*** 주요 키워드 ***

(1) 소프트웨어 재사용
(2) 소프트웨어 재공학 (Re-engineering)
(3) 소프트웨어 역공학
(4) 클라이언트/서버 모델
(5) CASE
(6) 2008년 기출문제(중복제거)
(7) 2009년 기출문제(중복제거)

(1) 소프트웨어 재사용

****

[기-09년3월][기-08년3월][기-99년4월]
1. 다음 중 소프트웨어를 재사용함으로써 얻는 이점이 아닌 것은?
가. 개발시간과 비용을 단축시킨다.
나. 소프트웨어 개발의 생산성을 높인다.
다. 프로젝트 실패의 위험을 줄여 준다.
라. 새로운 개발 방법론의 도입이 쉽다.

[기-07년5월]
2. 소프트웨어 재사용의 이점에 속하지 않는 것은?
가. 소프트웨어의 품질 향상
나. 소프트웨어의 개발시간과 비용감소
다. 소프트웨어의 생산성 증가
라. 소프트웨어 프로그래밍 언어의 종속

[기-07년3월]
3. 재사용 라이브러리가 가져야 할 속성이 아닌 것은?
가. 확장성
나. 비표준화된 요소 표현 형식
다. 재사용 요소들의 생성, 편집 등을 허용하는 연산
라. 편리한 접근, 탐색, 버전관리, 제어 변경

[기-07년3월]
4. 소프트웨어 부품에 적용되는 품질로서, 과학 계산용 라이브러리와 같이 이미 만들어진 프로그램을 사용하는 것을 의미하는 것은?
가. 신뢰성 나. 재사용성 다. 확장성 라. 유지보수성

[기-09년8월][기-07년3월][기-03년8월]
5. 소프트웨어 재사용의 이점으로 볼 수 없는 것은?
가. 개발 비용을 감소시킨다.
나. 프로그램 언어가 종속적이다.
다. 소프트웨어 품질을 향상시킨다.
라. 프로그램 생성 지식을 공유하게 된다.

[기-06년9월][기-05년3월][기-02년9월][기-99년10월]
6. 소프트웨어의 재사용(reusability)에 대한 효과와 거리가 먼 것은?
가. 사용자의 책임과 권한부여
나. 소프트웨어 품질 향상
다. 생산성 향상
라. 구축 방법에 대한 지식의 공유

[기-06년5월][기-04년3월]
7. 소프트웨어의 재사용으로 얻어지는 이익이 아닌 것은?
가. 표준화의 원칙을 무시할 수 있다.
나. 프로젝트의 개발 위험을 줄여줄 수 있다.
다. 프로젝트의 개발기간과 비용을 줄일 수 있다.
라. 개발자의 생산성을 향상시킬 수 있다.

[기-05년5월]
8. 소프트웨어 재사용에 가장 많이 이용되는 것은?
가. Data 나. Test Case
다. Source Code 라. Project Plan

[기-05년3월]
9. 소프트웨어 재사용을 통한 장점이라고 볼 수 없는 것은?
가. 개발 시간과 비용을 감소시킨다.
나. 소프트웨어 품질을 향상시킨다.
다. 생산성을 증가시킨다.
라. 고급 프로그래머 배출이 용이하다.

[기-04년5월]
10. 소프트웨어의 재사용과 관련된 내용 중 가장 적절한 설명은?
가. 시스템 명세, 설계, 코드 그리고 다른 팀에 의해 작성된 문서를 공유함으로 소프트웨어 개발을 복잡하게 만든다.
나. 소프트웨어를 재사용함으로써 유지 보수 비용이 높아진다.
다. 모든 소프트웨어를 개발할 때는 반드시 소프트웨어를 재사용하여야만 한다.
라. 소프트웨어의 개발 생산성과 품질을 높이려는 주요 방법이다.

