PHƯƠNG PHÁP PHÁT TRIỂN PHẦN MỀM

**Question 1**

**ICONIX Process tập trung vao giai đoạn nao trong vòng đoi phat triển phần mềm?**

\*A. Phân tích và thiết kế

B. Lập kế hoạch dự án

C. Quản lý dự án

D. Kiểm thử phần mềm

**Question 2**

**Quy trình ICONIX giúp thu hẹp khoảng cách giữa lý thuyết và thực tế như thế nào?**

A.Chuyển đổi nhanh từ yêu cầu thành mã nguồn mà không qua phân tích

B. Tập trung vào thiết kế trực tiếp từ mã nguồn

\*C. Liên kết cac use case với các đối tượng trong hệ thống

D. Phát triển hệ thống dựa trên mô hình UML đầy đủ

**Question 3**

**Yêu cầu chức năng trả lời cho câu hỏi nào?**

A. Ai sẽ là người sử dụng hệ thống?

B. Hệ thống được triển khai như thế nào?

\*C. Hệ thống có thể làm gì?

D. Làm thế nào để thiết kế hệ thống?

**Question 4**

**Quy trình ICONIX cố gắng giải quyết vấn đề gì trước đây?**

\*A. Cồng kềnh và nhiều thứ không cần thiết trong dự án

thực tế

B. Thiếu tài liệu mô tả hệ thống

C. Thiếu mô tả chức năng của mã nguồn

D. Thiếu sự chi tiết trong các mô hình

**Question 5**

**Sự khác biệt chính giữa yêu cầu chức năng và yêu cầu hành vi trong quy trình ICONIX là gì?**

A. Yêu cầu chức năng mô tả chi tiết các lớp đối tượng,còn yêu cầu hành vi mô tả dữ liệu hệ thống

B. Cá hai đều là các bước đầu tiên của qua trình phân

tích hệ thống

C. Yêu cầu chức năng miêu tả cách hệ thống hoạt

động, còn yêu cầu hành vi miêu tả giao diện người

dùng

\*D. Yêu cầu chức năng mô ta những gì hệ thống cần

làm, còn yêu cầu hành vi mô tả cách hệ thống và

người dùng tương tác

**Question 6**

**Trong quy trình ICONIX, hoạt động nao giup làm ro và loại bỏ sự mơ hồ trong yêu cầu hệ thống?**

A. Vẽ sơ đồ lớp

B. Kiểm thử đơn vị

\*C. Phân tích độ bền (robustness analysis)

D. Viết mã nguồn

**Question 7**

**Trong quy trình ICONIX, mô hình miền có vai trò gì?**

\*A. Định nghĩa phạm vi và tạo nền tảng cho việc xây

dựng use case

B. Giúp xác định các yêu cầu không chức năng

C. Định nghĩa cách hệ thống sẽ triển khai

D. Đảm bảo tính khả dụng của hệ thống

**Question 8**

**Việc sử dụng sơ đồ trình tự (sequence diagram) trong quy trình ICONIX nhằm mục đích gì?**

A. Để phân tích tính bảo mật của hệ thống

\*B. Để phân bổ hành vi cho các lớp đối tượng

C. Để phân tích tính khả thi của dự án

D. Để xác định các yêu cầu không chức năng

**Question 9**

**Trong quy trình ICONIX, tại sao cần thực hiện đánh giá thiết kế sơ bộ (Preliminary Design Review)?**

A. Để xác định tất ca các lỗi trong mã nguồn

\*B. Để đảm bảo mô hình miền và use case khớp nhau

C. Để kiểm tra tính chính xác cua mô hình dữ liệu

D. Để kiểm thử toàn bộ hệ thống

**Question 10**

**Làm thế nào để một sơ đồ độ bền giúp liên kết use case với mô hình đối tượng?**

A. Xác định các lỗi tiềm ẩn trong thiết kế

B. Liệt kê tất ca các yêu cầu chức năng

\*C.Xác định các đối tượng miền và các thông điệp truyền giữa chúng

D. Tạo ra các đối tượng giao diện (GUI)

**Question 11**

**Khi cần kiểm tra việc tích hợp hệ thống dựa trên các use case trong ICONIX Process, ta nên bat đầu từ đâu?**

A. Kiểm tra cơ sở dữ liệu

B. Kiểm tra giao diện người dùng

C. Viết mã nguồn cho tất cả các chức năng

\*D. Kiểm tra kịch bản chính và các kịch bản thay thế

**Question 12**

**Sau khi hoàn thành giai đoạn phân tích và thiết kế sơ bộ, nếu phát hiện sự khác biệt giữa use case và sơ đồ lớp, ta sẽ xử lý như thế nào?**

A. Kiểm tra mã nguồn

B. Bó qua sự khác biệt

C. Thực hiện kiểm thử đơn vị

\*D. Cập nhật sơ đồ lớp và use case để đảm bảo chúng

khớp với nhau

**Question 13**

**Khi thực hiện sơ đồ trình tự, điều gì cần được xem xét đầu tiên để đảm bảo thiết kế đúng?**

A. Các bảng cơ sở dữ liệu

B. Các yêu cầu phi chức năng

C. Sơ đồ lớp

\*D. Các đối tượng biên và thực thể từ sơ đồ độ bền

**Question 14**

**Tại sao quy trình ICONIX được coi là phù hợp với các dự án theo phương pháp phát triển Agile?**

A. Nó tập trung vao việc phát triển toàn bộ hệ thống ngay từ đầu

B. Nó yêu cầu một kế hoạch chi tiết và nghiêm ngặt từ đầu đến cuối

\*C. Nó cho phép lặp lại và phản hồi nhanh chóng dựa trên từng use case

D. Nó không cần mô hình hóa các yêu cầu

**Question 15**

**Trong quá trình kiểm thử đơn vị, các bài kiểm thử được viết dựa trên:**

A. Kịch bản chính từ sơ đồ trình tự

B. Sơ đồ lớp

\*C. Use case và các controllers từ sơ đồ độ bền

D. Mô hình miền

**Question 16**

**Quy trình ICONIX bao gồm bao nhiêu sơ đồ UML cốt lõi?**

A. 2

B. 6

\*C. 4

D. 14

**Question 17**

**Câu hỏi chính mà quy trình ICONIX muốn trả lời là gì?**

\*A. Làm the nao đe có the đi từ use case đến ma nguồn

B. Làm thế nao để lập trình hướng đối tượng

C. Làm thế nào để đi từ người dùng đến sản phẩm

D. Làm the nao để sử dụng được hết các loại sơ đồ UML

**Question 18**

**Đối với quy trình ICONIX, bước nào là quan trọng nhất để đảm bảo yêu cầu và thiết kế đồng nhất với nhau?**

A. Kiểm thử tích hợp

B. Kiểm thử đơn vị

C. Viết mã nguồn

\*D. Đánh giá thiết kế sơ bộ (PDR)

**Question 19**

**Trong quy trình ICONIX, các yêu cầu hành vi được mô tả như thế nào?**

A. Thông qua mô hình lớp

B. Thông qua các bảng cơ sở dữ liệu

C. Thông qua mô hình dữ liệu

\*D. Thông qua use case mô tả cách người dùng và hệ thống tương tác

**Question 20**

**Làm the nao để một sơ đồ độ bền giup liên kết use case với mô hình đối tượng?**

A. Xác định các lỗi tiềm ẩn trong thiết kế

B. Liệt kê tất cả các yêu cầu chức năng

C. Tạo ra các đối tượng giao diện (GUI)

\*D.Xác định các đối tượng miền và các thông điệp truyền giữa chúng

**Question 21**

**Việc điều chỉnh mô hình miền trong ICONIX xảy ra khi nào?**

\*A. Khi phat hien cac đoi tượng mới trong qua trình thiết kế

B. Sau khi kiểm thử tích hợp

C. Trong quá trình kiểm thử đơn vị

D. Chỉ sau khi viết mã

**Question 22**

**Khi nao ta cần thực hiện đanh gia thiết kế chi tiết (Critical Design Review) trong quy trình ICONIX ?**

A. Sau khi kiểm thử đơn vị

B. Trước khi thiết kế sơ đồ lớp

\*C. Sau khi hoan thành sơ đồ trình tự và cập nhật mô hình tĩnh

D. Sau khi viết mã nguồn

**Question 23**

**ICONIX Process là phương pháp tiếp cận nào?**

Phương pháp tuần tự

\*B. Phương pháp tối giản nhưng hoạt động

Phương pháp xoắn ốc

Phương pháp nhanh chóng

**Question 24**

**Khi cần liên kết giua các use case và các đối tượng trong hệ thống, ta sử dụng sơ đồ nào trong quy trình ICONIX?**

A. Sơ đồ use case

\*B. Sơ đồ độ bền (robustness diagram)

C. Sơ đồ trình tự (sequence diagram)

D. Sơ đồ miền (domain model)

**Question 25**

**Ta cần làm gì để đảm bảo rằng quá trình phân tích trong quy trình ICONIX tạo ra thiết kế có thể thực hiện được?**

A. Viết 1mã nguồn ngay lập tức

B. Phát triển sơ đồ lớp

C. Thực hiện phân tích độ bền (robustness analysis) để xác định các đối tượng cần thiết

D. Kiểm thử toàn bộ hệ thống

**Question 26**

**Trong qua trình phát triển phần mềm, nếu có sự thay đổi về yêu cầu, bước nào trong ICONIX Process có thể giúp thích ứng với thay đổi đó?**

A. Cập nhật sơ đồ trình tự và thiết kế chi tiết

B. Thay đổi mô hình lớp

C. Viết mã nguồn mới

D. Kiểm thử toàn bộ hệ thống

**Question 27**

**Đầu vào của quy trình ICONX là sơ đồ nào?**

A. Sequence diagram

\*B. Use case diagram

C. Robustness diagram

D. Class diagram

**Question 28**

**Thông tin nào trong sơ đồ trình tự được dùng để bổ sung vào sơ đồ lớp**

\*A. Thông điệp

B. Đối tượng biên

C. Tác nhân

D. Đối tượng thực thể

**Question 29**

**Mục tiêu chính của buổi đánh giá thiết kế quan trọng (CDR) là gì?**

A. Đánh giá các yêu cầu hệ thốn

B. Xác định các sơ đồ lớp

C. Kiểm thử mã nguồ

\*D. Kiểm tra tính nhất quán giữa thiết kế và yêu cầu

**Bài 2**

**Question 1**

**Mô hình miền (domain model) trong quy trình ICONIX là gì?**

A. Là một bản thiết kế cơ sở dữ liệu

\*B. Là “từ điển” thuật ngữ trong dự án

C. Là sơ đồ quy trình nghiệp vụ

D. Là một bảng dữ liệu quan hệ

**Question 2**

**Mô hình miền khác gì so với bảng thuật ngữ dự án (project glossary)?**

A. Mô hình miền chỉ chứa thông tin về các lớp (classes) trong dự án.

B. Mô hình miền chỉ sử dụng trong giai đoạn kiểm thử dự án.

\*C. Mô hình miền là một biểu đồ được minh họa, hiển thị mối quan hệ giữa các thuật ngữ.

D. Mô hình miền không có mối quan hệ với bảng thuật ngữ dự án.

**Question 3**

**Mô hình miền là cơ sở đầu tiên cho việc xây dựng sơ đồ nào trong quy trình ICONIX?**

A. Sơ đồ mạnh mẽ

\*B. Sơ đồ use case

C. Sơ đồ hoạt động

D. Sơ đồ trình tự

**Question 4**

**Điều nào sau đây không nên thực hiện khi xây dựng mô hình miền?**

\*A. Đưa các lớp giao diện người dùng vào mô hình miền.

B. Sắp xếp các lớp dựa trên các khái niệm chính trong miền vấn đề.

C. Sử dụng mô hình miền như một bảng thuật ngữ của dự án.

D. Giới hạn thời gian tạo mô hình miền ban đầu trong vài giờ.

**Question 5**

**Tại sao nên bắt đầu với mô hình miền (domain model) trước khi viết use case?**

A. Vì use case không liên quan đến mô hình miền.

B. Vì use case nên được viết trước để định nghĩa chi tiết.

\*C. Vì nó giúp ràng buộc use case vào thực tế và hệ thống sẽ được thiết kế.

D. Vì mô hình miền phức tạp hơn và nên được thực hiện trước.

**Question 6**

**Điều nào sau đây không nên thực hiện khi xây dựng mô hình miền?**

\*A. Sử dụng quan hệ tổng quát hóa (generalization) để đơn giản hóa mối quan hệ.

B. Thêm nhiều lớp hơn để giải thích.

C. Bỏ bớt các lớp.

D. Giữ nguyên và tiếp tục.

**Question 7**

**Nếu phát hiện mô hình miền thiếu một số đối tượng quan trọng, ta nên làm gì?**

A. Tạo một sơ đồ mới.

\*B. Cập nhật mô hình miền để bao gồm các đối tượng bị thiếu.

C. Bỏ qua những đối tượng này.

D. Xây dựng lại mô hình từ đầu.

**Question 8**

**Mối quan hệ nào dưới đây là ví dụ về quan hệ "is-a"?**

A. Mục Đơn Hàng thuộc Giỏ Hàng.

B. Quyển Sách có Mục Lục.

\*C. Con Mèo là một Loài Động Vật.

