



저작권 안내

이 자료는 시나공 카페 회원을 대상으로 하는 자료로서 개인적인 용도로만 사용할 수 있습니다. 허락 없이 복제하거나 다른 매체에 옮겨 실을 수 없으며, 상업적 용도로 사용할 수 없습니다.

※ 다음 문제를 읽고 알맞은 것을 골라 답안카드의
답란(①, ②, ③, ④)에 표기하시오.

제1과목 : 소프트웨어 설계

- ※ 다음 문제를 읽고 알맞은 것을 골라 답안카드의 답란(①, ②, ③, ④)에 표기하시오.

제1과목 : 소프트웨어 설계

 - GoF(Gang of Four)의 디자인 패턴에서 행위 패턴에 속하는 것은?
 - Builder
 - Visitor
 - Prototype
 - Bridge
 - 객체지향 프로그램에서 데이터를 추상화하는 단위는?
 - 메소드
 - 클래스
 - 상속성
 - 메시지
 - 객체지향 기법에서 클래스들 사이의 '부분-전체(Part-Whole)' 관계 또는 '부분(is-a-part-of)'의 관계로 설명되는 연관성을 나타내는 용어는?
 - 일반화
 - 추상화
 - 캡슐화
 - 집단화
 - 객체지향 분석 방법론 중 E-R 다이어그램을 사용하여 객체의 행위를 모델링하며, 객체 식별, 구조식별, 주체 정의, 속성 및 관계 정의, 서비스 정의 등의 과정으로 구성되는 것은?
 - Coad와 Yourdon 방법
 - Booch 방법
 - Jacobson 방법
 - Wirfs-Brock's 방법
 - 코드 설계에서 일정한 일련번호를 부여하는 방식의 코드는?
 - 연상 코드
 - 블록 코드
 - 순차 코드
 - 표의 숫자 코드
 - 소프트웨어 설계 시 구축된 플랫폼의 성능특성 분석에 사용되는 측정 항목이 아닌 것은?
 - 응답시간(Response Time)
 - 가용성(Availability)
 - 사용률(Utilization)
 - 서버 튜닝(Server Tuning)
 - 자료 사전에서 자료의 생략을 의미하는 기호는?
 - { }
 - **
 - =
 - ()
 - 검토회의 전에 요구사항 명세서를 미리 배포하여 사전 검토한 후 짧은 검토 회의의를 통해 오류를 조기에 검출하는데 목적을 두는 요구사항 검토 방법은?
 - 빌드 검증
 - 동료 검토
 - 워크 스루
 - 개발자 검토
 - CASE가 갖고 있는 주요 기능이 아닌 것은?
 - 그래픽 지원
 - 소프트웨어 생명주기 전 단계의 연결
 - 언어 번역
 - 다양한 소프트웨어 개발 모형 지원
 - XP(eXtreme Programing)의 5가지 가치로 거리가 먼 것은?
 - 용기
 - 의사소통
 - 정형 분석
 - 피드백
 - DBMS 분석 시 고려사항으로 거리가 먼 것은?
 - 가용성
 - 성능
 - 네트워크 구성도
 - 상호 호환성
 - HIPO(Hierarchy Input Process Output)에 대한 설명으로 거리가 먼 것은?
 - 상향식 소프트웨어 개발을 위한 문서화 도구이다.
 - HIPO 차트 종류에는 가시적 도표, 총체적 도표, 세부적 도표가 있다.
 - 기능과 자료의 의존 관계를 동시에 표현할 수 있다.
 - 보기 쉽고 이해하기 쉽다.
 - UI 설계 원칙에서 누구나 쉽게 이해하고 사용할 수 있어야 한다는 것은?
 - 유효성
 - 직관성
 - 무결성
 - 유연성
 - 럼바우(Rumbaugh)의 객체지향 분석 절차를 가장 바르게 나열한 것은?
 - 객체 모형 → 동적 모형 → 기능 모형
 - 객체 모형 → 기능 모형 → 동적 모형
 - 기능 모형 → 동적 모형 → 객체 모형
 - 기능 모형 → 객체 모형 → 동적 모형
 - 데이터 흐름도(DFD)의 구성요소에 포함되지 않는 것은?
 - Process
 - Data Flow
 - Data Store
 - Data Dictionary
 - UML 확장 모델에서 스테레오 타입 객체를 표현할 때 사용하는 기호로 맞는 것은?
 - << >>
 - (())
 - { { }
 - [[]]
 - 트랜잭션이 올바르게 처리되고 있는지 데이터를 감시하고 제어하는 미들웨어는?
 - RPC
 - ORB
 - TP monitor
 - HUB
 - 소프트웨어 개발 방법 중 요구사항 분석(Requirements Analysis)과 거리가 먼 것은?
 - 비용과 일정에 대한 제약설정
 - 타당성 조사
 - 요구사항 정의 문서화
 - 설계 명세서 작성

19. 공통 모듈에 대한 명세 기법 중 해당 기능에 대해 일관되게 이해되고 한 가지로 해석될 수 있도록 작성하는 원칙은?

- ① 상호작용성 ② 명확성
③ 독립성 ④ 내용성

20. UML 모델에서 사용하는 Structural Diagram에 속하지 않은 것은?

- ① Class Diagram ② Object Diagram
③ Component Diagram ④ Activity Diagram

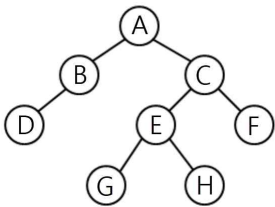
제2과목 : 소프트웨어 개발

21. 평가 점수에 따른 성적부여는 다음 표와 같다. 이를 구현한 소프트웨어를 경계 값 분석 기법으로 테스트 하고자 할 때 다음 중 테스트 케이스의 입력 값으로 옳지 않은 것은?

평가점수	성적
80~100	A
60~79	B
0~59	C

- ① 59 ② 80
③ 90 ④ 101

22. 다음 트리의 차수(degree)와 단말 노드(terminal node)의 수는?



- ① 차수 : 4, 단말 노드 : 4
② 차수 : 2, 단말 노드 : 4
③ 차수 : 4, 단말 노드 : 8
④ 차수 : 2, 단말 노드 : 8

23. 검증 검사 기법 중 개발자의 장소에서 사용자가 개발자가 앞에서 행하는 기법이며, 일반적으로 통제된 환경에서 사용자와 개발자가 함께 확인하면서 수행되는 검사는?

- ① 동치 분할 검사 ② 형상 검사
③ 알파 검사 ④ 베타 검사

24. 하향식 통합에 있어서 모듈 간의 통합 시험을 위해 일시적으로 필요한 조건만을 가지고 임시로 제공되는 시험용 모듈을 무엇이라고 하는가?

- ① Stub ② Driver
③ Procedure ④ Function

25. 소프트웨어 품질 측정을 위해 개발자 관점에서 고려해야 할 항목으로 거리가 먼 것은?

- ① 정확성 ② 무결성
③ 사용성 ④ 간결성

26. 소프트웨어 테스트에서 오류의 80%는 전체 모듈의 20% 내에서 발견된다는 법칙은?

- ① Brooks의 법칙 ② Boehm의 법칙
③ Pareto의 법칙 ④ Jackson의 법칙

27. 디지털 저작권 관리(DRM)의 기술 요소가 아닌 것은?

- ① 크랙 방지 기술 ② 정책 관리 기술
③ 암호화 기술 ④ 방화벽 기술

28. 인터페이스 보안을 위해 네트워크 영역에 적용될 수 있는 솔루션과 거리가 먼 것은?

- ① IPSec ② SMTP
③ SSL ④ S-HTTPS

29. 인터페이스 구현 검증도구 중 아래에서 설명하는 것은?

- 서비스 호출, 컴포넌트 재사용 등 다양한 환경을 지원 하는 테스트 프레임워크
- 각 테스트 대상 분산 환경에 데몬을 사용하여 테스트 대상 프로그램을 통해 테스트를 수행하고, 통합하여 자동화하는 검증 도구

- ① xUnit ② STAF
③ FitNesse ④ RubyNode

30. SW 패키징 도구 활용 시 고려 사항과 거리가 먼 것은?

- ① 패키징 시 사용자에게 배포되는 SW이므로 보안을 고려한다.
② 사용자 편의성을 위한 복잡성 및 비효율성 문제를 고려한다.
③ 보안상 단일 기종에서만 사용할 수 있도록 해야 한다.
④ 제품 SW종류에 적합한 암호화 알고리즘을 적용한다.

31. 소프트웨어 형상 관리의 의미로 적절한 것은?

- ① 비용에 관한 사항을 효율적으로 관리하는 것
② 개발 과정의 변경 사항을 관리하는 것
③ 테스트 과정에서 소프트웨어를 통합하는 것
④ 개발 인력을 관리하는 것

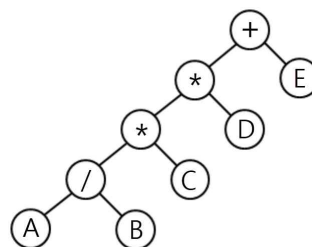
32. White Box Testing에 대한 설명으로 옳지 않은 것은?

- ① Base Path Testing, Boundary Value Analysis가 대표적인 기법이다.
② Source Code의 모든 문장을 한 번 이상 수행함으로써 진행된다.
③ 모듈 안의 작동을 직접 관찰할 수 있다.
④ 산출물의 각 기능별로 적절한 프로그램의 제어구조에 따라 선택, 반복 등의 부분들을 수행함으로써 논리적 경로를 점검한다.

33. 외계인 코드(Alien Code)에 대한 설명으로 옳은 것은?

- ① 프로그램의 로직이 복잡하여 이해하기 어려운 프로그램을 의미한다.
② 아주 오래되거나 참고문서 또는 개발자가 없어 유지보수 작업이 어려운 프로그램을 의미한다.
③ 오류가 없어 디버깅 과정이 필요 없는 프로그램을 의미한다.
④ 사용자가 직접 작성한 프로그램을 의미한다.

34. 다음 트리를 전위 순회(Preorder Traversal)한 결과는?



- ① + * A B / * C D E

- ② $A \cdot B / C \cdot D \cdot E +$
 ③ $A / B \cdot C \cdot D + E$
 ④ $+ \cdot \cdot / A \cdot B \cdot C \cdot D \cdot E$

35. 알고리즘 시간 복잡도 $O(1)$ 이 의미하는 것은?

- ① 컴퓨터 처리가 불가
 ② 알고리즘 입력 데이터 수가 한 개
 ③ 알고리즘 수행시간이 입력 데이터 수와 관계없이 일정
 ④ 알고리즘 길이가 입력 데이터보다 작음

36. 정렬된 N개의 데이터를 처리하는데 $O(\log_2 N)$ 의 시간이 소요되는 정렬 알고리즘은?

- ① 선택 정렬 ② 삽입 정렬
 ③ 버블 정렬 ④ 합병 정렬

37. ISO/IEC 9126의 소프트웨어 품질 특성 중 기능성 (Functionality)의 하위 특성으로 옳지 않은 것은?

- ① 학습성 ② 적합성
 ③ 정확성 ④ 보안성

38. EAI(Enterprise Application Integration)의 구축 유형으로 옳지 않은 것은?

- ① Point-to-Point ② Hub & Spoke
 ③ Message Bus ④ Tree

39. 소스코드 품질 분석 도구 중 정적 분석 도구가 아닌 것은?

- ① pmd ② cppcheck
 ③ valMeter ④ checkstyle

40. 반정규화(Denormalization) 유형 중 중복 테이블을 추가하는 방법에 해당하지 않는 것은?

- ① 빌드 테이블의 추가
 ② 집계 테이블의 추가
 ③ 진행 테이블의 추가
 ④ 특정 부분만을 포함하는 테이블 추가

제3과목 : 데이터베이스 구축

41. SQL의 분류 중 DDL에 해당하지 않는 것은?

- ① UPDATE ② ALTER
 ③ DROP ④ CREATE

42. STUDENT 테이블에 독일어과 학생 50명, 중국어과 학생 30명, 영어영문학과 학생 50명의 정보가 저장되어 있을 때, 다음 두 SQL문의 실행 결과 튜플 수는? (단, DEPT 컬럼은 학과명)

- ① SELECT DEPT FROM STUDENT;
 ② SELECT DISTINCT DEPT FROM STUDENT;

- ① ① a 3, ② b 3 ③ ① a 50, ② b 3
 ③ ① a 130, ② b 3 ④ ① a 130, ② b 130

43. 다음 두 릴레이션에서 외래키로 사용된 것은? (단, 밑줄 친 속성은 기본키이다.)

- 과목(과목번호, 과목명)
 수강(수강번호, 학번, 과목번호, 학기)

- ① 수강번호 ② 과목번호
 ③ 학번 ④ 과목명

44. 정규화 과정 중 1NF에서 2NF가 되기 위한 조건은?

- ① 1NF를 만족하고 모든 도메인이 원자값이어야 한다.
 ② 1NF를 만족하고, 키가 아닌 모든 애틀리뷰트들이 기본키에 이행적으로 함수 종속되지 않아야 한다.
 ③ 1NF를 만족하고 다치 종속이 제거되어야 한다.
 ④ 1NF를 만족하고 키가 아닌 모든 속성이 기본키에 대하여 완전 함수적 종속 관계를 만족해야 한다.

45. 데이터 무결성 제약조건 중 “개체 무결성 제약” 조건에 대한 설명으로 맞는 것은?

- ① 릴레이션 내의 튜플들이 각 속성의 도메인에 지정된 값만을 가져야 한다.
 ② 기본키에 속해 있는 애틀리뷰트는 널 값이나 중복 값을 가질 수 없다.
 ③ 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다.
 ④ 외래키 값은 참조 릴레이션의 기본키 값과 동일해야 한다.

46. 이행적 함수 종속 관계를 의미하는 것은?

- ① $A \rightarrow B$ 이고 $B \rightarrow C$ 일 때, $A \rightarrow C$ 를 만족하는 관계
 ② $A \rightarrow B$ 이고 $B \rightarrow C$ 일 때, $C \rightarrow A$ 를 만족하는 관계
 ③ $A \rightarrow B$ 이고 $B \rightarrow C$ 일 때, $B \rightarrow A$ 를 만족하는 관계
 ④ $A \rightarrow B$ 이고 $B \rightarrow C$ 일 때, $C \rightarrow B$ 를 만족하는 관계

47. DML에 해당하는 SQL 명령으로만 나열된 것은?

- ① DELETE, UPDATE, CREATE, ALTER
 ② INSERT, DELETE, UPDATE, DROP
 ③ SELECT, INSERT, DELETE, UPDATE
 ④ SELECT, INSERT, DELETE, ALTER

48. 데이터베이스 시스템에서 삽입, 갱신, 삭제 등의 이벤트가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL은?

- ① 트리거(Trigger) ② 무결성(Integrity)
 ③ 잠금(Lock) ④ 복귀(Rollback)

49. 데이터베이스의 논리적 설계(Logical Design) 단계에서 수행 하는 작업이 아닌 것은?

- ① 레코드 집합의 분석 및 설계
 ② 논리적 데이터베이스 구조로 매핑(mapping)
 ③ 트랜잭션 인터페이스 설계
 ④ 스키마의 평가 및 정제

50. E-R 모델의 표현 방법으로 옳지 않은 것은?

- ① 개체타입 : 사각형 ② 관계타입 : 마름모
 ③ 속성 : 오각형 ④ 연결 : 선

51. 병행제어의 로킹(Locking) 단위에 대한 설명으로 옳지 않은 것은?

- ① 데이터베이스, 파일, 레코드 등은 로킹 단위가 될 수 있다.
 ② 로킹 단위가 작아지면 로킹 오버헤드가 감소한다.
 ③ 로킹 단위가 작아지면 데이터베이스 공유도가 증가한다.
 ④ 한꺼번에 로킹 할 수 있는 객체의 크기를 로킹 단위라고 한다.

52. 뷰(View)에 대한 설명으로 옳지 않은 것은?

- ① 뷰는 CREATE 문을 사용하여 정의한다.
 ② 뷰는 데이터의 논리적 독립성을 제공한다.
 ③ 뷰를 제거할 때에는 DROP 문을 사용한다.
 ④ 뷰는 저장장치 내에 물리적으로 존재한다.

53. 하나의 애틀리뷰트가 가질 수 있는 원자값들의 집합을 의미 하는 것은?

- ① 도메인 ② 튜플
 ③ 엔티티 ④ 다형성

54. 관계대수 연산에서 두 릴레이션이 공통으로 가지고 있는 속성을 이용하여 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 연산은?

- ① \bowtie
② \supset
③ π
④ σ

55. 분산 데이터베이스 목표 중 “데이터베이스의 분산된 물리적 환경에서 특정 지역의 컴퓨터 시스템이나 네트워크에 장애가 발생해도 데이터 무결성이 보장된다.”는 것과 관계있는 것은?

- ① 장애 투명성 ② 병행 투명성
③ 위치 투명성 ④ 중복 투명성

56. 다음 설명의 ()안에 들어갈 내용으로 적합한 것은?

후보키는 릴레이션에 있는 모든 튜플에 대해 유일성과 ()을 모두 만족시켜야 한다.

- ① 중복성 ② 최소성
③ 참조성 ④ 동일성

57. 다음 SQL문의 실행 결과는?

```
SELECT 가격 FROM 도서가격
WHERE 책번호=(SELECT 책번호 FROM 도서 WHERE
책명='자료구조');
```

[도서]		[도서가격]	
책번호	책명	책번호	가격
111	운영체제	111	20,000
222	자료구조	222	25,000
333	컴퓨터구조	333	10,000
		444	15,000

- ① 10,000 ② 15,000
③ 20,000 ④ 25,000

58. 데이터 제어언어(DCL)의 기능으로 옳지 않은 것은?

- ① 데이터 보안
- ② 논리적, 물리적 데이터 구조 정의
- ③ 무결성 유지
- ④ 병행수행 제어

59. 참조 무결성을 유지하기 위하여 DROP문에서 부모 테이블의 항목 값을 삭제할 경우 자동적으로 자식 테이블의 해당 레코드를 삭제하기 위한 옵션은?