[기-08년5월][기-03년5월]
11. 소프트웨어 컴포넌트(Component) 재사용의 이점이라고 볼 수 없는 항목은?
가. 소프트웨어의 품질 향상
나. 개발 담당자의 생산성 향상
다. 개발 비용의 절감
라. 응용소프트웨어의 보안 유지

[기-03년3월][기-07년9월]
12. 소프트웨어 재사용에 대한 설명으로 옳지 않은 것은?
가. 개발 시간과 비용을 감소시킨다.
나. 프로젝트 실패의 위험을 줄여 준다.
다. 재사용 부품의 크기가 작을수록 재사용 확률이 낮다.
라. 소프트웨어 개발자의 생산성을 증가시킨다.

[기-02년5월]
13. 소프트웨어 재사용에 대한 설명으로 옳은 것은?
가. 기계중심언어는 주기억장치를 매우 효과적으로 사용 할 수 있고 실행속도를 최적화 시킬 수 있다는 점에서 재 사용성이 가장 뛰어난 프로그래밍 언어이다.
나. 1990년대의 클래스, 객체 등의 소프트웨어 요소는 소프트웨어 재사용성을 크게 향상시켰다.
다. 노후된 시스템에 대한 재분석, 문서화 작업을 통해 공학적으로 우수한 시스템을 만드는 것을 의미한다.
라. 소프트웨어 재사용은 경제성을 고려하여 사용자의 요구가 있을 때만 적용하는 것이 바람직하다.

[기-01년6월]

14. 소프트웨어 재사용에 관한 설명으로 거리가 먼 것은?
가. 소프트웨어의 개발 생산성과 품질을 높이려는 방법이다.
나. 소프트웨어 재사용의 방법에는 합성중심(composition- based)과 생성 중심(generation-based) 방법으로 나눌 수 있다.
다. 재사용 부품의 크기는 클수록 재사용율이 높다.
라. 소프트웨어의 재사용은 프로젝트의 실패 위험을 줄일 수 있다.

[기-01년3월]
15. 소프트웨어 재사용으로 인한 영향이 아닌 것은?
가. 품질 향상 나. 생산성 인하
다. 생산성 향상 라. 비용 절감

[기-00년3월]
16. 재사용 컴포넌트(Component)들이 많이 있어도 그들을 찾아내는 것이 어려울 경우, 가장 주된 원인은 무엇인가?
가. 프로그램 언어의 차이
나. 분류(classification)의 문제
다. 객체지향방법과 전통적인 방법의 상충
라. 통합성의 문제

(2) 소프트웨어 재공학 (Re-engineering)

****
[기-07년5월][기-03년5월]
17. 다음 설명에 해당하는 것은?

- 기존 시스템을 이용하여 보다 나은 시스템을 구축하고 새로운 기능을 추가하여 소프트웨어 성능을 향상시킴 - 주요 활동으로 분석, 개조, 역공학, 이식 등이 있음

가. 소프트웨어 재공학(Software Re-engineering)
나. 소프트웨어 분석(Software analysis)
다. 소프트웨어 프로그래밍(Software Programming)
라. 소프트웨어 개발(Software Development)

[기-07년3월]
18. 기존에 있던 소프트웨어를 파기하지 않고 변경된 사용자의 요구사항이나 수정된 환경으로 기존 소프트웨어를 수정 보완하여 재구축하자는 개념은?
가. 소프트웨어 재공학(reenineering)
나. 소프트웨어 재판매(resale)
다. 소프트웨어 재정의(redefine)
라. 소프트웨어 재조정(readjust)

[기-05년9월]
19. 소프트웨어 재공학의 개념으로 옳지 않은 것은?
가. 재공학은 유지보수에 대한 장기적인 전략적 고려와 많은 비용, 시간, 자원을 요구한다.
나. 재공학은 유지보수성, 생산성, 품질의 향상을 목적으로 한다.
다. 재공학은 형식의 변경과 재 설계 과정을 포함한다.
라. 재공학은 자사 소프트웨어를 대상으로 소스코드 이상의 추상화 수준으로 명세화하는 과정이다.

