```
public interface Rentable {
    void rent(Customer customer, int days);

    void returnVehicle();
    }
    class Car extends Vehicle implements Rentable {
    private final double baseRate = 50;

    public Car(String brand, String model, String registrationNumber) {
        super(brand, model, registrationNumber);
    }

        @Override
        public double calculateRentalCost(int days) {
            return days * baseRate;
        }

        @Override
        public boolean isAvailableForRental() {
            return isAvailableForRental();
        }
        public void rent(Customer customer, int days) {
        if (!isAvailableForRental()) {
            System.out.println("Car is not available for rental.");
            return;
        }
        setAvailable(false);
            System.out.println("Car rented by " + customer.getName() + " for " +
days + " days .");
        }
        public void returnVehical() {
        setAvailable(true);
            System.out.println("Car has been returned.");
        }
    }
    class RentalAgency {
    public void rentVehicle(Car car, Customer customer, int days) {
        car.rent(customer, days);
    }
    public void rentVehicle(Motorcycle motorcycle, Customer customer, int days) {
        motorcycle.rent(customer, days);
    }
    }
```