**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

Bài tập cá nhân nhóm 11

Sinh viên thực hiện:

Nguyễn Văn Hiền



**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

Part 1: Câu hỏi trắc nghiệm

Câu 1: Lớp nào đại diện cho các đối tượng có dữ liệu và trạng thái cần quản lý trong hệ thống?

Đáp án đúng: C. Lớp thực thể

Giải thích:

* Lớp thực thể (Entity Class) lưu trữ dữ liệu và trạng thái của hệ thống.
* Lớp biên (A) xử lý giao tiếp với người dùng hoặc hệ thống bên ngoài.
* Lớp điều khiển (B) quản lý luồng công việc.
* Lớp giao diện (D) chỉ để hiển thị thông tin.

Câu 2: Lớp biên trong hệ thống có vai trò gì?

Đáp án đúng: C. Giao tiếp với người dùng hoặc hệ thống bên ngoài

Giải thích:

* Lớp biên (Boundary Class) giúp giao tiếp với người dùng hoặc hệ thống khác.
* Xử lý dữ liệu (A) thuộc về lớp thực thể.
* Điều khiển luồng công việc (B) thuộc về lớp điều khiển.
* Lưu trữ dữ liệu (D) thuộc về lớp thực thể.

Câu 3: Quan hệ nào giữa các lớp thể hiện sự kế thừa?

Đáp án đúng: D. Inheritance

Giải thích:

* Kế thừa (Inheritance) cho phép một lớp con mở rộng hoặc thay đổi hành vi của lớp cha.
* Association (A): Quan hệ liên kết giữa các lớp.
* Aggregation (B): Một lớp chứa một hoặc nhiều lớp khác nhưng lớp con vẫn tồn tại độc lập.
* Composition (C): Một lớp chứa lớp khác, nhưng lớp con không thể tồn tại độc lập.

Câu 4: Sơ đồ lớp mô tả gì?

Đáp án đúng: B. Các lớp và quan hệ giữa các lớp trong hệ thống

Giải thích:

* Sơ đồ lớp (Class Diagram) mô tả cấu trúc hệ thống bằng cách hiển thị các lớp, thuộc tính, phương thức và quan hệ giữa chúng.
* A (Luồng xử lý): Sơ đồ trình tự (Sequence Diagram) biểu diễn.
* C (Giao diện người dùng): Không phải mục đích của sơ đồ lớp.
* D (Thứ tự xử lý): Thuộc sơ đồ tuần tự.

Câu 5: Quan hệ Include giữa các use case được dùng khi nào?

Đáp án đúng: B. Một use case cần gọi một use case khác để hoàn thành chức năng

Giải thích:

* Include dùng để biểu diễn rằng một use case này phụ thuộc vào use case khác để hoàn thành.
* A (Mở rộng use case): Thuộc về quan hệ Extend.
* C (Kế thừa use case khác): Không có khái niệm kế thừa trong use case.
* D (Use case dùng bởi hệ thống khác): Không phải quan hệ Include.

Câu 6: Scenario là gì?

Đáp án đúng: B. Một kịch bản mô tả cách hệ thống và người dùng tương tác

Giải thích:

* Scenario (Kịch bản) mô tả một tình huống sử dụng cụ thể của hệ thống.
* A (Sơ đồ lớp): Không liên quan.
* C (Luồng dữ liệu): Liên quan đến sơ đồ DFD.
* D (Thiết kế giao diện): Không phải mục đích của Scenario.

Câu 7: Quan hệ nào biểu diễn một lớp chứa lớp khác nhưng lớp con vẫn có thể tồn tại độc lập?

Đáp án đúng: A. Aggregation

Giải thích:

* Aggregation (Tổng hợp): Một lớp có thể chứa lớp khác, nhưng lớp con vẫn có thể tồn tại độc lập.
* B (Composition): Lớp con không thể tồn tại nếu lớp cha bị xóa.
* C (Association): Chỉ là quan hệ liên kết giữa các lớp, không ám chỉ chứa đựng.
* D (Inheritance): Không phải quan hệ chứa đựng.

Câu 8: Sơ đồ tuần tự mô tả điều gì?

