



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 표기하십시오.

제1과목 소프트웨어 설계

1. 소프트웨어 개발 방법론 중 애자일 모형의 핵심 가치에 대한 설명으로 옳지 않는 것은?

- ① 계약 협상보다는 고객과 협업에 더 가치를 둔다.
- ② 변화에 반응하기 보다는 계획과 일정을 준수하는 것에 더 가치를 둔다.
- ③ 프로세스와 도구보다는 개인과 상호작용에 더 가치를 둔다.
- ④ 방대한 문서보다는 실행되는 SW에 더 가치를 둔다.

2. 다음 보기에 제시된 스크럼(Scrum) 개발 과정을 진행 순서에 맞게 나열한 것은?

<보기>

- ㉠ 스프린트 검토 회의(Sprint Review)
- ㉡ 스프린트 계획 회의(Sprint Planning Meeting)
- ㉢ 일일 스크럼 회의(Daily Scrum Meeting)
- ㉣ 스프린트(Sprint)
- ㉤ 스프린트 회고(Sprint Retrospective)

- ① ㊦ → ㊧ → ㊨ → ㊩ → ㊪
② ㊦ → ㊪ → ㊧ → ㊨ → ㊩
③ ㊦ → ㊪ → ㊨ → ㊧ → ㊩
④ ㊦ → ㊨ → ㊩ → ㊧ → ㊪

3. 새로운 시스템을 개발하기 위해 현행 시스템의 하드웨어 구성을 파악하고자 한다. 이 때 고려해야 할 사항이 아닌 것은?

- ① 운용 서버의 주요 사양
- ② 운용 서버의 용도
- ③ 운용 서버의 이중화 여부
- ④ 운용 서버의 사용 기간

4. 웹 애플리케이션 서버(WAS)와 관련된 요구사항 식별 시 고려해야 할 사항이 아닌 것은?

- ① 가용성 ② 성능
③ 이식성 ④ 기술 지원

5. 요구사항의 유형 중 기능적 요구사항(Functional requirement)에 대한 설명으로 옳지 않은 것은?

- ① 시스템이 무엇을 하는지, 어떤 기능을 하는지에 대한 사항
- ② 프로젝트의 원활한 수행을 위한 관리 방법에 대한 사항
- ③ 시스템이 반드시 수행해야 하는 기능
- ④ 사용자가 시스템을 통해 제공받기를 원하는 기능

6. 요구사항 분석 기법 중 정형 분석(Formal Analysis)에 대한 설명으로 옳지 않은 것은?

- ① 구문(Syntax)과 의미(Semantics)를 갖는 정형화된 언어를 사용한다.
- ② 요구사항을 수학적 기호로 표현하여 분석한다.
- ③ 요구사항 분석의 마지막 단계에서 이루어진다.
- ④ 요구사항을 만족시키기 위한 구성 요소를 식별한다.

7. 다음 설명에 해당하는 요구사항 확인 기법은?

- 사용자가 실제로 사용될 환경에서 요구사항들이 모두 충족되는지 사용자 입장에서 확인하는 과정이다.
- 각각의 요구사항을 어떻게 확인할 것인지에 대한 계획이 필요하다.

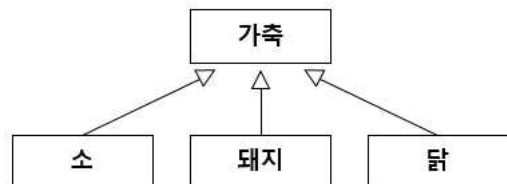
- ① 인수 테스트 ② 모델 검증
③ 프로토타이핑 ④ 요구사항 검토

8. 다음 두 개체 간의 관계는 UML 다이어그램에서 어떤 관계에 해당하는가?



- ① 연관(Association) 관계
- ② 집합(Aggregation) 관계
- ③ 포함(Composition) 관계
- ④ 일반화(Generalization) 관계

9. UML의 다이어그램에서 관계를 완성하고자 한다. 다음 관계의 표현으로 가장 적합한 것은?



- ① 의존 관계(Dependency)
- ② 실체화 관계(Realization)
- ③ 일반화 관계(Generalization)
- ④ 연관 관계(Association)

10. 다음 <실무 사례>에 해당하는 사용자 인터페이스는 무엇인가?

<실무 사례>

시중에 출시된 VR 기기들은 사용자의 위치와 방향, 움직임
을 추적하는 기능을 탑재하고 있어, 사용자는 앞·뒤·좌·우
로 이동하거나 팔을 휘두르거나, 허공에서 버튼을 누르는
등 다양한 동작을 통해 기기를 조작하는 것이 가능하다.

1회 대비

- ① CLI(Command Line Interface)
- ② GUI(Graphical User Interface)
- ③ VUI(Voice User Interface)
- ④ NUI(Natural User Interface)

11. 웹 사이트 개발 시 고려해야 할 웹의 3요소에 해당하지 않는 것은?

- ① 웹 가용성(Web Availability)
- ② 웹 표준(Web Standards)
- ③ 웹 접근성(Web Accessibility)
- ④ 웹 호환성(Cross Browsing)

12. 다음 중 사용자 인터페이스(User Interface)의 설계 도구에 대한 설명으로 옳지 않은 것은?

- ① 화면 설계 도구에는 파워포인트, 스토리보드, 와이어프레임, 목업 등이 있다.
- ② 목업은 실제 화면과 유사한 동적인 형태의 모형이다.
- ③ 프로토타입에는 페이퍼 프로토타입과 디지털 프로토타입이 있다.
- ④ 와이어프레임은 기획 단계에서 페이지 레이아웃이나 구성 요소 등 뼈대를 설계하는 단계이다.

13. 다음의 정황 시나리오에서 추출된 요구사항으로 옳지 않은 것은?

- PC를 통해 다른 프로그램으로 작성했던 표와 그래프를 불러와 발표용 자료를 만들었다.
- 스마트폰으로 해당 자료를 확인하는 한편 잘못된 곳을 수정하였다.
- 발표 주제와 관련된 사람들에게 해당 자료를 공유하여 문제점은 없는지 자문을 요청하였다.

- ① 외부 프로그램의 개체를 가져와 사용할 수 있어야 한다.
- ② PC와 스마트폰에서 모두 구현이 가능해야 한다.
- ③ 공유가 가능해야 한다.
- ④ 공유된 자료를 동시에 열람할 수 있어야 한다.

14. UI의 유용성 평가에 대한 설명으로 틀린 것은?

- ① 사용자 측면에서 복잡한 시스템을 얼마나 편리하게 사용할 수 있는지를 평가하는 것이다.
- ② 유용성이 좋은 UI의 설계를 위해서는 사용자 모형과 개발자 모형 간의 간극이 커야 한다.
- ③ UI의 주된 목적은 유용성이 뛰어난 UI를 제작하는 것이다.
- ④ 시스템의 문제점을 찾아내고 개선 방향을 제시하기 위한 조사 과정이다.

15. 소프트웨어 아키텍처 설계의 기본 원리에 대한 설명으로 틀린 것은?

- ① 소프트웨어 아키텍처 설계의 기본 원리에는 모듈화, 추상화, 단계적 분해, 정보은닉이 있다.
- ② 모듈화는 소프트웨어의 수정 및 재사용, 유지 관리 등을 용이하도록 기능들을 모듈 단위로 나누는 것이다.
- ③ 추상화는 문제의 세부적인 개념을 설계한 후 차례로 통합하여 포괄적인 개념을 구성하는 것이다.
- ④ 정보은닉은 다른 모듈이 접근하거나 변경하지 못하도록 한 모듈 내부에 포함된 절차와 자료들의 정보를 감추는 것이다.

16. 아키텍처 패턴 중 장애 허용 시스템과 병렬 컴퓨팅 시스템에서 주로 활용되는 패턴으로, 메인 컴포넌트에서 서브 컴포넌트로 작업을 분할하여 배정한 후, 처리된 결과물을 다시 돌려받는 방식의 패턴을 무엇이라 하는가?

- ① 클라이언트-서버 ② 파이프-필터
- ③ 마스터-슬레이브 ④ 모델-뷰-컨트롤러

17. 모듈 결합도의 종류 중 스탬프 결합도(Stamp Coupling)에 대한 설명으로 틀린 것은?

- ① 모듈 간의 인터페이스로 배열이나 레코드 등의 자료 구조가 전달될 때의 결합도이다.
- ② 결합의 정도가 자료 결합도보다 강하고, 제어 결합도보다 약하다.
- ③ 두 모듈이 동일한 자료 구조를 조회하는 경우의 결합도이다.
- ④ 자료 구조의 변화가 다른 모듈에 영향을 끼치지 않는다.

18. 다음 설명에 해당하는 코드의 종류는 무엇인가?

코드화 대상 항목의 성질, 즉 길이, 넓이, 부피, 지름, 높이 등의 물리적 수치를 그대로 코드에 적용시키는 방법이다. 대표적인 예로 가로가 55cm, 세로가 120cm인 합판을 코드로 표현하면 '55-120'이 된다.

- ① 표의 숫자 코드(Significant Digit Code)
- ② 순차 코드(Sequence Code)
- ③ 연상 코드(Mnemonic Code)
- ④ 블록 코드(Block Code)

19. 다음이 설명하는 미들웨어는?

- 은행 계정, 항공기와 버스 예약 업무 등 온라인 트랜잭션 업무에서 트랜잭션을 처리하고 감시하는 미들웨어이다.
- 사용자 수가 증가하여도 빠른 응답 속도를 유지해야 하는 업무에 적합한 미들웨어이다.

- ① TP-Monitor ② RPC
- ③ WAS ④ DB

20. 송·수신 시스템 사이에서 교환되는 데이터 중 인터페이스 표준 항목에 대한 설명으로 틀린 것은?

- ① 시스템 공통부와 거래 공통부로 구분된다.
- ② 송·수신 시스템을 연계하는데 표준적으로 필요한 데이터를 의미한다.
- ③ 시스템 공통부의 구성 정보에는 인터페이스 ID, 전송 시스템 정보, 서비스 코드 정보 등이 있다.
- ④ 거래 공통부의 구성 정보에는 응답 결과 정보, 장애 정보 등이 있다.

제2과목 소프트웨어 개발

21. 다음 설명에 해당하는 자료 구조는 무엇인가?

- 동일한 자료형의 데이터들이 같은 크기로 나열되어 순서를 갖고 있는 집합이다.
- 기억장소의 추가가 어렵고, 데이터 삭제 시 데이터가 저장되어 있던 기억장소는 빈 공간으로 남아있다.
- 데이터마다 동일한 이름의 변수를 사용한다.

- ① 리스트(List) ② 배열(Array)
- ③ 스택(Stack) ④ 트리(Tree)

22. 데이터의 독립성에 대한 설명으로 옳지 않은 것은?

- ① 응용 프로그램과 데이터베이스를 독립시킨 것은 논리적 독립성이다.
- ② 논리적 독립성은 데이터의 구조가 변경되어도 응용 프로그램에 영향을 주지 않는 것을 의미한다.
- ③ 물리적 독립성은 응용 프로그램과 하드웨어를 독립시킨 것이다.
- ④ 물리적 독립성은 DBMS가 변경되어도 응용 프로그램에 영향을 주지 않는 것을 의미한다.

23. 다음 괄호에 들어갈 적합한 용어는 무엇인가?

데이터베이스의 상태를 변환시키는 하나의 논리적 기능을 수행하기 위한 작업의 단위를 트랜잭션이라고 하며, 이를 제어하기 위해 사용하는 COMMIT, ROLLBACK과 같은 명령어를 ()이라고 한다.

- ① SQL ② DDL
③ TCL ④ DML

24. 절차형 SQL에 대한 설명으로 옳지 않은 것은?

- ① BEGIN ~ END 형식으로 작성되는 블록(Block) 구조로 구성된다.
② DBMS 엔진에서 직접 실행되기 때문에 일반적인 프로그래밍 언어에 비해 효율이 뛰어나다.
③ 프로그래밍 언어와 같이 연속적인 실행이나 분기, 반복 등의 제어가 가능하다.
④ 기능별 모듈화가 가능하다.

25. 모듈 간 통신 방식을 구현하기 위해 사용되는 IPC의 메소드에 속하지 않는 것은?

- ① Adapter ② Shared Memory
③ Semaphores ④ Message Queueing

26. 다음 중 ISO/IEC/IEEE 29119-3 표준에 따른 테스트 케이스의 구성 요소가 아닌 것은?

- ① 테스트 항목(Test Item)
② 테스트 스크립트(Test Script)
③ 입력 명세(Input Specification)
④ 환경 설정(Environmental Needs)

27. 협업 도구에 대한 설명으로 옳지 않은 것은?

- ① 그룹웨어, 협업 소프트웨어 등으로도 불린다.
② 개발에 참여하는 사람들이 서로 원활히 프로젝트를 수행할 수 있도록 도와주는 도구를 의미한다.
③ 협업 도구의 사용이 익숙하지 않더라도 일단 사용한다면 생산성의 향상을 기대할 수 있다.
④ 진행 현황 및 일정 등을 관리해주거나, 팀원 간 정보 공유를 위한 도구 등이 협업 도구에 속한다.

28. 모듈별로 생성한 실행 파일들을 묶어 배포용 설치 파일을 만드는 것을 무엇이라 하는가?

- ① 패키징(Packaging) ② 빌딩(Building)
③ 모듈화(Modularity) ④ 컴파일링(Compiling)

29. 릴리즈 노트(Release Note)를 통해 확인할 수 있는 정보가 아닌 것은?

- ① 테스트 진행 방법에 대한 결과
② 소프트웨어에서 사용한 외부 라이브러리
③ 소프트웨어 사양에 대한 개발팀의 정확한 준수 여부
④ 소프트웨어에 포함된 전체 기능

30. 디지털 저작권 관리(DRM)에 대한 설명으로 잘못된 것은?

- ① 아날로그 콘텐츠인 경우에는 디지털로 변환하여 DRM 패키징을 수행한다.
② 패키징을 수행하면 콘텐츠에 암호화된 저작권자의 전자서명이 포함된다.
③ 패키징된 콘텐츠의 라이선스 정보는 DRM 컨트롤러에 등록된다.
④ 크기가 작은 콘텐츠는 실시간으로 패키징되어 배포되며, 크기가 크면 미리 패키징을 수행한 후 배포된다.

31. 소프트웨어 사용자 매뉴얼에 대한 설명으로 틀린 것은?

- ① 소프트웨어가 설치된 이후 사용자가 소프트웨어를 사용하는데 필요한 내용들을 기록한다.
② 매뉴얼은 서브시스템 단위로 작성한다.
③ 소프트웨어 배포 후 발생할 수 있는 변경사항의 관리를 위해 매뉴얼의 버전을 관리한다.
④ 컴포넌트 명세서와 컴포넌트 구현 설계서를 토대로 작성한다.

32. 소프트웨어의 개발 과정에서 소프트웨어의 변경 사항을 관리하기 위해 개발된 일련의 활동을 형상 관리라고 한다. 다음 중 형상 관리의 기능 중 형상 통제에 대한 설명으로 옳은 것은?

- ① 형상 항목에 대한 변경 요구를 검토하여 현재의 베이스 라인이 잘 반영될 수 있도록 조정하는 작업
② 베이스 라인의 무결성을 평가하기 위해 확인, 검증, 검열 과정을 통해 공식적으로 승인하는 작업
③ 형상 관리 대상을 계층(Tree) 구조로 구분하여 수정 및 추적이 용이하도록 하는 작업
④ 소프트웨어 업그레이드나 유지 보수 과정에서 생성된 다른 버전의 형상 항목을 관리하는 것

33. 다음 설명에 해당하는 소프트웨어 버전 관리 도구는 무엇인가?

- 클라이언트/서버 방식의 버전 관리 도구로, 서버에서 최신 버전의 파일들과 변경 내역이 관리되는 오픈 소스 도구이다.
- 모든 개발 작업은 trunk에서 수행되며, 추가 작업은 branches 안에 별도의 디렉토리를 만들어 작업을 완료한 후 병합한다.
- 커밋할 때마다 리버전이 1씩 증가한다.

- ① SCCS ② DCVS
③ SVN ④ GIT

34. 다음 중 애플리케이션 테스트가 필요한 이유로 가장 옳지 않은 것은?