- | | |
|------------|--------------|
| ① CLUSTER | ② CASCADE |
| ③ SET-NULL | ④ RESTRICTED |

60. 트랜잭션의 특성 중 다음 설명에 해당하는 것은?

트랜잭션의 연산은 데이터베이스에 모두 반영되든지 아니면 전혀 반영되지 않아야 한다.

- ① Durability
- ② Share
- ③ Consistency
- ④ Atomicity

61. UNIX의 셸(Shell)에 관한 설명으로 옳지 않은 것은?

- ① 명령어 해석기이다.
- ② 시스템과 사용자 간의 인터페이스를 담당한다.
- ③ 여러 종류의 셸이 있다.
- ④ 프로세스, 기억장치, 입출력 관리를 수행한다.

62. TCP/IP 프로토콜 중 전송계층 프로토콜은?

- [illegible]

63. C언어에서 비트 논리연산자에 해당하지 않는 것은?

- ① ^ ② ? ③ & ④ ~

64. 시스템에서 모듈 사이의 결합도(Coupling)에 대한 설명으로 옳은 것은?

- ① 한 모듈 내에 있는 처리요소들 사이의 기능적인 연관 정도를 나타낸다.
- ② 결합도가 높으면 시스템 구현 및 유지보수 작업이 쉽다.
- ③ 모듈 간의 결합도를 약하게 하면 모듈 독립성이 향상된다.
- ④ 자료결합도는 내용결합도보다 결합도가 높다.

65. 스레드(Thread)에 대한 설명으로 옳지 않은 것은?

- ① 한 개의 프로세스는 여러 개의 스레드를 가질 수 없다.
- ② 커널 스레드의 경우 운영체제에 의해 스레드를 운용한다.
- ③ 사용자 스레드의 경우 사용자가 만든 라이브러리를 사용하여 스레드를 운용한다.
- ④ 스레드를 사용함으로써 하드웨어, 운영체제의 성능과 응용 프로그램의 처리율을 향상시킬 수 있다.

66. C언어에서 배열 b[5]의 값은?

```
static int b[9]={1, 2, 3};
```

- ① 0 ② 1 ③ 2 ④ 3

67. 은행가 알고리즘(Banker's Algorithm)은 교착상태의 해결 방법 중 어떤 기법에 해당하는가?

- ① Avoidance ② Detection
③ Prevention ④ Recovery

68. IEEE 802.11 워킹 그룹의 무선 LAN 표준화 현황 중 QoS 강화를 위해 MAC 지원 기능을 채택한 것은?

- ① 802.11a ② 802.11b
③ 802.11g ④ 802.11e

69. TCP/IP 네트워크에서 IP주소를 MAC 주소로 변환하는 프로토콜은?

- ① UDP ② ARP
③ TCP ④ ICMP

70. HRN(Highest Response-ratio Next)스케줄링 방식에 대한 설명으로 옳지 않은 것은?

- ① 대기 시간이 긴 프로세스일 경우 우선순위가 높아진다.
- ② SJF 기법을 보완하기 위한 방식이다.
- ③ 긴 작업과 짧은 작업 간의 지나친 불평등을 해소할 수 있다.
- ④ 우선순위를 계산하여 그 수치가 가장 낮은 것부터 높은 순으로 우선순위가 부여된다.

71. 교착 상태 발생의 필요충분조건이 아닌 것은?

- ① 상호 배제(mutual exclusion)
- ② 점유와 대기(hold and wait)
- ③ 환형 대기(circular wait)
- ④ 선점(preemption)

72. 다음의 페이지 참조 열(Page reference)에 대해 페이지 교체 기법으로 선입선출 알고리즘을 사용할 경우 페이지 부재(Page Fault) 횟수는? (단, 할당된 페이지 프레임 수는 3 이고, 처음에는 모든 프레임이 비어 있다.)

<페이지 참조열>

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0

- ① 13 ② 14
③ 15 ④ 20
73. C언어에서 사용할 수 없는 변수명은?
① student2019 ② text-color
③ _korea ④ amount
74. IPv6에 대한 설명으로 틀린 것은?
① 128비트의 주소 공간을 제공한다.
② 인증 및 보안 기능을 포함하고 있다.
③ 패킷 크기가 64Kbyte로 고정되어 있다.
④ IPv6 확장 헤더를 통해 네트워크 기능 확장이 용이하다.
75. 프로세스 상태의 종류가 아닌 것은?
① Ready ② Running
③ Request ④ Exit
76. IPv6의 주소체제로 거리가 먼 것은?
① Unicast ② Anycast
③ Broadcast ④ Multicast
77. 응집도가 가장 낮은 것은?
① 기능적 응집도 ② 시간적 응집도
③ 절차적 응집도 ④ 우연적 응집도
78. JAVA 언어에서 접근제한자가 아닌 것은?
① public ② protected
③ package ④ private
79. 스크립트 언어가 아닌 것은?
① PHP ② Cobol
③ Basic ④ Python
80. OSI-7계층에서 종단 간 신뢰성 있고 효율적인 데이터를 전송하기 위해 오류 검출과 복구, 흐름 제어를 수행하는 계층은?
① 전송 계층 ② 세션 계층
③ 표현 계층 ④ 응용 계층

제5과목 : 정보시스템 구축 관리

81. 웹과 컴퓨터 프로그램에서 용량이 적은 데이터를 교환하기 위해 데이터 객체를 속성-값의 쌍 형태로 표현하는 형식으로 자바스크립트(JavaScript)를 토대로 개발되어진 형식은?
① Python ② XML
③ JSON ④ WEB SEVER
82. 최대 홉수를 15로 제한한 라우팅 프로토콜은?
① RIP ② OSPF
③ Static ④ EIGRP

83. IP 또는 ICMP의 특성을 악용하여 특정 사이트에 집중적으로 데이터를 보내 네트워크 또는 시스템의 상태를 불안으로 만드는 공격 방법은?

① TearDrop ② Smishing
③ Qshing ④ Smurfing

84. CMM(Capability Maturity Model) 모델의 레벨로 옳지 않은 것은?

① 최적단계 ② 관리단계
③ 정의단계 ④ 계획단계

85. 여러 개의 독립된 통신장치가 UWB(Ultra Wide Band)기술 또는 블루투스 기술을 사용하여 통신망을 형성하는 무선 네트워크 기술은?

① PICONET ② SCRUM
③ NFC ④ WI-SUN

86. COCOMO model 중 기관 내부에서 개발된 중소 규모의 소프트웨어로 일괄 자료 처리나 과학기술 계산용, 비즈니스 자료 처리용으로 5만 라인 이하의 소프트웨어를 개발하는 유형은?

① embeded ② organic
③ semi-detached ④ semi-embeded

87. 컴퓨터 사용자의 키보드 움직임을 탐지해 ID, 패스워드 등 개인의 중요한 정보를 몰래 빼가는 해킹 공격은?

① Key Logger Attack ② Worm
③ Rollback ④ Zombie Worm

88. LOC 기법에 의하여 예측된 총 라인수가 50000라인, 프로그래머의 월 평균 생산성이 200라인, 개발에 참여할 프로그래머가 10인 일 때, 개발 소요 기간은?

① 25개월 ② 50개월
③ 200개월 ④ 2000개월

89. Rayleigh-Norden 곡선의 노력 분포도를 이용한 프로젝트 비용 산정기법은?

① Putnam 모형 ② 델파이 모형
③ COCOMO 모형 ④ 기능점수 모형

90. 소인수 분해 문제를 이용한 공개키 암호화 기법에 널리 사용되는 암호 알고리즘 기법은?

① RSA ② ECC
③ PKI ④ PRM

91. 프로토타입을 지속적으로 발전시켜 최종 소프트웨어 개발까지 이르는 개발방법으로 위험관리가 중심인 소프트웨어 생명주기 모형은?

① 나선형 모형 ② 델파이 모형
③ 폭포수 모형 ④ 기능점수 모형

92. 다음 설명의 정보보안 침해 공격 관련 용어는?

인터넷 사용자의 컴퓨터에 침입해 내부 문서 파일 등을 암호화해 사용자가 열지 못하게 하는 공격으로, 암호 해독용 프로그램의 전달을 조건으로 사용자에게 돈을 요구하기도 한다.

① Smishing ② C-brain
③ Trojan Horse ④ Ransomware

93. 백도어 탐지 방법으로 틀린 것은?

① 무결성 검사 ② 닫힌 포트 확인
③ 로그 분석 ④ SetUID 파일 검사

94. 메모리상에서 프로그램의 복귀 주소와 변수사이에 특정 값을 저장해 두었다가 그 값이 변경되었을 경우 오버플로우 상태로 가정하여 프로그램 실행을 중단하는 기술은?

- ① 모드체크 ② 리커버리 통제
③ 시스로그 ④ 스택가드

95. 소프트웨어 개발 프레임워크를 적용할 경우 기대효과로 거리가 먼 것은?

- ① 품질 보증 ② 시스템 복잡도 증가
③ 개발 용이성 ④ 변경 용이성

96. 시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있는 보안 요소는?

- ① 기밀성 ② 부인방지
③ 가용성 ④ 무결성

97. 다음이 설명하는 용어로 옳은 것은?

- 오픈 소스를 기반으로 한 분산 컴퓨팅 플랫폼이다.
- 일반 PC급 컴퓨터들로 가상화된 대형 스토리지를 형성한다.
- 다양한 소스를 통해 생성된 빅데이터를 효율적으로 저장하고 처리한다.

- ① 하둡(Hadoop)
② 비컨(Beacon)
③ 포스퀘어(Foursquare)
④ 맴리스터(Memristor)

98. 크래커가 침입하여 백도어를 만들어 놓거나, 설정 파일을 변경했을 때 분석하는 도구는?

- ① trace ② tripwire
③ udpdump ④ cron

99. 폭포수 모형의 특징으로 거리가 먼 것은?

- ① 개발 중 발생한 요구사항을 쉽게 반영할 수 있다.
② 순차적인 접근방법을 이용한다.
③ 단계적 정의와 산출물이 명확하다.
④ 모형의 적용 경험과 성공사례가 많다.

100. 테일러링(Tailoring) 개발 방법론의 내부 기준에 해당하지 않는 것은?

- ① 납기/비용 ② 기술환경
③ 구성원 능력 ④ 국제표준 품질기준

정답 및 해설

1. ②	2. ②	3. ④	4. ①	5. ③	6. ④	7. ④	8. ③	9. ③	10. ③
11. ③	12. ①	13. ②	14. ①	15. ④	16. ①	17. ③	18. ④	19. ②	20. ④
21. ③	22. ②	23. ③	24. ①	25. ④	26. ③	27. ④	28. ②	29. ②	30. ③
31. ②	32. ①	33. ②	34. ④	35. ③	36. ④	37. ①	38. ④	39. ③	40. ①
41. ①	42. ③	43. ②	44. ④	45. ②	46. ①	47. ③	48. ①	49. ①	50. ③
51. ②	52. ④	53. ①	54. ①	55. ①	56. ②	57. ④	58. ②	59. ②	60. ④
61. ④	62. ④	63. ②	64. ③	65. ①	66. ①	67. ①	68. ④	69. ②	70. ④
71. ④	72. ②	73. ②	74. ③	75. ③	76. ③	77. ④	78. ③	79. ②	80. ①
81. ③	82. ①	83. ④	84. ④	85. ①	86. ②	87. ①	88. ①	89. ①	90. ①
91. ①	92. ④	93. ②	94. ④	95. ②	96. ④	97. ①	98. ②	99. ①	100. ④

- 1 GoF의 디자인 패턴은 유형에 따라 생성 패턴 5개, 구조 패턴 7개, 행위 패턴 11개 총 23개의 패턴으로 구성됩니다. ①, ③번은 생성 패턴, ④번은 구조 패턴에 해당합니다.

[병행학습]

디자인 패턴의 종류

- 생성 패턴(Creational Pattern)
 - 추상 팩토리(Abstract Factory), 빌더(Builder), 팩토리 메소드(Factory Method), 프로토타입(Prototype), 싱글톤(Singleton)
 - 구조 패턴(Structural Pattern)
 - 어댑터(Adapter), 브리지(Bridge), 컴포지트(Composite), 데코레이터(Decorator), 퍼사드(Facade), 플라이웨이트(Flyweight), 프록시(Proxy)
 - 행위 패턴(Behavioral Pattern)
 - 책임 연쇄(Chain of Responsibility), 커맨드(Command), 인터프리터(Interpreter), 반복자(Iterator), 중재자(Mediator), 메멘토(Memento), 옵서버(Observer), 상태(State), 전략(Strategic), 템플릿 메소드(Template Method), 방문자(Visitor)
- 2 객체지향 프로그래밍 언어의 구성 요소에는 객체, 클래스, 메시지가 있으며, 유사한 객체들을 묶어 추상화하는 것은 클래스(Class)입니다.

[병행학습]

객체지향 프로그래밍 언어의 구성 요소

- 객체(Object)
 - 데이터(속성)과 이를 처리하기 위한 연산(메소드)을 결합시킨 실체이다.
 - 데이터 구조와 그 위에서 수행되는 연산들을 가지고 있는 소프트웨어 모듈이다.
 - 속성(Attribute) : 한 클래스 내에 속한 객체들이 가지고 있는 데이터 값들을 단위별로 정의하는 것으로서 성질, 분류, 식별, 수량 또는 현재 상태 등을 표현한다.
 - 메소드(Method) : 객체가 메시지를 받아 실행해야 할 때 구체적인 연산을 정의하는 것으로, 객체의 상태를 참조하거나 변경하는 수단이 된다.
- 클래스(Class)
 - 두 개 이상의 유사한 객체들을 묶어서 하나의 공통된 특성을 표현하는 요소이다. 즉 공통된 특성과 행위를 갖는 객체의 집합이라고 할 수 있다.
 - 객체의 유형 또는 타입(Object Type)을 의미한다.
- 메시지(Message)

- 객체들 간에 상호작용을 하는데 사용되는 수단으로 객체의 메소드(동작, 연산)를 일으키는 외부의 요구 사항이다.
- 메시지를 받은 객체는 대응하는 연산을 수행하여 예상된 결과를 반환하게 된다.

- 3 클래스들 사이의 ‘부분-전체(Part-Whole)’ 관계 또는 ‘부분(is-a-part-of)’의 관계와 같이 하나의 사물이 다른 사물에 포함되어 있는 관계를 집합 또는 집단 관계라고 합니다.

[병행학습]

관계(Relationships)

- 연관(Association) 관계 : 2개 이상의 사물이 서로 관련되어 있음
- 집합(Aggregation) 관계 : 하나의 사물이 다른 사물에 포함되어 있는 관계
- 포함(Composition) 관계 : 집합 관계의 특수한 형태로, 포함하는 사물의 변화가 포함되는 사물에게 영향을 미치는 관계
- 일반화(Generalization) 관계 : 하나의 사물이 다른 사물에 비해 더 일반적인지 구체적인지를 표현하는 관계
- 의존(Dependency) 관계 : 연관 관계와 같이 사물 사이에 서로 연관은 있으나 필요에 의해 서로에게 영향을 주는 짧은 시간 동안만 연관을 유지하는 관계
- 실체화(Realization) 관계 : 사물이 할 수 있거나 해야 하는 기능(행위, 인터페이스)으로 서로를 그룹화 할 수 있는 관계

- 4 객체지향 분석 방법론 중 E-R 다이어그램을 사용하여 객체의 행위를 모델링하며, 객체 식별, 구조식별, 주체 정의, 속성 및 관계 정의, 서비스 정의 등의 과정으로 구성되는 것을 Coad와 Yourdon 방법이라고 합니다.

[병행학습]

객체지향 분석의 방법론

- Rumbaugh(럼바우) 방법 : 가장 일반적으로 사용되는 방법으로 분석 활동을 객체 모델, 동적 모델, 기능 모델로 나누어 수행하는 방법이다.
- Booch(부치) 방법 : 미시적(Micro) 개발 프로세스와 거시적(Macro) 개발 프로세스를 모두 사용하는 분석 방법으로, 클래스와 객체들을 분석 및 식별하고 클래스의 속성과 연산을 정의한다.
- Jacobson 방법 : Use Case를 강조하여 사용하는 분석 방법이다.
- Coad와 Yourdon 방법 : E-R 다이어그램을 사용하여 객체

1, 2회

의 행위를 모델링하며, 객체 식별, 구조 식별, 주제 정의, 속성과 인스턴스 연결 정의, 연산과 메시지 연결 정의 등의 과정으로 구성하는 기법이다.

- **Wirfs-Brock 방법** : 분석과 설계 간의 구분이 없고, 고객 명세서를 평가해서 설계 작업까지 연속적으로 수행하는 기법이다.

- 5 순차 코드는 자료의 발생순서, 크기순서 등 일정 기준에 따라서 최초의 자료부터 차례로 일련번호를 부여하는 방법입니다.

[병행학습]

코드(Code)

- 컴퓨터를 이용하여 자료를 처리하는 과정에서 분류·조합 및 집계를 용이하게 하고, 특정 자료의 추출을 쉽게 하기 위해서 사용하는 기호
- **순차 코드(Sequence Code)** : 자료의 발생순서, 크기순서 등 일정 기준에 따라서 최초의 자료부터 차례로 일련번호를 부여하는 방법
- **블록 코드(Block Code)** : 코드화 대상 항목 중에서 공통성이 있는 것끼리 블록으로 구분하고, 각 블록 내에서 일련번호를 부여하는 방법
- **10진 코드(Decimal Code)** : 코드화 대상 항목을 0~9까지 10진 분할하고, 다시 그 각각에 대하여 10진 분할하는 방법을 필요한 만큼 반복하는 방법
- **그룹 분류 코드(Group Classification Code)** : 코드화 대상 항목을 일정 기준에 따라 대분류, 중분류, 소분류 등으로 구분하고, 각 그룹 안에서 일련번호를 부여하는 방법
- **연상 코드(Mnemonic Code)** : 코드화 대상 항목의 명칭이나 약호와 관계있는 숫자나 문자, 기호를 이용하여 코드를 부여하는 방법
- **표의 숫자 코드(Significant Digit Code)** : 코드화 대상 항목의 성질, 즉 길이, 넓이, 부피, 지름, 높이 등의 물리적 수치를 그대로 코드에 적용시키는 방법
- **합성 코드(Combined Code)** : 필요한 기능을 하나의 코드로 수행하기 어려운 경우 2개 이상의 코드를 조합하여 만드는 방법

- 6 서버 튜닝은 서버의 성능 개선을 의미하는 것으로, 성능특성 분석에 사용되는 측정 항목이 될 수 없습니다.