Đáp án đúng: B. Thứ tự các thông điệp được trao đổi giữa các đối tượng

Giải thích:

* Sơ đồ tuần tự (Sequence Diagram) hiển thị thứ tự các thông điệp được gửi giữa các đối tượng theo thời gian.
* A (Quan hệ giữa các lớp): Thuộc sơ đồ lớp.
* C (Cấu trúc dữ liệu): Không liên quan.
* D (Chức năng hệ thống): Không phải mục tiêu của sơ đồ tuần tự.

Câu 9: Lớp điều khiển trong mô hình MVC tương ứng với thành phần nào?

Đáp án đúng: C. Control

Giải thích:

* Control: Quản lý luồng công việc và logic xử lý.
* A (Model): Xử lý dữ liệu.
* B (View): Hiển thị giao diện.
* D (Entity): Không có trong MVC.

Câu 10: Để biểu diễn quan hệ giữa các lớp, ta sử dụng sơ đồ nào?

Đáp án đúng: C. Sơ đồ lớp

Giải thích:

* Sơ đồ lớp (Class Diagram) giúp biểu diễn quan hệ giữa các lớp như kế thừa, tổng hợp, liên kết.
* A (Sơ đồ tuần tự): Hiển thị thứ tự các thông điệp.
* B (Sơ đồ cộng tác): Biểu diễn quan hệ giao tiếp, không tập trung vào quan hệ lớp.
* D (Sơ đồ use case): Biểu diễn tương tác giữa người dùng và hệ thống.

Part 2: Câu hỏi ngắn:

1. Lớp thực thể là gì?

Trả lời:

* Lớp thực thể (Entity Class) là lớp chứa dữ liệu và trạng thái của hệ thống.
* Nó thường được ánh xạ với các bảng trong cơ sở dữ liệu.

Ví dụ:

* Trong hệ thống quản lý sinh viên, lớp SinhVien có các thuộc tính như Mã sinh viên, Họ tên, Ngày sinh.

2. Lớp điều khiển có vai trò gì trong hệ thống?

Trả lời:

* Lớp điều khiển (Control Class) có nhiệm vụ quản lý luồng xử lý và điều hướng dữ liệu giữa lớp giao diện và lớp thực thể.
* Nó không lưu trữ dữ liệu mà chỉ xử lý logic nghiệp vụ.

Ví dụ:

* Trong hệ thống đặt hàng, lớp OrderController sẽ xử lý logic khi khách hàng đặt hàng.

3. Scenario là gì?

Trả lời:

* Scenario là một kịch bản mô tả cách hệ thống và người dùng tương tác trong một trường hợp cụ thể.
* Nó thể hiện một chuỗi hành động theo thời gian.

Ví dụ:

* Scenario cho use case "Đăng nhập":
  1. Người dùng nhập tên đăng nhập và mật khẩu.
  2. Hệ thống kiểm tra thông tin.
  3. Nếu đúng → Truy cập thành công.
  4. Nếu sai → Hiển thị thông báo lỗi.

4. Quan hệ Include giữa các use case là gì?

Trả lời:

* Quan hệ Include trong Use Case biểu diễn một use case phụ thuộc vào một use case khác để hoàn thành chức năng.

Ví dụ:

* Use case "Thanh toán" Include "Xác thực thẻ tín dụng", vì hệ thống cần xác thực thẻ trước khi thực hiện thanh toán.

5. Mục đích của sơ đồ lớp là gì?

Trả lời:

* Sơ đồ lớp (Class Diagram) dùng để biểu diễn cấu trúc hệ thống, gồm các lớp, thuộc tính, phương thức và quan hệ giữa chúng.

Ví dụ:

* Trong hệ thống quản lý thư viện, sơ đồ lớp có:
  + Lớp "Sách" (thuộc tính: Tên sách, Tác giả, ISBN).
  + Lớp "Thành viên" (thuộc tính: Họ tên, Mã thành viên).
  + Mối quan hệ: Một Thành viên có thể mượn nhiều Sách.

6. Quan hệ Aggregation khác gì so với Composition?

Trả lời:

* Aggregation (Tổng hợp): Một lớp chứa lớp khác, nhưng lớp con vẫn có thể tồn tại độc lập.
* Composition (Thành phần): Một lớp chứa lớp khác, nhưng lớp con không thể tồn tại nếu lớp cha bị xóa.