- ① 테스트의 역량 강화를 위해
② 새로운 오류의 유입을 예방하기 위해
③ 사용자의 요구사항이나 기대 수준 등을 만족시키기 위해
④ 프로그램 실행 전에 오류를 발견하여 예방하기 위해

35. 구분 기반 테스트, 결정 기반 테스트 등 소프트웨어 내부의 논리 흐름에 따라 테스트 케이스를 작성하고 확인하는 테스트를 무엇이라 하는가?

- ① 구조 기반 테스트 ② 명세 기반 테스트
③ 경험 기반 테스트 ④ 정적 테스트

36. 다음 중 블랙박스 테스트에 속하지 않는 테스트 방식은?

- ① 동등 분할 검사 ② 제어 구조 검사
③ 경계값 분석 ④ 오류 예측 검사

37. 다음 괄호에 들어갈 적합한 테스트는 무엇인가?

()는 이미 테스트된 프로그램의 테스팅을 반복하는 것으로, 통합 테스트로 인해 변경된 모듈이나 컴포넌트에 새로운 오류가 있는지 확인하는 테스트이다. ()는 수정한 모듈이나 컴포넌트가 다른 부분에 영향을 미치는지, 오류가 생기지 않았는지 테스트하여 새로운 오류가 발생하지 않음을 보증하기 위해 반복 테스트한다.

- ① 회귀 테스트
② 혼합식 통합 테스트
③ 상향식 통합 테스트
④ 하향식 통합 테스트

1회 대비

38. 테스트 데이터에 대한 설명 중 틀린 것은?

- ① 시스템의 기능이나 적합성 등을 테스트하기 위해 만든 데이터 집합이다.
- ② 소프트웨어의 기능을 차례대로 테스트할 수 있도록 만든 데이터이다.
- ③ 잘못된 테스트 데이터는 잘못된 결과를 도출하기 때문에 신중하게 작업해야 한다.
- ④ 테스트 데이터는 실제 운영되는 데이터가 아닌 스크립트를 통해서 인위적으로 만든 데이터만을 이용해야 한다.

39. 테스트 하네스(Test Harness)의 구성 요소에 해당하지 않는 것은?

- ① 테스트 드라이버(Test Driver)
- ② 테스트 블록(Test Block)
- ③ 테스트 케이스(Test Case)
- ④ mock 오브젝트(Mock Object)

40. ESB(Enterprise Service Bus)에 대한 설명으로 틀린 것은?

- ① 애플리케이션 간 연계, 데이터 변환 등의 인터페이스를 제공하는 솔루션이다.
- ② 서비스 보다는 애플리케이션 중심의 통합을 지향한다.
- ③ 범용적 사용을 위해 애플리케이션과의 결합도가 약하다.
- ④ 관리 및 보안 유지가 쉽고, 높은 수준의 품질 지원이 가능하다.

제3과목 데이터베이스 구축

41. 데이터 모델에 표시하는 요소 중의 하나로, 데이터의 무결성 보장을 위해 저장될 수 있는 실제 데이터에 조건을 설정하여 해당 조건에 부합하지 않으면 저장되지 않도록 하는 것은 무엇인가?

- ① Operation ② Relationship
- ③ Constraint ④ Structure

42. 다음은 개체 구성 방식에 따라 속성(Attribute)을 분류한 것이다. 설명이 틀린 것은?

- ① 기본키 속성은 개체를 식별할 수 있는 속성을 의미한다.
- ② 외래키 속성은 다른 개체와의 관계에서 만들어진 속성이다.
- ③ 외래키 속성과 관계된 개체가 변경되어도 외래키 속성에는 영향을 주지 않는다.
- ④ 일반 속성은 기본키와 외래키에 속하지 않는 속성들을 의미한다.

43. 다음의 설명에 해당하는 관계는 무엇인가?

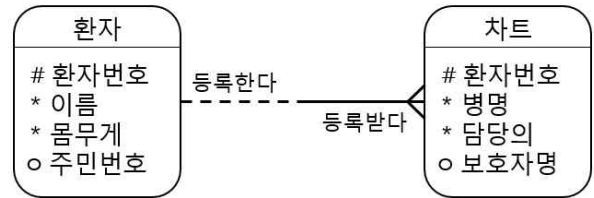
- <도시> 개체와 <시민> 개체는 시민이 도시에 거주하는 관계이다.
- 한 명의 시민은 하나의 도시만 거주할 수 있다.
- 하나의 도시는 여러 명의 시민을 거주시킬 수 있다.

- ① 1:0 ② 1:1
- ③ 1:N ④ N:M

44. 주 식별자(Primary Identifier)의 속성이 두 개 이상인 경우 속성들을 하나의 속성으로 묶어 사용하는 식별자를 무엇이라 하는가?

- ① 대리 식별자(Surrogate Identifier)
- ② 복합 식별자(Composit Identifier)
- ③ 외부 식별자(Foreign Identifier)
- ④ 보조 식별자(Alternate Identifier)

45. 다음 그림에 해당하는 E-R 모델 표기법에 대한 설명으로 틀린 것은?



- ① 개체를 모서리가 둥근 박스로 표시한다.
- ② 속성이 반드시 저장되어야 하는 경우 속성명 앞에 'o'를, 선택적인 경우 '*'를 표시한다.
- ③ 개체명은 박스 안 가장 위에 표시한다.
- ④ 관계는 관계 표기 기호를 사용하여 표시한 후 해당 개체의 역할을 동사적 단어로 입력한다.

46. 다음 E-R 모델에서 '고객'과 '주문서'는 개체 테이블이고, '주문'은 관계(Relationship)로 만들어진 테이블이다. 다음 E-R 모델에 대한 설명으로 가장 옳지 않은 것은?



- ① <고객> 테이블과 <주문서> 테이블은 <주문> 테이블을 관계로 하여 연결되어 있다.
- ② <고객> 테이블에 있는 '고객번호'와 <주문서> 테이블에 있는 '주문번호'는 기본키이고, <주문> 테이블에 있는 '고객번호'와 '주문번호'는 외래키이다.
- ③ <고객>과 <주문>의 관계는 1:N이고, <주문서>와 <주문>의 관계는 1:1이다.
- ④ <고객>과 <주문서>의 관계는 N:M이 된다.

47. SQL문에서 사원(사번, 이름, 부서) 테이블에 사번이 1201, 이름이 "홍길동", 부서가 "인사"인 튜플을 삽입하는 명령으로 옳은 것은?

- ① INSERT FROM 사원(사번, 이름, 부서) VALUES(1201, '홍길동', '인사');
- ② INSERT 사원 INTO VALUES(1201, '홍길동', '인사');
- ③ INSERT TO 사원(사번, 이름, 부서) VALUES(1201, '홍길동', '인사');
- ④ INSERT INTO 사원 VALUES(1201, '홍길동', '인사');

48. 테이블 PRODUCT(CODE, NAME, TYPE, SALES)에 대해 다음과 같은 SQL문이 실행되었을 때 그 결과에 대한 설명으로 가장 옳지 않은 것은?

```
SELECT TYPE, COUNT(*), SUM(SALES)
FROM PRODUCT
GROUP BY TYPE
HAVING SUM(SALES) >= 300
ORDER BY TYPE DESC;
```

- ① TYPE 속성을 기준으로 그룹화 되어 출력된다.
- ② 각 튜플들은 TYPE 속성을 기준으로 내림차순 출력된다.
- ③ TYPE 속성으로 그룹화된 튜플들의 SALES 속성 합계가 300 미만인 경우 출력되지 않는다.
- ④ HAVING의 조건 "SUM(SLAES) >= 300"을 WHERE 조건으로 바꿔도 동일한 결과가 출력된다.

49. 다음 Oracle로 작성된 SQL문에 대한 설명으로 가장 옳지 않은 것은?
(단, MOD(A, B)는 A를 B로 나누어 나머지를 구하는 함수이다.)

```
CREATE PROCEDURE GET_DEPT(i IN INT) IS
BEGIN
    IF (MOD(i, 10) = 2) THEN
        INSERT INTO 회원(코드, 분류) VALUES(i, 'VIP');
    ELSE
        INSERT INTO 회원(코드, 분류) VALUES(i, '일반');
    END IF;
END;
```

- ① RUN GET_DEPT(30)을 입력하면 해당 프로시저가 실행된다.
- ② 파라미터로 i를 전달받아 수행되는 프로시저이다.
- ③ i를 10으로 나눈 나머지의 값에 따라 수행되는 코드가 다르다.
- ④ i의 값이 302이면 <회원> 테이블의 '코드' 속성에는 i의 값 302가, '분류' 속성에는 'VIP'가 저장된다.

50. 관계형 데이터베이스에서 사용하는 용어 중 카디널리티(Cardinality)에 대한 설명으로 옳은 것은?

- ① 디그리 또는 차수라고 한다.
- ② 테이블에 속한 튜플의 수를 의미한다.
- ③ 속성이 취할 수 있는 원자값들의 집합이다.
- ④ 파일 구조상의 데이터 항목 또는 필드에 해당한다.

51. 데이터 무결성을 강화하는 방법에 대한 설명으로 틀린 것은?

- ① 데이터를 조작하는 프로그램에 무결성 검증 코드를 삽입한다.
- ② 데이터베이스의 트리거 개체를 생성하여 무결성 검증 코드를 삽입한다.
- ③ 제약 조건을 설정하여 무결성을 유지한다.
- ④ 네트워크 채널을 통제하여 무결성 조건에 부합하지 않는 데이터를 반송한다.

52. 릴레이션 R의 차수가 3이고 카디널리티가 4이며, 릴레이션 S의 차수가 5이고 카디널리티가 6일 때, 두 개의 릴레이션에 대해 카디션 프로덕트를 수행하면 새롭게 생성되는 릴레이션의 차수와 카디널리티는 얼마인가?

- ① 8, 24 ② 15, 10
③ 7, 11 ④ 12, 30

53. 정규화(Normalization)의 각 정규형에 대한 설명으로 틀린 것은?

- ① 1NF는 모든 도메인이 원자값만으로 구성되어 있다.
- ② 2NF는 모든 속성이 기본키에 완전 함수적 종속이다.
- ③ 3NF는 모든 속성이 다치 종속의 결정자에 함수적 종속 관계를 만족한다.
- ④ BCNF는 모든 결정자가 후보키이다.

54. 인덱스 대상 속성 설정 기준에 대한 설명으로 가장 옳지 않은 것은?

- ① 인덱스 속성의 분포도가 20% 이상인 속성
- ② 수정이 빈번하지 않은 속성
- ③ Primary key 또는 Unique 제약 조건이 설정된 속성
- ④ 임-출력 자료 등에서 조회 및 출력 조건으로 사용되는 속성

55. 다음 설명에 해당하는 용어는 무엇인가?

본산 데이터베이스의 4대 목표 중 하나로, 동일 데이터가 여러 곳에 중복되어 있더라도 사용자는 마치 하나의 데이터만 존재하는 것처럼 사용하고, 시스템은 자동으로 여러 자료에 대한 작업을 수행한다.

- ① 위치 투명성 ② 중복 투명성
③ 병행 투명성 ④ 장애 투명성

56. 데이터베이스에서 데이터 저장 시 데이터의 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법을 무엇이라 하는가?

- | | |
|--------------|-------------|
| ① Cluster | ② Index |
| ③ Constraint | ④ Partition |

57. 후보키(Candidate Key)에 대한 설명으로 틀린 것은?

- ① 기본키로 사용할 수 있는 속성들을 말한다.
- ② 구성에 따라 후보키가 없는 릴레이션이 존재할 수 있다.
- ③ 유일성과 최소성을 만족한다.
- ④ 튜플을 유일하게 식별하기 위해 사용하는 속성들의 부분집합이다.

58. 데이터베이스 보안의 접근통제에 대한 설명으로 틀린 것은?

- ① DAC은 신원에 따라 접근 권한을 부여하는 통제 방식이다.
- ② MAC은 객체별, 사용자별 등급을 비교하여 접근 권한을 부여한다.
- ③ 접근통제의 보안 모델 중 하나인 무결성 모델은 기밀성 보장이 최우선인 모델이다.
- ④ 접근통제의 조건에는 값 중속 통제, 다중 사용자 통제, 컨텍스트 기반 통제가 있다.

59. 서버와 저장장치를 연결하는 스토리지의 종류 중 하나로, 서버와 저장장치를 연결하는 전용 네트워크를 별도로 구성하는 방식을 무엇이라 하는가?

- ① DAS ② NAS
③ DES ④ SAN

60. 논리 데이터 모델의 슈퍼타입 속성들을 각각의 서브타입에 추가하여 서브타입들을 물리 데이터 모델의 개별 테이블로 만드는 변환 방법에 대한 설명으로 가장 옳지 않은 것은?

- ① 각 서브타입 속성들의 선택 사양이 불명확한 경우에 유리하다.
- ② 처리할 때마다 서브타입 유형을 구분할 필요가 없다.
- ③ 여러 개의 테이블로 통합하므로 테이블당 크기가 감소하여 전체 테이블 스캔시 유리하다.
- ④ UID(Unique Identifier)의 유지 관리가 어렵다.

제4과목 프로그래밍 언어 활용

61. 다음 중 C언어의 변수명 작성에 대한 설명으로 가장 옳지 않은 것은?

- ① 대·소문자를 구분하지 않는다.
- ② 영문, 숫자, 언더바 등을 사용할 수 있다.
- ③ 글자 수에 제한이 없다.
- ④ 예약어는 변수명으로 사용할 수 없다.

62. 다음 중 C언어에서 사용하는 기억 클래스의 종류가 아닌 것은?

- ① Automatic Variable ② Dynamic Variable
③ Register Variable ④ Static Variable

63. C언어의 do ~ while문에 대한 설명 중 옳지 않은 것은?

- ① while문의 조건이 처음부터 거짓일 때는 while문 내부의 코드를 실행하지 않는다.
- ② while문의 맨 마지막에 “;”이 필요하다.
- ③ 무조건 한 번은 실행하고 경우에 따라서는 여러 번 실행하는 처리에 사용하면 유용하다.
- ④ 조건이 참이면 실행할 문장을 반복한다.

64. C언어의 포인터 형(Pointer Type)에 대한 설명으로 틀린 것은?

- ① 변수의 주소를 지정할 때 사용하는 변수를 포인터 변수라고 한다.
- ② 포인터 변수는 기억장소의 번지를 기억하는 정적 변수이다.
- ③ 보통 변수의 번지를 참조하려면 번지 연산자 &를 변수 앞에 쓴다.
- ④ 실행문에서는 간접 연산자 *를 사용하여 포인터 변수가 지시하고 있는 내용을 참조한다.

65. 다음 C 코드의 결과 값은?

```
#include <stdio.h>
main( ) {
    int a = 1, sum = 0;
    while (a++ < 10)
        sum += a * 3 >= sum ? a : 1;
    printf("%d", sum);
}
```

- ① 11 ② 30
- ③ 39 ④ 54

66. 다음 JAVA 코드의 결과 값은?

```
public class Test {
    public static void main(String[] args) {
        int array[] = { 1, 3, 5, 7, 9 };
        for (int a : array)
            if(a%3==0) System.out.printf("%");
    }
}
```

- ① * ② **
- ③ *** ④ ****

67. 다음 중 printf() 함수의 사용이 잘못된 것은?

- ① printf("%d", 152);
- ② printf("%f", "programming");
- ③ printf("%c", 'e');
- ④ printf("%e", 1.5e+3);

68. 연산자에 대한 설명으로 가장 옳지 않은 것은?

- ① 이항 연산자는 대입 연산자보다 먼저 계산한다.
- ② 시프트 연산자는 결합규칙에 따라 왼쪽부터 계산한다.
- ③ 대입 연산자는 조건 연산자보다 나중에 계산된다.
- ④ 단항 연산자에는 &, !, ++, -- 등이 있다.

69. 개발 언어의 선정 기준에 대한 설명으로 틀린 것은?

- ① 개발 소프트웨어의 목적에 적합해야 한다.
- ② 개발자들의 이해도보다는 언어의 성능을 우선해야 한다.
- ③ 많은 사람들이 활용하고 있어 자료를 쉽게 찾을 수 있어야 한다.
- ④ 이식성이 좋아야 한다.

70. UNIX에서 커널(Kernel)에 대한 설명으로 옳지 않은 것은?

- ① 프로세스 관리, 기억장치 관리 등을 담당한다.
- ② UNIX 시스템의 중심부에 해당한다.
- ③ 컴퓨터 부팅시 주기억장치에 적재되어 상주하면서 실행된다.
- ④ 사용자와 시스템 간의 인터페이스를 제공한다.

