * @param consumo El nuevo valor del consumo de gas en km/l.
79      */
80     3 usages
81     public void setConsumoGas(double consumo) {
82         this.consumoGas = consumo;
83     }
84
85     /**
86      * Establece los kilómetros recorridos por el automóvil.
87      *
88      * @param km Los nuevos kilómetros recorridos.
89      */
90     3 usages
91     public void setKmRecorridos(double km) {
92         this.kmRecorridos = km;
93     }
94 }
```

Código de clase principal con tres objetos

```
1  /**
2   * UPTx - Fundamentos de Programación Orientada a Objetos
3   * Practica No. 4      Grupo: 5H
4   * Tema: Clase Carro
5   * Integrantes: Eder López Villarreal, Jazmin Portillo Michicol, Karen Leticia García Vásquez, Eric Jhonathan Anaya Márquez
6   * Objetivo: Realizar una aplicación donde se realice los registros de un coche
7   *              para establecer su mantenimiento, consumo de gas y km recorrido.
8   * Modificación: 18/ene/2024
9   */
10
11  import P00.Carro;
12  public class Main {
13      public static void main(String[] args) {
14
15          // Crea tres objetos Carro
16          Carro carro1 = new Carro("Toyota", "Corolla", 2020);
17          Carro carro2 = new Carro("Honda", "Civic", 2019);
18          Carro carro3 = new Carro("Ford", "Focus", 2021);
19
20          // Actualiza atributos de los carros
21          carro1.setConsumoGas(10.5);
22          carro1.setKmRecorridos(14700.0);
23          carro2.setKmRecorridos(15000.0);
24          carro2.setConsumoGas(12.0); // Establecer el consumo de gas antes de los kilómetros recorridos
25          carro3.setConsumoGas(15.0);
26          carro3.setKmRecorridos(20000.0);
27      }
28  }
```

```
27
28  // Muestra valores en consola
29  System.out.println("Carro 1:");
30  System.out.println("Marca: " + carro1.getMarca());
31  System.out.println("Modelo: " + carro1.getModelo());
32  System.out.println("Año de fabricación: " + carro1.getAñoFabricacion());
33  System.out.println("Consumo de gas: " + carro1.getConsumoGas() + " km/l");
34  System.out.println("Kilómetros recorridos: " + carro1.getKmRecorridos() + " km\n");
35
36  System.out.println("Carro 2:");
37  System.out.println("Marca: " + carro2.getMarca());
38  System.out.println("Modelo: " + carro2.getModelo());
39  System.out.println("Año de fabricación: " + carro2.getAñoFabricacion());
40  System.out.println("Consumo de gas: " + carro2.getConsumoGas() + " km/l");
41  System.out.println("Kilómetros recorridos: " + carro2.getKmRecorridos() + " km\n");
42
43  System.out.println("Carro 3:");
44  System.out.println("Marca: " + carro3.getMarca());
45  System.out.println("Modelo: " + carro3.getModelo());
46  System.out.println("Año de fabricación: " + carro3.getAñoFabricacion());
47  System.out.println("Consumo de gas: " + carro3.getConsumoGas() + " km/l");
48  System.out.println("Kilómetros recorridos: " + carro3.getKmRecorridos() + " km");
49
50  }
```

Funcionalidad



```
Carro 1:
Marca: Toyota
Modelo: Corolla
Año de fabricación: 2020
Consumo de gas: 10.5 km/l
Kilómetros recorridos: 14700.0 km

Carro 2:
Marca: Honda
Modelo: Civic
Año de fabricación: 2019
Consumo de gas: 12.0 km/l
Kilómetros recorridos: 15000.0 km

Carro 3:
Marca: Ford
Modelo: Focus
Año de fabricación: 2021
Consumo de gas: 15.0 km/l
Kilómetros recorridos: 20000.0 km

Process finished with exit code 0
```