



시험에 나오는 것만 공부한다!

시나공시리즈

모의고사 & 정답 및 해설 2020년 4회 대비 정보처리기사 필기



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

1. 다음 설명에 해당하는 인터페이스 요구사항 검토 방법은?

요구사항 명세서 작성자를 제외한 다른 검토 전문가들이 요구사항 명세서를 확인하면서 결함을 발견하는 형태의 검토 방법

- ① 동료검토 ② 워크스루
- ③ 인스펙션 ④ CASE 도구

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

- ① 소프트웨어 생명주기 전 단계를 연결한다.
- ② 다양한 소프트웨어 개발 모형을 지원한다.
- ③ 소프트웨어의 개발 기법을 실용화 할 수 있고, 문서화를 쉽게 작성할 수 있다.
- ④ 상위, 중위, 하위 케이스로 분류된다.

3. 자료 사전(Data Dictionary)의 표기법에 대한 설명으로 옳지 않은 것은?

- ① = : 자료의 정의 ② + : 자료의 연결
- ③ () : 자료의 설명 ④ { } : 자료의 반복

4. 객체지향의 구성 요소에 대한 설명으로 옳지 않은 것은?

- ① 객체가 수행하는 기능으로, 객체가 갖는 데이터(속성, 상태)를 처리하는 알고리즘을 메소드라고 한다.
- ② 부모 클래스의 모든 속성과 연산을 자식 클래스가 물려받는 것을 상속이라고 한다.
- ③ 하나의 메시지에 대해 각각의 객체가 가지고 있는 고유한 방법으로 응답할 수 있는 능력은 캡슐화이다.
- ④ 공통된 속성과 연산을 갖는 객체의 집합을 클래스라고 한다.

5. 린바우의 분석 기법에서 다음 설명에 해당하는 것은?

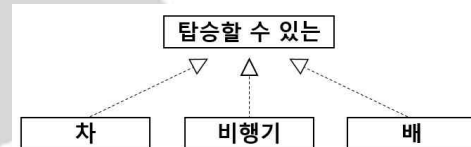
- 시스템에서 요구되는 객체를 찾아내어 속성과 연산 식별 및 객체들 간의 관계를 규정하여 객체 다이어그램으로 표시하는 것이다.
- 분석 활동의 세 가지 모델 중 가장 중요하며 선행되어야 할 모델링이다.

- ① 기능 모델링 ② 동적 모델링
- ③ 객체 모델링 ④ 정적 모델링

6. GoF의 디자인 패턴에 대한 설명으로 가장 옳지 않은 것은?

- ① 재사용할 수 있는 기본형 코드들이 포함되어 있다.
- ② 학습 및 연구용으로 주로 활용되며, 현업에서는 많이 활용되지 않는다.
- ③ 변형을 가하거나 특정 요구사항을 반영하면 다른 패턴으로 변화될 수 있다.
- ④ 생성 패턴, 구조 패턴, 행위 패턴으로 구분된다.

7. 아래의 UML 모델에서 각 클래스 간의 관계로 옳은 것은?



- ① 실체화 관계 ② 의존 관계
- ③ 일반화 관계 ④ 추상화 관계

8. UML에서 활용되는 다이어그램의 이름과 설명의 연결이 올바르지 않은 것은?

- ① 클래스 다이어그램 : 시퀀스 다이어그램과 같이 동작에 참여하는 객체들이 주고받는 메시지를 표현하는데, 메시지뿐만 아니라 객체들 간의 연관까지 표현한다.
- ② 배치 다이어그램 : 결과물, 프로세스, 컴포넌트 등 물리적 요소들의 위치를 표현한다.
- ③ 유스케이스 다이어그램 : 사용자의 요구를 분석하는 것으로, 기능 모델링 작업에 사용한다.
- ④ 활동 다이어그램 : 시스템이 어떤 기능을 수행하는지 객체의 처리 로직이나 조건에 따른 처리의 흐름을 순서에 따라 표현한다.

9. ISO/IEC 9126에서 소프트웨어가 다른 환경에서도 얼마나 쉽게 적용할 수 있는지 정도를 나타내는 품질특성은?

- ① 변경성 ② 안정성
- ③ 이식성 ④ 친밀성

10. 새로 들어온 제품에 대해 명칭이나 약호와 관계있는 숫자, 문자, 기호 등을 이용하여 코드를 부여하고자 한다. 이 때 가장 적합한 코드의 종류는?

- ① 구분 코드(Block Code)
- ② 그룹 분류 코드(Group Classification Code)
- ③ 연상 코드(Mnemonic Code)
- ④ 순차 코드(Sequence Code)

11. UI 설계 지침에 대한 설명으로 옳지 않은 것은?

- ① 다양한 계층이 사용할 수 있도록 설계해야 한다.
- ② 오류가 발생해도 사용자가 인지할 수 없도록 설계해야 한다.
- ③ 사용자가 쉽게 이해하고 편리하게 사용할 수 있는 환경을 제공해야 한다.
- ④ 조작 방법을 단순화시켜 인지적 부담을 감소시켜야 한다.

4회 대비

12. 다음 설명에 해당하는 디자인 패턴은?

- 생성 패턴의 하나로, 하나의 객체를 생성하면 생성된 객체를 어디서든 참조할 수 있지만, 여러 프로세스가 동시에 참조할 수는 없다.
- 클래스 내에서 인스턴스가 하나뿐임을 보장하며, 불필요한 메모리 낭비를 최소화 할 수 있다.

- ① 추상 팩토리 ② 빌더
③ 프로토타입 ④ 싱글톤

13. 다음 중 객체지향 분석의 방법론에 속하지 않는 것은?

- ① Coad와 Yourdon 방법 ② Booch 방법
③ Jacobson 방법 ④ Function Point 방법

14. UML의 다이어그램에서 사물(Things)에 대한 설명으로 옳지 않은 것은?

- ① 구조 사물은 시스템의 개념적, 물리적 요소를 표현한다.
② 행동 사물은 설명이나 제약에 따른 요소들의 행위를 표현한다.
③ 그룹 사물의 대표적인 것으로 패키지(Package)가 있다.
④ 주해 사물의 대표적인 것으로 노트(Note)가 있다.

15. 객체 지향 미들웨어 중 하나로 코바(CORBA) 표준 스펙을 구현한 소프트웨어는?

- ① ORB ② TP-Monitor
③ RPC ④ MOM

16. 사용자의 요구사항을 충분히 분석할 목적으로, 시스템의 일부분 또는 시제품을 일시적으로 간결히 구현한 다음 다시 요구사항을 반영하는 과정을 반복하는 점진적 개발 생명주기를 갖는 모델은?

- ① Waterfall Model ② Spiral Model
③ Prototype Model ④ Agile Model

17. 다음 설명에 해당하는 도표는?

시스템의 기능을 여러 개의 고유 모듈들로 분할하여 이들 간의 인터페이스를 계층 구조로 표현한 것으로, 가시적 도표(Visual Table of Contents), 총체적 도표(Overview Diagram), 세부적 도표(Detail Diagram)가 있다.

- ① Flow Chart ② Burn-down Chart
③ Visual Diagram ④ HIPO Chart

18. 요구사항 분석 기법 중 개념 모델링(Conceptual Modeling)에 대한 설명으로 가장 옳지 않은 것은?

- ① 기능 요구사항과 비기능 요구사항으로 분류한다.
② 모델링 표기는 주로 UML을 사용한다.
③ 현실 세계의 상황을 단순화하여 개념적으로 표현한 것을 모델 이라고 한다.
④ 문제가 발생하는 상황을 쉽게 이해시키고 해결책을 설명할 수 있다.

19. 시스템의 일부가 결함 또는 고장으로 기능이 정지되더라도 해당 부분의 기능만 수행이 불가능할 뿐 전체 시스템은 정상적으로 수행이 가능해야 하는 시스템에 가장 적합한 아키텍처 패턴은?

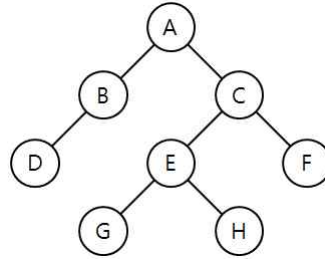
- ① Client - Server Pattern ② Blackboard Pattern
③ Peer-To-Peer Pattern ④ Master-Slave Pattern

20. Niklaus Wirth에 의해 제안된 하향식 설계 전략으로, 문제를 상위의 중요 개념으로부터 하위의 개념으로 구체화시키는 분할 기법은?

- ① 정보 은닉 ② 캡슐화
③ 단계적 분해 ④ 상속

제2과목 소프트웨어 개발

21. 다음 트리에서 터미널 노드 수는?



- ① 2 ② 3
③ 4 ④ 8

22. 애플리케이션 테스트 중 코딩 직후 수행하는 테스트로, 인터페이스, 외부적 I/O, 자료 구조, 독립적 기초 경로, 오류 처리 경로, 경계 조건 등을 검사하는 테스트는?

- ① 단위 테스트 ② 통합 테스트
③ 시스템 테스트 ④ 인수 테스트

23. 소프트웨어 유지보수 유형 중 환경의 변화를 기존의 소프트웨어에 반영하기 위해 수행하는 활동으로, 주로 프로그래밍 환경이나 주변 장치 또는 시스템 요소가 향상되었을 때 대처하는 활동은?

- ① Preventive Maintenance
② Adaptive Maintenance
③ Corrective Maintenance
④ Perfective Maintenance

24. 모듈에 관한 다음 설명에서 괄호에 들어갈 알맞은 용어는?

()은/는 비용과 개발 시간을 절약하기 위해 이미 개발된 기능들을 파악하고 재구성하여 새로운 시스템 또는 기능 개발에 사용하기 적합하도록 최적화 시키는 작업이다.

- ① Reverse Engineering
② Re-engineering
③ Reuse
④ Repository

25. 상향식(Bottom Up) 통합 테스트에서 상위 모듈 없이 하위 모듈이 있는 경우 하위 모듈을 구동할 때 사용 하는 도구는?

- ① Test Stub ② Test Oracle
③ Test Script ④ Test Driver

26. 다음 물리데이터 저장소의 설계에 대한 설명 중 클러스터드 인덱스 테이블(Clustered Index Table)에 대한 설명으로 옳은 것은?

- ① 현재 사용되는 대부분의 DBMS에서 사용되는 형태이다.
② 키(Key)의 순서에 따라 데이터가 저장되는 테이블이다.
③ 대용량의 테이블을 작은 논리적 단위로 나눈 테이블이다.
④ 데이터베이스에서 일반 테이블처럼 이용할 수 있는 파일로, 데이터베이스 내에 객체로 존재한다.

27. 디지털 저작권 관리(DRM)에 대한 설명으로 가장 옳지 않은 것은?

- ① 콘텐츠의 생성, 유통, 이용까지의 전 과정에 걸쳐 사용되는 디지털 콘텐츠 관리 및 보호 기술이다.
② 패키징 수행 전 라이선스 정보를 클리어링 하우스에 등록 한다.
③ 원본 콘텐츠가 아날로그인 경우에는 변환 작업을 수행한 뒤 패키징을 한다.
④ 암호화, 식별 기술, 크랙 방지, 인증 등의 기술이 사용된다.

28. 소프트웨어 패키징에 대한 설명으로 옳지 않은 것은?

- ① 소스 코드는 향후 관리를 고려하여 모듈화하여 패키징한다.
- ② 소프트웨어 개발 주기가 끝날 때마다 패키징을 수행한다.
- ③ 사용자의 시스템 환경, 즉 운영체제(OS), CPU, 메모리 등에 필요한 최소 환경을 정의한다.
- ④ 다양한 환경에서 소프트웨어를 손쉽게 사용할 수 있도록 커스터마이징된 형태로 패키징한다.

29. 소프트웨어 버전 등록 과정으로 올바른 것은?

- | | |
|----------------|-----------------|
| ㉠ 가져오기(Import) | ㉡ 인출(Check-Out) |
| ㉢ 예치(Commit) | ㉣ 동기화(Update) |
| ㉤ 차이(Diff) | |

- ① ㉠ → ㉡ → ㉢ → ㉣ → ㉤
- ② ㉠ → ㉢ → ㉡ → ㉣ → ㉤
- ③ ㉠ → ㉣ → ㉢ → ㉡ → ㉤
- ④ ㉠ → ㉡ → ㉢ → ㉣ → ㉤

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

- ① 스택의 삽입 및 삭제
- ② 이진 검색
- ③ 합병 정렬
- ④ 파보나치 수열

31. EAI(Enterprise Application Integration)의 구축 유형 중 애플리케이션 사이에 미들웨어를 두어 확장성 및 대용량 처리에 특화된 구축 방법은?

- ① Point to Point
- ② Hybrid
- ③ ESB
- ④ Hub & Spoke

32. 스택에서 순서가 A, B, C, D로 정해진 입력 자료를, push → push → pop → push → pop → push → pop → pop으로 연산 했을 때 출력은?

- ① C, B, D, A
- ② B, C, D, A
- ③ B, C, A, D
- ④ C, B, A, D

33. 형상 관리(SCM)의 기능이나 목적 대한 설명으로 가장 옳지 않은 것은?

- ① 그래프 구조로 구분하여 수정 및 추적이 용이하도록 한다.
- ② 유지 보수 과정에서 생성된 다른 버전의 형상 항목 또한 관리 대상에 포함된다.
- ③ 베이스 라인이 잘 반영되도록 조정한다.
- ④ 개발의 전체 비용을 줄이고 방해 요인을 최소화한다.

34. 다음 자료에 대하여 삽입(insertion) 정렬 기법을 사용하여 오름차순으로 정렬하고자 한다. 1회전 후의 결과는?

5, 4, 3, 2, 1

- ① 4, 3, 2, 1, 5
- ② 3, 4, 5, 2, 1
- ③ 4, 5, 3, 2, 1
- ④ 1, 2, 3, 4, 5

35. 다음 전위식(prefix)을 후위식(postfix)으로 옳게 표현한 것은?

- / * A + B C D E

- ① A B C + * D / E -
- ② A B * C D / + E -
- ③ A B * C + D / E -
- ④ A B C + D / * E -

36. 다음 중 블랙박스 테스트(Black Box Test)의 종류로만 묶여진 것은?

- ① 동치 분할 검사, 데이터 흐름 검사
- ② 경계값 분석, 기초 경로 검사
- ③ 원인-효과 그래프 검사, 조건 검사
- ④ 비교 검사, 오류 예측 검사

37. 클린 코드 작성 원칙 중 한 번에 한 가지를 처리하도록 코드를 작성하고 클래스/메소드/함수 등의 단위로 분리하는 원칙은?

- ① 가독성
- ② 단순성
- ③ 추상화
- ④ 의존성 배제

38. 소프트웨어의 품질 특성과 평가를 위한 표준 지침으로서 ISO/IEC 9126을 개선하여 2011년 발표한 국제 표준은?

- ① ISO/IEC 12119
- ② ISO/IEC 14598
- ③ ISO/IEC 15504
- ④ ISO/IEC 25010

39. 다음 중 APM(Application Performance Monitoring) 도구에 속하지 않는 것은?

- ① Nagios
- ② Zabbix
- ③ FitNesse
- ④ Scouter

40. 인터페이스 보안을 위해 데이터베이스 영역에 해당하는 보안 사항으로 옳지 않은 것은?

- ① 데이터베이스, 스키마, 엔티티 접근 권한
- ② 프로시저, 트리거 등 데이터베이스 동작 객체에 대한 보안
- ③ 애플리케이션 코드 상의 보안
- ④ 데이터의 암호화 및 익명화

제3과목 데이터베이스 구축

41. 속성(Attribute)에 대한 설명으로 틀린 것은?

- ① 속성은 개체의 특성을 기술한다.
- ② 속성은 데이터베이스를 구성하는 가장 작은 논리적 단위이다.
- ③ 속성은 파일 구조상 데이터 항목 또는 데이터 필드에 해당된다.
- ④ 속성의 수를 "Cardinality"라고 한다.

