### 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.