



Fundamentos de POO

Ingeniería en
Tecnologías de la
Información

KAREN LETICIA GARCIA VASQUEZ

EDER LÓPEZ VILLARREAL

JAZMIN PORTILLO MICHICOL

ERIC JHONATHAN ANAYA MARQUEZ

5H T/V

DOCENTE

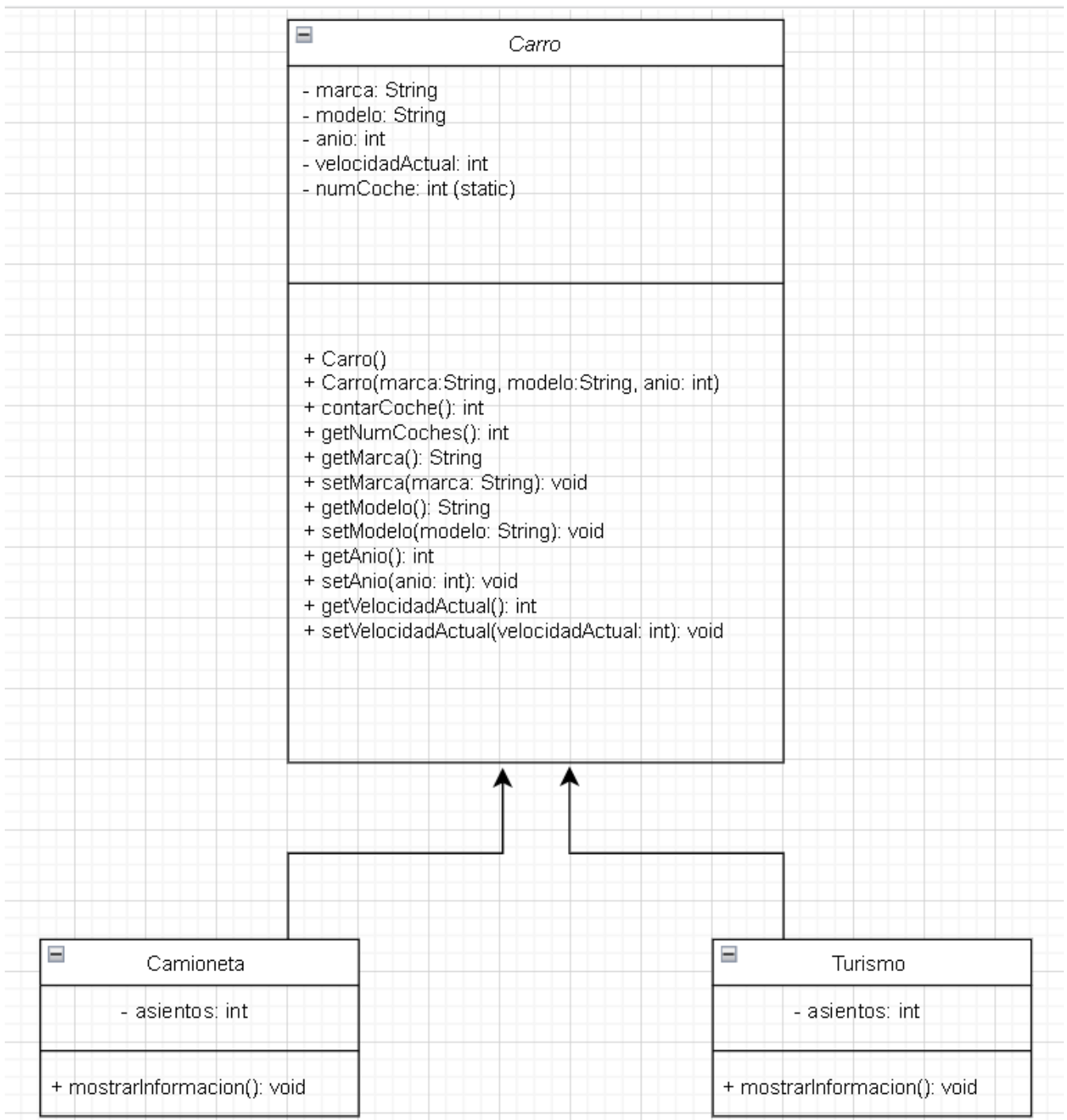
MTRO. SAUL OLAF LOAIZA

Practica 6

Clase Carro con herencia

30/01/2024

Diagrama UML del clasificador de la clase Carro con herencia



Código de clase Carro con herencia

```
1 package P00;
2
3 // La clase Carro representa un vehículo genérico.
4 public class Carro {
5     private String marca; // Marca del carro
6     private String modelo; // Modelo del carro
7     private int anio; // Año de fabricación del carro
8     private int velocidadActual; // Velocidad actual del carro
9     private static int numCoche; // Contador de cantidad de carros creados
10
11     // Constructores
12
13     // Constructor por defecto de la clase Carro.
14     public Carro() {
15         numCoche++; // Incrementa el contador de coches
16     }
17
18     // Constructor de la clase Carro con parámetros.
```

```
19     public Carro(String marca, String modelo, int anio) {
20         this.marca = marca; // Asigna la marca del carro
21         this.modelo = modelo; // Asigna el modelo del carro
22         this.anio = anio; // Asigna el año de fabricación del carro
23         numCoche++; // Incrementa el contador de coches
24     }
25
26     // Métodos
27
28     // Método estático que devuelve la cantidad total de coches creados.
29     public static int contarCoche() { return numCoche; // Devuelve la cantidad total de coches creados }
30
31     // Método que devuelve la cantidad total de coches creados.
32     public int getNumCoches() { return numCoche; // Devuelve la cantidad total de coches creados }
33
34     /**
35      * Método que devuelve la marca del carro.
36      * @return la marca del carro.
37      */
38     public String getMarca() { return marca; // Devuelve la marca del carro }
39
40     // Método que establece la marca del carro.
```

```
Practica6 Version control Current File
Main.java Carro.java x Camioneta.java Turismo.java

47 > public void setMarca(String marca) { this.marca = marca; // Establece la marca del carro }
50
51 // Método que devuelve el modelo del carro.