42. 병행제어의 목적으로 옳지 않은 것은?

- ① 사용자에게 대한 응답시간 최소화
- ② 시스템 활용도 최대화
- ③ 데이터베이스 일관성 유지
- ④ 데이터베이스 공유도 최소화

43. 다음의 성적 테이블에서 학생별 점수의 평균을 구하기 위한 SQL문으로 옳은 것은?

성명	과목	점수
홍길동	국어	80
홍길동	영어	68
홍길동	수학	97
강감찬	국어	58
강감찬	영어	97
강감찬	수학	65

- ① SELECT 성명, (AVG)점수 FROM 성적 ORDER BY 성명;
- ② SELECT 성명, AVG(점수) FROM 성적 ORDER BY 성명;
- ③ SELECT 성명, (AVG)점수 FROM 성적 GROUP BY 성명;
- ④ SELECT 성명, AVG(점수) FROM 성적 GROUP BY 성명;

44. 관계대수에 대한 설명으로 옳지 않은 것은?

- ① 원하는 릴레이션을 정의하는 방법을 제공하며 비절차적 언어이다.
- ② 릴레이션 조작을 위한 연산의 집합으로 피연산자와 결과가 모두 릴레이션이다.
- ③ 일반 집합 연산과 순수 관계 연산으로 구분된다.

4회 대비

④ 질의에 대한 해를 구하기 위해 수행해야 할 연산의 순서를 명시한다.

45. 정규화 과정에서 부분적 함수 종속성을 제거하는 단계는?

- ① 1NF → 2NF ② 2NF → 3NF
③ 3NF → BCNF ④ BCNF → 4NF

46. DML에 해당하는 것으로만 나열된 것은?

㉠ SELECT	㉡ UPDATE
㉢ INSERT	㉣ GRANT

- ① ㉠, ㉡, ㉢ ② ㉠, ㉡, ㉣
③ ㉠, ㉢, ㉣ ④ ㉠, ㉡, ㉢, ㉣

47. 릴레이션 R_1 에 저장된 튜플이 릴레이션 R_2 에 있는 튜플을 참조하려면 참조되는 튜플이 반드시 R_2 에 존재해야 한다는 무결성 규칙은?

- ① 개체 무결성 규칙(Entity Integrity Rule)
② 참조 무결성 규칙(Referential Integrity Rule)
③ 영역 무결성 규칙(Domain Integrity Rule)
④ 트리거 규칙(Trigger Rule)

48. 아래와 같은 결과를 만들어 내는 SQL문은?

[공급자 Table]

공급자번호	공급자명	위치
16	대신공업사	수원
27	삼진사	서울
39	삼양사	인천
62	진아공업사	대전
70	신촌상사	서울

[결과]

공급자번호	공급자명	위치
16	대신공업사	수원
70	신촌상사	서울

- ① SELECT * FROM 공급자 WHERE 공급자명 LIKE '%신%'
② SELECT * FROM 공급자 WHERE 공급자명 LIKE '대%'
③ SELECT * FROM 공급자 WHERE 공급자명 LIKE '%사'
④ SELECT * FROM 공급자 WHERE 공급자명 LIKE '_사'

49. 조건을 만족하는 릴레이션의 수평적 부분집합으로 구성하며, 연산자의 기호는 그리스 문자 시그마(σ)를 사용하는 관계대수 연산은?

- ① Select ② Project
③ Join ④ Division

50. 다음 설명에 해당하는 종속성은?

A, B, C 3개의 속성을 가진 릴레이션 R에서 어떤 복합 속성(A, C)에 대응하는 B 값의 집합이 A 값에만 종속되고 C 값에는 무관할 때 'A → B'로 표기한다.

- ① 부분 함수적 종속 ② 완전 함수적 종속
③ 조인 종속 ④ 다치 종속

51. 반정규화(Denormalization)에 대한 설명으로 가장 옳지 않은 것은?

- ① 데이터의 일관성 및 정합성이 저하될 수 있다.
② 일관성과 무결성을 우선으로 할지, 데이터베이스의 성능과 단순화를 우선으로 할지를 결정해야 한다.
③ 반정규화 방법에는 테이블 통합, 테이블 분할 등이 있다.
④ 테이블 통합은 다른 서버에 저장된 테이블을 자주 이용해야 하는 경우 사용한다.

52. 분산 데이터베이스에 대한 설명으로 거리가 먼 것은?

- ① 분산 제어가 용이하다.
② 지역 자치성이 높다.
③ 효율성과 융통성이 높다.
④ DBMS가 수행할 기능이 간단하다.

53. 시스템 카탈로그에 대한 설명으로 옳지 않은 것은?

- ① 시스템 카탈로그는 DBMS가 생성하고 유지하는 데이터베이스 내의 특별한 테이블들의 집합체이다.
② 일반 사용자도 시스템 카탈로그의 내용을 검색할 수 있다.
③ 시스템 카탈로그에 대한 갱신은 데이터베이스의 무결성 유지를 위하여 사용자가 직접 갱신해야 한다.
④ 시스템 카탈로그 내의 각 테이블은 DBMS에서 지원하는 개체들에 관한 정보를 포함한다.

54. 다음 중 교착상태가 발생할 수 있는 필요 충분 조건은?

- ① 중단 조건(Preemption)
② 환형 대기(Circular Wait)
③ 기아 상태(Starvation)
④ 동기화(Synchronization)

55. 데이터베이스에서 개념적 설계 단계에 대한 설명으로 틀린 것은?

- ① 산출물로 ER-D가 만들어진다.
② DBMS에 독립적인 개념 스키마를 설계한다.
③ 트랜잭션 인터페이스를 설계한다.
④ 논리적 설계 단계의 앞 단계에서 수행된다.

56. 트랜잭션에서 SQL 문들에 의해 수행된 모든 갱신을 취소시켜 데이터베이스를 트랜잭션의 첫 구문이 실행되기 전 상태로 되돌리는 트랜잭션 연산은?

- ① ROLLBACK ② UPDATE
③ CANCEL ④ COMMIT

57. 다음 표와 같은 판매실적 테이블에서 서울지역에 한하여 판매액 내림차순으로 지점명과 판매액을 출력하고자 한다. 가장 적절한 SQL 구문은?

[테이블명 : 판매실적]

도시	지점명	판매액
서울	강남 지점	330
서울	강북 지점	168
광주	광주 지점	197
서울	강서 지점	158
서울	강동 지점	197
대전	대전 지점	165

- ① SELECT 지점명, 판매액 FROM 판매실적
WHERE 도시 = "서울"
ORDER BY 판매액 DESC;
② SELECT 지점명, 판매액 FROM 판매실적
ORDER BY 판매액 DESC;
③ SELECT 지점명, 판매액 FROM 판매실적
WHERE 도시 = "서울" ASC;
④ SELECT * FROM 판매실적
WHEN 도시 = "서울"
ORDER BY 판매액 DESC;

4회 대비

58. 다음 설명에 해당하는 트랜잭션의 특성은?

- 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.

- ① Atomicity ② Consistency
③ Isolation ④ Durability

59. 자료 구조 중 선형 구조만으로 나열된 것은?

- ① 트리, 그래프
② 트리, 그래프, 스택, 큐
③ 트리, 배열, 스택, 큐
④ 배열, 스택, 큐

60. 동적 SQL의 설명으로 틀린 것은?

- ① SQL 구성 시 커서(Cursor)를 통해 처리한다.
② NVL 함수 없이 로직을 통해 SQL을 작성한다.
③ 문자열 변수에 SQL 코드를 넣어 처리한다.
④ 실행속도가 느리고 사전 검사가 불가능하다.

제4과목 프로그래밍 언어 활용

61. 같은 타입의 데이터 집합을 만들어 저장할 때 사용하는 자료형으로, 중괄호 안에 콤마로 구분하여 값을 나열하는 것은?

- ① Integer Type ② Array Type
③ Boolean Type ④ Character String Type

62. UNIX에 대한 설명으로 옳지 않은 것은?

- ① 상당 부분 C 언어를 사용하여 작성되었으며, 이식성이 우수하다.
② 사용자는 하나 이상의 작업을 백그라운드에서 수행할 수 있어 여러 개의 작업을 병행 처리할 수 있다.
③ 셸(Shell)은 프로세스 관리, 기억장치 관리, 입출력 관리 등의 기능을 수행한다.
④ 두 사람 이상의 사용자가 동시에 시스템을 사용할 수 있어 정보와 유틸리티들을 공유하는 편리한 작업 환경을 제공한다.

63. 다음 C 프로그램의 결과 값은?

```
main( ) {
    int sum = 0;
    for(int a = 20; a > 5; a -= 5)
        sum += a;
    printf("%d", sum);
}
```

- ① 53 ② 50
③ 45 ④ 38

64. 다음 설명에 해당하는 결합도는?

모듈 간의 인터페이스로 배열이나 레코드 등의 자료 구조가 전달될 때의 결합도로, 두 모듈이 동일한 자료 구조를 조회하는 경우의 결합도이다.

- ① 스템프 결합도 ② 제어 결합도
③ 외부 결합도 ④ 내용 결합도

65. 다음 중 C언어의 변수명으로 사용할 수 없는 것은?

- ① a2bc ② dir
③ _sum ④ return

66. 다음 Java 프로그램의 결과 값은?

```
public static void main(String[] args) {
    int a = 7, b = 11;
    int c = a > b ? a * b : b % a;
    System.out.println(c);
}
```

- ① 4 ② 18
③ 28 ④ 77

67. 네트워크 주소 할당 시 할당된 네트워크 주소를 다시 여러 개의 작은 네트워크로 나누어 사용하는 것은?

- ① IPv6 ② Multicast
③ Subnetting ④ DNS

68. 기억장치 계층 구조에 대한 설명으로 가장 옳지 않은 것은?

- ① 상위부터 하위까지 레지스터, 캐시, 주기억장치, 보조기억장치로 구분된다.
② 상위의 기억장치일수록 접근 속도가 빠르다.
③ 하위의 기억장치일수록 용량이 크고 저가이다.
④ 주기억장치에 있는 데이터는 CPU가 직접 액세스할 수 없다.

69. 다음 응집도를 강한 것부터 약한 것으로 순서대로 나열한 것은?

- ㉠ 교환적 응집도 ㉡ 순차적 응집도
㉢ 논리적 응집도 ㉣ 기능적 응집도

- ① ㉢ → ㉠ → ㉡ → ㉣
② ㉣ → ㉡ → ㉠ → ㉢
③ ㉢ → ㉡ → ㉠ → ㉣
④ ㉣ → ㉠ → ㉡ → ㉢

70. OSI 7계층에서 TCP는 어떤 계층에 해당하는가?

- ① 세션 계층 ② 네트워크 계층
③ 전송 계층 ④ 데이터 링크 계층

71. TCP/IP의 네트워크 액세스 계층의 프로토콜에 속하지 않는 것은?

- ① Ethernet ② HDLC
③ X.25 ④ ARP

72. 프로토콜의 기능 중 흐름 제어(Flow Control)에 대한 설명으로 옳은 것은?

- ① 전송할 데이터를 전송에 알맞은 일정 크기의 작은 블록으로 자르고 결합하는 기능
② 단편화된 데이터에 프로토콜 제어 정보를 추가하는 기능
③ 수신 측의 처리 능력에 따라 송신 측에서 송신하는 데이터의 전송량이나 전송 속도를 조절하는 기능
④ 송·수신 측이 같은 상태를 유지하도록 타이밍을 맞추는 기능

73. 배치 프로그램의 자동 수행 주기 중 사용자가 특정 조건을 설정해두고 해당 조건이 충족될 때만 수행되도록 하는 것은?

- ① 정기 배치 ② 이벤트성 배치
③ On-Demand 배치 ④ 사용자 배치

74. 다음 중 운영체제(OS)가 아닌 것은?

- ① Prezi ② Windows
③ Linux ④ Android

4회 대비

75. RR(Round-Robin) 스케줄링에 대한 설명으로 옳지 않은 것은?

- ① 일괄처리 시스템을 위해 고안된 방식이다.
- ② FCFS 기법과 동일하게 준비상태 큐에 먼저 들어온 프로세스가 먼저 CPU를 할당받는다.
- ③ 시간 할당량이 커지면 FCFS 스케줄링과 같은 효과를 얻을 수 있다.
- ④ 할당된 시간의 크기가 작을수록 작은 프로세스들에게 유리하다.

76. IEEE 802.4의 표준안 내용으로 맞는 것은?

- ① 토큰 버스 LAN ② 블루투스
- ③ CSMA/CD LAN ④ 무선 LAN

77. 프로세스가 자원을 기다리고 있는 시간에 비례하여 우선순위를 부여함으로써 무기한 연기 문제를 방지하는 기법은?

- ① Deadly Embrace ② Reusable
- ③ Circular Wait ④ Aging

78. Windows의 명령 프롬프트 창에서 사용하는 명령어 중 파일의 속성을 변경할 때 사용하는 명령어는?

- ① DIR ② CHMOD
- ③ ATTRIB ④ CHKDSK

79. 스크립트 언어의 특징에 대한 설명으로 옳지 않은 것은?

- ① 컴파일 없이 바로 실행할 수 있다.
- ② 런타임 오류가 많이 발생한다.
- ③ 소스 코드를 쉽고 빠르게 수정할 수 있다.
- ④ 개발 시간과 실행 속도가 빠르다.

80. C언어의 포인터 변수에 대한 설명으로 옳지 않은 것은?

- ① 변수의 주소를 저장할 때 사용하는 변수이다.
- ② 변수의 주소를 알아낼 때는 변수 앞에 #을 붙인다.
- ③ 선언 시 변수명 앞에 간접 연산자 *를 붙인다.
- ④ 동적 메모리 영역인 힙 영역에 접근하는 변수이다.

제5과목 : 정보시스템 구축 관리

81. 단말 장치 간 Point-to-Point 방식으로 연결시킨 형태로, 단말 장치의 추가/제거 및 기밀 보호가 어렵고, 하나라도 고장 나면 전체 통신망에 영향을 미치는 네트워크의 설치 형태는?

- ① 성형 ② 링형
- ③ 버스형 ④ 망형

82. Secure SDLC의 설계 단계에서 수행해야 할 보안 활동으로 적합하지 않은 것은?

- ① 보안대책, 소요예산, 사고 발생 시 영향 범위와 대응책 등을 수립한다.
- ② 외부의 사이버 공격으로부터 개발 환경을 보호하기 위해 네트워크를 분리하거나 방화벽을 설치한다.
- ③ 각 정보의 보안 수준을 보안 요소별로 등급을 구분하여 분류한다.
- ④ 출입통제, 개발 공간 제한, 폐쇄회로 등의 감시설비를 설치한다.

83. 미국 카네기멜론 대학교의 소프트웨어 공학연구소(SEI)에서 개발한 것으로, 소프트웨어 개발 조직의 업무 능력 및 조직의 성숙도를 평가하는 모델은?

- ① ISO/IEC 12207 ② CMMI
- ③ SPICE ④ SCRUM

84. 다음 Java 코드에서 밑줄로 표시된 부분에는 어떤 보안 약점이 존재하는가?

```
public static void main(String[] args) {
    int a = 5;
    a = func(a);
}
static int func(int a) {
    return a <= 5 ? func(a) : 3;
}
```

- ① 종료되지 않는 반복문 또는 재귀함수
- ② 널 포인터 역참조
- ③ 하드코딩된 암호화 키 사용
- ④ 초기화되지 않은 변수 사용

85. Java에서 외부의 접근을 제한하는 접근 제한자 중 클래스 내부와 패키지 내부에서만 접근이 가능하며, 외부 패키지나 해당 클래스를 상속받은 외부의 하위 클래스에서는 접근이 불가능한 접근 제한자는?

- ① Public ② Protected
- ③ Default ④ Private

86. CASE에 대한 설명으로 거리가 먼 것은?

- ① 자동화된 기법을 통해 소프트웨어 품질이 향상된다.
- ② 소프트웨어 부품의 재사용성이 향상된다.
- ③ 프로토타입 모델에 위험 분석 기능을 추가한 생명주기 모형이다.
- ④ 소프트웨어 도구와 방법론의 결합이다.

