



시험에 나오는 것만 공부한다!

시나공시리즈

모의고사 & 정답 및 해설 2020년 2회 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. 애자일 모형의 실무 적용 시 준수해야 할 사항이 아닌 것은?

- ① 유용한 소프트웨어를 빠르고, 지속적으로 제공하여 고객을 만족시킨다.
- ② 개발 막바지라도 요구사항 변경을 적극 수용한다.
- ③ 최소한 몇 개월 단위로 실행되는 소프트웨어를 제공한다.
- ④ 기술적 우수성과 좋은 설계에 지속적인 관심을 기울이면 민첩성이 향상된다.

2. 다음이 설명하고 있는 생명주기 모델은?

- 여러 번의 개발 과정을 거쳐 완벽한 최종 소프트웨어를 개발한다.
- 점진적으로 개발 과정이 반복되므로 누락되거나 추가된 요구사항을 첨가할 수 있다.

- ① 애자일 모형(Agile Model)
- ② 프로토타입 모형(Prototype Model)
- ③ 나선형 모형(Spiral Model)
- ④ 폭포수 모형(Waterfall Model)

3. 다음 중 스크럼(Scrum)에 대한 설명으로 잘못된 것은?

- ① 팀원들이 스스로 팀을 구성하며, 개발 작업의 모든 것을 스스로 해결할 수 있어야 한다.
- ② 개발에 필요한 요구사항에 우선순위를 부여한 제품기능 목록(Product Backlog)을 작성한다.
- ③ 소멸(Burn-down) 차트를 통해 작업의 진행 상황을 확인할 수 있다.
- ④ 스프린트 검토 회의(Sprint Review)를 통해 스프린트 동안 발생한 문제점을 파악하고 이에 대한 해결 방안을 모색한다.

4. 다음 중 XP의 주요 실천 방법에 대한 설명 중 옳지 않은 것은?

- ① 개발자가 실제 코드를 작성하기 전에 테스트 케이스를 먼저 작성한다.
- ② 릴리즈 기간을 길게 반복하여 고객의 요구 변화에 신속히 대응한다.
- ③ 모듈 단위로 나눠서 개발된 코드들은 하나의 작업이 마무리될 때마다 지속적으로 통합된다.
- ④ 다른 사람과 함께 프로그래밍을 수행함으로써 개발에 대한 책임을 공동으로 나눠 갖는 환경을 조성한다.

5. 다음 UML의 구성 요소 중 관계에 대한 설명으로 옳지 않은 것은?

- ① 포함(Composition) 관계 : 하나의 사물이 다른 사물에 포함되어 있는 관계로 두 사물이 서로 독립적이다.
- ② 실체화(Realization) 관계 : 사물이 할 수 있거나 해야 하는 기능으로 서로를 그룹화 할 수 있는 관계를 표현한다.
- ③ 일반화(Generalization) 관계 : 하나의 사물이 다른 사물에 비해 더 일반적인지 구체적인지를 표현한다.
- ④ 의존(Dependency) 관계 : 사물 사이에 서로 연관은 있으나 필요에 의해 서로에게 영향을 주는 짧은 시간 동안만 연관을 유지하는 관계를 표현한다.

6. 다음이 설명하고 있는 UML 다이어그램은?

결과물, 프로세스, 컴포넌트 등 물리적인 자원의 위치를 표시하는 것으로, 구현 단계에서 사용된다.

- ① 배치 다이어그램(Deployment Diagram)
- ② 순차 다이어그램(Sequence Diagram)
- ③ 객체 다이어그램(Object Diagram)
- ④ 상태 다이어그램(State Diagram)

7. 다음 중 사용자 인터페이스(UI)를 설계할 때 고려할 사항으로 거리가 먼 것은?

- ① 사용자가 쉽게 이해하고 편리하게 사용할 수 있는 환경을 제공해야 한다.
- ② 메인 화면에 주요 기능을 노출시켜 최대한 조작이 쉽도록 설계해야 한다.
- ③ 버튼이나 조작 방법 등을 새로운 방식으로 제공해야 한다.
- ④ 사용자가 개념적으로 쉽게 인지할 수 있도록 설계해야 한다.

8. 다음 중 사용자 인터페이스를 설계할 때 사용하는 도구로 옳지 않은 것은?

- ① 와이어프레임(Wireframe)
- ② 이클립스(Eclipse)
- ③ 유스케이스(Use Case)
- ④ 스토리보드(Story Board)

9. 다음 중 유스케이스(Use Case)에 관한 설명으로 잘못된 것은?

- ① 완성된 유스케이스에 대해 유스케이스 명세서를 작성한다.
- ② 사용자의 요구사항을 정리하고 기록하기 위한 도구이다.
- ③ 유스케이스는 일반적으로 다이어그램 형식으로 작성된다.
- ④ 현재 진행 상태 등을 공유하기 위해 사용한다.