52 > public String getModelo() { return modelo; // Devuelve el modelo del carro }
55
56 // Método que establece el modelo del carro.
57 > public void setModelo(String modelo) { this.modelo = modelo; // Establece el modelo del carro }
60
61 //Método que devuelve el año de fabricación del carro.
62 > public int getAnio() { return anio; // Devuelve el año de fabricación del carro }
65
66 // Método que establece el año de fabricación del carro.
67 > public void setAnio(int anio) { this.anio = anio; // Establece el año de fabricación del carro }
70
71 // Método que devuelve la velocidad actual del carro.
72 > public int getVelocidadActual() { return velocidadActual; // Devuelve la velocidad actual del carro }
75
76 //Método que establece la velocidad actual del carro.
77
78 3 usages
public void setVelocidadActual(int velocidadActual) {
```

```
Practica6 Version control Current File
Main.java Carro.java x Camioneta.java Turismo.java

79 this.velocidadActual = velocidadActual; // Establece la velocidad actual del carro
80 }
81 }
```

Código de clase Camioneta hija

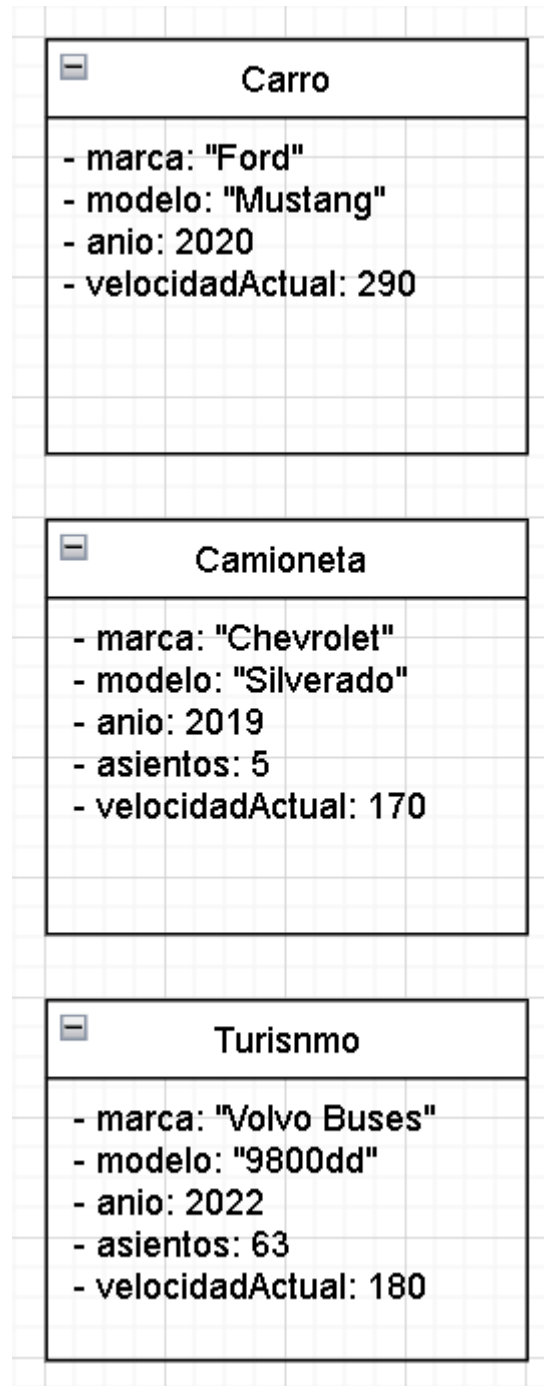
```
1 package P00;
2
3 // La clase Camioneta representa un tipo específico de vehículo de carga.
3 usages
4 public class Camioneta extends Carro {
5     2 usages
6     private final int asientos; // Número de asientos de la camioneta
7
8     // Constructor de la clase Camioneta.
9     1 usage
10    public Camioneta(String marca, String modelo, int anio, int asientos) {
11        super(marca, modelo, anio); // Llama al constructor de la clase base (Carro)
12        this.asientos = asientos; // Asigna el número de asientos de la camioneta
13    }
14
15    // Método que muestra la información de la camioneta.
16    1 usage
17    public void mostrarInformacion() {
18        System.out.println("Marca: " + getMarca()); // Muestra la marca de la camioneta
19        System.out.println("Modelo: " + getModelo()); // Muestra el modelo de la camioneta
20        System.out.println("Año: " + getAnio()); // Muestra el año de fabricación de la camioneta
21        System.out.println("Número de asientos: " + asientos); // Muestra el número de asientos de la camioneta
22        System.out.println("Velocidad actual: " + getVelocidadActual()); // Muestra la velocidad actual de la camioneta
23    }
24 }
```