87. 다음이 설명하고 있는 라우팅 프로토콜은?

• 자율 시스템(AS) 간의 라우팅 프로토콜로, EGP의 단점을 보완하기 위해 만들어졌다.
• 초기에 라우터들이 연결될 때는 전체 경로 제어표(라우팅 테이블)를 교환하고, 이후에는 변화된 정보만을 교환한다.

- ① IGP ② OSPF
- ③ RIP ④ BGP

88. 가상의 클라이언트로 위장하여 3-way-handshake 과정을 의도적으로 중단시킴으로써 공격 대상자인 서버가 대기 상태에 놓여 정상적인 서비스를 수행하지 못하게 하는 공격 방법은?

- ① Smurfing ② SYN Flooding
- ③ Ping of Death ④ Land Attack

89. 다음 설명에 해당하는 정보 기술 용어는?

플래시 애니메이션 기술과 웹 서버 애플리케이션 기술을 통합하여 기존 HTML 보다 역동적이고 인터랙티브한 웹페이지를 제공하는 신개념의 플래시 웹페이지 제작 기술

- ① RIA ② PIA
- ③ Mashup ④ Semantic Web

90. 긴 시간동안 안정적인 서비스 운영을 위해 장애 발생 시 즉시 다른 시스템으로 대체 가능한 환경을 구축하는 메커니즘을 가리키는 HW 관련 기술 용어는?

- ① 고가용성 ② RAID
- ③ 컴패니언 스크린 ④ MEMS

91. 다음 설명에 해당하는 COCOMO 유형은?

컴파일러, 인터프리터와 같은 유틸리티 개발에 적합한 유형으로, 30만(300KDSI) 라인 이하의 소프트웨어에 적합한 비용 산정 기법이다.

4회 대비

- ① 조직형 ② 반분리형
③ 중간형 ④ 발전형

92. 암호화 알고리즘 RSA에 대한 설명으로 옳지 않은 것은?

- ① 1978년 MIT의 라이베스트(Rivest), 샤미르(Shamir), 애들먼(Adelman)에 의해 개발되었다.
② 큰 숫자를 소인수 분해 하기 어렵다는 것에 기반하여 만들어졌다.
③ 공개키와 개인키를 사용하는 공개키 암호화 기법을 사용한다.
④ 암호화/복호화 속도가 빨라 휴대용 단말기에서 많이 사용된다.

93. 프로젝트 관리(Project Management)의 관리 유형과 내용의 연결이 올바르지 않은 것은?

- ① 일정 관리 - 작업 순서, 작업 기간 산정 등
② 비용 관리 - 비용 산정, 비용 통제, 자원 산정 등
③ 인력 관리 - 팀 편성, 조직 정의, 팀 개발
④ 위험 관리 - 위험 식별, 위험 평가, 위험 통제 등

94. 다음 중 스트림 암호화 방식에 해당하는 암호화 알고리즘을 모두 고른 것은?

- ㉠ LFSR ㉡ DES ㉢ SEED
㉣ RC4 ㉤ ARIA ㉥ AES

- ① ㉠, ㉢ ② ㉡, ㉢
③ ㉡, ㉣, ㉥ ④ ㉠, ㉢, ㉥

95. 다음 빈 칸에 알맞은 기술은?

()은/는 필요한 모든 것에 RFID 태그를 부착하여 수집한 정보를 무선으로 수집할 수 있도록 구성한 네트워크이다.

- ① GIS ② SON
③ USN ④ UWB

96. 자기 복제 능력은 없지만 정상적인 기능을 하는 프로그램으로 위장하여 프로그램 내에 숨어 있다가 해당 프로그램이 동작할 때 활성화되어 부작용을 일으키는 것은?

- ① 웜 ② 좀비 PC
③ 트로이 목마 ④ 랜섬웨어

97. 하향식 비용 산정 기법에 대한 설명으로 옳지 않은 것은?

- ① 과거의 유사한 경험을 바탕으로 비용을 산정하는 비과학적인 방법이다.
② 프로젝트의 전체 비용을 산정한 후 각 작업별로 비용을 세분화한다.
③ 개인적이고 주관적일 수 있으며, 이를 보완하기 위해 많은 전문가로 구성하기도 한다.
④ 하향식 비용 산정 기법에는 델파이 기법, LOC 기법 등이 있다.

98. 소프트웨어 개발 프로젝트에 참여하는 관련자 중 시스템에 사용되는 모든 리소스 정의 및 각 리소스별로 적절한 보안 요구사항을 적용하는 직무는?

- ① 프로젝트 관리자 ② 요구사항 분석가
③ 아키텍트 ④ 설계자

99. 참조 모니터(Reference Monitor)에 대한 설명으로 옳지 않은 것은?

- ① 보호대상의 객체에 대한 접근통제를 수행하는 추상머신이다.
② 참조 모니터를 실제로 구현한 것이 보안 커널이다.
③ 데이터베이스를 참조하여 객체에 대한 접근 허가 여부를 결정한다.
④ 참조 모니터와 보안 커널의 3가지 특징에는 격리성, 완전성, 원자성이 있다.

100. 소프트웨어의 구현 단계에서 발생할 수 있는 보안 취약점들을 최소화하기 위해 보안 요소들을 고려하며 코딩하는 것은?

- ① Secure SDLC ② Secure Coding
③ Secure Framework ④ Secure OS

정답 및 해설

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

1 문제의 지문은 인스펙션(Inspection)에 대한 설명입니다.

[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.

2 CASE는 상위, 하위, 통합 케이스로 분류됩니다.

[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 자동화해주는 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
- CASE의 분류
 - 상위(Upper) CASE : 소프트웨어 생명 주기의 전반부에서 사용되는 것으로, 문제를 기술(Description)하고 계획하며 요구 분석과 설계 단계를 지원하는 CASE
 - 하위(Lower) CASE : 소프트웨어 생명 주기의 후반부에서 사용되는 것으로 코드의 작성과 테스트, 문서화하는 과정을 지원하는 CASE

- 통합(Integrate) CASE : 소프트웨어 생명 주기에 포함되는 전체 과정을 지원하기 위한 CASE

3 자료 사전에서 ()는 자료의 생략 또는 생략 가능한 자료를 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

4 하나의 메시지에 대해 각각의 객체(클래스)가 가지고 있는 고유한 방법(특성)으로 응답할 수 있는 능력을 다형성(Polymorphism)이라고 합니다.

[병행학습]

객체지향 관련 용어

- 메소드(Method) : 객체의 상태를 참조하거나 변경하는 수단이 되는 것으로, 동작(Operation)이라고도 함
- 캡슐화(Encapsulation) : 데이터(속성)와 데이터를 처리하는 함수를 하나로 묶는 것
- 정보 은닉(Information Hiding) : 캡슐화에서 가장 중요한 개념으로, 다른 객체에게 자신의 정보를 숨기고 자신의 연산만을 통하여 접근을 허용하는 것
- 추상화(Abstraction) : 불필요한 부분을 생략하고 객체의 속성 중 가장 중요한 것에만 중점을 두어 개략화하는 것, 즉 모델화하는 것
- 상속성(Inheritance) : 이미 정의된 상위 클래스(부모 클래스)의 모든 속성과 연산을 하위 클래스가 물려받는 것
- 다형성(Polymorphism) : 메시지에 의해 객체(클래스)가 연산을 수행하게 될 때 하나의 메시지에 대해 각 객체(클래스)가 가지고 있는 고유한 방법으로 응답할 수 있는 능력

- 5 림바우의 분석 기법 중 지문에 제시된 내용은 객체 모델링의 특징입니다.

[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 연관 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.

- 6 GoF의 디자인 패턴은 수많은 디자인 패턴들 중 가장 일반적인 사례에 적용될 수 있는 패턴들을 분류하여 정리함으로써, 지금까지도 소프트웨어 공학이나 현업에서 가장 많이 사용되는 디자인 패턴입니다.

[병행학습]

디자인 패턴(Design Pattern)

- 각 모듈의 세분화된 역할이나 모듈들 간의 인터페이스와 같은 코드를 작성하는 수준의 세부적인 구현 방안을 설계할 때 참조할 수 있는 전형적인 해결 방식 또는 예제를 의미한다.
- 재사용할 수 있는 기본형 코드들이 포함되어 있다.
- ‘바퀴를 다시 발명하지 마라(Don't reinvent the wheel)’라는 말과 같이, 개발 과정 중에 문제가 발생하면 새로 해결책을 구상하는 것보다 문제에 해당하는 디자인 패턴을 참고하여 적용하는 것이 더 효율적이다.
- 한 패턴에 변형을 가하거나 특정 요구사항을 반영하면 유사한 형태의 다른 패턴으로 변화되는 특징이 있다.
- 1995년 GoF(Gang of Four)라고 불리는 에릭 감마(Erich Gamma), 리처드 헬름(Richard Helm), 랄프 존슨(Ralph Johnson), 존 블리시디스(John Vissides)가 처음으로 구체화 및 체계화하였다.
- GoF의 디자인 패턴은 수많은 디자인 패턴들 중 가장 일반적인 사례에 적용될 수 있는 패턴들을 분류하여 정리함으로써, 지금까지도 소프트웨어 공학이나 현업에서 가장 많이 사용되는 디자인 패턴이다.
- GoF의 디자인 패턴은 유형에 따라 생성 패턴 5개, 구조 패턴 7개, 행위 패턴 11개 총 23개의 패턴으로 구성된다.

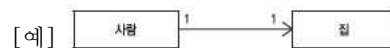
- 7 그림은 사물이 할 수 있거나 해야 하는 기능(행위)으로 서로를 그룹화 할 수 있는 관계를 표현하고 있습니다. 차, 비행기, 배는 모두 ‘탑승할 수 있는’ 기능으로 그룹화 할 수 있습니다.

[병행학습]

관계(Relationships)

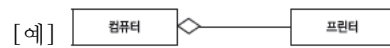
· 연관(Association) 관계

- 2개 이상의 사물이 서로 관련되어 있음



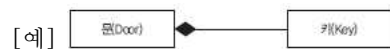
· 집합(Aggregation) 관계

- 하나의 사물이 다른 사물에 포함되어 있는 관계



· 포함(Composition) 관계

- 집합 관계의 특수한 형태로, 포함하는 사물의 변화가 포함되는 사물에게 영향을 미치는 관계



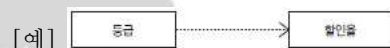
· 일반화(Generalization) 관계

- 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

- 실체화 관계는 사물이 할 수 있거나 해야 하는 기능(행위, 인터페이스)으로 서로를 그룹화 할 수 있는 관계



- 8 클래스 다이어그램은 클래스와 클래스가 가지는 속성, 클래스 사이의 관계를 표현하는 다이어그램이며, ①번은 커뮤니케이션 다이어그램에 대한 설명입니다.

[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

- 9 소프트웨어가 다른 환경에서도 얼마나 쉽게 적용할 수 있는지를 나타내는 품질특성을 이식성 또는 호환성이라고 합니다.

[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)

- 10 코드화 대상 항목의 명칭이나 약호와 관계있는 숫자나 문자, 기호를 이용하여 코드를 부여하는 방법을 연상 코드라고 합니다.

[병행학습]

코드(Code)

- 컴퓨터를 이용하여 자료를 처리하는 과정에서 분류·조합 및 집계를 용이하게 하고, 특정 자료의 추출을 쉽게 하기 위해서 사용하는 기호이다.
- 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

- 11 사용자 인터페이스(UI)는 오류가 발생했을 때 사용자가 쉽게 인지할 수 있도록 설계해야 합니다.

[병행학습]

사용자 인터페이스(UI; User Interface)의 설계 지침

사용자 중심	사용자가 쉽게 이해하고 편리하게 사용할 수 있는 환경을 제공하며, 실사용자에 대한 이해가 바탕이 되어야 함
일관성	버튼이나 조작 방법 등을 일관성 있게 제공하므로 사용자가 쉽게 기억하고 습득할 수 있게 설계해야 함
단순성	조작 방법을 단순화시켜 인지적 부담을 감소시켜야 함
결과 예측 가능	작동시킬 기능만 보고도 결과를 미리 예측할 수 있게 설계해야 함
가시성	메인 화면에 주요 기능을 노출시켜 최대한 조작이 쉽도록 설계해야 함
표준화	기능 구조와 디자인을 표준화하여 한번 학습한 이후에는 쉽게 사용할 수 있도록 설계해야 함
접근성	사용자의 연령, 성별, 인종 등 다양한 계층이 사용할 있도록 설계해야 함
명확성	사용자가 개념적으로 쉽게 인지할 수 있도록 설계해야 함
오류 발생 해결	오류가 발생하면 사용자가 쉽게 인지할 수 있도록 설계해야 함

- 12 문제의 지문에 제시된 내용은 싱글톤(Singleton)의 특징입니다.

[병행학습]

디자인 패턴 - 생성 패턴(Creational Pattern)

· 추상 팩토리(Abstract Factory)

- 구체적인 클래스에 의존하지 않고, 인터페이스를 통해서 서로 연관·의존하는 객체들의 그룹으로 생성하여 추상적으로 표현한다.
- 연관된 서브 클래스를 묶어 한 번에 교체하는 것이 가능하다.

· 빌더(Builder)

- 작게 분리된 인스턴스를 건축 하듯이 조합하여 객체를 생성한다.
- 객체의 생성 과정과 표현 방법을 분리하고 있어, 동일한 객체 생성에서도 서로 다른 결과를 만들어 낼 수 있다.

· 팩토리 메소드(Factory Method)

- 객체 생성을 서브 클래스에서 처리하도록 분리하여 캡슐화한 패턴이다.
- 상위 클래스에서 인터페이스만 정의하고 실제 생성은 서브 클래스가 담당한다.

· 프로토타입(Prototype)

- 원본 객체를 복제하는 방법으로 객체를 생성하는 패턴이다.
- 일반적인 방법으로 객체를 생성하며, 비용이 큰 경우 주로 이용한다.

· 싱글톤(Singleton)

- 하나의 객체를 생성하면 생성된 객체를 어디서든 참조할

- 수 있지만, 여러 프로세스가 동시에 참조할 수는 없다.
- 클래스 내에서 인스턴스가 하나뿐임을 보장하며, 불필요한 메모리 낭비를 최소화 할 수 있다.

- 13 기능 점수(Function Point)는 비용 산정 모형의 하나로, 객체 지향 분석의 방법론이 아닙니다.

[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 객체의 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.

- 14 행동 사물(Behavioral Things)은 시간과 공간에 따른 요소들의 행위를 표현하는 것입니다. ②번은 주해 사물(Annotation Things)에 대한 설명입니다.

[병행학습]

UML - 사물(Things)

- 모델을 구성하는 가장 중요한 기본 요소로, 다이어그램 안에서 관계가 형성될 수 있는 대상들을 말한다.
- **구조 사물(Structural Things)**
 - 시스템의 개념적, 물리적 요소를 표현
 - 클래스(Class), 유스케이스(Use Case), 컴포넌트(Component), 노트(Node) 등
- **행동 사물(Behavioral Things)**
 - 시간과 공간에 따른 요소들의 행위를 표현
 - 상호작용(Interaction), 상태 머신(State Machine) 등
- **그룹 사물(Grouping Things)**
 - 요소들을 그룹으로 묶어서 표현
 - 패키지(Package)
- **주해 사물(Annotation Things)**
 - 추가적인 설명이나 제약조건 등을 표현
 - 노트(Note)

- 15 객체 지향 미들웨어로 코바(CORBA) 표준 스펙을 구현한 미들웨어는 객체 요청 브로커(ORB, Object Request Broker)입니다.

