Câu hỏi ngắn chương 6

1. Lớp thực thể là gì?
    - Lớp thực thể (Entity Class) mô tả các thực thể trong hệ thống phần mềm, thường ánh xạ trực tiếp đến bảng trong cơ sở dữ liệu. Lớp thực thể lưu trữ dữ liệu có trạng thái và thường tồn tại lâu dài trong hệ thống.
2. Lớp điều khiển có vai trò gì trong hệ thống?
    - Lớp điều khiển (Control Class) đóng vai trò trung gian giữa lớp giao diện (Boundary Class) và lớp thực thể (Entity Class). Nó chứa logic xử lý nghiệp vụ và điều phối luồng dữ liệu trong hệ thống.
3. Scenario là gì?
    - Scenario (kịch bản sử dụng) mô tả cách một tác nhân (user hoặc hệ thống khác) tương tác với hệ thống để thực hiện một chức năng cụ thể.
4. Quan hệ Include giữa các use case là gì?
    - Quan hệ Include thể hiện rằng một use case bao gồm một use case khác, có nghĩa là một use case không thể hoạt động độc lập mà bắt buộc phải gọi đến use case khác để hoàn thành chức năng.
5. Mục đích của sơ đồ lớp là gì?
    - Sơ đồ lớp giúp mô tả cấu trúc tĩnh của hệ thống bằng cách thể hiện các lớp, thuộc tính, phương thức và mối quan hệ giữa các lớp. Nó hỗ trợ việc thiết kế và triển khai phần mềm một cách rõ ràng.
6. Quan hệ Aggregation khác gì so với Composition?
    - Aggregation thể hiện mối quan hệ "có nhưng không sở hữu", nghĩa là đối tượng con có thể tồn tại độc lập với đối tượng cha. Trong khi đó, Composition thể hiện mối quan hệ "sở hữu mạnh", đối tượng con phụ thuộc hoàn toàn vào đối tượng cha và sẽ bị hủy khi đối tượng cha bị hủy.
7. Sơ đồ tuần tự là gì?
    - Sơ đồ tuần tự (Sequence Diagram) là một loại sơ đồ UML mô tả sự tương tác giữa các đối tượng trong hệ thống theo thứ tự thời gian, thể hiện luồng thông điệp được gửi giữa các đối tượng.
8. Quan hệ Extend giữa các use case là gì?
    - Quan hệ Extend mô tả việc một use case có thể mở rộng thêm chức năng cho một use case khác trong một số điều kiện nhất định.
9. Lớp biên có vai trò gì trong hệ thống?

- Lớp biên (Boundary Class) đóng vai trò giao tiếp giữa hệ thống và môi trường bên ngoài (người dùng hoặc hệ thống khác), quản lý tương tác, xử lý đầu vào từ người dùng và hiển thị kết quả.

10. Sơ đồ cộng tác là gì?

- Sơ đồ cộng tác (Collaboration Diagram) là một sơ đồ UML thể hiện sự tương tác giữa các đối tượng và sự trao đổi thông điệp giữa chúng trong một tình huống cụ thể.

Câu hỏi tình huống chương 6

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?

Trả lời:

- Đánh giá mức độ ảnh hưởng của yêu cầu mới đối với các scenario hiện có.
- Cập nhật hoặc bổ sung scenario để đảm bảo hệ thống đáp ứng yêu cầu mới.
- Kiểm tra lại các use case liên quan để đảm bảo sự nhất quán.

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

Trả lời:

- Sử dụng mô hình BCE (Boundary, Control, Entity) để xác định các lớp điều khiển.
- Lớp điều khiển chịu trách nhiệm xử lý logic nghiệp vụ và điều phối dữ liệu giữa lớp giao diện và lớp thực thể.
- Xác định các use case chính, sau đó phân tích các hành động để tách biệt logic điều khiển.

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?

Trả lời:

- Phân tích yêu cầu mới để xác định các lớp cần bổ sung hoặc sửa đổi.
- Cập nhật sơ đồ lớp bằng cách thêm các phương thức, thuộc tính hoặc quan hệ mới.
- Đánh giá tác động đến các phần khác của hệ thống để đảm bảo tính toàn vẹn.

4. Khi viết các scenario cho use case "Đăng ký khóa học", nhóm phát triển gặp tình huống có nhiều trường hợp ngoại lệ. Làm thế nào để xử lý tình huống này?