[병행학습]

- **응답시간(Response Time)** : 요청을 전달한 시간부터 응답이 도착할 때까지 걸린 시간
- **가용성(Availability)** : 시스템을 사용할 필요가 있을 때 즉시 사용 가능한 정도
- **사용률(Utilization)** : 의뢰한 작업을 처리하는 동안의 CPU 사용량, 메모리 사용량, 네트워크 사용량 등 자원 사용률

- 7 ①번은 자료의 반복을, ②번은 자료의 설명을, ③번은 자료의 정의를 의미합니다.

[병행학습]

자료 사전(DD; Data Dictionary)

- 자료 흐름도에 있는 자료를 더 자세히 정의하고 기록한 것이며, 이처럼 데이터를 설명하는 데이터를 데이터의 데이터 또는 메타 데이터(Meta Data)라고 한다.
- 자료 흐름도에 시각적으로 표시된 자료에 대한 정보를 체계적이고 조직적으로 모아 개발자나 사용자가 편리하게 사용할 수 있다.
- 자료 사전에서 사용되는 표기 기호
 - = : 자료의 정의

- + : 자료의 연결
- () : 자료의 생략
- [] : 자료의 선택
- { } : 자료의 반복
- ** : 자료의 설명

- 8 검토회의 전에 요구사항 명세서를 미리 배포하여 사전 검토한 후 짧은 검토 회의를 통해 오류를 조기에 검출하는데 목적을 두는 요구사항 검토 방법은 워크 스루입니다.

[병행학습]

요구사항 검토(Requirements Review)

- 요구사항 명세서의 오류 확인 및 표준 준수 여부 등의 결함 여부를 검토 담당자들이 수작업으로 분석하는 방법
- 종류
 - 동료검토(Peer Review) : 요구사항 명세서 작성자가 명세서 내용을 직접 설명하고 동료들이 이를 들으면서 결함을 발견하는 형태의 검토 방법이다.
 - 워크스루(Walk Through) : 검토 회의 전에 요구사항 명세서를 미리 배포하여 사전 검토한 후에 짧은 검토 회의를 통해 결함을 발견하는 형태의 검토 방법이다.
 - 인스펙션(Inspection) : 요구사항 명세서 작성자를 제외한 다른 검토 전문가들이 요구사항 명세서를 확인하면서 결함을 발견하는 형태의 검토 방법이다.

- 9 CASE의 주요 기능에는 소프트웨어 생명주기 전 단계의 연결, 다양한 소프트웨어 개발 모형 지원, 그래픽 지원 등이 있습니다.

[병행학습]

CASE(Computer Aided Software Engineering)

- CASE는 소프트웨어 개발 과정에서 사용되는 요구 분석, 설계, 구현, 검사 및 디버깅 과정 전체 또는 일부를 컴퓨터와 전용 소프트웨어 도구를 사용하여 자동화하는 것이다.
- 소프트웨어, 하드웨어, 데이터베이스, 테스트 등을 통합하여 소프트웨어를 개발하는 환경을 조성한다.
- 소프트웨어 생명 주기의 전체 단계를 연결해 주고 자동화해 주는 통합된 도구를 제공해 주는 기술이다.
- 소프트웨어 개발 도구와 방법론이 결합된 것으로, 정형화된 구조 및 방법(메커니즘)을 소프트웨어 개발에 적용하여 생산성 향상을 구현하는 공학 기법이다.
- 소프트웨어 개발의 모든 단계에 걸쳐 일관된 방법론을 제공하는 자동화 도구(CASE Tool)들을 지원하고, 개발자들은 이 도구를 사용하여 소프트웨어 개발의 표준화를 지향하며, 자동화의 이점을 얻을 수 있게 해준다.
- **CASE의 주요 기능** : 소프트웨어 생명주기 전 단계의 연결, 다양한 소프트웨어 개발 모형 지원, 그래픽 지원 등

- 10 XP의 5가지 핵심 가치에는 의사소통(Communication), 단순성(Simplicity), 용기(Courage), 존중(Respect), 피드백(Feedback)이 있습니다.

[병행학습]

XP(eXtreme Programming)

- XP는 수시로 발생하는 고객의 요구사항에 유연하게 대응하기 위해 고객의 참여와 개발 과정의 반복을 극대화하여 개발 생산성을 향상시키는 방법이다.
- XP는 짧은 반복적인 개발 주기, 단순한 설계, 고객의 적극적인 참여를 통해 소프트웨어를 빠르게 개발하는 것을 목적으로 한다.
- 릴리즈의 기간을 짧게 반복하면서 고객의 요구사항 반영에 대한 가시성을 높인다.
- 릴리즈 테스트마다 고객을 직접 참여시킴으로써 요구한

- 기능이 제대로 작동하는지 고객이 직접 확인할 수 있다.
- 비교적 소규모 인원의 개발 프로젝트에 효과적이다.
- **XP의 5가지 핵심 가치** : 의사소통(Communication), 단순성(Simplicity), 용기(Courage), 존중(Respect), 피드백(Feedback)

- 11 DBMS 분석 시 고려사항에는 가용성, 성능, 기술 지원, 상호 호환성, 구축 비용이 있습니다.

[병행학습]

DBMS 관련 요구사항 식별 시 고려사항

- 가용성
 - 시스템의 장시간 운영으로 인해 발생할 수 있는 운영체제 고유의 장애 발생 가능성
 - DBMS의 결함 등으로 인한 패치 설치를 위한 재가동
 - 백업이나 복구의 편의성
 - DBMS 이중화 및 복제 지원
- 성능
 - 대규모 데이터 처리 성능(분할 테이블 지원 여부)
 - 대용량 트랜잭션 처리 성능
 - 튜닝 옵션의 다양한 지원
 - 최소화된 설정과 비용 기반 질의 최적화 지원
- 기술 지원
 - 제작업체의 안정적인 기술 지원
 - 여러 사용자들 간의 정보 공유
 - 오픈 소스 여부
- 상호 호환성
 - 설치 가능한 운영체제의 종류
 - JDBC, ODBC와의 호환 여부
- 구축 비용
 - 라이선스 정책 및 비용
 - 유지관리 비용
 - 총 소유 비용(TCO)

- 12 HIPO는 시스템의 분석 및 설계나 문서화할 때 사용되는 기법으로, 하향식 소프트웨어 개발을 위한 문서화 도구입니다.

[병행학습]

HIPO(Hierarchy Input Process Output)

- 시스템의 분석 및 설계나 문서화할 때 사용되는 기법으로, 시스템 실행 과정인 입력, 처리, 출력의 기능을 나타낸다.
- 기본 시스템 모델은 입력, 처리, 출력으로 구성되며, 하향식 소프트웨어 개발을 위한 문서화 도구이다.
- 체계적인 문서 관리가 가능하다.
- 기호, 도표 등을 사용하므로 보기 쉽고 이해하기도 쉽다.
- 기능과 자료의 의존 관계를 동시에 표현할 수 있다.
- 변경, 유지보수가 용이하다.
- 시스템의 기능을 여러 개의 고유 모듈들로 분할하여 이들 간의 인터페이스를 계층 구조로 표현한 것을 HIPO Chart라고 한다.
- HIPO Chart의 종류
 - 가시적 도표(도식 목차) : 시스템의 전체적인 기능과 흐름을 보여주는 계층(Tree) 구조도
 - 총체적 도표(총괄도표, 개요 도표) : 프로그램을 구성하는 기능을 기술한 것으로 입력, 처리, 출력에 대한 전반적인 정보를 제공하는 도표
 - 세부적 도표(상세 도표) : 총체적 도표에 표시된 기능을 구성하는 기본 요소들을 상세히 기술하는 도표

- 13 누구나 쉽게 이해하고 사용할 수 있어야 한다는 UI의 설계 원칙은 직관성입니다.

[병행학습]

사용자 인터페이스(UI)의 기본 원칙

- **직관성** : 누구나 쉽게 이해하고 사용할 수 있어야 한다.
- **유효성** : 사용자의 목적을 정확하고 완벽하게 달성해야 한다.
- **학습성** : 누구나 쉽게 배우고 익힐 수 있어야 한다.
- **유연성** : 사용자의 요구사항을 최대한 수용하고 실수를 최소화해야 한다.

- 14 린바우 분석 기법의 활동은 객체 모델링, 동적 모델링, 기능 모델링 순으로 이루어집니다.

[병행학습]

림바우(Rumbaugh)의 분석 기법

- 모든 소프트웨어 구성 요소를 그래픽 표기법을 이용하여 모델링하는 기법으로, 객체 모델링 기법(OMT, Object-Modeling Technique)이라고도 한다.
- 분석 활동은 객체 모델링, 동적 모델링, 기능 모델링을 통해 이루어진다.
- 객체 모델링(Object Modeling)
 - 정보 모델링이라고도 하며, 시스템에서 요구되는 객체를 찾아내어 속성과 연산 식별 및 객체들 간의 관계를 규정하여 객체 다이어그램으로 표시하는 것이다.
 - 분석 활동의 세 가지 모델 중 가장 중요하며 선행되어야 할 모델링이다.
- 동적 모델링(Dynamic Modeling)
 - 상태 다이어그램(상태도)을 이용하여 시간의 흐름에 따른 객체들 간의 제어 흐름, 상호 작용, 동작 순서 등의 동적인 행위를 표현하는 모델링이다.
 - 동적 모델링에서는 객체나 클래스의 상태, 사건을 중심으로 다룬다.
- 기능 모델링(Functional Modeling)
 - 자료 흐름도(DFD)를 이용하여 다수의 프로세스들 간의 자료 흐름을 중심으로 처리 과정을 표현한 모델링이다.
 - 어떤 데이터를 입력하여 어떤 결과를 구할 것인지를 표현하는 것이다.

- 15 데이터(자료) 흐름도는 요구사항 분석에서 자료의 흐름 및 변환 과정과 기능을 도형 중심으로 기술하는 방법으로, 구성 요소에는 프로세스(Process), 자료 흐름(Data Flow), 자료 저장소(Data Store), 단말(Terminator)이 있습니다.

[병행학습]

자료 흐름도(DFD; Data Flow Diagram)

- 요구사항 분석에서 자료의 흐름 및 변환 과정과 기능을 도형 중심으로 기술하는 방법으로 자료 흐름 그래프, 버블 차트라고도 한다.
- 시스템 안의 프로세스와 자료 저장소 사이에 자료의 흐름을 나타내는 그래프로 자료 흐름과 처리를 중심으로 하는 구조적 분석 기법에 이용된다.
- 자료 흐름도는 자료 흐름과 기능을 자세히 표현하기 위해 단계적으로 세분화된다.
- 자료는 처리(Process)를 거쳐 변환될 때마다 새로운 이름이 부여되며, 처리는 입력 자료가 발생하면 기능을 수행한 후 출력 자료를 산출한다.
- 자료 흐름도 구성 요소 표기법
 - 프로세스(Process)
 - ▶ 자료를 변환시키는 시스템의 한 부분(처리 과정)을 나타내며 처리, 기능, 변환, 버블이라고도 한다.
 - ▶ 원이나 둥근 사각형으로 표시하고 그 안에 프로세스 이름을 기입한다.
 - 자료 흐름(Data Flow)
 - ▶ 자료의 이동(흐름)이나 연관관계를 나타낸다.
 - ▶ 화살표 위에 자료의 이름을 기입한다.

1, 2회

- 자료 저장소(Data Store)
 - ▶ 시스템에서의 자료 저장소(파일, 데이터베이스)를 나타낸다.
 - ▶ 도형 안에 자료 저장소 이름을 기입한다.
- 단말(Terminator)
 - ▶ 시스템과 교신하는 외부 개체로, 입력 데이터가 만들어지고 출력 데이터를 받는다(정보의 생산자와 소비자).
 - ▶ 도형 안에 이름을 기입한다.

16 UML에서 표현하는 기본 기능 외에 추가적인 기능을 표현하는 스테레오 타입은 길러멧(Guillemet, << >>)이라고 부르는 겹화살괄호 사이에 기능을 기술합니다.

17 TP-Monitor(Transaction Processing Monitor)는 항공기나 철도 예약 업무 등과 같은 온라인 트랜잭션 업무에서 트랜잭션을 처리 및 감시하는 미들웨어입니다.

[병행학습]

미들웨어(Middleware)의 개념 및 종류

- DB(DataBase) : 데이터베이스 벤더(vendor)에서 제공하는 클라이언트에서 원격의 데이터베이스와 연결하기 위한 미들웨어
- RPC(Remote Procedure Call) : 응용 프로그램의 프로시저를 사용하여 원격 프로시저를 마치 로컬 프로시저처럼 호출하는 방식의 미들웨어
- MOM(Message Oriented Middleware) : 메시지 기반의 비동기형 메시지를 전달하는 방식의 미들웨어
- ORB(Object Request Broker) : 객체 지향 미들웨어로 코바(CORBA) 표준 스펙을 구현한 미들웨어
- WAS(Web Application Server) : 사용자의 요구에 따라 변하는 동적인 콘텐츠를 처리하기 위해 사용되는 미들웨어

18 요구사항 분석(Requirement Analysis)은 개발 대상에 대한 사용자의 요구사항 중 명확하지 않거나 모호하여 이해되지 않는 부분을 발견하고 이를 걸러내는 요구사항 개발 프로세스의 한 과정으로, 실제 명세서 작성을 하는 과정은 요구사항 명세(Requirement Specification)에 해당합니다.

[병행학습]

요구사항 분석(Requirement Analysis)

- 개발 대상에 대한 사용자의 요구사항 중 명확하지 않거나 모호하여 이해되지 않는 부분을 발견하고 이를 걸러내기 위한 과정이다.
- 사용자 요구사항의 타당성을 조사하고 비용과 일정에 대한 제약을 설정한다.
- 내용이 중복되거나 하나로 통합되어야 하는 등 서로 상충되는 요구사항이 있으면 이를 해결한다.
- 도출된 요구사항들을 토대로 소프트웨어의 범위를 파악한다.
- 도출된 요구사항들을 토대로 소프트웨어와 주변 환경이 상호 작용하는 방법을 이해한다.

19 공통 모듈에 대한 명세 기법 중 해당 기능에 대해 일관되게 이해되고 한 가지로 해석될 수 있도록 작성하는 원칙을 명확성(Clarity)이라고 합니다.

[병행학습]

공통 모듈의 명세 기법

- 정확성(Correctness) : 시스템 구현 시 해당 기능이 필요하다는 것을 알 수 있도록 정확히 작성한다.
- 명확성(Clarity) : 해당 기능을 이해할 때 중의적으로 해석

되지 않도록 명확하게 작성한다.

- 완전성(Completeness) : 시스템 구현을 위해 필요한 모든 것을 기술한다.
- 일관성(Consistency) : 공통 기능들 간 상호 충돌이 발생하지 않도록 작성한다.
- 추적성(Traceability) : 기능에 대한 요구사항의 출처, 관련 시스템 등의 관계를 파악할 수 있도록 작성한다.

20 구조적 다이어그램에는 클래스, 객체, 컴포넌트, 배치, 복합체 구조, 패키지 다이어그램이 있습니다.

[병행학습]

UML 다이어그램의 종류

- 구조적(Structural) 다이어그램의 종류
 - 클래스 다이어그램(Class Diagram) : 클래스와 클래스가 가지는 속성, 클래스 사이의 관계를 표현
 - 객체 다이어그램(Object Diagram) : 클래스에 속한 사물(객체)들, 즉 인스턴스(Instance)를 특정 시점의 객체와 객체 사이의 관계로 표현
 - 컴포넌트 다이어그램(Component Diagram) : 실제 구현 모듈인 컴포넌트 간의 관계나 컴포넌트 간의 인터페이스를 표현
 - 배치 다이어그램(Deployment Diagram) : 결과물, 프로세스, 컴포넌트 등 물리적 요소들의 위치를 표현
 - 복합체 구조 다이어그램(Composite Structure Diagram) : 클래스나 컴포넌트가 복합 구조를 갖는 경우 그 내부 구조를 표현
 - 패키지 다이어그램(Package Diagram) : 유스케이스나 클래스 등의 모델 요소들을 그룹화한 패키지들의 관계를 표현
- 행위(Behavioral) 다이어그램의 종류
 - 유스케이스 다이어그램(Use Case Diagram) : 사용자의 요구를 분석하는 것으로 기능 모델링 작업에 사용함
 - 시퀀스 다이어그램(Sequence Diagram) : 상호 작용하는 시스템이나 객체들이 주고받는 메시지를 표현
 - 커뮤니케이션 다이어그램(Communication Diagram) : 시퀀스 다이어그램과 같이 동작에 참여하는 객체들이 주고받는 메시지를 표현하는데, 메시지뿐만 아니라 객체들 간의 연관까지 표현
 - 상태 다이어그램(State Diagram) : 하나의 객체가 자신이 속한 클래스의 상태 변화 혹은 다른 객체와의 상호 작용에 따라 상태가 어떻게 변화하는지를 표현
 - 활동 다이어그램(Activity Diagram) : 시스템이 어떤 기능을 수행하는지 객체의 처리 로직이나 조건에 따른 처리의 흐름을 순서에 따라 표현
 - 상호작용 개요 다이어그램(Interaction Overview Diagram) : 상호작용 다이어그램 간의 제어 흐름을 표현
 - 타이밍 다이어그램(Timing Diagram) : 객체 상태 변화와 시간 제약을 명시적으로 표현

21 경계값 분석 기법은 입력 조건의 경계값을 테스트 케이스로 선정하여 검사하는 기법으로, 성적이 분리되는 평가점수의 경계값인 101, 100, 80, 79, 60, 59, 0, -1이 적절한 입력값에 해당한다.