71. 다음 중 주기억장치 관리 전략과 거리가 먼 것은?

- ① Paging 전략 ② Placement 전략
- ③ Fetch 전략 ④ Replacement 전략

72. 주기억장치에 완전히 비어 있는 3개의 페이지가 있다. 페이지 교체 방법으로 FIFO를 사용할 때 다음의 순서로 페이지 번호를 요청할 경우 페이지 부재(Fault)는 몇 번 발생하는가?

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

- ① 5 ② 6
- ③ 7 ④ 8

73. 다음 중 워킹 셋(Working Set)에 대한 설명으로 옳지 않은 것은?

- ① 프로그램의 구역성(Locality) 특징을 이용한다.
- ② 주기억장치에 적재되지 않으면 스래싱이 발생할 수 있다.
- ③ 프로세스 실행중에는 크기가 변하지 않는다.
- ④ 워킹 셋에 속한 페이지를 참조하면 프로세스의 기억장치 사용은 안정상태가 된다.

74. 스레드(Thread)에 대한 설명으로 옳지 않은 것은?

- ① 스레드는 동일 프로세스 환경에서 서로 독립적인 다중 수행이 불가능하다.
- ② 스레드는 프로세스의 일부 특성을 갖고 있기 때문에 경량(Light Weight) 프로세스라고도 한다.
- ③ 사용자 스레드의 경우 사용자가 만든 라이브러리를 사용하여 스레드를 운용한다.
- ④ 스레드를 사용함으로써 하드웨어, 운영체제의 성능과 응용 프로그램의 처리율을 향상시킬 수 있다.

75. 다음 설명에 해당하는 용어는 무엇인가?

준비 상태에서 대기하고 있는 프로세스 중 하나가 스케줄링되어 중앙처리장치를 할당받아 실행 상태로 전이되는 과정이다.

- ① 실행(Run) ② 대기(Wait)
- ③ 준비(Ready) ④ 디스패치(Dispatch)

76. IPv6에 대한 설명으로 틀린 것은?

- ① 주소의 확장성, 융통성, 연동성이 뛰어나다.
- ② 더 많은 IP 주소를 지원할 수 있도록 주소의 크기는 64비트이다.
- ③ 인종성, 기밀성 등의 지원으로 보안 성능이 향상되었다.
- ④ 유니캐스트, 멀티캐스트, 애니캐스트를 지원한다.

77. OSI 7계층의 데이터 링크 계층 기능에 대한 설명으로 적합하지 않은 것은?

- ① 종단 시스템 간에 End-to-End 통신을 지원한다.
- ② 에러 제어, 흐름 제어 등의 기능을 수행한다.
- ③ 대표적 프로토콜로는 HDLC, LAPB 등이 있다.
- ④ 프레임의 시작과 끝을 구분하기 위한 프레임의 동기화 기능을 한다.

78. 다음과 같은 기능을 가지고 있는 프로토콜은?

- 양방향 연결과 가상 회선 연결 형태의 서비스를 제공한다.
- 스트림 위주로 데이터를 전달한다.
- 신뢰성 있는 경로를 확립하고 메시지 전송을 감독한다.
- 순서 제어, 오류 제어, 흐름 제어 기능을 제공한다.

- ① RTCP(Real-Time Control Protocol)
- ② FTP(File Transfer Protocol)
- ③ UDP(User Datagram Protocol)
- ④ TCP(Transmission Control Protocol)

1회 대비

79. 프레임워크(Framework)의 특성 중 제어의 역흐름에 대한 설명으로 옳은 것은?

- ① 모듈화를 강화하고 설계 및 구현의 변경에 따른 영향을 최소화한다.
- ② 재사용 가능한 모듈들을 제공하여 생산성을 향상시킨다.
- ③ 다형성(Polymorphism)을 통해 인터페이스의 확장이 가능하다.
- ④ 객체들의 제어를 개발자가 아닌 프레임워크가 담당한다.

80. 패키지 소프트웨어와 전용 개발 소프트웨어에 대한 설명으로 틀린 것은?

- ① 전용 개발 소프트웨어는 패키지 소프트웨어에 비해 인력과 시간이 절약된다.
- ② 전용 개발 소프트웨어는 개발자의 역량에 따라 품질이 크게 달라진다.
- ③ 패키지 소프트웨어는 소프트웨어에 대한 라이선스가 판매자에게 있다.
- ④ 패키지 소프트웨어는 기능 요구사항의 70% 이상을 충족하는 소프트웨어가 존재하는 경우에만 이용하는 것이 좋다.

제5과목 정보시스템 구축 관리

81. 다음 중 SPICE(ISO/IEC 15504)의 프로세스 범주에 속하지 않는 것은?

- ① Customer-Supplier Process
- ② Engineering Process
- ③ Maintenance Process
- ④ Management Process

82. 소프트웨어 개발 방법론의 테일러링(Tailoring)에 대한 설명으로 틀린 것은?

- ① 프로젝트 상황 및 특성에 맞도록 절차, 사용기법 등을 수정 및 보완한다.
- ② 프로젝트의 규모, 복잡도, 구성원 등에 따라 적용되는 다양한 테일러링 기법이 존재한다.
- ③ 목표 환경, 요구사항 등의 내부적 요건을 고려해야 한다.
- ④ 보유 기술, 프로젝트 규모 등의 외부적 요건을 고려해야 한다.

83. 다음 중 소프트웨어 비용을 결정할 때 사용되는 프로젝트 요소에 대한 설명으로 옳지 않은 것은?

- ① 제품의 복잡도 : 응용, 유틸리티, 시스템 소프트웨어 등 소프트웨어의 종류에 따라 달라지는 문제의 난이도를 의미한다.
- ② 시스템의 크기 : 대형, 소형 등 소프트웨어의 규모에 따라 개발할 시스템의 규모를 의미한다.
- ③ 요구되는 신뢰도 : 정확성, 견고성, 완전성, 일관성 등 프로그램이 일정한 기간 내에 주어진 조건하에서 필요한 기능을 수행하는 정도를 의미한다.
- ④ 개발 기간 : 소프트웨어를 개발하는 기간을 의미한다.

84. 전문가 감정 기법의 주관적인 편견을 보완하기 위해 많은 전문가의 의견을 종합하여 산정하는 비용 산정 기법은?

- ① 기능 점수 모형 ② 델파이 기법
- ③ LOC 기법 ④ COCOMO 모형

85. 원시 코드 라인 수(LOC) 기법에 의하여 예측된 총 라인 수가 25,000라인, 개발에 참여할 프로그래머가 5명, 프로그래머들의 평균 생산성이 월간 500라인일 때 개발에 소요되는 기간은?

- ① 8개월 ② 10개월
- ③ 11개월 ④ 20개월

86. 2명의 개발자가 5개월에 걸쳐 15,000 라인의 코드를 개발하였을 때, 월별(Person Month) 생산성 측정을 위한 계산 방식으로 가장 적합한 것은?

- ① 15,000 / 2 ② 15,000 / (5 × 2)
- ③ 15,000 / 5 ④ (2 × 15,000) / 5

87. 다음 중 정보통신기술에 대한 설명으로 적당하지 않은 것은?

- ① 메시 네트워크(Mesh Network)는 차세대 이동통신, 홈네트워킹, 공공 안전 등 특수 목적을 위한 새로운 방식의 네트워크 기술이다.
- ② 와이선(Wi-SUN)은 스마트 그리드와 같은 장거리 무선 통신을 필요로 하는 사물 인터넷(IoT) 서비스를 위한 저전력 장거리 통신 기술이다.
- ③ 피코넷(PICONET)은 여러 개의 독립된 통신장치가 블루투스 기술이나 UWB 통신기술을 사용하여 통신망을 형성하는 무선 네트워크 기술이다.
- ④ NFC는 각종 컴퓨팅 자원을 중앙 컴퓨터에 두고 인터넷 기능을 갖는 단말기로 언제 어디서나 인터넷을 통해 컴퓨터 작업을 수행할 수 있는 환경을 의미한다.

88. 다음 중 통신망의 구성 형태에 대한 설명으로 틀린 것은?

- ① 버스형(Bus)은 기밀 보장이 어렵고, 통신 회선의 길이에 제한이 있다.
- ② 성형(Star)에 연결된 각 단말장치들은 중앙 컴퓨터를 통하여 데이터를 교환한다.
- ③ 망형(Mesh)은 많은 단말장치로부터 많은 양의 통신을 필요로 하는 경우에 유리하며, LAN에서 가장 많이 사용되는 방식이다.
- ④ 계층형(Tree)은 일정 지역의 단말장치까지는 하나의 통신 회선으로 연결하고, 이웃하는 단말장치는 일정 지역 내에 설치된 중간 단말장치로부터 다시 연결하는 형태이다.

89. 다음 설명에 해당하는 용어는 무엇인가?

소프트웨어를 제공하는 입장에서는 약의적이지 않은 유용한 소프트웨어라고 주장할 수 있지만 사용자 입장에서는 유용할 수도 있고 약의적일 수도 있는 소프트웨어이다.

- ① 매시업(Mashup)
- ② RIA
- ③ 그레이웨어(Grayware)
- ④ 뉴럴링크(Neuralink)

90. 현실의 사물을 소프트웨어로 가상화한 모델로, 현실의 사물을 대신해 다양한 상황을 모의 실험하기 위한 용도로 사용되는 것은 무엇인가?

- ① 디지털 트윈(Digital Twin)
- ② 소프트웨어 에스크로(Software Escrow)
- ③ 서비스형 소프트웨어(SaaS)
- ④ 오픈 그리드 서비스 아키텍처(OGSA)

91. 행정안전부가 보안 활동의 주체로서 수행해야 하는 역할이 아닌 것은?

- ① 보안 정책을 총괄한다.
- ② 보안 법규, 제도 등을 정비한다.
- ③ 보안 가이드를 개발한다.
- ④ 보안 전문가의 양성을 위한 업무를 수행한다.

99. 다음에서 설명하는 네트워크 공격 방법은 무엇인가?

- ① APT ② Smishing
③ Qshing ④ DDoS Attack

100. 침입 탐지 시스템(IDS)에 대한 설명으로 틀린 것은?

- ① M-DISC ② Memristor
③ MEMS ④ RAID

- ① 프로세스의 논리적 구역을 지정하여 구역을 벗어나는 행위를 제한한다.
- ② 동일 시간에 하나의 프로세스만 수행되도록 하여 동시 실행으로 발생하는 보안 취약점을 제거한다.
- ③ 사용자별로 특정 장비만 사용하도록 제한한다.
- ④ 내부 정보를 암호화 하는 방법이 자원 보호 방법 중 가장 단순한 방법에 속한다.

- ① 연쇄 복귀(Cascading Rollback)
- ② 모순성(Inconsistency)
- ③ 갱신 분실(Lost Update)
- ④ 비완료 의존성(Uncommitted Dependency)

- ① 호환성(Compatibility)
- ② 기밀성(Confidentiality)
- ③ 무결성(Integrity)
- ④ 가용성(Availability)

- ① 데이터 표준화는 중요한 데이터가 저장되는 특정 오브젝트를 대상으로 한다.
- ② 명칭은 업무적 보편성을 갖는 이름으로 정의한다.
- ③ 사용 목적과 유사한 데이터는 형식을 일관되게 정의한다.
- ④ 적합성을 위해 기본값, 허용 범위 등을 사전에 지정한다.

- ① Rivest Shamir Adleman
- ② Data Encryption Standard
- ③ Hash Algorithm
- ④ Public Key Infrastructure

정답 및 해설

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

- 1 애자일 모형(Agile Model)은 변화에 기민하고 유연하게 대응하는 것이 특징인 모형으로, 계획을 따르기 보다는 변화에 반응하는 것에 더 가치를 둡니다.

[병행학습]

애자일 개발 4가지 핵심 가치

- 프로세스와 도구보다는 개인과 상호작용에 더 가치를 둔다.
- 방대한 문서보다는 실행되는 SW에 더 가치를 둔다.
- 계약 협상보다는 고객과 협업에 더 가치를 둔다.
- 계획을 따르기 보다는 변화에 반응하는 것에 더 가치를 둔다.

- 2 스크럼(Scrum) 개발 과정은 계획한 내용을 토대로 일정 기간 동안 스프린트를 수행하면서 진행 상황을 매일 점검하고 하나의 스프린트가 끝나면 검토한 후 진행을 되돌아봅니다.

- 3 새로운 시스템 개발에 있어서 하드웨어 구성 파악 시 고려해야 할 사항에는 운영 서버의 용도, 수량, 주요 사양, 이중화 적용 여부가 있습니다.

- 4 웹 애플리케이션 서버(WAS)에 대한 요구사항 식별 시 고려해야 할 사항에는 가용성, 성능, 기술 지원, 구축 비용이 있습니다.

[병행학습]

애플리케이션 서버(WAS) 관련 요구사항 식별 시 고려사항

- 가용성 : 장시간 운영 시 발생할 수 있는 고유의 장애 발생 가능성
- 성능 : 대규모의 트랜잭션 처리 능력 및 다양한 옵션 지원
- 기술 지원 : 제조업체의 안정적인 기술 지원 및 여러 사용자들 간의 정보 공유
- 구축 비용 : 라이선스, 유지관리 등에 대한 비용

- 5 ②번은 비기능 요구사항(Non-functional requirements) 중 하나인 프로젝트 관리 요구사항에 해당합니다.

[병행학습]

기능 요구사항(Functional Requirements)

- 시스템이 무엇을 하는지, 어떤 기능을 하는지에 대한 사항
- 시스템의 입력이나 출력으로 무엇이 포함되어야 하는지, 시스템이 어떤 데이터를 저장하거나 연산을 수행해야 하는지에 대한 사항
- 시스템이 반드시 수행해야 하는 기능
- 사용자가 시스템을 통해 제공받기를 원하는 기능

비기능 요구사항(Non-functional Requirements)

- 시스템 장비 구성 요구사항

- 성능 요구사항
- 인터페이스 요구사항
- 데이터 요구사항
- 테스트 요구사항
- 보안 요구사항
- 품질 요구사항
- 제약사항
- 프로젝트 관리 요구사항
- 프로젝트 지원 요구사항

- 6 정형 분석은 구문(Syntax)과 의미(Semantics)를 갖는 정형화된 언어를 이용해 요구사항을 수학적 기호로 표현한 후 이를 분석하는 과정입니다. ④번은 요구사항 할당(Requirement Allocation)에 대한 설명입니다.

[병행학습]

요구사항 분석 기법

개발 대상에 대한 사용자의 요구사항 중 명확하지 않거나 모호한 부분을 걸러내기 위한 방법

- **요구사항 분류(Requirement Classification)** : 요구사항을 명확히 확인할 수 있도록 분류함
- **개념 모델링(Conceptual Modeling)** : 요구사항을 보다 쉽게 이해할 수 있도록 현실 세계의 상황을 단순화하여 개념적으로 표현한 것을 모델이라고 하며, 이러한 모델을 만드는 과정을 모델링이라고 함
- **요구사항 할당(Requirement Allocation)** : 요구사항을 만족시키기 위한 구성 요소를 식별하는 것
- **요구사항 협상(Requirement Negotiation)** : 요구사항이 서로 충돌될 경우 이를 적절히 해결하는 과정
- **정형 분석(Formal Analysis)** : 구문(Syntax)과 의미(Semantics)를 갖는 정형화된 언어를 이용해 요구사항을 수학적 기호로 표현한 후 이를 분석하는 과정

- 7 인수 테스트는 사용자가 실제로 사용할 환경에서 요구사항들이 모두 충족되는지 사용자 입장에서 확인하는 과정입니다.

[병행학습]

요구사항 확인 기법

요구사항 개발 과정을 거쳐 문서화된 요구사항 관련 내용을 확인하고 검증하는 방법이다.