[기-01년9월][기-05년5월][기-03년8월]
20. 소프트웨어의 위기를 해결하기 위해 개발의 생산성이 아닌 유지보수의 생산성으로 해결하려는 방법을 의미하는 것은?
가. 소프트웨어 재사용
나. 소프트웨어 재공학
다. 클라이언트/서버 소프트웨어 공학
라. 전통적 소프트웨어공학

[기-03년3월][기-00년10월]
21. 소프트웨어 리엔지니어링(reengineering)의 목표 중 거리가 먼 것은?
가. 복잡한 시스템을 다루는 방법 구현
나. 다른 뷰의 생성
다. 기존 시스템의 해킹
라. 잃어버린 정보의 복구 및 제거

[기-01년3월]
22. 소프트웨어 재공학에 관한 설명으로 옳지 않은 것은?
가. 소프트웨어 재공학은 CASE의 정보저장소와는 무관하다.
나. 재공학 활동은 분석, 재구성, 역공학, 이식 활동 등으로 구분할 수 있다.
다. 소프트웨어 재공학도 자동화된 도구를 사용하여 소프트웨어를 분석하고 수정하는 과정을 포함한다.
라. 소프트웨어 재공학의 일반적인 개념은 데이터와 기능들의 개조 및 개선을 가해 유지보수 용이성을 향상시키자는 것이다.

[기-99년8월]
23. 소프트웨어 리엔지니어링(re-engineering)의 목표 중 거리가 먼 것은 어느 것인가?
가. 복잡한 시스템을 다루는 방법
나. 다른 뷰의 생성
다. 기존 시스템의 해킹
라. 잃어버린 정보의 복구 및 제거

[기-09년8월][기-08년3월][기-99년4월]
24. 소프트웨어 재공학의 필요성이 대두된 주된 이유는?
가. 요구사항 분석의 문제 나. 설계의 문제
다. 구현의 문제 라. 유지보수의 문제

(3) 소프트웨어 역공학

****
[기-08년3월][기-06년3월][기-02년3월][기-00년10월]
25. 현재 프로그램으로부터 데이터, 아키텍처, 그리고 절차에 관한 분석 및 설계 정보를 추출하는 과정은?
가. 재공학(re-engineering)
나. 역공학(reverse engineering)
다. 순공학(forward engineering)
라. 재사용(reuse)

[기-09년5월][기-08년5월][기-02년9월]
26. 소프트웨어 분석하여 소프트웨어 개발과정과 데이터 처리 과정을 설명하는 분석 및 설계 정보를 재발견하거나 다시 만들어 내는 작업을 무엇이라 하는가?
가. 순공학 나. 역공학 다. 재구축 라. 전공학

(4) 클라이언트/서버 모델

****

[기-03년3월][기-00년3월]

27. 클라이언트/서버(Client/Server) 모델에서의 소프트웨어 개발에 대한 설명으로 옳지 않은 것은?

가. 사용자의 요구사항은 서버의 데이터베이스 시스템에 영향을 미친다.
나. 병목현상을 없애기 위해 비즈니스 로직을 분리하여 관리할 수 있다.
다. 미들웨어의 사용은 서버와 클라이언트의 작업량을 증가시켰다.
라. 대부분 네트워크로 연결되어 있고 인증 작업을 필요로 한다.

[기-99년4월]

28. 다음 중 클라이언트/서버 시스템을 위한 소프트웨어 요소에 해당되지 않는 것은?

가. 객체지향 요소 나. 어플리케이션 요소
다. 데이터베이스 요소 라. 프리젠테이션/상호작용 요소

(5) CASE

****

[기-05년3월][기-01년6월]

29. CASE에 대한 설명으로 옳지 않은 것은?

가. 소프트웨어의 개발과정을 자동화함으로써 생산성을 증대시키고자 하는 목적으로 개발되었다.
나. CASE는 소프트웨어 개발의 모든 단계에 걸쳐 일관된 방법론을 지원한다.
다. CASE를 사용함으로 개발의 표준화를 지향하고, 자동화의 이점을 얻을 수 있다.
라. CASE는 시스템의 개발 속도를 빠르게 하지만 재사용성은 떨어진다.