[병행학습]

경계값 분석(Boundary Value Analysis)

- 블랙박스 테스트의 종류 중 하나로, 입력 자료에만 치중한 동치 분할 기법을 보완하기 위한 기법이다.
- 입력 조건의 중간값보다 경계값에서 오류가 발생될 확률이 높다는 점을 이용하여 입력 조건의 경계값을 테스트 케이스로 선정하여 검사하는 기법이다.

22 트리의 차수는 가장 차수가 많은 노드의 차수이고, 단말 노드는 자식이 하나도 없는 노드입니다. A, C, E의 차수 2가 차수 중 가장 높으므로 트리의 차수는 2가 되고, 자식이 하나도 없는 노드는 D, G, H, F로 총 4개가 됩니다.

23 검증 검사 기법 중 개발자의 장소에서 사용자가 개발자 앞에서 행하는 기법이며, 일반적으로 통제된 환경에서 사용자와 개발자가 함께 확인하면서 수행되는 검사를 알파 테스트라고 합니다.

[병행학습]

인수 테스트의 종류

- **사용자 인수 테스트** : 사용자가 시스템 사용의 적절성 여부를 확인
- **운영상의 인수 테스트** : 시스템 관리자가 시스템 인수 시 수행하는 테스트 기법으로, 백업/복원 시스템, 재난 복구, 사용자 관리, 정기 점검 등을 확인
- **계약 인수 테스트** : 계약상의 인수/검수 조건을 준수하는지 여부를 확인
- **규정 인수 테스트** : 소프트웨어가 정부 지침, 법규, 규정 등 규정에 맞게 개발되었는지 확인
- **알파 테스트** : 개발자의 장소에서 사용자가 개발자 앞에서 행하는 테스트 기법으로, 통제된 환경에서 행해지며, 오류와 사용상의 문제점을 사용자와 개발자가 함께 확인하면서 기록한다.
- **베타 테스트** : 선정된 최종 사용자가 여러 명의 사용자 앞에서 행하는 테스트 기법으로, 실업무를 가지고 사용자가 직접 테스트하는 것으로, 개발자에 의해 제어되지 않은 상태에서 테스트가 행해지며, 발견된 오류와 사용상의 문제점을 기록하고 개발자에게 주기적으로 보고한다.

24 하향식 통합 테스트에서 모듈 간의 통합 시험을 위해 일시적으로 필요한 조건만을 가지고 임시로 제공되는 시험용 모듈을 스텝(Stub)이라고 합니다.

[병행학습]

하향식 통합 테스트(Top Down Integration Test)

- 프로그램의 상위 모듈에서 하위 모듈 방향으로 통합하면서 테스트하는 기법이다.
- 주요 제어 모듈을 기준으로 하여 아래 단계로 이동하면서 통합하는데, 이때 깊이 우선 통합법이나 넓이 우선 통합법을 사용한다.
- 테스트 초기부터 사용자에게 시스템 구조를 보여줄 수 있다.
- 상위 모듈에서는 테스트 케이스를 사용하기 어렵다.
- 하향식 통합 방법의 절차
 - ① 주요 제어 모듈은 작성된 프로그램을 사용하고, 주요 제어 모듈의 종속 모듈들은 스텝(Stub)으로 대체한다.
 - ② 깊이 우선 또는 넓이 우선 등의 통합 방식에 따라 하위 모듈인 스텝들이 한 번에 하나씩 실제 모듈로 교체된다.
 - ③ 모듈이 통합될 때마다 테스트를 실시한다.
 - ④ 새로운 오류가 발생하지 않음을 보증하기 위해 회귀 테스트를 실시한다.
- ※ **테스트 스텝(Test Stub)** : 제어 모듈이 호출하는 타 모듈의 기능을 단순히 수행하는 도구로, 일시적으로 필요한 조건만을 가지고 있는 시험용 모듈

25 소프트웨어 품질 요구사항에서 고려되어야 할 특성 중 무결성과 간결성이라는 특성은 존재하지 않습니다. 하지만 무결성은 정확성을 보장하기 위해 기본적으로 전제되어야 하는 특성입니다.

[병행학습]

ISO/IEC 9126의 소프트웨어 품질 특성

- **기능성(Functionality)** : 적절성/정합성(Suitability), 정밀성/정확성(Accuracy), 상호 운용성(Interoperability), 보안성(Security), 호환성(Compliance)
- **신뢰성(Reliability)** : 성숙성(Maturity), 고장 허용성(Fault Tolerance), 회복성(Recoverability)
- **사용성(Usability)** : 이해성(Understandability), 학습성(Learnability), 운용성(Operability), 친밀성(Attractiveness)
- **효율성(Efficiency)** : 시간 효율성(Time Behaviour), 자원 효율성(Resource Behaviour)
- **유지 보수성(Maintainability)** : 분석성(Analyzability), 변경성(Changeability), 안정성(Stability), 시험성(Testability)
- **이식성(Portability)** : 적용성(Adaptability), 설치성(Installability), 대체성(Replaceability), 공존성(Co-existence)

26 소프트웨어 테스트에서 오류의 80%는 전체 모듈의 20% 내에서 발견된다는 법칙은 파레토 법칙(Pareto Principle)입니다.

[병행학습]

애플리케이션 테스트의 기본 원리

- 애플리케이션 테스트는 소프트웨어의 잠재적인 결함을 줄일 수 있지만 소프트웨어에 결함이 없다고 증명할 수는 없다. 즉 완벽한 소프트웨어 테스트는 불가능하다.
- 애플리케이션의 결함은 대부분 개발자의 특성이나 애플리케이션의 기능적 특징 때문에 특정 모듈에 집중되어 있다. 애플리케이션의 20%에 해당하는 코드에서 전체 80%의 결함이 발견된다고 하여 파레토 법칙을 적용하기도 한다.
- 애플리케이션 테스트에서는 동일한 테스트 케이스로 동일한 테스트를 반복하면 더 이상 결함이 발견되지 않는 ‘살충제 패러독스(Pesticide Paradox)’ 현상이 발생한다. 살충제 패러독스를 방지하기 위해서 테스트 케이스를 지속적으로 보완 및 개선해야 한다.
- 애플리케이션 테스트는 소프트웨어 특징, 테스트 환경, 테스트 역량 등 정황(Context)에 따라 테스트 결과가 달라질 수 있으므로, 정황에 따라 테스트를 다르게 수행해야 한다.
- 소프트웨어의 결함을 모두 제거해도 사용자의 요구사항을 만족시키지 못하면 해당 소프트웨어는 품질이 높다고 말할 수 없다. 이것을 오류-부재의 궤변(Absence of Errors Fallacy)이라고 한다.
- 테스트와 위험은 반비례한다. 테스트를 많이 하면 할수록 미래에 발생할 위험을 줄일 수 있다.
- 테스트는 작은 부분에서 시작하여 점점 확대하며 진행해야 한다.
- 테스트는 개발자와 관계없는 별도의 팀에서 수행해야 한다.

27 방화벽 기술은 디지털 저작권 관리 기술이 아닌 기업이나 조직 내부의 네트워크와 인터넷 간에 전송되는 정보를 선별하여 수용·거부·수정하는 기능을 가진 침입 차단 시스템입니다.

[병행학습]

디지털 저작권 관리(DRM)의 기술 요소

- **암호화(Encryption)** : 콘텐츠 및 라이선스를 암호화하고 전자 서명을 할 수 있는 기술
- **키 관리(Key Management)** : 콘텐츠를 암호화한 키에 대한 저장 및 분배 기술

1, 2회

- **암호화 파일 생성(Packager)** : 콘텐츠를 암호화된 콘텐츠로 생성하기 위한 기술
- **식별 기술(Identification)** : 콘텐츠에 대한 식별 체계 표현 기술
- **저작권 표현(Right Expression)** : 라이선스의 내용 표현 기술
- **정책 관리(Policy Management)** : 라이선스 발급 및 사용에 대한 정책 표현 및 관리 기술
- **크랙 방지(Tamper Resistance)** : 크랙에 의한 콘텐츠 사용 방지 기술
- **인증(Authentication)** : 라이선스 발급 및 사용의 기준이 되는 사용자 인증 기술

28 SMTP(Simple Mail Transfer Protocol)는 전자 우편을 교환하는 서비스로, 인터페이스 보안과는 무관합니다.

[병행학습]

영역별 인터페이스 보안

- 네트워크 영역
 - 인터페이스 송·수신 간 스니핑(Sniffing) 등을 이용한 데이터 탈취 및 변조 위협을 방지하기 위해 네트워크 트래픽에 대한 암호화를 설정한다.
 - 암호화는 인터페이스 아키텍처에 따라 IPsec, SSL, S-HTTP 등의 다양한 방식으로 적용한다.
- 애플리케이션 영역
 - 소프트웨어 개발 보안 가이드를 참조하여 애플리케이션 코드 상의 보안 취약점을 보완하는 방향으로 애플리케이션 보안 기능을 적용한다.
- 데이터베이스 영역
 - 데이터베이스, 스키마, 엔티티의 접근 권한과 프로시저(Procedure), 트리거(Trigger) 등 데이터베이스 동작 객체의 보안 취약점에 보안 기능을 적용한다.
 - 개인 정보나 업무상 민감한 데이터의 경우 암호화나 익명화 등 데이터 자체의 보안 방안도 고려한다.

29 인터페이스 구현 검증 도구 중 서비스 호출, 컴포넌트 재사용 등 다양한 환경을 지원하는 테스트 프레임워크는 STAF입니다.

[병행학습]

인터페이스 구현 검증 도구

- **xUnit** : Java(Junit), C++(Cppunit), .Net(Nunit) 등 다양한 언어를 지원하는 단위 테스트 프레임워크
- **STAF** : 서비스 호출 및 컴포넌트 재사용 등 다양한 환경을 지원하는 테스트 프레임워크
- **FitNesse** : 웹 기반 테스트케이스 설계, 실행, 결과 확인 등을 지원하는 테스트 프레임워크
- **NTAF** : FitNesse의 장점인 협업 기능과 STAF의 장점인 재사용 및 확장성을 통합한 NHN(Naver)의 테스트 자동화 프레임워크
- **Selenium** : 다양한 브라우저 및 개발 언어를 지원하는 웹 애플리케이션 테스트 프레임워크
- **watir** : Ruby를 사용하는 애플리케이션 테스트 프레임워크

30 SW 패키징 도구 사용 시에는 사용자의 편의성을 고려하여 다양한 기종과 호환되도록 해야 합니다.

[병행학습]

제품 소프트웨어 패키징 도구

- 제품 소프트웨어의 배포를 위한 패키징 시 디지털 콘텐츠의 지적 재산권의 보호 및 관리 기능과 안전한 유통과 배포를 보장하는 도구이다.

- 제품 소프트웨어 패키징 도구는 불법적인 복제로부터 디지털 콘텐츠의 지적 재산권을 보호해 주는 사용 권한 제어 기술, 패키징 기술, 라이선스 관리, 권한 통제 기술 등을 포함한다.
- 제품 소프트웨어 패키징 도구 활용 시 고려할 사항은 다음과 같다.
 - 패키징 시 사용자에게 배포되는 소프트웨어이므로 내부 콘텐츠에 대한 암호화 및 보안을 고려한다.
 - 다른 여러 콘텐츠 및 단말기 간 DRM(디지털 저작권 관리) 연동을 고려한다.
 - 사용자의 편의성을 위한 복잡성 및 비효율성 문제를 고려한다.
 - 제품 소프트웨어의 종류에 적합한 암호화 알고리즘을 적용한다.

31 형상 관리는 소프트웨어의 개발 과정에서 소프트웨어의 변경 사항을 관리하기 위해 개발된 일련의 활동을 의미합니다.

[병행학습]

형상 관리(SCM; Software Configuration Management)

- 소프트웨어의 개발 과정에서 소프트웨어의 변경 사항을 관리하기 위해 개발된 일련의 활동이다.
- 소프트웨어 변경의 원인을 알아내고 제어하며, 적절히 변경되고 있는지 확인하여 해당 담당자에게 통보한다.
- 형상 관리는 소프트웨어 개발의 전 단계에 적용되는 활동이며, 유지보수 단계에서도 수행된다.
- 형상 관리는 소프트웨어 개발의 전체 비용을 줄이고, 개발 과정의 여러 방해 요인이 최소화되도록 보증하는 것을 목적으로 한다.
- 형상 관리 기능의 종류
 - 형상 식별 : 형상 관리 대상에 이름과 관리 번호를 부여하고, 계층(Tree) 구조로 구분하여 수정 및 추적이 용이하도록 하는 작업
 - 버전 제어 : 소프트웨어 업그레이드나 유지 보수 과정에서 생성된 다른 버전의 형상 항목을 관리하고, 이를 위해 특정 절차와 도구(Tool)를 결합시키는 작업
 - 형상 통제(변경 관리) : 식별된 형상 항목에 대한 변경 요구를 검토하여 현재의 기준선(Base Line)이 잘 반영될 수 있도록 조정하는 작업
 - 형상 감사 : 기준선의 무결성을 평가하기 위해 확인, 검증, 검열 과정을 통해 공식적으로 승인하는 작업
 - 형상 기록(상태 보고) : 형상의 식별, 통제, 감사 작업의 결과를 기록·관리하고 보고서를 작성하는 작업

32 대표적인 화이트박스 테스트의 종류에는 기초 경로 검사, 제어 구조 검사가 있습니다. 경계값 분석(Boundary Value Analysis)은 블랙박스 테스트의 종류에 해당합니다.

[병행학습]

화이트 박스 테스트의 종류

- **기초 경로 검사** : 대표적인 화이트박스 테스트 기법
- **제어 구조 검사**
 - 조건 검사(Condition Testing) : 프로그램 모듈 내에 있는 논리적 조건을 테스트하는 테스트 케이스 설계 기법
 - 루프 검사(Loop Testing) : 프로그램의 반복(Loop) 구조에 초점을 맞춰 실시하는 테스트 케이스 설계 기법
 - 데이터 흐름 검사(Data Flow Testing) : 프로그램에서 변수의 정의와 변수 사용의 위치에 초점을 맞춰 실시하는 테스트 케이스 설계 기법

블랙 박스 테스트의 종류

- **동치 분할 검사(Equivalence Partitioning Testing)** : 입력 자료에 초점을 맞춰 테스트 케이스를 만들고 검사하는

방법으로 동등 분할 기법이라고도 함

- **경계값 분석(Boundary Value Analysis)** : 입력 자료에만 치중한 동치 분할 기법을 보완하기 위한 기법
- **원인-효과 그래프 검사(Cause-Effect Graphing Testing)** : 입력 데이터 간의 관계와 출력에 영향을 미치는 상황을 체계적으로 분석한 다음 효용성이 높은 테스트 케이스를 선정하여 검사하는 기법
- **오류 예측 검사(Error Guessing)** : 과거의 경험이나 확인자의 감각으로 테스트하는 기법
- **비교 검사(Comparison Testing)** : 여러 버전의 프로그램에 동일한 테스트 자료를 제공하여 동일한 결과가 출력되는지 테스트하는 기법

- 33 아주 오래되거나 참고문서 또는 개발자가 없어 유지보수 작업이 어려운 프로그램을 외계인 코드(Alien Code)라고 합니다. 소프트웨어에 대한 변경이 자주 발생하는 유지보수 시 자주 사용되는 용어입니다.

[병행학습]

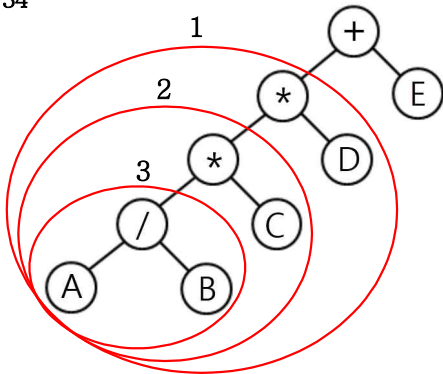
유지보수의 부작용

- 유지보수 활동을 통해 예기치 못한 부작용이 발생할 수 있으며 그 종류는 다음과 같다.
- **코딩 부작용** : 코딩 내용의 변경으로 인해 발생하는 부작용
- **자료 부작용** : 자료나 자료 구조의 변경으로 인해 발생하는 부작용
- **문서화 부작용** : 자료 코드에 대한 변경이 설계문서나 사용자가 사용하는 매뉴얼에 적용되지 않을 때에 발생하는 부작용

외계인 코드(Alien Code)

- 외계인 코드는 아주 오래 전에 개발되어 유지보수 작업이 매우 어려운 프로그램을 의미한다.
- 일반적으로 15년 전 또는 그 전에 개발된 프로그램을 의미하며, 문서화(Documentation)를 철저히 해두면 방지할 수 있다.

34



- ① Preorder는 Root → Left → Right이므로 +1E입니다.
- ② 1은 *2D이므로 +*2DE입니다.
- ③ 2는 *3C이므로 +**3CDE입니다.
- ④ 3은 /AB이므로 +**/ABCDE입니다.

- 35 O(1)은 빅오 표기법의 시간 복잡도를 표기하는 방법의 하나로, 입력 데이터 수에 관계없이 문제 해결에 하나의 단계만을 거친다는 것을 의미합니다.

[병행학습]

빅오 표기법(Big-O Notation)

- 알고리즘의 실행시간이 최악일 때를 표기하는 방법으로, 신뢰성이 떨어지는 오메가 표기법이나 평가하기 까다로운 세타 표기법에 비해 성능을 예측하기 용이하여 주로 사용

되는 표기법이다.