10. 소프트웨어 아키텍처 설계의 기본 원리 중 추상화(Abstraction)에 대한 설명으로 가장 옳지 않은 것은?

- ① 문제의 전체적이고 포괄적인 개념을 설계한 후 차례로 세분화하여 구체화시켜 나가는 것이다.
- ② 추상화는 최소의 비용으로 실제 상황에 대처할 수 있고, 시스템의 구조 및 구성을 대략적으로 파악할 수 있게 해준다.
- ③ 데이터 추상화는 자세한 수행 과정을 정의하지 않고, 전반적인

2회 대비

흐름만 파악할 수 있게 설계하는 방법이다.

- ④ 제어 추상화는 이벤트 발생의 정확한 절차나 방법을 정의하지 않고, 대표할 수 있는 표현으로 대체하는 방법이다.

11. 다음 중 UI 프로토타입의 장·단점에 대한 설명으로 잘못된 것은?

- ① 사용자를 설득하고 이해시키기 쉽다.
 ② 사전에 오류를 발견할 수 있다.
 ③ 반복적인 개선 및 보완 작업이 없어 작업 시간을 최소화할 수 있다.
 ④ 부분적으로 프로토타이핑을 진행하다보면 중요한 작업이 생략될 수 있다.

12. 소프트웨어 아키텍처의 설계 과정을 순서대로 가장 옳게 나열한 것은?

- ㉠ 시스템과 서브시스템의 타입을 결정한다.
 ㉡ 서브시스템의 기능과 서브시스템 간의 인터페이스를 정의한다.
 ㉢ 표준 아키텍처를 설계한다.
 ㉣ 요구사항을 분석하여 전체 시스템의 설계 목표를 설정한다.

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

13. 다음 중 아키텍처 패턴(Architecture Pattern)에 관한 설명 중 가장 옳지 않은 것은?

- ① 검증된 구조로 개발하기 때문에 오류가 적어 개발시간을 단축할 수 있다.
 ② 시스템의 구조를 이해하는 것이 어려워 개발에 참여한 개발자에 의해서만 유지보수가 수행될 수 있다.
 ③ 서브시스템들에 대한 역할을 정의하고 있으며, 그들 간의 인터페이스에 대한 지침 등이 포함되어 있다.
 ④ 이해관계자들이 공통된 아키텍처를 공유할 수 있어 의사소통이 간편해진다.

14. 다음 중 객체(Object)에 대한 특성을 설명한 것으로 가장 옳지 않은 것은?

- ① 식별성을 가진다.
 ② 상호 연관성에 의한 관계가 형성된다.
 ③ 행위에 대한 특징은 나타낼 수 없다.
 ④ 일정한 기억장소를 가지고 있다.

15. 다음이 설명하고 있는 결합도(Coupling)는?

모듈이 파라미터나 인수로 다른 모듈에게 데이터를 넘겨주고 호출 받은 모듈은 받은 데이터에 대한 처리 결과를 다시 돌려주는 유형의 모듈 결합도이다.

- ① 내용 결합도(Content Coupling)
 ② 외부 결합도(External Coupling)
 ③ 제어 결합도(Control Coupling)
 ④ 데이터 결합도(Data Coupling)

16. 다음 중 모듈의 응집도(Cohesion)에 대한 설명 중 틀린 것은?

- ① 시간적 응집도는 특정 시간에 처리되는 몇 개의 기능을 모아 하나의 모듈로 작성할 경우의 응집도이다.
 ② 우연적 응집도는 한 모듈 내부의 한 기능 요소에 의한 출력 자료가 다음 기능 요소의 입력 자료로서 제공되는 형태의 응집도이다.
 ③ 기능적 응집도는 모듈 내부의 모든 기능 요소들이 단일 문제와 연관되어 수행될 경우의 응집도이다.
 ④ 논리적 응집도는 유사한 성격을 갖거나 특정 형태로 분류되는

처리 요소들로 하나의 모듈이 형성되는 경우의 응집도이다.

17. 운영체제와 해당 운영체제에서 실행되는 응용 프로그램 사이에서 운영체제가 제공하는 서비스 이외에 추가적인 서비스를 제공하는 소프트웨어를 무엇이라 하는가?

- ① Middleware ② ORM
 ③ OOP ④ Interface

18. 송수신 데이터의 처리 방식 중 대량의 데이터를 처리할 때 사용하는 방식은?

- ① 실시간 방식 ② 분산 처리 방식
 ③ 배치 방식 ④ 지연 처리 방식

19. 다음 중 미들웨어(Middleware)들의 개별적인 특징으로 옳지 않은 것은?