[기-05년3월]

30. 다음의 자동화 예측 도구들 중 Rayleigh-Norden 곡선과 Putnam 의 예측모델에 기반을 둔 것은?

가. SLIM 나. ESTIMACS
다. SPQR/20 라. WICOMO

[기-08년3월][기-03년8월][기-06년5월][기-00년3월]

31. CASE(Computer-Aided Software Engineering)에 대한 설명으로 옳지 않은 것은?

가. 소프트웨어 개발의 작업들을 자동화하는 것이다.
나. 소프트웨어 도구와 방법론의 결합이다.
다. 소프트웨어의 생산성 문제를 해결할 수 있다.
라. 개발과정이 빠른 대신 재사용성이 떨어진다.

[기-07년5월][기-06년3월][기-05년3월][기-02년5월]

32. 소프트웨어 생명 주기의 전체 단계를 연결시켜 주고 자동화 시켜 주는 통합된 도구를 제공해 주는 것은?

가. UIMS 나. CASE 다. OOD 라. SADT

[기-07년3월]

33. CASE(Computer Aided Software Engineering)에 대한 설명으로 옳지 않은 것은?

가. 프로그램의 구현과 유지보수 작업만을 중심으로 소프트웨어 생산성 문제를 해결한다.
나. 소프트웨어 생명 주기의 전체 단계를 연결해 주고 자동화 해주는 통합된 도구를 제공한다.
다. 개발 과정의 속도를 향상 시킨다.
라. 소프트웨어 부품의 재사용을 가능하게 한다.

[기-06년9월]

34. CASE 사용에 관한 설명으로 옳지 않은 것은?

가. 소프트웨어 부품의 재사용성이 낮아진다.
나. 소프트웨어 개발을 신속하게 할 수 있다.
다. 유지보수를 간단하고 간편하게 수행 할 수 있다.
라. 자동화 기법을 통하여 소프트웨어 품질이 향상된다.

[기-05년9월][기-99년4월]

35. CASE(Computer Aided Software Engineering)에 대한 설명 중 틀린 것은?

가. CASE는 상위(upper) CASE, 중위(midium) CASE, 하위(Lower) CASE, 통합(integrated) CASE의 4가지 형태로 나눌 수 있다.
나. 통합 CASE는 소프트웨어 개발 주기 전체과정을 지원한다.
다. 상위 CASE는 요구분석과 설계단계를 지원한다.
라. 하위 CASE는 코드를 작성하고 테스트하며 문서화하는 과정을 지원한다.

[기-05년5월]

36. CASE(Computer Aided Software Engineering)에 관한 설명으로 거리가 먼 것은?

가. 소프트웨어 공학의 여러 작업들을 자동화하는 것이다.
나. 소프트웨어 수명주기의 어느 부분을 지원하느냐에 따라 organic case, semi-detached case, embedded case 로 분류할 수 있다.
다. 소프트웨어 시스템의 문서화 및 명세화를 위한 그래픽 기능을 제공한다.
라. 자료흐름도 등의 다이어그램을 쉽게 작성하게 해주는 소프트웨어도 CASE 도구이다.

[기-05년3월]

37. 소프트웨어 자동화도구인 CASE 에 대한 설명으로 부적절한 것은?

가. 차세대 CASE 도구는 통합화, 지능화로 정의될 수 있다.
나. 설계지식이 없을 때 CASE 를 사용하면 효과적이다.
다. CASE 정보저장소에는 데이터, 프로세스, 다이아그램, 규칙 등에 관한 정보가 저장된다.
라. CASE 시스템은 다이아그램도구, 설계분석기, 코드생성기, 정보저장소, 프로젝트관리도구, 재공학도구, 프로토타이핑도구 등으로 구성된다.