- O(1)
 - 입력값(n)에 관계 없이 일정하게 문제 해결에 하나의 단계만을 거친다.
 - [예] 스택의 삽입(Push), 삭제(Pop)
- O(logn)
 - 문제 해결에 필요한 단계가 입력값(n) 또는 조건에 의해 감소한다.
 - [예] 이진 트리(Binary Tree), 이진 검색(Binary Search)
- O(n)
 - 문제 해결에 필요한 단계가 입력값(n)과 1:1의 관계를 가진다.
 - [예] for문
- O(nlogn)
 - 문제 해결에 필요한 단계가 n(log2n)번만큼 수행된다.
 - [예] 힙 정렬(Heap Sort), 2-Way 합병 정렬(Merge Sort)
- O(n²)
 - 문제 해결에 필요한 단계가 입력값(n)의 제곱만큼 수행된다.
 - [예] 삽입 정렬(Insertion Sort), 쉘 정렬(Shell Sort), 선택 정렬(Selection Sort), 버블 정렬(Bubble Sort), 퀵 정렬(Quick Sort)
- O(2ⁿ)
 - 문제 해결에 필요한 단계가 2의 입력값(n) 제곱만큼 수행된다.
 - [예] 피보나치 수열(Fibonacci Sequence)

- 36 nlogn에 해당하는 정렬 알고리즘에는 힙 정렬과 2-Way 합병 정렬이 있습니다. ①, ②, ③번의 각 정렬의 시간 복잡도는 O(n²)입니다.

- 37 ISO/IEC 9126의 기능성(Functionality)의 하위 특성에는 적절성/정합성(Suitability), 정밀성/정확성(Accuracy), 상호운용성(Interoperability), 보안성(Security), 호환성(Compliance)이 있습니다.

- 38 EAI는 기업 내 각종 애플리케이션 및 플랫폼 간의 정보 전달, 연계, 통합 등 상호 연동이 가능하게 해주는 솔루션으로, 구축 유형에는 Point-to-Point, Hub & Spoke, Message Bus(ESB), Hybrid가 있습니다.

[병행학습]

EAI(Enterprise Application Integration)

- 기업 내 각종 애플리케이션 및 플랫폼 간의 정보 전달, 연계, 통합 등 상호 연동이 가능하게 해주는 솔루션이다.
- EAI는 비즈니스 간 통합 및 연계성을 증대시켜 효율성 및 각 시스템 간의 확정성(Determinacy)을 높여 준다.
- EAI의 구축 유형
 - Point-to-Point : 가장 기본적인 애플리케이션 통합 방식으로, 애플리케이션을 1 : 1로 연결하며 변경 및 재사용이 어렵다.
 - Hub & Spoke : 단일 집중인 허브 시스템을 통해 데이터를 전송하는 중앙 집중형 방식으로, 확장 및 유지 보수가 용이하지만 허브 장애 발생 시 시스템 전체에 영향을 미친다.
 - Message Bus(ESB 방식) : 애플리케이션 사이에 미들웨어를 두어 처리하는 방식으로, 확장성이 뛰어나며 대용량 처리가 가능하다.
 - Hybrid : Hub & Spoke와 Message Bus의 혼합 방식으

로, 그룹 내에서는 Hub & Spoke 방식을, 그룹 간에는 Message Bus 방식을 사용한다.

- 39 소스 코드 품질 분석 도구는 정적 분석 도구와 동적 분석 도구로 나뉘며, 정적 분석 도구에는 pmd, cppcheck, SonarQube, checkstyle, ccm, cobertura 등이 있고, 동적 분석 도구에는 Avalanche, Valgrind 등이 있다.

[병행학습]

소스 코드 품질 분석 도구

- 소스 코드의 코딩 스타일, 코드에 설정된 코딩 표준, 코드의 복잡도, 코드에 존재하는 메모리 누수 현상, 스레드 결함 등을 발견하기 위해 사용하는 분석 도구로, 크게 정적 분석 도구와 동적 분석 도구로 나뉜다.
- 정적 분석 도구
 - 작성한 소스 코드를 실행하지 않고 코딩 표준이나 코딩 스타일, 결함 등을 확인하는 코드 분석 도구이다.
 - 비교적 애플리케이션 개발 초기의 결함을 찾는 데 사용되고, 개발 완료 시점에서는 개발된 소스 코드의 품질을 검증하는 차원에서 사용된다.
 - 동적 분석 도구로는 발견하기 어려운 결함을 찾아내고, 소스 코드에서 코딩의 복잡도, 모델 의존성, 불일치성 등을 분석할 수 있다.
 - 종류 : pmd, cppcheck, SonarQube, checkstyle, ccm, cobertura 등
- 동적 분석 도구
 - 작성한 소스 코드를 실행하여 코드에 존재하는 메모리 누수, 스레드 결함 등을 분석하는 도구이다.
 - 종류 : Avalanche, Valgrind 등

- 40 반정규화에서 중복 테이블을 추가하는 방법에는 집계 테이블의 추가, 진행 테이블의 추가, 특정 부분만을 포함하는 테이블의 추가가 있습니다.

[병행학습]

반정규화 - 중복 테이블 추가

- 여러 테이블에서 데이터를 추출해서 사용해야 하거나 다른 서버에 저장된 테이블을 이용해야 하는 경우 중복 테이블을 추가하여 작업의 효율성을 향상시킬 수 있다.
- 중복 테이블을 추가하는 경우
 - 정규화로 인해 수행 속도가 느려지는 경우
 - 많은 범위의 데이터를 자주 처리해야 하는 경우
 - 특정 범위의 데이터만 자주 처리해야 하는 경우
 - 처리 범위를 줄이지 않고는 수행 속도를 개선할 수 없는 경우
- 중복 테이블의 추가 방법
 - 집계 테이블의 추가 : 집계 데이터를 위한 테이블을 생성하고, 각 원본 테이블에 트리거(Trigger)를 설정하여 사용하는 것으로, 트리거의 오버헤드(Overhead)에 유의해야 한다.
 - 진행 테이블의 추가 : 이력 관리 등의 목적으로 추가하는 테이블로, 적절한 데이터 양의 유지와 활용도를 높이기 위해 기본키를 적절히 설정한다.
 - 특정 부분만을 포함하는 테이블의 추가 : 데이터가 많은 테이블의 특정 부분만을 사용하는 경우 해당 부분만으로 새로운 테이블을 생성한다.

- 41 DDL(데이터 정의어)의 3가지 명령어는 CREATE, ALTER, DROP입니다. UPDATE는 DML(데이터 조작어)의 명령어입니다.

[병행학습]

- DML(데이터 조작어) : SELECT, INSERT, UPDATE,

DELETE

- DCL(데이터 제어어) : COMMIT, ROLLBACK, GRANT, REVOKE

- 42 ㉠ STUDENT 테이블에서 DEPT를 검색합니다. 총 130개의 튜플이 들어 있고 검색 조건이 없으므로 튜플의 수는 130입니다.

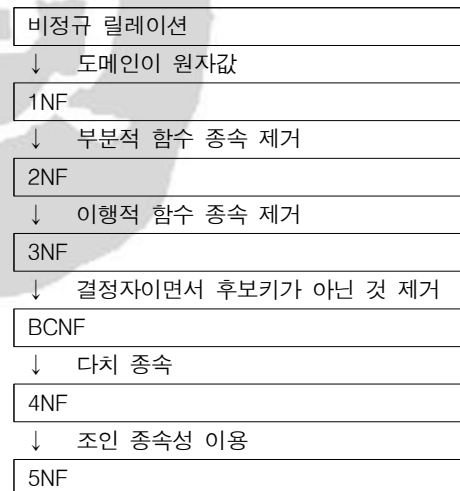
㉢ STUDENT 테이블에서 DEPT를 검색하는 데 중복된 결과는 처음의 한 개만 검색에 포함시킵니다. 독일어과 50개 튜플의 DEPT 속성의 값이 같으므로 1개, 중국어과 30개 튜플의 DEPT 속성의 값이 같으므로 1개, 영어영문학과 50개 튜플의 DEPT 속성의 값이 같으므로 1개를 검색에 포함시킴으로 3개의 튜플이 검색됩니다.

- 43 두 릴레이션에 공통으로 존재하는 속성명은 '과목번호'입니다. <과목> 릴레이션의 '과목번호'는 기본키 속성으로 동일한 속성값이 존재할 수 없고, <수강> 릴레이션의 '과목번호'는 일반 속성으로 여러 속성값이 존재할 수 있으므로 <수강> 릴레이션의 '과목번호' 속성이 <과목> 릴레이션의 기본키 속성 '과목번호'를 참조하는 외래키가 됩니다.

- 44 제2정규형(2NF)이 되기 위해서는 제1정규형(1NF)을 만족하고, 기본키가 아닌 모든 속성이 기본키에 대하여 완전 함수적 종속을 만족해야 합니다. ①번은 제1정규형, ②번은 제3정규형, ③번은 제4정규형이 되기 위한 조건입니다.

[병행학습]

정규화 과정



- 45 개체 무결성 제약 조건은 기본 테이블의 기본키를 구성하는 어떤 속성도 Null 값이나 중복값을 가질 수 없다는 규정입니다. ①번은 도메인 무결성, ③, ④번은 참조 무결성에 대한 설명입니다.

[병행학습]

무결성 제약 조건

- 데이터베이스에 저장된 데이터 값과 그것이 표현하는 현실 세계의 실제값이 일치하는 정확성을 의미한다.
- 무결성 제약 조건은 데이터베이스에 들어 있는 데이터의 정확성을 보장하기 위해 부정확한 자료가 데이터베이스 내에 저장되는 것을 방지하기 위한 제약 조건을 말한다.
- 무결성의 종류에는 개체 무결성, 도메인 무결성, 참조 무결성, 사용자 정의 무결성 등이 있다.
- 개체 무결성(Entity Integrity, 실체 무결성)
 - 개체 무결성은 기본 테이블의 기본키를 구성하는 어떤 속성도 Null 값이나 중복값을 가질 수 없다는 규정이다.

- 도메인 무결성(Domain Integrity, 영역 무결성)
 - 도메인 무결성은 주어진 속성 값이 정의된 도메인에 속한 값이어야 한다는 규정이다.
- 참조 무결성(Referential Integrity)
 - 참조 무결성은 외래키 값은 Null이거나 참조 릴레이션의 기본키 값과 동일해야 한다.
 - 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다는 규정이다.
- 사용자 정의 무결성
 - 사용자 정의 무결성(User-Defined Integrity)은 속성 값들이 사용자가 정의한 제약조건에 만족해야 한다는 규정이다.

46 이행적 함수 종속 DMS $A \rightarrow B$ 이고 $B \rightarrow C$ 일 때, $A \rightarrow C$ 를 만족하는 관계를 의미합니다.

47 DML(데이터 조작어)의 4가지 명령어는 SELECT, INSERT, DELETE, UPDATE입니다.

[병행학습]

- DDL(데이터 정의어) : CREATE, ALTER, DROP
- DCL(데이터 제어어) : COMMIT, ROLLBACK, GRANT, REVOKE

48 문제에 제시된 내용은 트리거(Trigger)에 대한 설명입니다.

[병행학습]

트리거(Trigger)

- 데이터베이스 시스템에서 데이터의 삽입(Insert), 갱신(Update), 삭제(Delete) 등의 이벤트(Event)가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL이다.
- 트리거는 데이터베이스에 저장되며, 데이터 변경 및 무결성 유지, 로그 메시지 출력 등의 목적으로 사용된다.
- 트리거의 구문에는 DCL(데이터 제어어)을 사용할 수 없으며, DCL이 포함된 프로시저나 함수를 호출하는 경우에도 오류가 발생한다.
- 트리거에 오류가 있는 경우 트리거가 처리하는 데이터에도 영향을 미치므로 트리거를 생성할 때 세심한 주의가 필요하다.

49 레코드 집중의 분석은 물리적 설계 단계에서 수행하는 작업입니다.

[병행학습]

논리적 설계(데이터 모델링)

- 논리적 설계 단계란 현실 세계에서 발생하는 자료를 컴퓨터가 이해하고 처리할 수 있는 물리적 저장장치에 저장할 수 있도록 변환하기 위해 특정 DBMS가 지원하는 논리적 자료 구조로 변환시키는 과정이다.
- 개념 세계의 데이터를 필드로 기술된 데이터 타입과 이 데이터 타입들 간의 관계로 표현되는 논리적 구조의 데이터로 모델화한다.
- 개념적 설계가 개념 스키마를 설계하는 단계라면 논리적 설계에서는 개념 스키마를 평가 및 정제하고 DBMS에 따라 서로 다른 논리적 스키마를 설계하는 단계이다.
- 트랜잭션의 인터페이스를 설계한다.
- 관계형 데이터베이스라면 테이블을 설계하는 단계이다.

물리적 설계(데이터 구조화)

- 물리적 설계란 논리적 설계 단계에서 논리적 구조로 표현된 데이터를 디스크 등의 물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 과정이다.
- 물리적 설계 단계에서는 다양한 데이터베이스 응용에 대해 처리 성능을 얻기 위해 데이터베이스 파일의 저장 구조

및 액세스 경로를 결정한다.

- 저장 레코드의 형식, 순서, 접근 경로와 같은 정보를 사용하여 데이터가 컴퓨터에 저장되는 방법을 묘사한다.

50 E-R 모델에서 속성은 타원으로 표현합니다.

[병행학습]

E-R 도형

사각형	개체(Entity) 타입
다이아몬드(마름모)	관계(Relationship) 타입
타원	속성(Attribute)
이중 타원	다중값 속성(복합 속성)
밑줄 타원	기본 키 속성
복수 타원	복합 속성
선, 링크	개체 타입과 속성을 연결

51 로킹 단위가 작아지면 로크 수가 많아 관리하기 복잡해 오버헤드가 증가합니다.

[병행학습]

로킹 단위(Locking Granularity)

- 로킹 단위는 병행제어에서 한꺼번에 로킹할 수 있는 객체의 크기를 의미한다.
- 데이터베이스, 파일, 레코드, 필드 등은 로킹 단위가 될 수 있다.
- 로킹 단위가 크면 로크 수가 작아 관리하기 쉽지만 병행성 수준이 낮아지고 데이터베이스 공유도가 저하된다.
- 로킹 단위가 작으면 로크 수가 많아 관리하기 복잡해 오버헤드가 증가하지만 병행성 수준이 높아지고, 데이터베이스 공유도가 증가한다.

52 뷰(View)는 저장장치 내에 물리적으로 존재하지 않는 가상 테이블입니다.

[병행학습]

뷰(View)의 개념

- 뷰는 사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블이다.
- 뷰는 저장장치 내에 물리적으로 존재하지 않지만, 사용자에게는 있는 것처럼 간주된다.

뷰(View)의 특징

- 뷰는 기본 테이블로부터 유도된 테이블이기 때문에 기본 테이블과 같은 형태의 구조를 사용하며, 조작도 기본 테이블과 거의 같다.
- 뷰는 가상 테이블이기 때문에 물리적으로 구현되어 있지 않다.
- 데이터의 논리적 독립성이 어느 정도 보장된다.
- 필요한 데이터만 뷰로 정의해서 처리할 수 있기 때문에 관리가 용이하고 명령문이 간단해진다.
- 뷰를 통해서만 데이터에 접근하게 하면 뷰에 나타나지 않는 데이터를 안전하게 보호하는 효율적인 기법으로 사용할 수 있다.
- 기본 테이블의 기본키를 포함한 속성(열) 집합으로 뷰를 구성해야만 삽입, 삭제, 갱신 연산이 가능하다.
- 정의된 뷰는 다른 뷰의 정의에 기초가 될 수 있다.
- 하나의 뷰를 삭제하면 그 뷰를 기초로 정의된 다른 뷰도 자동으로 삭제된다.

53 하나의 애트리뷰트가 가질 수 있는 원자값들의 집합을 의미하는 것은 도메인(Domain)입니다.

[병행학습]

관계형 데이터베이스 관련 용어

• 튜플(Tupel)

- 튜플은 릴레이션을 구성하는 각각의 행을 말한다.
- 튜플은 속성의 모임으로 구성된다.
- 파일 구조에서 레코드와 같은 의미이다.
- 튜플의 수를 카디널리티(Cardinality) 또는 기수, 대응수라고 한다.

• 속성(Attribute)

- 속성은 데이터베이스를 구성하는 가장 작은 논리적 단위이다.
- 파일 구조상의 데이터 항목 또는 데이터 필드에 해당된다.
- 속성은 개체의 특성을 기술한다.
- 속성의 수를 디그리(Degree) 또는 차수라고 한다.

• 도메인(Domain)

- 도메인은 하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자(Atomic)값들의 집합이다.
- 도메인은 실제 애트리뷰트 값이 나타날 때 그 값의 합법 여부를 시스템이 검사하는데에도 이용된다.

[예] 성별 애트리뷰트의 도메인은 '남'과 '여'로, 그 외의 값은 입력될 수 없다.

- 54 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 Join 연산자의 기호는 입니다.

[병행학습]

순수 관계 연산자

• Select

- 릴레이션에 존재하는 튜플 중에서 선택 조건을 만족하는 튜플의 부분집합을 구하여 새로운 릴레이션을 만드는 연산이다.
- 릴레이션의 행(가로)에 해당하는 튜플을 구하는 것이므로 수평 연산이라고도 한다.
- 연산자의 기호는 시그마(σ)를 사용한다.

• Project

- 주어진 릴레이션에서 속성 리스트(Attribute List)에 제시된 속성 값만을 추출하여 새로운 릴레이션을 만드는 연산이다. 단 연산 결과에 중복이 발생하면 중복이 제거된다.
- 릴레이션의 열(세로)에 해당하는 Attribute를 추출하는 것이므로 수직 연산자라고도 한다.
- 연산자의 기호는 파이(π)를 사용한다.

