comments.java

```java
import java.util.ArrayList;//used to create dynamic arrays to store the list of available vehicles.
import java.util.List;//Used for creating a List object that can hold multiple types of objects (Vehicle).
import java.util.Scanner;//taking input from the user via the command line.
// Interface defining the structure of a rental system
interface RentalStructure {
    void displayVehicles(); // Method to display available vehicles
    void bookVehicle(); // Method to book a vehicle
    void addVehicle(); // Method to add a new vehicle
}
// Abstract class representing a vehicle
abstract class Vehicle {
    private String company; // Company of the vehicle
    private String model; // Model of the vehicle
    private int year; // Year of manufacture
    private int costPHour; // Cost per hour for renting
    private boolean available; // Availability status of the vehicle

    // Constructor to initialize vehicle details
    public Vehicle(String company, String model, int year, int costPHour) {
        this.company = company;
        this.model = model;
        this.year = year;
        this.costPHour = costPHour;
        this.available = true; // Initialize availability to true
    }
    // Getter methods for vehicle details
    public String getCompany() {
        return company;
    }
    public String getModel() {
        return model;
    }
    public int getYear() {
        return year;
    }
    public int getCostPHour() {
        return costPHour;
    }
    public boolean isAvailable() {
```

```
        return available;
    }
    // Setter method to update availability status
    public void setAvailable(boolean available) {
        this.available = available;
    }
    // Abstract method to display details of the vehicle (to be implemented by subclasses)
    public abstract void displayDetails();
}
// Class representing a two-wheeler vehicle
class TwoWheeler extends Vehicle {
    // Constructor to initialize two-wheeler details
    public TwoWheeler(String company, String model, int year, int costPHour) {
        super(company, model, year, costPHour);
    }
    // Method to display details of the two-wheeler
    public void displayDetails() {
        System.out.println("Two Wheeler: " + getCompany() + " " + getModel() + " (" + getYear() + ")");
        System.out.println("Cost per Hour: Rs." + getCostPHour());
    }
}
// Class representing a four-wheeler vehicle
class FourWheeler extends Vehicle {
    // Constructor to initialize four-wheeler details
    public FourWheeler(String company, String model, int year, int costPHour) {
        super(company, model, year, costPHour);
    }
    // Method to display details of the four-wheeler
    public void displayDetails() {
        System.out.println("Four Wheeler: " + getCompany() + " " + getModel() + " (" + getYear() + ")");
        System.out.println("Cost per Hour: Rs." + getCostPHour());
    }
}
// Class representing the rental system
class RentalSystem implements RentalStructure {
    private List<Vehicle> availableVehicles; // List of available vehicles

    // Constructor to initialize the rental system with some default vehicles
    public RentalSystem() {
        availableVehicles = new ArrayList<>();
        availableVehicles.add(new TwoWheeler("Harley", "Davidson", 2021, 500));
        availableVehicles.add(new TwoWheeler("RoyalEnfield", "Hunter", 2020, 150));
```

```
        availableVehicles.add(new FourWheeler("Maruti", "Swift", 2019, 200));
        availableVehicles.add(new FourWheeler("Hyundai", "i20", 2020, 250));
    }
    // Method to display available vehicles
public void displayVehicles() {
    System.out.println("Available Vehicles:");
    System.out.println("---------------------------------");
    // Iterate through the list of available vehicles
    for (int i = 0; i < availableVehicles.size(); i++) {
        System.out.print((i + 1) + ".");
        // Call displayDetails method of each vehicle
        availableVehicles.get(i).displayDetails();
        System.out.println("---------------------------------");
    }
}
// Method to book a vehiclE
public void bookVehicle() {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Available Vehicles:");
    displayVehicles();  // Calling displayVehicles method
    System.out.println("Enter the vehicle number to book: ");
    int vehicleNumber = scanner.nextInt();
    if (vehicleNumber < 1 || vehicleNumber > availableVehicles.size()) {
        System.out.println("Invalid vehicle number!");
        return;
    }
    Vehicle selectedVehicle = availableVehicles.get(vehicleNumber - 1);
    if (!selectedVehicle.isAvailable()) {
        System.out.println("This vehicle is not available for booking!");
        return;
    }
    System.out.println("Enter the number of hours for rent: ");
    int hours = scanner.nextInt();
    System.out.println("Enter your name: ");
    String name = scanner.next();
    System.out.println("Enter your mobile number: ");
    String mobile = scanner.next();
    System.out.println("Enter the pickup point: ");
    String pickupPoint = scanner.next();
    System.out.println("Enter the return point: ");
    String returnPoint = scanner.next();
    int totalCost = selectedVehicle.getCostPHour() * hours;
```

```
124     System.out.println("---------------------------------\nVehicle Booked\n---------------------------------");
125     System.out.println("Bill Details:\n---------------------------------");
126     System.out.println("Name: " + name);
127     System.out.println("Mobile: " + mobile);
128     System.out.println("Pickup Point: " + pickupPoint);
129     System.out.println("Return Point: " + returnPoint);
130     System.out.println("Vehicle: " + selectedVehicle.getCompany() + " " + selectedVehicle.getModel() + " (" +
    selectedVehicle.getYear() + ")");
131     System.out.println("Cost per Hour: Rs." + selectedVehicle.getCostPHour());
132     System.out.println("Total Cost: Rs." + totalCost);
133     System.out.println("---------------------------------");
134 
135     // Mark the selected vehicle as not available
136     selectedVehicle.setAvailable(false);
137 }
138 
139 // Method to add a new vehicle
140 public void addVehicle() {
141     Scanner scanner = new Scanner(System.in);
142     System.out.println("---------------------------------\nAdding a new vehicle...\n---------------------------------");
143     System.out.println("Enter vehicle type (1 for Two Wheeler, 2 for Four Wheeler): ");
144     int vehicleType = scanner.nextInt();
145     System.out.println("Enter company: ");
146     String company = scanner.next();
147     System.out.println("Enter model: ");
148     String model = scanner.next();
149     System.out.println("Enter year: ");
150     int year = scanner.nextInt();
151     System.out.println("Enter cost per hour: ");
152     int costPHour = scanner.nextInt();
153     if (vehicleType == 1) {
154         // Creating new TwoWheeler object and adding it to availableVehicles list
155         availableVehicles.add(new TwoWheeler(company, model, year, costPHour));
156     } else if (vehicleType == 2) {
157         // Creating new FourWheeler object and adding it to availableVehicles list
158         availableVehicles.add(new FourWheeler(company, model, year, costPHour));
159     } else {
160         System.out.println("Invalid vehicle type!");
161     }
162     System.out.println("---------------------------------\nNew vehicle added successfully!\n---------------------------------"
    );
163 }
164 
```

```java
}

// Main class
public class Main{
    public static void main(String[] args) {
        RentalSystem obj = new RentalSystem();
        Scanner scanner = new Scanner(System.in);
        int choice;
        do {
            System.out.println("---------------------------------");
            System.out.println("---------------------------------");
            System.out.println("BandiLo-The Vehicle RentalSystem");
            System.out.println("---------------------------------");
            System.out.println("1. Display available vehicles");
            System.out.println("2. Book vehicle");
            System.out.println("3. Add new vehicle");
            System.out.println("4. Exit");
            System.out.println("---------------------------------");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            switch (choice) {
                case 1 -> obj.displayVehicles();
                case 2 -> obj.bookVehicle();
                case 3 -> obj.addVehicle();
                case 4 -> System.out.println("Exiting...");
                default -> System.out.println("Invalid choice!");
            }
        } while (choice != 4);
        scanner.close();
    }
}

```