

Báo cáo Bài tập: Public Transportation Station Management System

1. Phân tích OOA

- Đối tượng chính: Station, Vehicle, ExpressBus, Passenger, Schedule.
- Thuộc tính:
 - Station: name, location, type, schedules
 - Vehicle: route, capacity, status, bookedSeats
 - ExpressBus (kế thừa Vehicle): speed, fewer stops
 - Passenger: name, id, tickets
- Phương thức:
 - Station: addSchedule(), removeSchedule(), showSchedules()
 - Vehicle: assignToStation(), calculateTravelTime(), displayInfo(), bookPassenger()
 - ExpressBus: calculateTravelTime() (ghi đè), displayInfo() (ghi đè)
 - Passenger: book(), cancel(), showTickets()
- Quan hệ kế thừa: ExpressBus kế thừa Vehicle.

2. Thiết kế và cài đặt bằng C++

Trong chương trình, mình xây dựng 4 lớp chính:

- Vehicle: lớp cơ bản, có thông tin tuyến đường, sức chứa, tình trạng.
- ExpressBus: kế thừa Vehicle, thêm tốc độ và ghi đè phương thức tính thời gian đi lại.
- Station: quản lý lịch trình (tối đa 10), thêm hoặc xóa phương tiện.
- Passenger: có tên, id, vé đã đặt, có thể đặt hoặc hủy vé.

Mỗi lớp đều có constructor và destructor để thể hiện vòng đời đối tượng.

- Constructor: khởi tạo giá trị ban đầu cho các thuộc tính.
- Destructor: được gọi khi object bị hủy, mình tận dụng để in ra thông báo giúp quan sát.

Cái hay của từ khóa virtual:

Nếu mình gọi phương thức qua con trỏ hoặc tham chiếu kiểu Vehicle nhưng trỏ đến ExpressBus, thì nhờ có virtual, chương trình sẽ chạy đúng hàm ở lớp con. Nếu không có virtual, nó sẽ gọi sai (theo kiểu biến). Đây là điểm quan trọng để thể hiện đa hình trong OOP.

3. Kiểm thử và kết quả

Trong phần kiểm thử, mình tạo một số đối tượng Station, Vehicle, ExpressBus, Passenger. Sau đó:

- Thêm lịch trình cho Station.
- Đặt vé cho Passenger, khi hết chỗ sẽ báo lỗi.
- In thông tin của Vehicle và ExpressBus để thấy sự khác biệt.
- Tính toán thời gian đi lại với cùng quãng đường, ExpressBus nhanh hơn Vehicle.
- Quan sát destructor in ra thông báo khi object bị hủy.

Mình cũng thêm các dòng phân cách bằng cout để kết quả hiển thị rõ ràng và dễ đọc hơn.

5. Cách mình dùng LLM (ChatGPT)

Mình có nhờ ChatGPT gợi ý cấu trúc các lớp và quan hệ kế thừa để tiết kiệm thời gian suy nghĩ. Sau đó mình chỉnh code lại cho đơn giản, phù hợp trình độ sinh viên.

Mình cũng nhờ ChatGPT dịch đề, giải thích rõ constructor và destructor, và bổ sung destructor cho đầy đủ.

Code cuối cùng mình tự kiểm thử và chỉnh sửa theo nhu cầu bài tập.

-----Hết-----