D. Người Dùng có Tài Khoản.

**Question 9**

**Quan hệ "has-a" trong mô hình miền là loại quan hệ gì?**

A. Quan hệ một-nhiều.

\*B. Tập hợp (aggregation).

C. Tổng quát hóa.

D. Kế thừa.

**Question 10**

**Vai trò của mô hình miền trong quá trình phát triển hệ thống là gì?**

\*A. Để đảm bảo giao tiếp rõ ràng, nhất quán giữa các thành viên dự án.

B. Để mô hình hóa cơ sở dữ liệu.

C. Để tối ưu hóa mã nguồn.

D. Để kiểm thử hệ thống.

**Question 11**

**Nếu phát hiện hai thuật ngữ "Book" và "Books" trong mô hình miền, bước tiếp theo bạn nên làm gì?**

\*A. Xác định và loại bỏ sự trùng lặp, giữ lại một trong hai thuật ngữ.

B. Giữ lại cả hai thuật ngữ.

C. Tạo lớp mới cho mỗi thuật ngữ.

D. Thêm thuộc tính cho cả hai.

**Question 12**

**Quá trình nào giúp phát hiện các đối tượng bị thiếu trong mô hình miền sau khi phiên tạo mô hình ban đầu hoàn thành?**

A. Triển khai hệ thống.

B. Phân tích mã nguồn.

\*C. Phân tích use case và sơ đồ độ bền (robustness diagrams).

D. Kiểm thử hệ thống.

**Question 13**

**Lợi ích của việc tổ chức mô hình miền dựa trên các trừu tượng trong thế giới thực là gì?**

A. Giúp giảm bớt tài liệu yêu cầu.

\*B. Giúp mô hình bền vững hơn trước những thay đổi của yêu cầu phần mềm.

C. Giúp tăng tốc độ triển khai hệ thống.

D. Làm cho mô hình khó thay đổi hơn khi yêu cầu thay đổi.

**Question 14**

**Mục tiêu chính của mô hình miền là gì?**

\*A. Giúp tạo ra một ngôn ngữ chung để truyền đạt thông tin rõ ràng giữa các thành viên dự án

B. Để xác định các yêu cầu phi chức năng

C. Để thiết kế cơ sở dữ liệu

D. Để xây dựng mô hình dữ liệu chính xác

**Question 15**

**Mô hình miền đại diện cho phần nào của hệ thống?**

A. Mô tả các trường hợp sử dụng chi tiết.

B. Chỉ mô tả các giao diện người dùng.

\*C. Phần tĩnh mô tả cấu trúc của hệ thống.

D. Phần động mô tả hành vi của hệ thống.

**Question 16**

**Trong mô hình miền, tại sao không nên nhầm lẫn giữa đối tượng và bảng cơ sở dữ liệu?**

\*A. Vì đối tượng đại diện cho một thể hiện duy nhất, trong khi bảng cơ sở dữ liệu đại diện cho một tập hợp.

B. Vì các đối tượng phức tạp hơn bảng dữ liệu

C. Vì bảng cơ sở dữ liệu không có quan hệ với các đối tượng

D. Vì bảng cơ sở dữ liệu chứa nhiều khóa ngoài

**Question 17**

**Tại sao việc xây dựng mô hình miền ban đầu chỉ nên kéo dài khoảng 2 giờ?**

A. Vì thời gian giới hạn của dự án

B. Vì không cần mô hình lớp chi tiết

\*C. Vì mô hình miền sẽ được cập nhật liên tục trong quá trình dự án.

D. Vì chỉ có các yêu cầu chức năng quan trọng

**Question 18**

**Lý do nên bắt đầu với mô hình miền thay vì use case là gì?**

A. Vì use case không cần phải rõ ràng ngay từ đầu

B. Vì mô hình miền dễ thực hiện hơn

\*C. Để đảm bảo use case được viết trong bối cảnh của mô hình đối tượng và tránh sự mơ hồ.

D. Để tiết kiệm thời gian

**Question 19**

**Tại sao khi tạo mô hình miền, cần tập trung vào các đối tượng trong thế giới thực?**

A. Vì các đối tượng trong thế giới thực đơn giản hơn các đối tượng trong phần mềm.

\*B. Vì phần mềm nên được thiết kế sao cho giống với cách thế giới thực hoạt động, và thế giới thực thay đổi ít thường xuyên hơn so với yêu cầu phần mềm.

C. Vì nó giúp tránh sử dụng quá nhiều tài nguyên trong quá trình phát triển phần mềm.

D. Vì các đối tượng trong thế giới thực không ảnh hưởng đến kiến trúc phần mềm.

**Question 20**

**Mối quan hệ nào sau đây là đúng khi nói về các mối quan hệ trong mô hình miền?**

A. Mối quan hệ giữa các lớp không quan trọng trong mô hình miền.

\*B. Mối quan hệ tổng hợp (has-a) thể hiện một đối tượng sở hữu một đối tượng khác và mối quan hệ tổng quát hóa (is-a) thể hiện sự kế thừa giữa các lớp.

C. Mối quan hệ tổng quát hóa (is-a) chỉ thể hiện một lớp cha có nhiều lớp con.

D. Tất cả các mối quan hệ trong mô hình miền đều là các mối quan hệ tuần tự.

**Question 21**

**Ý nghĩa của các mối quan hệ "has-a" và "is-a" trong mô hình miền là gì?**

A. Chúng không có trong mô hình miền.

B. Chúng chỉ áp dụng cho các lớp giao diện (interfaces).

C. Chúng là các mối quan hệ giữa các thuộc tính của lớp.

.\*D. Chúng mô tả cách các lớp chứa các lớp khác và các lớp được kế thừa.

**Question 22**

**Tại sao không nên kỳ vọng sơ đồ lớp cuối cùng sẽ khớp chính xác với mô hình miền?**

A. Vì sơ đồ lớp không quan trọng

B. Vì mô hình miền không liên quan đến sơ đồ lớp

.\*C. Vì mô hình miền đơn giản hơn và không chứa các chi tiết thiết kế cụ thể.

D. Vì sơ đồ lớp chỉ là mô hình tạm thời

**Question 23**

**Điểm khác biệt chính giữa một đối tượng và một bảng cơ sở dữ liệu là gì?**

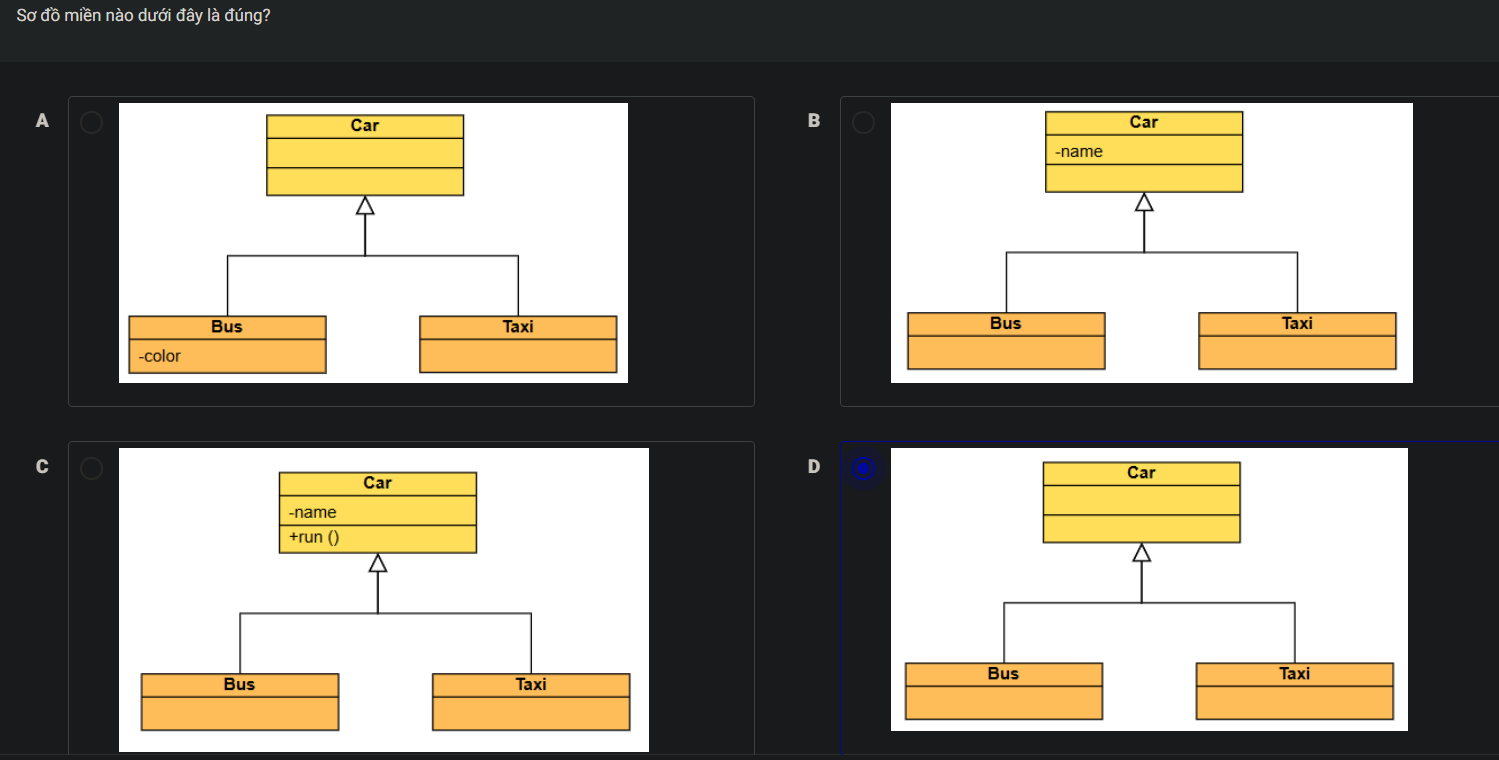
A. Bảng cơ sở dữ liệu không liên quan gì đến đối tượng trong mô hình miền.

B. Một đối tượng đại diện cho một tập hợp các bản ghi, trong khi bảng đại diện cho một bản ghi duy nhất.

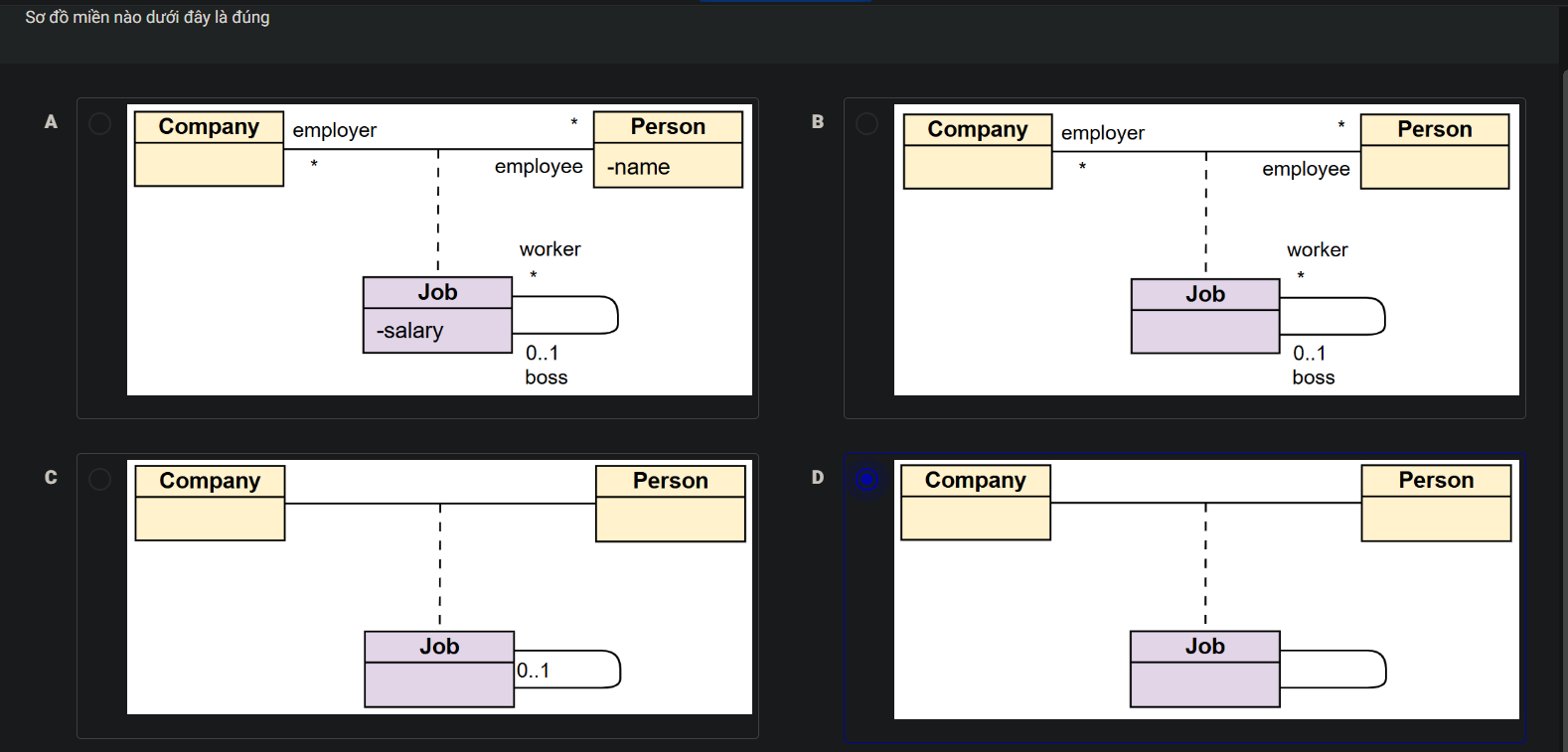
.\*C. Một đối tượng đại diện cho một thể hiện đơn lẻ, trong khi bảng cơ sở dữ liệu đại diện cho một tập hợp các bản ghi

