

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,
35     double maxSpeed){
36         this.make = make;
37         this.model = model;
38         this.year = year;
39         this.fuelType = fuelType;
40         this.fuelEfficiency = fuelEfficiency;
41         this.maxSpeed = maxSpeed;
42     }
43     double getFuelEfficiency(){
44         return fuelEfficiency;
45     }
46     double calculateDistance(double fuel){
47         return fuel * fuelEfficiency;
48     }
49     double getMaxSpeed(){
50         return maxSpeed;
51     }
52     void display9(){
53 }
```

```
52     System.out.println("Make: " + make + ", Model: " + model + ", Year: " + year + ",  
53     Fuel: " + fuelType);  
54 }  
55 class Truck extends Vehicle{  
56     Truck(String make, String model, int year, String fuelType, double fuelEfficiency,  
57     double maxSpeed){  
58         super(make, model, year, fuelType, fuelEfficiency, maxSpeed);  
59     }  
60 }  
61 class Car extends Vehicle{  
62     Car(String make, String model, int year, String fuelType, double fuelEfficiency,  
63     double maxSpeed){  
64         super(make, model, year, fuelType, fuelEfficiency, maxSpeed);  
65 }  
66 class Motorcycle extends Vehicle{  
67     Motorcycle(String make, String model, int year, String fuelType, double  
fuelEfficiency, double maxSpeed){  
68         super(make, model, year, fuelType, fuelEfficiency, maxSpeed);  
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 =
105         scan.nextDouble();
106     System.out.println("Max Speed: "); double motoCMaxSpeed = scan.nextDouble();
107     scan.nextLine(); // clear buffer
108
109     Motorcycle motCycl = new Motorcycle(motoCMake, motoCModel, motoCYear, motoCFuel,
110     motoCEfficiency, motoCMaxSpeed);
111
112     // Display Truck Info
113     System.out.println("\n==== Truck Info ===");
114     trck.display9();
115     System.out.println("Fuel Efficiency: " + trck.getFuelEfficiency() + " km/l");
116     System.out.println("Distance for 40L: " + trck.calculateDistance(40) + " km");
117     System.out.println("Max Speed: " + trck.getMaxSpeed() + " km/h");
118
119     // Display Car Info
120     System.out.println("\n==== Car Info ===");
121     car.display9();
122     System.out.println("Fuel Efficiency: " + car.getFuelEfficiency() + " km/l");
123     System.out.println("Distance for 40L: " + car.calculateDistance(40) + " km");
124     System.out.println("Max Speed: " + car.getMaxSpeed() + " km/h");
125
126     // Display Motorcycle Info
127     System.out.println("\n==== Motorcycle Info ===");
128     motCycl.display9();
129     System.out.println("Fuel Efficiency: " + motCycl.getFuelEfficiency() + " km/l");
130     System.out.println("Distance for 10L: " + motCycl.calculateDistance(10) + " km");
131     System.out.println("Max Speed: " + motCycl.getMaxSpeed() + " km/h");
132
133     scan.close();
134 }
```