

Câu 1:

- a.
- b.
- c.
- d.

Câu 2: Theo quy trình phát triển phần mềm hợp nhất RUP (Rational Unified Process), cách tiến hành các workflow là theo cơ chế:

- a. Tuần tự
- b. Song song
- ☒ c. Lập theo pipe-line
- d. Tùy theo sở thích

Câu 3: Ta sử dụng “hoạt động lặp” trong khi thực hiện:

- a. Từng workflow
- b. Từng bước nhỏ trong workflow
- c. Từng chu kỳ phát triển phần mềm
- ☒ d. Cả 3 câu trên đều đúng

Câu 4: UML cho phép:

- a. Đặc tả và xây dựng các artifacts cho 1 hệ thống cần tin học hóa
- b. Lập tài liệu và hiển thị trực quan các artifacts cho 1 hệ thống cần tin học hóa
- ☒ c. Cả 2 câu trên đều đúng
- d. Cả 2 câu trên đều sai

Câu 5: UML cung cấp:

- a. 2 ký hiệu mô tả thông tin cơ bản
- ☒ b. 3 ký hiệu mô tả thông tin cơ bản
- c. Nhiều ký hiệu mô tả thông tin cơ bản
- d. Cả 3 câu trên đều sai

Câu 6: Lược đồ use-case có thể chứa:

- a. Use-case, actor, mối quan hệ giữa các thành phần
- b. Use-case, actor, note
- ☒ c. Use-case, actor, note, mối quan hệ giữa các thành phần
- d. Cả 3 câu trên đều sai

Câu 7: Các mối quan hệ phổ biến giữa các actor là:

- ☒ a. Generalization
- b. Include, extend
- c. Cả 2 câu a và b đều đúng
- d. Cả 2 câu a và b đều sai

Câu 8: Các mối quan hệ phổ biến giữa các use-case là:

- a. Generalization
- ☒ b. Include, extend
- c. Cả 2 câu a và b đều đúng
- d. Cả 2 câu a và b đều sai

Câu 9: Công dụng thiết yếu của mối quan hệ tổng quát giữa 2 actor là:

- a. Giúp ta miêu tả được 2 actor tương ứng có mối quan hệ với nhau
- ☒ b. Cấu trúc lược đồ use-case trong sáng, dễ đọc, dễ hiểu hơn
- c. Giúp ta dễ dàng viết code cho các actor dựa vào tính thừa kế của ngôn ngữ hướng đối tượng
- d. Cả 3 câu trên đều đúng

Câu 10: Công dụng của mối quan hệ extends giữa 2 use-case là:

- a. Giúp ta miêu tả được 2 use-case tương ứng có mối quan hệ với nhau
- b. Cấu trúc lược đồ use-case trong sáng, dễ đọc, dễ hiểu hơn
- c. Giúp ta lập kế hoạch thực hiện các build
- ☒ d. Cả 3 câu trên đều đúng

Câu 11: Lược đồ use-case cho thấy:

- a. Toàn bộ chức năng của hệ thống phần mềm
- b. Toàn bộ chức năng của hệ thống phần mềm và actor
- c. Một số chức năng của hệ thống phần mềm
- ☒ d. Một số chức năng của hệ thống phần mềm và một số actor tương tác với các chức năng này

Câu 12: Mô hình use-case cho thấy:

- a. Toàn bộ các chức năng của hệ thống phần mềm
- ☒ b. Toàn bộ chức năng của hệ thống phần mềm và actor
- c. Một số chức năng của hệ thống phần mềm
- d. Một số chức năng của hệ thống phần mềm và một số actor tương tác với các chức năng này

Câu 13: Lược đồ hoạt động miêu tả được:

- ☒ a. Giải thuật thực hiện chức năng của use-case tương ứng
- b. 1 kịch bản thực hiện chức năng của use-case tương ứng
- c. Sự tương tác giữa các đối tượng
- d. Sự tương tác giữa các actor và các đối tượng cũng như giữa các đối tượng

Câu 14: Lược đồ tuần tự miêu tả được:

- a. Giải thuật thực hiện chức năng của use-case tương ứng
- ☒ b. 1 kịch bản thực hiện chức năng của use-case tương ứng
- c. Sự tương tác giữa các đối tượng
- d. Sự tương tác giữa các actor và các đối tượng cũng như giữa các đối tượng

Câu 15: Lược đồ cộng tác miêu tả được:

- a. Giải thuật thực hiện chức năng của use-case tương ứng
- b. 1 kịch bản thực hiện chức năng của use-case tương ứng
- c. Sự tương tác giữa các đối tượng
- ☒ d. Sự tương tác giữa các actor và các đối tượng cũng như giữa các đối tượng

Câu 16: Mô hình use-case được xây dựng ở workflow:

- ☒ a. Nắm bắt yêu cầu
- b. Phân tích
- c. Thiết kế
- d. Hiện thực

Câu 17: Lược đồ class phân tích có thể chứa:

- a. Use-case, actor, note, mối quan hệ giữa các thành phần
- ☒ b. Actor, class, interface và mối quan hệ giữa chúng
- c. Actor, class, đối tượng và mối quan hệ giữa chúng
- d. Cả 3 câu trên đều sai

Câu 18: Ở cấp phân tích:

- a.
- b.
- c.
- d.

Câu 19: Mỗi class phân tích có thể:

- ☒ a. Tham gia thực hiện 1 chức năng nào đó của phần mềm
- b. Thực hiện 1 yêu cầu không chức năng nào đó của phần mềm
- c. Cả 2 câu trên đều đúng
- d. Cả 2 câu a và b đều sai

Câu 20: Ở cấp phân tích, để miêu tả hành vi động của các đối tượng phân tích, ta thường dùng:

- ☒ a. Lược đồ cộng tác
- b. Lược đồ trình tự
- c. Lược đồ hoạt động
- d. Cả 3 câu trên đều đúng

Câu 21: Mối quan hệ giữa 2 mô hình phân tích và mô hình use-case là:

- a. Vi mô và vĩ mô
- b. Tổng quát hóa
- ☒ c. Hiện thực
- d. Cả 3 câu trên đều đúng

Câu 22: Trong hoạt động tìm actor và use-case của workflow nắm bắt yêu cầu:

- ☒ a. Ta nên tìm actor trước rồi tìm use-case dựa trên actor tìm được
- b. Ta nên tìm actor và use-case đồng thời bởi 2 nhóm khác nhau để hiệu quả
- c. Ta nên tìm use-case trước rồi tìm actor dựa trên use-case tìm được
- d. Ta nên làm theo sở thích của mình để được thoải mái, nhờ đó đạt được kết quả tốt nhất

Câu 23: Theo quy trình phát triển phần mềm RUP, kiến trúc của hệ thống phần mềm là:

- ☒ a. Các module chứa chức năng cấu thành phần mềm và mối quan hệ giữa chúng
- b. Các góc nhìn (view) của các mô hình
- c. Các mô hình
- d. Cả 3 câu trên đều đúng

Câu 24: Việc đặc tả chi tiết thông tin về use-case là của:

- a. Kiến trúc sư
- b. Người phân tích hệ thống phần mềm
- ☒ c. Người đặc tả use-case
- d. Cả 3 worker trên phải hợp tác với nhau

Câu 25: Việc phân tích 1 class là của:

- ☒ a. Người phân tích hệ thống phần mềm
- b. Kỹ sư thành phần (component engineer)
- c. Kỹ sư use-case (use-case engineer)
- d. Cả 3 worker trên phải hợp tác với nhau

Câu 26: Phát họa prototype giao diện với người dùng được làm trong:

- ☒ a. Hoạt động nắm bắt yêu cầu của phần mềm
- b. Hoạt động phân tích yêu cầu phần mềm
- c. Hoạt động thiết kế
- d. Hoạt động viết code

Câu 27: Worker nào xuất hiện trong hầu hết các hoạt động (workflow) phát triển phần mềm:

- a. Chuyên gia lĩnh vực
- b. Phân tích viên
- ☒ c. Kiến trúc sư phần mềm
- d. Kỹ sư use-case

Câu 28: Các package phân tích cần có:

- a. Tính kết dính thấp (cohesion) và tính nối ghép thấp (coupling)
- b. Tính kết dính thấp (cohesion) và tính nối ghép cao (coupling)
- ☒ c. Tính kết dính cao (cohesion) và tính nối ghép thấp (coupling)
- d. Tính kết dính cao (cohesion) và tính nối ghép cao (coupling)

Câu 29: Mô hình nào có thể chứa chi tiết dư thừa, không nhất quán:

- ☒ a. Mô hình use-case
- b. Mô hình phân tích
- c. Mô hình thiết kế
- d. Mọi mô hình đều phải đúng đắn, nhất quán và không thể chứa chi tiết dư thừa

Câu 30: Trong mối quan hệ kết hợp (association) giữa 2 class A và B thì "role name" ở đầu B sẽ:

- a. Trở thành tên thuộc tính trong class B
- b. Trở thành tên thuộc tính trong class A
- c. Trở thành tên thuộc tính trong class B và điều này thường được làm tự động bởi tiện ích hỗ trợ phát triển phần mềm
- ☒ d. Trở thành tên thuộc tính trong class A và điều này thường được làm tự động bởi tiện ích hỗ trợ phát triển phần mềm

Câu 31: Để miêu tả chu trình sống của một đối tượng, ta dùng:

- a. Lược đồ hoạt động (activity diagram)
- ☒ b. Lược đồ "statechart"
- c. Lược đồ trình tự (sequence diagram)
- d. Cả 3 câu trên đều sai

Câu 32: Mục đích của lược đồ thành phần (component diagram) là:

- a. Xây dựng 1 phiên bản phần mềm chạy được
- b. Đặc tả database vật lý
- c. Tổ chức source code thành những thành phần và mối liên hệ giữa chúng
- ☒ d. Cả 3 câu trên đều đúng

Câu 33: Mục đích của các cơ chế nối rộng (extensibility mechanism) là:

- a. Giảm thiểu số lượng cấu tử cơ bản mà vẫn miêu tả được nhiều loại phần tử khác nhau
- b. Nối rộng một cấu tử mô hình thành nhiều phần tử khác nhau
- c. Làm phức tạp thêm cho ngôn ngữ UML
- ☒ d. Chỉ có câu a và b đúng

Câu 34: Lược đồ class thiết kế chứa:

- ☒ a. Các class thiết kế và các mối quan hệ giữa chúng
- b. Các đối tượng thiết kế và các tương tác giữa chúng
- c. Cả a và b đều đúng
- d. Cả a và b đều sai

Câu 35: Thuộc tính chất lượng phần mềm nào không ở thời điểm chạy:

- a. Availability
- ☒ b. Modifiability
- c. Performace
- d. Cả 3 câu trên đều sai

Câu 36: Yêu cầu về “security” là:

- a. Yêu cầu chức năng của phần mềm
- b. Thuộc tính chất lượng của phần mềm
- ☒ c. Tùy theo quan điểm phân loại mà chọn a hay b
- d. Cả 3 câu trên đều sai

Câu 37: Nội dung “thị trường về phần mềm sẽ tăng rất mạnh trong 8 tháng tới” đòi hỏi phần mềm cần có tính:

- a. Modifiability
- ☒ b. Buildability
- c. Usability
- d. Cả 3 câu trên đều sai

Câu 38: Để đặc tả 1 yêu cầu không chức năng, ta dùng:

- a. Bảng đặc tả use-case
- ☒ b. Kịch bản thuộc tính chất lượng
- c. Cả 2 câu a và b đều đúng
- d. Cả 2 câu a và b đều sai

Câu 39: User có thể quan sát thấy yêu cầu phi chức năng nào:

- a. Scalability
- b. Modifiability
- c. Buildability
- ☒ d. Cả 3 câu trên đều sai