D. Một đối tượng luôn chứa nhiều thuộc tính hơn bảng cơ sở dữ liệu.

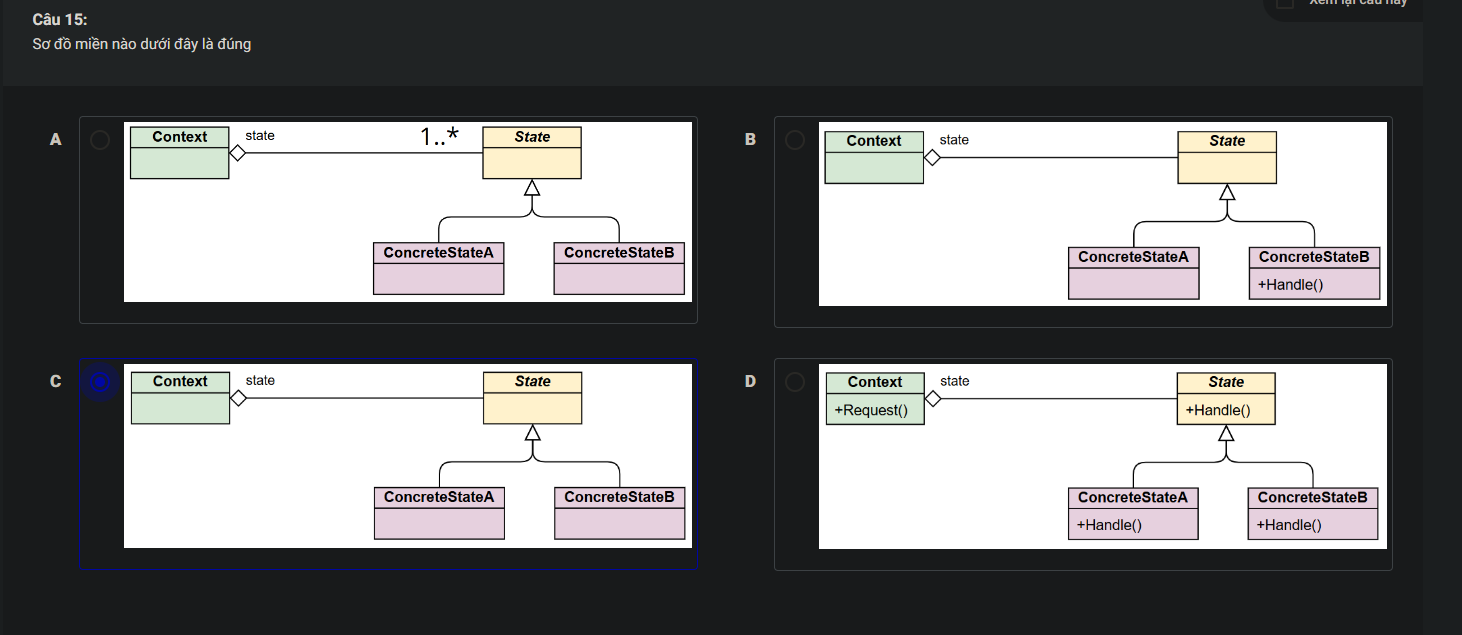
**Question 24:**

****

**Question 25:**

****

**Question 26:**

****

**Question 27**

**Mục tiêu của phiên tạo mô hình miền ban đầu là gì?**

A. Tìm ra các lỗi phần mềm trong hệ thống.

B. Tạo ra mô hình hoàn chỉnh mà không cần sửa đổi sau này.

\*C. Xây dựng một phiên bản sơ bộ để có thể điều chỉnh trong quá trình làm việc.

D. Hoàn thiện mô hình miền ngay từ đầu..

**Question 28**

**Nguyên tắc nào đúng khi nói về quan hệ giữa các đối tượng trong mô hình miền?**

A. Tránh sử dụng bất kỳ quan hệ nào để giữ mô hình đơn giản.

B. Không sử dụng quan hệ tổng quát hóa (is-a) và tổng hợp (has-a).

\*C. Sử dụng quan hệ tổng quát hóa (is-a) và tổng hợp (has-a) để thể hiện cách các đối tượng liên quan với nhau.

D. Chỉ sử dụng các quan hệ tuần tự để kết nối các đối tượng.

**Question 29**

**Mô hình miền được minh họa như thế nào?**

\*A. Như một sơ đồ lớp (class diagram) đơn giản với các đường nối giữa các lớp.

B. Như một sơ đồ trình tự (sequence diagram).

C. Như một sơ đồ hoạt động (activity diagram).

D. Như một sơ đồ triển khai (deployment diagram).

**Question 30**

**Mô hình miền hiển thị loại quan hệ nào giữa các lớp trong miền?**

A. Quan hệ tuần tự (sequential relationships).

B. Quan hệ phụ thuộc (dependency relationships).

\*C. Quan hệ tổng hợp (aggregation) và tổng quát hóa (generalization).

D. Quan hệ đồng thời (concurrent relationships).

**Bài 3**

**Question 1**

**Use case giúp trả lời câu hỏi nào liên quan đến hệ thống?**

A. Cách hệ thống lưu trữ dữ liệu trong cơ sở dữ liệu.

B. Hệ thống sẽ được triển khai như thế nào?

C. Sự phân bổ nguồn lực phát triển trong dự án.

\* D. Người dùng của hệ thống đang cố gắng làm gì và trải nghiệm người dùng ra sao?

**Question 2**

**Tại sao không nên thiết kế, lập trình, hoặc ước tính trực tiếp từ tài liệu yêu cầu chức năng (functional specification)?**

A. Vì tài liệu yêu cầu chức năng chỉ dành cho nhóm quản lý.

\* B. Vì thiết kế trực tiếp từ tài liệu yêu cầu chức năng giống như chơi trò “chọn số ngẫu nhiên”, cần có thêm bước khám phá để làm rõ yêu cầu.

C. Vì tài liệu yêu cầu chức năng thường có nhiều lỗi.

D. Vì tài liệu yêu cầu chức năng không liên quan đến quá trình thiết kế phần mềm.

**Question 3**

**Điều gì làm cho yêu cầu chức năng (functional requirements) trở thành bước đầu tiên trong quá trình thu thập yêu cầu?**

A. Vì yêu cầu chức năng chỉ bao gồm các yêu cầu kỹ thuật.

\* B. Vì yêu cầu chức năng là tập hợp các yêu cầu ban đầu từ các bên liên quan, giúp xác định các yêu cầu hành vi cần làm rõ hơn.

C. Vì yêu cầu chức năng cung cấp một bản thiết kế chi tiết của hệ thống.

D. Vì yêu cầu chức năng thay thế hoàn toàn các trường hợp sử dụng.

**Question 4**

**Yếu tố nào sau đây giúp ích đáng kể trong việc xác định use case?**

A. Phân tích cấu trúc cơ sở dữ liệu.

\* B. Tạo storyboards và nguyên mẫu (UI mockups, demo chức năng cơ bản).

C. Viết tài liệu quản lý dự án.

D. Thực hiện kiểm thử hiệu năng.

**Question 5**

**Khi viết use case, điều gì cần được lưu ý ngoài kịch bản chính (sunny-day scenario)?**

A. Chỉ viết về cách hệ thống hoạt động trong điều kiện tối ưu.

B. Tập trung vào tài liệu kỹ thuật thay vì các tình huống sử dụng thực tế.

C. Chỉ cần tập trung vào kịch bản chính mà không cần các kịch bản khác.

\* D. Phân tích các kịch bản thay thế (rainy-day scenarios) để mô tả cách hệ thống phản ứng khi xảy ra lỗi hoặc khi người dùng sử dụng các tính năng ít phổ biến.

**Question 6**

**Trong ngữ cảnh của use case, “rainy-day scenario” là gì?**

A. Một kịch bản mô tả cách người dùng sử dụng hệ thống trong điều kiện lý tưởng.

B. Một kịch bản mô tả giao diện người dùng.

C. Một kịch bản mô tả các tính năng hiệu năng của hệ thống.

\* D. Một kịch bản mô tả cách hệ thống hoạt động trong các tình huống không mong muốn hoặc khi người dùng sử dụng các tính năng ít phổ biến.

**Question 7**

**Tại sao việc ghi nhận các "rainy-day scenarios" (kịch bản không mong muốn) lại quan trọng khi viết use case?**

\* A. Vì các kịch bản này cung cấp thông tin về cách hệ thống cần phản hồi khi có lỗi hoặc khi người dùng tương tác với các tính năng ít được sử dụng, đảm bảo hệ thống hoạt động toàn diện và ổn định.

B. Vì các kịch bản này không ảnh hưởng đến cách hệ thống hoạt động trong thực tế.

C. Vì các kịch bản này chỉ hữu ích trong giai đoạn kiểm thử hệ thống.

D. Vì các kịch bản này giúp xác định những tính năng không cần thiết trong hệ thống.

**Question 8**

**Tại sao việc viết use case nên tập trung vào sự tương tác giữa người dùng và hệ thống?**

A. Để xác định các lỗi trong hệ thống.

\* B. Để mô tả rõ ràng các hành vi cần thiết mà hệ thống phải thực hiện.

C. Để tạo sơ đồ trạng thái.

D. Để đảm bảo tính an toàn của hệ thống.

**Question 9**

**Tại sao việc viết use case theo cấu trúc "noun-verb-noun" lại quan trọng?**

\*A. Để tạo ra một thiết kế hướng đối tượng dễ dàng từ use case.

B. Để dễ dàng mô hình hóa cơ sở dữ liệu.

C. Để tạo sơ đồ lớp.

D. Để viết mã nguồn trực tiếp.

**Question 10**

**Trong quá trình viết use case, nếu người dùng thực hiện một hành động không dự đoán trước, bạn sẽ làm gì?**

A. Thay đổi mô hình miền.

\* B. Thêm vào "rainy-day scenario" trong use case để mô tả hành vi này.

C. Yêu cầu người dùng không thực hiện hành vi đó.

D. Bỏ qua yêu cầu hành vi này.

**Question 11**

**Nếu phát hiện ra một yêu cầu mới trong quá trình viết use case, bạn sẽ làm gì?**

A. Bỏ qua yêu cầu này.

**\* B. Cập nhật use case để bao gồm yêu cầu mới.**

C. Viết mã nguồn cho yêu cầu này.

D. Thay đổi toàn bộ thiết kế hệ thống.

**Question 12**

**Nếu mô hình miền thay đổi trong quá trình phát triển, điều gì cần thực hiện với use case?**

A. Thay đổi mô hình lớp.

**\* B. Cập nhật use case để phù hợp với mô hình miền mới.**

C. Bỏ qua mô hình miền.

D. Tạo sơ đồ trình tự mới.

**Question 13**

**Tại sao nên xây dựng mô hình miền ban đầu trước khi viết use case?**

A. Để tránh phải sửa đổi mô hình miền sau khi phân tích use case.

B. Để đảm bảo mô hình miền hoàn chỉnh ngay từ đầu.

**\* C. Để thiết lập bối cảnh cho các thuật ngữ được sử dụng trong use case và cung cấp cơ sở cho việc phát triển use case.**

D. Để mô tả chi tiết tất cả các tương tác giữa người dùng và hệ thống ngay từ đầu.

**Question 14**

**Viết một đoạn văn bản mô tả một "sunny-day scenario" cho chức năng Đăng nhập trong hệ thống bán hàng trực tuyến, sử dụng cấu trúc "noun-verb-noun".**

A. Người dùng nhập mật khẩu và đăng nhập thành công.

B. Hệ thống hiển thị màn hình đăng nhập.

\* C. Người dùng nhập tên đăng nhập và mật khẩu, sau đó nhấn nút Đăng nhập. Hệ thống xác thực thông tin và đăng nhập người dùng.

D. Người dùng vào trang đăng nhập, kiểm tra thông tin và đăng nhập thành công.

**Question 15**

**Mô tả một "rainy-day scenario" cho tình huống người dùng nhập sai mật khẩu.**

A. Hệ thống chấp nhận mọi đầu vào và cho phép người dùng đăng nhập.

B. Hệ thống chuyển người dùng đến trang chính.

\* C. Hệ thống hiển thị thông báo lỗi "Mật khẩu không đúng" và yêu cầu người dùng nhập lại mật khẩu.

D. Hệ thống không phản hồi gì và yêu cầu người dùng nhập lại thông tin.

**Question 16**

**Use case là gì trong quy trình ICONIX?**

A. Một báo cáo tài chính liên quan đến chi phí phát triển.

B. Một sơ đồ thể hiện các mối quan hệ giữa các lớp trong hệ thống.

C. Một công cụ kiểm thử phần mềm tự động.

\*D. Một cách có cấu trúc để ghi lại các yêu cầu hành vi của hệ thống.