Ví dụ:

* Aggregation: Một Lớp "Giảng viên" có thể dạy nhiều Môn học, nhưng nếu Giảng viên rời khỏi trường, các môn học vẫn tồn tại.
* Composition: Một Lớp "Ô tô" chứa Lớp "Động cơ", nếu xe bị hủy, động cơ cũng bị hủy theo.

7. Sơ đồ tuần tự là gì?

Trả lời:

* Sơ đồ tuần tự (Sequence Diagram) thể hiện thứ tự các thông điệp được trao đổi giữa các đối tượng theo thời gian.

Ví dụ:

* Trong hệ thống đăng nhập:
  1. Người dùng → Gửi yêu cầu đăng nhập.
  2. Hệ thống kiểm tra thông tin.
  3. Hệ thống phản hồi kết quả.

8. Quan hệ Extend giữa các use case là gì?

Trả lời:

* Quan hệ Extend thể hiện một use case mở rộng chức năng của một use case khác.

Ví dụ:

* Use case "Thanh toán" có thể được Extend bởi "Thanh toán bằng ví điện tử" khi người dùng chọn phương thức này.

9. Lớp biên có vai trò gì trong hệ thống?

Trả lời:

* Lớp biên (Boundary Class) giúp giao tiếp giữa hệ thống và người dùng hoặc hệ thống khác.
* Nó thường là các giao diện người dùng hoặc API kết nối.

Ví dụ:

* Lớp "Giao diện Đăng nhập" nhận thông tin từ người dùng và gửi dữ liệu đến lớp điều khiển.

10. Sơ đồ cộng tác là gì?

Trả lời:

* Sơ đồ cộng tác (Collaboration Diagram) mô tả sự tương tác giữa các đối tượng trong hệ thống.
* Nó thể hiện các đối tượng tham gia và cách chúng gửi thông điệp cho nhau.

Ví dụ:

* Trong hệ thống thương mại điện tử, sơ đồ cộng tác thể hiện cách "Khách hàng" gửi đơn hàng đến "Hệ thống", rồi "Hệ thống" gửi yêu cầu thanh toán đến "Ngân hàng".

Part 3: Câu hỏi thảo luận nhóm

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

Trả lời:

* Lớp thực thể (Entity Class):
  + Chứa dữ liệu và trạng thái của hệ thống.
  + Ví dụ: Lớp SinhVien lưu trữ thông tin sinh viên.
* Lớp biên (Boundary Class):
  + Giao tiếp với người dùng hoặc hệ thống bên ngoài.
  + Ví dụ: Lớp GiaoDienDangNhap nhận thông tin đăng nhập từ người dùng.
* Lớp điều khiển (Control Class):
  + Quản lý luồng xử lý và điều hướng dữ liệu giữa lớp giao diện và lớp thực thể.
  + Ví dụ: Lớp OrderController xử lý đơn hàng.

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

|  |  |  |
| --- | --- | --- |
| Đặc điểm | Aggregation (Tổng hợp) | Composition (Thành phần) |
| Mối quan hệ | Một lớp chứa lớp khác, nhưng lớp con có thể tồn tại độc lập | Một lớp chứa lớp khác, nhưng lớp con không thể tồn tại nếu lớp cha bị xóa |
| Ví dụ | Lớp "Giảng viên" có thể giảng dạy nhiều "Môn học", nhưng nếu giảng viên nghỉ dạy, môn học vẫn tồn tại | Lớp "Ô tô" chứa "Động cơ", nếu ô tô bị hủy, động cơ cũng bị hủy theo |

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

* Sơ đồ lớp mô tả cấu trúc hệ thống, giúp phát triển phần mềm dễ dàng hơn.
* Nó giúp xác định các lớp, thuộc tính, phương thức và quan hệ giữa chúng.
* Là công cụ quan trọng trong giai đoạn thiết kế và phát triển hệ thống.

