Fernandez Earl John Vincent

21-ITE-03

ACTIVITY:3-OOP

Vehicle Management System: Understanding Inheritance and Polymorphim

<?php

// Base class Vehicle

class Vehicle {

protected $make;

protected $model;

protected $year;

public function \_\_construct($make, $model, $year) {

$this->make = $make;

$this->model = $model;

$this->year = $year;

}

final public function getDetails() {

return "{$this->year} {$this->make} {$this->model}";

}

public function describe() {

return "This is a vehicle.";

}

}

// Car class extending Vehicle

class Car extends Vehicle {

private $numberOfDoors;

public function \_\_construct($make, $model, $year, $numberOfDoors) {

parent::\_\_construct($make, $model, $year);

$this->numberOfDoors = $numberOfDoors;

}

public function describe() {

return "This is a car with {$this->numberOfDoors} doors.";

}

}

// Final Motorcycle class extending Vehicle

final class Motorcycle extends Vehicle {

public function describe() {

return "This is a motorcycle.";

}

}

// ElectricVehicle Interface

interface ElectricVehicle {

public function chargeBattery();

}

// ElectricCar class extending Car and implementing ElectricVehicle

class ElectricCar extends Car implements ElectricVehicle {

public function chargeBattery() {

return "The battery is being charged.";

}

public function describe() {

return "This is an electric car.";

}

}

// Testing the implementation

$car = new Car('Toyota', 'Camry', 2020, 4);

echo $car->getDetails() . "\n";

echo $car->describe() . "\n";

$motorcycle = new Motorcycle('Harley-Davidson', 'Sportster', 2021);

echo $motorcycle->getDetails() . "\n";

echo $motorcycle->describe() . "\n";

$electricCar = new ElectricCar('Tesla', 'Model 3', 2022, 4);

echo $electricCar->getDetails() . "\n";

echo $electricCar->describe() . "\n";

echo $electricCar->chargeBattery() . "\n";

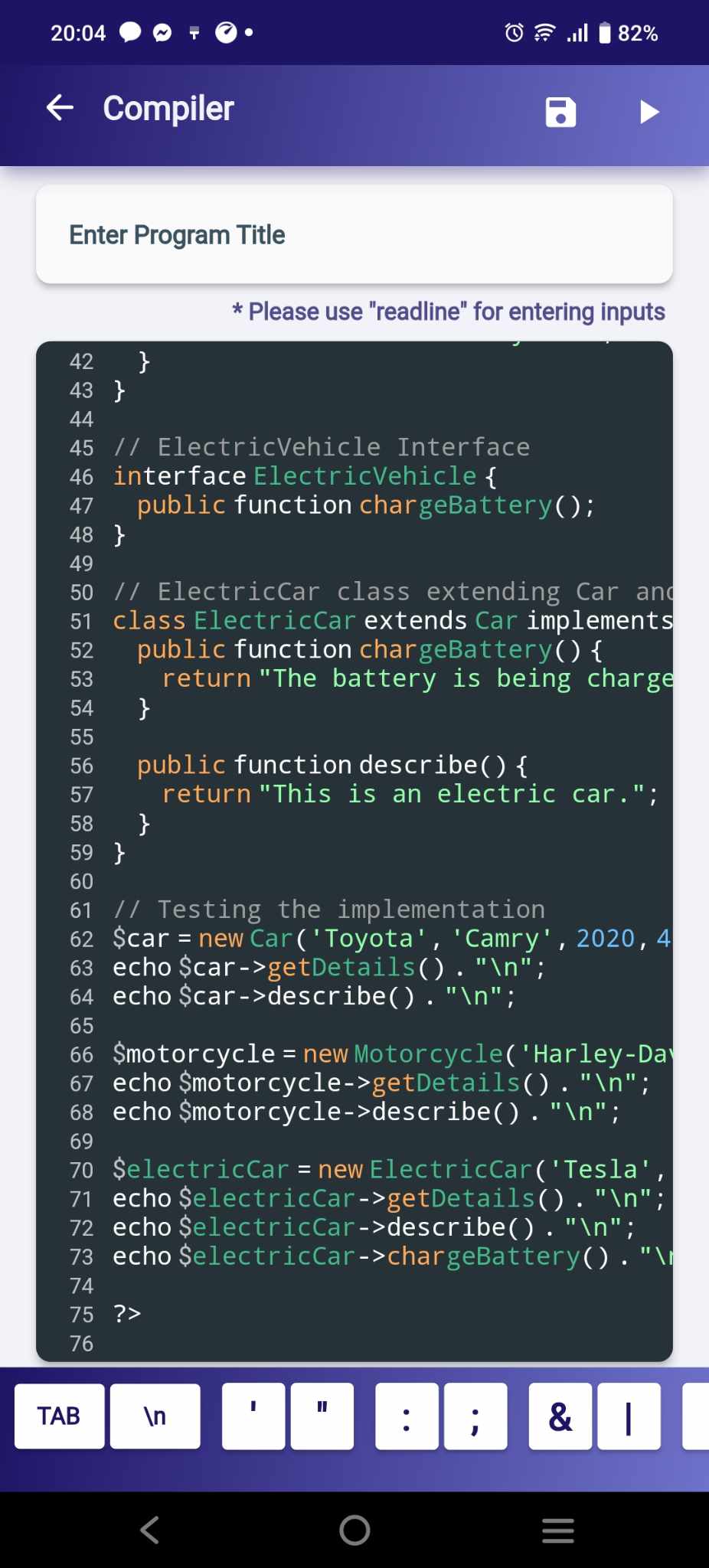
?>

Explanation///

- \*\*Inheritance:\*\* The `Car` and `Motorcycle` classes inherit from `Vehicle`, reusing the core properties while adding their specific behaviors.

- \*\*Polymorphism:\*\* The `describe()` method is overridden in each derived class to exhibit different behaviors based on the type of vehicle.

- \*\*Final Keyword:\*\* The `Motorcycle` class is marked as `final`, preventing further inheritance. Similarly, the `getDetails()` method in the `Vehicle` class is `final` to ensure consistency.

- \*\*Interface Implementation:\*\* The `ElectricVehicle` interface is implemented by the `ElectricCar` class, adding the functionality to charge a battery.