**Question 17**

**Use case cung cấp điều gì cho quá trình thiết kế hệ thống?**

\*A. Một nền tảng để thiết kế và ước tính thời gian, công sức một cách đáng tin cậy.

B. Một cách để dự đoán lỗi trong phần mềm.

C. Một phương pháp để viết tài liệu dự án theo cách tổng quát.

D. Một danh sách các công cụ cần sử dụng trong dự án.

**Question 18**

**Trong quá trình viết use case, điều gì cần được chú ý nhất?**

A. Xác định yêu cầu phi chức năng.

B. Tạo ra nhiều sơ đồ lớp.

C. Viết mã nguồn trực tiếp từ use case.

\*D. Sử dụng các thuật ngữ rõ ràng và nhất quán với mô hình miền.

**Question 19**

**Khi mô hình miền và use case không khớp nhau, bạn nên làm gì?**

A. Kiểm tra lại mô hình dữ liệu.

\*B. Cập nhật mô hình miền hoặc use case để đảm bảo sự khớp nhau.

C. Làm việc lại với khách hàng để thu thập thêm yêu cầu.

D. Bỏ qua sự không khớp.

**Question 20**

**Khi hệ thống gặp lỗi trong quá trình kiểm thử và không thực hiện đúng theo use case, bước tiếp theo là gì?**

A. Thay đổi yêu cầu ban đầu.

B. Xây dựng lại hệ thống từ đầu.

\*C. Chỉnh sửa mã nguồn để phản ánh đúng hành vi trong use case

D. Tạo thêm các sơ đồ trạng thái.

**Question 21**

**Khi viết use case, tại sao cần tránh sử dụng câu bị động?**

A. Để dễ dàng viết mã nguồn.

B. Để giảm bớt sự phức tạp trong thiết kế.

\*C. Để đảm bảo rõ ràng ai đang thực hiện hành động và hệ thống phản hồi như thế nào.

D. Để mô tả hệ thống theo yêu cầu.

**Question 22**

**Theo quy trình ICONIX, tại sao mô hình miền ban đầu thường được coi là không hoàn chỉnh?**

A. Vì tất cả các yêu cầu đều phải được xác định rõ ràng trước khi bắt đầu xây dựng mô hình miền.

\*B. Vì mô hình miền ban đầu chỉ là một bản sơ bộ và sẽ được cải thiện dần qua quá trình phân tích use case.

C. Vì mô hình miền ban đầu không liên quan đến use case.

D. Vì không thể bao gồm tất cả các chức năng trong mô hình đầu tiên.

**Question 23**

**Hãy tạo một mô hình miền ban đầu cho một hệ thống bán sách trực tuyến. Bạn sẽ chọn những thực thể nào và tại sao?**

A. Chọn các thực thể liên quan đến hệ thống thanh toán và giao hàng, vì đó là phần quan trọng nhất trong hệ thống bán sách.  
\*B**.** Chọn các thực thể như Sách, Tác giả, và Giỏ hàng, vì đây là các thực thể chính của hệ thống bán sách trực tuyến và phù hợp với nhu cầu người dùng.  
C. Chọn các thực thể liên quan đến quản trị hệ thống, vì nó giúp đảm bảo hệ thống vận hành đúng cách.  
D. Chọn tất cả các thực thể có thể xuất hiện trong hệ thống, để không bỏ sót thông tin nào.

**Question 24**

**Trong quá trình viết use case, bạn phát hiện hệ thống cần phản hồi khi người dùng nhập sai mật khẩu. Hãy mô tả một "rainy-day scenario" cho tình huống này.**

\*A. Hệ thống hiển thị thông báo lỗi "Mật khẩu không đúng" và yêu cầu người dùng nhập lại mật khẩu.

B. Hệ thống không phản hồi gì và yêu cầu người dùng nhập lại thông tin.

C. Hệ thống chấp nhận mọi đầu vào và cho phép người dùng đăng nhập.

D. Hệ thống chuyển người dùng đến trang chính.

Dưới đây là câu hỏi và đáp án theo định dạng bạn đã yêu cầu:

**Question 25**

**Mục đích chính của use case là gì?**

A. Thể hiện cấu trúc dữ liệu của hệ thống.

\*B. Giúp nắm bắt các yêu cầu hành vi của hệ thống để tạo ra thiết kế.

C. Tìm ra cấu trúc tĩnh của hệ thống.

D. Tạo ra một danh sách các lỗi tiềm ẩn trong hệ thống.

**Question 26**

**Mục tiêu của use case trong quy trình hướng đối tượng là gì?**

A. Để tạo một tài liệu mô tả toàn bộ quá trình phát triển.

\*B. Là bước đầu tiên trong việc thiết kế hướng đối tượng và để nhanh chóng đạt được mã chất lượng cao.

C. Để xác định chi tiết các cấu trúc dữ liệu trong cơ sở dữ liệu.

D. Để làm tài liệu trừu tượng về yêu cầu.

**Question 27**

**Khi phát hiện một đối tượng không được mô tả trong use case nhưng có liên quan đến hệ thống, bạn sẽ làm gì?**

A. Bỏ qua đối tượng này.

B. Xóa bỏ đối tượng.

C. Chỉ thay đổi mô hình miền.

\*D. Thêm đối tượng vào use case và mô hình miền.

**Question 28**

**Khi xây dựng use case, việc tổ chức use case với các tác nhân (actors) có mục đích gì?**

A. Để đảm bảo tính toàn vẹn của hệ thống.

\*B. Để làm rõ vai trò của từng người dùng và chức năng tương ứng họ thực hiện.

C. Để tối ưu hóa mã nguồn.

D. Để phân tích mô hình dữ liệu.

**Question 29**

**Vai trò của use case trong việc bổ sung cho yêu cầu chức năng là gì?**

A. Để thay thế hoàn toàn tài liệu yêu cầu chức năng.

\*B. Để cung cấp một phương pháp có cấu trúc giúp ghi lại các yêu cầu hành vi, hỗ trợ quá trình thiết kế.

C. Để lập danh sách các lỗi tiềm ẩn trong hệ thống.

D. Để xác định các trường hợp thử nghiệm phần mềm.

**Question 30**

**Điều gì xảy ra với mô hình miền khi các use case được phát triển?**

A. Use case sẽ thay thế hoàn toàn mô hình miền.

B. Mô hình miền sẽ giữ nguyên và không thay đổi.

C. Mô hình miền sẽ chỉ được sử dụng lại khi hệ thống hoàn thành.

\*D. Mô hình miền sẽ được cập nhật và cải thiện liên tục dựa trên thông tin và thay đổi từ use case.

**Bài 4**

**Question 1**

**Điều gì nên được sử dụng để bổ sung mô tả use case?**

A. Các bảng số liệu phân tích.

B. Các tài liệu kỹ thuật chi tiết.

C. Các bảng màu thiết kế giao diện.

\*D. Các nguyên mẫu GUI hoặc bản phác thảo màn hình.

**Question 2**

**Nguyên tắc nào là quan trọng nhất khi xây dựng mô hình miền?**

A. Đảm bảo rằng tất cả các đối tượng đều được triển khai chi tiết.

B. Bao gồm tất cả các khía cạnh kỹ thuật.

C. Chỉ sử dụng ngôn ngữ kỹ thuật.

\*D. Bao quát ít nhất 80% các khái niệm quan trọng từ miền vấn đề.

**Question 3**

**Một trong các bước đầu tiên khi đánh giá use case là gì?**

A. Chuyển các yêu cầu chức năng vào use case.

\*B. Loại bỏ mọi thứ nằm ngoài phạm vi.

C. Tạo sơ đồ giao diện người dùng.

D. Thêm các yêu cầu mới vào mô hình.

**Question 4**

**Khi mô tả hành vi người dùng và phản hồi của hệ thống, điều gì cần được tránh?**

\*A. Kết hợp ngôn ngữ bị động và chủ động.

B. Mô tả chi tiết từng bước.

C. Sử dụng ngôn ngữ cụ thể.

D. Liên kết các yêu cầu với giao diện người dùng.

**Question 5**

**Điều gì sẽ xảy ra nếu các yêu cầu chức năng bị lẫn vào văn bản mô tả use case?**

A. Tăng tính rõ ràng của tài liệu.

B. Đơn giản hóa quá trình kiểm thử.

\*C. Gây nhầm lẫn khi phân tích và thiết kế.

D. Tăng tính linh hoạt khi cập nhật yêu cầu.

**Question 6**

**Việc tổ chức các use case thành các gói giúp ích như thế nào trong quy trình phát triển?**

A. Đơn giản hóa việc liên kết các yêu cầu với các thành phần phần mềm.

B. Giảm chi phí phát triển.

\*C. Cung cấp cái nhìn tổng quan rõ ràng hơn về các chức năng của hệ thống.

D. Tăng tốc độ viết tài liệu.

**Question 7**

**Tại sao cần có các "kịch bản thay thế" trong mỗi use case?**

A. Để đáp ứng các yêu cầu cấp cao của khách hàng.

B. Để đảm bảo rằng tất cả các yêu cầu chức năng được thực hiện.

C. Để đơn giản hóa mô tả use case.

\*D. Để xử lý các tình huống đặc biệt hoặc lỗi tiềm ẩn trong hệ thống.

**Question 8**

**Việc xác định các liên kết "is-a" và "has-a" trong mô hình miền có vai trò gì?**

A. Đảm bảo rằng các yêu cầu được đáp ứng đầy đủ.

B. Loại bỏ nhu cầu tạo các lớp đối tượng.

\*C. Tăng tính chính xác trong việc mô tả các mối quan hệ giữa các đối tượng.

D. Làm giảm độ phức tạp của hệ thống.

**Question 9**

**Trong buổi đánh giá yêu cầu, tại sao cần tạo nguyên mẫu (prototype) cho giao diện người dùng?**

A. Để giảm chi phí phát triển.

B. Để tạo tài liệu thiết kế chi tiết hơn.

\*C. Để xác minh rằng các yêu cầu được thể hiện đúng trong thiết kế giao diện.

D. Để tăng số lượng các yêu cầu được xử lý.

**Question 10**

**Tại sao nên sử dụng các bước "Hành động của người dùng / Phản hồi của hệ thống" trong mô tả use case?**

A. Để tăng độ phức tạp của tài liệu.

B. Để giảm thời gian thực hiện kiểm thử.

\*C. Để làm rõ cách hệ thống tương tác với người dùng.

D. Để đảm bảo rằng tất cả các yêu cầu kỹ thuật đều được đáp ứng.

**Question 11: Vai trò của việc tạo các nguyên mẫu giao diện (prototypes) trong buổi đánh giá yêu cầu là gì?**

A. Để tập trung vào các yêu cầu trừu tượng hơn.

B. Để xây dựng các chức năng phức tạp hơn.

\*C. Để xác định các hành vi người dùng mà tài liệu chưa nêu rõ.

D. Để giảm chi phí phát triển phần mềm.

**Question 12**

**Việc chuyển đổi từ giọng bị động sang giọng chủ động trong mô tả use case có tác dụng gì?**

A. Giúp tài liệu trở nên dễ kiểm tra hơn trong giai đoạn kiểm thử.

B. Tăng tính chi tiết trong mô tả giao diện người dùng.

C. Làm giảm số lượng bước cần thiết trong tài liệu.

\*D. Làm cho văn bản rõ ràng hơn về các hành động của người dùng và hệ thống.

**Question 13**

**Tại sao cần tạo ra các kịch bản "trời mưa" trong mỗi use case?**

A. Để giảm số lượng các yêu cầu chức năng cần xem xét.

\*B. Để dự đoán và xử lý các lỗi hoặc trường hợp ngoại lệ có thể xảy ra.

C. Để giúp hệ thống linh hoạt hơn trong việc thay đổi yêu cầu.

D. Để đáp ứng các yêu cầu về hiệu suất của hệ thống.

**Question 14**

**Bạn đang thực hiện đánh giá một use case, nhưng mô tả quá trừu tượng và khó hiểu. Bạn sẽ làm gì để cải thiện tài liệu?**

\*A. Cụ thể hóa các hành động của người dùng và phản hồi của hệ thống bằng cách sử dụng giọng chủ động.

B. Sử dụng các từ ngữ chuyên môn để làm tăng tính chuyên sâu của tài liệu.

C. Bỏ qua các phần trừu tượng và tập trung vào giao diện người dùng.

D. Chuyển đổi toàn bộ văn bản sang dạng bảng biểu.

**Question 15**

**Trong quá trình đánh giá, khách hàng phàn nàn rằng hệ thống không đáp ứng được yêu cầu của họ, mặc dù yêu cầu đã được phê duyệt trước đó. Bạn sẽ xử lý thế nào?**

A. Tạm dừng dự án và yêu cầu khách hàng cung cấp yêu cầu mới.

B. Chuyển yêu cầu đến nhóm phát triển mà không thay đổi gì.

\*C. Thực hiện lại buổi đánh giá yêu cầu với khách hàng để tìm ra sự khác biệt và cập nhật tài liệu.

D. Loại bỏ ý kiến của khách hàng vì không phù hợp với giai đoạn hiện tại.

**Question 16**

**Ai cần tham gia vào buổi đánh giá yêu cầu?**

\*A. Đại diện khách hàng, người dùng cuối, nhân viên marketing, và các bên liên quan khác.

