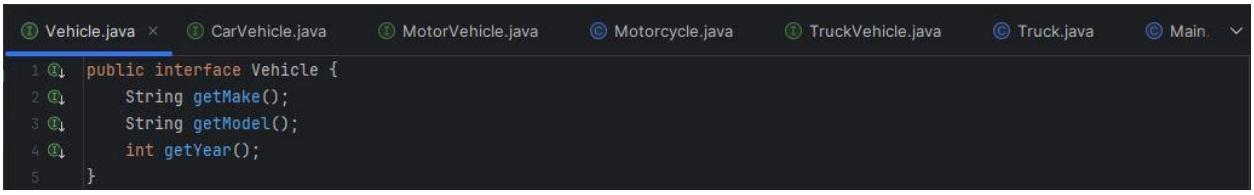
CS1102-01 | PROGRAMMING ASSIGNMENT UNIT 6 BONAVENTURE OKOYE

VEHICLE MANAGEMENT SOFTWARE APPLICATION

This project is a command-line based vehicle management system that enables users to manage a variety of vehicles, including cars, motorcycles, and trucks. The system allows for the creation of different vehicle types, setting their specific attributes, and displaying their details. It consists of four interfaces – Vehicle, CarVehicle, MotorVehicle, TruckVehicle – and three main classes: Car, Motorcycle and Truck.

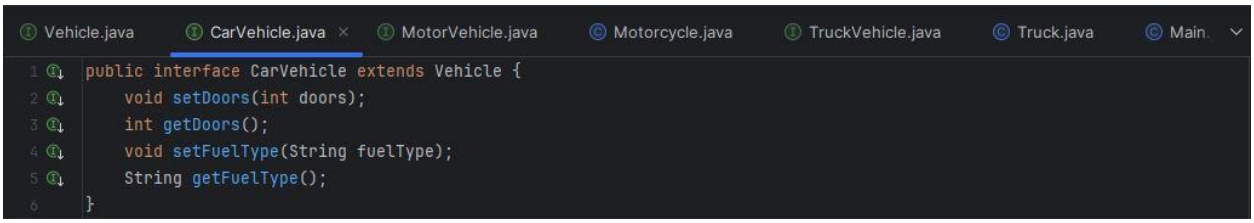
Vehicle Interface:

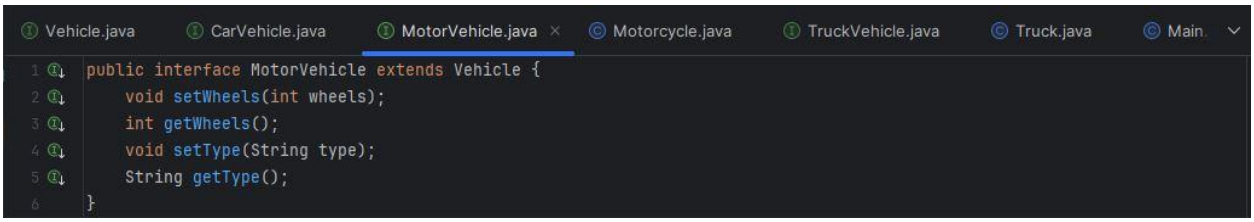


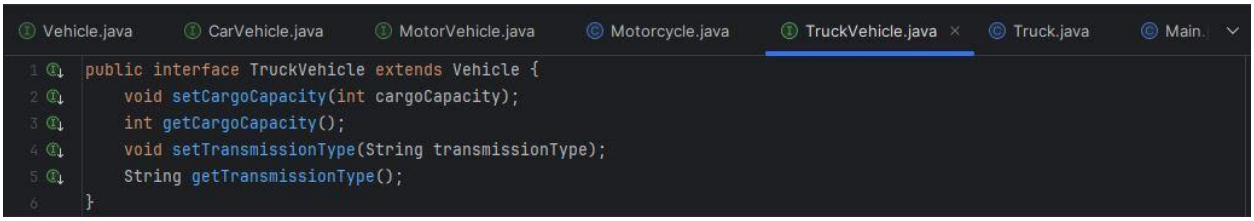
The Vehicle interface defines the common methods that all vehicle types must implement, such as retrieving the make, model, and year of manufacture.