[병행학습]

미들웨어(Middleware)의 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
- **TP-Monitor(Transaction Processing Monitor)** : 항공기

나 철도 예약 업무 등과 같은 온라인 트랜잭션 업무에서 트랜잭션을 처리 및 감시하는 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

- 16 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 확실히 매듭짓고 그 결과를 철저히 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 건본품(Prototype)을 만들어 최종 결과물을 예측하는 모형
- **나선형 모형(Spiral Model, 점진적 모형)** : 폭포수 모형과 프로토타입 모형의 장점에 위험 분석 기능을 추가한 모형으로, 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

- 17 문제의 지문은 HIPO 차트에 대한 설명입니다.

[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 표현한다.
- 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

- 18 기능 요구사항과 비기능 요구사항을 분류하는 것은 요구사항 분류(Requirement Classification) 기법입니다.

[병행학습]

개념 모델링(Conceptual Modeling)

- 요구사항을 보다 쉽게 이해할 수 있도록 현실 세계의 상황을 단순화하여 개념적으로 표현한 것을 모델이라고 하며, 이러한 모델을 만드는 과정을 모델링이라고 한다.
- 모델은 문제가 발생하는 상황을 쉽게 이해시키고 해결책을 설명할 수 있으므로 실세계 문제에 대한 모델링은 소프트웨어

4회 대비

어 요구사항 분석의 핵심이다.

- 개념 모델은 문제의 주제인 개체(Entity)들과 그들 간의 관계 및 종속성을 반영한다.
- 요구사항을 이해하는 이해관계자별로 관점이 다양하므로 그에 맞게 개념 모델도 다양하게 표현되어야 한다.
- 개념 모델의 종류에는 유스케이스 다이어그램(Use Case Diagram), 데이터 흐름 모델(Data Flow Model), 상태 모델(State Model), 목표기반 모델(Goal-Based Model), 사용자 인터랙션(User Interactions), 객체 모델(Object Model), 데이터 모델(Data Model) 등이 있다.
- 모델링 표기는 주로 UML(Unified Modeling Language)을 사용한다.

- 19 문제에 제시된 내용은 장애 허용 시스템(FTS, Fault Tolerance System)의 개념입니다. 장애 허용 시스템은 마스터-슬레이브 패턴을 주로 사용합니다.

[병행학습]

마스터-슬레이브 패턴(Master-Slave Pattern)

- 마스터 컴포넌트에서 슬레이브 컴포넌트로 작업을 분할한 후, 슬레이브 컴포넌트에서 처리된 결과물을 다시 돌려받는 방식으로 작업을 수행하는 패턴이다.
- 마스터 컴포넌트는 모든 작업의 주체이고, 슬레이브 컴포넌트는 마스터 컴포넌트의 지시에 따라 작업을 수행하여 결과를 반환한다.
- 장애 허용 시스템과 병렬 컴퓨팅 시스템에서 주로 활용된다.

- 20 문제에 제시된 내용은 단계적 분해(Stepwise Refinement)의 개념입니다.

[병행학습]

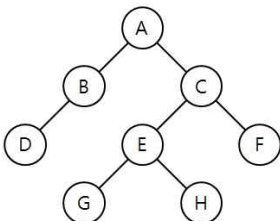
아키텍처 설계의 기본 원리

- **모듈화(Modularity)** : 소프트웨어의 성능을 향상시키거나 시스템의 수정 및 재사용, 유지 관리 등이 용이하도록 시스템의 기능들을 모듈 단위로 나누는 것
- **추상화(Abstraction)** : 문제의 전체적이고 포괄적인 개념을 설계한 후 차례로 세분화하여 구체화시켜 나가는 것
- **단계적 분해(Stepwise Refinement)** : 문제를 상위의 중요 개념으로부터 하위의 개념으로 구체화시키는 분할 기법
- **정보 은닉(Information Hiding)** : 한 모듈 내부에 포함된 절차와 자료들의 정보가 감추어져 다른 모듈이 접근하거나 변경하지 못하도록 하는 기법

- 21 터미널 노드(단말 노드)란 자식이 하나도 없는 노드를 말합니다. 제시된 그림에서 자식이 없는 터미널 노드는 D, F, G, H로 총 4개입니다.

[병행학습]

트리(Tree) 관련 용어



- **노드(Node)** : 트리의 기본 요소로서 자료 항목과 다른 항목에 대한 가지(Branch)를 합친 것
예) A, B, C, D, E, F, G, H
- **디그리(Degree, 차수)** : 각 노드에서 뻗어 나온 가지의 수
예) A = 2, B = 1, C = 2, D = 0, E = 2, FGH = 0
- **단말 노드(Terminal Node)=잎 노드(Leaf Node)** : 자식이

하나도 없는 노드, 즉 디그리가 0인 노드

예) D, F, G, H

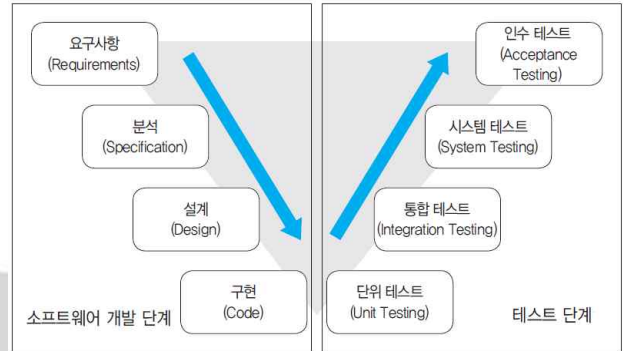
- **트리의 디그리** : 노드들의 디그리 중에서 가장 많은 수
예) 노드 A, C, E의 디그리 2가 트리의 디그리이다.

- 22 문제에 제시된 내용은 단위 테스트의 개념입니다.

[병행학습]

V-모델

애플리케이션 테스트와 소프트웨어 개발 단계를 연결하여 표현한 것이다.



- 23 문제에 제시된 내용은 적응 보수(Adaptive Maintenance)의 개념입니다.

[병행학습]

유지보수의 유형

- **수정(Corrective) 보수** = **수리·교정·정정·하자 보수** : 시스템을 운영하면서 검사 단계에서 발견하지 못한 오류를 찾아 수정하는 활동
- **적응(Adaptive) 보수** = **환경 적응, 조정 보수** : 소프트웨어의 수명 기간 중에 발생하는 환경의 변화(하드웨어, 운영체제 등)를 기존의 소프트웨어에 반영하기 위하여 수행하는 활동
- **완전화(Perfective) 보수** = **기능 개선, 기능 보수** : 소프트웨어의 본래 기능에 새로운 기능을 추가하거나 성능을 개선하기 위해 소프트웨어를 확장시키는 활동으로, 유지보수 활동 중 가장 큰 업무 및 비용을 차지하는 활동임
- **예방(Preventive) 보수** : 미래에 유지보수를 용이하게 하거나 기능을 향상시키기 위해 소프트웨어를 변경하는 활동

- 24 문제의 지문으로 제시된 내용은 재사용(Reuse)에 대한 설명입니다.

[병행학습]

재사용(Reuse)

- 비용과 개발 시간을 절약하기 위해 이미 개발된 기능들을 파악하고 재구성하여 새로운 시스템 또는 기능 개발에 사용하기 적합하도록 최적화 시키는 작업이다.
- 재사용을 위해서는 누구나 이해할 수 있고 사용이 가능하도록 사용법을 공개해야 한다.
- 재사용되는 대상은 외부 모듈과의 결합도는 낮고, 응집도는 높아야 한다.
- 재사용 규모에 따른 분류
 - 함수와 객체 : 클래스나 메소드 단위의 소스 코드를 재사용
 - 컴포넌트 : 컴포넌트 자체에 대한 수정 없이 인터페이스를 통해 통신하는 방식으로 재사용
 - 애플리케이션 : 공통된 기능들을 제공하는 애플리케이션을 공유하는 방식으로 재사용

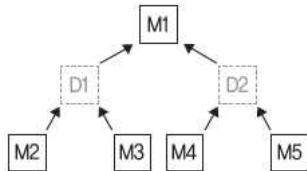
- 25 하향식 통합 테스트에서는 테스트 스텝을, 상향식 통합 테스트에서는 테스트 드라이버를 사용합니다.

[병행학습]

테스트 드라이버와 테스트 스텝의 차이점

• 드라이버(Driver)

- 필요시기 : 상위 모듈 없이 하위 모듈이 있는 경우 하위 모듈 구동
- 테스트 방식 : 상향식(Bottom Up) 테스트
- 개념도

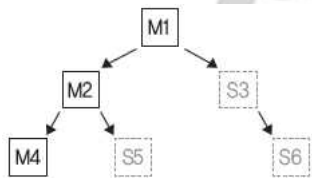


- 차이점

- ▶ 이미 존재하는 하위 모듈과 존재하지 않는 상위 모듈 간의 인터페이스 역할을 한다.
- ▶ 소프트웨어 개발이 완료되면 드라이버는 본래의 모듈로 교체된다.

• 스텝(Stub)

- 필요시기 : 상위 모듈은 있지만 하위 모듈이 없는 경우 하위 모듈 대체
- 테스트 방식 : 하향식(Top-Down) 테스트
- 개념도



- 차이점

- ▶ 일시적으로 필요한 조건만을 가지고 임시로 제공되는 가짜 모듈의 역할을 한다.
- ▶ 시험용 모듈이기 때문에 일반적으로 드라이버보다 작성하기 쉽다.

※ 드라이버와 스텝의 공통점 : 소프트웨어 개발과 테스트를 병행할 경우 이용

- 26 ①번은 기본 테이블, ③번은 파티셔닝 테이블, ④번은 외부 테이블에 대한 설명입니다.

[병행학습]

클러스터드 인덱스 테이블(Clustered Index Table)

- 클러스터드 인덱스 테이블은 기본키(Primary Key)나 인덱스키의 순서에 따라 데이터가 저장되는 테이블이다.
- 클러스터 인덱스 테이블은 일반적인 인덱스를 사용하는 테이블에 비해 접근 경로가 단축된다.

- 27 라이선스 정보는 패키징 수행 시 클리어링 하우스에 등록됩니다.

[병행학습]

디지털 저작권 관리(DRM; Digital Right Management)

- 저작권자가 배포한 디지털 콘텐츠가 저작권자가 의도한 용도로만 사용되도록 디지털 콘텐츠의 생성, 유통, 이용까지의 전 과정에 걸쳐 사용되는 디지털 콘텐츠 관리 및 보호 기술이다.
- 원본 콘텐츠가 아날로그인 경우에는 디지털로 변환한 후 패키저(Packager)에 의해 DRM 패키징을 수행한다.
- 콘텐츠의 크기에 따라 음원이나 문서와 같이 크기가 작은 경우에는 사용자가 콘텐츠를 요청하는 시점에서 실시간으

로 패키징을 수행하고, 크기가 큰 경우에는 미리 패키징을 수행한 후 배포한다.

- 패키징을 수행하면 콘텐츠에는 암호화된 저작권자의 전자서명이 포함되고 저작권자가 설정한 라이선스 정보가 클리어링 하우스(Clearing House)에 등록된다.
- 사용자가 콘텐츠를 사용하기 위해서는 클리어링 하우스에 등록된 라이선스 정보를 통해 사용자 인증과 콘텐츠 사용 권한 소유 여부를 확인받아야 한다.
- 종량제 방식을 적용한 소프트웨어의 경우 클리어링 하우스를 통해 서비스의 실제 사용량을 측정하여 이용한 만큼의 요금을 부과한다.

- 28 소프트웨어는 커스터마이징(맞춤형) 형태로 패키징하는 것이 아니라 다양한 환경에서 소프트웨어를 손쉽게 사용할 수 있도록 가장 일반적인 배포 형태로 패키징해야 합니다.

[병행학습]

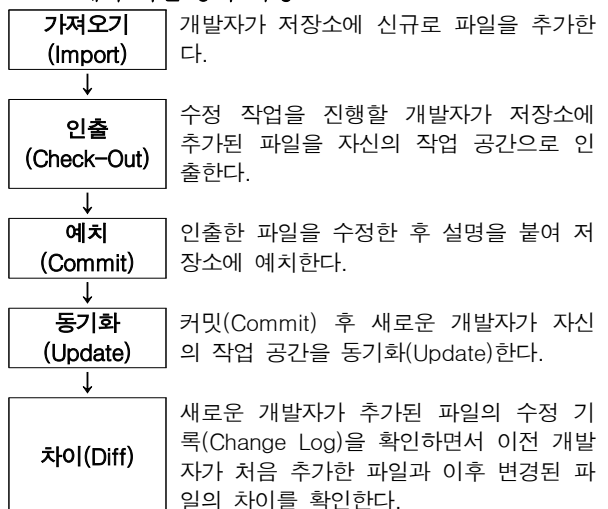
소프트웨어 패키징

- 모듈별로 생성한 실행 파일들을 묶어 배포용 설치 파일을 만드는 것을 말한다.
- 개발자가 아니라 사용자를 중심으로 진행한다.
- 소스 코드는 향후 관리를 고려하여 모듈화하여 패키징한다.
- 사용자가 소프트웨어를 사용하게 될 환경을 이해하여, 다양한 환경에서 소프트웨어를 손쉽게 사용할 수 있도록 일반적인 배포 형태로 패키징한다.
- 사용자를 중심으로 진행되는 작업이므로 사용자의 편의성 및 실행 환경을 우선적으로 고려해야 한다.
- 사용자의 시스템 환경, 즉 운영체제(OS), CPU, 메모리 등에 필요한 최소 환경을 정의한다.
- UI(User Interface)는 사용자가 눈으로 직접 확인할 수 있도록 시각적인 자료와 함께 제공하고 매뉴얼과 일치시켜 패키징한다.
- 소프트웨어는 단순히 패키징하여 배포하는 것으로 끝나는 것이 아니라 하드웨어와 함께 관리될 수 있도록 Managed Service 형태로 제공하는 것이 좋다.
- 고객의 편의성을 고려한 안정적인 배포가 중요하다.
- 다양한 사용자의 요구사항을 반영할 수 있도록 패키징의 변경 및 개선에 대한 관리를 항상 고려한다.

- 29 소프트웨어 버전 등록 순서는 가져오기(Import), 인출(Check-Out), 예치(Commit), 동기화(Update), 차이(Diff) 순입니다.

[병행학습]

소프트웨어 버전 등록 과정



- 30 빅오 표기법 중 $O(\log_2 N)$ 의 시간 복잡도를 가진 알고리즘에는 이진 트리와 이진 검색이 있습니다.

[병행학습]

빅오 표기법(Big-O Notation)

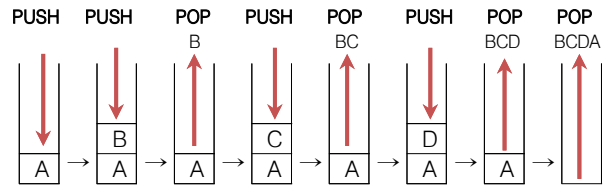
- 알고리즘의 실행시간이 최악일 때를 표기하는 방법으로, 신뢰성이 떨어지는 오메가 표기법이나 평가하기 까다로운 세타 표기법에 비해 성능을 예측하기 용이하여 주로 사용되는 표기법이다.
- $O(1)$
 - 입력값(n)에 관계 없이 일정하게 문제 해결에 하나의 단계만을 거친다.
 - [예] 스택의 삽입(Push), 삭제(Pop)
- $O(\log_2 n)$
 - 문제 해결에 필요한 단계가 입력값(n) 또는 조건에 의해 감소한다.
 - [예] 이진 트리(Binary Tree), 이진 검색(Binary Search)
- $O(n)$
 - 문제 해결에 필요한 단계가 입력값(n)과 1:1의 관계를 가진다.
 - [예] for문
- $O(n \log_2 n)$
 - 문제 해결에 필요한 단계가 $n(\log_2 n)$ 번만큼 수행된다.
 - [예] 힙 정렬(Heap Sort), 2-Way 합병 정렬(Merge Sort)
- $O(n^2)$
 - 문제 해결에 필요한 단계가 입력값(n)의 제곱만큼 수행된다.
 - [예] 삽입 정렬(Insertion Sort), 셸 정렬(Shell Sort), 선택 정렬(Selection Sort), 버블 정렬(Bubble Sort), 퀵 정렬(Quick Sort)
- $O(2^n)$
 - 문제 해결에 필요한 단계가 2의 입력값(n) 제곱만큼 수행된다.