- ① RPC : 항공기나 철도 예약 업무 등과 같은 온라인 트랜잭션 업무에서 트랜잭션을 처리 및 감시하는 미들웨어이다.
 ② MOM : 메시지 기반의 비동기형 메시지를 전달하는 방식의 미들웨어이다.
 ③ ORB : 객체 지향 미들웨어로, 코바(CORBA) 표준 스펙을 구현한 미들웨어이다.
 ④ DB : 데이터베이스 벤더에서 제공하는 클라이언트에서 원격의 데이터베이스와 연결하기 위한 미들웨어이다.

20. 다음 중 GoF의 디자인 패턴에 대한 설명으로 틀린 것은?

- ① 수많은 디자인 패턴들 중 가장 일반적인 사례에 적용될 수 있는 패턴들을 분류하여 정리한 것이다.
 ② 디자인 패턴은 서브시스템에 속하는 컴포넌트들과 그 관계를 설계하기 위한 참조 모델을 말한다.
 ③ 디자인 패턴은 아키텍처 패턴보다 상위 수준의 설계에 사용된다.
 ④ 디자인 패턴은 유형에 따라 생성, 구조, 행위로 구분한다.

제2과목 소프트웨어 개발

21. 자료들을 반드시 연속적으로 배열시키지는 않고 임의의 기억공간에 기억시키되, 자료 항목의 순서에 따라 노드의 포인터 부분을 이용하여 서로 연결시킨 자료 구조를 무엇이라고 하는가?

- ① 배열(Array)
 ② 연속 리스트(Contiguous List)
 ③ 연결 리스트(Linked List)
 ④ 스택(Stack)

22. 데이터베이스의 정의 중 통합된 데이터(Integrated Data)에 대한 설명으로 옳은 것은?

- ① 컴퓨터가 접근할 수 있는 저장 매체에 저장된 자료이다.
 ② 자료의 중복을 배제한 데이터의 모임이다.
 ③ 조직의 고유한 업무를 수행하는 데 존재 가치가 확실하고 없어서는 안 될 반드시 필요한 자료이다.
 ④ 여러 응용 시스템들이 공동으로 소유하고 유지하는 자료이다.

23. 다음 중 소프트웨어 설치 매뉴얼에 포함될 내용에 해당하지 않는 것은?

- ① 소프트웨어 사용 시 함께 실행하면 안 되는 응용 프로그램의 목록
 ② 설치 방법이나 설치 환경이 잘못된 경우 표시될 수 있는 메시지에 대한 설명
 ③ 설치 전 사용자의 설치 환경에 따라 점검해야 할 사항들

- ④ 설치와 관련하여 기술적인 지원이나 소프트웨어에 대한 서비스를 원할 경우 문의할 수 있는 연락처

24. 다음 중 인수 테스트(Acceptance Test)에 대한 설명으로 가장 옳지 않은 것은?

- ① 사용자의 요구사항을 충족하는지에 중점을 두고 테스트한다.
 ② 문제가 없으면 사용자는 소프트웨어를 인수하게 되고, 프로젝트는 종료된다.
 ③ 개발한 소프트웨어가 인수되기 전 개발자가 수행하는 테스트이다.
 ④ 종류에는 사용자 인수 테스트, 알파 테스트, 베타 테스트 등이 있다.

25. 다음 중 테스트 시나리오의 작성 방법에 대한 설명 중 틀린 것은?

- ① 소프트웨어를 하나의 시나리오로 통합하여 작성해야 한다.
 ② 사용자의 요구사항과 설계 문서 등을 토대로 작성해야 한다.
 ③ 유스케이스(Use Case) 간 업무 흐름이 정상적인지 확인할 수 있도록 작성해야 한다.
 ④ 모듈 간의 연계가 정상적인지 확인할 수 있도록 작성해야 한다.

26. 다음 중 결함 관리 프로세스에 대한 설명으로 가장 옳지 않은 것은?

- ① 전체 프로세스에 대한 결함 관리 일정, 인력, 업무 프로세스 등을 확보하여 계획을 수립해야 한다.
 ② 결함 관리 DB를 구축하여 발견된 결함을 관리한다.
 ③ 결함은 테스터, 개발팀 리더, QA 담당자 등이 검토한 후 개발자에게 전달된다.
 ④ DB에 기록된 결함들은 '수정 → 검토 → 재확인' 순서로 처리된다.

27. 애플리케이션의 성능을 측정하기 위한 테스트 중 하나로, 애플리케이션이 과부하 상태에서 어떻게 작동하는지 확인하는 테스트는 무엇인가?

- ① 부하 테스트 ② 스트레스 테스트
 ③ 연계 테스트 ④ 침투 테스트

28. 다음 중 연계 테스트에 대한 설명으로 틀린 것은?

