Vehicle Abstract class

package com.rental.vehicles;

public abstract class Vehicle {

// Private encapsulated fields

private final String vehicleId;

private final String model;

private final double baseRentalRate;

private boolean isAvailable;

// Constructor

public Vehicle(String vehicleId, String model, double baseRentalRate) {

if (baseRentalRate <= 0) {

throw new IllegalArgumentException("Base rental rate must be positive.");

}

this.vehicleId = vehicleId;

this.model = model;

this.baseRentalRate = baseRentalRate;

this.isAvailable = true;

}

// Getters

public String getVehicleId() { return vehicleId; }

public String getModel() { return model; }

public double getBaseRentalRate() { return baseRentalRate; }

public boolean isAvailable() { return isAvailable; }

// Setters with validation

public void setAvailable(boolean available) {

this.isAvailable = available;

}

// Abstract methods

public abstract double calculateRentalCost(int days);

public abstract boolean isAvailableForRental();

// toString() method

@Override

public String toString() {

return "Vehicle{" +

"vehicleId='" + vehicleId + '\'' +

", model='" + model + '\'' +

", baseRentalRate=" + baseRentalRate +

", isAvailable=" + isAvailable +

'}';

}

}

Car class

package com.rental.vehicles;

import com.rental.interfaces.Rentable;

public class Car extends Vehicle implements Rentable {

private final boolean hasGPS;

public Car(String vehicleId, String model, double baseRentalRate, boolean hasGPS) {

super(vehicleId, model, baseRentalRate);

this.hasGPS = hasGPS;

}