Trong bài này, mình bắt đầu từ yêu cầu thực tế: quản lý các loại phương tiện trong một đội xe. Theo lý thuyết OOP, bước 1 là xác định thực thể (class) và thuộc tính,hành vi cốt lõi. Mình chọn lớp Vehicle làm đại diện cho phương tiện với hai thuộc tính cơ bản: type (loại) và maxSpeed (tốc độ tối đa). Sau đó mở rộng thêm fuelType để phản ánh đặc trưng vận hành. Về hành vi, phương thức accelerate mô phỏng hành động tăng tốc, còn displayInfo in đầy đủ thông tin để quan sát trạng thái đối tượng.

Khó khăn chủ yếu là tốc độ âm hay không, và phản hồi thế nào khi nhập sai. Mình quyết định chuẩn hóa maxSpeed về 0 nếu dữ liệu âm và để accelerate xử lý tình huống này bằng thông điệp rõ ràng. Ngoài ra, để đảm bảo độ bền vững khi chạy thực tế, mình bổ sung các hàm nhập liệu an toàn thay vì phụ thuộc vào luồng cin trực tiếp, giúp tránh lỗi runtime khi người dùng nhập chữ thay vì số.

Cách tiếp cận OOP giúp cô lập dữ liệu và hành vi liên quan trong lớp Vehicle.Mình xây dựng một bộ test nhỏ gồm các trường hợp bình thường và biên (maxSpeed = 0, maxSpeed âm, loại nhiên liệu khác nhau) để kiểm tra. Qua nhiều lần biên dịch và chạy, mình tinh chỉnh thông báo, quy ước đặt tên CamelCase, và chú thích mã để mã nguồn sạch, dễ đọc, và bám sát quy trình đã học.