CarVehicle, MotorVehicle and TruckVehicle Interfaces:

The CarVehicle, MotorVehicle, and TruckVehicle interfaces extend the Vehicle interface and include methods specific to cars, motorcycles, and trucks, respectively. These methods allow for setting and retrieving attributes such as the number of doors, fuel type, number of wheels, type of motorcycle, cargo capacity, and transmission type.







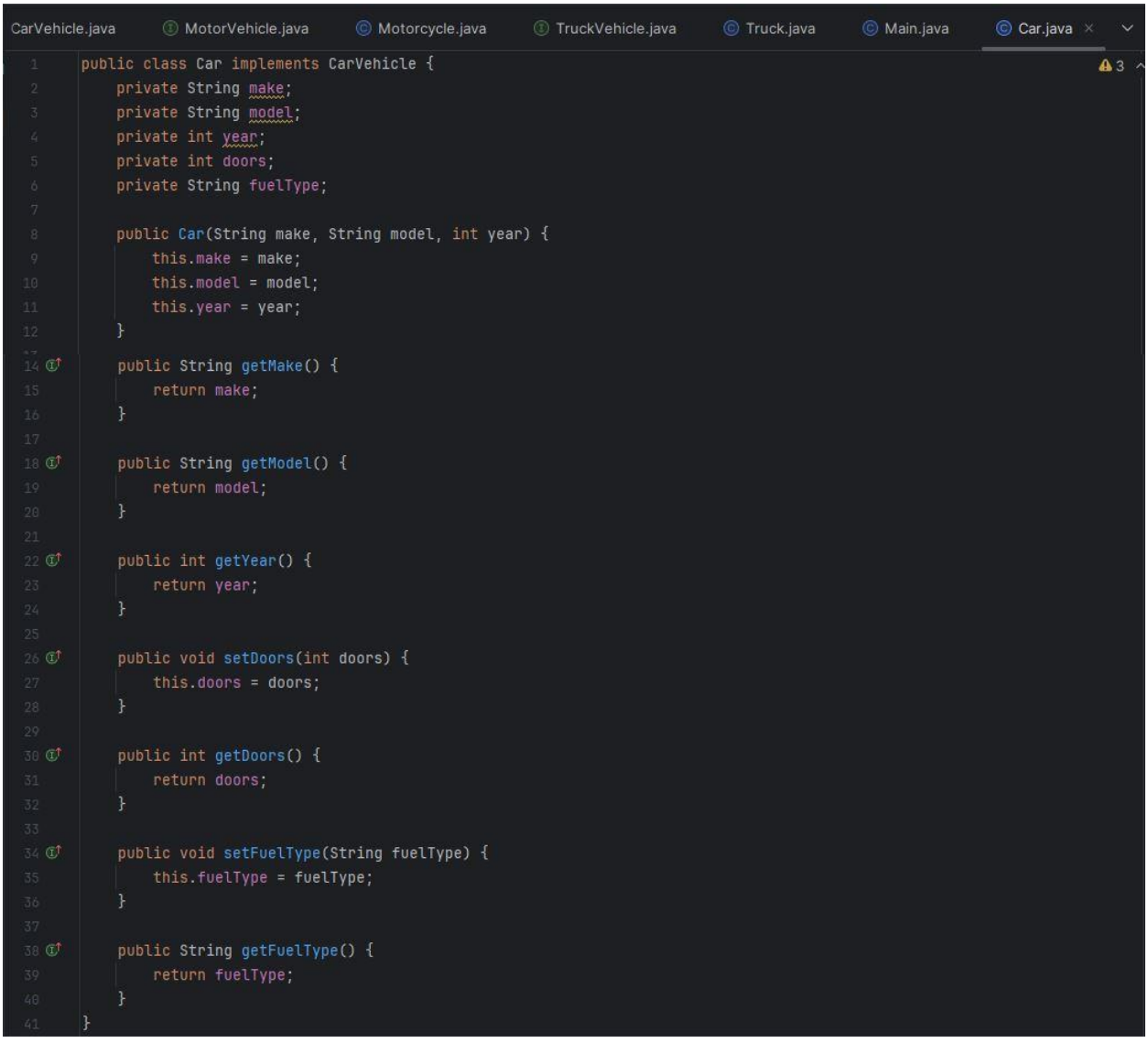
Car, Motorcycle and Truck Classes:

The Car, Motorcycle, and Truck classes implement their respective interfaces, providing the necessary fields and methods to represent and manage each vehicle type. The classes include private fields for the attributes specific to each vehicle type and provide getter and setter methods for these fields.