Câu 40: Yêu cầu phi chức năng nào ảnh hưởng lớn nhất đến chi phí thay đổi phần mềm:

- a. Availability
- b. Usability
- ☒ c. Modifiability
- d. Cả 3 câu trên đều đúng

Câu 41: Kiểm thử (Testing) là:

- ☒ a. Chạy phần mềm trên máy với dữ liệu giả hầu tìm lỗi của phần mềm
- b. Xem xét mã nguồn để tìm lỗi phần mềm
- c. Cả a và b đều đúng
- d. Cả a và b đều sai

Câu 42: kiểm tra tĩnh (Inspection) là:

- ☒ a. Xem xét tài liệu nào đó về phần mềm để tìm lỗi phần mềm
- b. Xem xét mã nguồn để tìm lỗi phần mềm
- c. Chạy phần mềm thủ công bằng con người để tìm lỗi phần mềm
- d. Cả 3 câu trên đều sai

Câu 43: Quy trình V&V bao gồm:

- a. Testing phần mềm
- b. Inspection phần mềm
- c. Kiểm tra việc dùng phần mềm bởi người dùng
- ☒ d. Cả 3 câu trên đều đúng

Câu 44: Kỹ thuật nào được dùng chủ yếu trong quy trình V&V:

- ☒ a. Kiểm thử hộp đen
- b. Kiểm thử hộp trắng
- c. Kiểm thử hộp xám
- d. Cả 3 câu trên đều đúng

Câu 45: Đường thi hành phần mềm là:

- ☒ a. 1 kịch bản thi hành cụ thể, bắt đầu từ lệnh đầu tiên cho đến lệnh làm kết thúc phần mềm
- b. Là thuật giải mà phần mềm thi hành
- c. Cả a và b đều đúng
- d. Cả a và b đều sai

Câu 46: Vì sao ta không thể kiểm thử hết tất cả đường thi hành phần mềm:

- a. Vì số lượng thường quá nhiều
- b. Vì có đường quá dài
- ☒ c. Cả a và b đều đúng
- d. Cả a và b đều sai

Câu 47: Chọn phát biểu đúng:

- a. Phủ các lệnh thường tốt hơn phủ các nhánh
- b. Phủ các lệnh và phủ các nhánh tương đương nhau
- ☒ c. Phủ các nhánh thường tốt hơn phủ các lệnh
- d. Cả 3 câu trên đều sai

Câu 48: Khi kiểm thử đơn vị, hoạt động nào cần thực hiện nhất:

- a. Kiểm thử giao tiếp sử dụng đơn vị đó
- ☒ b. Kiểm thử các đường thi hành của đơn vị
- c. Kiểm thử đời sống các biến được dùng
- d. Cả 3 câu trên đều sai

Câu 49: Khi kiểm thử thành phần, hoạt động nào cần thực hiện nhất:

- ☒ a. Kiểm thử giao tiếp sử dụng thành phần đó
- b. Kiểm thử các đường thi hành của thành phần
- c. Kiểm thử đời sống các biến được dùng
- d. Cả 3 câu trên đều sai

Câu 50: Khi kiểm thử hệ thống phần mềm, hoạt động nào cần thực hiện nhất:

- a. Kiểm thử giao tiếp sử dụng với người dùng
- b. Kiểm thử đời sống các biến được dùng
- ☒ c. Kiểm thử các yêu cầu phi chức năng của phần mềm
- d. Cả 3 câu trên đều sai

Câu 51: Để dừng và thoát bất thường khỏi 1 hàm chức năng, ta nên dùng:

- a. Lệnh return
- ☒ b. Lệnh throw
- c. Cả 2 câu a và b đều đúng
- d. Cả 2 câu a và b đều sai

Câu 52: Để đặt tên cho 1 hằng gọi nhớ, ta nên:

- a. Dùng tiếp đầu ngữ "C" để nhấn mạnh nó là Constant
- ☒ b. Dùng các ký tự chữ hoa
- c. Dùng các ký tự chữ thường
- d. Đặt tên thoải mái miễn sao gợi ý tốt nhất đến ý nghĩa của hằng đó