Código de clase Turismo hija

```

1 package P00;
2
3 // La clase Turismo representa un tipo específico de vehículo de turismo.
4 public class Turismo extends Carro {
5     private final int asientos; // Número de asientos del turismo
6
7     // Constructor de la clase Turismo.
8     public Turismo(String marca, String modelo, int anio, int asientos) {
9         super(marca, modelo, anio); // Llama al constructor de la clase base (Carro)
10        this.asientos = asientos; // Asigna el número de asientos del turismo
11    }
12
13    // Método que muestra la información del turismo.
14    public void mostrarInformacion() {
15        System.out.println("Marca: " + getMarca()); // Muestra la marca del turismo
16        System.out.println("Modelo: " + getModelo()); // Muestra el modelo del turismo
17        System.out.println("Año: " + getAnio()); // Muestra el año de fabricación del turismo
18        System.out.println("Número de asientos: " + asientos); // Muestra el número de asientos del turismo
19        System.out.println("Velocidad actual: " + getVelocidadActual()); // Muestra la velocidad actual del turismo
20    }
21 }
```

Diagrama UML de los objetos para verificar la prueba



Clase principal Main

```

1  import P00.Carro;
2  import P00.Camioneta;
3  import P00.Turismo;
4
5  public class Main {
6      public static void main(String[] args) {
7          // Crear un objeto de tipo Carro
8          Carro carro = new Carro( marca: "Ford", modelo: "Mustang", anio: 2020);
9          carro.setVelocidadActual(290); // Definir la velocidad actual del carro
10
11         // Crear un objeto de tipo Camioneta
12         Camioneta camioneta = new Camioneta( marca: "Chevrolet", modelo: "Silverado", anio: 2019, asientos: 5);
13         camioneta.setVelocidadActual(170); // Definir la velocidad actual de la camioneta
14
15         // Crear un objeto de tipo Turismo (Autobús)