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

|  |  |  |
| --- | --- | --- |
| Đặc điểm | Sơ đồ tuần tự (Sequence Diagram) | Sơ đồ cộng tác (Collaboration Diagram) |
| Mục đích | Mô tả thứ tự các thông điệp được gửi giữa các đối tượng | Mô tả quan hệ giữa các đối tượng và cách chúng tương tác |
| Cấu trúc | Dựa trên dòng thời gian, các thông điệp được gửi theo thứ tự | Dựa trên mối quan hệ giữa các đối tượng, không tập trung vào thời gian |
| Ví dụ | Hệ thống đăng nhập: Người dùng gửi yêu cầu → Hệ thống kiểm tra → Trả về kết quả | Hệ thống đặt hàng: Khách hàng tương tác với giỏ hàng, giỏ hàng gửi dữ liệu đến hệ thống thanh toán |

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

* Lớp điều khiển (Control Class) chịu trách nhiệm xử lý logic nghiệp vụ.
* Nó nhận dữ liệu từ giao diện (View), xử lý và cập nhật dữ liệu lên Model.

Ví dụ trong hệ thống MVC:

* Người dùng nhập dữ liệu vào form đăng nhập (View).
* Controller kiểm tra thông tin, nếu đúng sẽ cấp quyền truy cập.
* Kết quả sẽ được cập nhật trên Model và hiển thị trên View.

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

* Scenario giúp mô tả cách hệ thống hoạt động trong từng tình huống thực tế.
* Giúp phát hiện lỗi tiềm ẩn và đảm bảo yêu cầu hệ thống đầy đủ.

Ví dụ: Scenario cho use case "Đăng nhập"

* Người dùng nhập thông tin đăng nhập.
* Hệ thống kiểm tra dữ liệu.
* Nếu đúng, hiển thị trang chủ; nếu sai, báo lỗi.

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

* Thu thập yêu cầu từ khách hàng & người dung
* Sử dụng kỹ thuật phỏng vấn, khảo sát và phân tích quy trình.
* Xác định đầy đủ các tác nhân (actors) và các kịch bản (scenarios).
* Xem xét các tình huống ngoại lệ (exception cases) để tránh bỏ sót.

Ví dụ:

* Khi phân tích hệ thống **bán vé online**, cần xem xét cả trường hợp **thanh toán thất bại**.

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

* **Use case:** Mô tả **một chức năng chính của hệ thống**.
* **Scenario:** Mô tả **tình huống cụ thể** của một use case.

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

Ưu điểm:

* Dễ hiểu, trực quan khi mô tả cách hệ thống hoạt động.
* Giúp kiểm tra thứ tự thực thi của hệ thống.
* Dễ dàng kiểm tra lỗi logic và đảm bảo đúng yêu cầu.

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

* Viết kịch bản đầy đủ, rõ ràng và có chi tiết về dữ liệu đầu vào, đầu ra.
* Xác định các tình huống bình thường & tình huống ngoại lệ
* Kiểm tra lại với người dùng để đảm bảo không thiếu sót

Part 4: 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?

Trả lời:

* Xác nhận lại yêu cầu mới với khách hàng để đảm bảo rằng nó thực sự cần thiết.
* Xem xét ảnh hưởng của yêu cầu mới đối với các phần hiện có của hệ thống.
* Cập nhật lại các scenario và sơ đồ liên quan như use case, sơ đồ lớp nếu cần.
* Đánh giá rủi ro và tác động đến tiến độ dự án.

Ví dụ:

* Nếu khách hàng muốn thêm tính năng "Gửi email nhắc hạn trả sách", nhóm phát triển cần cập nhật lại use case "Mượn sách" và thêm lớp mới trong sơ đồ lớp.

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:

* Xác định các luồng nghiệp vụ chính, từ đó tìm ra các chức năng cần điều khiển.
* Dựa vào sơ đồ use case để xác định những phần cần xử lý logic.
* Tạo một lớp điều khiển cho mỗi nhóm chức năng liên quan để tránh quá tải cho một lớp duy nhất.
* Tránh việc lớp điều khiển xử lý quá nhiều logic, hãy tách nhỏ nếu cần.

Ví dụ:

* Trong hệ thống đặt hàng, OrderController sẽ xử lý logic cho tạo đơn hàng, tính tổng giá, gửi xác nhận.

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:

* Xác định chức năng mới ảnh hưởng đến lớp nào trong sơ đồ lớp.
* Bổ sung hoặc sửa đổi các lớp hiện có để hỗ trợ chức năng mới.
* Cập nhật quan hệ giữa các lớp nếu có sự thay đổi trong cách tương tác.
* Kiểm tra tác động đến các mô-đun khác để tránh xung đột.