Trả lời:

- Viết các scenario ngoại lệ để mô tả cách hệ thống phản hồi khi xảy ra lỗi.

- Bao gồm các trường hợp như nhập sai thông tin, quá tải hệ thống, khóa tài khoản, v.v.
- Đảm bảo kiểm thử kỹ các scenario ngoại lệ để hệ thống hoạt động ổn định.

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?

Trả lời:

- Xem xét lại sơ đồ lớp để xác định vai trò của từng đối tượng.
- Nếu đối tượng không có chức năng cụ thể, có thể cần loại bỏ hoặc hợp nhất với các đối tượng khác.
- Đảm bảo rằng mỗi đối tượng trong sơ đồ tuần tự đều có một vai trò quan trọng trong quy trình nghiệp vụ.

6. Sau khi xây dựng sơ đồ lớp, nhóm phát triển phát hiện ra một số quan hệ giữa các lớp bị sai. Hãy đề xuất cách sửa chữa.

Trả lời:

- Xác định loại quan hệ chính xác (kế thừa, liên kết, kết tập, hợp thành) dựa trên nghiệp vụ.
- Sửa chữa các sai sót bằng cách sử dụng ký hiệu UML phù hợp.
- Kiểm tra lại sơ đồ để đảm bảo tính logic và nhất quán với yêu cầu 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.

Trả lời:

- Sử dụng các mối quan hệ kế thừa, bao gồm (include) hoặc mở rộng (extend) trong sơ đồ use case.
- Xác định các điểm giao thoa giữa các use case để tạo ra sự liên kết hợp lý.
- Kiểm tra lại yêu cầu khách hàng để đảm bảo mô hình phản ánh đúng nghiệp vụ.

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.

Trả lời:

- Lớp biên (Boundary Class) là lớp giao tiếp giữa hệ thống và người dùng hoặc hệ thống bên ngoài.
- Ví dụ: Các form nhập liệu, API handlers, bộ kiểm tra dữ liệu đầu vào.
- Đảm bảo các lớp biên chỉ thực hiện nhiệm vụ giao tiếp và không chứa logic nghiệp vụ.

9. Khách hàng yêu cầu thêm chức năng mới sau khi các scenario đã được hoàn thiện. Nhóm phát triển nên làm gì?

Trả lời:

- Phân tích yêu cầu mới và đánh giá tác động lên các scenario hiện có.
- Nếu cần thiết, cập nhật hoặc bổ sung scenario để phản ánh chức năng mới.
- Kiểm thử lại hệ thống để đảm bảo các thay đổi không gây lỗi cho chức năng cũ.

10. Trong quá trình xây dựng sơ đồ cộng tác, một số đối tượng không tương tác đúng theo yêu cầu. Hãy đề xuất cách giải quyết.

Trả lời:

- Xác định lại vai trò của các đối tượng trong hệ thống.
- Kiểm tra mối quan hệ giữa các đối tượng và đảm bảo chúng thực hiện đúng nhiệm vụ.
- Nếu cần, sửa đổi luồng tương tác để phù hợp với yêu cầu nghiệp vụ.

Câu hỏi thảo luận nhóm chương 6

1. Thảo luận về vai trò của từng loại lớp (thực thể, biên, điều khiển) trong hệ thống.

- Lớp thực thể (Entity): Lớp thực thể chịu trách nhiệm lưu trữ và quản lý dữ liệu cốt lõi của hệ thống, đại diện cho các đối tượng trong thế giới thực (ví dụ: Khách hàng, Sản phẩm). Chúng thường chứa các thuộc tính và logic nghiệp vụ cơ bản liên quan đến dữ liệu.
- Lớp biên (Boundary): Lớp biên đóng vai trò giao tiếp giữa hệ thống và người dùng hoặc các hệ thống bên ngoài. Đây là nơi xử lý các giao diện người dùng (UI) hoặc API, đảm bảo dữ liệu được truyền tải đúng cách.
- Lớp điều khiển (Control): Lớp điều khiển hoạt động như trung gian giữa lớp biên và lớp thực thể, điều phối luồng xử lý logic nghiệp vụ. Nó nhận yêu cầu từ lớp biên, xử lý và tương tác với lớp thực thể để trả về kết quả.