- **모델 검증(Model Verification)** : 요구사항 분석 단계에서 개발된 모델이 요구사항을 충족시키는지 검증하는 것
- **프로토타입(Prototyping)** : 초기 도출된 요구사항을 토대로 프로토타입(초기 모형)을 만든 후 대상 시스템의 개발이 진행되는 동안 도출되는 요구사항을 반영하면서 지속적인

로 프로토타입을 재작성하는 과정

- **요구사항 검토(Requirement Reviews)** : 문서화된 요구사항을 훑어보면서 확인하는 것으로 가장 일반적인 요구사항 검증 방법
- **인수 테스트(Acceptance Tests)** : 사용자가 실제로 사용할 환경에서 요구사항들이 모두 충족되는지 사용자 입장에서 확인하는 과정

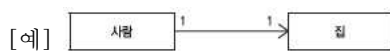
- 8 포함 관계는 집합 관계의 특수한 형태로, 포함하는 사물의 변화가 포함되는 사물에게 영향을 미치는 관계를 표현합니다. 문제의 관계 표현에서 '사원번호'에 해당하는 '사원명'은 하나이며, 퇴직하여 '사원번호'가 없어지게 된다면 '사원명' 또한 필요가 없어집니다.

[병행학습]

관계(Relationships)

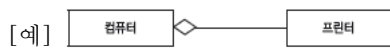
• 연관(Association) 관계

- 2개 이상의 사물이 서로 관련되어 있음



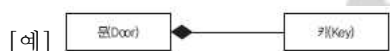
• 집합(Aggregation) 관계

- 하나의 사물이 다른 사물에 포함되어 있는 관계



• 포함(Composition) 관계

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



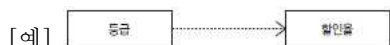
• 일반화(Generalization) 관계

- 하나의 사물이 다른 사물에 비해 더 일반적인지 구체적인지를 표현하는 관계



• 의존(Dependency) 관계

- 연관 관계와 같이 사물 사이에 서로 연관성은 있으나 필요에 의해 서로에게 영향을 주는 짧은 시간 동안만 연관을 유지하는 관계



- 9 그림은 하나의 사물이 다른 사물에 비해 더 일반적인지 구체적인 인지를 표현하는 일반화(Generalization) 관계를 표현하고 있습니다. 가축을 구체적으로 표현하면 소, 돼지, 닭이 되고 반대로 소, 돼지, 닭을 일반적으로 표현하면 가축이 됩니다.

- 10 사용자의 동작으로 기기를 조작하는 인터페이스는 NUI입니다.

[병행학습]

사용자 인터페이스의 종류

- **CLI(Command Line Interface)** : 명령과 출력이 텍스트 형태로 이뤄지는 인터페이스
- **GUI(Graphical User Interface)** : 아이콘이나 메뉴를 마우스로 선택하여 작업을 수행하는 그래픽 환경의 인터페이스
- **NUI(Natural User Interface)** : 사용자의 말이나 행동으로 기기를 조작하는 인터페이스
- **VUI(Voice User Interface)** : 사람의 음성으로 기기를 조작하는 인터페이스

- 11 웹의 3요소에는 웹 표준, 웹 접근성, 웹 호환성이 있습니다.

[병행학습]

웹의 3요소

- **웹 표준(Web Standards)** : 웹에서 사용되는 규칙 또는 기술을 의미하는 것으로, 웹 사이트 작성 시 이용하는 HTML, JavaScript 등에 대한 규정, 웹 페이지가 다른 기종이나 플랫폼에서도 구현되도록 제작하는 기법 등을 포함함
- **웹 접근성(Web Accessibility)** : 누구나, 어떠한 환경에서도 웹 사이트에서 제공하는 모든 정보를 접근하여 이용할 수 있도록 보장하는 것
- **웹 호환성(Cross Browsing)** : 하드웨어나 소프트웨어 등이 다른 환경에서도 모든 이용자에게 동등한 서비스를 제공하는 것

- 12 목업은 실제 화면과 유사하지만 실제로는 구현되지 않은 정적인 형태의 모형입니다.

[병행학습]

UI 설계 도구

- **와이어프레임(Wireframe)** : 페이지에 대한 개략적인 레이아웃이나 UI 요소 등에 대한 뼈대를 설계하는 단계
- **목업(Mockup)** : 디자인, 사용 방법 설명, 평가 등을 위해 와이어프레임보다 좀 더 실제 화면과 유사하게 만든 정적인 형태의 모형
- **스토리보드(Story Board)** : 와이어프레임에 콘텐츠에 대한 설명, 페이지 간 이동 흐름 등을 추가한 문서
- **프로토타입(Prototype)** : 와이어프레임나 스토리보드 등에 인터랙션을 적용함으로써 실제 구현된 것처럼 테스트가 가능한 동적인 형태의 모형
- **유스케이스(Use Case)** : 사용자가 원하는 목표를 달성하기 위해 수행할 내용을 기술한 것

- 13 정황 시나리오는 사용자의 요구사항을 도출하기 위해 작성하는 것으로, 사용자가 목표를 달성하기 위해 수행하는 방법을 순차적으로 묘사한 것입니다. 지문에 공유에 대한 내용은 있지만 동시에 열람에 관한 내용은 없습니다.

[병행학습]

정황 시나리오

- 사용자의 요구사항을 도출하기 위해 작성하는 것으로, 사용자가 목표를 달성하기 위해 수행하는 방법을 순차적으로 묘사한 것이다.
- 요구사항 정의에 사용되는 초기 시나리오이다.
- 개발하는 서비스의 모습을 상상하는 첫 번째 단계로 사용자 관점에서 시나리오를 작성해야 한다.
- 사용자가 주로 사용하는 기능 위주로 작성해야 한다.
- 함께 발생하는 기능들은 하나의 시나리오에 통합한다.
- 육하원칙에 따라 간결하고 명확하게 작성한다.
- 작성된 시나리오는 외부 전문가 또는 경험이 풍부한 사람에게 검토를 의뢰한다.

- 14 유용한 UI를 설계하기 위해서는 UI의 구조, 기능, 가치 등에 대해 사용자가 생각하는 사용자 모형과 시스템 설계자가 만들려고 하는 개발자 모형 간의 간극을 최소화해야 합니다.

[병행학습]

UI의 유용성 평가

- 사용자 측면에서 복잡한 시스템을 얼마나 편리하게 사용할 수 있는지를 평가하는 것이다.
- 시스템의 문제점을 찾아내고 개선 방향을 제시하기 위한 조사 과정이다.
- 유용한 UI를 설계하기 위해서는 UI의 구조, 기능, 가치 등에 대해 사용자가 생각하는 사용자 모형과 시스템 설계자가 만들려고 하는 개발자 모형 간의 차이를 최소화해야 한다.
- 사용자 모형과 개발자 모형 간의 차이가 발생하는 원인

- **실행 차** : 사용자가 원하는 목적과 실행 기능이 다르기 때문에 발생함
- **평가 차** : 사용자가 원하는 목적과 실행 결과가 다르기 때문에 발생함

- 15 소프트웨어 아키텍처 설계의 기본 원리 중 추상화는 문제의 전체적이고 포괄적인 개념을 설계한 후 차례로 세분화하여 구체화시켜 나가는 것입니다.

[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

- 16 마스터 컴포넌트에서 슬레이브 컴포넌트로 작업을 분할한 후, 슬레이브 컴포넌트로부터 처리된 결과물을 다시 돌려받는 방식으로 작업을 수행하는 패턴을 마스터-슬레이브 패턴(Master-Slave Pattern)이라고 합니다.

[병행학습]

주요 아키텍처 패턴(Patterns)의 종류

- **레이어 패턴(Layers pattern)** : 시스템을 계층(Layer)으로 구분하여 구성하는 고전적인 방법 중의 하나로 각각의 서브 시스템들이 계층 구조를 이루며, 상위 계층은 하위 계층에 대한 서비스 제공자가 되고, 하위 계층은 상위 계층의 클라이언트가 됨
- **클라이언트-서버 패턴** : 하나의 서버 컴포넌트와 다수의 클라이언트 컴포넌트로 구성되는 패턴으로, 클라이언트가 서버에 요청하고 응답을 받아 사용자에게 제공하는 방식
- **파이프-필터 패턴** : 데이터 스트림 절차의 각 단계를 필터(Filter) 컴포넌트로 캡슐화하여 파이프(Pipe)를 통해 데이터를 전송하는 패턴
- **모델-뷰-컨트롤러 패턴** : 서브시스템을 모델(Model), 뷰(View), 컨트롤러(Controllor)의 세 부분으로 구조화하는 패턴

- 17 스템프 결합도에서 자료 구조의 어떠한 변화, 즉 포맷이나 구조의 변화는 그것을 조회하는 모든 모듈 및 변화되는 필드를 실제로 조회하지 않는 모듈에도 영향을 미칩니다.

[병행학습]

결합도(Coupling)

- 결합도는 모듈 간에 상호 의존하는 정도 또는 두 모듈 사이의 연관 관계를 의미한다.
- 다양한 결합도로 모듈을 구성할 수 있으나 결합도가 약할수록 품질이 높고, 강할수록 품질이 낮다.
- 결합도가 강하면 시스템 구현 및 유지보수 작업이 어렵다.
- 결합도의 종류
 - **자료 결합도(Data Coupling)** : 모듈 간의 인터페이스가 자료 요소로만 구성될 때
 - **스탬프(검인) 결합도(Stamp Coupling)** : 모듈 간의 인터페이스로 배열이나 레코드 등의 자료 구조가 전달될 때
 - **제어 결합도(Control Coupling)** : 어떤 모듈이 다른 모듈 내부의 논리적인 흐름을 제어하기 위해 제어 신호를 이용하여 통신하거나 제어 요소를 전달할 때

- **외부 결합도(External Coupling)** : 어떤 모듈에서 선언한 데이터(변수)를 외부의 다른 모듈에서 참조할 때
- **공통(공유) 결합도(Common Coupling)** : 공유되는 공통 데이터 영역을 여러 모듈이 사용할 때
- **내용 결합도(Content Coupling)** : 한 모듈이 다른 모듈의 내부 기능 및 그 내부 자료를 직접 참조하거나 수정할 때

- 18 표의 숫자 코드는 코드화 대상 항목의 성질, 즉 길이, 넓이, 부피, 지름, 높이 등의 물리적 수치를 그대로 코드에 적용시키는 방법으로, 유효 숫자 코드라고도 합니다.

[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

- 19 트랜잭션(Transaction)을 처리(Processing)하고 감시(Monitoring)하는 미들웨어는 TP-Monitor입니다. 다른 미들웨어에 대한 개념도 확인하고 넘어가세요.

[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

- 20 거래 공통부의 구성 정보에는 직원 정보, 승인자 정보, 기기 정보, 매체 정보 등이 있습니다. 응답 결과 정보, 장애 정보는 시스템 공통부의 구성 정보입니다.

[병행학습]

인터페이스 표준 항목

- 인터페이스 표준 항목은 송·수신 시스템을 연계하는데 표준적으로 필요한 데이터를 의미한다.
- 인터페이스 표준 항목은 시스템 공통부와 거래 공통부로 나뉜다.
- 시스템 공통부
 - 시스템 간 연동 시 필요한 공통 정보이다.
 - 구성 정보에는 인터페이스 ID, 전송 시스템 정보, 서비스 코드 정보, 응답 결과 정보, 장애 정보 등이 있다.
- 거래 공통부
 - 시스템들이 연동된 후 송·수신 되는 데이터를 처리할 때 필요한 정보이다.
 - 구성 정보에는 직원 정보, 승인자 정보, 기기 정보, 매체 정보 등이 있다.

- 21 동일한 자료형의 데이터들이 같은 크기로 나열되어 순서를 갖고 있는 집합 형태의 자료구조는 배열입니다.

[병행학습]**자료 구조의 종류**

배열 (Array)	동일한 자료형의 데이터들이 같은 크기로 나열되어 순서를 갖고 있는 집합
리스트 (List)	배열 또는 포인터를 이용하여 일정한 순서에 의해 나열된 자료 구조
스택 (Stack)	리스트의 한쪽 끝으로만 자료의 삽입, 삭제 작업이 이루어지는 자료 구조
큐 (Queue)	리스트의 한쪽에서는 삽입 작업이 이루어지고 다른 한쪽에서는 삭제 작업이 이루어지도록 구성한 자료 구조
트리 (Tree)	정점(Node, 노드)과 선분(Branch, 가지)을 이용하여 사이클을 이루지 않도록 구성한 그래프(Graph)의 특수한 형태

- 22 물리적 독립성은 데이터베이스 시스템의 성능 향상을 위해 새로운 디스크를 추가하는 등의 단순한 하드웨어 변경으로부터 응용 프로그램이 영향을 받지 않는 성질을 가리킵니다. 응용 프로그램과 데이터베이스 시스템을 연결해주는 DBMS의 변경은 응용 프로그램에 영향을 주지 않을 수가 없습니다.

[병행학습]**데이터의 독립성**

- 종속성에 대비되는 말로 DBMS의 궁극적 목표이다.
- 데이터의 독립성에는 논리적 독립성과 물리적 독립성이 있다.
- **논리적 독립성** : 응용 프로그램과 데이터베이스를 독립시킴으로써, 데이터의 논리적 구조를 변경시키더라도 응용 프로그램은 변경되지 않음
- **물리적 독립성** : 응용 프로그램과 보조기억장치 같은 물리적 장치를 독립시킴으로써, 데이터베이스 시스템의 성능 향상을 위해 새로운 디스크를 도입하더라도 응용 프로그램에는 영향을 주지 않고 데이터의 물리적 구조만을 변경함

- 23 트랜잭션을 제어하기 위해서 사용하는 명령어를 TCL (Transaction Control Language)이라고 하며, TCL의 종류에는 COMMIT, ROLLBACK, SAVEPOINT가 있습니다. TCL은 기능을 제어하는 명령어라는 공통점으로 DCL의 일부로 분류하기도 합니다. 보기로 나온 다른 용어들의 개념도 확인하고 넘어가세요.

- **SQL** : 관계형 데이터베이스를 지원하는 언어로, 데이터 정의어(DDL), 데이터 조작어(DML), 데이터 제어어(DCL)로 구분됨

- **데이터 정의어(DDL)** : SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의하거나 변경 또는 삭제할 때 사용하는 언어
- **데이터 조작어(DML)** : 데이터베이스 사용자가 응용 프로그램이나 질의어를 통하여 저장된 데이터를 실질적으로 처리하는 데 사용되는 언어
- **데이터 제어어(DCL)** : 데이터의 보안, 무결성, 회복, 병행 수행 제어 등을 정의하는 데 사용되는 언어

- 24 절차형 SQL은 프로그래밍 언어와 같이 연속적인 실행이나 분기, 반복 등의 제어가 가능한 SQL이지만, 일반적인 프로그래밍 언어에 비해 효율은 떨어집니다.

[병행학습]**절차형 SQL**

- C, JAVA 등의 프로그래밍 언어와 같이 연속적인 실행이나 분기, 반복 등의 제어가 가능한 SQL을 의미한다.
- 일반적인 프로그래밍 언어에 비해 효율은 떨어지지만 단일 SQL 문장으로 처리하기 어려운 연속적인 작업들을 처리하는데 적합하다.
- 절차형 SQL을 활용하여 다양한 기능을 수행하는 저장 모듈을 생성할 수 있다.
- DBMS 엔진에서 직접 실행되기 때문에 입·출력 패킷이 적은 편이다.
- BEGIN ~ END 형식으로 작성되는 블록(Block) 구조로 되어 있기 때문에 기능별 모듈화가 가능하다.
- 종류에는 프로시저, 트리거, 사용자 정의 함수가 있다.

- 25 IPC는 모듈 간 통신 방식을 구현하기 위해 사용되는 대표적인 프로그래밍 인터페이스 집합이며, IPC의 메소드 중 Adapter라는 메소드는 없습니다.

[병행학습]**IPC의 대표 메소드**

- **Shared Memory** : 다수의 프로세스가 공유 가능한 메모리를 구성하여 프로세스 간 통신을 수행함
- **Socket** : 네트워크 소켓을 이용하여 네트워크를 점유하는 프로세스들 간 통신을 수행함
- **Semaphores** : 공유 자원에 대한 접근 제어를 통해 프로세스 간 통신을 수행함
- **Pipes/named Pipes** : 'Pipe'라고 불리는 선입선출 형태로 구성된 메모리를 여러 프로세스가 공유하여 통신을 수행함
- **Message Queueing** : 메시지가 발생하면 이를 전달하는 형태로 프로세스 간 통신을 수행함

- 26 테스트 스크립트는 ISO/IEC/IEEE 29119-3 표준에 따른 테스트 케이스의 구성 요소에 포함되지 않습니다.