- 31 문제의 설명은 Message Bus(ESB) 방식에 대한 설명입니다.

[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 집점인 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 방식으로, 그룹 내에서는 Hub & Spoke 방식을, 그룹 간에는 Message Bus 방식을 사용한다.
- 32 PUSH는 스택에 자료를 입력하는 명령이고, POP은 스택에서 자료를 출력하는 명령입니다. 문제에 제시된 대로 PUSH와 POP을 수행하면 다음의 순서로 입출력이 발생합니다.



- 33 형상 관리는 수정 및 추적이 용이하도록 계층(Tree) 구조로 구분합니다.

[병행학습]

형상 관리의 목적

형상 관리는 소프트웨어 개발의 전체 비용을 줄이고, 개발 과정의 여러 방해 요인이 최소화되도록 보증하는 것을 목적으로 한다.

형상 관리 기능

- 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

- 34 삽입 정렬은 두 번째 자료부터 시작하여 그 앞(왼쪽)의 자료들과 비교하여 삽입할 위치를 지정한 후 자료를 뒤로 옮기고 지정한 자리에 자료를 삽입하여 정렬하는 알고리즘입니다. 즉 두 번째 자료는 첫 번째 자료, 세 번째 자료는 두 번째와 첫 번째 자료, 네 번째 자료는 세 번째, 두 번째, 첫 번째 자료와 비교한 후 자료가 삽입될 위치를 찾습니다.

초기 자료 :

5	4	3	2	1
---	---	---	---	---

① 1회전 :

5	4	3	2	1
---	---	---	---	---

 →

4	5	3	2	1
---	---	---	---	---

두 번째 값 4를 첫 번째 값과 비교하여 첫 번째 자리에 삽입하고 5를 한 칸 뒤로 이동시킵니다.

② 2회전 :

4	5	3	2	1
---	---	---	---	---

 →

3	4	5	2	1
---	---	---	---	---

세 번째 값 3을 첫 번째, 두 번째 값과 비교하여 4자리에 삽입하고 4, 5는 한 칸씩 뒤로 이동시킵니다.

③ 3회전 :

3	4	5	2	1
---	---	---	---	---

 →

2	3	4	5	1
---	---	---	---	---

네 번째 값 2를 첫 번째, 두 번째, 세 번째 값과 비교하여 3자리에 삽입하고 3, 4, 5는 한 칸씩 뒤로 이동시킵니다.

④ 4회전 :

2	3	4	5	1
---	---	---	---	---

 →

1	2	3	4	5
---	---	---	---	---

다섯 번째 값 1을 처음부터 비교하여 2자리에 삽입하고 나머지를 한 칸씩 뒤로 이동시킵니다.

- 35 전위식은 연산자를 해당 피연산자 두 개의 앞(왼쪽)으로 이동시킨 것입니다. 그러므로 연산자와 피연산자 2개를 묶은 후 연산자를 피연산자 두 개의 뒤(오른쪽)로 옮겨 놓으면 후위식이 됩니다.

- ① 왼쪽으로 인접한 연산자 1개와 피연산자 2개를 묶습니다.

- / * A + B C D E

↓
(- (/ (* A (+ B C)) D) E)

4회 대비

- ② 연산자를 피연산자 두 개의 뒤로 옮깁니다.

$$(-((/*A(+ B C)) D) E)$$

$$\downarrow$$

$$(((A(B C +) *) D /) E -)$$

- ③ 괄호를 제거합니다.

$A B C + * D / E -$

- 36 기초 경로 검사, 데이터 흐름 검사, 조건 검사는 화이트 박스 검사 기법입니다.

[병행학습]

화이트박스 테스트(White Box Test)

- 모듈의 원시 코드를 오픈시킨 상태에서 원시 코드의 논리적인 모든 경로를 테스트하여 테스트 케이스를 설계하는 방법이다.
- 설계된 절차에 초점을 둔 구조적 테스트로 프로시저 설계의 제어 구조를 사용하여 테스트 케이스를 설계하며, 테스트 과정의 초기에 적용된다.
- 모듈 안의 작동을 직접 관찰한다.
- 원시 코드(모듈)의 모든 문장을 한 번 이상 실행함으로써 수행된다.
- 프로그램의 제어 구조에 따라 선택, 반복 등의 분기점 부분들을 수행함으로써 논리적 경로를 제어한다.
- 종류 : 기초 경로 검사, 제어 구조 검사(조건 검사, 루프 검사, 데이터 흐름 검사) 등

블랙박스 테스트(Black Box Test)

- 소프트웨어가 수행할 특정 기능을 알기 위해서 각 기능이 완전히 작동되는 것을 입증하는 테스트로, 기능 테스트라고도 한다.
- 사용자의 요구사항 명세를 보면서 테스트하는 것으로, 주로 구현된 기능을 테스트한다.
- 소프트웨어 인터페이스에서 실시되는 테스트이다.
- 부정확하거나 누락된 기능, 인터페이스 오류, 자료 구조나 외부 데이터베이스 접근에 따른 오류, 행위나 성능 오류, 초기화와 종료 오류 등을 발견하기 위해 사용되며, 테스트 과정의 후반부에 적용된다.
- 종류 : 동치 분할 검사, 경계값 분석, 원인-효과 그래프 검사, 오류 예측 검사, 비교 검사 등

- 37 문제에 제시된 내용은 단순성의 개념입니다.

[병행학습]

클린 코드 작성 원칙

- 가독성
 - 누구든지 코드를 쉽게 읽을 수 있도록 작성한다.
 - 코드 작성 시 이해하기 쉬운 용어를 사용하거나 들여쓰기 기능 등을 사용한다.
- 단순성
 - 코드를 간단하게 작성한다.
 - 한 번에 한 가지를 처리하도록 코드를 작성하고 클래스/메소드/함수 등을 최소 단위로 분리한다.
- 의존성 배제
 - 코드가 다른 모듈에 미치는 영향을 최소화한다.
 - 코드 변경 시 다른 부분에 영향이 없도록 작성한다.
- 중복성 최소화
 - 코드의 중복을 최소화한다.
 - 중복된 코드는 삭제하고 공통된 코드를 사용한다.
- 추상화
 - 상위 클래스/메소드/함수에서는 간략하게 애플리케이션

의 특성을 나타내고, 상세 내용은 하위 클래스/메소드/함수에서 구현한다.

- 38 ISO/IEC 9126을 개정한 표준은 ISO/IEC 25010입니다.

[병행학습]

소프트웨어 품질 표준

- ISO/IEC 9126 : 소프트웨어의 품질 특성과 평가를 위한 표준 지침
- ISO/IEC 12119 : ISO/IEC 9126을 준수한 품질 표준으로 테스트 절차도 규정함
- ISO/IEC 14598 : 소프트웨어 품질의 측정, 평가에 필요 절차를 규정한 표준으로, 개발자, 구매자, 평가자 별로 제품 평가 활동을 규정함
- ISO/IEC 25010 : ISO/IEC 9126을 개정하여 만든 소프트웨어 제품에 대한 국제 표준으로 호환성과 보안성이 강화됨

- 39 FitNesse는 인터페이스 구현 검증 도구로, 웹 기반 테스트케이스를 지원하는 테스트 프레임워크입니다.

[병행학습]

APM(Application Performance Management/Monitoring)

- 애플리케이션의 성능 관리를 위해 접속자, 자원 현황, 트랜잭션 수행 내역, 장애 진단 등 다양한 모니터링 기능을 제공하는 도구를 의미합니다.
- APM은 리소스 방식과 엔드투엔드(End-to-End)의 두 가지 유형이 있습니다.
 - 리소스 방식 : Nagios, Zabbix, Cacti 등
 - 엔드투엔드 방식 : VisualVM, 제니퍼, 스카우터 등

- 40 애플리케이션 코드는 애플리케이션 영역에 해당하는 인터페이스 보안 사항입니다.

[병행학습]

인터페이스 보안 기능

· 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 적용함

· 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.

- 41 속성(열)의 수는 디그리(Degree)라고 하고, 튜플(행)의 수를 카디널리티(Cardinality)라고 합니다.

[병행학습]

릴레이션의 구성 요소

- 릴레이션(Relation) : 데이터들을 표(Table)의 형태로 표현한 것으로, 구조를 나타내는 릴레이션 스키마와 실제 값들인 릴레이션 인스턴스로 구성됨
- 인스턴스(Instance) : 데이터 개체를 구성하고 있는 속성들에 데이터 타입이 정의되어 구체적인 데이터 값을 갖고 있는 것을 말함
- 튜플(Tuple)
 - 튜플은 릴레이션을 구성하는 각각의 행을 말한다.

4회 대비

- 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 검사하는 데에도 이용된다.
예) 성별 애트리뷰트의 도메인은 '남'과 '여'로, 그 외의 값은 입력될 수 없다.

- 42 병행제어(Concurrency Control)란 다중 프로그램의 이점을 활용하여 동시에 여러 개의 트랜잭션들이 병행수행 될 때, 동시에 실행되는 트랜잭션들이 데이터베이스의 일관성을 파괴하지 않도록 트랜잭션 간의 상호작용을 제어하는 것으로, ①, ②, ③번의 목적과 데이터베이스의 공유도를 최대화하기 위해 사용합니다.

- 43 AVG는 AVG(속성명)의 형태로 사용해야 하며, 그룹함수이기 때문에 GROUP BY절과 함께 사용되어야 합니다.

- ① SELECT 성명, AVG(점수)
- ② FROM 성적
- ③ GROUP BY 성명;

- ① '성명'과 '점수'의 평균을 표시한다.
- ② <성적> 테이블을 대상으로 검색한다.
- ③ '성명'을 기준으로 그룹을 지정한다.

- 44 관계대수는 원하는 릴레이션을 정의하는 방법을 제공하는 것은 맞지만 절차적인 특징을 가지고 있습니다. 원하는 정보가 무엇이라는 것만 정의하는 비절차적인 특징을 가지는 것은 관계해석입니다.

[전문가의 조언]

관계해석

- 코드(E. F. Codd)가 수학의 Predicate Calculus(술어 해석)에 기반을 두고 관계 데이터베이스를 위해 제안했다.
- 관계해석은 원하는 정보가 무엇이라는 것만 정의하는 비절차적 특징을 지닌다.
- 원하는 정보를 정의할 때는 계산 수식을 사용한다.
- 튜플 관계해석과 도메인 관계해석이 있다.
- 기본적으로 관계해석과 관계대수는 관계 데이터베이스를 처리하는 기능과 능력 면에서 동등하다.
- 질의어로 표현한다.

- 45 정규화 과정 중 결정자이면서 부분적 함수 종속성을 제거하는 과정은 1NF(제1정규형)에서 2NF(제2정규형)으로 정규화하는 과정입니다.

[병행학습]

정규화 과정

비정규 릴레이션

↓ 도메인이 원자값

1NF

↓ 부분적 함수 종속 제거

2NF

↓ 이행적 함수 종속 제거

3NF

↓ 결정자이면서 후보키가 아닌 것 제거

BCNF

↓ 다치 종속 제거

4NF

↓ 조인 종속성 이용

5NF

정규화 단계 암기 요령

정규화라는 출소자가 말했다.

두부이겨다줘 ≡ 도부이결다조

도메인이 원자값

부분적 함수 종속 제거

이행적 함수 종속 제거

결정자이면서 후보키가 아닌 것 제거

다치 종속 제거

조인 종속성 이용

- 46 SELECT, UPDATE, INSERT는 DML이고, GRANT는 DCL입니다.

[병행학습]

데이터베이스 언어

- DDL(데이터 정의어)

- DDL(Data Define Language)은 SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의하거나 변경 또는 삭제할 때 사용하는 언어이다.
- 데이터베이스 관리자나 데이터베이스 설계자가 사용한다.
- 명령어 : CREATE, ALTER, DROP

- DML(데이터 조작어)

- DML(Data Manipulation Language)은 데이터베이스 사용자가 응용 프로그램이나 질의어를 통하여 저장된 데이터를 실질적으로 처리하는 데 사용되는 언어이다.
- 데이터베이스 사용자와 데이터베이스 관리 시스템 간의 인터페이스를 제공한다.
- 명령어 : SELECT, INSERT, DELETE, UPDATE

- DCL(데이터 제어어)

- DCL(Data Control Language)은 데이터의 보안, 무결성, 회복, 병행 수행 제어 등을 정의하는 데 사용되는 언어이다.
- 데이터베이스 관리자가 데이터 관리를 목적으로 사용한다.
- 명령어 : COMMIT, ROLLBACK, GRANT, REVOKE

- 47 참조되는 튜플이 반드시 참조 테이블에 존재해야 한다는 무결성 규칙은 참조 무결성 규칙입니다.

[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, 실제 무결성) : 개체 무결성은 기본 테이블의 기본키를 구성하는 어떤 속성도 Null 값이나 중복값을 가질 수 없다는 규정

4회 대비

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

48 보기별로 제시된 조건의 결과를 확인해 보세요.

① LIKE '%신%' : 공급자명에 '신'이 포함된 레코드

공급자번호	공급자명	위치
16	대신공업사	수원
70	신촌상사	서울

② LIKE '대%' : 공급자명이 '대'로 시작하는 레코드

공급자번호	공급자명	위치
16	대신공업사	수원

③ LIKE '%사' : 공급자명이 '사'로 끝나는 레코드

공급자번호	공급자명	위치
16	대신공업사	수원
27	삼진사	서울
39	삼양사	인천
62	진아공업사	대전
70	신촌상사	서울

④ LIKE '_사' : 공급자명이 '사'로 끝나고 두 글자인 레코드

공급자번호	공급자명	위치

[병행학습]

대표 문자

%	모든 문자를 대표
_	한 자리 문자를 대표
#	한 자리 숫자를 대표

49 시그마(σ)를 사용하는 관계대수 연산은 Select입니다.

[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 $R(X)$ 와 $S(Y)$ 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.

50 문제의 지문은 다치 종속(Multi Valued Dependency, 다가 종속)에 대한 설명입니다.

· **부분 함수적 종속** : 어떤 테이블 R 에서 속성 A 가 다른 속성 집합 B 전체에 대해 함수적 종속이면서 속성 집합 B 의 어떠한 진부분 집합에도 함수적 종속일 때, 속성 A 는 속성 집합 B 에 부분 함수적 종속이라고 함

· **완전 함수적 종속** : 어떤 테이블 R 에서 속성 A 가 다른 속성 집합 B 전체에 대해 함수적 종속이지만 속성 집합 B 의 어떠한 진부분 집합 C (즉, $C \subset B$)에는 함수적 종속이 아닐 때 속성 A 는 속성 집합 B 에 완전 함수적 종속이라고 함

· **조인 종속** : 어떤 릴레이션 R 의 속성에 대한 부분집합 A, B, C 가 있다고 할 때 만일 릴레이션 R 이 자신의 프로젝트션(Projection) A, B, C 를 모두 조인한 결과가 자신과 동일한 경우 릴레이션 R 은 조인 종속(A, B, C)을 만족한다고 함

51 테이블 통합은 두 개의 테이블이 조인(Join)되는 경우가 많아 하나의 테이블로 합쳐 사용하는 것이 성능 향상에 도움이 될 경우 사용합니다. 다른 서버에 저장된 테이블을 자주 이용해야 하는 경우에는 중복 테이블을 추가해야 합니다.