2. So sánh sự khác nhau giữa Aggregation và Composition.

- Aggregation: Là mối quan hệ "có một" (has-a) yếu hơn, trong đó các đối tượng con có thể tồn tại độc lập với đối tượng cha. Ví dụ: Một đội bóng (Team) và cầu thủ (Player), cầu thủ có thể rời đội mà vẫn tồn tại.
- Composition: Là mối quan hệ "có một" mạnh hơn, trong đó đối tượng con không thể tồn tại nếu không có đối tượng cha. Ví dụ: Một ngôi nhà (House) và phòng (Room), nếu ngôi nhà bị phá hủy, các phòng cũng không còn tồn tại.

3. Thảo luận về tầm quan trọng của việc xây dựng sơ đồ lớp trong quá trình phân tích hệ thống.

- Sơ đồ lớp đóng vai trò quan trọng trong việc phân tích hệ thống vì nó cung cấp một cái nhìn tổng quan về cấu trúc tĩnh của hệ thống. Cụ thể:

- Xác định các đối tượng và mối quan hệ: Sơ đồ lớp giúp xác định các lớp, thuộc tính, phương thức và mối quan hệ giữa chúng (kế thừa, liên kết, phụ thuộc), từ đó làm rõ cấu trúc hệ thống.
- Hỗ trợ thiết kế và triển khai: Sơ đồ lớp là nền tảng để lập trình viên triển khai mã nguồn, đảm bảo tính nhất quán giữa thiết kế và thực thi.
- Tăng khả năng bảo trì: Việc có sơ đồ lớp rõ ràng giúp dễ dàng hiểu và bảo trì hệ thống trong tương lai, đặc biệt khi có thay đổi hoặc mở rộng.
- Giao tiếp trong nhóm: Sơ đồ lớp là công cụ giao tiếp hiệu quả giữa các thành viên trong nhóm phát triển, giúp thống nhất ý tưởng và yêu cầu.

4. Phân biệt sơ đồ tuần tự và sơ đồ cộng tác.

- Sơ đồ tuần tự (Sequence Diagram):
  - Tập trung vào luồng thời gian của các tương tác giữa các đối tượng trong hệ thống.
  - Hiển thị các thông điệp được gửi giữa các đối tượng theo thứ tự thời gian, thường được biểu diễn theo trục thời gian dọc.
  - Thích hợp để mô tả chi tiết một kịch bản cụ thể (use case) và cách các đối tượng tương tác theo thời gian.
- Sơ đồ cộng tác (Collaboration Diagram):
  - Tập trung vào mối quan hệ không gian giữa các đối tượng và cách chúng tương tác.
  - Hiển thị các đối tượng và thông điệp giữa chúng mà không nhấn mạnh vào thời gian, thường sử dụng số thứ tự để chỉ ra trình tự.
  - Thích hợp để mô tả cấu trúc tổng thể của các đối tượng và mối quan hệ giữa chúng trong một kịch bản.

5. Thảo luận về vai trò của lớp điều khiển trong mô hình MVC.

- Trong mô hình MVC (Model-View-Controller), lớp điều khiển (Controller) đóng vai trò quan trọng:
  - Trung gian giữa Model và View: Controller nhận yêu cầu từ người dùng (thông qua View), xử lý logic và tương tác với Model để cập nhật dữ liệu, sau đó trả kết quả về View để hiển thị.
  - Điều phối luồng xử lý: Controller đảm bảo các yêu cầu được xử lý đúng cách, giúp tách biệt logic nghiệp vụ (Model) và giao diện người dùng (View).
  - Tăng tính tái sử dụng: Nhờ Controller, Model và View có thể hoạt động độc lập, giúp dễ dàng thay đổi hoặc tái sử dụng các thành phần mà không ảnh hưởng đến nhau.
  - Quản lý tương tác: Controller xử lý các sự kiện từ người dùng (như nhấn nút, nhập dữ liệu) và quyết định cách phản hồi.

6. Tại sao cần viết các scenario khi phân tích hệ thống?

- Việc viết các scenario (kịch bản) khi phân tích hệ thống là cần thiết vì:
  - Hiểu rõ yêu cầu: Scenario giúp mô tả chi tiết cách hệ thống hoạt động trong các tình huống cụ thể, từ đó làm rõ yêu cầu của người dùng.
  - Kiểm tra tính khả thi: Scenario cho phép nhóm phát triển kiểm tra xem hệ thống có đáp ứng được các trường hợp sử dụng thực tế hay không.