- ① 연계 시스템과 송·수신 모듈, 서버 등이 정상적으로 동작하는지 확인하는 활동이다.
 ② 연계 응용프로그램의 단위 테스트 케이스와 연계 테스트 케이스를 작성해야 한다.
 ③ 송·수신 시스템 간의 연계 테스트를 먼저 수행한 후 연계 응용프로그램의 단위 테스트를 수행한다.
 ④ 연계 서버 또는 송·수신용 어댑터, 연계를 위한 IP 및 포트 개방 등의 환경을 구축해야 한다.

29. 다음 설명에 해당하는 소프트웨어는 무엇인가?

접속자, 자원 현황, 트랜잭션 수행 내역, 장애 진단 등 다양한 모니터링 기능을 제공하는 소프트웨어로, 리소스 방식과 엔드투엔드(end-to-end) 방식 두 가지 유형으로 구분된다.

- ① APM ② ESB
 ③ EAI ④ Optimizer

30. 다음 중 데이터의 보안, 무결성, 회복, 병행 수행 제어 등을 정의하는데 사용되는 SQL 언어를 무엇이라 하는가?

- ① RDB(Relational Database)
 ② DDL(Data Define Language)
 ③ DML(Data Manipulation Language)
 ④ DCL(Data Control Language)

31. 디지털 저작권 관리(DRM)에 대한 다음 설명에서 괄호에 들어갈 가장 적합한 용어는 무엇인가?

()는 저작권에 대한 사용 권한, 라이선스 발급, 사용량에 따른 결제 관리 등을 수행하는 곳으로, 배포하려는 소프트웨어의 패키징을 수행하면 콘텐츠에는 암호화된 저작권자의 전자서명이 포함되고 저작권자가 설정한 라이선스 정보가 ()에 등록된다.

- ① 패키저(Packager)
 ② 클리어링 하우스(Clearing House)
 ③ 콘텐츠 분배자(Contents Distributor)
 ④ DRM 컨트롤러(DRM Controller)

32. 소프트웨어 버전 관리 도구 중 분산 저장소 방식에 대한 설명으로 틀린 것은?

- ① 버전 관리 자료가 하나의 원격 저장소와 분산된 개발자 PC의 로컬 저장소에 함께 저장되어 관리되는 방식이다.
 ② 로컬 저장소에서 버전 관리가 가능하므로 원격 저장소에 문제가 생겨도 로컬 저장소의 자료를 이용하여 작업 할 수 있다.
 ③ 파일을 잘못 복사하거나 다른 위치로 복사하는 것에 대비하기 위해 파일의 변경 사항을 데이터베이스에 기록하여 관리한다.
 ④ 종류에는 Git, GNU arch, DCVS, Bazaar 등이 있다.

33. 다음 중 블랙박스(Black Box) 테스트에 대한 설명으로 가장 옳지 않은 것은?

- ① 각 기능이 완전히 작동되는 것을 입증하는 테스트로, 기능 테스트라고도 한다.
 ② 소프트웨어 인터페이스에서 실시되는 테스트이다.
 ③ 프로그램의 제어 구조에 따라 선택, 반복 등의 분기점 부분들을 수행함으로써 논리적 경로를 제어한다.
 ④ 종류에는 동치 분할 검사, 경계값 분석, 원인-효과 그래프 검사, 오류 예측 검사, 비교 검사 등이 있다.

34. 다음 설명에 해당하는 테스트의 종류는 무엇인가?

대표적인 화이트박스 테스트 기법의 하나로, 테스트 케이스 설계자가 절차적 설계의 논리적 복잡성을 측정할 수 있게 해주는 테스트이다. 이 테스트의 결과는 실행 경로의 기초를 정의하는 지침으로 사용된다.

- ① 기초 경로 검사 ② 제어 구조 검사
 ③ 동치 분할 검사 ④ 오류 예측 검사

35. 다음 중 하향식 통합 테스트(Top Down Integration Test)에 대한 설명으로 가장 옳지 않은 것은?

- ① 상위 모듈에서 하위 모듈 방향으로 통합하면서 테스트하는 기법이다.
 ② 테스트 초기부터 사용자에게 시스템 구조를 보여줄 수 있다.
 ③ 상위 모듈에서는 테스트 케이스를 사용하기 어렵다.
 ④ 주요 제어 모듈의 종속 모듈들은 클러스터(Cluster)로 대체한다.

36. 다음 설명에 해당하는 모듈 연계 방법은 무엇인가?

• 기업 내 각종 애플리케이션 및 플랫폼 간의 정보 전달, 연계, 통합 등 상호 연동이 가능하게 해주는 솔루션이다.
 • 비즈니스 간 통합 및 연계성을 증대시켜 효율성 및 각 시스템 간의 확정성(Determinacy)을 높인다.
 • 서비스 보다는 애플리케이션 중심의 통합이 이루어진다.

- ① ESB(Enterprise Service Bus)
 ② EAI(Enterprise Application Integration)
 ③ IDE(Integrated Development Environment)
 ④ Middleware

37. 다음 중 소프트웨어 패키징에 대한 설명으로 가장 옳지 않은 것은?