Car Class:

The Car class implements the CarVehicle interface, which extends the Vehicle interface. This class represents a car and includes private fields for make, model, year, doors, and fuelType. It provides the following methods:

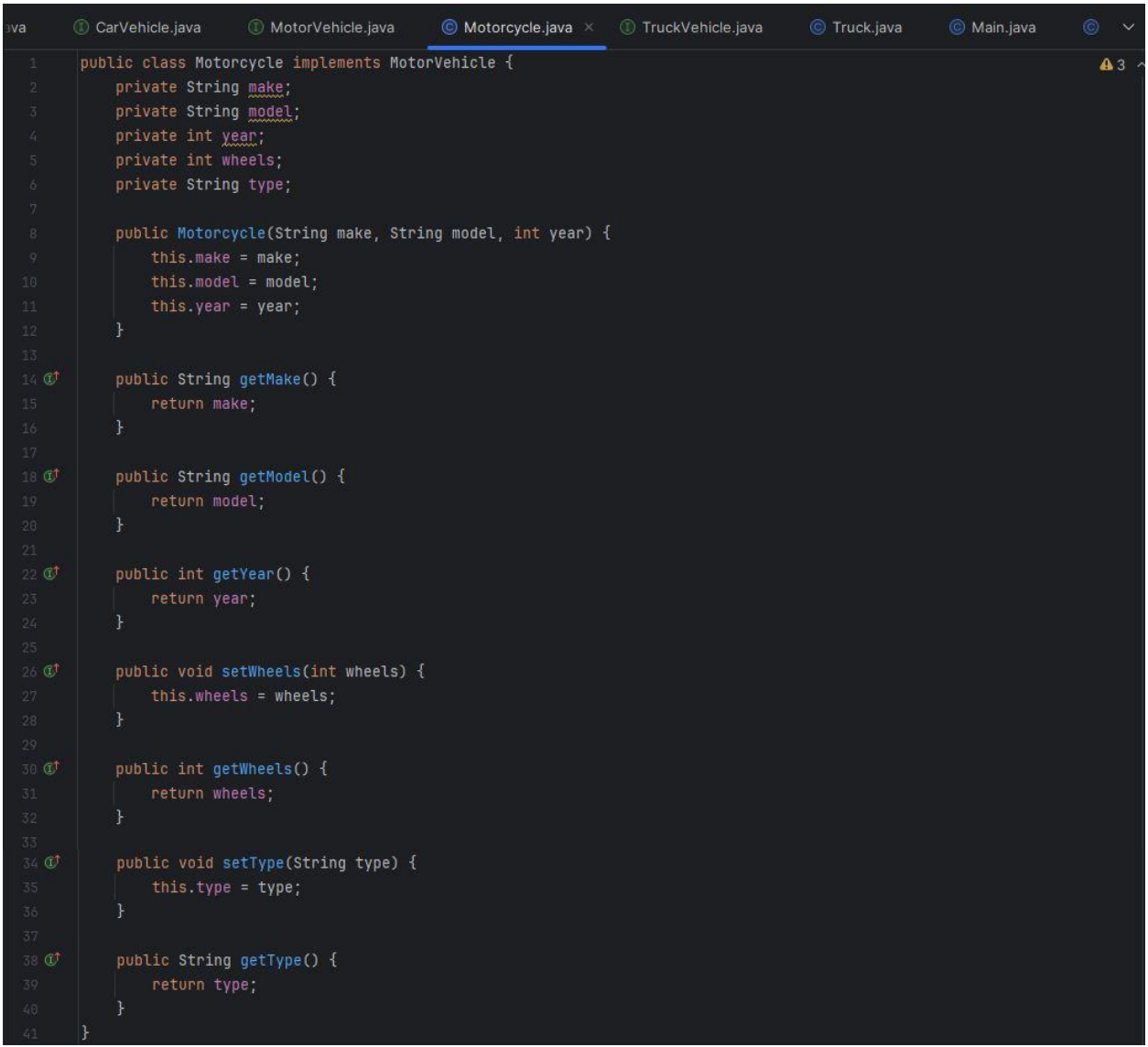
- getMake(), getModel(), and getYear(): These methods retrieve the make, model, and year of manufacture of the car, respectively.
- setDoors(int doors) and getDoors(): These methods set and retrieve the number of doors of the car.
- setFuelType(String fuelType) and getFuelType(): These methods set and retrieve the fuel type (petrol, diesel, or electric) of the car.