• Join

- 공통 속성을 중심으로 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 연산이다.
- 연산자의 기호는 \bowtie 를 사용한다.

• Division

- $X \supset Y$ 인 두 개의 릴레이션 $R(X)$ 과 $S(Y)$ 가 있을 때, R 의 속성이 S 의 속성값을 모두 가진 튜플에서 S 가 가진 속성을 제외한 속성만을 구하는 연산이다.
- 연산자의 기호는 \div 를 사용한다.

- 55 문제에 제시된 내용과 관계있는 분산 데이터베이스 목표는 장애 투명성(Failure Transparency)입니다.

[병행학습]

분산 데이터베이스(Distributed Database)

- 분산 데이터베이스는 논리적으로는 같은 시스템에 속하지만 물리적으로는 컴퓨터 네트워크를 통해 분산되어 있는 데이터베이스로 목표는 다음과 같다.
- 위치 투명성(Location Transparency) : 접근하려는 데이

터베이스의 실제 위치를 알 필요 없이 단지 데이터베이스의 논리적인 명칭만으로 접근할 수 있음

- **중복 투명성(Replication Transparency)** : 동일한 데이터가 여러 곳에 중복되어 있더라도 사용자는 마치 하나의 데이터만 존재하는 것처럼 사용할 수 있고, 시스템은 자동으로 여러 데이터에 대한 작업을 수행함
- **병행 투명성(Concurrency Transparency)** : 분산 데이터베이스와 관련된 다수의 트랜잭션들이 동시에 실행되더라도 그 트랜잭션들의 수행 결과는 서로 영향을 받지 않음
- **장애 투명성(Failure Transparency)** : 트랜잭션, DBMS, 네트워크, 컴퓨터 장애에도 불구하고 트랜잭션은 정확하게 수행됨

- 56 후보키는 릴레이션에 있는 모든 튜플에 대해 유일성과 최소성을 모두 만족시켜야 합니다.

[병행학습]

키(Key)의 개념 및 종류

- 키(Key)는 데이터베이스에서 조건에 만족하는 튜플을 찾거나 순서대로 정렬할 때 기준이 되는 속성이다.
- **슈퍼키(Super Key)** : 한 릴레이션 내에 있는 속성들의 집합으로 구성된 키로, 릴레이션을 구성하는 모든 튜플에 대해 유일성(Unique)은 만족하지만, 최소성(Minimality)은 만족하지 못함
- **후보키(Candidate Key)** : 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용되는 속성들의 부분집합으로, 유일성과 최소성을 모두 만족함
- **기본키(Primary Key)** : 후보키 중에서 특별히 선정된 키로 중복된 값과 NULL 값을 가질 수 없음
- **대체키(Alternate Key)** : 후보키 중에서 선정된 기본키를 제외한 나머지 후보키를 의미함
- **외래키(Foreign Key)** : 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미하며, 릴레이션 간의 관계를 표현할 때 사용함

- 57 문제의 질의문은 하위 질의가 있는 질의문입니다. 먼저 WHERE 조건에 지정된 하위 질의의 SELECT문을 검색합니다. 그리고 검색 결과를 본 질의의 조건에 있는 '책번호' 속성과 비교합니다.

① **SELECT 책번호 FROM 도서 WHERE 책명 = '자료구조';**
: '도서' 테이블에서 '책명' 속성의 값이 '자료구조'와 같은 레코드의 '책번호' 속성의 값을 검색합니다. 결과는 '222'입니다.

② **SELECT 가격 FROM 도서가격 WHERE 책번호 = '222';**
: '도서가격' 테이블에서 '책번호' 속성의 값이 '222'와 같은 레코드의 '가격' 속성의 값을 검색합니다. 결과는 '25,000'입니다.

- 58 논리적, 물리적 데이터 구조를 정의하는 것은 데이터 정의 언어(DDL; Data Definition Language)의 기능입니다.

[병행학습]

DDL(Data Define Language, 데이터 정의어)

- SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의하거나 변경 또는 삭제할 때 사용하는 언어이다.
- 논리적 데이터 구조와 물리적 데이터 구조의 사상을 정의한다.
- 데이터베이스 관리자나 데이터베이스 설계자가 사용한다.
- DDL(데이터 정의어)의 세 가지 유형
 - CREATE : SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의한다.
 - ALTER : TABLE에 대한 정의를 변경하는 데 사용한다.

- DROP : SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 삭제한다.

DML(Data Manipulation Language, 데이터 조작어)

- DML은 데이터베이스 사용자가 응용 프로그램이나 질의어를 통하여 저장된 데이터를 실질적으로 처리하는 데 사용되는 언어이다.
- 데이터베이스 사용자와 데이터베이스 관리 시스템 간의 인터페이스를 제공한다.
- DML(데이터 조작어)의 네 가지 유형
 - SELECT : 테이블에서 조건에 맞는 튜플을 검색한다.
 - INSERT : 테이블에 새로운 튜플을 삽입한다.
 - DELETE : 테이블에서 조건에 맞는 튜플을 삭제한다.
 - UPDATE : 테이블에서 조건에 맞는 튜플의 내용을 변경한다.

DCL(Data Control Language, 데이터 제어어)

- DCL은 데이터의 보안, 무결성, 회복, 병행 수행 제어 등을 정의하는 데 사용되는 언어이다.
- 데이터베이스 관리자가 데이터 관리를 목적으로 사용한다.
- DCL(데이터 제어어)의 종류
 - COMMIT : 명령에 의해 수행된 결과를 실제 물리적 디스크로 저장하고, 데이터베이스 조작 작업이 정상적으로 완료되었음을 관리자에게 알려준다.
 - ROLLBACK : 데이터베이스 조작 작업이 비정상적으로 종료되었을 때 원래의 상태로 복구한다.
 - GRANT : 데이터베이스 사용자에게 사용 권한을 부여한다.
 - REVOKE : 데이터베이스 사용자의 사용 권한을 취소한다.

- 59 문제에 제시된 내용은 CASCADE에 대한 설명입니다.

[병행학습]

- CASCADE : 제거할 요소를 참조하는 다른 모든 개체를 함께 제거함. 즉 주 테이블의 데이터 제거 시 각 외래키와 관계를 맺고 있는 모든 데이터를 제거하는 참조 무결성 제약 조건을 설정하기 위해 사용됨
- RESTRICTED : 다른 개체가 제거할 요소를 참조중일 때는 제거를 취소함

- 60 문제의 지문에 제시된 트랜잭션의 특성은 원자성(Atomicity)입니다.

[병행학습]

트랜잭션의 특성

- Atomicity(원자성)
 - 트랜잭션의 연산은 데이터베이스에 모두 반영되도록 완료(Commit)되든지 아니면 전혀 반영되지 않도록 복구(Rollback)되어야 한다.
 - 트랜잭션 내의 모든 명령은 반드시 완벽히 수행되어야 하며, 모두가 완벽히 수행되지 않고 어느 하나라도 오류가 발생하면 트랜잭션 전부가 취소되어야 한다.
- Consistency(일관성)
 - 트랜잭션이 그 실행을 성공적으로 완료하면 언제나 일관성 있는 데이터베이스 상태로 변환한다.
 - 시스템이 가지고 있는 고정 요소는 트랜잭션 수행 전과 트랜잭션 수행 완료 후의 상태가 같아야 한다.
- Isolation(독립성, 격리성, 순차성)
 - 둘 이상의 트랜잭션이 동시에 병행 실행되는 경우 어느 하나의 트랜잭션 실행중에 다른 트랜잭션의 연산이 끼어들 수 없다.
 - 수행중인 트랜잭션은 완전히 완료될 때까지 다른 트랜잭션에서 수행 결과를 참조할 수 없다.
- Durability(영속성, 지속성) : 성공적으로 완료된 트랜잭션

의 결과는 시스템이 고장나더라도 영구적으로 반영되어야 함

- 61 프로세스 관리, 기억장치 관리, 파일 관리 등의 작업을 수행하는 것은 커널(Kernel)입니다.

[병행학습]

UNIX 시스템의 구성

커널(Kernel)	<ul style="list-style-type: none"> • UNIX의 가장 핵심적인 부분 • 하드웨어를 보호(캡슐화)하고, 프로그램들과 하드웨어 간의 인터페이스 역할을 담당함 • 프로세스, 기억장치, 파일, 입·출력 관리, 프로세스 간 통신, 데이터 전송 및 변환 등 여러 가지 기능을 수행함 • 컴퓨터 부팅 시 주기억장치에 적재되어 상주하면서 실행됨
셸(Shell)	<ul style="list-style-type: none"> • 사용자의 명령어를 인식하여 프로그램을 호출하고, 명령을 수행하는 명령어 해석기 • 시스템과 사용자 간의 인터페이스를 담당함 • DOS의 COMMAND.COM과 같은 기능을 수행함 • 주기억장치에 상주하지 않고, 명령어가 포함된 파일 형태로 존재하며 보조기억장치에서 교체 처리가 가능함 • 종류 : Bourne Shell, C Shell, Korn Shell 등

- 62 TCP는 전송 계층, 나머지는 응용 계층의 프로토콜입니다.

[병행학습]

TCP/IP의 구조

OSI	TCP/IP	기능
응용 계층 표현 계층 세션 계층	응용 계층	<ul style="list-style-type: none"> • 응용 프로그램 간의 데이터 송·수신을 제공 • TELNET, FTP, SMTP, SNMP, DNS, HTTP 등
전송 계층	전송 계층	<ul style="list-style-type: none"> • 호스트들 간의 신뢰성 있는 통신을 제공 • TCP, UDP
네트워크 계층	인터넷 계층	<ul style="list-style-type: none"> • 데이터 전송을 위한 주소 지정, 경로 설정을 제공 • IP, ICMP, IGMP, ARP, RARP
데이터 링크 계층 물리 계층	네트워크 액세스 계층	<ul style="list-style-type: none"> • 실제 데이터(프레임)를 송·수신하는 역할 • Ethernet, IEEE 802, HDLC, X.25, RS-232C, ARQ 등

- 63 C 언어의 비트 연산자에는 &, ^, |, ~, <<, >> 등이 있습니다.

[병행학습]

비트 연산자

비트별(0, 1)로 연산하여 결과를 얻는 연산자이다.

연산자	의미	비고
&	and	모든 비트가 1일 때만 1
^	xor	모든 비트가 같으면 0, 하나라도 다르면 1
	or	모든 비트 중 한 비트라도 1이면 1
~	not	각 비트의 부정, 0이면 1, 1이면 0
<<	왼쪽 시프트	비트를 왼쪽으로 이동
>>	오른쪽 시프트	비트를 오른쪽으로 이동

- 64 ① 결합도는 모듈 상호 간에 의존하는 정도를 의미합니다. 한 모듈 내에 있는 처리 요소들 사이의 기능적인 연관 정도를 나타내는 것은 응집도입니다.
- ② 결합도가 높으면 시스템 구현 및 유지보수 작업이 어렵습니다.
- ④ 자료 결합도는 내용 결합도보다 결합도가 낮습니다.

[병행학습]

결합도(Coupling)

- 모듈 간에 상호 의존하는 정도 또는 두 모듈 사이의 연관 관계를 의미한다.
- 다양한 결합으로 모듈을 구성할 수 있으나 결합도가 약할 수록 품질이 높고, 강할수록 품질이 낮다.
- 결합도가 강하면 시스템 구현 및 유지보수 작업이 어렵다.
- 결합도의 종류(낮음에서 높음순)
 - 자료 결합도(Data Coupling)** : 모듈 간의 인터페이스가 자료 요소로만 구성될 때의 결합도
 - 스탬프(검인) 결합도(Stamp Coupling)** : 모듈 간의 인터페이스로 배열이나 레코드 등의 자료 구조가 전달될 때의 결합도
 - 제어 결합도(Control Coupling)** : 어떤 모듈이 다른 모듈 내부의 논리적인 흐름을 제어하기 위해 제어 신호를 이용하여 통신하거나 제어 요소를 전달하는 결합도
 - 외부 결합도(External Coupling)** : 어떤 모듈에서 외부로 선언한 데이터(변수)를 다른 모듈에서 참조할 때의 결합도
 - 공통(공유) 결합도(Common Coupling)** : 공유되는 공통 데이터 영역을 여러 모듈이 사용할 때의 결합도
 - 내용 결합도(Content Coupling)** : 한 모듈이 다른 모듈의 내부 기능 및 그 내부 자료를 직접 참조하거나 수정할 때의 결합도

- 65 한 개의 프로세스는 여러 개의 스레드를 가질 수 있습니다.

[병행학습]

스레드(Thread)

- 프로세스 내에서의 작업 단위로서 시스템의 여러 자원을 할당받아 실행하는 프로그램의 단위이다.
- 하나의 프로세스에 하나의 스레드가 존재하는 경우에는 단일 스레드, 하나 이상의 스레드가 존재하는 경우에는 다중 스레드라고 한다.
- 프로세스의 일부 특성을 갖고 있기 때문에 경량(Light Weight) 프로세스라고도 한다.
- 스레드 기반 시스템에서 스레드는 독립적인 스케줄링의 최소 단위로서 프로세스의 역할을 담당한다.
- 동일 프로세스 환경에서 서로 독립적인 다중 수행이 가능하다.

스레드의 분류

- 사용자 수준의 스레드
 - 사용자가 만든 라이브러리를 사용하여 스레드를 운용한다.
 - 속도는 빠르지만 구현이 어렵다.
- 커널 수준의 스레드
 - 운영체제의 커널에 의해 스레드를 운용한다.
 - 구현이 쉽지만 속도가 느리다.

- 66 C언어의 변수 선언 시 앞에 static을 붙이면 정적 변수로 선언됩니다. 정적 변수는 초기화를 생략하면 자동으로 0으로 초기화 됩니다. 또한 정적 변수가 아니라 static을 생략하여 자동 변수로 선언하더라도 배열의 초기 값을 일부만 지정하면 배열의 첫 번째 요소부터 지정한 값이 입력되고, 나머지

요소는 0으로 초기화 됩니다.

[병행학습]

지역 클래스의 종류

- 자동 변수(Automatic Variable)
 - 함수나 코드의 범위를 한정하는 블록 내에서 선언되는 변수이다.
 - 함수나 블록이 실행되는 동안에만 존재하며 이를 벗어나면 자동으로 소멸된다.
 - 초기화하지 않으면 쓰레기값(Garbage Value)이 저장된다.
- 외부 변수(External Variable)
 - 현재 파일이나 다른 파일에서 선언된 변수나 함수를 참조(reference)하기 위한 변수이다.
 - 외부 변수는 함수 밖에서 선언한다.
 - 함수가 종료된 뒤에도 값이 소멸되지 않는다.
 - 초기화하지 않으면 자동으로 0으로 초기화 된다.
 - 다른 파일에서 선언된 변수를 참조할 경우 초기화 할 수 없다.
- 정적 변수(Static Variable)
 - 함수나 블록 내에서 선언하는 내부 정적 변수와 함수 외부에서 선언하는 외부 정적 변수가 있다.
 - 내부 정적 변수는 선언한 함수나 블록 내에서만 사용할 수 있고, 외부 정적 변수는 모든 함수에서 사용할 수 있다.
 - 두 변수 모두 함수나 블록이 종료된 뒤에도 값이 소멸되지 않는다.
 - 초기화는 변수 선언 시 한 번만 할 수 있으며, 초기화를 생략하면 자동으로 0으로 초기화 된다.
- 레지스터 변수(Register Variable)
 - 메모리가 아닌 CPU 내부의 레지스터에 기억영역을 할당받는 변수이다.
 - 자주 사용되는 변수를 레지스터에 저장하여 처리 속도를 높이기 위해 사용한다.
 - 함수나 블록이 실행되는 동안에만 존재하며 이를 벗어나면 자동으로 소멸된다.
 - 레지스터의 사용 개수는 한정되어 있어 데이터를 저장할 레지스터가 없는 경우 자동 변수로 취급되어 메모리에 할당된다.
 - CPU에 저장되어 메모리 주소를 가질 수 없기 때문에 변수의 주소를 구하는 주소 연산자(&)를 사용할 수 없다.

- 67 은행가 알고리즘은 교착상태의 해결 방법 중 회피 기법(Avoidance)에 해당합니다.

[병행학습]

교착상태 해결 방법

예방 기법(Prevention)

- 교착상태가 발생하지 않도록 사전에 시스템을 제어하는 방법으로, 교착상태 발생의 네 가지 조건 중에서 어느 하나를 제거(부정)함으로써 수행된다.
- 상호 배제(Mutual Exclusion) 부정 : 한 번에 여러 개의 프로세스가 공유 자원을 사용할 수 있도록 함
- 점유 및 대기(Hold and Wait) 부정 : 프로세스가 실행되기 전 필요한 모든 자원을 할당하여 프로세스 대기를 없애거나 자원이 점유되지 않은 상태에서에서만 자원을 요구하도록 함
- 비선점(Non-preemption) 부정 : 자원을 점유하고 있는 프로세스가 다른 자원을 요구할 때 점유하고 있는 자원을 반납하고, 요구한 자원을 사용하기 위해 기다리게 함
- 환형 대기(Circular Wait) 부정 : 자원을 선형 순서로

1, 2회

분류하여 고유 번호를 할당하고, 각 프로세스는 현재 점유한 자원의 고유 번호보다 앞이나 뒤 어느 한쪽 방향으로만 자원을 요구하도록 하는 것

- **회피 기법(Avoidance)** : 교착상태가 발생할 가능성을 배제하지 않고 교착상태가 발생하면 적절히 피해나가는 방법으로, 주로 은행원 알고리즘(Banker's Algorithm)이 사용됨
- **발견(Detection) 기법** : 시스템에 교착 상태가 발생했는지 점검하여 교착 상태에 있는 프로세스와 자원을 발견하는 것으로, 자원 할당 그래프 등을 사용함
- **회복(Recovery) 기법** : 교착 상태를 일으킨 프로세스를 종료하거나 교착 상태의 프로세스에 할당된 자원을 선점하여 프로세스나 자원을 회복하는 것

- 68 QoS 강화를 위해 MAC 지원 기능을 채택한 802.11 버전은 802.11e입니다.

