

**Câu 26:** Đơn vị đo lường đời sống 1 phần mềm là:

- a. Bước lặp
- ☒ b. Chu kỳ
- c. Năm
- d. Công đoạn (phase)

**Câu 27:** Theo qui trình RUP, cách tiến hành các workflow là:

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

**Câu 28:** 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 29:** UML cho phép:

- a. Đặc tả và xây dựng các artifacts cho 1 hệ thống cần tin học hóa.
- b. Lặp 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 30:** UML cung cấp:

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

**Câu 31:** 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 32:** 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 33:** 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 34:** Công dụng thiết yếu của mối quan hệ tổng quát hóa 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 chúng 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 35:** Công dụng thiết yếu của mối quan hệ extends giữa 2 use-case là:

- a. Giúp chúng 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 buid
- ☒ d. Cả 3 câu trên đều đúng

**Câu 36:** 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 các 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 37:** Mô hình 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 các 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 38:** 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 39:** 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 40:** 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 41:** Mô hình use-case được xây dựng ở workflow

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

**Câu 42:** 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 43:** Ở cấp phân tích:

- a. Chỉ có 1 loại class phân tích
- b. Chỉ có 2 loại class phân tích
- ☒ c. Chỉ có 3 loại class phân tích
- d. Có thể có nhiều loại class phân tích tùy vào chức năng cần tích

**Câu 44:** 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 a và b đều đúng
- d. Cả 2 câu a và b đều sai

**Câu 45:** 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 46:** Mỗi quan hệ giữa 2 mô hình phân tích và mô hình use-case là:

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

**Câu 47:** 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 48:** Theo qui trình phát triển phần mềm RUP (Rational Unified Process), 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 49:** 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 50:** 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 hỏi trắc nghiệm ôn thi giữa kỳ (2)

**Câu 15:** Công nghệ phần mềm được đặc trưng bởi các yếu tố:

- ☒ a. Qui trình, phương pháp và công cụ
- b. Kỹ thuật, công cụ và con người
- c. Lý thuyết, phương pháp và công cụ
- d. Con người, thời đại và công cụ

**Câu 16:** Phát họa prototype giao diện với người dùng được làm trong:

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

**Câu 17:** 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 18:** Để miêu tả hành vi động của các đối tượng phân tích, ta dùng chủ yếu

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

**Câu 19:** Các package phân tích 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 20:** 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 21:** ActiveX Control là:

- a. Đối tượng phần mềm
- ☒ b. Module phần mềm
- c. Đối tượng phần mềm có thể tương tác trực tiếp với người dùng
- d. Phần mềm giao tiếp trực tiếp với user và chỉ chạy được trên Windows

**Câu 22:** Mô hình usecase nghiệp vụ (Business usecase model):

- a. Nhất thiết phải được xây dựng cho bất kỳ việc phát triển phần mềm nào
- b. Nên được xây dựng cho các chương trình liên quan đến nghiệp vụ
- c. Không cần thiết trong quá trình .....
- d.

**Câu 23:**

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

**Câu 24:** 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 được làm tự động bởi Rose 2000
- d. Trở thành tên thuộc tính trong class A và điều này được làm tự động bởi Rose 2000

**Câu 25:** Đặc tính round-trip trong Rose 2000 cho phép:

- ☒ a. Cập nhật source code từ các thay đổi ở mô hình thiết kế và ngược lại
- b. Cập nhật mô hình thiết kế từ các thay đổi ở mô hình phân tích và ngược lại
- c. Cập nhật mô hình phân tích từ các thay đổi ở mô hình user case và ngược lại
- d. Cả 3 câu trên đều đúng

**Câu 26:** Để miêu tả chu trình sống của một đối tượng, ta dùng:

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

**Câu 27:**

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

**Câu 28:**

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

**Câu 29:** Mục đích của biểu 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 30:** Một thành phần trong đối tượng Java có thuộc tính truy xuất default là:

- a. Private
- b. Protected
- c. Public
- ☒ d. Friendly

**Câu 31:** Tính đa hình được hiện thực bởi:

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

**Câu 32:**

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

**Câu 33:**

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

**Câu 34:** Mục đích sử dụng cơ chế nới 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. Nới 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 35:** Để thêm các component trong file component sẵn có vào Project Rose 2000:

- a. Ta dùng WE duyệt file component rồi drag nó vào cửa sổ miêu tả cây Project
- b. Ta dùng WE duyệt file component rồi drag nó vào cửa sổ working
- c. Cả 2 câu trên đều đúng
- d.

**Câu 36:**

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

**Câu 37:**

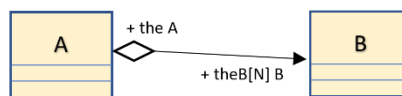
- a.
- b.
- c.
- d.

**Câu 38:**

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

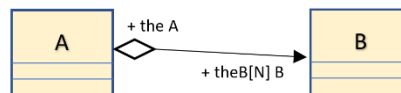
**Câu 39:** Xem hình bên, khi sinh mã VC++ tự động bởi Rose 200x thì trong class A có phát biểu nào sau đây:

- a. `public: B theB;`
- b. `public: B* theB;`
- ☒ c. `public: B* theB[N];`
- d. `public: B theB[N];`



**Câu 40:** Xem hình trên, khi sinh mã VB tự động bởi Rose 200x thì trong class A có phát biểu nào sau đây:

- a. `public Dim theB As B`
- b. `public Dim theB As Collection`
- ☒ c. `public Dim theB[N]_B_ As B`
- d. `public Dim theB[N]_B_ As Collection`



**Câu 41:** Ngôn ngữ C++:

- a. Không hỗ trợ kiểu trừu tượng của .....
- b.
- c.
- d.

**Câu 42:**

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

**Câu 43:**

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

**Câu 44:**

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

**Câu 45:** Trong qui trình phát triển phần mềm hợp nhất, các workflow:

- a. Phải được thực hiện tuần tự
- b. Phải được thực hiện tuần tự nhưng có thể lặp lại nhiều lần
- c. Có thể được thực hiện đồng thời
- ☒ d. Có thể được thực hiện đồng thời nhưng có thể lặp lại nhiều lần