B. Nhóm quản lý dự án.

C. Chỉ đội ngũ phát triển.

D. Chỉ các lập trình viên.

**Question 17**

**Tại sao cần xem xét các yêu cầu trước khi bắt đầu mã hóa?**

A. Để giảm chi phí phát triển.

B. Để tiết kiệm thời gian lập trình.

C. Để tránh các thay đổi yêu cầu trong tương lai.

\*D. Để xác minh rằng các yêu cầu không bị hiểu sai.

**Question 18**

**Một use case không có "kịch bản thay thế" thường cho thấy điều gì?**

A. Nó đã được phân tích đầy đủ.

B. Nó chỉ dành cho các hệ thống nhỏ.

C. Nó không cần thiết trong quy trình phát triển.

\*D. Nó thiếu các kịch bản xử lý lỗi hoặc trường hợp đặc biệt.

**Question 19**

**Tại sao việc mời đại diện khách hàng tham gia buổi đánh giá yêu cầu lại quan trọng?**

A. Để xác định chi phí thực hiện dự án.

B. Để tăng cường mối quan hệ giữa khách hàng và nhà phát triển.

\*C. Để đảm bảo rằng yêu cầu hệ thống phản ánh chính xác nhu cầu thực tế của khách hàng.

D. Để tiết kiệm thời gian hoàn thành dự án.

**Question 20**

**Khi mô tả các bước trong use case, tại sao cần liên kết các bước này với các đối tượng miền?**

\*A. Để đảm bảo rằng mô hình miền hỗ trợ trực tiếp các yêu cầu của hệ thống.

B. Để tăng tính trừu tượng của tài liệu.

C. Để tránh sử dụng các thuật ngữ không cần thiết.

D. Để giảm bớt số lượng các bước.

**Question 21**

**Tại sao việc viết use case theo ngôn ngữ chủ động lại quan trọng?**

A. Để tách biệt yêu cầu chức năng khỏi giao diện người dùng.

**\*B. Để giúp lập trình viên hiểu rõ hơn về cách triển khai yêu cầu.**

C. Để làm cho tài liệu trở nên dễ đọc hơn với khách hàng.

D. Để giảm thời gian kiểm thử hệ thống.

**Question 22**

**Tại sao việc đánh giá các use case cần có sự tham gia của cả người dùng cuối?**

A. Người dùng cuối chịu trách nhiệm tài trợ cho dự án.

B. Người dùng cuối có kinh nghiệm về quy trình phát triển.

C. Người dùng cuối thường có quyền quyết định về mô hình thiết kế.

\*D. Người dùng cuối có thể cung cấp phản hồi thực tế về cách hệ thống sẽ được sử dụng.

**Question 23**

**Làm thế nào việc truy vết (traceability) các yêu cầu đến các use case giúp ích cho dự án?**

A. Tăng độ chi tiết của mô hình miền.

B. Giảm nhu cầu kiểm tra hệ thống.

C. Giảm thời gian viết các use case.

\*D. Đảm bảo rằng mọi yêu cầu đều được thực hiện và kiểm tra.

**Question 24**

**Tại sao việc đặt tên rõ ràng cho các đối tượng trong mô hình miền lại quan trọng?**

A. Để giúp khách hàng dễ hiểu hơn về các yêu cầu.

B. Để đơn giản hóa tài liệu hóa hệ thống.

\*C. Để đảm bảo rằng các lập trình viên hiểu rõ vai trò của từng đối tượng.

D. Để tăng tốc độ hoàn thành mô hình.

**Question 25**

**Một use case mô tả rằng "Hệ thống phải cho phép người dùng tìm kiếm sản phẩm". Để cải thiện, bạn sẽ làm gì?**

A. Xóa use case này vì nó quá cơ bản.

\*B. Thay đổi câu mô tả thành: "Người dùng nhập từ khóa tìm kiếm, hệ thống hiển thị danh sách sản phẩm phù hợp."

C. Tăng cường mô tả bằng cách thêm các bước xử lý lỗi vào phần chính.

D. Liên kết use case với mô hình miền mà không thay đổi nội dung.

**Question 26**

**Mục đích của buổi xem xét yêu cầu (requirements review session) là gì?**

A. Để kiểm tra tính khả thi của các yêu cầu kỹ thuật.

B. Để thiết kế giao diện người dùng cho hệ thống.

\*C. Để đảm bảo rằng hệ thống được mô tả khớp với các yêu cầu.

D. Để đánh giá khả năng mở rộng của hệ thống.

**Question 27**

**Lợi ích của việc sử dụng ngôn ngữ chủ động trong mô tả use case là gì?**

A. Loại bỏ hoàn toàn các yêu cầu chức năng.

B. Giảm thời gian viết tài liệu.

\*C. Làm cho các yêu cầu rõ ràng hơn và dễ hiểu hơn.

D. Hạn chế số lượng lỗi khi mã hóa.

**Question 28**

**Làm thế nào để tránh tình trạng "intermangled" trong đặc tả use case?**

\*A. Ghi lại các yêu cầu chức năng trong tài liệu riêng biệt.

B. Kết hợp các yêu cầu vào mô hình miền.

C. Sử dụng các thuật ngữ trừu tượng.

D. Loại bỏ hoàn toàn các yêu cầu chức năng.

**Question 29**

**Việc sử dụng giao diện người dùng (GUI) giúp cải thiện đặc tả use case như thế nào?**

A. Tăng khả năng tái sử dụng của các yêu cầu chức năng.

B. Loại bỏ nhu cầu sử dụng sơ đồ use case.

C. Giảm thời gian phát triển giao diện.

\*D. Làm rõ hành vi của người dùng và phản hồi của hệ thống.

**Question 30**

**Lợi ích của việc liên kết use case với mô hình miền là gì?**

A. Tăng tốc độ phát triển giao diện người dùng.

B. Loại bỏ các lớp dư thừa trong thiết kế.

C. Giảm độ phức tạp của mã nguồn.

\*D. Đảm bảo rằng các mô tả use case phù hợp với các đối tượng trong miền.

**Bài 5**

### **Question 1**

**Vai trò của các bộ điều khiển (controllers) trong sơ đồ độ bền là gì?**

A. Được sử dụng để lưu trữ dữ liệu người dùng.

**\*B. Là cầu nối giữa các đối tượng biên và đối tượng thực thể.**

C. Là các lớp trình bày giao diện.

D. Chỉ đơn thuần là các lớp cơ sở dữ liệu.

### **Question 2**

**Tại sao sơ đồ độ bền được coi là một bước thiết kế khái niệm ban đầu?**

A. Nó loại bỏ sự cần thiết của phân tích yêu cầu.

B. Nó thay thế hoàn toàn sơ đồ trình tự (sequence diagram).

**\*C. Nó giúp xác minh yêu cầu trước khi chuyển sang thiết kế chi tiết.**

D. Nó chỉ cần thiết trong các dự án nhỏ.

### **Question 3**

**Các mũi tên trên sơ đồ độ bền có ý nghĩa gì?**

A. Chúng chỉ định hướng của các use case.

B. Chúng không có ý nghĩa cụ thể.

**\*C. Chúng đại diện cho mối liên hệ về dữ liệu hoặc điều khiển giữa các đối tượng.**

D. Chúng chỉ ra cấu trúc của cơ sở dữ liệu.

### **Question 4**

**Điều gì thường xảy ra trong quá trình phân tích độ bền?**

A. Tất cả các yêu cầu được loại bỏ.

**\*B. Các use case được viết lại để loại bỏ sự mơ hồ.**

C. Mô hình miền được hoàn thiện ngay lập tức.

D. Các đối tượng được chuyển đổi thành mã nguồn.

### **Question 5**

**Tại sao các đối tượng biên cần được đặt tên rõ ràng?**

**\*A. Để giảm mâu thuẫn trong mô hình.**

B. Để tăng tính trừu tượng của sơ đồ.

C. Để đơn giản hóa thiết kế.

D. Để đảm bảo rằng chúng có thể được chuyển trực tiếp sang mã nguồn.

### **Question 6**

**Điều gì xảy ra nếu không thể tạo một sơ đồ độ bền từ một use case?**

A. Các đối tượng không cần thiết phải được xác định.

B. Hệ thống có thể được phát triển mà không cần sơ đồ.

**\*C. Use case cần được xem xét và chỉnh sửa.**

D. Sơ đồ độ bền không cần thiết cho trường hợp này.

### **Question 7**

**Khi nào một sơ đồ độ bền được coi là hoàn chỉnh?**

A. Khi tất cả các đối tượng đều là thực thể.

**\*B. Khi nó bao gồm cả kịch bản chính và các kịch bản thay thế.**

C. Khi nó chỉ bao gồm các đối tượng thực thể và biên.

D. Khi không có mũi tên nào trên sơ đồ.

### **Question 8**

**Tại sao các đối tượng thực thể và biên được coi là danh từ, trong khi bộ điều khiển là động từ?**

A. Để giảm sự phức tạp trong quá trình thiết kế.

**\*B. Để đảm bảo cấu trúc rõ ràng và dễ hiểu cho sơ đồ.**

C. Để tối ưu hóa việc chuyển đổi sang mã nguồn

D. Để làm rõ ranh giới giữa các lớp.

### **Question 9**

**Bạn phát hiện một đối tượng trên sơ đồ độ bền không có trong mô hình miền. Bạn sẽ làm gì?**

**\*A. Thêm đối tượng này vào mô hình miền.**

B. Thay thế đối tượng bằng một bộ điều khiển.

C. Đưa đối tượng vào danh sách cần kiểm tra sau.

D. Loại bỏ đối tượng vì không phù hợp.

### **Question 10**

**Tại sao các sơ đồ độ bền thường mất nhiều thời gian để hoàn thành nếu văn bản use case chưa được viết đúng cách?**

A. Vì cần kiểm tra lại toàn bộ mô hình miền.

B. Vì các đối tượng không được tự động tạo ra từ văn bản.

C. Vì các use case phức tạp hơn sơ đồ.

**\*D. Vì sơ đồ yêu cầu tên gọi và mối quan hệ chính xác từ văn bản.**

### **Question 11**

**Trong sơ đồ độ bền, các bộ điều khiển thường đại diện cho điều gì?**

**\*A. Các hành động hoặc chức năng logic.**

B. Các lớp trong cơ sở dữ liệu.

C. Các đối tượng tĩnh trong hệ thống.

D. Các giao diện người dùng.

### **Question 12**

**Tại sao các mũi tên trong sơ đồ độ bền không cần chỉ rõ hướng?**

A. Vì hướng không ảnh hưởng đến logic của sơ đồ.

B. Vì các mũi tên không cần thiết trong sơ đồ.

**\*C. Vì sơ đồ chỉ tập trung vào mối liên hệ, không phải luồng dữ liệu.**

D. Vì các mũi tên được sử dụng để chỉ ra trạng thái, không phải hành động.

### **Question 13**

**Làm thế nào để đảm bảo rằng một sơ đồ độ bền bao quát toàn bộ use case?**

A. Chỉ tập trung vào các đối tượng thực thể quan trọng.

B. Loại bỏ các đối tượng không cần thiết khỏi sơ đồ.

C. Thêm nhiều bộ điều khiển để kiểm soát các luồng dữ liệu.

**\*D. Đảm bảo rằng tất cả các bước từ văn bản đều xuất hiện trên sơ đồ.**

### **Question 14**

**Khi sử dụng một công cụ CASE để vẽ sơ đồ độ bền, lợi ích lớn nhất là gì?**

**\*A. Dễ dàng liên kết văn bản use case với sơ đồ.**

B. Tăng độ phức tạp của sơ đồ.

C. Tự động phát hiện các lỗi logic

D. Loại bỏ nhu cầu viết lại văn bản use case.

### **Question 15**

**Đối tượng nào dưới đây không phải là đối tượng biên?**

A. Hộp thoại lỗi

B. Màn hình đăng nhập

C. Hòm thư (mailbox)

**\*D. Bảng tài khoản**

### **Question 16**

**Các đối tượng nào thường được phân loại trong sơ đồ độ bền?**

**\*A. Đối tượng biên, đối tượng thực thể, và bộ điều khiển.**

B. Đối tượng giao diện, đối tượng miền, và đối tượng chức năng.

C. Đối tượng mô hình, đối tượng giao diện, và đối tượng lớp.

D. Đối tượng giao diện, đối tượng giao tiếp, và đối tượng quản lý.

### **Question 17**

**Quy tắc nào KHÔNG đúng trong sơ đồ độ bền?**

A. Nouns có thể nói chuyện với verbs.

**\*B. Nouns (danh từ) không thể nói chuyện với verbs (động từ).**

C. Nouns không thể nói chuyện với nhau.

D. Verbs có thể nói chuyện với verbs.

### **Question 18**

**Các bộ điều khiển trong sơ đồ độ bền có thể được chuyển đổi thành gì trong sơ đồ trình tự?**

A. Các lớp thực thể.

B. Các lớp biên.

**\*C. Các phương thức hoặc thông điệp.**

D. Các đối tượng.

### **Question 19**

**Một sơ đồ độ bền thiếu bộ điều khiển (controller) giữa đối tượng biên và đối tượng thực thể. Điều này ảnh hưởng như thế nào đến hệ thống?**