- Hỗ trợ kiểm thử: Các scenario cung cấp cơ sở để xây dựng các trường hợp kiểm thử (test case), đảm bảo hệ thống hoạt động đúng như mong đợi.
- Giao tiếp với khách hàng: Scenario giúp khách hàng và nhóm phát triển hiểu rõ hơn về cách hệ thống sẽ hoạt động, giảm thiểu hiểu lầm.
- Phát hiện lỗi sớm: Viết scenario giúp phát hiện các vấn đề hoặc thiếu sót trong yêu cầu trước khi tiến hành thiết kế và triển khai.

7. Làm thế nào để đảm bảo rằng các use case được trich đầy đủ và chính xác?

- Để đảm bảo các use case được trích xuất đầy đủ và chính xác, cần thực hiện các bước sau:
    - Thu thập yêu cầu kỹ lưỡng: Phỏng vấn và làm việc chặt chẽ với các bên liên quan (stakeholders) để hiểu rõ nhu cầu và mục tiêu của hệ thống.
    - Phân tích quy trình nghiệp vụ: Nghiên cứu các quy trình hiện tại để xác định tất cả các tác nhân (actors) và cách họ tương tác với hệ thống.
    - Sử dụng kỹ thuật phân tích: Áp dụng các kỹ thuật như phân tích luồng công việc, quan sát người dùng, hoặc tổ chức các buổi thảo luận nhóm (brainstorming).
    - Xác định các trường hợp sử dụng chính và phụ: Đảm bảo bao quát cả các trường hợp thông thường (main flow) và các trường hợp ngoại lệ (alternative/exceptional flow).
    - Xác nhận với khách hàng: Trình bày các use case cho khách hàng và các bên liên quan để xác nhận tính chính xác và đầy đủ.
    - Sử dụng công cụ hỗ trợ: Sử dụng các công cụ như sơ đồ luồng dữ liệu (DFD) hoặc sơ đồ hoạt động (Activity Diagram) để hỗ trợ việc trích xuất use case.

8. Thảo luận về mối quan hệ giữa use case và scenario.

- Use case: Là một mô tả tổng quát về một chức năng của hệ thống, bao gồm tất cả các kịch bản có thể xảy ra khi tác nhân (actor) tương tác với hệ thống để đạt được một mục tiêu cụ thể. Ví dụ: "Đăng nhập hệ thống" là một use case.
- Scenario: Là một trường hợp cụ thể trong use case, mô tả một luồng sự kiện cụ thể (thường là luồng chính hoặc luồng phụ). Ví dụ: Trong use case "Đăng nhập hệ thống", một scenario có thể là "Người dùng nhập đúng tên và mật khẩu" (luồng chính) hoặc "Người dùng nhập sai mật khẩu" (luồng phụ).
- Mối quan hệ:
    - Một use case thường bao gồm nhiều scenario, đại diện cho các cách khác nhau để hoàn thành chức năng.
    - Scenario là cách để chi tiết hóa use case, giúp làm rõ các bước và điều kiện cụ thể.
    - Use case cung cấp cái nhìn tổng quan, trong khi scenario đi sâu vào chi tiết từng trường hợp.

9. Phân tích ưu và nhược điểm của việc sử dụng sơ đồ tuần tự trong thiết kế hệ thống.

- Ưu điểm:
    - Rõ ràng về luồng thời gian: Sơ đồ tuần tự thể hiện rõ thứ tự các thông điệp và tương tác giữa các đối tượng theo thời gian, giúp dễ hiểu cách hệ thống hoạt động.
    - Hỗ trợ phân tích chi tiết: Phù hợp để phân tích các kịch bản cụ thể, đặc biệt là các luồng chính và luồng ngoại lệ.
    - Dễ giao tiếp: Sơ đồ tuần tự dễ hiểu đối với cả kỹ thuật viên và các bên liên quan không chuyên về kỹ thuật.
    - Hỗ trợ kiểm thử: Dựa vào sơ đồ tuần tự, nhóm kiểm thử có thể xây dựng các trường hợp kiểm thử chi tiết.
- Nhược điểm:
    - Khó quản lý với hệ thống phức tạp: Khi hệ thống có quá nhiều đối tượng và tương tác, sơ đồ tuần tự có thể trở nên rối và khó đọc.
    - Không thể hiện mối quan hệ không gian: Sơ đồ tuần tự không tập trung vào cấu trúc tổng thể của các đối tượng, mà chỉ chú trọng vào thời gian.
    - Khó bảo trì: Nếu hệ thống thay đổi, việc cập nhật sơ đồ tuần tự có thể tốn thời gian và dễ gây nhầm lẫn.
    - Hạn chế trong việc tái sử dụng: Sơ đồ tuần tự thường được xây dựng cho từng kịch bản cụ thể, nên không thể hiện được tính tổng quát của hệ thống.