[병행학습]**ISO/IEC/IEEE 29119-3 표준의 케이스 구성 요소**

- **식별자(Identifier)** : 항목 식별자, 일련번호
- **테스트 항목(Test Item)** : 테스트 대상(모듈 또는 기능)
- **입력 명세(Input Specification)** : 입력 데이터 또는 테스트 조건
- **출력 명세(Output Specification)** : 테스트 케이스 수행 시 예상되는 출력 결과
- **환경 설정(Environmental Needs)** : 필요한 하드웨어나 소프트웨어의 환경
- **특수 절차 요구(Special Procedure Requirement)** : 테스트 케이스 수행 시 특별히 요구되는 절차
- **의존성 기술(Inter-case Dependencies)** : 테스트 케이스 간의 의존성

27 협업 도구에 익숙하지 않거나 이용할 의지가 없으면 협업 도구가 오히려 협업의 방해 요소가 될 수 있습니다.

28 빌딩(Building)은 소스 코드 파일들을 실행 파일로 변환하는 것, 모듈화(Modularity)는 시스템을 각 기능별로 나누는 것, 컴파일링(Compiling)은 소스 코드를 기계어로 변환하는 것을 의미합니다.

29 소프트웨어에서 사용한 외부 라이브러리는 릴리즈 노트를 통해 확인할 수 있는 정보가 아닙니다. 릴리즈 노트를 통해 확인할 수 있는 정보들은 다음과 같습니다.

- 테스트 진행 방법에 대한 결과
- 소프트웨어 사양에 대한 개발팀의 정확한 준수 여부
- 소프트웨어에 포함된 전체 기능
- 소프트웨어의 사용 환경
- 서비스의 내용과 개선 사항

30 패키징된 콘텐츠의 라이선스 정보는 DRM 컨트롤러가 아닌 클리어링 하우스에 등록됩니다.

- **클리어링 하우스(Clearing House)** : 디지털 저작권 라이선스의 중개 및 발급을 수행하는 곳으로, 디지털 저작물의 이용 내역을 근거로 저작권료의 정산 및 분배가 수행됨

[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 패키지(Packager)에 의해 DRM 패키징을 수행한다.
- 콘텐츠의 크기에 따라 음원이나 문서와 같이 크기가 작은 경우에는 사용자가 콘텐츠를 요청하는 시점에서 실시간으로 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 사용권한 소유 여부를 확인받아야 한다.
- 종량제 방식을 적용한 소프트웨어의 경우 클리어링 하우스를 통해 서비스의 실제 사용량을 측정하여 이용한 만큼의 요금을 부과한다.

31 소프트웨어 사용자 매뉴얼은 개별적으로 동작이 가능한 컴포넌트 단위로 작성해야 합니다.

[병행학습]

소프트웨어 사용자 매뉴얼

- 사용자가 소프트웨어를 사용하는 과정에서 필요한 내용을 문서로 기록한 설명서와 안내서이다.
- 사용자가 소프트웨어 사용에 필요한 절차, 환경 등의 제반 사항이 모두 포함되도록 작성한다.
- 소프트웨어 배포 후 발생할 수 있는 오류에 대한 패치나 기능에 대한 업그레이드를 위해 매뉴얼의 버전을 관리한다.
- 개별적으로 동작이 가능한 컴포넌트 단위로 매뉴얼을 작성한다.
- 사용자 매뉴얼은 컴포넌트 명세서와 컴포넌트 구현 설계서를 토대로 작성한다.
- 목차 및 개요, 서문, 기본 사항 등이 기본적으로 포함되어야 한다.

32 형상 항목에 대한 변경 요구를 검토하여 현재의 베이스 라인이 잘 반영될 수 있도록 조정하는 작업을 형상 통제라고 합니다. ②번은 형상 감사, ③번은 형상 식별, ④번은 버전 제어에 대한 설명입니다.

[병행학습]

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

33 보기 중 클라이언트/서버 방식의 버전 관리 도구는 SVN입니다. SCCS는 공유 폴더 방식, DCVS와 GIT은 분산 저장소 방식을 사용하는 소프트웨어 버전 관리 도구입니다.

[병행학습]

소프트웨어 버전 관리 도구

- 공유 폴더 방식
 - 버전 관리 자료가 로컬 컴퓨터의 공유 폴더에 저장되어 관리되는 방식
 - 종류 : SCCS, RCS, PVCS, QVCS 등
- 클라이언트/서버 방식
 - 버전 관리 자료가 중앙 시스템(서버)에 저장되어 관리되는 방식
 - 종류 : CVS, SVN(Subversion), CVSNT, Clear Case, CMVC, Perforce 등
- 분산 저장소 방식
 - 버전 관리 자료가 하나의 원격 저장소와 분산된 개발자 PC의 로컬 저장소에 함께 저장되어 관리되는 방식
 - 종류 : Git, GNU arch, DCVS, Bazaar, Mercurial, TeamWare, Bitkeeper, Plastic SCM 등

34 테스트의 역량은 테스트 수행 전에 테스터가 갖춰줘야 할 조건입니다.

35 소프트웨어 내부의 논리 흐름에 따라 테스트 케이스를 작성하고 확인하는 테스트를 구조 기반 테스트라고 합니다. 나머지 보기로 제시된 테스트의 개념도 확인하고 넘어가세요.

- **명세 기반 테스트** : 사용자의 요구사항에 대한 명세를 빠짐 없이 테스트 케이스로 만들어 구현하고 있는지 확인하는 테스트
- **경험 기반 테스트** : 유사 소프트웨어나 기술 등에 대한 테스트의 경험을 기반으로 수행하는 테스트
- **정적 테스트** : 프로그램을 실행하지 않고 명세서나 소스 코드를 대상으로 분석하는 테스트

36 블랙 박스 테스트에는 동치(동등) 분할 검사, 경계값 분석, 원인-효과 그래프 검사, 오류 예측 검사, 비교 검사가 있습니다. 제어 구조 검사는 화이트박스 테스트에 속합니다.

[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

37 통합 테스트로 인해 변경된 모듈이나 컴포넌트에 새로운 오류가 있는지 확인하는 테스트를 회귀 테스트라고 합니다. 나머지 보기로 제시된 테스트의 개념도 확인하고 넘어가세요.

- 혼합식 통합 테스트 : 하위 수준에서는 상향식 통합, 상위 수준에서는 하향식 통합을 사용하여 최적의 테스트를 지원하는 방식
- 상향식 통합 테스트 : 프로그램의 하위 모듈에서 상위 모듈 방향으로 통합하면서 테스트하는 기법
- 하향식 통합 테스트 : 프로그램의 상위 모듈에서 하위 모듈 방향으로 통합하면서 테스트하는 기법

38 테스트 데이터의 종류에는 선행된 연산에 의해 만든 데이터, 실제 운영되는 데이터를 복제한 실제 데이터, 스크립트를 통해서 인위적으로 만든 가상 데이터 등이 있습니다.

[병행학습]

테스트 데이터

- 시스템의 기능이나 적합성 등을 테스트하기 위해 만든 데이터 집합
- 소프트웨어의 기능을 차례대로 테스트할 수 있도록 만든 데이터
- 잘못된 데이터는 잘못된 결과를 도출하기 때문에 효율적인 테스트를 위해서는 올바른 테스트 데이터를 준비해야 함
- 테스트 데이터 종류
 - 실제 데이터 : 선행된 연산에 의해 만들거나 실제 운영되는 데이터를 복제한 데이터
 - 가상 데이터 : 스크립트를 통해서 인위적으로 만든 데이터

39 테스트 블록(Test Block)은 테스트 하네스의 구성 요소에 속하지 않습니다.

[병행학습]

테스트 하네스(Test Harness)의 구성 요소

- 테스트 드라이버(Test Driver) : 테스트 대상의 하위 모듈을 호출하고, 파라미터를 전달하고, 모듈 테스트 수행 후의 결과를 도출하는 도구
- 테스트 스텝(Test Stub) : 제어 모듈이 호출하는 타 모듈의 기능을 단순히 수행하는 도구로, 일시적으로 필요한 조건만

을 가지고 있는 테스트용 모듈

- 테스트 스위트(Test Suites) : 테스트 대상 컴포넌트나 모듈, 시스템에 사용되는 테스트 케이스의 집합
- 테스트 케이스(Test Case) : 사용자의 요구사항을 정확하게 준수했는지 확인하기 위한 입력 값, 실행 조건, 기대 결과 등으로 만들어진 테스트 항목의 명세서
- 테스트 스크립트(Test Script) : 자동화된 테스트 실행 절차에 대한 명세서
- 목 오브젝트(Mock Object) : 사전에 사용자의 행위를 조건부로 입력해 두면, 그 상황에 맞는 예정된 행위를 수행하는 객체

40 ESB는 애플리케이션 보다는 서비스 중심의 통합을 지향합니다.

[병행학습]

ESB(Enterprise Service Bus)

- 애플리케이션 간 연계, 데이터 변환, 웹 서비스 지원 등 표준 기반의 인터페이스를 제공하는 솔루션이다.
- 애플리케이션 통합 측면에서 EAI와 유사하지만 애플리케이션 보다는 서비스 중심의 통합을 지향한다.
- 특정 서비스에 국한되지 않고 범용적으로 사용하기 위하여 애플리케이션과의 결합도(Coupling)를 약하게(Loosely) 유지한다.
- 관리 및 보안 유지가 쉽고, 높은 수준의 품질 지원이 가능하다.

41 데이터 모델에 표시하는 구조(Structure), 연산(Operation), 제약 조건(Constraint) 중 실제 데이터에 조건을 설정하는 것은 제약 조건(Constraint)입니다.

[병행학습]

데이터 모델에 표시할 요소

- 구조(Structure) : 논리적으로 표현된 개체 타입들 간의 관계로서 데이터 구조 및 정적 성질 표현
- 연산(Operation) : 데이터베이스에 저장된 실제 데이터를 처리하는 작업에 대한 명세로서 데이터베이스를 조작하는 기본 도구
- 제약 조건(Constraint) : 데이터베이스에 저장될 수 있는 실제 데이터의 논리적인 제약 조건

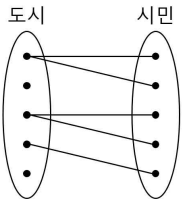
42 외래키는 다른 개체와의 '관계(Relationship)'를 통해 생성된 속성이므로, 관계 있는 개체가 변경되면 외래키 속성도 영향을 받게 됩니다.

[병행학습]

속성의 종류

- 속성의 특성에 따른 분류
 - 기본 속성(Basic Attribute) : 업무 분석을 통해 정의한 속성
 - 설계 속성(Designed Attribute) : 원래 업무상 존재하지 않고 설계 과정에서 도출해내는 속성
 - 파생 속성(Derived Attribute) : 다른 속성으로부터 계산이나 변형 등의 영향을 받아 발생하는 속성
- 개체 구성 방식에 따른 분류
 - 기본키 속성(Primary Key Attribute) : 개체를 식별할 수 있는 속성
 - 외래키 속성(Foreign Key Attribute) : 다른 개체와의 관계에서 포함된 속성
 - 일반 속성 : 개체에 포함되어 있고 기본키, 외래키에 포함되지 않은 속성

- 43 <도시>의 각 원소는 <시민>의 원소 여러 개와 대응하고 있지만, <시민>의 각 원소는 <도시>의 원소 한 개와 대응하는 관계이므로 일 대 다(1:N) 관계가 됩니다.



[병행학습]

관계의 형태

- 일 대 일(1 : 1) : 개체 집합 A의 각 원소가 개체 집합 B의 원소 한 개와 대응하는 관계
 - 일 대 다(1 : N) : 개체 집합 A의 각 원소는 개체 집합 B의 원소 여러 개와 대응하고 있지만, 개체 집합 B의 각 원소는 개체 집합 A의 원소 한 개와 대응하는 관계
 - 다 대 다(N : M) : 개체 집합 A의 각 원소는 개체 집합 B의 원소 여러 개와 대응하고, 개체 집합 B의 각 원소도 개체 집합 A의 원소 여러 개와 대응하는 관계
- 44 주 식별자의 속성이 두 개 이상인 경우 속성들을 하나의 속성으로 묶어 사용하는 식별자를 대리 식별자라고 합니다.

[병행학습]

식별자의 분류

- 주 식별자/보조 식별자
 - 주 식별자(Primary Identifier) : 개체를 대표하는 유일한 식별자
 - 보조 식별자(Alternate Identifier) : 주 식별자를 대신하여 개체를 식별할 수 있는 속성
 - 내부 식별자/외부 식별자
 - 내부 식별자(Internal Identifier) : 개체 내에서 스스로 만들어지는 식별자
 - 외부 식별자(Foreign Identifier) : 다른 개체와의 관계(Relationship)에 의해 외부 개체의 식별자를 가져와 사용하는 식별자
 - 단일 식별자/복합 식별자
 - 단일 식별자(Single Identifier) : 주 식별자가 한 가지 속성으로만 구성된 식별자
 - 복합 식별자(Composit Identifier) : 주 식별자가 두 개 이상의 속성으로 구성된 식별자
 - 원조 식별자/대리 식별자
 - 원조 식별자(Original Identifier) : 업무에 의해 만들어지는 가공되지 않은 원래의 식별자로, 본질 식별자라고도 함
 - 대리 식별자(Surrogate Identifier) : 주 식별자의 속성이 두 개 이상인 경우 속성들을 하나의 속성으로 묶어 사용하는 식별자
- 45 그림의 표기법은 바커 표기법(Barker Notation)입니다. 바커 표기법은 반드시 값이 저장되어야 하는 속성에는 '*' (Mandatory)를 표시하고, 값이 저장될 수도, 안될 수도 있는 속성에는 'o' (Optional)을 표시합니다.

[병행학습]

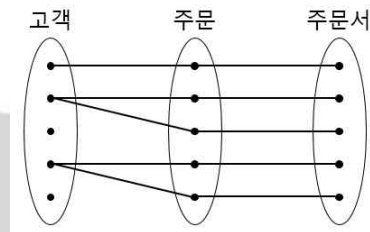
바커 표기법(Barker Notation)

- 영국 컨설팅 회사 CACI에서 개발하였고, 리처드 바커(Richard Barker)에 의해 정립되었다.
- 바커 표기법은 개체, 속성, 기본키 속성, 관계 등을 다음과 같이 표현한다.
- 개체는 모서리가 둥근 박스로 표시하고 개체명은 박스 안 가장 위에 표시한다.

- 속성은 반드시 값이 저장되어야 하는 경우 *(Mandatory)를 표시하고, 값이 저장될 수도, 안될 수도 있는 경우 o (Optional)을 표시한다.
- 관계는 관계 표기 기호를 사용하여 표시한 후 해당 개체의 역할을 동사적 단어로 입력한다.
- 관계 표기 기호

기호	의미
————	필수(Mandatory)
-----	선택적(Optional)
<	다중(Multiple)

- 46 한 명의 <고객>은 여러 <주문>이 가능하고, 각 <주문>은 하나의 <주문서>를 가지므로, 한 명의 <고객>은 여러 <주문서>를 갖는 일 대 다(1:N) 관계가 형성됩니다.



- 47 INSERT문의 기본 형식은 'INSERT INTO 테이블명(속성명) VALUES(값)'이지만, 테이블의 모든 속성에 값을 삽입할 때는 속성명을 생략하여 'INSERT INTO 테이블명 VALUES(값)'으로 사용할 수 있습니다.

- INSERT INTO 사원 : <사원> 테이블에 삽입하라.
- VALUES(1201, '홍길동', '인사') : 첫 번째 필드부터 순서대로 1201, '홍길동', '인사'를 삽입하라.

- 48 WHERE는 개개의 튜플에 조건을 지정하는 것이고, HAVING은 그룹에 대해 조건을 지정하는 명령이므로 HAVING 대신 WHERE를 사용하면 동일한 결과가 출력되지 않습니다. 또한 WHERE 조건문에는 SUM, AVG 등과 같은 그룹 함수를 사용할 수도 없습니다. 문제에 제시된 코드를 살펴보면 다음과 같습니다.

```

1 SELECT TYPE, COUNT(*), SUM(SALES)
2 FROM PRODUCT
3 GROUP BY TYPE
4 HAVING SUM(SALES) >= 300
5 ORDER BY TYPE DESC;
```

- ① 'TYPE', 개수, 'SALES'의 합계를 표시합니다.
- ② <PRODUCT> 테이블을 대상으로 검색합니다.
- ③ 'TYPE'을 기준으로 그룹을 지정합니다.
- ④ 그룹 중 'SALES'의 합계가 300 이상인 자료만을 대상으로 합니다.
- ⑤ 'TYPE'을 기준으로 내림차순 정렬합니다.