[기-04년9월]

38. CASE가 제공하는 기능으로 거리가 먼 것은?

가. 개발기간의 단축
나. 개발 방법론의 생성
다. 소프트웨어 품질향상
라. 소프트웨어 개발 단계의 표준화

[기-04년5월][기-02년3월][기-00년7월]

39. 소프트웨어 개발 과정에서 사용되는 요구 분석, 설계, 구현, 검사 및 디버깅 과정을 컴퓨터와 전용의 소프트웨어 도구를 사용하여 자동화하는 것을 무엇이라고 하는가?

가. CAT(Computer Aided Testing)

나. CAD/CAM(Computer Aided Design and Manufacturing)
다. CASE(Computer Aided Software Engineering)
라. CAI(Computer Aided Instruction)

[기-03년5월]
40. CASE 에 관한 설명으로 옳지 않은 것은?
가. 소프트웨어 개발비용을 절약할 수 있다.
나. 자동화된 검사를 통해 소프트웨어 품질을 향상시킨다.
다. 모듈의 수가 증가하므로 개발 기간이 늘어난다.
라. 프로그램의 유지 보수를 간편하게 한다.

[기-02년9월]
41. CASE(computer Aided Software Engineering) 툴(tool)의 장점으로 거리가 먼 것은?
가. 자동화 기법을 통해 소프트웨어 품질이 향상된다.
나. 개발 기간이 짧다.
다. CASE 표준이 있어서 대부분의 CASE 툴 들은 호환된다.
라. 개발 비용이 절감된다.

[기-08년9월][기-02년5월]
42. CASE 도구의 정보저장소(Repository)에 대한 설명으로 거리가 먼 것은?
가. 일반적으로 정보저장소는 도구들과 생명 주기 활동, 사용자들, 응용 소프트웨어들 사이의 통신과 소프트웨어 시스템 정보의 공유를 향상시킨다.
나. 초기의 소프트웨어 개발 환경에서는 사람이 정보 저장소 역할을 했지만 오늘날에는 응용프로그램이 정보 저장소 역할을 담당한다.
다. 정보 저장소는 도구들의 통합, 소프트웨어 시스템의 표준화, 소프트웨어 시스템 정보의 공유, 시스템웨어 재사용성의 기본이 된다.
라. 소프트웨어 시스템 구성 요소들과 시스템 정보가 정보 저장소에 의해 관리되므로 소프트웨어 시스템의 유지보수가 용이해진다.

[기-01년3월]
43. CASE(Computer Aided Software Engineering)에 대한 설명으로 거리가 먼 것은?
가. 개발도구와 개발 방법론이 결합된 것이다.
나. 시스템 개발과정의 일부 또는 전체를 자동화하는 것이다.
다. 유지보수성을 높이기 위해 기존 소프트웨어를 재구성하고 새로운 기술을 적응시키는 것이다.
라. 정형화된 구조 및 메커니즘을 소프트웨어 개발에 적용하여 소프트웨어 생산성 향상을 구현하는 공학 기법이다.

[기-06년5월]
44. 다음은 무엇에 대한 설명인가?

"획기적인 결과를 성취하기 위한 비즈니스 프로세스에 대한 연구, 비즈니스 프로세스 구현, 비즈니스 프로세스의 근본적인 변경"

가. BPR 나. CRC 다. EBP 라. DFD

[기-07년9월]
45. CASE에 대한 설명으로 거리가 먼 것은?
가. 자동화된 기법을 통해 소프트웨어 품질이 향상된다.
나. 소프트웨어 부품의 재사용성이 향상된다.
다. 프로토타입 모델에 위험 분석 기능을 추가한 생명주기 모형이다.
라. 소프트웨어 도구와 방법론의 결합이다.