Motorcycle Class:

The Motorcycle class implements the MotorVehicle interface, which extends the Vehicle interface. This class represents a motorcycle and includes private fields for make, model, year, wheels, and type. It provides the following methods:

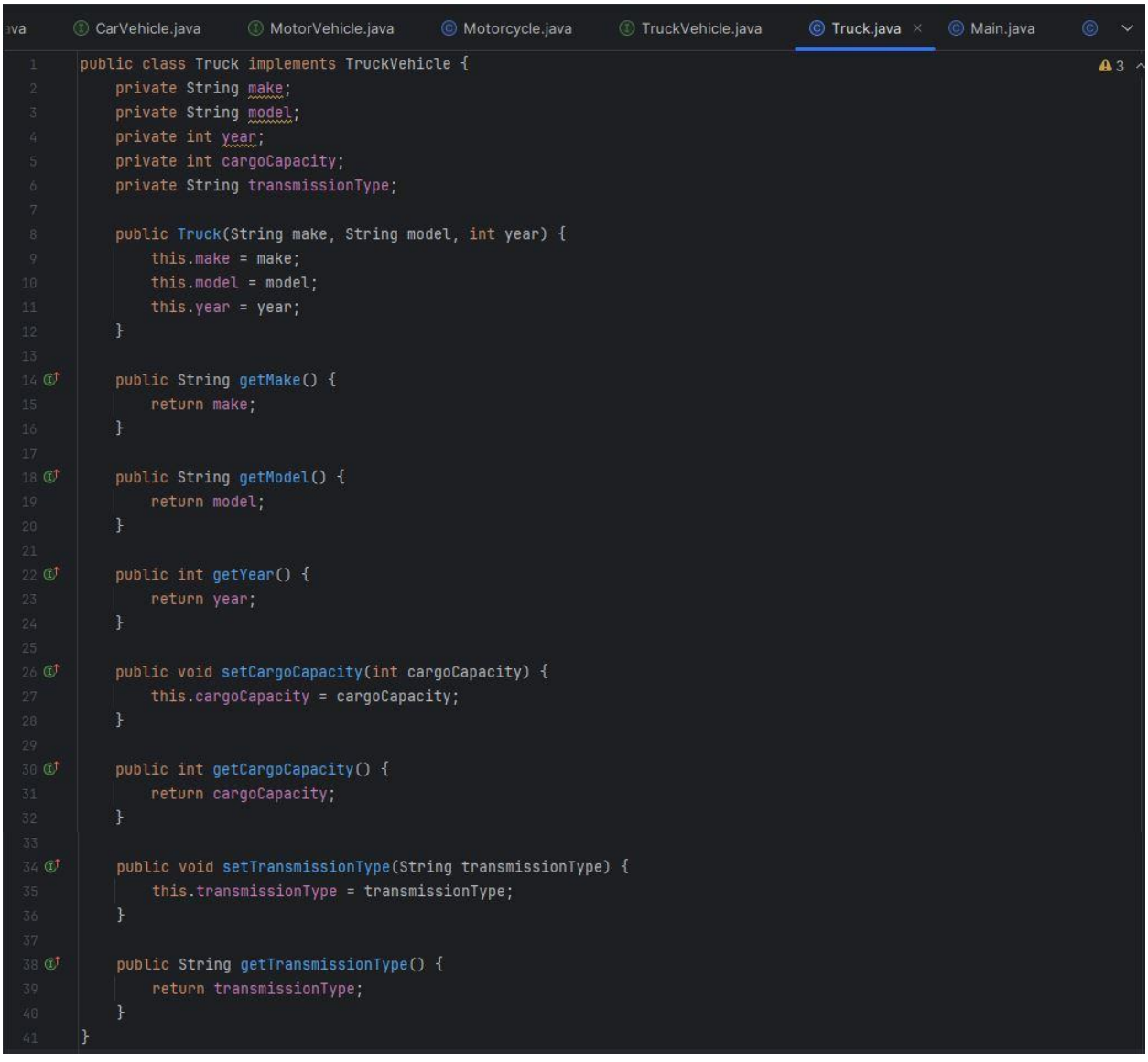
- getMake(), getModel(), and getYear(): These methods retrieve the make, model, and year of manufacture of the motorcycle, respectively.
- setWheels(int wheels) and getWheels(): These methods set and retrieve the number of wheels of the motorcycle.
- setType(String type) and getType(): These methods set and retrieve the type of motorcycle (sport, cruiser, or off-road).