- 49 프로시저의 실행 명령어는 EXECUTE, EXEC, CALL 3가지입니다. 문제에 제시된 코드를 살펴보면 다음과 같습니다.

```

1 CREATE PROCEDURE GET_DEPT(i IN INT) IS
2 BEGIN
3 IF (MOD(i, 10) = 2) THEN
4 INSERT INTO 회원(코드, 분류) VALUES(i, 'VIP');
5 ELSE
6 INSERT INTO 회원(코드, 분류) VALUES(i, '일반');
7 END IF;
8 END;
```


1회 대비

- ① 호출 프로그램이 전달한 값을 'i'에 저장한 후 사용하는 프로시저 'GET_DEPT'를 생성합니다.
- ② i를 10으로 나눈 나머지가 2이면 ③번을 수행하고 아니면 ④번을 수행합니다.
- ③ <회원> 테이블의 '코드', '분류' 속성에 각각 i의 값과 'VIP'를 입력합니다.
- ④ <회원> 테이블의 '코드', '분류' 속성에 각각 i의 값과 '일반'을 입력합니다.
- 50 테이블에 속한 튜플의 수를 카디널리티(Cardinality)라고 합니다. ①번은 속성의 수, ③번은 도메인(Domain), ④번은 속성(Attribute)에 대한 설명입니다.
- [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.
- 51 데이터 무결성은 애플리케이션(프로그램), 데이터베이스 트리거, 제약 조건을 이용하여 강화할 수 있습니다.
- [병행학습]**
데이터 무결성 강화 방법
- 애플리케이션
 - 데이터 생성, 수정, 삭제 시 무결성 조건을 검증하는 코드를 데이터를 조작하는 프로그램 내에 추가한다.
 - 데이터베이스에서는 데이터 무결성 제약을 정의할 수 없으므로 복잡한 규칙 등은 애플리케이션 내에서 처리한다.
 - **장점** : 사용자 정의 같은 복잡한 무결성 조건의 구현이 가능하다.
 - **단점** : 소스 코드에 분산되어 있어 관리가 힘들고, 개별적인 시행으로 인해 적정성 검토가 어렵다.
 - 데이터베이스 트리거
 - 트리거 이벤트에 무결성 조건을 실행하는 절차형 SQL을 추가한다.
 - **장점** : 통합 관리가 가능하고, 복잡한 요구 조건의 구현이 가능하다.
 - **단점** : 운영 중 변경이 어렵고, 사용상 주의가 필요하다.
 - 제약 조건
 - 데이터베이스에 제약 조건을 설정하여 무결성을 유지한다.
 - **장점** : 통합 관리 가능, 간단한 선언으로 구현 가능, 변경 용이, 오류 데이터 발생 방지 등이 있다.
 - **단점** : 복잡한 제약 조건의 구현과 예외적인 처리가 불가능하다.

- 52 카티션 프로덕트(Cartesian Product), 즉 교차곱은 두 릴레이션의 차수(Degree, 속성의 수)는 더하고, 카디널리티(튜플의 수)는 곱하면 됩니다. 즉 차수는 $3+5=8$ 이고, 카디널리티는 $4 \times 6 = 24$ 입니다.

- 53 3NF는 기본키가 아닌 모든 속성이 기본키에 대해 이행적 종속을 만족하지 않는 정규형입니다. ③번의 내용은 4NF에 대한 설명입니다.

[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
4NF
↓ 조인 종속성 이용
5NF

정규화 단계 암기 요령

정규화가 두부가게에 가서 모든 두부를 달라고 하니 주인이 깜짝 놀라 말했다.

두부이걸다줘? ⇨ 도부이겔다조

도메인이 원자값

부분적 함수 종속 제거

이행적 함수 종속 제거

결정자이면서 후보키가 아닌 것 제거

다치 종속

조인 종속성 이용

- 54 인덱스 대상 속성(컬럼) 선정 기준에서 인덱스 속성의 적절한 분포도는 10~15% 이내입니다.

[병행학습]**인덱스 대상 컬럼 선정 기준**

- 인덱스 컬럼의 분포도가 10~15% 이내인 컬럼
 - 분포도 = (컬럼값의 평균 Row 수 / 테이블의 총 Row 수) × 100
- 분포도가 10~15% 이상이어도 부분 처리를 목적으로 하는 컬럼
- 입·출력 장표 등에서 조회 및 출력 조건으로 사용되는 컬럼
- 인덱스가 자동 생성되는 기본키와 Unique키 제약 조건을 사용한 컬럼
- 가능한 한 수정이 빈번하지 않은 컬럼
- ORDER BY, GROUP BY, UNION이 빈번한 컬럼
- 분포도가 좁은 컬럼은 단독 인덱스로 생성
- 인덱스들이 자주 조합되어 사용되는 경우 하나의 결합 인덱스(Concatenate Index)로 생성

- 55 문제의 지문에 제시된 내용은 분산 데이터베이스의 4대 목표 중 중복 투명성(Replication Transparency)에 대한 설명입니다.

[병행학습]**분산 데이터베이스의 목표**

- **위치 투명성(Location Transparency)** : 액세스하려는 데이터베이스의 실제 위치를 알 필요 없이 단지 데이터베이스의 논리적인 명칭만으로 액세스할 수 있음
- **중복 투명성(Replication Transparency)** : 동일 데이터가 여러 곳에 중복되어 있더라도 사용자는 마치 하나의 데이터만 존재하는 것처럼 사용하고, 시스템은 자동으로 여러 자료에 대한 작업을 수행함
- **병행 투명성(Concurrency Transparency)** : 분산 데이터베이스와 관련된 다수의 트랜잭션들이 동시에 실행되더라도 그 트랜잭션의 결과는 영향을 받지 않음
- **장애 투명성(Failure Transparency)** : 트랜잭션, DBMS, 네트워크, 컴퓨터 장애에도 불구하고 트랜잭션을 정확하게 처리함

- 56 데이터베이스에서 데이터 저장 시 데이터의 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법을 클러스터(Cluster)라고 합니다.
- **인덱스(Index)** : 데이터 레코드를 빠르게 접근하기 위해 <키 값, 포인터> 쌍으로 구성되는 데이터 구조
 - **제약 조건(Constraint)** : 데이터베이스에 저장될 수 있는 실제 데이터의 논리적인 제약 조건
 - **파티션(Partition)** : 대용량의 테이블이나 인덱스를 작은 논리적 단위인 파티션으로 나누는 것

- 57 관계형 데이터베이스에서 하나의 릴레이션에는 중복된 튜플이 존재할 수 없으므로 모든 릴레이션에는 반드시 하나 이상의 후보키가 있어야 합니다.

[병행학습]

후보키(Candidate Key)

- 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용하는 속성들의 부분집합, 즉 기본키로 사용할 수 있는 속성들을 말한다.
 - 하나의 릴레이션내에서는 중복된 튜플들이 있을 수 없으므로 모든 릴레이션에는 반드시 하나 이상의 후보키가 존재한다.
 - 후보키는 릴레이션에 있는 모든 튜플에 대해서 유일성과 최소성을 만족시켜야 한다.
 - **유일성(Unique)** : 하나의 키 값으로 하나의 튜플만을 유일하게 식별할 수 있어야 한다.
 - **최소성(Minimality)** : 모든 레코드들을 유일하게 식별하는 데 꼭 필요한 속성으로만 구성되어야 한다.
- [예] <학생> 릴레이션에서 '학번'이나 '주민번호'는 다른 레코드를 유일하게 구별할 수 있는 기본키로 사용할 수 있으므로 후보키이다.

- 58 무결성 모델은 기밀성 모델에서 발생하는 불법적인 정보 변경을 방지하기 위해 무결성을 기반으로 개발된 모델로, 무결성 보장이 최우선인 모델입니다. 기밀성 보장이 최우선인 모델은 기밀성 모델입니다.

[병행학습]

접근통제

- 데이터가 저장된 객체와 이를 사용하려는 주체 사이의 정보 흐름을 제한하는 것이다.
- 접근통제는 데이터에 대해 다음과 같은 통제를 함으로써 자원의 불법적인 접근 및 파괴를 예방한다.
- 접근통제 기술에는 임의 접근통제(DAC), 강제 접근통제(MAC)가 있다.
- **임의 접근통제(DAC, Discretionary Access Control)** : 데이터에 접근하는 사용자의 신원에 따라 접근 권한을 부여하는 방식

- **강제 접근통제(MAC, Mandatory Access Control)** : 주체와 객체의 등급을 비교하여 접근 권한을 부여하는 방식
- 접근통제의 3요소는 접근통제 정책, 접근통제 메커니즘, 접근통제 보안모델이다.

- 59 서버와 저장장치를 연결하는 스토리지의 종류 중 하나로, 서버와 저장장치를 연결하는 전용 네트워크를 별도로 구성하는 방식을 SAN(Storage Area Network)이라고 합니다. DES는 암호화 알고리즘이고 나머지 보기는 다른 종류의 스토리지입니다.

[병행학습]

스토리지(Storage)의 종류

- **DAS(Direct Attached Storage)** : 서버와 저장장치를 전용 케이블로 직접 연결하는 방식
- **NAS(Network Attached Storage)** : 서버와 저장장치를 네트워크를 통해 연결하는 방식
- **SAN(Storage Area Network)** : DAS의 빠른 처리와 NAS의 파일 공유 장점을 혼합한 방식으로, 서버와 저장 장치를 연결하는 전용 네트워크를 별도로 구성하는 방식

- 60 서브타입 기준 테이블 변환은 각 서브타입 속성들의 선택 사양이 명확한 경우에 유리합니다.

[병행학습]

서브타입 기준 테이블 변환

- 서브타입 기준의 테이블 변환은 슈퍼타입 속성들을 각각의 서브타입에 추가하여 서브타입들을 개별적인 테이블로 만드는 것이다.
- 서브타입에 속성이나 관계가 많이 포함된 경우 적용한다.
- 장점
 - 각 서브타입 속성들의 선택 사양이 명확한 경우에 유리하다.
 - 처리할 때마다 서브타입 유형을 구분할 필요가 없다.
 - 여러 개의 테이블로 통합하므로 테이블당 크기가 감소하여 전체 테이블 스캔시 유리하다.
- 단점
 - 수행 속도가 감소할 수 있다.
 - 복잡한 처리를 하는 SQL의 통합이 어렵다.
 - 부분 범위에 대한 처리가 곤란해진다.
 - 여러 테이블을 통합한 뷰는 조회만 가능하다.
 - UID(Unique Identifier, 식별자)의 유지 관리가 어렵다

- 61 C언어에서 변수명은 대·소문자를 구분합니다.

[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.

- 62 C언어에서 제공하는 기억 클래스에는 자동 변수(Automatic Variable), 외부 변수(External Variable), 정적 변수(Static Variable), 레지스터 변수(Register Variable)가 있습니다.

[병행학습]

기억 클래스의 종류

- **자동 변수(Automatic Variable)** : 함수나 코드의 범위를 한정하는 블록 내에서 선언되는 변수

1회 대비

- **외부 변수(External Variable)** : 현재 파일이나 다른 파일에서 선언된 변수나 함수를 참조(reference)하기 위한 변수
- **정적 변수(Static Variable)** : 함수나 블록 내에서 선언하는 내부 정적 변수와 함수 외부에서 선언하는 외부 정적 변수
- **레지스터 변수(Register Variable)** : 레지스터 변수는 메모리가 아닌 CPU 내부의 레지스터에 기억영역을 할당받는 변수

63 do~while문은 실행할 문의 조건이 처음부터 거짓일 때도 while문 내부의 코드를 최소 한 번은 실행합니다. ①번의 설명에 해당하는 반복문에는 for문, while문이 있습니다.

64 포인터 변수는 필요에 의해 동적으로 할당되는 메모리 영역인 힙 영역에 접근하는 동적 변수입니다.

[병행학습]

포인터(Pointer)

- 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.

65 사용된 코드의 의미는 다음과 같습니다.

```
#include <stdio.h>
main( ) {
    ① int a = 1, sum = 0;
    ② while (a++ < 10)
    ③     sum += a * 3 >= sum ? a : 1;
    ④ printf("%d", sum);
}
```

- 정수형 변수 a와 sum을 선언한 후 각각 1과 0으로 초기화한다.
- a가 10보다 작은 동안 ③번을 반복하여 수행한다. 조건을 비교한 후 a는 1 증가한다.
- a*3이 sum보다 크면 a를 sum에 누적하고, 아니면 1을 sum에 누적한다.
- sum의 값을 출력한다.

반복문 실행에 따른 변수의 변화는 다음과 같습니다.

반복횟수	②의 조건	a	③의 조건	sum
		1		0
1	1<10 (참)	2	6>0 (참)	2
2	2<10 (참)	3	9>2 (참)	5
3	3<10 (참)	4	12>5 (참)	9
4	4<10 (참)	5	15>9 (참)	14
5	5<10 (참)	6	18>14 (참)	20
6	6<10 (참)	7	21>20 (참)	27
7	7<10 (참)	8	24>27 (거짓)	28
8	8<10 (참)	9	27>28 (거짓)	29
9	9<10 (참)	10	30>29 (참)	39
	10<10 (거짓)	11		

66 사용된 코드의 의미는 다음과 같습니다.

```
public class Test {
    public static void main(String[] args) {
        ① int array[] = { 1, 3, 5, 7, 9 };
        ② for (int a : array)
        ③     if(a%3==0) System.out.printf("%*");
    }
}
```

- 정수형 배열 array를 선언하고 다음과 같이 초기화한다.
array

1	3	5	7	9
---	---	---	---	---
- array 배열의 각 요소를 a에 저장하면서 배열의 요소 수만큼 ③번을 반복 수행한다.
- a를 3으로 나눈 나머지가 0이면 “*”을 출력하고, 아니면 ②번으로 돌아간다.

반복문 실행에 따른 변수의 변화는 다음과 같습니다.

반복횟수	a	a%3	출력
1	1	1	
2	3	0	*
3	5	2	*
4	7	1	*
5	9	0	**

67 서식 지정자 ‘%f’는 소수점을 포함하는 실수를 입·출력하는데 사용합니다. “programming”과 같은 문자열을 출력하기 위해서는 서식 지정자 ‘%s’를 사용해야 합니다.

[병행학습]

서식 문자열

서식 문자열	의미
%d	정수형 10진수를 입·출력하기 위해 지정한다.
%u	부호없는 정수형 10진수를 입·출력하기 위해 지정한다.
%o	정수형 8진수를 입·출력하기 위해 지정한다.
%x	정수형 16진수를 입·출력하기 위해 지정한다.
%c	문자를 입·출력하기 위해 지정한다.
%s	문자열을 입·출력하기 위해 지정한다.
%f	소수점을 포함하는 실수를 입·출력하기 위해 지정한다.
%e	지수형 실수를 입·출력하기 위해 지정한다.
%ld	long형 10진수를 입·출력하기 위해 지정한다.
%lo	long형 8진수를 입·출력하기 위해 지정한다.
%lx	long형 16진수를 입·출력하기 위해 지정한다.
%p	주소를 16진수로 입·출력하기 위해 지정한다.

68 단항 연산자에는 !, ~, ++, --, sizeof가 있습니다. &는 비트 끼리 and 연산을 수행하는 이항 연산자입니다.

[병행학습]

연산자의 종류

- **단항 연산자** : !(논리 not), ~(비트 not), ++(증가), --(감소), sizeof(기타)
- **이항 연산자**
 - 산술 연산자 : *, /, %(나머지), +, -
 - 시프트 연산자 : <<, >>
 - 관계 연산자 : <, <=, >=, >, ==(같다), !=(같지 않다)
 - 비트 연산자 : &(비트 and), ^(비트 xor), |(비트 or)
 - 논리 연산자 : &&(논리 and), ||(논리 or)
- **삼항 연산자** : (조건식) ? (참) : (거짓)

- 대입 연산자 : =, +=, -=, *=, /=, %=, <<=, >>= 등
- 순서 연산자 : ,

69 개발 언어의 선정 시 개발 언어의 성능보다는 개발 언어에 대한 개발자들의 이해도와 활용도를 먼저 고려해야 합니다.

