**Vehicle Rental System Documentation**

**Overview**

This system is designed to manage vehicle rentals. It allows admins to manage a fleet of vehicles (cars, motorcycles, and trucks), while customers can rent or return vehicles. The system also supports vehicle availability, rental cost calculation, and basic authentication for administrative tasks.

**Key Classes and Interfaces**

1. **Admin**
   * Manages a fleet of vehicles and handles administrative tasks such as adding vehicles to the fleet and displaying available vehicles.
   * **Methods**:
     + verifyPassword(String inputPassword): Verifies the admin's password.
     + addVehicle(Vehicle vehicle): Adds a vehicle to the fleet.
     + displayFleet(): Displays all vehicles in the fleet.
     + displayAvailableVehicles(): Displays only available vehicles.
     + getVehicleById(String vehicleId): Retrieves a vehicle by its ID.
2. **App**
   * The main entry point of the application where users interact with the system.
   * **Features**:
     + Menu-driven interface to rent or return vehicles, add a vehicle (admin), display the fleet (admin), or exit.
     + Admin verification before performing administrative actions (adding a vehicle or displaying the fleet).
     + Allows users to rent vehicles by selecting available ones and entering rental details.
3. **Vehicle**
   * An abstract class that represents a generic vehicle.
   * **Attributes**:
     + vehicleId: Unique identifier for the vehicle.
     + model: The model of the vehicle.
     + baseRentalRate: Daily rental rate for the vehicle.
     + isAvailable: Availability status.
   * **Abstract Methods**:
     + calculateRentalCost(int days): Calculates the rental cost based on the number of rental days.
     + isAvailableForRental(): Determines if the vehicle is available for rental.
4. **Car, Motorcycle, and Truck** (Concrete subclasses of Vehicle)
   * Each class represents a specific type of vehicle (car, motorcycle, truck) and implements the Rentable interface.
   * **Methods**:
     + calculateRentalCost(int days): Returns the rental cost for the respective vehicle.
     + isAvailableForRental(): Checks if the vehicle is available.
     + rent(Customer customer, int days): Rents the vehicle to a customer if available.
     + returnVehicle(): Marks the vehicle as returned and available for rental.
5. **Rentable (Interface)**
   * Defines the contract for vehicles that can be rented.
   * **Methods**:
     + rent(Customer customer, int days): Initiates a rental for the vehicle.
     + returnVehicle(): Marks the vehicle as returned.
6. **Customer**
   * Represents a customer with a name who can rent vehicles.
   * **Attributes**:
     + name: The customer's name.
   * **Methods**:
     + getName(): Retrieves the customer's name.

**Flow of the Application**

* **Admin Actions**: Admin users can add vehicles to the fleet or display the fleet after verifying their password.
* **Customer Actions**: Customers can rent available vehicles by selecting a vehicle from the list, specifying rental days, and providing their name.
* **Vehicle Return**: When a customer returns a vehicle, it is marked as available again for future rentals.

**Vehicle Types and Rental Cost Calculation**

* Each vehicle (Car, Motorcycle, and Truck) calculates the rental cost based on its daily rate and the number of rental days.
* Rental cost is calculated using the formula:

rentalCost = baseRentalRate \* rentalDays;