**Ôn thi cuối kỳ Công nghệ phần mềm**

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

1. Tuần tự
2. Song song
3. Lập theo pipe-line
4. Tùy theo sở thích

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

1. Từng workflow
2. Từng bước nhỏ trong workflow
3. Từng chu kỳ phát triển phần mềm
4. Cả 3 câu trên đều đúng

**Câu 4:** UML cho phép:

1. Đặc tả và xây dựng các artifacts cho 1 hệ thống cần tin học hóa
2. Lập tài liệu và hiển thị trực quan các artifacts cho 1 hệ thống cần tin học hóa
3. Cả 2 câu trên đều đúng
4. Cả 2 câu trên đều sai

**Câu 5:** UML cung cấp:

1. 2 ký hiệu mô tả thông tin cơ bản
2. 3 ký hiệu mô tả thông tin cơ bản
3. Nhiều ký hiệu mô tả thông tin cơ bản
4. Cả 3 câu trên đều sai

Câu 6: Lược đồ use-case có thể chứa:

1. Use-case, actor, mối quan hệ giữa các thành phần
2. Use-case, actor, note
3. Use-case, actor, note, mối quan hệ giữa các thành phần
4. Cả 3 câu trên đều sai

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

1. Generalization
2. Include, extend
3. Cả 2 câu a và b đều đúng
4. Cả 2 câu a và b đều sai

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

1. Generalization
2. Include, extend
3. Cả 2 câu a và b đều đúng
4. Cả 2 câu a và b đều sai

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

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

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

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

Câu 11: Lược đồ use-case cho thấy:

1. Toàn bộ chức năng của hệ thống phần mềm
2. Toàn bộ chức năng của hệ thống phần mềm và actor
3. Một số chức năng của hệ thống phần mềm
4. 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:

1. Toàn bộ các chức năng của hệ thống phần mềm
2. Toàn bộ chức năng của hệ thống phần mềm và actor
3. Một số chức năng của hệ thống phần mềm
4. 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:

1. Giải thuật thực hiện chức năng của use-case tương ứng
2. 1 kịch bản thực hiện chức năng của use-case tương ứng
3. Sự tương tác giữa các đối tượng
4. 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:

1. Giải thuật thực hiện chức năng của use-case tương ứng
2. 1 kịch bản thực hiện chức năng của use-case tương ứng
3. Sự tương tác giữa các đối tượng
4. 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:

1. Giải thuật thực hiện chức năng của use-case tương ứng
2. 1 kịch bản thực hiện chức năng của use-case tương ứng
3. Sự tương tác giữa các đối tượng
4. 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:

1. Nắm bắt yêu cầu
2. Phân tích
3. Thiết kế
4. Hiện thực

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

1. Use-case, actor, note, mối quan hệ giữa các thành phần
2. Actor, class, interface và mối quan hệ giữa chúng
3. Actor, class, đối tượng và mối quan hệ giữa chúng
4. Cả 3 câu trên đều sai

Câu 18: Ở cấp phân tích:



Câu 19: Mỗi class phân tích có thể:

1. Tham gia thực hiện 1 chức năng nào đó của phần mềm
2. Thực hiện 1 yêu cầu không chức năng nào đó của phần mềm
3. Cả 2 câu trên đều đúng
4. 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:

1. Lược đồ cộng tác
2. Lược đồ trình tự
3. Lược đồ hoạt động
4. Cả 3 câu trên đều đúng

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

1. Vi mô và vĩ mô
2. Tổng quát hóa
3. Hiện thực
4. Cả 3 câu trên đều đúng

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

1. Ta nên tìm actor trước rồi tìm use-case dựa trên actor tìm được
2. Ta nên tìm actor và use-case đồng thời bởi 2 nhóm khác nhau để hiệu quả
3. Ta nên tìm use-case trước rồi tìm actor dựa trên use-case tìm được
4. 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:

1. Các module chứa chức năng cấu thành phần mềm và mối quan hệ giữa chúng
2. Các góc nhìn (view) của các mô hình
3. Các mô hình
4. Cả 3 câu trên đều đúng

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

1. Kiến trúc sư
2. Người phân tích hệ thống phần mềm
3. Người đặc tả use-case
4. Cả 3 worker trên phải hợp tác với nhau

Câu 25: Việc phân tích 1 class là của:

1. Người phân tích hệ thống phần mềm
2. Kỹ sư thành phần (component engineer)
3. Kỹ sư use-case (use-case engineer)
4. Cả 3 worker trên phải hợp tác với nhau

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

1. Hoạt động nắm bắt yêu cầu của phần mềm
2. Hoạt động phân tích yêu cầu phần mềm
3. Hoạt động thiết kế
4. Hoạt động viết code

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

1. Chuyên gia lĩnh vực
2. Phân tích viên
3. Kiến trúc sư phần mềm
4. Kỹ sư use-case

Câu 28: Các package phân tích cần có:

1. Tính kết dính thấp (cohesion) và tính nối ghép thấp (coupling)
2. Tính kết dính thấp (cohesion) và tính nối ghép cao (coupling)
3. Tính kết dính cao (cohesion) và tính nối ghép thấp (coupling)
4. 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:

1. Mô hình use-case
2. Mô hình phân tích
3. Mô hình thiết kế
4. 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:

1. Trở thành tên thuộc tính trong class B
2. Trở thành tên thuộc tính trong class A
3. 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
4. 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