Programming Assignment Unit 6

Godfrey Ouma

University of the People

CS 1102: Programming 1

Ruth Alabi

December 28, 2023

```java
package textio;
import java.util.Scanner;

//Vehicle interface specifying common methods
interface Vehicle {
 String getMake();
 String getModel();
 int getYear();
 void setMake(String make);
 void setModel(String model);
 void setYear(int year);
}

//CarVehicle interface extending Vehicle with car-specific methods
interface CarVehicle extends Vehicle {
 int getNumDoors();
 String getFuelType();
 void setNumDoors(int numDoors);
 void setFuelType(String fuelType);
}

//MotorVehicle interface extending Vehicle with motorcycle-specific methods
interface MotorVehicle extends Vehicle {
 int getNumWheels();
 String getMotorcycleType();
 void setNumWheels(int numWheels);
 void setMotorcycleType(String motorcycleType);
}

//TruckVehicle interface extending Vehicle with truck-specific methods
interface TruckVehicle extends Vehicle {
 double getCargoCapacity();
 String getTransmissionType();
 void setCargoCapacity(double cargoCapacity);
 void setTransmissionType(String transmissionType);
}

//Car class implementing CarVehicle interface
class Car implements CarVehicle {
 private String make;
 private String model;
 private int year;
 private int numDoors;
 private String fuelType;

 // Implementing methods from Vehicle interface
 @Override
 public String getMake() {
     return make;
 }

 @Override
 public String getModel() {
     return model;
 }

 @Override
 public int getYear() {
     return year;
```

```
 }

 @Override
 public void setMake(String make) {
     this.make = make;
 }

 @Override
 public void setModel(String model) {
     this.model = model;
 }

 @Override
 public void setYear(int year) {
     this.year = year;
 }

 // Implementing methods from CarVehicle interface
 @Override
 public int getNumDoors() {
     return numDoors;
 }

 @Override
 public String getFuelType() {
     return fuelType;
 }

 @Override
 public void setNumDoors(int numDoors) {
     this.numDoors = numDoors;
 }

 @Override
 public void setFuelType(String fuelType) {
     this.fuelType = fuelType;
 }
}

abstract //Motorcycle class implementing MotorVehicle interface
class Motorcycle implements MotorVehicle {
 // Implement the Motorcycle class according to the MotorVehicle interface
 // ...

 // Implement required methods from Vehicle and MotorVehicle interfaces
 // ...
}

//Truck class implementing TruckVehicle interface
abstract class Truck implements TruckVehicle {
 // Implement the Truck class according to the TruckVehicle interface
 // ...

 // Implement required methods from Vehicle and TruckVehicle interfaces
 // ...
}
public class VehicleInformationSystem {

    public static void main(String[] args) {
```

```
        Scanner scanner = new Scanner(System.in);

        // Implementing a main program that allows users to interactively create, input,
and display details of various vehicle objects.
        Car car = new Car();
        System.out.println("Enter car details:");
        System.out.print("Make: ");
        car.setMake(scanner.nextLine());
        System.out.print("Model: ");
        car.setModel(scanner.nextLine());
        System.out.print("Year: ");
        car.setYear(scanner.nextInt());
        System.out.print("Number of doors: ");
        car.setNumDoors(scanner.nextInt());
        scanner.nextLine(); // Clear buffer
        System.out.print("Fuel type: ");
        car.setFuelType(scanner.nextLine());

        // Display car details
        System.out.println("\nCar Details:");
        System.out.println("Make: " + car.getMake());
        System.out.println("Model: " + car.getModel());
        System.out.println("Year: " + car.getYear());
        System.out.println("Number of doors: " + car.getNumDoors());
        System.out.println("Fuel type: " + car.getFuelType());

        // Close scanner
        scanner.close();

    }

}
```

Console

```
Enter car details:
Make: Mercedes-Benz
Model: GLS 600
Year: 2022
Number of doors: 5
Fuel type: Diesel

Car Details:
Make: Mercedes-Benz
Model: GLS 600
Year: 2022
Number of doors: 5
Fuel type: Diesel
```

Documentation

Interfaces: The provided code includes interfaces like Vehicle, CarVehicle, MotorVehicle, and TruckVehicle. These interfaces define methods for retrieving and setting specific vehicle details, ensuring a consistent structure across different vehicle types.

Classes: The classes Car, Motorcycle, and Truck implement the respective interfaces and contain member variables along with methods to handle vehicle attributes.

Main Program: The main method in the VehicleInformationSystem class facilitates interactive creation, input, and display of car details. It uses a Scanner object to receive user inputs and sets/display car attributes accordingly.