A. Làm tăng độ phức tạp của mã nguồn.

B. Gây ra lỗi khi triển khai hệ thống.

**\*C. Làm mất đi cấu trúc "danh từ-động từ-danh từ".**

D. Không ảnh hưởng vì các đối tượng này có thể nói chuyện trực tiếp.

### **Question 20**

### **Làm thế nào để phát hiện một use case cần được viết lại khi vẽ sơ đồ độ bền?**

A. Khi văn bản use case quá ngắn.

B. Khi số lượng đối tượng vượt quá giới hạn.

**\*C. Khi sơ đồ độ bền không tuân theo quy tắc "danh từ-động từ-danh từ".**

D. Khi không có đối tượng biên nào được xác định.

D. Khi trang đó không có dữ liệu động.

### **Question 21**

**Khi nào cần sử dụng đối tượng hiển thị (Display controller) trong sơ đồ độ bền?**

A. Khi không thể xác định bộ điều khiển cụ thể.

B. Khi cần kết nối hai đối tượng thực thể.

**\*C. Khi một trang hoặc màn hình cần được khởi tạo và hiển thị.**

D. Khi trang đó không có dữ liệu động.

### **Question 22**

**Nếu một use case có nhiều kịch bản thay thế, sơ đồ độ bền nên xử lý thế nào?**

A. Tạo các sơ đồ riêng biệt cho từng kịch bản.

B. Không cần xử lý kịch bản thay thế.

**\*C. Bao gồm tất cả các kịch bản thay thế trong cùng một sơ đồ.**

D. Chỉ tập trung vào kịch bản chính.

### **Question 23**

**Tại sao không nên sử dụng tên công nghệ như "JSP" trong sơ đồ độ bền?**

A. Vì nó không tương thích với mô hình miền.

**\*B. Vì sơ đồ độ bền tập trung vào thiết kế khái niệm, không phải chi tiết công nghệ.**

C. Vì điều này sẽ làm tăng độ phức tạp của sơ đồ.

D. Vì các công nghệ luôn thay đổi.

### **Question 24**

**Tại sao sơ đồ độ bền lại đóng vai trò cầu nối giữa phân tích yêu cầu và thiết kế chi tiết?**

A. Vì nó thay thế mô hình miền và sơ đồ trình tự.

B. Vì nó chỉ tập trung vào giao diện người dùng.

**\*C. Vì nó xác minh rằng các yêu cầu có thể chuyển đổi thành các đối tượng thiết kế.**

D. Vì nó loại bỏ nhu cầu tạo sơ đồ trình tự.

### **Question 25**

**Mục tiêu chính của phân tích độ bền (robustness analysis) là gì?**

A. Xác định tất cả các kịch bản thay thế.

B. Phát triển mã nguồn ngay lập tức.

C. Tạo giao diện người dùng hoàn chỉnh.

**\*D. Liên kết các use case với các đối tượng.**

### **Question 26**

**Những đối tượng nào thường được lấy từ mô hình miền?**

**\*A. Đối tượng thực thể.**

B. Đối tượng giao tiếp.

C. Bộ điều khiển.

D. Đối tượng biên.

### **Question 27**

**Tại sao cần kiểm tra tính chính xác giữa sơ đồ độ bền và văn bản use case?**

A. Để tránh việc tạo ra các đối tượng không cần thiết.

B. Để giảm độ phức tạp của sơ đồ.

C. Để tăng tính trừu tượng của hệ thống.

**\*D. Để xác định các thành phần còn thiếu.**

### **Question 28**

**Điều gì không được thực hiện trong phân tích độ bền?**

A. Tạo mối liên hệ giữa các use case và các đối tượng.

B. Loại bỏ sự mơ hồ trong văn bản use case.

C. Xác định các lớp thực thể mới.

**\*D. Xóa bỏ các use case không liên quan.**

### **Question 29**

**Một sơ đồ độ bền thường có mối liên hệ nào với văn bản mô tả use case?**

A. Mối liên hệ nhiều:1.

B. Mối liên hệ 1: nhiều.

C. Không có mối liên hệ.

**\*D. Mối liên hệ 1:1.**

### **Question 30**

**Vai trò chính của các đối tượng biên trong sơ đồ độ bền là gì?**

A. Là các thành phần trong cơ sở dữ liệu.

**\*B. Là giao diện giữa hệ thống và thế giới bên ngoài.**

C. Là các yếu tố không liên quan đến giao diện người dùng.

D. Là các lớp chứa logic xử lý chính.

**Bài 6**

Question 1

**Highlighter Test trong Preliminary Design Review (PDR) được sử dụng để làm gì?**

A. Đánh dấu các lỗi logic

B. Xác minh rằng các sơ đồ trình tự đã hoàn thành.

**\*C. Kiểm tra sự khớp giữa văn bản use case và sơ đồ độ mạnh.**

D. Đánh giá mức độ chi tiết của mã nguồn.

Question 3

**Mục tiêu chính của 6 bước đơn giản trong PDR là gì?**

A. Tạo một mô hình chi tiết cho hệ thống.

B. Tăng tốc quá trình kiểm thử.

C. Xác định hành vi cho từng lớp thực thể.

**\*D. Đảm bảo sơ đồ độ bền không vi phạm quy tắc phân tích.**

Question 4

**Tại sao khách hàng không nên tham gia thiết kế chi tiết sau PDR?**

A. Vì thiết kế chi tiết không ảnh hưởng đến yêu cầu.

B. Vì nhóm phát triển cần sự tập trung.

C. Vì khách hàng không đủ hiểu biết kỹ thuật.

**\*D. Vì các yêu cầu đã được xác định trước đó.**

Question 5

**Làm thế nào để phát hiện lỗi trong sơ đồ độ bền?**

**\*A. Sử dụng Highlighter Test.**

B. Loại bỏ các kịch bản thay thế.

C. Yêu cầu khách hàng kiểm tra.

D. Đọc lại toàn bộ mã nguồn.

Question 6

**Tại sao phải đảm bảo các màn hình (giao diện) và lớp thực thể có tên rõ ràng?**

A. Để giảm độ phức tạp của sơ đồ.

B. Để đảm bảo khách hàng hiểu rõ.

**\*C. Để tránh mâu thuẫn khi chuyển sang thiết kế chi tiết.**

D. Để đơn giản hóa quá trình kiểm thử.

Question 7

**Khi phát hiện một lỗi giao tiếp danh từ-danh từ trong sơ đồ độ bền, bạn sẽ làm gì?**

A. Bỏ qua lỗi này vì không ảnh hưởng đến hệ thống.

B. Yêu cầu khách hàng sửa lại yêu cầu.

C. Chuyển đổi danh từ thành động từ.

**\*D. Thêm một controller để xử lý giao tiếp.**

Question 8

**Một sơ đồ độ bền không khớp với văn bản use case. Bạn sẽ làm gì?**

A. Bỏ qua vấn đề vì nó không ảnh hưởng nhiều.

**\*B. Điều chỉnh sơ đồ hoặc văn bản để chúng đồng bộ.**

C. Loại bỏ sơ đồ và chỉ sử dụng văn bản.

D. Thêm các thành phần mới vào sơ đồ.

Question 9

**Khách hàng yêu cầu thêm một kịch bản thay thế trong PDR. Bạn sẽ xử lý thế nào?**

**\*A. Thảo luận với nhóm phát triển và khách hàng để quyết định.**

B. Chỉ thêm kịch bản vào sơ đồ.

C. Bỏ qua yêu cầu vì đã hoàn thành PDR.

D. Cập nhật sơ đồ độ bền và văn bản để bổ sung kịch bản.

Question 10

**Tại sao khách hàng nên được tham gia đánh giá sơ đồ trong PDR?**

A. Để xác định các lớp thực thể mới.

**\*B. Để đảm bảo rằng các yêu cầu được hiểu đúng.**

C. Để đánh giá hiệu suất hệ thống.

D. Để kiểm tra các thuật toán phức tạp.

Question 11

**Tại sao tên gọi trong sơ đồ độ bềncần rõ ràng và nhất quán?**

A. Để đơn giản hóa việc kiểm thử.

B. Để khách hàng không đặt thêm yêu cầu.

C. Để giảm số lượng controllers cần thiết.

**\*D. Để tránh nhầm lẫn khi chuyển sang thiết kế chi tiết.**

Question 12

**Điều gì xảy ra nếu các controllers không giao tiếp đúng cách với các lớp thực thể?**

A. Mô hình miền sẽ tự động điều chỉnh.

B. Sơ đồ trình tự sẽ khắc phục lỗi này.

C. Các lớp thực thể vẫn hoạt động bình thường.

**\*D. Hệ thống sẽ thiếu logic điều phối dữ liệu.**

Question 13

**Sơ đồ độ bền giúp ích gì cho thiết kế chi tiết?**

A. Giảm số lượng kịch bản cần kiểm tra.

B. Hỗ trợ triển khai mã nguồn ngay lập tức.

**\*C. Xác định rõ các đối tượng và luồng dữ liệu.**

D. Loại bỏ sự cần thiết của sơ đồ trình tự.

Question 14

**Bạn nhận thấy rằng một sơ đồ độ bền không bao gồm kịch bản lỗi cho một use case quan trọng. Bạn sẽ làm gì?**

A. Chuyển vấn đề này sang giai đoạn kiểm thử.

B. Loại bỏ sơ đồ này vì không phù hợp.

C. Thêm các controllers tạm thời để xử lý lỗi trong sơ đồ.

**\*D. Xem xét lại văn bản use case để xác định hành vi lỗi cần thêm.**

Question 15

**Trong một sơ đồ độ bền, bạn nhận thấy có giao tiếp trực tiếp giữa hai entity objects mà không qua controller. Điều này vi phạm quy tắc gì và cách xử lý nào phù hợp nhất?**

A. Quy tắc alternate courses; chuyển luồng dữ liệu vào boundary object.

B. Quy tắc giao tiếp controller-to-boundary; thêm thuộc tính vào entity objects.

C. Quy tắc liên kết trực tiếp; xóa một trong hai entity objects.

**\*D. Quy tắc giao tiếp danh từ-danh từ; thêm controller để điều phối giao tiếp.**

Question 16

**Ai nên tham gia vào buổi PDR?**

**\*A. Nhóm phát triển, đại diện khách hàng, và quản lý dự án.**

B. Chỉ khách hàng.

C. Chỉ nhóm phát triển.

D. Chỉ đội kiểm thử.

Question 17

**Preliminary Design Review (PDR) nhằm mục đích gì?**

A. Thêm yêu cầu mới từ khách hàng.

B. Bắt đầu giai đoạn phát triển chi tiết.

C. Đánh giá mã nguồn đã hoàn thành.

**\*D. Đảm bảo rằng các sơ đồ độ bền, mô hình miền và văn bản use case khớp nhau**

Question 18

**Điều gì cần tránh khi tạo sơ đồ độ bền trong PDR?**

**\*A. Đi sâu vào thiết kế chi tiết.**

B. Đặt tên rõ ràng cho các màn hình và thực thể.

C. Bao gồm cả kịch bản chính và kịch bản thay thế.

D. Sử dụng controllers để liên kết các đối tượng.

Question 19

**Tại sao cần phải liệt kê hành vi của tất cả các kịch bản thay thế?**

A. Vì chúng được triển khai trực tiếp trong mã nguồn.

**\*B. Vì chúng chiếm hơn 50% độ phức tạp của phần mềm.**

C. Vì chúng được khách hàng ưu tiên.

D. Vì chúng giúp đơn giản hóa thiết kế chi tiết.

Question 20

**Làm thế nào để kiểm tra rằng một sơ đồ độ bền đầy đủ?**

A. Đảm bảo rằng chỉ có các lớp thực thể xuất hiện.

B. Đếm số lượng controllers trong sơ đồ.

C. Xem xét xem sơ đồ có chi tiết như sơ đồ trình tự không.

**\*D. Xác minh rằng tất cả các kịch bản chính và thay thế được thể hiện.**

Question 21

**Điều gì cần làm nếu phát hiện một đối tượng mới trong sơ đồ độ bền?**

A. Tạo một lớp thực thể mới ngay lập tức.

B. Bỏ qua đối tượng mới nếu không ảnh hưởng đến sơ đồ.

C. Loại bỏ đối tượng để tránh làm phức tạp sơ đồ.

**\*D. Thêm đối tượng vào mô hình miền và đánh giá sự liên kết.**

Question 22

**Tại sao không nên loại bỏ các bước kiểm tra sơ đồ trong PDR?**

A. Vì kiểm tra không ảnh hưởng đến kết quả cuối cùng.

**\*B. Vì các lỗi nhỏ có thể dẫn đến vấn đề lớn trong thiết kế chi tiết.**

C. Vì khách hàng yêu cầu giữ nguyên sơ đồ ban đầu.

D. Vì sơ đồ độ bền không cần thay đổi sau PDR.

Question 23

**Một sơ đồ độ bền thể hiện luồng dữ liệu không rõ ràng giữa các controllers và entity objects. Bạn sẽ làm gì để làm rõ vấn đề này?**

