4 Объекты и классы: принципы ООП в Java и наследование классов, передача метода объектов.

Задача

Ваша программа должна быть организована по пакетам:

- Пакет `vehicles` для классов `Vehicle`, `Car`, `ElectricCar`
- Пакет `app` для тестового класса `TestCar`.

Используя программу, выполненную во 2 и 3 практических работах, внести следующие изменения:

1. Добавить абстрактный класс Vehicle, который будет представлять общие свойства всех транспортных средств. В этот класс включите следующие общие поля для транспортных средств: `model` (модель); `license` (номерной знак); `color` (цвет); `year` (год выпуска); `ownerName` (имя владельца); `insuranceNumber` (страховой номер); `engineType` (тип двигателя, поле должно быть защищённым для наследования). Определите абстрактный метод `vehicleType()`, который будет возвращать тип транспортного средства. Добавьте методы для получения и изменения значений полей (геттеры и сеттеры).

- Изменить класс Car, чтобы он наследовал Vehicle. Реализуйте абстрактный метод vehicleType(), чтобы он возвращал "Car". В конструкторе класса `Car` используйте поля и методы родительского класса.

- Изменить класс ElectricCar, чтобы он наследовал Car. Добавьте в класс поле batteryCapacity (емкость аккумулятора) и методы для работы с ним. Реализуйте метод vehicleType(), который будет возвращать "Electric Car". Используйте protected-поле engineType для установки значения "Electric" в классе ElectricCar.

- Использовать полиморфизм в тестовом классе для работы с объектами Car и ElectricCar через ссылки на родительские классы. Создайте объекты `Car` и `ElectricCar`, измените их свойства с помощью сеттеров, и выведите информацию на экран с помощью метода `toString()`.

- Включить инкапсуляцию: убедитесь, что поля каждого класса имеют доступ через методы (геттеры и сеттеры), а не напрямую.