- ① 모듈별로 생성한 실행 파일들을 묶어 배포용 설치 파일을 만드는 것이다.
- ② 다양한 환경에서 소프트웨어를 손쉽게 사용할 수 있도록 일반적인 배포 형태로 패키징한다.
- ③ 다양한 사용자의 요구사항을 반영할 수 있도록 패키징의 변경 및 개선에 대한 관리를 항상 고려한다.
- ④ 개발자를 중심으로 작업이 진행되므로 개발자의 편의성 및 실행 환경을 우선적으로 고려해야 한다.

38. 애플리케이션 테스트에서 동일한 테스트를 반복하면 더 이상 결함이 발견되지 않는 현상을 가리키는 용어는?

- ① 결함 집중(Defect Clustering)
- ② 파레토 법칙(Pareto Principle)
- ③ 살충제 패러독스(Pesticide Paradox)
- ④ 오류-부재의 켈변(Absence of Errors Fallacy)

39. 다음 중 인터페이스 보안 기능에 대한 설명으로 가장 옳지 않은 것은?

- ① 인터페이스 기능과 보안 취약점을 기반으로 네트워크, 애플리케이션, 데이터베이스 영역에 인터페이스 보안 기능을 적용한다.
- ② 데이터 탈취 및 변조 위협을 방지하기 위해 네트워크 트래픽에 대한 암호화를 설정한다.
- ③ 애플리케이션 코드 상의 보안 취약점을 보완하여 애플리케이션 영역에 보안 기능을 적용한다.
- ④ 데이터베이스에서 민감한 데이터의 암호화를 위해 IPsec, SSL, S-HTTP 등의 방법을 사용한다.

40. 코드의 자동 생성, 컴파일, 빌드, 디버그, 라이브러리 관리 등 개발을 위한 다양한 기능이나 툴을 하나의 인터페이스로 제공하는 소프트웨어는 무엇인가?

- ① 빌드 도구(Build Tool)
- ② 통합 개발 환경(IDE) 도구
- ③ 형상 관리(Configuration Management) 도구
- ④ 그룹웨어(Groupware)

제3과목 데이터베이스 구축

41. 다음 중 데이터베이스 설계에 대한 설명으로 잘못된 것은?

- ① 사용자의 요구를 분석하여 특정 DBMS로 데이터베이스를 구축하는 과정이다.
- ② 무결성, 일관성 회복, 보안 등의 다양한 요소들을 고려해야 한다.
- ③ 데이터 모델링은 현실의 자료를 논리적 자료 구조로 변환시키는 과정이다.
- ④ 데이터 구조화는 트랜잭션의 인터페이스와 테이블을 설계하는 단계이다.

42. 다음 중 데이터 모델의 개체(Entity)에 대한 설명으로 틀린 것은?

- ① 속성은 개체가 가지고 있는 특성이다.
- ② 개체 타입은 속성으로만 기술된 개체의 정의를 의미한다.
- ③ 개체 인스턴스는 개체를 구성하고 있는 각 속성들이 값을 가져 하나의 개체를 나타내는 것을 의미한다.
- ④ 속성을 개체 어커런스(Occurrence)라고도 한다.

43. 다음 중 릴레이션(Relation)의 특징에 대한 설명으로 가장 옳지 않은 것은?

- ① 한 릴레이션에는 동일한 튜플이 존재할 수 있다.
- ② 속성의 명칭은 유일해야 하지만, 속성을 구성하는 값은 동일한 값이 있을 수 있다.
- ③ 속성의 값은 논리적으로 더 이상 쪼갤 수 없는 원자값만을 저장한다.
- ④ 릴레이션 스키마를 구성하는 속성들 간의 순서는 중요하지 않다.

44. 다음 설명에 해당하는 키의 종류는 무엇인가?

- 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미한다.
- 참조되는 릴레이션의 기본키와 대응되어 릴레이션 간에 참조 관계를 표현하는데 중요한 도구이다.
- 참조 릴레이션의 기본키에 없는 값은 입력할 수 없다.

- ① 후보키(Candidate Key)
- ② 외래키(Foreign Key)
- ③ 기본키(Primary Key)
- ④ 슈퍼키(Super Key)

45. <ORDERS> 릴레이션과 <CUSTOMERS> 릴레이션에 공통으로 존재하는 'OC' 속성을 기준으로 두 릴레이션을 합치고자 할 때, 이를 관계대수식으로 올바르게 표현한 것은?

- ① $ORDERS \bowtie_{OC=OC} CUSTOMERS$
- ② $\pi_{OC}(ORDERS) \cap \pi_{OC}(CUSTOMERS)$
- ③ $ORDERS[OC \div OC] CUSTOMERS$
- ④ $\pi_{OC}(ORDERS) \cup \pi_{OC}(CUSTOMERS)$

46. 다음 중 제 3정규형에서 BCNF 수행 시의 작업으로 옳은 것은?