Truck Class:

The Truck class implements the TruckVehicle interface, which extends the Vehicle interface. This class represents a truck and includes private fields for make, model, year, cargoCapacity, and transmissionType. It provides the following methods:

- getMake(), getModel(), and getYear(): These methods retrieve the make, model, and year of manufacture of the truck, respectively.
- setCargoCapacity(int cargoCapacity) and getCargoCapacity(): These methods set and retrieve the cargo capacity (in tons) of the truck.
- setTransmissionType(String transmissionType) and getTransmissionType(): These methods set and retrieve the transmission type (manual or automatic) of the truck.



Main Class:

CarVehicle.java | MotorVehicle.java | Motorcycle.java | TruckVehicle.java | Truck.java | Main.java | Car.java

```java
public class Main {
    public static void main(String[] args) {
        Car car = new Car("Toyota", "Camry", 2021);
        car.setDoors(4);
        car.setFuelType("Petrol");

        Motorcycle motorcycle = new Motorcycle("Yamaha", "YZF-R1", 2020);
        motorcycle.setWheels(2);
        motorcycle.setType("Sport");

        Truck truck = new Truck("Ford", "F-150", 2019);
        truck.setCargoCapacity(3);
        truck.setTransmissionType("Automatic");

        System.out.println("Car details:");
        System.out.println("Make: " + car.getMake());
        System.out.println("Model: " + car.getModel());
        System.out.println("Year: " + car.getYear());
        System.out.println("Doors: " + car.getDoors());
        System.out.println("Fuel type: " + car.getFuelType());

        System.out.println("\nMotorcycle details:");
        System.out.println("Make: " + motorcycle.getMake());
        System.out.println("Model: " + motorcycle.getModel());
        System.out.println("Year: " + motorcycle.getYear());
        System.out.println("Wheels: " + motorcycle.getWheels());
        System.out.println("Type: " + motorcycle.getType());

        System.out.println("\nTruck details:");
        System.out.println("Make: " + truck.getMake());
        System.out.println("Model: " + truck.getModel());
        System.out.println("Year: " + truck.getYear());
        System.out.println("Cargo capacity: " + truck.getCargoCapacity() + " tons");
        System.out.println("Transmission type: " + truck.getTransmissionType());
    }
}
```

Output:

Run | Main

```
"C:\Program Files\Java\jdk-21.0.2\bin\java.exe" "-javaagent:C:\Program Files\Intellij\ideaIC-2023.3.3.win\lib\idea_rt.jar=57562:C:\Program Files\Intellij\ideaIC-2023.3.3.win\bin"
Car details:
Make: Toyota
Model: Camry
Year: 2021
Doors: 4
Fuel type: Petrol

Motorcycle details:
Make: Yamaha
Model: YZF-R1
Year: 2020
Wheels: 2
Type: Sport

Truck details:
Make: Ford
Model: F-150
Year: 2019
Cargo capacity: 3 tons
Transmission type: Automatic

Process finished with exit code 0
```