Ví dụ:

* Nếu khách hàng muốn thêm "Chức năng đánh giá sách", có thể cần thêm lớp Review liên kết với lớp Sách.

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:

* Xác định và phân loại các trường hợp ngoại lệ (ví dụ: Nhập sai thông tin, khóa học đã đầy, lỗi kết nối).
* Viết các scenario riêng biệt cho từng ngoại lệ quan trọng.
* Làm việc với khách hàng để hiểu cách xử lý khi xảy ra ngoại lệ.
* Cập nhật sơ đồ use case nếu cần để phản ánh đúng tình huống thực tế.

Ví dụ:

* Scenario ngoại lệ: Nếu học viên nhập sai mã khóa học, hệ thống phải hiển thị lỗi và cho phép nhập lại.

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:

* Kiểm tra lại sơ đồ lớp để xem đối tượng đó có thực sự cần thiết hay không.
* Xác định lại trách nhiệm của từng đối tượng để đảm bảo vai trò rõ ràng.
* Loại bỏ các đối tượng không cần thiết hoặc điều chỉnh để phù hợp với thực tế.
* Tái cấu trúc sơ đồ tuần tự nếu cần để phản ánh đúng luồng xử lý.

Ví dụ:

* Nếu trong sơ đồ tuần tự của Quản lý mượn sách, có một đối tượng "LibrarianHelper" không thực hiện hành động nào, nó có thể bị loại bỏ.

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:

* Rà soát lại sơ đồ lớp để xác định lỗi trong các mối quan hệ (kế thừa, liên kết, tổng hợp, thành phần).
* Kiểm tra yêu cầu hệ thống để đảm bảo rằng sơ đồ lớp phản ánh đúng yêu cầu.
* Sửa đổi hoặc thay đổi kiểu quan hệ giữa các lớp để phù hợp với thực tế.

Ví dụ:

* Nếu sơ đồ lớp hiện tại cho rằng "Sinh viên" kế thừa từ "Giảng viên", nhưng thực tế hai đối tượng này không có quan hệ kế thừa, cần sửa thành mối quan hệ Association.

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 quan hệ Include và Extend để làm rõ mối quan hệ giữa các use case.
* Xác định các hành động chung mà nhiều use case có thể sử dụng lại.
* Vẽ sơ đồ use case để minh họa mối quan hệ giữa các chức năng.

Ví dụ:

* Trong hệ thống đặt hàng, "Thanh toán" có thể Include "Xác thực thẻ tín dụng", trong khi "Hủy đơn hàng" có thể Extend "Xem đơn hàng".

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) giúp hệ thống giao tiếp với người dùng hoặc hệ thống bên ngoài.
* Trong hệ thống quản lý bán hàng, có thể có các lớp biên như:
  + "GiaoDienBanHang" (giao diện cho nhân viên bán hàng).
  + "APIThanhToan" (tương tác với hệ thống thanh toán online).
  + "GiaoDienQuanLyDonHang" (giao diện cho quản lý theo dõi đơn hàng).

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:

* Thảo luận với khách hàng để hiểu rõ yêu cầu mới.
* Phân tích tác động của yêu cầu mới đến hệ thống hiện tại.
* Cập nhật scenario để phản ánh sự thay đổi.
* Chỉnh sửa sơ đồ lớp, sơ đồ use case và các tài liệu liên quan.
* Đánh giá rủi ro và thời gian thực hiện.

Ví dụ:

* Nếu khách hàng yêu cầu thêm "Chức năng hoàn tiền" sau khi hệ thống đã xong, cần cập nhật use case Thanh toán để phản ánh thay đổi này.

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:

* Rà soát lại yêu cầu hệ thống để xác định lại vai trò của từng đối tượng.
* Kiểm tra sơ đồ lớp và sơ đồ use case để xem cách các đối tượng liên kết với nhau.
* Chỉnh sửa sơ đồ cộng tác để phản ánh đúng luồng tương tác.
* Loại bỏ các đối tượng không cần thiết hoặc thêm đối tượng bị thiếu.

Ví dụ:

* Nếu trong hệ thống ATM, "Khách hàng" không gửi yêu cầu rút tiền trực tiếp đến "Máy ATM", có thể phải sửa lại luồng tương tác.