[병행학습]

반정규화(Denormalization)

- 시스템의 성능 향상, 개발 및 운영의 편의성 등을 위해 정규화된 데이터 모델을 통합, 중복, 분리하는 과정으로, 의도적으로 정규화 원칙을 위배하는 행위이다.
- 반정규화를 수행하면 시스템의 성능이 향상되고 관리 효율성은 증가하지만 데이터의 일관성 및 정합성이 저하될 수 있다.
- 과도한 반정규화는 오히려 성능을 저하시킬 수 있다.
- 반정규화를 위해서는 사전에 데이터의 일관성과 무결성을 우선으로 할지, 데이터베이스의 성능과 단순화를 우선으로 할지를 결정해야 한다.
- 반정규화 방법에는 테이블 통합, 테이블 분할, 중복 테이블 추가, 중복 속성 추가 등이 있다.

테이블 통합	두 개의 테이블이 조인(Join)되는 경우가 많아 하나의 테이블로 합쳐 사용하는 것이 성능 향상에 도움이 될 경우 수행함
테이블 분할	<ul style="list-style-type: none"> · 테이블을 수직 또는 수평으로 분할하는 것 · 수평 분할(Horizontal Partitioning) : 레코드(Record)를 기준으로 테이블을 분할 · 수직 분할(Vertical Partitioning) : 하나의 테이블에 속성이 너무 많을 경우 속성을 기준으로 테이블을 분할
중복 테이블 추가	여러 테이블에서 데이터를 추출해서 사용해야 하거나 다른 서버에 저장된 테이블을 이용해야 하는 경우 중복 테이블을 추가하여 작업의 효율성을 향상시킬 수 있음
중복 속성 추가	조인해서 데이터를 처리할 때 데이터를 조회하는 경로를 단축하기 위해 자주 사용하는 속성을 하나 더 추가하는 것

52 분산 데이터베이스는 DBMS가 수행할 기능이 복잡합니다.

[병행학습]

분산 데이터베이스의 장·단점

장점	<ul style="list-style-type: none"> 지역 자치성이 높음 효용성과 융통성이 높음 자료의 공유성이 향상됨 신뢰성 및 가용성이 높음 분산 제어가 가능함 시스템 성능이 향상됨 점증적 시스템 용량 확장이 용이함
단점	<ul style="list-style-type: none"> DBMS가 수행할 기능이 복잡함 처리 비용이 증가함 데이터베이스 설계가 어려움 잠재적 오류가 증가함 소프트웨어 개발 비용이 증가함

- 53 시스템 카탈로그는 데이터베이스에 변화가 있을 때마다 DBMS가 스스로 생성하고 유지합니다. 사용자는 일반 질의어를 이용해 시스템 카탈로그의 내용을 검색할 수는 있지만 직접 갱신할 수는 없습니다.

[병행학습]

시스템 카탈로그(System Catalog)

- 시스템 카탈로그는 시스템 그 자체에 관련이 있는 다양한 객체에 관한 정보를 포함하는 시스템 데이터베이스이다.
- 시스템 카탈로그는 데이터베이스에 포함되는 모든 데이터 객체에 대한 정의나 명세에 관한 정보를 유지·관리하는 시스템 테이블이다.
- 데이터 정의어의 결과로 구성되는 기본 테이블, 뷰, 인덱스, 패키지, 접근 권한 등의 데이터베이스 구조 및 통계 정보를 저장한다.
- 카탈로그들이 생성되면 자료 사전(Data Dictionary)에 저장되기 때문에 좁은 의미로는 카탈로그를 자료 사전이라고도 한다.
- 카탈로그에 저장된 정보를 메타 데이터(Meta-Data)라고 한다.
- 시스템 카탈로그 자체도 시스템 테이블로 구성되어 있어 일반 이용자도 SQL을 이용하여 내용을 검색해 볼 수 있다.
- INSERT, DELETE, UPDATE문으로 카탈로그를 갱신하는 것은 허용되지 않는다.
- 카탈로그는 DBMS가 스스로 생성하고, 유지한다.

- 54 교착상태(Deadlock)의 4가지 필요 충분 조건은 상호 배제(Mutual Exclusion), 점유와 대기(Hold and Wait), 비선점(Non-preemption), 환형 대기(Circular Wait)입니다.

[병행학습]

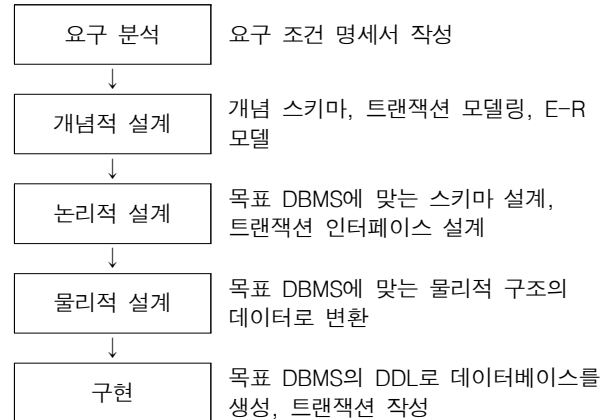
교착상태 발생의 필요 충분 조건

- 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
- 환형 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

- 55 트랜잭션 인터페이스를 설계하는 단계는 논리적 설계 단계입니다.

[병행학습]

데이터베이스의 설계 순서



- 56 데이터베이스를 트랜잭션의 첫 구문이 실행되기 전 상태로 되돌리는 트랜잭션 연산은 ROLLBACK입니다.

[병행학습]

데이터 제어어(DCL)

COMMIT	데이터베이스 조작 작업이 정상적으로 완료 되었음을 관리자에게 알려 줌
ROLLBACK	데이터베이스 조작 작업이 비정상적으로 종료되었을 때 원래의 상태로 복구함
GRANT	데이터베이스 사용자에게 사용 권한을 부여함
REVOKE	데이터베이스 사용자의 사용 권한을 취소함

- 57 ‘지점명’과 ‘판매액’을 출력하므로 “SELECT 지점명, 판매액”입니다.
- ‘<판매실적> 테이블을 참조하므로 “FROM 판매실적”입니다.
 - “서울” 지역에 한 한다고 했으니 “WHERE 도시 = ‘서울’”입니다.
 - ‘판매액’을 기준으로 내림차순으로 출력한다고 했으니 “ORDER BY 판매액 DESC”입니다.

[병행학습]

SELECT 문의 일반 형식

```
SELECT [테이블명.]속성명1, [테이블명.]속성명2, ...
FROM 테이블명1, 테이블명2, ...
[WHERE 조건]
[GROUP BY 속성명1, 속성명2, ...]
[HAVING 조건]
[ORDER BY 속성명 [ASC|DESC]];
```

· SELECT절

- 속성명** : 검색하여 불러올 속성(열) 또는 속성을 이용한 수식을 지정함

▶ 기본 테이블을 구성하는 모든 속성을 지정할 때는 ‘*’를 기술한다.

▶ 두 개 이상의 테이블을 대상으로 검색할 때는 ‘테이블명.속성명’으로 표현한다.

- FROM절** : 질의에 의해 검색될 데이터들을 포함하는 테이블명을 기술함

- WHERE절** : 검색할 조건을 기술함

- GROUP BY절** :

- 특정 속성을 기준으로 그룹화하여 검색할 때 그룹화한 속성을 지정한다.

- 일반적으로 GROUP BY절은 그룹 함수와 함께 사용된다.

- HAVING절** : GROUP BY와 함께 사용되며, 그룹에 대한

4회 대비

조건을 지정함

• ORDER BY절

- 특정 속성을 기준으로 정렬하여 검색할 때 사용한다.
- 속성명 : 정렬의 기준이 되는 속성명을 기술함
- [ASC|DESC] : 정렬 방식으로서 'ASC'는 오름차순, 'DESC'는 내림차순이며, 생략하면 오름차순으로 지정됨

58 문제의 지문에 제시된 내용은 독립성(Isolation)에 대한 설명입니다.

[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 트랜잭션의 결과는 시스템이 고장나더라도 영구적으로 반영되어야 함

59 트리와 그래프는 비선형 자료 구조이고, 배열, 스택, 큐는 선형 자료 구조입니다.

[병행학습]

자료 구조의 분류

선형 구조	스택, 큐, 데크, 배열, 연결 리스트
비선형 구조	트리, 그래프

60 SQL 구성 시 커서(Cursor)를 통해 처리하는 것은 정적 SQL입니다.

[병행학습]

동적 SQL(Dynamic SQL)

- 개발 언어에 삽입되는 SQL 코드를 문자열 변수에 넣어 처리하는 것으로, 조건에 따라 SQL 구문을 동적으로 변경하여 처리할 수 있다.
- 동적 SQL은 사용자로부터 SQL문의 일부 또는 전부를 입력받아 실행할 수 있다.
- 동적 SQL은 값이 입력되지 않을 경우 사용하는 NVL 함수를 사용할 필요가 없다.
- 동적 SQL은 응용 프로그램 수행 시 SQL이 변형될 수 있으므로 프리컴파일 할 때 구문 분석, 접근 권한 확인 등을 할 수 없다.
- 동적 SQL은 정적 SQL에 비해 속도가 느리지만, 상황에 따라 다양한 조건을 첨가하는 등 유연한 개발이 가능하다.

61 같은 타입의 데이터 집합을 저장할 때 사용하는 자료형은 배열 타입(Array Type)입니다.

[병행학습]

- 정수 타입(Integer Type) : 정수, 즉 소수점이 없는 숫자를 저장할 때 사용함
- 부동 소수점 타입(Floating Point Type) : 소수점 이하가 있는 실수를 저장할 때 사용함
- 문자 타입(Character Type)
 - 한 문자를 저장할 때 사용한다.
 - 작은따옴표(' ') 안에 표시한다.
- 문자열 타입(Character String Type)
 - 문자열을 저장할 때 사용한다.
 - 큰따옴표(" ") 안에 표시한다.
- 불린 타입(Boolean Type)
 - 조건의 참(True), 거짓(False) 여부를 판단하여 저장할 때 사용한다.
 - 기본값은 거짓(False)이다.
- 배열 타입(Array Type)
 - 같은 타입의 데이터 집합을 만들어 저장할 때 사용한다.
 - 데이터는 중괄호({ }) 안에 콤마(,)로 구분하여 값들을 나열한다.

62 셸(Shell)은 사용자의 명령어를 인식하여 프로그램을 호출하고 명령을 수행하는 명령어 해석기입니다. 프로세스 관리, 기억장치 관리, 입출력 관리 등의 기능을 수행하는 것은 커널(Kernel)입니다.

[병행학습]

커널(Kernel)

- UNIX의 가장 핵심적인 부분이다.
- 컴퓨터가 부팅될 때 주기억장치에 적재된 후 상주하면서 실행된다.
- 하드웨어를 보호하고, 프로그램과 하드웨어 간의 인터페이스 역할을 담당한다.
- 프로세스(CPU 스케줄링) 관리, 기억장치 관리, 파일 관리, 입·출력 관리, 프로세스간 통신, 데이터 전송 및 변환 등 여러 가지 기능을 수행한다.

셸(Shell)

- 사용자의 명령어를 인식하여 프로그램을 호출하고 명령을 수행하는 명령어 해석기이다.
- 시스템과 사용자 간의 인터페이스를 담당한다.
- DOS의 COMMAND.COM과 같은 기능을 수행한다.
- 주기억장치에 상주하지 않고, 명령어가 포함된 파일 형태로 존재하며 보조기억장치에서 교체 처리가 가능하다.
- 파이프라인 기능을 지원하고 입·출력 재지정을 통해 출력과 입력의 방향을 변경할 수 있다
- 공용 셸(Bourne Shell, C Shell, Korn Shell)이나 사용자 자신이 만든 셸을 사용할 수 있다.

63

```
main( ) {
  ① int sum = 0;
  ② for(int a = 20; a > 5; a -= 5)
    ③ sum += a;
  ④ printf("%d", sum);
}
```

- ① 정수형 변수 sum을 선언하고 0으로 초기화한다.
- ② 반복 변수 a가 20부터 5씩 감소하면서 5보다 큰 동안 ③번을 반복 수행한다.
- ③ sum에 a의 값을 누적한다.
- ④ sum의 값을 출력한다.
- ※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같습니다.

반복횟수	a	sum
		0
1	20	20
2	15	35
3	10	45
	5	

- 64 문제의 지문에 제시된 내용은 스탬프 결합도(Stamp Coupling)에 대한 설명입니다.

[병행학습]

결합도(Coupling)

- 모듈 간에 상호 의존하는 정도 또는 두 모듈 사이의 연관 관계를 의미한다.
- 다양한 결합으로 모듈을 구성할 수 있으나 결합도가 약할수록 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 return은 함수의 결과 값을 반환하는 예약어이므로 변수명으로 사용할 수 없습니다.

[병행학습]

C언어의 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.

66

```
public static void main(String[] args) {
  ❶ int a = 7, b = 11;
  ❷ int c = a > b ? a * b : b % a;
  ❸ System.out.println(c);
}
```

- 정수형 변수 a와 b를 선언하고 각각 7과 11로 초기화한다.
 - 정수형 변수 c를 선언하고, a가 b보다 크면 a*b의 값으로, 아니면 b%a의 값으로 초기화한다.
 - c의 값을 출력한다.
- ∴ 11을 7로 나눈 몫은 1이고, 나머지는 4입니다.

- 67 할당된 네트워크 주소를 다시 여러 개의 작은 네트워크로 나누어 사용하는 것을 서브네팅(Subnetting)이라고 합니다.

[병행학습]

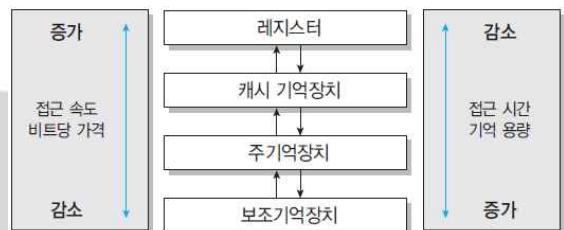
서브네팅(Subnetting)

- 할당된 네트워크 주소를 다시 여러 개의 작은 네트워크로 나누어 사용하는 것을 말한다.
- 4바이트의 IP 주소 중 네트워크 주소와 호스트 주소를 구분하기 위한 비트를 서브넷 마스크(Subnet Mask)라고 하며, 이를 변경하여 네트워크 주소를 여러 개로 분할하여 사용한다.
- 서브넷 마스크는 각 클래스마다 다르게 사용된다.

- 68 주기억장치의 프로그램과 데이터는 CPU가 직접 액세스할 수 있습니다.

[병행학습]

기억장치 계층 구조의 특징



- 계층 구조에서 상위의 기억장치일수록 접근 속도와 접근 시간이 빠르지만, 기억 용량이 적고 고가이다.
- 주기억장치는 각기 자신의 주소를 갖는 워드 또는 바이트들로 구성되어 있으며, 주소를 이용하여 액세스할 수 있다.
- 레지스터, 캐시 기억장치, 주기억장치의 프로그램과 데이터는 CPU가 직접 액세스할 수 있으나 보조기억장치에 있는 프로그램이나 데이터는 직접 액세스할 수 없다.
- 보조기억장치에 있는 데이터는 주기억장치에 적재된 후 CPU에 의해 액세스될 수 있다.

- 69 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

4회 대비

- **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

70 TCP는 전송 계층(Transport Layer)에 해당합니다.

[병행학습]

OSI(Open System Interconnection) 7계층

- 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

71 ARP(Address Resolution Protocol, 주소 분석 프로토콜)는 TCP/IP의 인터넷 계층에서 호스트의 IP 주소를 호스트와 연결된 네트워크 접속 장치의 물리적 주소(MAC Address)로 바꾸는 역할을 수행합니다.

[병행학습]

네트워크 액세스 계층의 주요 프로토콜

- **Ethernet(IEEE 802.3)** : CSMA/CD 방식의 LAN
- **IEEE 802** : LAN을 위한 표준 프로토콜
- **HDLC** : 비트 위주의 데이터 링크 제어 프로토콜
- **X.25** : 패킷 교환망을 통한 DTE와 DCE 간의 인터페이스를 제공하는 프로토콜
- **RS-232C** : 공중 전화 교환망(PSTN)을 통한 DTE와 DCE 간의 인터페이스를 제공하는 프로토콜

72 수신 측의 처리 능력에 따라 송신 측에서 송신하는 데이터의 전송량이나 전송 속도를 조절하는 기능을 흐름 제어라고 합니다.