- ① 이행적 함수 종속성 제거
- ② 다치 종속 제거
- ③ 모든 결정자가 후보키가 되도록 분해
- ④ 부분 함수 종속성 제거

47. 의도적으로 정규화 원칙을 위배하여 시스템의 성능 향상, 개발 및 운영의 편의성 등을 위해 정규화된 데이터 모델을 통합, 중복, 분리하는 과정을 무엇이라 하는가?

- ① 정규화(Normalization)
- ② 반정규화(Denormalization)
- ③ 조인(Join)
- ④ 분할(Partitioning)

48. 다음 중 인덱스(Index) 설계 시 고려해야 할 사항으로 가장 옳지 않은 것은?

- ① 새로 추가되는 인덱스는 기존 액세스 경로에 영향을 미칠 수 있다.
- ② 인덱스를 지나치게 많이 만들면 오버헤드가 발생한다.
- ③ 인덱스를 위한 추가적인 저장 공간은 필요하지 않다.
- ④ 인덱스와 테이블 데이터의 저장 공간이 분리되도록 설계해야 한다.

49. 다음 설명에 해당하는 데이터베이스 개체는 무엇인가?

- 사용자에게 접근이 허용된 자료만을 제한적으로 보여주 기 위해 하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블이다.
- 기본 테이블로부터 유도된 테이블이기 때문에 기본 테이블과 같은 형태의 구조를 사용하며, 조작도 기본 테이블과 거의 같다.

- ① 뷰(View)
- ② 인덱스(Index)
- ③ 클러스터(Cluster)
- ④ 파티션(Partition)

2회 대비

50. 다음 중 클러스터(Cluster)에 대한 설명으로 가장 옳지 않은 것은?

- ① 데이터 액세스 효율을 향상시키기 위해 데이터 접근이 많은 테이블을 여러 곳에 분산 저장하는 물리적 저장 방법이다.
- ② 클러스터링으로 지정된 컬럼 값의 순서대로 저장된다.
- ③ 여러 개의 테이블이 하나의 클러스터에 저장된다.
- ④ 클러스터링 된 테이블은 데이터 조회 속도가 향상되지만 데이터 입력, 수정, 삭제에 대한 성능은 저하된다.

51. 범위 분할로 분할한 다음 해시 함수를 적용하여 다시 분할하는 파티셔닝 방식으로, 범위 분할한 파티션이 너무 커서 관리가 어려울 때 유용한 파티셔닝 방식은 무엇인가?

- ① 범위 분할(Range Partitioning)
- ② 해시 분할(Hash Partitioning)
- ③ 조합 분할(Composite Partitioning)
- ④ 인덱스 분할(Index Partitioning)

52. 다음 중 분산 데이터베이스에 대한 설명으로 틀린 것은?

- ① 분산 데이터베이스를 구성하기 위해서는 자체적으로 처리 능력이 있는 분산 처리기가 필요하다.
- ② 사이트나 회선의 고장으로부터 회복할 수 있는 기능이 있어야 한다.
- ③ 네트워크를 통해 원격 접근 기능이 있어야 한다.
- ④ DBMS가 수행할 기능이 복잡해지지만 개발 및 처리 비용이 감소한다.

53. 다음 <회원> 테이블에서 "SELECT COUNT(*) FROM 회원 GROUP BY 가입방식 ORDER BY 가입방식 DESC;"를 수행했을 때 나타날 결과로 알맞은 것은?

<회원>

아이디	성별	가입일	가입방식
bakdows	남	2017-04-03	인터넷
nanacle	여	2018-11-05	매장
se7en34	여	2018-05-30	매장
ori_2132	남	2019-09-29	인터넷
dmsgud33	남	2019-12-15	인터넷

- ①

COUNT(*)
5
- ②

COUNT(*)
3
2
- ③

COUNT(*)
2
3
- ④

가입방식	COUNT(*)
매장	2
인터넷	3

54. 사용자 ID가 "guest03"인 사람에게 <EMPLIST> 테이블에 대한 조회 권한을 부여하는 SQL문으로 옳은 것은?

- ① GRANT SELECT ON EMPLIST TO guest03;
- ② GRANT SELECT ON guest03 TO EMPLIST;
- ③ GRANT ALL ON EMPLIST TO guest03;
- ④ GRANT ALL ON guest03 TO EMPLIST;

55. 다음 SQL문에 대한 설명으로 가장 옳지 않은 것은?