A. Bỏ qua vấn đề này nếu không ảnh hưởng lớn đến thiết kế chi tiết.

B. Loại bỏ các controllers không cần thiết.

C. Tạo một sơ đồ trình tự để thay thế sơ đồ độ bền.

**\*D. Xác định và thêm các thuộc tính cụ thể cho entity objects để làm rõ luồng dữ liệu**

Question 24

**Sau khi hoàn thành PDR, bước tiếp theo là gì?**

A. Gửi tài liệu yêu cầu cho khách hàng.

B. Thêm các lớp thực thể mới.

C. Kiểm thử hệ thống.

**\*D. Phát triển chi tiết thiết kế.**

Question 25

**Bạn cần thêm một thực thể mới vào mô hình miền trong quá trình PDR. Bước đầu tiên là gì?**

A. Chỉnh sửa toàn bộ sơ đồ trình tự.

B. Yêu cầu nhóm phát triển tạo mã nguồn.

C. Xác định vị trí của thực thể trong sơ đồ độ bền.

**\*D. Thêm thực thể trực tiếp vào mô hình miền.**

Question 26

**Nếu sơ đồ độ bền có quá nhiều chi tiết, bạn sẽ làm gì?**

A. Loại bỏ các controllers không cần thiết.

**\*B. Chia sơ đồ thành các sơ đồ nhỏ hơn.**

C. Xóa bỏ các kịch bản thay thế.

D. Chỉ tập trung vào các lớp thực thể.

Question 27

**Điều gì xảy ra nếu không liệt kê đầy đủ các alternate courses?**

A. Sơ đồ trình tự sẽ thiếu tính hợp lệ.

B. Các yêu cầu chính bị bỏ sót.

C. Mô hình miền trở nên không đầy đủ.

**\*D. Hệ thống không xử lý được các trường hợp đặc biệt.**

Question 28

**Tại sao các sơ đồ không nên đi vào thiết kế chi tiết trong PDR?**

**\*A. Để đảm bảo rằng tất cả các bên đều hiểu rõ.**

B. Vì sơ đồ độ bền không cần chi tiết.

C. Vì sơ đồ chi tiết gây nhầm lẫn.

D. Vì thiết kế chi tiết chỉ dành cho đội kiểm thử.

Question 29

**Tại sao các thuộc tính của lớp thực thể cần được xác định trước khi mã hóa?**

A. Để kiểm tra tính chính xác của giao diện người dùng

**\*B. Để hoàn thiện mô hình miền**

C. Để tối ưu hóa hiệu suất mã nguồn.

D. Để đảm bảo rằng sơ đồ trình tự có đủ thông tin.

Question 30

**Preliminary Design Review (PDR) là giai đoạn cuối cùng mà khách hàng có thể trực tiếp góp ý vào điều gì?**

A. Quy trình kiểm thử.

B. Thiết kế chi tiết và mã nguồn.

C. Lựa chọn ngôn ngữ lập trình.

**\*D. Các sơ đồ và yêu cầu chức năng.**

### **Question 31**

**Quy tắc nào sau đây vi phạm trong PDR**

A. Actors chỉ liên kết với boundary objects.

**\*B. Controllers phải giao tiếp trực tiếp với entity objects.**

C. Alternate courses có thể bị bỏ qua.

D. Không có giao tiếp danh từ-danh từ giữa các đối tượng.

Bài 7

### Question 1

**Vai trò của Technical Architect (Kiến trúc kỹ thuật) là gì?**

A. Đảm bảo rằng khách hàng hiểu rõ thiết kế chi tiết.

B. Tạo mã nguồn cho toàn bộ hệ thống.

C. Hoàn thành kiểm thử hệ thống.

**\*D. Xây dựng kiến trúc đúng, truyền đạt và đảm bảo tất cả các bên hiểu rõ.**

### Question 2

**Một kiến trúc tốt dựa trên yếu tố nào?**

A. Dự đoán xu hướng thị trường.

B. Công nghệ mới nhất trên thị trường.

C. Tham khảo các dự án tương tự.

**\*D. Yêu cầu kinh doanh và dịch vụ đã phân tích kỹ lưỡng.**

### Question 3

**Thành phần chính trong mô hình MVC là gì?**

A. Model, Component, Database.

B. Controller, API, Component.

C. View, Database, API.

**\*D. Model, View, Controller.**

### Question 4

**Spring Framework sử dụng phương pháp nào để quản lý dependency giữa các lớp?**

A. Direct Dependency Injection.

B. Command and Control.

C. Event-Driven Model

**\*D. Inversion of Control (IoC).**

### Question 5

**Các DAO trong Spring Framework đại diện cho điều gì?**

A. Các thuộc tính trong giao diện người dùng.

**\*B. Các giao diện truy cập dữ liệu.**

C. Các đối tượng lưu trữ trong bộ nhớ.

D. Các đối tượng truy cập trình duyệt.

### Question 6

**Trong Internet Bookstore, Spring Framework sử dụng đối tượng nào để truyền dữ liệu giữa controller và view?**

A. Domain Model.

B. JSP Handler.

C. Command Object.

**\*D. ModelAndView.**

### Question 7

**Vai trò của DispatcherServlet trong Spring Framework là gì?**

A. Gửi yêu cầu trực tiếp tới view.

B. Lưu trữ dữ liệu tạm thời.

**\*C. Chuyển yêu cầu của người dùng tới controller phù hợp.**

D. Tạo kết nối cơ sở dữ liệu.

### Question 8

**Bạn cần tăng tính mở rộng (scalability) của hệ thống sử dụng Spring Framework. Bạn sẽ làm gì?**

A. Tối ưu hóa mô hình DAO để giảm truy vấn dư thừa.

**\*B. Sử dụng Microservices Architecture thay vì Layered Architecture**

C. Thêm nhiều controllers vào hệ thống.

D. Tăng số lượng giao diện (view) để phân phối yêu cầu.

### Question 9

**Khi sử dụng Spring Framework, làm thế nào để quản lý dependency phức tạp giữa các lớp?**

**\*A. Sử dụng Inversion of Control (IoC) để quản lý dependency.**

B. Tạo các controller bổ sung để giảm sự phụ thuộc.

C. Tăng số lượng DAO để quản lý dependency tốt hơn.

D. Loại bỏ dependency giữa các lớp.

### Question 10

**Nếu một lớp controller trong Spring Framework phải xử lý quá nhiều logic, bạn sẽ cải thiện thiết kế như thế nào?**

A. Tăng số lượng view để xử lý logic.

**\*B. Chuyển một phần logic vào service layer.**

C. Tách controller thành nhiều lớp nhỏ hơn.

D. Gộp controller với DAO để giảm bớt các tầng.

### Question 11

**Trong hệ thống sử dụng Spring, bạn cần theo dõi tất cả các giao dịch (transactions). Thành phần nào phù hợp nhất để triển khai điều này?**

**\*A. Aspect-Oriented Programming (AOP).**

B. DAO

C. DispatcherServlet.

D. Dependency Injection (DI).

### Question 12

**Một lỗi phổ biến khi thiết kế kiến trúc kỹ thuật là gì?**

A. Dùng AOP cho tất cả các tầng của hệ thống.

**\*B. Không tối ưu hóa các query trong DAO.**

C. Không sử dụng các view phức tạp.

D. Quá phụ thuộc vào Model và View.

### Question 13

**Một hệ thống sử dụng kiến trúc Layered Architecture, nhưng hiệu suất kém khi số lượng giao dịch tăng. Bạn sẽ xử lý thế nào?**

A. Gộp tất cả các tầng thành một module.

B. Loại bỏ DispatcherServlet khỏi cấu trúc.

C. Tăng số lượng Model trong MVC.

**\*D. Sử dụng caching trong DAO để giảm số lượng truy vấn cơ sở dữ liệu.**

### Question 14

**Khi cần tích hợp nhiều loại cơ sở dữ liệu, bạn sẽ điều chỉnh thành phần nào trong Spring Framework?**

A. View để kết nối trực tiếp với cơ sở dữ liệu.

B. Controller để hỗ trợ nhiều giao diện.

**\*C. DAO để cung cấp abstraction phù hợp cho nhiều cơ sở dữ liệu.**

D. DispatcherServlet để điều chỉnh dữ liệu.

### Question 15

**Một nhóm phát triển tích hợp nhiều loại cơ sở dữ liệu nhưng không sử dụng abstraction trong DAO. Hậu quả nào có thể xảy ra, và bạn sẽ đưa ra giải pháp nào?**

**\*A. Hậu quả: Code phụ thuộc mạnh vào cơ sở dữ liệu cụ thể; giải pháp: sử dụng abstraction để giảm phụ thuộc.**

B. Hậu quả: Lỗi bảo mật; giải pháp: sử dụng AOP để bảo vệ DAO.

C. Hậu quả: Hiệu suất thấp; giải pháp: thêm caching vào view.

D. Hậu quả: Hệ thống không thể mở rộng; giải pháp: giảm số lượng DAO.

### Question 16

**Mô hình "Layered Architecture" trong TA thường phân tách hệ thống thành những gì?**

A. Các lớp giao diện, dịch vụ và bảo mật.

B. Các lớp kiểm thử, mã hóa và triển khai.

C. Các yếu tố thiết kế giao diện người dùng.

**\*D. Mô hình dữ liệu, gói/thành phần và triển khai**

### Question 17

**Điểm mạnh chính của mô hình DAO là gì?**

A. Hỗ trợ mọi giao diện người dùng

B. Loại bỏ hoàn toàn các lỗi giao diện.

C. Tăng tốc độ kết nối cơ sở dữ liệu.

**\*D. Truy cập dữ liệu mà không phụ thuộc vào loại cơ sở dữ liệu.**

### Question 18

**Trong một hệ thống e-commerce, DispatcherServlet không thể xử lý yêu cầu đúng cách. Bạn sẽ kiểm tra thành phần nào đầu tiên?**

A. DAO để đảm bảo dữ liệu đã được truy cập đúng.

**\*B. Cấu hình DispatcherServlet trong file XML hoặc annotation.**

C. Controllers trong Spring Framework.

D. Các thành phần view để đảm bảo dữ liệu được hiển thị.

### Question 19

**Làm thế nào để áp dụng mô hình MVC để đảm bảo hệ thống dễ dàng bảo trì?**

**\*A. Phân chia rõ ràng giữa Model, View, và Controller.**

B. Loại bỏ Model để giảm sự phức tạp.

C. Đưa tất cả logic xử lý vào View để dễ kiểm tra.

D. Gộp các View và Controller thành một module.

### Question 20

**Nếu hệ thống có nhiều dependency, cách nào giúp giảm độ phức tạp khi triển khai?**

A. Tạo thêm các thành phần trực tiếp để giảm số lớp.

B. Tăng số lượng controller trong hệ thống.

C. Chuyển toàn bộ dependency vào view.

**\*D. Dùng Inversion of Control (IoC) để quản lý dependency.**

### Question 21

**Lợi ích chính của Spring Framework là gì?**

A. Tích hợp tốt với Visual Basic.

B. Tăng tốc độ xử lý giao diện người dùng.

C. Giúp tạo giao diện đồ họa dễ dàng.

**\*D. Đơn giản hóa việc quản lý dữ liệu và giảm mã lặp lại.**

### Question 22

**Spring Framework hỗ trợ kiến trúc nào trong Internet Bookstore?**

A. Microservices Architecture.

**\*B. Layered Architecture.**

C. Event-Driven Architecture (EDA)

D. Service-Oriented Architecture (SOA).

### Question 23

**AOP trong Spring Framework được sử dụng phổ biến cho mục đích gì?**

A. Quản lý phiên người dùng.

B. Tạo giao diện đồ họa người dùng.

**\*C. Logging, user authentication, và transactions.**

D. Tối ưu hóa mã nguồn.

### Question 24

**Khi tối ưu hóa hiệu suất của hệ thống, bạn cần điều chỉnh thành phần nào trước?**

A. Giao diện người dùng (View).

B. AOP

C. Dependency Injection.

**\*D. Data Access Object (DAO).**

### Question 25

**Trong một dự án lớn, bạn nhận thấy rằng các lớp DAO chứa nhiều logic không liên quan đến truy cập dữ liệu. Điều này vi phạm nguyên tắc nào trong Spring Framework và bạn sẽ xử lý như thế nào?**

A. Vi phạm nguyên tắc dependency injection; loại bỏ DAO khỏi hệ thống.

B. Vi phạm nguyên tắc bảo mật; chuyển logic vào controller.

**\*C. Vi phạm nguyên tắc phân tách trách nhiệm; chuyển logic sang service layer.**

D. Vi phạm nguyên tắc caching; thêm bộ nhớ đệm cho DAO.

### Question 26

**Technical Architecture cần được hoàn thành trước giai đoạn nào?**

A. Tạo use case

**\*B. Thiết kế chi tiết**

C. Phân tích độ bền

D. Viết code

### Question 27

**Một trong những lỗi phổ biến nhất trong Technical Architecture là gì?**

A. Phụ thuộc vào DAO cho tất cả các tầng.

B. Sử dụng bảo mật quá mức cần thiết.

**\*C. Không xem xét khả năng mở rộng (scalability).**

D. Chọn framework mới nhất.

### Question 28

**Trong kiến trúc MVC, nếu bạn cần hiển thị dữ liệu theo cách tùy chỉnh nhưng vẫn đảm bảo bảo mật, bạn sẽ làm gì?**

A. Dùng DAO để bảo mật dữ liệu trước khi gửi đến view.

B. Tạo một controller chuyên biệt để xử lý bảo mật cho mỗi loại dữ liệu.

**\*C. Sử dụng AOP để quản lý bảo mật dữ liệu trước khi gửi đến view.**

D. Chuyển toàn bộ dữ liệu thô vào view và xử lý tại đó.

### Question 29

**Mục tiêu chính của Technical Architecture (TA) là gì?**

A. Thiết kế giao diện người dùng chi tiết.

B. Hoàn thành kiểm thử hệ thố

C. Tạo mô hình miền chi tiết cho hệ thống.

**\*D. Xác định cấu trúc hệ thống nhằm đáp ứng các yêu cầu kinh doanh và dịch vụ.**

### Question 30

**Trong hệ thống Spring Framework, bạn nhận thấy DispatcherServlet xử lý sai yêu cầu từ người dùng. Nguyên nhân nào có thể xảy ra, và bạn sẽ kiểm tra thành phần nào trước?**

