

Câu hỏi Trắc nghiệm

1. Lớp nào đại diện cho dữ liệu cần quản lý trong hệ thống?
C. Lớp thực thể (Entity class)
2. Lớp biên trong hệ thống có vai trò gì?
C. Giao tiếp với người dùng hoặc hệ thống bên ngoài
3. Quan hệ nào thể hiện sự kế thừa?
D. Inheritance
4. Sơ đồ lớp mô tả:
A. Các lớp và quan hệ giữa các lớp trong hệ thống
5. Quan hệ Include giữa các use case dùng khi:
B. Một use case cần gọi một use case khác để hoàn thành chức năng
6. Scenario là gì?
B. Một kịch bản mô tả cách hệ thống và người dùng tương tác
7. Quan hệ nào biểu diễn việc một lớp chứa lớp khác nhưng vẫn độc lập?
A. Aggregation
8. Sơ đồ tuần tự mô tả điều gì?
B. Thứ tự các thông điệp được trao đổi giữa các đối tượng
9. Lớp điều khiển trong mô hình MVC là:
C. Control
10. Để biểu diễn quan hệ giữa các lớp, ta dùng sơ đồ:
C. Sơ đồ lớp

Câu hỏi ngắn

1. Lớp thực thể là gì?
Là lớp biểu diễn dữ liệu và trạng thái cần lưu trữ trong hệ thống (Entity class).
2. Lớp điều khiển có vai trò gì?
Điều phối luồng công việc, xử lý logic giữa giao diện và dữ liệu.
3. Scenario là gì?
Là một kịch bản mô tả chi tiết cách người dùng tương tác với hệ thống trong một tình huống cụ thể.

4. Quan hệ Include giữa các use case là gì?
Một use case bao gồm (include) một use case khác để hoàn thành chức năng.
5. Mục đích của sơ đồ lớp là gì?
Mô tả cấu trúc tĩnh của hệ thống, gồm các lớp và mối quan hệ giữa chúng.
6. Khác nhau giữa Aggregation và Composition?
Aggregation: quan hệ “có – chứa” nhưng các phần tử vẫn tồn tại độc lập.
Composition: quan hệ “có – chứa” nhưng phần tử phụ thuộc hoàn toàn vào lớp cha.
7. Sơ đồ tuần tự là gì?
Biểu diễn trình tự gửi – nhận thông điệp giữa các đối tượng theo thời gian.
8. Quan hệ Extend giữa các use case là gì?
Một use case có thể được mở rộng bằng một use case khác trong một điều kiện cụ thể.
9. Lớp biên có vai trò gì?
Là lớp trung gian giữa hệ thống và người dùng/hệ thống ngoài, xử lý giao diện.
10. Sơ đồ cộng tác là gì?
Mô tả sự tương tác giữa các đối tượng và các thông điệp được trao đổi.

Câu hỏi Thảo luận nhóm

1. Vai trò từng loại lớp:
 - Biên (Boundary): giao tiếp ngoài.
 - Điều khiển (Control): xử lý logic.
 - Thực thể (Entity): quản lý dữ liệu.
2. So sánh Aggregation vs Composition:
 - Aggregation: tồn tại độc lập.
 - Composition: phụ thuộc.
3. Tầm quan trọng của sơ đồ lớp:
Là nền tảng thiết kế hướng đối tượng, giúp xác định cấu trúc hệ thống.
4. Phân biệt sơ đồ tuần tự và sơ đồ cộng tác:
Cả hai mô tả tương tác, nhưng sơ đồ tuần tự nhấn mạnh *thời gian*, còn cộng tác nhấn mạnh *quan hệ giữa đối tượng*.
5. Vai trò lớp điều khiển trong MVC:
Kết nối View và Model, xử lý yêu cầu người dùng.

6. Vì sao cần viết scenario:
Giúp hiểu rõ hành vi hệ thống, kiểm thử dễ hơn.
7. Đảm bảo use case đủ và đúng:
Kiểm tra yêu cầu khách hàng, mapping với chức năng thực tế.
8. Quan hệ giữa use case và scenario:
Scenario là các tình huống cụ thể hóa từ use case.
9. Ưu nhược điểm sơ đồ tuần tự:
Ưu: rõ ràng về thời gian. Nhược: khó đọc khi có nhiều đối tượng.
10. Cải thiện chất lượng scenario:
Mô tả rõ, đầy đủ trường hợp ngoại lệ, logic thực tế.

Câu hỏi tình huống

1. Trong quá trình phân tích hệ thống quản lý thư viện, nhóm phát triển phát hiện một số yêu cầu mới từ khách hàng sau khi đã viết xong các scenario. Nhóm phát triển nên xử lý như thế nào?