Câu 53: Để đặt tên cho 1 interface của đối tượng, ta nên:

- a. Dùng tiếp đầu ngữ "C" để nhấn mạnh nó là Class
- ☒ b. Dùng tiếp đầu ngữ "I" để nhấn mạnh nó là Interface
- c. Dùng tiếp đầu ngữ "O" để nhấn mạnh nó là Object
- d. Đặt tên thoải mái miễn sao gợi ý tốt nhất đến ý nghĩa của Interface đó

Câu 54: Để đặt tên cho 1 interface của đối tượng, ta nên:

- a. Dùng tiếp đầu ngữ "C" để nhấn mạnh nó là Class
- ☒ b. Dùng tiếp đầu ngữ "I" để nhấn mạnh nó là Interface
- c. Dùng tiếp đầu ngữ "O" để nhấn mạnh nó là Object
- d. Đặt tên thoải mái miễn sao gợi ý tốt nhất đến ý nghĩa của Interface đó

Câu 55: Để đặt tên cho 1 thành phần static của class, ta nên:

- a. Dùng tiếp đầu ngữ "@"
- b. Dùng tiếp đầu ngữ "S" để nhấn mạnh nó là static
- ☒ c. Dùng tiếp đầu ngữ "_"
- d. Đặt tên thoải mái miễn sao gợi ý tốt nhất đến ý nghĩa của thành phần static đó

Câu 56: Để đặt tên cho 1 biến cục bộ trong 1 hàm, ta nên:

- ☒ a. Dùng qui định ký hiệu Hungarian
- b. Dùng tiếp đầu ngữ "_" để nhấn mạnh nó là nội bộ
- c. Dùng tiếp đầu ngữ "in_" để nhấn mạnh nó là nội bộ
- d. Đặt tên thoải mái miễn sao gợi ý tốt nhất đến ý nghĩa của biến cục bộ đó

Câu 57: Trong giai đoạn coding phần mềm, ta sẽ:

- ☒ a. Kiểm thử đơn vị
- b. Kiểm thử thành phần
- c. Kiểm thử tổng thể phần mềm
- d. Cả 3 câu trên đều sai

Câu 58: Lệnh chú thích nên được dùng:

- a. Ở đầu mỗi tác vụ chức năng
- b. Ở câu lệnh hay đoạn lệnh thực hiện giải thuật phức tạp
- c. Ở câu lệnh gây hiệu ứng lề nguy hiểm
- ☒ d. Cả 3 câu trên đều đúng

Câu hỏi trắc nghiệm ôn thi cuối kỳ (2)

Câu 28: Để quản lý danh sách có thứ tự các phần tử, ta có thể dùng:

- a. Class tổng quát hóa HashSet<T>
- ☒ b. Class tổng quát hóa List<T>
- c. Tùy sở thích mà dùng a hay b
- d. Cả 3 câu trên đều không giải quyết được yêu cầu

Câu 29: Trong các mẫu thiết kế đã học, mẫu nào cung cấp 2 loại class pattern và object pattern:

- a. Composite
- ☒ b. Adapter
- c. Proxy
- d. Façade

Câu 30: Trong đại đa số mẫu thiết kế, phần tử cha đóng vai trò như

- a. 1 abstract class
- ☒ b. 1 interface sử dụng
- c. 1 class tổng quát hóa
- d. Cả 3 câu trên đều sai

Câu 31: Để tạo 1 ứng dụng multi-platform dễ dàng, dễ nâng cấp, ta nên dùng mẫu thiết kế:

- a. Adapter
- b. Abstract Factory
- ☒ c. Kết hợp các mẫu thiết kế trên
- d. Cả 3 câu trên đều sai

Câu 32: Mẫu thiết kế phục vụ việc thêm động trách nhiệm cho đối tượng:

- a. Virtual Proxy
- b. Remote Proxy
- c. Protection Proxy
- ☒ d. Smart Proxy

Câu 33: Chọn câu đúng nhất:

- a. Mẫu Smart Proxy là trường hợp đặc biệt của mẫu Decorator
- ☒ b. Mẫu Decorator là trường hợp đặc biệt của mẫu Smart Proxy
- c. 2 mẫu Smart Proxy và Decorator tương đương nhau
- d. Cả 3 câu trên đều sai

Câu 34: Khi client muốn tương tác với một họ trong tập hợp các họ đối tượng và việc chọn họ đối tượng được xác định tại thời điểm run-time, ta dùng mẫu thiết kế:

- a. Composite
- b. Template method
- ☒ c. Abstract Factory
- d. Factory method

Câu 35: Để một class chuyển quá trình thực hiện một nhiệm vụ nào đó cho một trong các class con nhưng cho phép ứng dụng xác định class con cụ thể, ta dùng mẫu thiết kế:

- a. Template method
- b. Factory method
- ☒ c. Cả 2 câu a và b đều đúng
- d. Cả 2 câu a và b đều sai

Câu 36: Mối quan hệ giữa các thành phần trong các mẫu thiết kế là:

- a. Hiện thực và thừa kế
- b. Kết hợp (association)
- c. Bao gộp
- ☒ d. Cả 3 câu trên đều đúng

Câu 37: Khi hệ thống cần độc lập với các đối tượng mà nó sinh ra và các đối tượng sinh ra thuộc cùng một class và có rất ít điểm khác biệt, ta dùng mẫu thiết kế:

- a. Abstract factory
- b. Method Factory
- ☒ c. Prototype
- d. Builder

Câu 38: Muốn xác định động tại thời điểm chạy tập các đối tượng xử lý 1 request nào đó, ta nên dùng mẫu thiết kế:

- a. Proxy
- b. Composite
- ☒ c. Chain of responsibility
- d. Cả 3 câu trên đều đúng

Câu 39: Muốn tập trung các hành vi giống nhau của các class để tránh trùng lặp, ta nên dùng mẫu thiết kế:

- a. Prototype
- b. Strategy
- ☒ c. Template method
- d. Factory method

Câu 40: Để giúp client tạo đối tượng phức hợp theo cơ chế tích lũy tăng dần, ta dùng mẫu thiết kế:

- a. Abstract Factory
- b. Factory method
- c. Prototype
- ☒ d. Builder

Câu 41: Muốn tránh việc dùng các lệnh kiểm tra số học có nhiều khả năng xảy ra, ta nên dùng mẫu thiết kế:

- a. State
- b. Strategy
- ☒ c. Cả 2 câu a và b đều đúng
- d. Cả 2 câu a và b đều sai

Câu 42: Muốn định nghĩa sự phụ thuộc 1-n giữa các đối tượng sao cho khi 1 đối tượng thay đổi trạng thái thì các đối tượng phụ thuộc được cảnh báo dấu hiệu chỉnh tự động, ta dùng mẫu thiết kế:

- a. Command
- b. Chain of responsibility
- ☒ c. Observer
- d. State

Câu 43: Mỗi mẫu thiết kế thường chứa:

- a. 1 vài loại thành phần
- ☒ b. Từ 3-5 loại thành phần
- c. Từ 4-10 loại thành phần
- d. Trên 10 loại thành phần

Câu 44: Đa số các mẫu thiết kế thuộc nhóm "Creational" thuộc loại

- ☒ a. Class pattern
- b. Object pattern
- c. Class pattern lẫn object pattern
- d. Cả 3 câu trên đều sai

Câu 45: Mẫu thiết kế Command thuộc loại:

- a. Class pattern
- ☒ b. Object pattern
- c. Class pattern lẫn object pattern
- d. Cả 3 câu trên đều sai

Câu 46: Để tạo ứng dụng cho phép người dùng dễ dàng cá nhân hóa hệ thống menu-bar, ta nên dùng mẫu thiết kế:

- a. Composite
- b. Adapter
- ☒ c. Command
- d. Observer

Câu 47: Trong ứng dụng cần tạo nhiều số ngẫu nhiên ở nhiều nơi, để đảm bảo ứng dụng chỉ dùng được 1 đối tượng tạo số ngẫu nhiên duy nhất, ta dùng mẫu thiết kế:

- a. Composite
- b. Adapter
- c. Command
- ☒ d. Singleton

Câu 48: Thường hệ thống đã cung cấp đối tượng tạo số ngẫu nhiên, để đặc tả đối tượng tạo số ngẫu nhiên theo interface của ứng dụng được dễ dàng, ta dùng mẫu thiết kế:

- a. Composite
- ☒ b. Adapter
- c. Command
- d. Singleton

Câu 49: Class Singleton cần phải có tính chất nào sau đây:

- a. Hàm khởi tạo phải có tầm vực private
- b. Chứa tác vụ static có nhiệm vụ trả về tham khảo đến đối tượng duy nhất của class
- c. Chứa thuộc tính static tham khảo đến đối tượng duy nhất của class
- ☒ d. Cả 3 câu trên đều đúng

Câu 50: Biểu thức gồm nhiều phần tử cấu thành như hằng, biến, giá trị cụ thể, ... Để đặc tả các thành phần cấu thành biểu thức sao cho cùng cung cấp interface duy nhất cho ứng dụng, ta dùng mẫu thiết kế:

- ☒ a. Composite
- b. Adapter
- c. Command
- d. Chain of responsibility

Câu 51: Mẫu thiết kế nào có sử dụng abstract class:

- a. Adapter & Composite
- b. Composite & Proxy
- c. Proxy & Abstract Factory
- ☒ d. Abstract Factory & Factory Method

Câu 52: Mẫu thiết kế nào có sử dụng mối quan hệ bao gộp:

- a. Adapter & Composite
- ☒ b. Composite & Observer
- c. Observer & Strategy
- d. Strategy & State

Câu 53: Mẫu thiết kế nào có sử dụng mối quan hệ kết hợp (association):

- a. Adapter & Composite
- b. Composite & Observer
- c. Observer & Strategy
- ☒ d. Adapter & Proxy

Câu 54: Hai mối quan hệ có mẫu thiết kế hỗ trợ là:

- a. Kết hợp & bao gộp
- ☒ b. Bao gộp & 1-n
- c. 1-n & hiện thực
- d. Hiện thực & thừa kế

Câu 55: Để quản lý 1 họ đối tượng được dùng trong phần mềm cho mỗi lần chạy, ta có thể dùng mẫu thiết kế:

- a. Abstract Factory
- b. Factory Method
- ☒ c. Cả 2 câu a và b đều đúng
- d. Cả 2 câu a và b đều sai

Câu 56: Điều kiện cần và đủ để 1 class trở thành Singleton là:

- a. Tất cả các constructor đều private
- ☒ b. Ít nhất 1 constructor không tham số và phải private
- c. Không chứa bất kỳ constructor nào
- d. Cả 3 câu trên đều sai

Câu 57: Điều kiện cần và đủ để 1 class trở thành Prototype là:

- ☒ a. Phải override tác vụ Clone()
- b. Không cần làm vì hết class Object đã có tác vụ Clone()
- c. Phải định nghĩa mới tác vụ Clone()
- d. Cả 3 câu trên đều sai

Câu 58: Tác vụ getInstance() của class Singleton cần phải có tính chất:

- a. Public
- b. Static
- ☒ c. Cả 2 câu a và b đều đúng
- d. Cả 2 câu a và b đều sai

Câu 59: Môi trường .Net hỗ trợ trực tiếp mẫu thiết kế:

- a. Singleton
- ☒ b. Prototype
- c. Strategy
- d. Observer

Câu 60: Trong class Singleton, lệnh new 1 đối tượng thuộc class Singleton:

- a. Phải xuất hiện chỉ 1 lần
- b. Phải nằm ngoài thân của tất cả các tác vụ
- ☒ c. Cả 2 câu a và b đều đúng
- d. Cả 2 câu a và b đều sai