[병행학습]

프로토콜의 기능

단편화와 재결합	송신 측에서 전송할 데이터를 전송에 알맞은 일정 크기의 작은 블록으로 자르는 작업을 단편화(Fragmentation)라 하고, 수신 측에서 단편화된 블록을 원래의 데이터로 모으는 것을 재결합(Reassembly)이라함
캡슐화	단편화된 데이터에 송·수신지 주소, 오류 검출 코드, 프로토콜 기능을 구현하기 위한 프로토콜 제어 정보 등의 정보를 부가하는 것으로, 요약화라고도 함
흐름 제어	수신 측의 처리 능력에 따라 송신 측에서 송신하는 데이터의 전송량이나 전송 속도를 조절하는 기능
오류 제어	전송중에 발생하는 오류를 검출하고 정정하여 데이터나 제어 정보의 파손에 대비하는 기능
동기화	송·수신 측이 같은 상태를 유지하도록 타이밍(Timing)을 맞추는 기능
순서 제어	전송되는 데이터 블록(PDU)에 전송 순서를 부여하는 기능으로, 연결 위주의 데이터 전송 방식에만 사용됨
주소 지정	데이터가 목적지까지 정확하게 전송될 수 있도록 목적지 이름, 주소, 경로를 부여하는 기능
다중화	한 개의 통신 회선을 여러 가입자들이 동시에 사용하도록 하는 기능
경로 제어	송·수신 측 간의 송신 경로 중에서 최적의 패킷 교환 경로를 설정하는 기능
전송 서비스	전송하려는 데이터가 사용하도록 하는 별도의 부가 서비스

73 배치 프로그램의 자동 수행 주기 중 특정 조건을 설정해두고 해당 조건이 충족될 때만 수행되는 것을 이벤트성 배치라고 합니다.

[병행학습]

배치 프로그램(Batch Program)

- 사용자와의 상호 작용 없이 여러 작업들을 미리 정해진 일련의 순서에 따라 일괄적으로 처리하는 것을 의미한다.
- 자동 수행 주기

정기 배치	일, 주, 월과 같이 정해진 기간에 정기적으로 수행됨
이벤트성 배치	특정 조건을 설정해두고 조건이 충족될 때만 수행됨
On-Demand 배치	사용자 요청 시 수행됨

· 필수 요소

대용량 데이터	대량의 데이터를 가져오거나, 전달하거나, 계산하는 등의 처리가 가능해야 함
자동화	심각한 오류가 발생하는 상황을 제외하고는 사용자의 개입 없이 수행되어야 함
견고성	잘못된 데이터나 데이터 중복 등의 상황으로 중단되는 일 없이 수행되어야 함
안정성/신뢰성	오류가 발생하면 오류의 발생 위치, 시간 등을 추적할 수 있어야 함
성능	다른 응용 프로그램의 수행을 방해하지 않아야 하고, 지정된 시간 내에 처리가

	완료되어야 함
--	---------

- 배치 스케줄러(Batch Scheduler) : 일괄 처리(Batch Processing) 작업이 설정된 주기에 맞춰 자동으로 수행되도록 지원해주는 도구

74 프레지(Presi)는 프레젠테이션 프로그램의 일종으로서, 사용자의 컴퓨터에 설치할 필요 없이 인터넷 접속을 통해 사용할 수 있는 특징이 있습니다.

[병행학습]

운영체제(OS; Operating System)

- 컴퓨터 시스템의 자원들을 효율적으로 관리하며, 사용자가 컴퓨터를 편리하고 효과적으로 사용할 수 있도록 환경을 제공하는 여러 프로그램의 모임이다.
- 컴퓨터 사용자와 컴퓨터 하드웨어 간의 인터페이스로서 동작하는 시스템 소프트웨어의 일종으로, 다른 응용 프로그램이 유용한 작업을 할 수 있도록 환경을 제공해준다.
- 프로세서(처리기, Processor), 기억장치(주기억장치, 보조 기억장치), 입·출력장치, 파일 및 정보 등의 자원을 관리한다.
- 자원을 효율적으로 관리하기 위해 자원의 스케줄링 기능을 제공한다.
- 사용자와 시스템 간의 편리한 인터페이스를 제공한다.
- 시스템의 각종 하드웨어와 네트워크를 관리·제어한다.
- 데이터를 관리하고, 데이터 및 자원의 공유 기능을 제공한다.
- 시스템의 오류를 검사하고 복구한다.
- 자원 보호 기능을 제공한다.
- 입·출력에 대한 보조 기능을 제공한다.
- 가상 계산기 기능을 제공한다.
- 종류 : Windows, Unix, Linux, MacOS, Android, iOS 등

75 라운드 로빈(RR) 기법은 시분할 시스템(Time Sharing System)을 위해 고안된 방식입니다.

[병행학습]

RR(Round-Robin)

- 시분할 시스템(Time Sharing System)을 위해 고안된 방식으로, FCFS 알고리즘을 선점 형태로 변형한 기법이다.
- FCFS 기법과 같이 준비상태 큐에 먼저 들어온 프로세스가 먼저 CPU를 할당받지만 각 프로세스는 시간 할당량(Time Slice, Quantum) 동안만 실행한 후 실행이 완료되지 않으면 다음 프로세스에게 CPU를 넘겨주고 준비상태 큐의 가장 뒤로 배치된다.
- 할당되는 시간이 클 경우 FCFS 기법과 같아지고, 할당되는 시간이 작을 경우 문맥교환 및 오버헤드가 자주 발생된다.
- 할당되는 시간의 크기가 작으면 작은 프로세스들에게 유리하다.

76 IEEE 802.4는 토큰 버스 방식의 매체 접근 제어(MAC) 계층에 관한 규약입니다.

[병행학습]

IEEE 802의 주요 표준 규격

802.1	전체의 구성, OSI 참조 모델과의 관계, 통신망 관리 등에 관한 규약
802.2	논리 링크 제어(LLC) 계층에 관한 규약
802.3	CSMA/CD 방식의 매체 접근 제어 계층에 관한 규약
802.4	토큰 버스 방식의 매체 접근 제어 계층에 관한 규약
802.5	토큰 링 방식의 매체 접근 제어 계층에 관한 규약

802.6	도시형 통신망(MAN)에 관한 규약
802.11	무선 LAN에 관한 규약
802.15	블루투스에 관한 규약

77 프로세스가 자원을 기다리고 있는 시간에 비례하여 우선순위를 부여함으로써 무기한 연기 문제를 방지하는 기법은 에이징(Aging)입니다.

[병행학습]

에이징(Aging) 기법

- 시스템에서 특정 프로세스의 우선순위가 낮아 무한정 기다리게 되는 경우, 한 번 양보하거나 기다린 시간에 비례하여 일정 시간이 지나면 우선순위를 한 단계씩 높여 가까운 시간 안에 자원을 할당받도록 하는 기법입니다.
- SJF나 우선순위 기법에서 발생할 수 있는 무한 연기 상태, 기아 상태를 예방할 수 있습니다.

78 Windows의 명령 프롬프트 창 또는 DOS 명령어 중 파일의 속성을 변경할 때 사용하는 명령어는 ATTRIB입니다.

[병행학습]

Windows 기본 명령어

DIR	파일 목록을 표시함
COPY	파일을 복사함
TYPE	파일의 내용을 표시함
REN	파일의 이름을 변경함
DEL	파일을 삭제함
MD	디렉터리를 생성함
CD	디렉터리의 위치를 변경함
CLS	화면의 내용을 지움
ATTRIB	파일의 속성을 변경함
FIND	파일을 찾음
CHKDSK	디스크 상태를 점검함
FORMAT	디스크 표면을 트랙과 섹터로 나누어 초기화함
MOVE	파일을 이동함

79 스크립트 언어는 개발 시간은 짧지만 실행 속도는 비교적 느린 편입니다.

[병행학습]

스크립트 언어(Script Language)의 특징

- 컴파일 없이 바로 실행하므로 결과를 바로 확인할 수 있다.
- 배우고 코딩하기 쉽다.
- 개발 시간이 짧다.
- 소스 코드를 쉽고 빠르게 수정할 수 있다.
- 코드를 읽고 해석해야 하므로 실행 속도가 느리다.
- 런타임 오류가 많이 발생한다.

80 포인터 변수에 주소를 저장하기 위해 변수의 주소를 알아낼 때는 변수 앞에 번지 연산자 &를 붙입니다.

[병행학습]

포인터와 포인터 변수

- 포인터는 변수의 주소를 말하며, C언어에서는 주소를 제어할 수 있는 기능을 제공한다.
- C언어에서 변수의 주소를 저장할 때 사용하는 변수를 포인터 변수라 한다.
- 포인터 변수를 선언할 때는 자료의 형을 먼저 쓰고 변수명 앞에 간접 연산자 *를 붙인다([예] int *a;).
- 포인터 변수에 주소를 저장하기 위해 변수의 주소를 알아낼

- 때는 변수 앞에 번지 연산자 &를 붙인다([예] a = &b;).
- 실행문에서 포인터 변수에 간접 연산자 *를 붙이면 해당 포인터 변수가 가리키는 곳의 값을 말한다([예] c = *a;).
- 포인터 변수는 필요에 의해 동적으로 할당되는 메모리 영역인 힙 영역에 접근하는 동적 변수이다.
- 포인터 변수의 용도
 - 연결된 자료 구조를 구성하기 위해 사용한다.
 - 동적으로 할당된 자료 구조를 지정하기 위해 사용한다.
 - 배열을 인수로 전달하기 위해 사용한다.
 - 문자열을 표현하기 위해 사용한다.
 - 커다란 배열에서 요소를 효율적으로 저장하기 위해 사용한다.
 - 메모리에 직접 접근하기 위해 사용한다.

81 문제에 제시된 내용은 링형(Ring)에 대한 설명입니다.

[병행학습]

통신망의 구성 형태

- **성형(Star, 중앙 집중형)** : 중앙에 중앙 컴퓨터가 있고, 이를 중심으로 단말장치들이 연결되는 중앙 집중식의 네트워크 구성 형태
- **링형(Ring, 루프형)** : 컴퓨터와 단말장치들을 서로 이웃하는 것끼리 포인트 투 포인트(Point-to-Point) 방식으로 연결시킨 형태
- **버스형(Bus)** : 한 개의 통신 회선에 여러 대의 단말장치가 연결되어 있는 형태
- **계층형(Tree, 분산형)** : 중앙 컴퓨터와 일정 지역의 단말장치까지는 하나의 통신 회선으로 연결시키고, 이웃하는 단말장치는 일정 지역 내에 설치된 중간 단말장치로부터 다시 연결시키는 형태
- **망형(Mesh)** : 모든 지점의 컴퓨터와 단말장치를 서로 연결한 형태로, 노드의 연결성이 높음

82 ③번은 Secure SDLC의 요구사항 분석 단계에서 수행해야 할 보안 활동입니다.

[병행학습]

Secure SDLC의 단계별 주요 보안 활동

- 요구사항 분석 단계
 - 보안 항목에 해당하는 요구사항을 식별하는 작업을 수행
 - 전산화되는 정보가 가지고 있는 보안 수준을 보안 요소별로 등급을 구분하여 분류
 - 조직의 정보보호 관련 보안 정책을 참고하여 소프트웨어 개발에 적용할 수 있는 보안 정책 항목들의 출처, 요구 수준, 세부 내용 등을 문서화
- 설계 단계
 - 식별된 보안 요구사항들을 소프트웨어 설계서에 반영하고, 보안 설계서를 작성
 - 소프트웨어에서 발생할 수 있는 위협을 식별하여 보안대책, 소요예산, 사고 발생 시 영향 범위와 대응책 등을 수립
 - 네트워크, 서버, 물리적 보안, 개발 프로그램 등 환경에 대한 보안통제 기준을 수립하여 설계에 반영
- 구현 단계
 - 표준 코딩 정의서 및 소프트웨어 개발 보안 가이드를 준수하며, 설계서에 따라 보안 요구사항들을 구현
 - 개발 과정 중에는 지속적인 단위 테스트를 통해 소프트웨어에 발생할 수 있는 보안 취약점을 최소화
 - 코드 점검 및 소스 코드 진단 작업을 통해 소스 코드의 안정성을 확보
- 테스트 단계
 - 설계 단계에서 작성한 보안 설계서를 바탕으로 보안 사항들이 정확히 반영되고 동작되는지 점검

- 동적 분석 도구 또는 모의 침투테스트를 통해 설계 단계에서 식별된 위협들의 해결여부를 검증
- 설계 단계에서 식별된 위협들 외에도 구현 단계에서 추가로 제시된 위협들과 취약점들을 점검할 수 있도록 테스트 계획을 수립하고 시행
- **유지보수 단계** : 이전 과정을 모두 수행하였음에도 발생할 수 있는 보안 사고들을 식별하고, 사고 발생 시 이를 해결하고 보안 패치를 실시

83 미국 카네기멜론 대학교의 소프트웨어 공학연구소(SEI)에서 개발한 것으로, 소프트웨어 개발 조직의 업무 능력 및 조직의 성숙도를 평가하는 모델은 능력 성숙도 통합 모델(CMMI, Capability Maturity Model Integration)입니다.

[병행학습]

- **ISO/IEC 12207** : ISO(국제표준화기구)에서 만든 표준 소프트웨어 생명 주기 프로세스로, 소프트웨어의 개발, 운영, 유지보수 등을 체계적으로 관리하기 위한 소프트웨어 생명 주기 표준을 제공
- **SPICE(Software Process Improvement and Capability dEtermination)** : 정보 시스템 분야에서 소프트웨어의 품질 및 생산성 향상을 위해 소프트웨어 프로세스를 평가 및 개선하는 국제 표준으로, 공식 명칭은 ISO/IEC 15504임
- **SCRUM** : 럭비에서 반칙으로 경기가 중단된 경우 양 팀의 선수들이 럭비공을 가운데 두고 상대방을 밀치기 위해 서로 대치해 있는 대형을 말하며, 스크럼은 이처럼 팀이 중심이 되어 개발의 효율성을 높인다는 의미가 내포된 용어임

84

```
public static void main(String[] args) {
    ① int a = 5;
    ② a = func(a);
}
    ③ static int func(int a) {
    ④ return a <= 5 ? func(a) : 3;
}
```

- ①번에서 정수형 변수 a가 선언되고 5로 초기화 된 후 ②번에서는 a의 값 5를 인수로 func() 메소드 ③번을 호출합니다. ④번이 처음 수행될 때 a는 5의 값을 갖고 있어 참(true)이므로 다시 자기를 호출하는 func(a)를 수행하게 되는데, a의 값이 변화될 수 있는 코드가 함수 내에 존재하지 않으므로 끊임 없이 func(a)를 호출하는 무한 반복(Loop)에 빠지게 됩니다.

85 클래스 및 패키지 내부에서는 접근이 가능하며, 하위 클래스나 패키지 외부에서는 접근이 불가능한 접근 제한자는 Default입니다.

[병행학습]

Java의 접근 제어자

한정자	클래스 내부	패키지 내부	하위 클래스	패키지 외부
Public	O	O	O	O
Protected	O	O	O	X
Default	O	O	X	X
Private	O	X	X	X

86 프로토타입 모델에 위험 분석 기능을 추가한 생명주기 모형은 나선형 모형입니다.

[병행학습]

CASE(Computer Aided Software Engineering)의 개념

- 소프트웨어 개발 과정에서 사용되는 요구 분석, 설계, 구현,

검사 및 디버깅 과정 전체 또는 일부를 컴퓨터와 전용 소프트웨어 도구를 사용하여 자동화하는 것이다.

- 소프트웨어 생명 주기의 전체 단계를 연결해 주고 자동화해주는 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.