Nhóm cần phân tích lại các yêu cầu mới, cập nhật use case diagram và scenario tương ứng. Nếu yêu cầu ảnh hưởng đến luồng xử lý, phải sửa hoặc thêm các lớp liên quan (Boundary, Control, Entity). Sau đó kiểm tra lại tính nhất quán giữa use case, class và sequence diagram.

2. Một nhóm phát triển gặp khó khăn khi xác định các lớp điều khiển trong hệ thống. Hãy đề xuất giải pháp.

Xác định lớp điều khiển dựa trên các luồng xử lý nghiệp vụ chính (use case). Mỗi use case thường có một lớp điều khiển tương ứng để điều phối luồng dữ liệu giữa lớp biên và lớp thực thể. Nếu khó xác định, hãy vẽ sơ đồ tuần tự (sequence diagram) để thấy rõ đối tượng nào đang chịu trách nhiệm điều khiển logic.

3. Sau khi hoàn thành sơ đồ lớp, khách hàng yêu cầu thêm một số chức năng mới. Nhóm phát triển cần làm gì để cập nhật sơ đồ lớp?
 - Xem xét các chức năng mới ảnh hưởng đến lớp nào (biên, điều khiển, thực thể).
 - Thêm lớp mới hoặc mở rộng lớp hiện có, bổ sung các thuộc tính, phương thức và quan hệ liên quan.
 - Cập nhật lại các sơ đồ khác (use case, sequence) để đồng bộ hóa toàn hệ thống.
4. Khi viết các scenario cho use case “Đăng ký khóa học”, nhóm phát triển gặp nhiều trường hợp ngoại lệ. Làm thế nào để xử lý tình huống này?
 - Ghi rõ main flow (luồng chính) và alternate/exception flow (luồng ngoại lệ).
 - Với mỗi ngoại lệ, mô tả nguyên nhân – cách xử lý – thông báo lỗi.
 - Đảm bảo tất cả các trường hợp ngoại lệ đều được xét đến trong use case và sequence diagram.

5. Trong quá trình xây dựng sơ đồ tuần tự, một số đối tượng không có vai trò rõ ràng. Nhóm phát triển nên làm gì?
 - Rà soát lại use case để hiểu rõ nghiệp vụ.
 - Nếu đối tượng không có chức năng cụ thể → loại bỏ hoặc gộp vào đối tượng khác.
 - Chỉ giữ lại những đối tượng thực sự gửi/nhận thông điệp trong luồng xử lý.
6. Sau khi xây dựng sơ đồ lớp, nhóm phát triển phát hiện một số quan hệ giữa các lớp bị sai. Hãy đề xuất cách sửa chữa.
 - Kiểm tra lại nghiệp vụ thực tế và mối quan hệ giữa các đối tượng trong use case.
 - Xem quan hệ đó nên là association, aggregation, composition hay inheritance.
 - Sửa lại trên sơ đồ, đảm bảo các lớp và thuộc tính liên kết đúng logic hệ thống.
7. Một nhóm phát triển gặp khó khăn khi mô tả các quan hệ giữa các use case. Hãy đề xuất giải pháp.
 - Dựa vào mối liên hệ nghiệp vụ thực tế để xác định:
 - Quan hệ Include: một use case cần gọi use case khác.
 - Quan hệ Extend: một use case mở rộng use case khác trong trường hợp đặc biệt.
 - Kiểm tra lại phạm vi của từng use case để tránh chồng chéo hoặc trùng lặp chức năng.
8. Trong dự án phát triển phần mềm quản lý bán hàng, nhóm phát triển cần xác định các lớp biên cho hệ thống. Hãy đưa ra đề xuất phù hợp.
 - Lớp biên là các lớp giao tiếp với người dùng hoặc hệ thống ngoài.
 - Ví dụ:
 - **FormDangNhap**, **FormHoaDon**, **API_KhachHang**, **GiaoDienBanHang**
 - Các lớp này chịu trách nhiệm hiển thị dữ liệu, nhận input, truyền yêu cầu đến lớp điều khiển.
9. Khách hàng yêu cầu thêm chức năng mới sau khi các scenario đã được hoàn thiện. Nhóm phát triển nên làm gì?
 - Thảo luận lại với khách hàng để xác định rõ yêu cầu mới.
 - Cập nhật scenario mới hoặc mở rộng use case cũ.
 - Sau đó chỉnh sửa sơ đồ lớp, sequence và test case cho phù hợp.
10. Trong quá trình xây dựng sơ đồ cộng tác, một số thông điệp không tương tác đúng yêu cầu. Hãy đề xuất cách giải quyết.
 - Kiểm tra lại thứ tự và hướng trao đổi thông điệp giữa các đối tượng.
 - Đảm bảo các thông điệp khớp với logic của scenario và sequence diagram.
 - Nếu sai, chỉnh lại vai trò của đối tượng hoặc thông điệp gửi/nhận để phù hợp với yêu cầu thực tế.