10. Thảo luận về cách cải thiện chất lượng kịch bản sử dụng (scenario) trong quá trình phân tích.

- Để cải thiện chất lượng kịch bản sử dụng (scenario) trong quá trình phân tích, có thể áp dụng các cách sau:
    - Xác định rõ mục tiêu: Đảm bảo mỗi scenario đều phục vụ một mục tiêu cụ thể của use case, tránh lan man hoặc không cần thiết.
    - Bao quát các trường hợp: Xây dựng scenario cho cả luồng chính (main flow), luồng phụ (alternative flow) và luồng ngoại lệ (exceptional flow) để đảm bảo tính đầy đủ.
    - Sử dụng ngôn ngữ rõ ràng: Viết scenario bằng ngôn ngữ đơn giản, dễ hiểu, tránh thuật ngữ kỹ thuật phức tạp để các bên liên quan đều có thể hiểu.
    - Kiểm tra tính khả thi: Đảm bảo rằng các scenario được viết là thực tế và có thể triển khai trong hệ thống.
    - Phân tích rủi ro: Xem xét các rủi ro hoặc vấn đề tiềm ẩn trong mỗi scenario và đưa ra cách xử lý.
    - Hợp tác với các bên liên quan: Làm việc chặt chẽ với khách hàng, người dùng cuối và nhóm phát triển để xác nhận và tinh chỉnh các scenario.
    - Sử dụng công cụ hỗ trợ: Áp dụng các công cụ như sơ đồ tuần tự hoặc sơ đồ hoạt động để trực quan hóa scenario, giúp dễ dàng phát hiện thiếu sót.
    - Kiểm tra và cải tiến liên tục: Thường xuyên xem xét và cập nhật các scenario để đảm bảo chúng luôn phù hợp với yêu cầu và thay đổi của hệ thống.

Câu hỏi trắc nghiệm chương 6

Câu 1 : Kỹ thuật nào sau đây thường được sử dụng để thu thập yêu cầu khách hàng?
Đáp án: B. Phỏng vấn khách hàng

Câu 2 : Trong bước tổng hợp kết quả yêu cầu, việc nào sau đây là cần thiết?
Đáp án: B. Phân loại yêu cầu và loại bỏ vòng lặp thông tin trùng lặp

Câu 3 : Use case mô tả:
Đáp án: B. Cách người dùng tương tác với hệ thống

Câu 4 : Tại sao cần xây dựng danh sách từ khóa chuyên môn khi tìm hiểu lĩnh vưc ứng dụng?
Đáp án: C. Để đảm bảo đội phát triển và khách hàng hiểu các thuật ngữ

Câu 5: Quan hệ nào sau đây trong use case mô tả một use case có thể mở rộng thêm chức năng trong một số điều kiện nhất định.
Đáp án: B. Extend

Câu 6 : Khi mô tả yêu cầu bằng ngôn ngữ tự nhiên, điều quan trọng nhất là gì?
Đáp án: C. Viết rõ ràng, dễ hiểu và tránh từ đa nghĩa

Câu 7: Mục tiêu chính của việc trích các use case là gì?
Đáp án: B. Mô tả các chức năng mà hệ thống cung cấp cho người dùng

Câu 8 : Quan hệ nào giữa các use case thể hiện việc một use case phải gọi một use case khác để thưc hiện chức năng đầy đủ?
Đáp án: A. Include

Câu 9 : Hoạt động nào sau đây là bước đầu tiên trong việc xây dựng mô hình nghiệp vụ?
Đáp án: B. Trích các use case

Câu 10 : Một yêu cầu chức năng là gì?
Đáp án: A. Một yêu cầu mô tả cách hệ thống xử lý dữ liệu