[병행학습]

802.11의 버전

- **802.11(초기 버전)** : 2.4GHz 대역 전파와 CSMA/CA 기술을 사용해 최고 2Mbps까지의 전송 속도를 지원함
- **802.11a** : 5GHz 대역의 전파를 사용하며, OFDM 기술을 사용해 최고 54Mbps까지의 전송 속도를 지원함
- **802.11b** : 802.11 초기 버전의 개선안으로 등장하였으며, 초기 버전의 대역 전파와 기술을 사용해 최고 11Mbps의 전송 속도로 기존에 비해 5배 이상 빠르게 개선되었음
- **802.11e** : 802.11의 부가 기능 표준으로, QoS 기능이 지원되도록 하기 위해 매체 접근 제어(MAC) 계층에 해당하는 부분을 수정하였음
- **802.11g** : 2.4GHz 대역의 전파를 사용하지만 5GHz 대역의 전파를 사용하는 802.11a와 동일한 최고 54Mbps까지의 전송 속도를 지원함
- **802.11n** : 2.4GHz 대역과 5GHz 대역을 사용하는 규격으로, 최고 600Mbps까지의 전송 속도를 지원함

- 69 TCP/IP 네트워크에서 IP 주소를 MAC 주소로 변환하는 프로토콜은 ARP(Address Resolution Protocol)입니다.

[병행학습]

- **UDP(User Datagram Protocol)** : 데이터 전송 전에 연결을 설정하지 않는 비연결형 서비스를 제공하고, TCP에 비해 상대적으로 단순한 헤더 구조를 가지므로, 오버헤드가 적음
- **TCP(Transmission Control Protocol)** : 신뢰성 있는 연결형 서비스를 제공하고, 패킷의 다중화, 순서 제어, 오류 제어, 흐름 제어 기능을 제공함
- **ICMP(Internet Control Message Protocol, 인터넷 제어 메시지 프로토콜)** : IP와 조합하여 통신중에 발생하는 오류의 처리와 전송 경로 변경 등을 위한 제어 메시지를 관리하는 역할을 하며, 헤더는 8Byte로 구성됨

- 70 HRN 스케줄링 방식은 우선순위를 계산하여 그 숫자가 가장 높은 것부터 낮은 순으로 우선순위가 부여됩니다.

[병행학습]

HRN(Highest Response-ratio Next)

- 실행 시간이 긴 프로세스에 불리한 SJF 기법을 보완하기 위한 것으로, 대기 시간과 서비스(실행) 시간을 이용하는 기법이다.
- 우선순위 계산 공식을 이용하여 서비스(실행) 시간이 짧은 프로세스나 대기 시간이 긴 프로세스에게 우선순위를 주어 CPU를 할당한다.
- 서비스 실행 시간이 짧거나 대기 시간이 긴 프로세스일

경우 우선순위가 높아진다.

- 우선순위를 계산하여 그 숫자가 가장 높은 것부터 낮은 순으로 우선순위가 부여된다.
- **우선순위 계산식** : (대기 시간+ 서비스 시간) / 서비스 시간

- 71 교착 상태 발생의 필요 충분 조건 4가지는 상호 배제, 점유와 대기, 환경 대기, 비전점입니다.

[병행학습]

교착상태 발생의 필요 충분 조건

- **상호 배제(Mutual Exclusion)** : 한 번에 한 개의 프로세스만이 공유 자원을 사용할 수 있어야 함
- **점유와 대기(Hold and Wait)** : 최소한 하나의 자원을 점유하고 있으면서 다른 프로세스에 할당되어 사용되고 있는 자원을 추가로 점유하기 위해 대기하는 프로세스가 있어야 함
- **비선점(Non-preemption)** : 다른 프로세스에 할당된 자원은 사용이 끝날 때까지 강제로 빼앗을 수 없어야 함
- **환경 대기(Circular Wait)** : 공유 자원과 공유 자원을 사용하기 위해 대기하는 프로세스들이 원형으로 구성되어 있어 자신에게 할당된 자원을 점유하면서 앞이나 뒤에 있는 프로세스의 자원을 요구해야 함

- 72 3개의 페이지를 수용할 수 있는 주기억장치이므로 다음 그림과 같이 3개의 페이지 프레임으로 표현할 수 있습니다.

참조 페이지	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0
페이지 프레임	7	7	7	2	2	2	2	4	4	4	0	0	0	0	0	0	0	7	7
		0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1	0
			1	1	1	1	0	0	0	3	3	3	3	3	2	2	2	2	2
부재 발생	●	●	●	●		●	●	●	●	●					●	●		●	●

참조 페이지가 페이지 프레임에 없을 경우 페이지 결함(부재)이 발생합니다. 초기에는 모든 페이지 프레임이 비어 있으므로 처음 7, 0, 1 페이지 적재 시 페이지 결함이 발생합니다. 선입선출(FIFO) 기법은 각 페이지가 주기억장치에 적재될 때마다 그때의 시간을 기억시켜 가장 먼저 들어와서 가장 오래 있었던 페이지를 교체하는 기법이므로, 참조 페이지 2를 참조할 때에는 7을 제거한 후 2를 가져오게 됩니다. 이와 같은 방식으로 모든 페이지 요청을 처리하고 나면 총 페이지 결함 발생 수는 14번입니다.

- 73 변수명에는 공백이나 *, +, -, / 등의 특수문자를 사용할 수 없습니다.

[병행학습]

변수명 작성 규칙

- 영문자, 숫자, _(under bar)를 사용할 수 있다.
- 첫 글자는 영문자나 _(under bar)로 시작해야 하며, 숫자는 올 수 없다.
- 글자 수에 제한이 없다.
- 공백이나 *, +, -, / 등의 특수문자를 사용할 수 없다.
- 대·소문자를 구분한다.
- 예약어를 변수명으로 사용할 수 없다.
- 변수 선언 시 문장 끝에 반드시 세미콜론(;)을 붙여야 한다.

- 74 IPv6의 패킷 크기는 제한이 없습니다. 패킷 크기가 64Kbyte로 고정되어 있는 것은 IPv4입니다.

[병행학습]

IPv6(Internet Protocol version 6)

- 현재 사용하고 있는 IP 주소 체계인 IPv4의 주소 부족 문제를 해결하기 위해 개발되었다.
- 16비트씩 8부분, 총 128비트로 구성되어 있다.

- 각 부분을 16진수로 표현하고, 콜론(:)으로 구분한다.
- IPv4에 비해 자료 전송 속도가 빠르고, IPv4와 호환성이 뛰어나다.
- 인증성, 기밀성, 데이터 무결성의 지원으로 보안 문제를 해결할 수 있다.
- IPv6의 주소 체계
 - 유니캐스트(Unicast) : 단일 송신자와 단일 수신자 간의 통신(1:1 통신에 사용)
 - 멀티캐스트(Multicast) : 단일 송신자와 다중 수신자 간의 통신(1:N 통신에 사용)
 - 애니캐스트(Anycast) : 단일 송신자와 가장 가까이 있는 단일 수신자 간의 통신(1:1 통신에 사용)

75 프로세스 상태의 종류에는 제출(Submit), 접수(Hold), 준비(Ready), 실행(Run), 대기(Wait), 종료(Terminated, Exit) 등이 있습니다.

[병행학습]

프로세스 상태 전이

- **제출(Submit)** : 작업을 처리하기 위해 사용자가 작업을 시스템에 제출한 상태
- **접수(Hold)** : 제출된 작업이 스펴 공간인 디스크의 할당 위치에 저장된 상태
- **준비(Ready)** : 프로세스가 프로세서를 할당받기 위해 기다리고 있는 상태
- **실행(Run)** : 준비상태 큐에 있는 프로세스가 프로세서를 할당받아 실행되는 상태
- **대기(Wait), 블록(Block)** : 프로세스에 입·출력 처리가 필요하면 현재 실행 중인 프로세스가 중단되고, 입·출력 처리가 완료될 때까지 대기하고 있는 상태
- **종료(Terminated, Exit)** : 프로세스의 실행이 끝나고 프로세스 할당이 해제된 상태

프로세스 상태 전이 관련 용어

- **Dispatch** : 준비 상태에서 대기하고 있는 프로세스 중 하나가 프로세서를 할당받아 실행 상태로 전이되는 과정
- **Wake Up** : 입·출력 작업이 완료되어 프로세스가 대기 상태에서 준비 상태로 전이 되는 과정
- **Spooling** : 입·출력장치의 공유 및 상대적으로 느린 입·출력장치의 처리 속도를 보완하고 다중 프로그래밍 시스템의 성능을 향상시키기 위해 입·출력할 데이터를 직접 입·출력장치에 보내지 않고 나중에 한꺼번에 입·출력하기 위해 디스크에 저장하는 과정

76 IPv6의 주소 체계로는 유니캐스트(Unicast), 멀티캐스트(Multicast), 애니캐스트(Anycast)가 있습니다.

77 응집도를 강한 것에서 약한 것 순으로 나열하면 '기능적 응집도 → 순차적 응집도 → 교환(통신)적 응집도 → 절차적 응집도 → 시간적 응집도 → 논리적 응집도 → 우연적 응집도'입니다.

[병행학습]

응집도(Cohesion)

- 정보 은닉 개념을 확장한 것으로, 명령어나 호출문 등 모듈의 내부 요소들의 서로 관련되어 있는 정도, 즉 모듈이 독립적인 기능으로 정의되어 있는 정도를 의미한다.
- 다양한 기준으로 모듈을 구성할 수 있으나 응집도가 강할수록 품질이 높고, 약할수록 품질이 낮다.
- 응집도의 종류(강함에서 낮음순)
 - **기능적 응집도(Functional Cohesion)** : 모듈 내부의 모든 기능 요소들이 단일 문제와 연관되어 수행될 경우의 응집도

- **순차적 응집도(Sequential Cohesion)** : 모듈 내 하나의 활동으로부터 나온 출력 데이터를 그 다음 활동의 입력 데이터로 사용할 경우의 응집도
- **교환(통신)적 응집도(Communication Cohesion)** : 동일한 입력과 출력을 사용하여 서로 다른 기능을 수행하는 구성 요소들이 모였을 경우의 응집도
- **절차적 응집도(Procedural Cohesion)** : 모듈이 다수의 관련 기능을 가질 때 모듈 안의 구성 요소들이 그 기능을 순차적으로 수행할 경우의 응집도
- **시간적 응집도(Temporal Cohesion)** : 특정 시간에 처리되는 몇 개의 기능을 모아 하나의 모듈로 작성할 경우의 응집도
- **논리적 응집도(Logical Cohesion)** : 유사한 성격을 갖거나 특정 형태로 분류되는 처리 요소들로 하나의 모듈이 형성되는 경우의 응집도
- **우연적 응집도(Coincidental Cohesion)** : 모듈 내부의 각 구성 요소들이 서로 관련 없는 요소로만 구성된 경우의 응집도

78 JAVA의 접근 제한자에는 Public, Default, Private, protected 등이 있습니다.

79 Cobol은 절차적 프로그래밍 언어에 해당합니다.

[병행학습]

스크립트 언어(Script Language)

- HTML 문서 안에 직접 프로그래밍 언어를 삽입하여 사용하는 것으로, 기계어로 컴파일 되지 않고 별도의 번역기가 소스를 분석하여 동작하게 하는 언어이다.
- 게시판 입력, 상품 검색, 회원 가입 등과 같은 데이터베이스 처리 작업을 수행하기 위해 주로 사용한다.
- 스크립트 언어는 클라이언트의 웹 브라우저에서 해석되어 실행되는 클라이언트용 스크립트 언어와 서버에서 해석되어 실행된 후 결과만 클라이언트로 보내는 서버용 스크립트 언어가 있다.
 - 서버용 스크립트 언어 : ASP, JSP, PHP, 파이썬(Python), 펄(Perl), 루비(Ruby) 등
 - 클라이언트용 스크립트 언어 : 자바 스크립트(JavaScript), VB 스크립트(Visual Basic Script) 등

80 문제에 제시된 내용은 OSI 계층 중 전송 계층에 대한 설명입니다.

[병행학습]

OSI(Open System Interconnection) 참조 모델

- 다른 시스템 간의 원활한 통신을 위해 ISO(국제표준화기구)에서 제안한 통신 규약(Protocol)이다.
- **OSI 7계층** : 하위 계층(물리 계층 → 데이터 링크 계층 → 네트워크 계층), 상위 계층(전송 계층 → 세션 계층 → 표현 계층 → 응용 계층)
- **물리 계층(Physical Layer)** : 전송에 필요한 두 장치 간의 실제 접속과 절단 등 기계적, 전기적, 기능적, 절차적 특성에 대한 규칙을 정의함
- **데이터 링크 계층(Data Link Layer)**
 - 두 개의 인접한 개방 시스템들 간에 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 함
 - 흐름 제어, 프레임 동기화, 오류 제어, 순서 제어
- **네트워크 계층(Network Layer, 망 계층)**
 - 개방 시스템들 간의 네트워크 연결을 관리하는 기능과 데이터의 교환 및 중계 기능을 함
 - 경로 설정(Routing), 트래픽 제어, 패킷 정보 전송
- **전송 계층(Transport Layer)**

- 종단 시스템(End-to-End) 간의 전송 연결 설정, 데이터 전송, 연결 해제 기능을 함
- 주소 설정, 다중화(데이터의 분할과 재조립), 오류 제어, 흐름 제어

• 세션 계층(Session Layer)

- 송·수신 측 간의 관련성을 유지하고 대화 제어를 담당함
- 대화(회화) 구성 및 동기 제어, 데이터 교환 관리 기능

• 표현 계층(Presentation Layer)

- 응용 계층으로부터 받은 데이터를 세션 계층에 맞게, 세션 계층에서 받은 데이터는 응용 계층에 맞게 변환하는 기능
- 코드 변환, 데이터 암호화, 데이터 압축, 구문 검색, 정보 형식(포맷) 변환, 문맥 관리 기능

• 응용 계층(Application Layer) : 사용자(응용 프로그램)가 OSI 환경에 접근할 수 있도록 응용 프로세스 간의 정보 교환, 전자 사서함, 파일 전송, 가상 터미널 등의 서비스를 제공함

- 81 문제에 제시된 내용은 JSON(JavaScript Object Notation)에 대한 설명입니다.

[병행학습]

- 파이썬(Python) : 객체지향 기능을 지원하는 대화형 인터프리터 언어로, 플랫폼에 독립적이고 문법이 간단하여 배우기 쉬움
- XML(eXtensible Markup Language) : 특수한 목적을 갖는 마크업 언어를 만드는 데 사용되는 다목적 마크업 언어
- 웹 서버(Web Server) : 클라이언트로부터 직접 요청을 받아 처리하는 서버로, 저장량의 정적 파일들을 제공함

- 82 최대 홉수를 15로 제한한 라우팅 프로토콜은 RIP(Routing Information Protocol)입니다.

[병행학습]

RIP(Routing Information Protocol)

- 현재 가장 널리 사용되는 라우팅 프로토콜이다.
- 소규모 동종의 네트워크(자율 시스템, AS) 내에서 효율적인 방법이다.
- 최대 홉(Hop)수를 15로 제한하므로 15 이상의 경우는 도달할 수 없는 네트워크를 의미하는데 이것은 대규모 네트워크에서는 RIP를 사용할 수 없음을 의미한다.
- 라우팅 정보를 30초마다 네트워크 내의 모든 라우터에 알리며, 180초 이내에 새로운 라우팅 정보가 수신되지 않으면 해당 경로를 이상 상태로 간주한다.

OSPF(Open Shortest Path First protocol)

- 대규모 네트워크에서 많이 사용되는 라우팅 프로토콜이다.
- 라우팅 정보에 변화가 생길 경우, 변화된 정보만 네트워크 내의 모든 라우터에 알린다.

- 83 문제에서 설명하는 공격 방법은 스머핑(Smurfing)입니다.

[병행학습]

- 티어드롭(TearDrop) : 데이터의 송·수신 과정에서 패킷의 크기가 커 여러 개로 분할되어 전송될 때 분할 순서를 알 수 있도록 Fragment Offset 값을 함께 전송하는데, TearDrop은 이 Offset 값을 변경시켜 수신 측에서 패킷을 재조립할 때 오류로 인한 과부하를 발생시킴으로써 시스템이 다운되도록 하는 공격 방법
- 스미싱(Smishing) : 문자 메시지(SMS)를 이용해 사용자의 개인 신용 정보를 빼내는 수법으로, 현재 각종 행사 안내, 경품 안내 등의 문자 메시지에 링크를 걸어 안드로이드 앱 설치 파일인 apk 파일을 설치하도록 유도하여 사용자

정보를 빼가는 수법으로 발전하고 있음

- 큐싱(Qshing) : QR코드를 통해 악성 앱의 다운로드를 유도하거나 악성 프로그램을 설치하도록 하는 금융사기 기법의 하나로, QR코드와 개인정보 및 금융정보를 낚는다(Fishing)는 의미의 합성 신조어임

- 84 CMM 모델은 초기, 관리, 정의, 정량적 관리, 최적화의 5단계로 구분합니다.

[병행학습]

CMM(Capability Maturity Model)

- 소프트웨어 개발 조직의 업무 능력 및 조직의 성숙도를 평가하는 모델로, 미국 카네기멜론 대학교의 소프트웨어 공학연구소(SEI)에서 개발하였다.
- CMM의 소프트웨어 프로세스 성숙도는 초기, 관리, 정의, 정량적 관리, 최적화의 5단계로 구분한다.

단계	프로세스	특징
초기(Initial)	정의된 프로세스 없음	작업자 능력에 따라 성공 여부 결정
관리(Managed)	규칙화된 프로세스	특정한 프로젝트 내의 프로세스 정의 및 수행
정의(Defined)	표준화된 프로세스	조직의 표준 프로세스를 활용하여 업무 수행
정량적 관리(Quantitatively Managed)	예측 가능한 프로세스	프로젝트를 정량적으로 관리 및 통제
최적화(Optimizing)	지속적 개선 프로세스	프로세스 역량 향상을 위해 지속적인 프로세스 개선

- 85 문제에서 설명하는 무선 네트워크 기술은 PICONET입니다.