[병행학습]

개발 언어의 선정 기준

적정성	개발하려는 소프트웨어의 목적에 적합해야 함
효율성	코드의 작성 및 구현이 효율적이어야 함
이식성	다양한 시스템 및 환경에 적용이 가능해야 함
친밀성	개발 언어에 대한 개발자들의 이해도와 활용도가 높아야 함
범용성	다른 개발 사례가 존재하고 여러 분야에서 활용되고 있어야 함

70 사용자와 시스템 간의 인터페이스를 제공하는 것은 셸(Shell)입니다.

[병행학습]

UNIX 시스템의 구성

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 보조 기억장치에서 교체 처리가 가능하다.
- 파이프라인 기능을 지원하고 입·출력 재지정을 통해 출력과 입력의 방향을 변경할 수 있다.
- 공용 Shell(Bourne Shell, C Shell, Korn Shell)이나 사용자 자신이 만든 Shell을 사용할 수 있다.

유틸리티(Utility Program)

- 일반 사용자가 작성한 응용 프로그램을 처리하는 데 사용한다.
- DOS에서의 외부 명령어에 해당된다.
- 유틸리티 프로그램에는 에디터, 컴파일러, 인터프리터, 디버거 등이 있다.

71 기억장치 관리 전략에는 반입(Fetch), 배치(Placement), 교체(Replacement) 전략이 있습니다. Paging 전략은 가상기억장치에서 사용되는 기법입니다.

[병행학습]

기억장치의 관리 전략

- **반입(Fetch) 전략** : 보조기억장치에 보관중인 프로그램이나 데이터를 언제 주기억장치로 적재할 것인지를 결정하는 전략
- **배치(Placement) 전략** : 새로 반입되는 프로그램이나 데이터를 주기억장치의 어디에 위치시킬 것인지를 결정하는 전략
- **교체(Replacement) 전략** : 주기억장치의 모든 영역이 이미 사용중인 상태에서 새로운 프로그램이나 데이터를 주기억

장치에 배치하려고 할 때, 이미 사용되고 있는 영역 중에서는 어느 영역을 교체하여 사용할 것인지를 결정하는 전략

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

참조페이지	0	1	2	3	2	1	4	0
페이지 프레임	0	0	0	3	3	3	3	3
		1	1	1	1	1	4	4
			2	2	2	2	2	0
부재 발생	●	●	●	●			●	●

※ ● : 페이지 부재 발생

참조 페이지가 페이지 테이블에 없을 경우 페이지 결함(부재)이 발생합니다. 초기에는 모든 페이지가 비어 있으므로 처음 0, 1, 2 페이지 적재 시 페이지 결함이 발생됩니다. FIFO 기법은 가장 먼저 들어와 있었던 페이지를 교체하는 기법이므로 참조 페이지 3을 참조할 때에는 0을 제거한 후 3을 가져오게 됩니다. 이러한 과정으로 모든 페이지에 대한 요구를 처리하고 나면 총 페이지 결함 발생 횟수는 6회입니다.

73 워킹 셋(Working Set)은 자주 참조하는 페이지들의 집합을 의미하는 것으로, 스래싱이 발생되지 않도록 워킹 셋을 계속하여 변경해 주기 때문에 프로세스 실행 중에 워킹 셋의 크기가 변경될 수 있습니다.

[병행학습]

워킹 셋(Working Set)

- 프로세스가 일정 시간 동안 자주 참조하는 페이지들의 집합이다.
- 데닝(Denning)이 제안한 프로그램의 움직임에 대한 모델로, 프로그램의 Locality 특징을 이용한다.
- 자주 참조되는 워킹 셋을 주기억장치에 상주시킴으로써 페이지 부재 및 페이지 교체 현상이 줄어들어 프로세스의 기억장치 사용이 안정된다.
- 시간이 지남에 따라 자주 참조하는 페이지들의 집합이 변화하기 때문에 워킹 셋은 시간에 따라 변경된다.

74 스레드(Thread)는 동일 프로세스 환경에서 서로 독립적인 다중 수행이 가능합니다.

[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 운용
 - 커널 수준의 스레드 : 운영체제의 커널에 의해 스레드를 운용
- **스레드 사용의 장점**
 - 하나의 프로세스를 여러 개의 스레드로 생성하여 병행성을 증진시킬 수 있다.

1회 대비

- 하드웨어, 운영체제의 성능과 응용 프로그램의 처리율을 향상시킬 수 있다.
- 응용 프로그램의 응답 시간(Response Time)을 단축시킬 수 있다.
- 실행 환경을 공유시켜 기억장소의 낭비가 줄어든다.
- 프로세스들 간의 통신이 향상된다.
- 스레드는 공통적으로 접근 가능한 기억장치를 통해 효율적으로 통신한다.

75 준비 상태에서 대기하고 있는 프로세스 중 하나가 스케줄링되어 중앙처리장치를 할당받아 실행 상태로 전이되는 과정을 디스패치(Dispatch)라고 합니다.

[병행학습]

프로세스 상태 전이 관련 용어

- **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), 보류, 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
- **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·출력 장치의 처리 속도를 보완하고 다중 프로그래밍 시스템의 성능을 향상시키기 위해 입·출력할 데이터를 직접 입·출력 장치에 보내지 않고 나중에 한꺼번에 입·출력하기 위해 디스크에 저장하는 과정
- **교통량 제어기(Traffic Controller)** : 프로세스의 상태에 대한 조사와 통보를 담당

76 IPv6은 16비트씩 8부분, 총 128비트로 구성되어 있습니다.

[병행학습]

IPv6(Internet Protocol version 6)

- 현재 사용하고 있는 IP 주소 체계인 IPv4의 주소 부족 문제를 해결하기 위해 개발되었다.
- 128비트의 긴 주소를 사용하여 주소 부족 문제를 해결할 수 있으며, IPv4에 비해 자료 전송 속도가 빠르다.
- 인증성, 기밀성, 데이터 무결성의 지원으로 보안 문제를 해결할 수 있다.
- IPv4와 호환성이 뛰어나다.
- 주소의 확장성, 융통성, 연동성이 뛰어나며, 실시간 흐름 제어로 향상된 멀티미디어 기능을 지원한다.
- Traffic Class, Flow Label을 이용하여 등급별, 서비스별로 패킷을 구분할 수 있어 품질 보장이 용이하다.

77 양단 간(End-to-End)의 통신(전송)을 지원하는 기능은 전송 계층(Transport Layer)의 대표적인 기능입니다.

[병행학습]

OSI 7계층

물리 계층	<ul style="list-style-type: none"> • 전송에 필요한 두 장치 간의 실제 접속과 절단 등 기계적, 전기적, 기능적, 절차적 특성에 대한 규칙을 정의함 • 물리적 전송 매체와 전송 신호 방식을 정의하며, RS-232C, X.21 등의 표준이 있음
데이터 링크 계층	<ul style="list-style-type: none"> • 두 개의 인접한 개방 시스템들 간에 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 함 • 송신 측과 수신 측의 속도 차이를 해결하기 위한 흐름 제어 기능을 함 • 프레임의 시작과 끝을 구분하기 위한 프레임의 동기화 기능을 함 • 오류의 검출과 회복을 위한 오류 제어 기능을 함 • 프레임의 순서적 전송을 위한 순서 제어 기능을 함 • HDLC, LAPB, LLC, LAPD, PPP 등의 표준이 있음
네트워크 계층	<ul style="list-style-type: none"> • 개방 시스템들 간의 네트워크 연결을 관리하는 기능과 데이터의 교환 및 중계 기능을 함 • 네트워크 연결을 설정, 유지, 해제하는 기능을 함 • 경로 설정(Routing), 데이터 교환 및 중계, 트래픽 제어, 패킷 정보 전송을 수행함 • 관련 표준으로는 X.25, IP 등이 있음
전송(트랜스포트) 계층	<ul style="list-style-type: none"> • 논리적 안정과 균일한 데이터 전송 서비스를 제공함으로써 종단 시스템(End-to-End) 간에 투명한 데이터 전송을 가능하게 함 • OSI 7계층 중 하위 3계층과 상위 3계층의 인터페이스(Interface)를 담당함 • 종단 시스템(End-to-End) 간의 전송 연결 설정, 데이터 전송, 연결 해제 기능을 함 • 주소 설정, 다중화, 오류 제어, 흐름 제어를 수행함 • TCP, UDP 등의 표준이 있음
세션 계층	<ul style="list-style-type: none"> • 송·수신측 간의 관련성을 유지하고 대화 제어를 담당하는 계층 • 대화(회화) 구성 및 동기 제어, 데이터 교환 관리 기능을 함 • 송·수신측의 대화(회화) 동기를 위해 전송하는 정보의 일정한 부분에 체크점을 두어 정보의 수신 상태를 체크하며, 이때의 체크점을 동기점이라고 함 • 동기점은 오류가 있는 데이터의 회복을 위해 사용하는 것으로, 종류에는 소동기점과 대동기점이 있음
표현 계층	<ul style="list-style-type: none"> • 응용 계층으로부터 받은 데이터를 세션 계층에 보내기 전에 통신에 적당한 형태로 변환하고, 세션 계층에서 받은 데이터는 응용 계층에 맞게 변환하는 기능을 함 • 서로 다른 데이터 표현 형태를 갖는 시스템 간의 상호 접속을 위해 필요한 계층 • 코드 변환, 데이터 암호화, 데이터 압축, 구문 검색, 정보 형식(포맷) 변환, 문맥 관리 기능을 함
응용 계층	<ul style="list-style-type: none"> • 사용자(응용 프로그램)가 OSI 환경에 접근할 수 있도록 서비스를 제공함 • 응용 프로세스 간의 정보 교환, 전자 사서함, 파일 전송 등의 서비스를 제공함

78 문제의 지문에 제시된 기능을 가지고 있는 프로토콜은 TCP(Transmission Control Protocol)입니다.

- **RTCP(Real-Time Control Protocol)** : RTP(Real-time Transport Protocol) 패킷의 전송 품질을 제어하기 위한 제어 프로토콜

- **FTP(File Transfer Protocol)** : 컴퓨터와 컴퓨터 또는 컴퓨터와 인터넷 사이에서 파일을 주고받을 수 있도록 하는 원격 파일 전송 프로토콜
- **UDP(User Datagram Protocol)** : 데이터 전송 전에 연결을 설정하지 않는 비연결형 서비스를 제공하는 프로토콜

[병행학습]

TCP/IP(Transmission Control Protocol/Internet Protocol)

- 인터넷에 연결된 서로 다른 기종의 컴퓨터들이 데이터를 주고받을 수 있도록 하는 표준 프로토콜이다.
- 1960년대 말 ARPA에서 개발하여 ARPANET(1972)에서 사용하기 시작했다.
- UNIX의 기본 프로토콜로 사용되었고, 현재 인터넷 범용 프로토콜로 사용된다.
- 다음과 같은 기능을 수행하는 TCP 프로토콜과 IP 프로토콜이 결합된 것을 의미한다.
- **TCP(Transmission Control Protocol)**
 - OSI 7계층의 전송 계층에 해당
 - 신뢰성 있는 연결형 서비스를 제공
 - 패킷의 다중화, 순서 제어, 오류 제어, 흐름 제어 기능을 제공
 - 스트림(Stream) 전송 기능 제공
 - TCP 헤더에는 Source/Destination Port Number, Sequence Number, Acknowledgment Number, Checksum 등이 포함된다.
- **IP(Internet Protocol)**
 - OSI 7계층의 네트워크 계층에 해당
 - 데이터그램을 기반으로 하는 비연결형 서비스를 제공
 - 패킷의 분해/조립, 주소 지정, 경로 선택 기능을 제공
 - 헤더의 길이는 최소 20Byte에서 최대 60Byte이다.
 - IP 헤더에는 Version, Header Length, Total Packet Length, Header Checksum, Source IP Address, Destination IP Address 등이 포함된다.

- 79 제어의 역흐름은 객체들의 제어를 개발자가 아닌 프레임워크가 담당합니다. ①번은 모듈화(Modularity), ②번은 재사용성(Reusability), ③번은 확장성(Extensibility)에 대한 내용입니다.

[병행학습]

프레임워크의 특성

- **모듈화(Modularity)** : 캡슐화를 통해 모듈화를 강화하고 설계 및 구현의 변경에 따른 영향을 최소화함으로써 소프트웨어의 품질을 향상시킴
- **재사용성(Reusability)** : 재사용 가능한 모듈들을 제공함으로써 개발자의 생산성을 향상시킴
- **확장성(Extensibility)** : 다형성(Polymorphism)을 통한 인터페이스 확장이 가능하여 다양한 형태와 기능을 가진 애플리케이션 개발이 가능함
- **제어의 역흐름(Inversion of Control)** : 개발자가 관리하고 통제해야 하는 객체들의 제어를 프레임워크에 넘김으로써 생산성을 향상시킴

- 80 외부 소프트웨어를 구매하여 사용하는 패키지 소프트웨어에 비해 직접 개발하는 전용 개발 소프트웨어가 많은 인력과 시간이 소요됩니다.

[병행학습]

패키지 소프트웨어

- **기능 요구사항** : 70% 이상 충족시키는 패키지 소프트웨어가 있는 경우 이용
- **안정성** : 품질이 검증되었고, 업계 표준 준용
- **라이선스** : 판매자

- **생산성** : 개발을 위한 인력과 시간이 절약됨
- **호환성** : 보장이 안 됨
- **유지보수** : 결함 발생 시 즉시 대응이 어려움

전용 개발 소프트웨어

- **기능 요구사항** : 모든 기능 요구사항 반영 가능
- **안정성** : 개발자의 역량에 따라 달라짐
- **라이선스** : 회사
- **생산성** : 개발을 위한 인력과 시간이 필요함
- **호환성** : 설계 단계부터 고려하여 개발
- **유지보수** : 결함 발생 시 즉시 대응이 가능

- 81 SPICE(ISO/IEC 15504)의 프로세스 범주에는 고객-공급자(Customer-Supplier), 공학(Engineering), 지원(Support), 관리(Management), 조직(Organization)이 있습니다.

[병행학습]

SPICE(ISO/IEC 15504)의 프로세스 범주

- **고객-공급자(Customer-Supplier) 프로세스**
 - 소프트웨어를 개발하여 고객에게 전달하는 것을 지원하고, 소프트웨어의 정확한 운용 및 사용을 위한 프로세스로 구성된다.
 - 구성 요소 : 인수, 공급, 요구 도출, 운영
 - 프로세스 수 : 10개
- **공학(Engineering) 프로세스**
 - 시스템과 소프트웨어 제품의 명세화, 구현, 유지보수를 하는데 사용되는 프로세스로 구성된다.
 - 구성 요소 : 개발, 소프트웨어 유지보수
 - 프로세스 수 : 9개
- **지원(Support) 프로세스**
 - 소프트웨어 생명 주기에서 다른 프로세스에 의해 이용되는 프로세스로 구성된다.
 - 구성 요소 : 문서화, 형상, 품질 보증, 검증, 확인, 리뷰, 감사, 품질 문제 해결
 - 프로세스 수 : 8개
- **관리(Management) 프로세스**
 - 소프트웨어 생명 주기에서 프로젝트 관리자에 의해 사용되는 프로세스로 구성된다.
 - 구성 요소 : 관리, 프로젝트 관리, 품질 및 위험 관리
 - 프로세스 수 : 4개
- **조직(Organization) 프로세스**
 - 조직의 업무 목적 수립과 조직의 업무 목표 달성을 위한 프로세스로 구성된다.
 - 구성 요소 : 조직 배치, 개선 활동 프로세스, 인력 관리, 기반 관리, 측정 도구, 재사용
 - 프로세스 수 : 9개

- 82 소프트웨어 개발 방법론의 테일러링 시 고려해야 할 외부적 요건에는 법적 제약사항과 표준 품질 기준이 있습니다. 보유 기술과 프로젝트 규모는 내부적 요건에 해당합니다.