CASE 사용의 이점

- 소프트웨어 개발 기간을 단축하고 개발 비용을 절감할 수 있다.
- 자동화된 기법을 통해 소프트웨어 품질이 향상된다.
- 소프트웨어의 유지보수를 간편하게 수행할 수 있다.
- 소프트웨어의 생산성이 향상되고 생산, 운용 활동을 효과적으로 관리·통제할 수 있다.
- 품질과 일관성을 효과적으로 제어할 수 있다.
- 소프트웨어 개발의 모든 단계에 걸친 표준을 확립할 수 있다.

- 87 문제에 제시된 지문은 BGP(Border Gateway Protocol)에 대한 설명입니다.

[병행학습]

라우팅 프로토콜

IGP	하나의 자율 시스템(AS) 내의 라우팅에 사용되는 프로토콜
RIP	<ul style="list-style-type: none"> • 현재 가장 널리 사용되는 라우팅 프로토콜 • 소규모 동종의 네트워크(자율 시스템, AS) 내에서 효율적인 방법임 • 인접해 있는 라우터와 라우팅 정보를 교환하는 대표적인 거리 벡터(Distance Vector) 라우팅 • 최대 홉(Hop) 수를 15로 제한함 • 라우팅 정보를 30초마다 네트워크 내의 모든 라우터에 알리며, 180초 이내에 새로운 라우팅 정보가 수신되지 않으면 해당 경로를 이상 상태로 간주함
OSPF	<ul style="list-style-type: none"> • 홉(Hop) 수에 제한이 없으므로 대규모 네트워크에서 많이 사용되는 라우팅 프로토콜 • 라우팅 정보에 변화가 있을 때에, 변화된 정보만 네트워크 내의 모든 라우터에 알리는 링크 상태(Link State) 라우팅
EGP	자율 시스템(AS) 간의 라우팅, 즉 게이트웨이 간의 라우팅에 사용되는 프로토콜
BGP	<ul style="list-style-type: none"> • 자율 시스템(AS) 간의 라우팅 프로토콜로, EGP의 단점을 보완하기 위해 만들어진 • 초기에 BGP 라우터들이 연결될 때에는 전체 경로 제어표를 교환하고, 이후에는 변화된 정보만을 교환함

- 88 신뢰성 있는 전송을 위해 수행하는 확인 작업인 3-way-handshake 과정을 의도적으로 중단시키는 공격 방법은 SYN Flooding입니다.

[병행학습]

- **Smurfing** : IP나 ICMP의 특성을 악용하여 엄청난 양의 데이터를 한 사이트에 집중적으로 보냄으로써 네트워크를 불능 상태로 만드는 공격 방법
- **Ping of Death** : 명령을 전송할 때 패킷의 크기를 인터넷 프로토콜 허용 범위(65,536 바이트) 이상으로 전송하여 공격 대상의 네트워크를 마비시키는 서비스 거부 공격 방법

• **Land** : 패킷을 전송할 때 송신 IP 주소와 수신 IP 주소를 모두 공격 대상의 IP 주소로 하여 공격 대상에게 전송하는 것으로, 이 패킷을 받은 공격 대상은 송신 IP 주소가 자신이므로 자신에게 응답을 수행하게 되는데, 이러한 패킷이 계속해서 전송될 경우 자신에 대해 무한히 응답하게 하는 공격

- 89 문제에 제시된 지문은 리치 인터넷 애플리케이션(RIA; Rich Internet Application)에 대한 설명입니다.

[병행학습]

- **개인정보 영향평가 제도(PIA; Privacy Impact Assessment)** : 개인 정보를 활용하는 새로운 정보시스템의 도입 및 기존 정보시스템의 중요한 변경 시 시스템의 구축·운영이 기업과 고객은 물론 국민의 사생활에 미칠 영향에 대해 미리 조사·분석·평가하는 제도
- **매시업(Mashup)** : 웹에서 제공하는 정보 및 서비스를 이용하여 새로운 소프트웨어나 서비스, 데이터베이스 등을 만드는 기술
- **시맨틱 웹(Semantic Web)** : 컴퓨터가 사람을 대신하여 정보를 읽고 이해하고 가공하여 새로운 정보를 만들어 낼 수 있도록 이해하기 쉬운 의미를 가진 차세대 지능형 웹

- 90 긴 시간동안 안정적인 서비스 운영을 위해 장애 발생 시 즉시 다른 시스템으로 대체 가능한 환경을 구축하는 메커니즘을 고가용성(HA, High Availability)라고 하며, 가용성을 극대화하는 방법에는 클러스터, 이중화 등이 있습니다.

[병행학습]

- **RAID(Redundant Array of Inexpensive Disk, Redundant Array of Independent Disk)** : 여러 개의 하드디스크로 디스크 배열을 구성하여 파일을 구성하고 있는 데이터 블록들을 서로 다른 디스크들에 분산 저장하여 디스크의 속도를 향상시키는 기술
- **컴패니언 스크린(Companion Screen)** : 앤 스크린(N Screen)의 한 종류로, TV 방송 시청 시 방송 내용을 공유하며 추가적인 기능을 수행할 수 있는 스마트폰, 태블릿PC 등을 의미함
- **멤스(MEMS; Micro-Electro Mechanical Systems)** : 초정밀 반도체 제조 기술을 바탕으로 센서, 액추에이터(Actuator) 등 기계 구조를 다양한 기술로 미세 가공하여 전자기기계적 동작을 할 수 있도록 한 초미세 장치

- 91 문제에 제시된 지문은 COCOMO의 유형 중 반분리형에 대한 설명입니다.

[병행학습]

COCOMO의 소프트웨어 규모별 유형

- **조직형(Organic Mode)**
 - 기관 내부에서 개발된 중·소 규모의 소프트웨어이다.
 - 일괄 자료 처리나 과학 기술 계산용, 비즈니스 자료 처리용으로 5만(50KDSI) 라인 이하의 소프트웨어를 개발하는 유형이다.
 - 사무 처리용, 업무용, 과학용 응용 소프트웨어 개발에 적합하다.
- **반분리형(Semi-Detached Mode)**
 - 조직형과 내장형의 중간형으로, 트랜잭션 처리 시스템이나 운영체제, 데이터베이스 관리 시스템 등의 30만(300KDSI) 라인 이하의 소프트웨어를 개발하는 유형이다.
 - 컴파일러, 인터프리터와 같은 유틸리티 개발에 적합하다.
- **내장형(Embedded Mode)**
 - 최대형 규모의 트랜잭션 처리 시스템이나 운영체제 등의

30만(300KDSI) 라인 이상의 소프트웨어를 개발하는 유형이다.

- 신호기 제어 시스템, 미사일 유도 시스템, 실시간 처리 시스템 등의 시스템 프로그램 개발에 적합하다.

COCOMO의 소프트웨어 구체화 정도별 유형

- **기본(Basic)형** : 소프트웨어의 크기(생산 코드 라인 수)와 개발 유형만을 이용하여 비용을 산정하는 모형
- **중간(Intermediate)형** : 기본형의 공식을 토대로 사용하나, 제품, 컴퓨터, 개발 요원, 프로젝트의 특성을 이용하여 비용을 산정하는 모형
- **발전(Detailed)형** : 중간형을 보완하여 만들어진 방법으로 개발 공정별로 보다 자세하고 정확하게 노력을 산출하여 비용을 산정하는 모형

- 92 공개키 암호화 기법은 알고리즘이 복잡하여 암호화/복호화 속도가 느립니다.

[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.

- 93 자원 산정은 인력 관리의 내용에 해당합니다.

[병행학습]

프로젝트 관리(Project Management)

- 주어진 기간 내에 최소의 비용으로 사용자를 만족시키는 시스템을 개발하기 위한 전반적인 활동이다.
- 관리 유형

일정 관리	작업 순서, 작업 기간 산정, 일정 개발, 일정 통제
비용 관리	비용 산정, 비용 예산 편성, 비용 통제
인력 관리	프로젝트 팀 편성, 자원 산정, 프로젝트 조직 정의, 프로젝트 팀 개발, 자원 통제, 프로젝트 팀 관리
위험 관리	위험 식별, 위험 평가, 위험 대처, 위험 통제
품질 관리	품질 계획, 품질 보증 수행, 품질 통제 수행

- 94 LFSR과 RC4가 스트림 암호화 방식에 해당하는 개인키 암호화 기법입니다.

[병행학습]

개인키 암호화(Private Key Encryption) 기법

- 개인키 암호화 기법은 동일한 키로 데이터를 암호화하고 복호화한다.
- 개인키 암호화 기법은 대칭 암호 기법 또는 단일키 암호화 기법이라고도 한다.
- 개인키 암호화 기법은 한 번에 하나의 데이터 블록을 암호화하는 블록 암호화 방식과, 평문과 동일한 길이의 스트림을 생성하여 비트 단위로 암호화 하는 스트림 암호화 방식으로 분류된다.
- 종류
 - 블록 암호화 방식 : DES, SEED, AES, ARIA
 - 스트림 암호화 방식 : LFSR, RC4
- **장점** : 암호화/복호화 속도가 빠르며, 알고리즘이 단순하고, 공개키 암호 기법보다 파일의 크기가 작음
- **단점** : 사용자의 증가에 따라 관리해야 할 키의 수가 상대적으로 많아짐

- 95 필요한 모든 것에 RFID 태그를 부착하여 수집한 정보를 무선으로 수집할 수 있도록 구성된 네트워크는 USN(Ubiquitous Sensor Network, 유비쿼터스 센서 네트워크)입니다.

[병행학습]

- **GIS(Geographic Information System, 지리 정보 시스템)** : 지리적인 자료를 수집·저장·분석·출력할 수 있는 컴퓨터 응용 시스템으로, 위성을 이용해 모든 사물의 위치 정보를 제공한다
- **SON(Self Organizing Network, 자동 구성 네트워크)** : 주변 상황에 맞추어 스스로 망을 구성하는 네트워크
- **UWB(Ultra WideBand, 초광대역)** : 짧은 거리에서 많은 양의 디지털 데이터를 낮은 전력으로 전송하기 위한 무선 기술로 무선 디지털 펄스라고도 하며, 블루투스나 비교되는 기술

- 96 문제에 제시된 내용은 트로이 목마(Trojan Horse)에 대한 설명입니다.

[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

97 LOC 기법은 상향식 비용 산정 기법에 속합니다.

[병행학습]

하향식 비용 산정 기법

- 과거의 유사한 경험을 바탕으로 전문 지식이 많은 개발자들이 참여한 회의를 통해 비용을 산정하는 비과학적인 방법이다.
- 프로젝트의 전체 비용을 산정한 후 각 작업별로 비용을 세분화한다.
- 하향식 비용 산정 기법에는 전문가 감정 기법, 델파이 기법 등이 있다.
- **전문가 감정 기법**
 - 조직 내에 있는 경험이 많은 두 명 이상의 전문가에게 비용 산정을 의뢰하는 기법이다.
 - 가장 편리하고 신속하게 비용을 산정할 수 있으며, 의뢰자로부터 믿음을 얻을 수 있다.
 - 새로운 프로젝트에는 과거의 프로젝트와 다른 요소들이 있다는 것을 간과할 수 있다.
 - 새로운 프로젝트와 유사한 프로젝트에 대한 경험이 없을 수 있다.
 - 개인적이고 주관적일 수 있다.
- **델파이 기법**
 - 전문가 감정 기법의 주관적인 편견을 보완하기 위해 많은 전문가의 의견을 종합하여 산정하는 기법이다.
 - 전문가들의 편견이나 분위기에 지배되지 않도록 한 명의 조정자와 여러 전문가로 구성된다.

98 소프트웨어 개발 프로젝트에서 시스템에 사용되는 모든 리소스를 정의하고 각 리소스별로 적절한 보안 요구사항을 적용하는 직무는 아키텍트(Architect)입니다.

[병행학습]

소프트웨어 개발 직무별 보안 활동

- **프로젝트 관리자(Project Manager)**
 - 응용 프로그램에 대한 보안 전략을 조직 구성원들에게 전달한다.
 - 조직 구성원들에게 응용 프로그램 보안 영향을 이해시킨다.
 - 조직의 상태를 모니터링 한다.
- **요구사항 분석가(Requirement Specifier)**
 - 아키텍트가 고려해야 할 보안 관련 비즈니스 요구사항을 설명한다.
 - 프로젝트 팀이 고려해야 할 구조 정의 및 해당 구조에 존재하는 자원에 대한 보안 요구사항을 정의한다.
- **아키텍트(Architect)**
 - 보안 오류가 발생하지 않도록 보안 기술 문제를 충분히 이해한다.
 - 시스템에 사용되는 모든 리소스 정의 및 각 리소스별로 적절한 보안 요구사항을 적용한다.

- **설계자(Designer)**
 - 특정 기술에 대해 보안 요구사항의 만족성 여부를 확인한다.
 - 문제 발생 시 최선의 문제 해결 방법을 결정한다.
 - 애플리케이션 보안 수준에 대한 품질 측정을 지원한다.
 - 많은 비용이 필요한 수정 요구사항을 최소화하기 위한 방법을 제공한다.
 - 다른 소프트웨어와 통합할 때 발생할 수 있는 보안 위험에 대해 이해해야 한다.
 - 소프트웨어에서 발견된 보안 위험에 대해 적절히 대응한다.
- **구현 개발자(Implementer)**
 - 구조화된 소프트웨어 개발 환경에서 프로그램을 원활히 구현할 수 있도록 시큐어 코딩 표준을 준수하여 개발한다.
 - 다른 사람이 소프트웨어의 안전 여부를 쉽게 확인할 수 있도록 문서화 한다.
- **테스트 분석가(Test Analyst)**
 - 소프트웨어 개발 요구사항과 구현 결과를 반복적으로 확인한다.
 - 테스트 분석가는 반드시 보안 전문가일 필요는 없지만 보안 위험에 대한 학습이나 툴(Tool) 사용법 정도는 숙지하고 있어야 한다.
- **보안 감사자(Security Auditor)**
 - 소프트웨어 개발 프로젝트의 현재 상태의 보안을 보장한다.
 - 요구사항 검토 시 요구사항의 적합성과 완전성을 확인한다.
 - 소프트웨어 개발 프로젝트의 전체 단계에서 활동한다.
 - 설계 단계에서는 보안 문제로 이어질 수 있는 사항이 있는지 확인한다.
 - 구현 단계에서는 보안 문제가 있는지 확인한다.

99 참조 모니터와 보안 커널의 3가지 특징에는 격리성, 검증 가능성, 완전성이 있습니다.

[병행학습]

참조 모니터(Reference Monitor)

- 보호대상의 객체에 대한 접근통제를 수행하는 추상머신이며, 이것을 실제로 구현한 것이 보안 커널이다.
- 참조 모니터는 보안 커널 데이터베이스(SKDB; Security Kernel Database)를 참조하여 객체에 대한 접근 허가 여부를 결정한다.
- 참조 모니터와 보안 커널의 3가지 특징
 - **격리성(Isolation)** : 부정 조작이 불가능해야 함
 - **검증가능성(Verifiability)** : 적절히 구현되었다는 것을 확인할 수 있어야 함
 - **완전성(Completeness)** : 우회가 불가능해야 함

100 소프트웨어의 구현 단계에서 발생할 수 있는 보안 취약점들을 최소화하기 위해 보안 요소들을 고려하며 코딩하는 것을 시큐어 코딩(Secure Coding)이라고 합니다.

[병행학습]

시큐어 코딩(Secure Coding)

- 소프트웨어의 구현 단계에서 발생할 수 있는 보안 취약점들을 최소화하기 위해 보안 요소들을 고려하며 코딩하는 것을 의미한다.
- 보안 취약점을 사전에 대응하여 안정성과 신뢰성을 확보하기 위해 사용된다.
- 보안 정책을 바탕으로 시큐어 코딩 가이드를 작성하고, 개발 참여자에게는 시큐어 코딩 교육을 실시해야 한다.