(6) 2008년 기출문제(중복제거)

****
[기-08년5월]
46. 소프트웨어 재공학은 어떤 유지보수 측면에서 소프트웨어 위기를 해결하려고 하는 방법인가?
가. 수정(Corrective) 유지 보수
나. 적응(Pdaptive) 유지 보수
다. 완전화(Perfective) 유지 보수
라. 예방(Preventive) 유지 보수

[기-08년5월]
47. CASE에 대한 설명으로 거리가 먼 것은?
가. 소프트웨어 모듈의 재사용성이 향상된다.
나. 자동화된 기법을 통해 소프트웨어 품질이 향상된다.
다. 소프트웨어 사용자들이 소프트웨어 사용 방법을 신속히 숙지할 수 있도록 개발된 자동화 패키지이다.
라. 소프트웨어 유지보수를 간편하게 수행할 수 있다.

[기-08년9월]
48. 소프트웨어 재공학이 재개발에 비해 갖는 주요한 장점으로 거리가 먼 것은?
가. 위험부담 감소
나. 비용 절감
다. 시스템 명세의 오류억제
라. 최신의 소프트웨어 공학기법 적용

(7) 2009년 기출문제(중복제거)

****
[기-09년3월]
49. 소프트웨어공학에서 CASE의 효과에 해당하지 않는 것은?
가. 소프트웨어 개발 주기의 표준안 확립
나. 소프트웨어 개발 기법의 실용화
다. 문서화의 용이성 제공
라. 시스템 수정 및 유지보수 확대

[기-09년3월]
50. 재공학의 목적으로 적합하지 않은 것은?
가. 소프트웨어의 수명을 연장시킨다.
나. 소프트웨어의 유지보수성을 향상시킨다.
다. 소프트웨어 개발 기간을 연장시켜 비용을 증가시킨다.
라. 소프트웨어에서 사용하고 있는 기술을 향상시킨다.

[기-09년5월]
51. 소프트웨어 재공학(Reengineering)에 관한 설명으로 거리가 먼 것은?

가. 현재의 시스템을 변경하거나 재구조화(Restructuring)하는 것이다.
나. 재구조화는 재공학의 한 유형으로 사용자의 요구사항이나 기술적 설계의 변경 없이 프로그램을 개선하는 것이다.
다. 재개발(Redevelopment)과 재공학은 동일한 의미이다.
라. 사용자의 요구사항을 변경시키지 않고, 기술적 설계를 변경하여 프로그램을 개선하는 것도 재공학이다.

[기-09년5월]
52. CASE(Computer-Aided Software Engineering)에 대한 설명으로 옳지 않은 것은?
가. 소프트웨어 부품의 재사용성을 향상시켜 준다.
나. Rayleigh-Norden 곡선의 노력 분포도를 기초로 한 생명 주기 예측 모형이다.
다. 소프트웨어 생명 주기의 모든 단계를 연결시켜 주고 자동화시켜 준다.
라. 소프트웨어의 유지보수를 용이하게 수행할 수 있도록 해 준다.

[기-09년8월]
53. 소프트웨어 역공학(Software reverse engineering)에 대한 설명으로 옳지 않은 것은?
가. 역공학의 가장 간단하고 오래된 형태는 재문서화라고 할 수 있다.
나. 기존 소프트웨어의 구성 요소와 그 관계를 파악하여 설계도를 추출한다.
다. 원시 코드를 분석하여 소프트웨어의 관계를 파악한다.
라. 대상 시스템이 없이 새로운 시스템으로 개선하는 변경 작업이다.

[SE05-발전적 주제]

1	2	3	4	5	6	7	8	9	10
라	라	나	나	나	가	가	다	라	라
11	12	13	14	15	16	17	18	19	20
라	다	나	다	나	나	가	가	라	나
21	22	23	24	25	26	27	28	29	30
다	가	다	라	나	나	다	가	라	가
31	32	33	34	35	36	37	38	39	40
라	나	가	가	가	나	나	나	다	다
41	42	43	44	45	46	47	48	49	50
다	나	다	가	다	라	다	라	라	다
51	52	53	54	55	56	57	58	59	60
다	나	라							