

Nine.java

```
1  /*
2  Write a Java program to create a vehicle class hierarchy. The base class should be
3  Vehicle, with subclasses Truck, Car and Motorcycle. Each subclass should have
4  properties such as make, model, year, and fuel type. Implement methods for calculating
5  fuel efficiency, distance traveled, and maximum speed.
6  Expected Output:
7  === Truck Info ===
8  Make: Scania, Model: R-Series, Year: 2021, Fuel: Diesel
9  Fuel Efficiency: 6.0 km/l
10 Distance for 40L: 240.0 km
11 Max Speed: 110.0 km/h
12 === Car Info ===
13 Make: Honda, Model: Civic, Year: 2023, Fuel: Petrol
14 Fuel Efficiency: 18.0 km/l
15 Distance for 40L: 720.0 km
16 Max Speed: 190.0 km/h
17 === Motorcycle Info ===
18 Make: Suzuki, Model: Gixxer, Year: 2022, Fuel: Petrol
19 Fuel Efficiency: 40.0 km/l
20 Distance for 10L: 400.0 km
21 Max Speed: 150.0 km/h
22 */
23
24 import java.util.Scanner;
25
26 class Vehicle{
27     String make;
28     String model;
29     int year;
30     String fuelType;
31     double fuelEfficiency;
32     double maxSpeed;
33
34     Vehicle(String make, String model, int year, String fuelType, double fuelEfficiency,
double maxSpeed){
35         this.make = make;
36         this.model = model;
37         this.year = year;
38         this.fuelType = fuelType;
39         this.fuelEfficiency = fuelEfficiency;
40         this.maxSpeed = maxSpeed;
41     }
42     double getFuelEfficiency(){
43         return fuelEfficiency;
44     }
45     double calculateDistance(double fuel){
46         return fuel * fuelEfficiency;
47     }
48     double getMaxSpeed(){
49         return maxSpeed;
50     }
51     void display9(){
```

```

52         System.out.println("Make: " + make + ", Model: " + model + ", Year: " + year + ",
Fuel: " + fuelType);
53     }
54 }
55 class Truck extends Vehicle{
56     Truck(String make, String model, int year, String fuelType, double fuelEfficiency,
double maxSpeed){
57         super(make, model, year, fuelType, fuelEfficiency, maxSpeed);
58     }
59 }
60 class Car extends Vehicle{
61     Car(String make, String model, int year, String fuelType, double fuelEfficiency,
double maxSpeed){
62         super(make, model, year, fuelType, fuelEfficiency, maxSpeed);
63     }
64 }
65 class Motorcycle extends Vehicle{
66     Motorcycle(String make, String model, int year, String fuelType, double
fuelEfficiency, double maxSpeed){
67         super(make, model, year, fuelType, fuelEfficiency, maxSpeed);
68     }
69 }
70
71 public class Nine {
72     public static void main(String[] args) {
73         Scanner scan = new Scanner(System.in);
74
75         // Truck input
76         System.out.println("Make: "); String tMake = scan.nextLine();
77         System.out.println("Model: "); String tModel = scan.nextLine();
78         System.out.println("Year: "); int tYear = scan.nextInt();
79         System.out.println("Fuel Type: "); String tFuel = scan.nextLine();
80         System.out.println("Fuel Efficiency: "); double tEfficiency = scan.nextDouble();
81         System.out.println("Max Speed: "); double tMaxSpeed = scan.nextDouble();
82         scan.nextLine();    // clear buffer
83
84         Truck trck = new Truck(tMake, tModel, tYear, tFuel, tEfficiency, tMaxSpeed);
85
86
87         // Car input
88         System.out.println("Make: "); String cMake = scan.nextLine();
89         System.out.println("Model: "); String cModel = scan.nextLine();
90         System.out.println("Year: "); int cYear = scan.nextInt();
91         System.out.println("Fuel Type: "); String cFuel = scan.nextLine();
92         System.out.println("Fuel Efficiency: "); double cEfficiency = scan.nextDouble();
93         System.out.println("Max Speed: "); double cMaxSpeed = scan.nextDouble();
94         scan.nextLine();    // clear buffer
95
96         Car car = new Car(cMake, cModel, cYear, cFuel, cEfficiency, cMaxSpeed);
97
98
99         // Motorcycle input
100        System.out.println("Make: "); String motoCMake = scan.nextLine();
101        System.out.println("Model: "); String motoCModel = scan.nextLine();

```

```

102     System.out.println("Year: "); int motoCYear = scan.nextInt();
103     System.out.println("Fuel Type: "); String motoCFuel = scan.nextLine();
104     System.out.println("Fuel Efficiency: "); double motoCEfficiency =
scan.nextDouble();
105     System.out.println("Max Speed: "); double motoCMaxSpeed = scan.nextDouble();
106     scan.nextLine();    // clear buffer
107
108     Motorcycle motCycl = new Motorcycle(motoCMake, motoCModel, motoCYear, motoCFuel,
motoCEfficiency, motoCMaxSpeed);
109
110     // Display Truck Info
111     System.out.println("\n=== Truck Info ===");
112     trck.display9();
113     System.out.println("Fuel Efficiency: " + trck.getFuelEfficiency() + " km/l");
114     System.out.println("Distance for 40L: " + trck.calculateDistance(40) + " km");
115     System.out.println("Max Speed: " + trck.getMaxSpeed() + " km/h");
116
117     // Display Car Info
118     System.out.println("\n=== Car Info ===");
119     car.display9();
120     System.out.println("Fuel Efficiency: " + car.getFuelEfficiency() + " km/l");
121     System.out.println("Distance for 40L: " + car.calculateDistance(40) + " km");
122     System.out.println("Max Speed: " + car.getMaxSpeed() + " km/h");
123
124     // Display Motorcycle Info
125     System.out.println("\n=== Motorcycle Info ===");
126     motCycl.display9();
127     System.out.println("Fuel Efficiency: " + motCycl.getFuelEfficiency() + " km/l");
128     System.out.println("Distance for 10L: " + motCycl.calculateDistance(10) + " km");
129     System.out.println("Max Speed: " + motCycl.getMaxSpeed() + " km/h");
130
131     scan.close();
132 }
133 }

```