16         Turismo autobus = new Turismo( marca: "Volvo Buses", modelo: "9800dd", anio: 2022, asientos: 63);
17         autobus.setVelocidadActual(180); // Definir la velocidad actual del autobús
18
19         // Mostrar información de los vehículos
20         System.out.println("\nInformación del carro:");
21         System.out.println("Marca: " + carro.getMarca());
22         System.out.println("Modelo: " + carro.getModelo());
23         System.out.println("Año: " + carro.getAnio());
24         System.out.println("Velocidad actual: " + carro.getVelocidadActual());
25
26         System.out.println("\nInformación de la camioneta:");

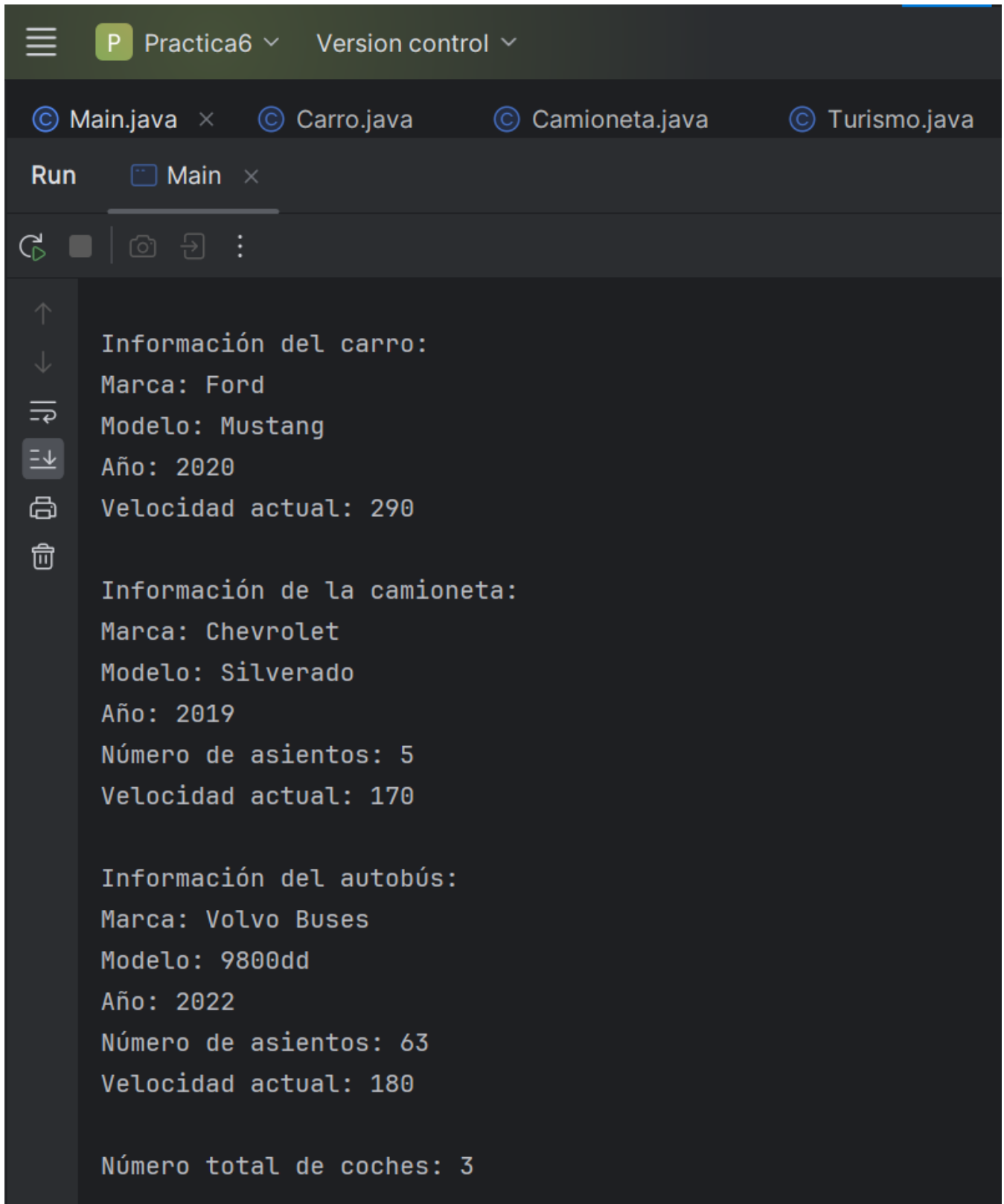
```

```

27         camioneta.mostrarInformacion();
28
29         System.out.println("\nInformación del autobús:");
30         autobus.mostrarInformacion();
31
32         // Contar la cantidad de coches creados
33         System.out.println("\nNúmero total de coches: " + Carro.contarCoche());
34     }
35 }

```


Prueba



The screenshot shows an IDE interface with a dark theme. At the top, there's a menu bar with a hamburger icon, a tab labeled 'P Practica6', and a dropdown menu 'Version control'. Below the menu bar, there are four open files: 'Main.java', 'Carro.java', 'Camioneta.java', and 'Turismo.java'. A 'Run' button and a 'Main' tab are visible. The main editor area displays the output of a Java program. On the left side of the editor, there are icons for navigating through the output: up, down, search, list, print, and delete. The output text is as follows:

```
Información del carro:  
Marca: Ford  
Modelo: Mustang  
Año: 2020  
Velocidad actual: 290  
  
Información de la camioneta:  
Marca: Chevrolet  
Modelo: Silverado  
Año: 2019  
Número de asientos: 5  
Velocidad actual: 170  
  
Información del autobús:  
Marca: Volvo Buses  
Modelo: 9800dd  
Año: 2022  
Número de asientos: 63  
Velocidad actual: 180  
  
Número total de coches: 3
```