A. Nguyên nhân: Sai dependency injection; kiểm tra cấu hình view

**\*B. Nguyên nhân: Sai cấu hình mapping; kiểm tra file cấu hình DispatcherServlet.**

C. Nguyên nhân: Sai logic kinh doanh; kiểm tra các lớp DAO.

D. Nguyên nhân: Sai caching; kiểm tra các transaction trong AOP.

Bài 8

### Question 1

**Mục tiêu chính của việc vẽ Sequence Diagrams là gì?**

A. Thể hiện cách dữ liệu được lưu trữ trong cơ sở dữ liệu.

B. Mô tả hành vi của từng đối tượng riêng lẻ.

C. Thể hiện cấu trúc tĩnh của hệ thống.

**\*D. Mô tả cách các đối tượng tương tác với nhau trong thời gian thực.**

### Question 2

**Sequence diagrams có mối quan hệ như thế nào với robustness diagrams?**

**\*A. Sequence diagrams mở rộng các controllers trong robustness diagrams thành các thông điệp.**

B. Robustness diagrams không liên quan đến sequence diagrams.

C. Sequence diagrams thay thế robustness diagrams.

D. Sequence diagrams hoàn toàn độc lập với robustness diagrams.

### Question 3

**Một use case thường cần bao nhiêu sequence diagrams?**

A. Không bắt buộc phải có sequence diagrams.

B. Một sequence diagram cho mỗi lớp trong hệ thống.

C. Chỉ cần một sequence diagram cho toàn bộ hệ thống.

**\*D. Mỗi use case cần một sequence diagram, bao gồm cả các kịch bản chính và thay thế.**

### Question 4

**Lợi ích chính của việc đặt text của use case bên cạnh sequence diagram là gì?**

A. Tăng tính thẩm mỹ cho diagram.

**\*B. Đảm bảo rằng thiết kế phù hợp với yêu cầu hành vi của hệ thống.**

C. Làm cho sequence diagram trở nên dễ đọc hơn.

D. Giảm thời gian phân tích yêu cầu.

### Question 5

**Các đường thẳng đứng (lifelines) trên sequence diagrams thể hiện điều gì?**

**\*A. Thời gian tồn tại của các đối tượng trong hệ thống.**

B. Các lỗi logic trong hệ thống.

C. Dữ liệu đang được truyền qua các lớp.

D. Cấu trúc tĩnh của đối tượng.

### Question 6

**Trong ICONIX Process, bước nào là bước chuẩn bị cuối cùng trước khi bắt đầu vẽ sequence diagrams?**

A. Cải tiến sơ đồ trình tự

B. Kiểm tra mô hình tĩnh

**\*C. Phân tích độ bền (Robustness analysis).**

D. Tinh chỉnh sơ đồ lớp

### Question 7

**Điều gì cần được kiểm tra trong sequence diagrams để đảm bảo tính chính xác của thiết kế?**

A. Các boundary objects có đúng tên không.

**\*B. Các thông điệp phù hợp với các use case text.**

C. Số lượng đối tượng trong mỗi diagram

D. Tính tương thích với cơ sở dữ liệu.

### Question 8

**Tại sao không nên vẽ flowcharts trên sequence diagrams?**

**\*A. Flowcharts không thể hiện được việc phân bổ hành vi giữa các lớp.**

B. Flowcharts không thích hợp cho các hệ thống lớn.

C. Flowcharts không được UML hỗ trợ.

D. Flowcharts làm mất sự liên kết giữa các use case và thiết kế.

### Question 9

**Nếu một use case có nhiều controllers nhưng các thông điệp không được chuyển đổi chính xác trên sequence diagrams, điều gì sẽ xảy ra?**

A. Thiết kế sẽ hoàn chỉnh nhưng không tối ưu

**\*B. Phân bổ hành vi sẽ không rõ ràng, gây khó khăn khi thiết kế chi tiết.**

C. Entity objects không thể xử lý dữ liệu đúng cách.

D. Các boundary objects sẽ không thể giao tiếp.

### Question 10

**Nếu một sequence diagram trở nên quá phức tạp với nhiều đối tượng và thông điệp, bạn nên làm gì để cải thiện?**

A. Chỉ tập trung vào flow chính và bỏ qua các chi tiết nhỏ.

**\*B. Chia nhỏ sequence diagram thành các diagram nhỏ hơn theo từng kịch bản.**

C. Gộp tất cả các entity objects thành một nhóm.

D. Loại bỏ một số boundary objects.

### Question 11

**Khi một use case có các boundary objects chưa được xác định rõ, bước đầu tiên bạn cần thực hiện là gì?**

**\*A. Xác định lại boundary objects trong robustness diagram.**

B. Bỏ qua boundary objects và tập trung vào controllers.

C. Chuyển các boundary objects thành entity objects.

D. Thêm các boundary objects giả định vào sequence diagram.

### Question 12

**Nếu một thông điệp trong sequence diagram không khớp với mô tả trong use case text, bạn sẽ làm gì?**

A. Chuyển thông điệp thành một controller mới.

B. Tạo một use case mới để phù hợp với thông điệp.

C. Loại bỏ thông điệp này khỏi sequence diagram.

**\*D. Chỉnh sửa sequence diagram để khớp với use case text.**

### Question 13

**Một sequence diagram thiếu các entity objects. Điều này sẽ ảnh hưởng gì đến hệ thống?**

A. Use case sẽ trở nên không cần thiết.

**\*B. Hệ thống sẽ thiếu các phương thức xử lý dữ liệu.**

C. Các controllers sẽ không thể hoạt động.

D. Các boundary objects sẽ không hiển thị đúng thông tin.

### Question 14

**Khi mô tả kịch bản thay thế trên sequence diagram, yếu tố nào cần được ưu tiên?**

A. Gộp kịch bản thay thế vào kịch bản chính.

B. Loại bỏ các thông điệp không quan trọng.

**\*C. Rõ ràng về luồng dữ liệu và hành vi.**

D. Tăng số lượng đối tượng trên diagram.

### Question 15

**Một use case có nhiều kịch bản thay thế nhưng sequence diagram chỉ mô tả kịch bản chính. Điều này ảnh hưởng gì đến hệ thống, và bạn sẽ làm gì để cải thiện?**

A. Ảnh hưởng: Boundary objects bị quá tải; cải thiện: Tăng số lượng boundary objects.

B. Ảnh hưởng: Kịch bản chính trở nên phức tạp; cải thiện: Loại bỏ các kịch bản thay thế không quan trọng.

**\*C. Ảnh hưởng: Hệ thống không thể xử lý các trường hợp ngoại lệ; cải thiện: Vẽ thêm sequence diagrams cho các kịch bản thay thế.**

D. Ảnh hưởng: Tăng độ phức tạp của class diagram; cải thiện: Thêm thông điệp vào kịch bản chính.

### Question 16

**Một controller từ robustness diagram thường được chuyển đổi thành gì trên sequence diagram?**

A. Một lớp dữ liệu.

B. Một thuộc tính của entity object.

C. Một boundary object mới.

**\*D. Một phương thức hoặc một tập hợp phương thức.**

### Question 17

**Khi nào cần chia nhỏ sequence diagram thành nhiều diagram khác nhau?**

A. Khi thiết kế chỉ có một kịch bản chính.

B. Khi không có thời gian để hoàn thành một diagram duy nhất.

C. Khi các boundary objects quá lớn.

**\*D. Khi use case quá phức tạp để thể hiện trong một diagram**

### Question 18

**Khi nào cần cập nhật class diagram dựa trên thông tin từ sequence diagram?**

A. Khi các thông điệp trên sequence diagram quá ngắn.

**\*B. Khi một phương thức mới được xác định trong sequence diagram.**

C. Khi một boundary object không có trong class diagram.

D. Khi một kịch bản thay thế không được bao phủ trong sequence diagram.

### Question 19

**Tại sao việc kiểm tra lifelines trên sequence diagram lại quan trọng?**

A. Để tăng tính thẩm mỹ của diagram.

B. Để xác định kịch bản thay thế

C. Để giảm số lượng thông điệp trong diagram.

**\*D. Để đảm bảo rằng các đối tượng tồn tại đúng thời gian cần thiết.**

### Question 20

**Một sequence diagram cho thấy một controller gửi thông điệp đến các đối tượng khác nhưng không có thông điệp trả về. Điều này có thể gây ra vấn đề gì trong thiết kế, và bạn sẽ xử lý như thế nào?**

A. Vấn đề: Controller quá tải; xử lý: Chuyển thông điệp sang boundary object.

B. Vấn đề: Mất tính bảo mật; xử lý: Dùng AOP để bảo vệ thông điệp.

**\*C. Vấn đề: Các phương thức không được hoàn thiện; xử lý: Thêm thông điệp trả về từ các đối tượng nhận.**

D. Vấn đề: Thiếu entity object; xử lý: Tạo thêm các entity object.

### Question 21

**Các thành phần chính trong một sequence diagram là gì?**

A. Actors, controllers, và dữ liệu.

B. Boundary objects, entity objects, và control objects.

**\*C. Actors, boundary objects, và messages.**

D. Boundary objects, use case text, và database models.

### Question 22

**Mối quan hệ giữa class diagrams và sequence diagrams là gì?**

A. Sequence diagrams thay thế class diagrams trong thiết kế chi tiết.

**\*B. Sequence diagrams hỗ trợ cập nhật và tinh chỉnh class diagrams.**

C. Sequence diagrams không liên quan đến class diagrams.

D. Sequence diagrams chỉ hỗ trợ việc kiểm tra use case.

### Question 23

**Nếu một boundary object không nhận được thông điệp cần thiết trên sequence diagram, điều gì nên được kiểm tra?**

A. Các lifelines để tăng thời gian tồn tại của boundary object.

B. Class diagram để xác minh tên các đối tượng.

**\*C. Text của use case để đảm bảo các hành vi được mô tả đầy đủ.**

D. Controllers để thêm thông điệp mới.

### Question 24

**Làm thế nào để kiểm tra rằng sequence diagram tuân thủ UML?**

**\*A. Đảm bảo rằng tất cả các thông điệp được đặt tên và có hướng rõ ràng.**

B. Giảm số lượng lifelines xuống tối thiểu.

C. Loại bỏ tất cả các boundary objects không cần thiết.

D. Tăng số lượng controllers trong diagram.

### Question 25

**Nếu một sequence diagram thể hiện một boundary object thực hiện quá nhiều logic xử lý, bạn sẽ xử lý như thế nào?**

**\*A. Chuyển logic xử lý từ boundary object sang controller.**

B. Gộp boundary object với entity object để giảm độ phức tạp.

C. Xóa các thông điệp không cần thiết.

D. Tăng số lượng boundary objects để giảm tải.

### Question 26

**Khi vẽ sequence diagrams, làm thế nào để đảm bảo rằng tất cả các hành vi quan trọng của hệ thống được bao phủ?**

A. Loại bỏ các entity objects không cần thiết.

B. Chỉ tập trung vào kịch bản chính của use case.

**\*C. Liệt kê tất cả các kịch bản chính và kịch bản thay thế trong use case.**

D. Sử dụng các boundary objects để thay thế controllers.

### Question 27

**Nếu các thông điệp trên sequence diagram quá phức tạp, bạn sẽ làm gì để làm rõ hơn?**

**\*A. Sử dụng các phương thức trừu tượng để đơn giản hóa thông điệp.**

B. Gộp tất cả thông điệp lại thành một phương thức.

C. Tăng số lượng boundary objects.

D. Loại bỏ các thông điệp không cần thiết.

### Question 28

**Trong ICONIX Process, mục tiêu nào không phải là mục tiêu chính của sequence diagrams?**

**\*A. Chuẩn hóa cấu trúc dữ liệu.**

B. Hoàn thiện phân bổ phương thức giữa các lớp.

C. Thể hiện cách các lớp tương tác trong suốt vòng đời của use case.

D. Phân bổ hành vi cho các lớp.

### Question 29

**Điều gì xảy ra nếu các controllers không được chuyển đổi thành thông điệp trên sequence diagrams?**

A. Tăng độ phức tạp của domain model.

**\*B. Thiết kế sẽ thiếu các phương thức cần thiết.**

C. Không ảnh hưởng gì đến thiết kế.

D. Làm giảm hiệu quả của các use case.

### Question 30

**Một sequence diagram không có bất kỳ boundary object nào. Điều này sẽ ảnh hưởng như thế nào đến thiết kế?**

A. Các entity objects sẽ không thể lưu trữ dữ liệu.

**\*B. Hệ thống sẽ thiếu giao diện để tương tác với người dùng.**

C. Diagram sẽ vẫn hoạt động bình thường.

D. Các controllers sẽ không thể xử lý logic.