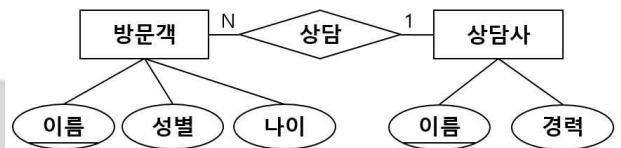
```
CREATE TABLE 상품(
    상품명 VARCHAR(10) PRIMARY KEY,
    단가 INT,
    재고량 INT,
    제조사 VARCHAR(20),
    FOREIGN KEY(제조사) REFERENCES 제조처(이름)
    ON DELETE SET NULL);
```

- ① '상품명' 속성을 기본키로 하는 테이블이다.
- ② '단가'와 '재고량' 속성은 정수를 저장한다.
- ③ '제조사' 속성은 외래키이다.
- ④ <제조처> 테이블의 '이름' 속성이 변경되면 '제조사' 속성의 값이 NULL로 변경된다.

56. 다음 중 트리거(Triiger)에 대한 설명으로 틀린 것은?

- ① 데이터 변경 및 무결성 유지, 로그 메시지 출력 등의 목적으로 사용된다.
- ② 트리거의 생성문에는 반드시 값을 반환하는 RETURN 명령어가 사용되어야 한다.
- ③ 데이터의 삽입, 갱신, 삭제 등의 이벤트가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL이다.
- ④ CREATE TRIGGER 명령어를 통해 생성된다.

57. 다음 E-R 다이어그램에 대한 설명으로 가장 옳지 않은 것은?



- ① '방문객'과 '상담사'는 개체 타입이다.
- ② '이름', '성별', '나이', '경력'은 속성에 해당한다.
- ③ '상담'은 '방문객'과 '상담사' 개체의 관계를 의미한다.
- ④ '방문객'과 '상담사'의 '상담' 관계는 일 대 다의 관계, 즉 한 사람의 방문객이 다수의 상담사를 만날 수 있음을 의미한다.

58. 다음 설명에 해당하는 DBMS 접속 기술은 무엇인가?

Java 언어로 다양한 종류의 데이터베이스에 접속하고 SQL문을 수행할 때 사용되는 표준 API로, 1997년 2월 썬 마이크로시스템에서 개발하였다. Java SE에 포함되어 있으며 접속하려는 DBMS에 대한 드라이버가 필요하다.

- ① JDBC
- ② ODBC
- ③ ORM
- ④ MyBatis

59. 다음 중 쿼리문의 성능을 최적화하기 위한 방법으로 가장 옳지 않은 것은?

- ① WHERE 절을 추가하여 일부 레코드만 조회하게 함으로써 조회에 들어가는 비용을 줄인다.
- ② 서브 쿼리에 특정 데이터가 존재하는지 확인할 때는 EXISTS 보다 IN을 활용한다.
- ③ 옵티마이저의 실행 계획이 잘못되었다고 판단되는 경우 힌트를 활용하여 실행 계획의 액세스 경로 및 조인 순서를 변경한다.
- ④ 실행 계획을 참고하여 인덱스를 추가하거나 기존 인덱스의 열 순서를 변경한다.

60. 다음 중 데이터 전환(Data Migration)에 대한 설명으로 틀린 것은?

- ① 추출(Extraction), 변환(Transformation), 적재(Load) 과정의 앞 글자를 따 ETL이라고도 한다.
- ② 운영 중인 기존 정보 시스템에 축적되어 있는 데이터를 추출하여 새로 개발할 정보 시스템에서 운영 가능하도록 변환한 후, 적재하는 일련의 과정이다.
- ③ 데이터 전환 계획서를 작성하여 대상 및 범위, 환경, 담당 조직 및 역할, 일정 등을 상세하게 기록해야 한다.
- ④ 데이터 전환 계획서에 기재되는 내용 중 비상 계획은 데이터 전환 과정 중에 오류가 발생할 경우에 대비한 데이터의 백업 및 복구 방안을 의미한다.

제4과목 프로그래밍 언어 활용

61. 웹 서버(Web Server)의 기능 중 하나로, 네트워크 트래픽의 포화를 방지하기 위해 응답 속도를 제한하는 기능을 무엇이라 하는가?

- ① HTTP/HTTPS ② 정적 파일 관리
③ 대역폭 제한 ④ 가상 호스팅

62. 다음 중 프레임워크(Framework)에 대한 설명으로 가장 옳지 않은 것은?

- ① 추상화를 통해 설계 및 구현의 변경에 따른 영향을 최소화함으로써 소프트웨어의 품질을 향상시킨다.
② 재사용 가능한 모듈들을 제공함으로써 생산성을 향상시킨다.
③ 인터페이스 확장을 통해 다양한 형태와 기능을 가진 애플리케이션 개발이 가능하다.
④ 객체들의 제어를 프레임워크에 넘길 수 있다.

63. 다음 중 C언어에서 사용하는 기본적인 데이터 형식이 아닌 것은?