[병행학습]

소프트웨어 개발 방법론 테일러링

- 프로젝트 상황 및 특성에 맞추어 정의된 소프트웨어 개발 방법론의 절차, 사용기법 등을 수정 및 보완하는 작업
- **고려해야 할 내부적 요건**
 - 목표 환경 : 시스템의 개발 환경과 유형이 서로 다른 경우
 - 요구사항 : 프로젝트의 생명 주기 활동에서 개발, 운영, 유지보수 등 프로젝트에서 우선적으로 고려할 요구사항이 서로 다른 경우
 - 프로젝트 규모 : 비용, 인력, 기간 등 프로젝트의 규모가 서로 다른 경우
 - 보유 기술 : 프로세스, 개발 방법론, 산출물 등이 서로

다른 경우

· **고려해야 할 외부적 요건**

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

· **테일러링 기법의 종류**

- 프로젝트 규모와 복잡도에 따른 테일러링 기법
- 프로젝트 구성원에 따른 테일러링 기법
- 팀내 방법론 지원에 따른 테일러링 기법
- 자동화에 따른 테일러링 기법

- 83 개발 기간은 소프트웨어 비용 결정 요소 중 생산성 요소에 해당합니다.

[병행학습]

소프트웨어 비용 결정 요소

· **프로젝트 요소**

- 제품 복잡도 : 소프트웨어의 종류에 따라 발생할 수 있는 문제점들의 난이도를 의미
- 시스템 크기 : 소프트웨어의 규모에 따라 개발해야 할 시스템의 크기를 의미
- 요구되는 신뢰도 : 일정 기간 내 주어진 조건하에서 프로그램이 필요한 기능을 수행하는 정도를 의미

· **자원 요소**

- 인적 자원 : 소프트웨어 개발 관련자들이 갖춘 능력 혹은 자질을 의미
- 하드웨어 자원 : 소프트웨어 개발 시 필요한 장비와 워드 프로세서, 프린터 등의 보조 장비를 의미
- 소프트웨어 자원 : 소프트웨어 개발 시 필요한 언어 분석기, 문서화 도구 등의 개발 지원 도구를 의미

· **생산성 요소**

- 개발자 능력 : 개발자들이 갖춘 전문지식, 경험, 이해도, 책임감, 창의력 등을 의미
- 개발 기간 : 소프트웨어를 개발하는 기간을 의미

- 84 전문가 감정 기법의 주관적인 편견을 보완하기 위해 많은 전문가의 의견을 종합하여 산정하는 비용 산정 기법은 델파이 기법입니다.

- **기능 점수(FP) 모형** : 총 기능 점수와 영향도를 이용하여 기능 점수(FP)를 구한 후 이를 이용해서 비용을 산정하는 기법
- **LOC 기법** : 소프트웨어 각 기능의 원시 코드 라인 수의 비관치, 낙관치, 기대치를 측정하여 예측치를 구하고 이를 이용하여 비용을 산정하는 기법
- **COCOMO 모형** : 원시 프로그램의 규모인 LOC(원시 코드 라인 수)에 의한 비용 산정 기법

[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.

- 85 · 노력 = LOC/1인당 월평균 생산 라인 수 = 25000/500 = 50
 · 개발 기간 = 노력/투입인원 = 50/5명 = 10개월
 ※ 단순히 '25,000라인 / (5명 * 500라인) = 10'으로 계산해도 됩니다.

- 86 생산성은 '원시 코드 라인 수/노력'이며, 노력은 소프트웨어를 한 달 간 개발하는 데 소요되는 총 인원 또는 한 사람을 기준으로 몇 개월에 걸쳐 개발했느냐를 나타냅니다. 이를 계산하는 방법은 '투입 인원 × 개발 기간'입니다.

- 87 NFC(Near Field Communication, 근거리 무선 통신)는 고주파(HF)를 이용한 근거리 무선 통신 기술입니다. ④번의 내용은 클라우드 컴퓨팅(Cloud Computing)에 대한 설명입니다.

- 88 LAN에서는 주로 버스(Bus)형이나 링(Ring)형 구조를 사용합니다. 망(Mesh)형은 주로 공중 통신망에서 사용됩니다.

[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

- 89 문제의 지문에서 설명하고 있는 소프트웨어는 그레이웨어(Grayware)입니다.

- **매시업(Mashup)** : 다수의 정보원이 제공하는 콘텐츠를 조합하여 하나의 서비스로 제공하는 웹 사이트 또는 애플리케이션
- **RIA(Rich Internet Application)** : 역동적이고 인터랙티브한 웹페이지를 제공하는 신개념의 플래시 웹페이지 제작 기술
- **뉴럴링크(Neuralink)** : 사람의 뇌와 컴퓨터를 결합하는 기술

- 90 현실의 사물을 소프트웨어로 가상화한 모델로, 현실의 사물을 대신해 다양한 상황을 모의 실험하기 위한 용도로 사용되는 것은 디지털 트윈(Digital Twin)입니다.

- **소프트웨어 에스크로(Software Escrow)** : 소프트웨어를 제3 기관에 임치하여 지식재산권을 보호하는 것
- **서비스형 소프트웨어(SaaS)** : 소프트웨어의 여러 기능 중에

서 사용자가 필요로 하는 서비스만 이용할 수 있도록 한 소프트웨어

- 오픈 그리드 서비스 아키텍처(OGSA) : 애플리케이션 공유를 위한 웹 서비스를 그리드 상에서 제공하기 위해 만든 개방형 표준

- 91 보안 가이드의 개발은 행정안전부가 아닌 한국인터넷진흥원(KISA)에서 수행해야 합니다.

[병행학습]

활동 주제별 개발 보안 역할

· 행정안전부

- 소프트웨어 개발 보안 정책을 총괄한다.
- 소프트웨어 개발 보안 관련 법규, 지침, 제도를 정비한다.
- 소프트웨어 보안 약점을 진단하는 사람의 양성 및 관련 업무를 수행한다.

· 한국인터넷진흥원(KISA)

- 소프트웨어 개발 보안 정책 및 가이드를 개발한다.
- 소프트웨어 개발 보안에 대한 기술을 지원하고, 교육과정 및 자격제도를 운영한다.

· 발주기관

- 소프트웨어 개발 보안의 계획을 수립한다.
- 소프트웨어 개발 보안 사업자 및 감리법인을 선정한다.
- 소프트웨어 개발 보안의 준수 여부를 점검한다.

· 사업자

- 소프트웨어 개발 보안 관련 기술 수준 및 적용 계획을 명시한다.
- 소프트웨어 개발 보안 관련 인력을 대상으로 교육을 실시한다.
- 소프트웨어 개발 보안 가이드를 참조하여 개발한다.
- 자체적으로 보안 약점을 진단하고 제거한다.
- 소프트웨어 보안 약점과 관련된 시정 요구사항을 이행한다.

· 감리법인

- 감리 계획을 수립하고 협의한다.
- 소프트웨어 보안 약점의 제거 여부 및 조치 결과를 확인한다.

- 92 소프트웨어의 안전 여부를 확인할 수 있도록 문서화하는 것은 구현 개발자의 역할입니다.

[병행학습]

설계자(Designer)가 수행해야 할 보안활동

- 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.

- 93 문제의 지문에서 괄호 안에 공통적으로 들어갈 용어는 메모리스터(Memristor)입니다.

- 엠디스크(Millennial DISC) : 한 번의 기록만으로 자료를 영구 보관할 수 있는 광 저장 장치
- 멤스(MEMS) : 기계 구조를 다양한 기술로 미세 가공하여 전기기계적 동작을 할 수 있도록 한 초미세 장치
- RAID(Redundant Array of Inexpensive Disk) : 오류 검출 및 복구를 위해 디스크 배열에 오류 검출 및 복구를 위한 여분의 디스크들을 추가하는 기술

- 94 내부 정보를 암호화 하는 방법은 암호화 분리이며, 암호화 분리는 Secure OS에서 구현되는 자원 보호 방법 중 구현이 가장 복잡합니다.

- ①번은 논리적 분리, ②번은 시간적 분리, ③번은 물리적 분리에 대한 내용입니다.

[병행학습]

Secure OS

- 기존의 운영체제(OS)에 내재된 보안 취약점을 해소하기 위해 보안 기능을 갖춘 커널을 이식하여 외부의 침입으로부터 시스템 자원을 보호하는 운영체제를 의미한다.
- 보안 커널은 보안 기능을 갖춘 커널을 의미하며, TCB를 기반으로 참조 모니터의 개념을 구현하고 집행한다.
- 보안 커널의 보호 대상에는 메모리와 보조기억장치, 그리고 그곳에 저장된 데이터, 하드웨어 장치, 자료 구조, 명령어, 각종 보호 메커니즘 등이 있다.
- 보호 방법(복잡한 것부터)

- 암호적 분리(Cryptographic Separation) : 내부 정보를 암호화하는 방법

- 논리적 분리(Logical Separation) : 프로세스의 논리적 구역을 지정하여 구역을 벗어나는 행위를 제한하는 방법

- 시간적 분리(Temporal Separation) : 동일 시간에 하나의 프로세스만 수행되도록 하여 동시 실행으로 발생하는 보안 취약점을 제거하는 방법

- 물리적 분리(Physical Separation) : 사용자별로 특정 장비만 사용하도록 제한하는 방법

- 95 병행제어 기법에서 병행수행되던 트랜잭션들 중 어느 하나에 문제가 생겨 Rollback하는 경우 다른 트랜잭션도 함께 Rollback되는 현상을 연쇄 복귀(Cascading Rollback)라고 합니다.

[병행학습]

병행수행의 문제점

- 갱신 분실(Lost Update) : 두 개 이상의 트랜잭션이 같은 자료를 공유하여 갱신할 때 갱신 결과의 일부가 없어지는 현상
- 비완료 의존성(Uncommitted Dependency) : 하나의 트랜잭션 수행이 실패한 후 회복되기 전에 다른 트랜잭션이 실패한 갱신 결과를 참조하는 현상
- 모순성(Inconsistency) : 두 개의 트랜잭션이 병행 수행될 때 원치 않는 자료를 이용함으로써 발생하는 문제
- 연쇄 복귀(Cascading Rollback) : 병행수행되던 트랜잭션들 중 어느 하나에 문제가 생겨 Rollback하는 경우 다른 트랜잭션도 함께 Rollback되는 현상

- 96 소프트웨어 개발이 있어 충족시켜야 할 3대 주요 보안 요소는 기밀성(Confidentiality), 무결성(Integrity), 가용성(Availability)이 있습니다.

[병행학습]

보안 요소

- 소프트웨어 개발에 있어 충족시켜야 할 요소 및 요건을 의미한다.
- 기밀성(Confidentiality) : 시스템 내의 정보와 자원은 인가된 사용자에게만 접근이 허용됨
- 무결성(Integrity) : 시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있음
- 가용성(Availability) : 인가받은 사용자는 언제라도 사용할 수 있음
- 인증(Authentication) : 시스템 내의 정보와 자원을 사용하려는 사용자가 합법적인 사용자인지를 확인하는 모든 행위
- 부인 방지(NonRepudiation) : 데이터를 송·수신한 자가 송·

수신 사실을 부인할 수 없도록 송·수신 증거를 제공함

- 97 데이터 표준화는 데이터 모델이나 DB에서 정의할 수 있는 모든 오브젝트를 대상으로 합니다.

[병행학습]

데이터 표준화

- 시스템을 구성하는 데이터 요소의 명칭, 정의, 형식, 규칙에 대한 원칙을 수립하고 적용하는 것을 의미한다.
- 데이터 모델이나 DB에서 정의할 수 있는 모든 오브젝트를 대상으로 데이터 표준화를 수행해야 한다.
- 데이터 표준의 종류

표준 단어	업무에서 사용하고 일정한 의미를 갖고 있는 최소 단위의 단어를 의미
표준 도메인	문자형, 숫자형, 날짜형, 시간형과 같이 컬럼을 성질에 따라 그룹핑한 개념
표준 코드	선택할 수 있는 값을 정형화하기 위해 기준에 맞게 이미 정의된 코드 값으로, 도메인의 한 유형
표준 용어	단어, 도메인, 코드 표준이 정의되면 이를 바탕으로 표준 용어를 구성

· 데이터 표준화의 대상

데이터 명칭	데이터를 유일하게 구별할 수 있는 유일성, 의미 전달의 충분성, 그리고 업무적 보편성을 갖는 이름으로 정의
데이터 정의	데이터를 제3자의 입장에서 쉽게 이해할 수 있도록 해당 데이터가 의미하는 범위 및 자격 요건 등을 규정
데이터 형식	업무 규칙 및 사용 목적과 유사한 데이터에 대해 일관되게 데이터 형식을 정의함으로써 데이터 입력 오류, 통제 위험 등을 최소화
데이터 규칙	기본 값, 허용 값, 허용 범위 등과 같이 발생할 수 있는 데이터 값을 사전에 지정함으로써 데이터의 정합성 및 완전성을 향상

- 98 임의의 길이의 입력 데이터나 메시지를 고정된 길이의 값이나 키(key)로 변환하는 알고리즘은 해시 알고리즘(Hash Algorithm)입니다.

- RSA(Rivest Shamir Adleman) : 큰 숫자는 소인수 분해하기 어렵다는 점을 기반으로 만들어진 공개키 암호화 알고리즘
- DES(Data Encryption Standard) : 미국 NBS에서 발표한 개인키 암호화 알고리즘
- PKI(Public Key Infrastructure, 공개키 기반 구조) : 공개키 암호 시스템을 안전하게 사용하고 관리하기 위한 정보 보호 표준 방식

[병행학습]

해시(Hash)

- 임의의 길이의 입력 데이터나 메시지를 고정된 길이의 값이나 키로 변환하는 것을 의미한다.
- 해시 알고리즘을 해시 함수라고 부르며, 해시 함수로 변환된 값이나 키를 해시값 또는 해시키라고 부른다.
- 데이터의 암호화, 무결성 검증을 위해 사용될 뿐만 아니라 정보보호의 다양한 분야에서 활용된다.
- 종류 : SHA 시리즈, MD5, N-NASH, SNEFRU 등

- 99 문제의 지문에 제시된 네트워크 공격 방법은 APT(Advanced Persistent Threats, 지능형 지속 위협)입니다.

- 스미싱(Smishing) : 문자 메시지(SMS)를 이용해 사용자의 개인 신용 정보를 빼내는 수법

- 쿼싱(Qshing) : QR코드를 통해 악성 앱의 다운로드를 유도하거나 악성 프로그램을 설치하도록 하는 금융사기 기법
- 분산 서비스 거부 공격(Distributed Denial of Service Attack) : 여러 곳에 분산된 공격 지점에서 한 곳의 서버에 대해 대량의 트래픽을 집중하여 자원을 고갈시키는 기법

- 100 ②번의 내용은 침입 방지 시스템에 대한 설명입니다.

- 침입 방지 시스템(IPS; Intrusion Prevention System) : 방화벽과 침입 탐지 시스템을 결합한 보안 솔루션

[병행학습]

침입 탐지 시스템(IDS; Intrusion Detection System)

- 컴퓨터 시스템의 비정상적인 사용, 오용, 남용 등을 실시간으로 탐지하는 시스템이다.
- 방화벽과 같은 침입 차단 시스템만으로는 내부 사용자의 불법적인 행동과 외부 해킹에 100% 완벽하게 대처할 수는 없다.
- 문제가 발생한 경우 모든 내·외부 정보의 흐름을 실시간으로 차단하기 위해 해커 침입 패턴에 대한 추적과 유해 정보 감시가 필요하다.
- 오용 탐지(Misuse Detection) : 미리 입력해 둔 공격 패턴이 감지되면 이를 알려준다.
- 이상 탐지(Anomaly Detection) : 평균적인 시스템의 상태를 기준으로 비정상적인 행위나 자원의 사용이 감지되면 이를 알려준다.
- 침입 탐지 시스템의 위치
 - 패킷이 라우터로 들어오기 전 : 네트워크에 시도되는 모든 공격을 탐지할 수 있음
 - 라우터 뒤 : 라우터에 의해 패킷 필터링을 통과한 공격을 탐지할 수 있음
 - 방화벽 뒤 : 내부에서 외부로 향하는 공격을 탐지할 수 있음
 - 내부 네트워크 : 내부에서 내부 네트워크의 해킹 공격을 탐지할 수 있음
 - DMZ : DMZ는 외부 인터넷에 서비스를 제공하는 서버가 위치하는 네트워크로, 강력한 외부 공격이나 내부 공격으로부터 중요 데이터를 보호하거나 서버의 서비스 중단을 방지할 수 있음