**PHẦN 1: TRẢ LỜI CÂU HỎI ÔN TẬP**

Câu hỏi 1: What is the difference between an open layer and a closed layer?

Câu trả lời: Một lớp “closed” nghĩa là khi một yêu cầu di chuyển từ lớp này sang lớp khác theo chiều từ trên xuống dưới, yêu cầu đó không được phép bỏ qua bất kỳ lớp trung gian nào mà phải đi qua lớp ngay bên dưới. Ngược lại, một lớp “open” cho phép một lớp nghiệp vụ (hoặc lớp bên trong) có thể truy cập trực tiếp đến lớp tiếp theo bên dưới mà không cần đi qua lớp trung gian, tạo ra sự linh hoạt trong việc định tuyến yêu cầu.

Câu hỏi 2: Describe the layers of isolation concept and what the benefits are of this concept.

Câu trả lời: Khái niệm “layers of isolation” chỉ ra rằng các lớp trong kiến trúc được cách ly với nhau, nghĩa là bất kỳ thay đổi nào ở một lớp (miễn là hợp đồng giao tiếp giữa các lớp không bị thay đổi) sẽ không ảnh hưởng đến các lớp khác. Ưu điểm của cách tiếp cận này là giúp bạn có thể thay thế, nâng cấp hoặc sửa đổi một lớp mà không làm gián đoạn toàn bộ hệ thống, từ đó giảm thiểu sự phụ thuộc lẫn nhau và nâng cao khả năng bảo trì cũng như mở rộng của hệ thống.

Câu hỏi 3: What is the architecture sinkhole anti-pattern?

Câu trả lời: Anti-pattern “architecture sinkhole” xảy ra khi một lớp hoặc một phần của kiến trúc trở thành “hố hút” – nhận và xử lý một tỷ lệ quá lớn các yêu cầu (ví dụ theo quy tắc 80-20, nếu 80% các yêu cầu đều rơi vào lớp này). Điều này cho thấy lớp đó đang bị quá tải và khó duy trì, từ đó báo hiệu rằng kiến trúc phân tầng có thể không phù hợp với miền vấn đề. Một giải pháp có thể là chuyển đổi các lớp thành “open” để giảm bớt sự phụ thuộc chặt chẽ, mặc dù điều này lại làm tăng độ phức tạp trong việc quản lý sự thay đổi của hệ thống.

Câu hỏi 4: What are some of the main architecture characteristics that would drive you to use a layered architecture?

Câu trả lời: Một số đặc tính kiến trúc khiến ta lựa chọn sử dụng kiến trúc phân tầng là khả năng tách biệt rõ ràng các mối quan tâm (như giao diện người dùng, nghiệp vụ và truy cập dữ liệu), giúp tăng cường khả năng bảo trì và cho phép thay thế, nâng cấp từng phần riêng biệt mà không ảnh hưởng đến toàn bộ hệ thống. Kiến trúc phân tầng cũng phù hợp cho các ứng dụng có quy mô nhỏ, đơn giản, đặc biệt khi ngân sách và thời gian phát triển hạn chế vì chi phí triển khai ban đầu thấp và dễ tiếp cận.

Câu hỏi 5: Why isn’t testability well supported in the layered architecture style?

Câu trả lời: Testability không được hỗ trợ tốt trong kiến trúc phân tầng bởi vì mặc dù các lớp được chia tách rõ ràng, nhưng trong thực tế, các lớp—đặc biệt là các lớp “closed”—thường có sự phụ thuộc lẫn nhau chặt chẽ. Điều này khiến cho việc kiểm thử đơn vị trở nên phức tạp, vì khi kiểm tra một lớp, bạn buộc phải giả lập hoặc khởi tạo toàn bộ chuỗi phụ thuộc từ lớp trên xuống dưới, làm tăng độ phức tạp của các bài kiểm thử.

Câu hỏi 6: Why isn’t agility well supported in the layered architecture style?

Câu trả lời: Agility (tính linh hoạt) không được hỗ trợ tốt trong kiến trúc phân tầng do cấu trúc cố định và sự phụ thuộc chặt chẽ giữa các lớp. Khi yêu cầu nghiệp vụ thay đổi, các lớp có thể bị ảnh hưởng lẫn nhau do liên kết cứng nhắc, khiến việc điều chỉnh, mở rộng hay thay đổi hệ thống diễn ra chậm chạp và tốn nhiều thời gian cũng như công sức.