[병행학습]

- SCRUM : 럭비에서 반칙으로 경기가 중단된 경우 양 팀의 선수들이 럭비공을 가운데 두고 상대방을 밀치기 위해 서로 대치해 있는 대형을 의미하는 것으로, 이처럼 팀이 중심이 되어 개발의 효율성을 높인다는 의미가 내포된 용어
- NFC(Near Field Communication, 근거리 무선 통신) : 고주파(HF)를 이용한 근거리 무선 통신 기술로, 13.56MHz 주파수를 이용해 10cm 내에서 최고 424Kbps의 속도로 데이터 전송을 지원함
- 와이선(Wi-SUN) : 스마트 그리드와 같은 장거리 무선 통신을 필요로 하는 사물 인터넷(IoT) 서비스를 위한 저전력 장거리(LPWA) 통신 기술로, 짧은 시간 동안 데이터 전송이 빈번한 검침 분야에 유용하며, 낮은 지연속도, 메시 네트워크 기반 확장성, 펌웨어 업그레이드 용이성 면에서 다른 저전력 장거리 통신 기술에 비해 우월함

- 86 문제에서 설명하는 COCOMO 모델의 소프트웨어 개발 유형은 조직형(Organic Mode)입니다.

[병행학습]

COCOMO의 소프트웨어 개발 유형

- 조직형(Organic Mode) : 기관 내부에서 개발된 중·소 규모의 소프트웨어로, 일괄 자료 처리나 과학 기술 계산용, 비즈니스 자료 처리용으로 5만(50KDSI) 라인 이하의 소프트웨어를 개발하는 유형이며, 사무 처리용, 업무용, 과학용 응용 소프트웨어 개발에 적합함
- 반분리형(Semi-Detached Mode) : 조직형과 내장형의 중간형으로, 트랜잭션 처리 시스템이나 운영체제, 데이터베이스 관리 시스템 등의 30만(300KDSI) 라인 이하의 소프트웨어를 개발하는 유형이며, 컴파일러, 인터프리터와 같

은 유틸리티 개발에 적합함

- **내장형(Embedded Mode)** : 최대형 규모의 트랜잭션 처리 시스템이나 운영체제 등의 30만(300KDSI) 라인 이상의 소프트웨어를 개발하는 유형으로, 신호기 제어 시스템, 미사일 유도 시스템, 실시간 처리 시스템 등의 시스템 프로그램 개발에 적합함

- 87 컴퓨터 사용자의 키보드 움직임을 탐지해 ID, 패스워드 등 개인의 중요한 정보를 몰래 빼가는 해킹 공격은 키로거 공격(Key Logger Attack)입니다.

[병행학습]

정보 보안 침해 공격 관련 용어

- **좀비(Zombie) PC** : 악성코드에 감염되어 다른 프로그램이나 컴퓨터를 조종하도록 만들어진 컴퓨터로, C&C(Command & Control) 서버의 제어를 받아 주로 DDoS 공격 등에 이용됨
- **C&C 서버** : 해커가 원격지에서 감염된 좀비 PC에 명령을 내리고 악성코드를 제어하기 위한 용도로 사용하는 서버를 의미함
- **봇넷(Botnet)** : 악성 프로그램에 감염되어 악의적인 의도로 사용될 수 있는 다수의 컴퓨터들이 네트워크로 연결된 형태를 의미함
- **웜(Worm)** : 네트워크를 통해 연속적으로 자신을 복제하여 시스템의 부하를 높임으로써 결국 시스템을 다운시키는 바이러스의 일종으로, 분산 서비스 거부 공격, 버퍼 오버플로 공격, 슬래머 등이 웜 공격의 한 형태임
- **제로 데이 공격(Zero Day Attack)** : 보안 취약점이 발견되었을 때 발견된 취약점의 존재 자체가 널리 공표되기도 전에 해당 취약점을 통하여 이루어지는 보안 공격으로, 공격의 신속성을 의미함
- **키로거 공격(Key Logger Attack)** : 컴퓨터 사용자의 키보드 움직임을 탐지해 ID, 패스워드, 계좌번호, 카드번호 등과 같은 개인의 중요한 정보를 몰래 빼가는 해킹 공격
- **랜섬웨어(Ransomware)** : 인터넷 사용자의 컴퓨터에 잠입해 내부 문서나 파일 등을 암호화해 사용자가 열지 못하게 하는 프로그램으로, 암호 해독용 프로그램의 전달을 조건으로 사용자에게 돈을 요구하기도 함
- **백도어(Back Door, Trap Door)** : 시스템 설계자가 서비스 기술자나 유지 보수 프로그램 작성자(Programmer)의 액세스 편의를 위해 시스템 보안을 제거하여 만들어놓은 비밀 통로로, 컴퓨터 범죄에 악용되기도 함
- **트로이 목마(Trojan Horse)** : 정상적인 기능을 하는 프로그램으로 위장하여 프로그램 내에 숨어 있다가 해당 프로그램이 동작할 때 활성화되어 부작용을 일으키는 것으로, 자기 복제 능력은 없음

- 88 프로그래머의 수가 10명이고, 월 평균 생산이 200 라인이라면 10명이 생산하는 월 생산은 2,000 라인입니다. 총 라인수가 50,000일 경우 개발에 소요되는 기간은 25(50,000 / 2,000)개월입니다.

[병행학습]

LOC 기법 산정 공식

- 노력(인월) = 개발 기간 × 투입 인원
= LOC / 1인당 월평균 생산 코드 라인 수
- 개발 비용 = 노력(인월) × 단위 비용(1인당 월평균 인건비)
- 개발 기간 = 노력(인월) / 투입 인원
- 생산성 = LOC / 노력(인월)

- 89 Rayleigh-Norden 곡선의 노력 분포도를 이용한 프로젝트 비용 산정 기법은 Putnam 모형입니다.

[병행학습]

- **델파이 모형** : 전문가 감정 기법의 주관적인 편견을 보완하기 위해 많은 전문가의 의견을 종합하여 산정하는 기법으로, 전문가들의 편견이나 분위기에 지배되지 않도록 한 명의 조정자와 여러 전문가로 구성됨
- **COCOMO 모형** : 원시 프로그램의 규모인 LOC(원시 코드 라인 수)에 의한 비용 산정 기법으로, 보험(Boehm)이 제안하였으며, 개발할 소프트웨어의 규모(LOC)를 예측한 후 이를 소프트웨어 종류에 따라 다르게 책정되는 비용 산정 방정식에 대입하여 비용을 산정함
- **기능점수 모형** : 알브레히트(Albrecht)가 제안한 것으로, 소프트웨어의 기능을 증대시키는 요인별로 가중치를 부여하고, 요인별 가중치를 합산하여 총 기능 점수를 산출하며 총 기능 점수와 영향도를 이용하여 기능 점수(FP)를 구한 후 이를 이용해서 비용을 산정하는 기법

- 90 소인수 분해 문제를 이용한 공개키 암호화 기법에 널리 사용되는 암호 알고리즘 기법은 RSA(Rivest Shamir Adleman)입니다.

[병행학습]

주요 암호화 알고리즘

- **SEED**
 - 1999년 한국인터넷진흥원(KISA)에서 개발한 블록 암호화 알고리즘이다.
 - 블록 크기는 128비트이며, 키 길이에 따라 128, 256으로 분류한다.
- **ARIA(Academy, Research Institute, Agency)**
 - 2004년 국가정보원과 산학연협회가 개발한 블록 암호화 알고리즘이다.
 - 블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류한다.
- **DES(Data Encryption Standard)**
 - 1975년 미국 NBS에서 발표한 개인키 암호화 알고리즘
 - 블록 크기는 64비트이며, 키 길이는 56비트이다.
- **AES(Advanced Encryption Standard)**
 - 2001년 미국 표준 기술 연구소(NIST)에서 발표한 개인키 암호화 알고리즘이다.
 - 블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류한다.
- **RSA(Rivest Shamir Adleman)**
 - 1978년 MIT의 라이베스트(Rivest), 샤미르(Shamir), 애들먼(Adelman)에 의해 제안된 공개키 암호화 알고리즘이다.
 - 소인수 분해 문제를 이용한 공개키 암호화 기법에 널리 사용된다.

- 91 문제에서 설명하는 소프트웨어 생명 주기 모형은 나선형 모형(Spiral Model)입니다.

[병행학습]

- **폭포수 모형(Waterfall Model)** : 폭포에서 한번 떨어진 물은 거슬러 올라갈 수 없듯이 소프트웨어 개발도 이전 단계로 돌아갈 수 없다는 전제하에 각 단계를 확실하게 매듭짓고 그 결과를 철저하게 검토하여 승인 과정을 거친 후에 다음 단계를 진행하는 개발 방법론
- **프로토타입 모형(Prototype Model, 원형 모형)** : 사용자의 요구사항을 정확히 파악하기 위해 실제 개발될 소프트웨어에 대한 견본품(Prototype)을 만들어 최종 결과물을 예측하는 모형
- **나선형 모형(Spiral Model, 점진적 모형)** : 폭포수 모형과 프로토타입 모형의 장점에 위험 분석 기능을 추가한 모형

으로, 나선을 따라 돌듯이 여러 번의 소프트웨어 개발 과정을 거쳐 점진적으로 완벽한 최종 소프트웨어를 개발함

• **애자일 모형(Agile Model)** : 고객의 요구사항 변화에 유연하게 대응할 수 있도록 일정한 주기를 반복하면서 개발과정을 진행하는 모형으로, 어느 특정 개발 방법론이 아니라 좋은 것을 빠르고 낭비 없게 만들기 위해 고객과의 소통에 초점을 맞춘 방법론을 통칭함

92 문제의 지문에 제시된 내용은 랜섬웨어(Ransomware)에 대한 설명입니다.

93 백도어 탐지 방법에는 무결성 검사, 열린 포트 확인, 로그 분석, SetUID 파일 검사 등이 있습니다.

94 문제에 제시된 내용은 스택가드(StackGuard)에 대한 설명입니다.

[병행학습]

- **모드체크(Mode Check)** : 영문, 숫자, 특수문자 등 입력될 수 있는 문자의 종류가 제한된 경우 입력 문자를 확인하여 이상 유무를 검색하는 것
- **시스로그(Syslog)** : 시스템 관리자가 송수신한 메시지나 명령, 시스템 이벤트 발생 내용, 시스템 작업과 관련된 정보 등을 저장한 자료의 집합

95 소프트웨어 개발 프레임워크를 적용할 경우 시스템 복잡도의 증가가 아닌 감소 효과를 기대할 수 있습니다.

[병행학습]

소프트웨어 개발 프레임워크

- **프레임워크(Framework)**는 소프트웨어 개발에 공통적으로 사용되는 구성 요소와 아키텍처를 일반화하여 손쉽게 구현할 수 있도록 여러 가지 기능들을 제공해주는 반제품 형태의 소프트웨어 시스템이다.
- **주요 기능** : 예외 처리, 트랜잭션 처리, 메모리 공유, 데이터 소스 관리, 서비스 관리, 쿼리 서비스, 로깅 서비스, 사용자 인증 서비스 등
- **스프링 프레임워크(Spring Framework)** : 자바 플랫폼을 위한 오픈 소스 경량형 애플리케이션 프레임워크로, 동적인 웹 사이트의 개발을 위해 다양한 서비스를 제공함
- **전자정부 프레임워크** : 우리나라의 공공부문 정보화 사업 시 효율적인 정보 시스템의 구축을 지원하기 위해 필요한 기능 및 아키텍처를 제공하는 프레임워크로, 개발 프레임워크의 표준 정립으로 응용 소프트웨어의 표준화, 품질 및 재사용성의 향상을 목적으로 함
- **닷넷 프레임워크(.NET Framework)** : Windows 프로그램의 개발 및 실행 환경을 제공하는 프레임워크로, Microsoft사에서 통합 인터넷 전략을 위해 개발하였으며, 실행을 관리하는 CLR(Common Language Runtime, 공용 언어 런타임)이라는 이름의 가상머신 상에서 작동함

96 시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있는 보안 요소는 무결성(Integrity)입니다.

[병행학습]

보안 요소

- 소프트웨어 개발에 있어 충족시켜야 할 요소 및 요건을 의미한다.
- 주요 보안 요소에는 기밀성, 무결성, 가용성이 있으며, 그 외에도 인증, 부인 방지 등이 있다.
 - 기밀성(Confidentiality) : 시스템 내의 정보와 자원은 인가된 사용자에게만 접근이 허용되며, 정보가 전송 중에 노출되더라도 데이터를 읽을 수 없음
 - 무결성(Integrity) : 시스템 내의 정보는 오직 인가된 사

용자만 수정할 수 있음

- 가용성(Availability) : 인가받은 사용자는 언제라도 사용할 수 있음
- 인증(Authentication) : 시스템 내의 정보와 자원을 사용하려는 사용자가 합법적인 사용자인지를 확인하는 모든 행위로, 대표적 방법으로는 패스워드, 인증용 카드, 지문 검사 등이 있음
- 부인 방지(NonRepudiation) : 데이터를 송·수신한 자가 송·수신 사실을 부인할 수 없도록 송·수신 증거를 제공함

97 문제의 지문에 제시된 내용은 하둡(Hadoop)에 대한 설명입니다.

[병행학습]

- **비컨(Beacon)** : 전파를 이용하는 무선 통신 기술에서 주기적으로 프레임 신호 동기를 맞추고, 송·수신 관련 시스템 정보를 전송하며, 수신 데이터 정보를 전달하는 신호 기술
- **포스퀘어(Foursquare)** : 특정 지역이나 장소를 방문할 때 자신의 위치를 지도에 표시하고, 방문한 곳의 정보를 남길 수 있는 기능을 제공하는 위치 기반 소셜 네트워크 서비스(SNS)
- **멤리스터(Memristor)** : 메모리(Memory)와 레지스터(Resistor)의 합성어로, 전류의 방향과 양 등 기존의 경험을 모두 기억하는 특별한 소자이며, 전원 공급이 끊어졌을 때도 직전에 통과한 전류의 방향과 양을 기억하기 때문에 다시 전원이 공급되면 기존의 상태가 그대로 복원됨

98 크래커가 침입하여 백도어를 만들어 놓거나, 설정 파일을 변경했을 때 분석하는 도구는 Tripwire입니다.

[병행학습]

데이터 무결성 검사 도구

- 시스템의 파일 변경 유무를 확인하고 파일이 변경되었을 경우 이를 관리자에게 알려주는 역할을 수행한다.
- 크래커나 허가받지 않은 내부 사용자들이 시스템에 침입하면 백도어를 만들어 놓거나 시스템 파일을 변경하여 자신의 흔적을 감추는데, 이를 무결성 검사 도구를 이용하여 감지할 수 있다.
- 데이터 무결성 도구는 현재 파일 및 디렉토리의 상태를 해시(Hash) 함수를 이용하여 DB 저장한 후 모니터링을 수행하다가 파일이나 디렉토리가 DB에 저장해둔 상태와 달라지는 경우 관리자에게 변경 사실을 알려준다.
- 대표적인 데이터 무결성 검사 도구에는 Tripwire, AIDE, Samhain, Claymore, Slipwire, Fcheck 등이 있다.

99 폭포수 모형은 이전 단계로 돌아갈 수 없으므로 개발 중 발생한 요구사항을 쉽게 반영할 수 없습니다.

[병행학습]

폭포수 모형(Waterfall Model)

- 폭포에서 한번 떨어진 물은 거슬러 올라갈 수 없듯이 소프트웨어 개발도 이전 단계로 돌아갈 수 없다는 전제하에 각 단계를 확실히 매듭짓고 그 결과를 철저히 검토하여 승인 과정을 거친 후에 다음 단계를 진행하는 개발 방법론이다.
- 폭포수 모형은 소프트웨어 공학에서 가장 오래되고 가장 폭넓게 사용된 전통적인 소프트웨어 생명 주기 모형으로, 고전적 생명 주기 모형이라고도 한다.
- 소프트웨어 개발 과정의 한 단계가 끝나야만 다음 단계로 넘어갈 수 있는 선형 순차적 모형이다.
- 제품의 일부가 될 매뉴얼을 작성해야 한다.
- 각 단계가 끝난 후에는 다음 단계를 수행하기 위한 결과물이 명확하게 산출되어야 한다.

- 두 개 이상의 과정이 병행하여 수행되지 않는다.

100 [전문가의 조언]

국제 표준 품질 기준은 테일러링 개발 방법론의 외부 기준에 해당합니다.

소프트웨어 개발 방법론 테일러링

- 프로젝트 상황 및 특성에 맞도록 정의된 소프트웨어 개발 방법론의 절차, 사용기법 등을 수정 및 보완하는 작업이다.
- 소프트웨어 개발 방법론 테일러링 작업 시 고려해야 할 사항에는 내부적 요건과 외부적 요건이 있다.
- 내부적 요건

목표 환경	시스템의 개발 환경과 유형이 서로 다른 경우 테일러링이 필요함
요구사항	프로젝트의 생명 주기 활동에서 개발, 운영, 유지보수 등 프로젝트에서 우선적으로 고려할 요구사항이 서로 다른 경우 테일러링이 필요함
프로젝트 규모	비용, 인력, 기간 등 프로젝트의 규모가 서로 다른 경우 테일러링이 필요함
보유 기술	프로세스, 개발 방법론, 산출물 등이 서로 다른 경우 테일러링이 필요함

- 외부적 요건

법적 제약사항	프로젝트별로 적용될 IT Compliance가 서로 다른 경우 테일러링이 필요함
표준 품질 기준	금융, 제도 등 분야별 표준 품질 기준이 서로 다른 경우 테일러링이 필요함