- ① char ② boolean
③ int ④ long double

64. 자주 사용되는 변수의 처리속도를 높이기 위해 사용하는 기억 클래스로, 메모리가 아닌 CPU 내부의 저장장소를 기억영역으로 할당받는 변수는 무엇인가?

- ① 자동 변수(Automatic Variable)
② 외부 변수(External Variable)
③ 정적 변수(Static Variable)
④ 레지스터 변수(Register Variable)

65. 다음 C 코드의 결과 값은?

```
#include <stdio.h>
main( ) {
    int a = 3, b = 5;
    int *c = &a;
    *c += ++b;
    printf("%d", a + b + *c);
}
```

- ① 11 ② 19
③ 23 ④ 24

66. 문자 배열 c[5]에 5개의 문자를 scanf() 함수를 이용해 입력받고자 할 때 함수의 사용이 올바른 것은?

- ① scanf("%5c", c); ② scanf("%5d", &c);
③ scanf("%5e", &c); ④ scanf("%5f", c);

67. 다음은 정수 배열 a에 0부터 9까지의 무작위 수를 저장하는 Java 코드이다. 괄호에 들어갈 알맞은 코드는 무엇인가?

```
import ( ).*;
public class Test {
    public static void main(String[] args) {
        Random rand = new Random( );
        int a[] = new int[6];
        for(int i = 0; i < 6; i++)
            a[i] = rand.nextInt(10);
    }
}
```

- ① java.lang ② java.util
③ java.io ④ java.net

68. C, Java 언어 등에서 사용되는 예약어로, 반복문 안에서 이후의 문장을 실행하지 않고 제어를 반복문의 처음으로 옮기는 명령어는 무엇인가?

- ① break ② continue
③ next ④ release

69. 다음 중 운영체제(OS)의 기능에 대한 설명으로 가장 옳지 않은 것은?

- ① 각종 입·출력 장치를 관리 및 제어한다.
② 시스템의 오류를 검사하고 복구한다.
③ 이기종 분산 데이터 시스템의 데이터 동기를 지원한다.
④ 사용자와 시스템 간의 편리한 인터페이스를 제공한다.

70. 다음 설명에 해당하는 것은 무엇인가?

사용자의 명령어를 인식하여 프로그램을 호출하고 명령을 수행하는 UNIX의 명령어 해석기이다. 시스템과 사용자 간의 인터페이스를 담당하며, 파이프라인 기능을 지원하고 입·출력 재지정을 통해 출력과 입력의 방향을 변경할 수 있다.

- ① 파이프(Pipe) ② 유틸리티(Utility)
③ 커널(Kernel) ④ 셸(Shell)

71. 주기억장치에 완전히 비어 있는 3개의 페이지가 있다. 페이지 교체 방법으로 LRU를 사용할 때 요청된 페이지 번호의 순서가 0, 1, 2, 0, 3, 1, 0, 2 인 경우 페이지 부재(Fault)는 몇 번 발생하는가?

- ① 5 ② 6 ③ 7 ④ 8

72. 다음과 같이 저장장치에 작업이 배치되어 있고 9K의 작업이 공간 할당을 요구할 때, 최적 적합(Best Fit) 전략을 사용한다면 어느 주소에 배치되는가?

영역A	8K
영역B	12K(사용중)
영역C	10K
영역D	15K

- ① 영역A ② 영역B
③ 영역C ④ 영역D

73. 다중 프로그래밍의 정도가 높아짐에 따라 주로 나타나는 현상으로, 프로세스 처리 시간보다 페이지 교체에 소요되는 시간이 더 많아지는 것을 무엇이라 하는가?

- ① 구역성(Locality)
② 워킹 셋(Working set)
③ 페이지 부재(Page Fault)
④ 스래싱(Thrashing)

74. 다음 중 프로세스에 대한 설명으로 가장 옳지 않은 것은?

- ① 프로세서가 할당되는 실체로서, 디스패치가 가능한 단위이다.
② 한 프로그램에 속한 작은 프로그램으로, 부프로그램이라고도 한다.
③ 프로세스에 대한 중요한 정보를 저장해 놓는 곳을 PCB라고 한다.
④ 프로세스 내의 작업 단위를 스레드라고 한다.

75. 다음 중 IP(Internet Protocol) 주소에 대한 설명으로 가장 옳지 않은 것은?

- ① A, B, C 클래스만이 네트워크 주소와 호스트 주소 체계의 구조를 가진다.
② D 클래스 주소는 유니캐스팅(Unicasting)을 사용하기 위해 예약되어 있다.

2회 대비

- ③ E 클래스는 실험적 주소로 공용되지 않는다.
 ④ 5개의 클래스(A, B, C, D, E)로 분류되어 있다.

76. OSI 참조 모델에서 시스템들 간에 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 흐름 제어, 프레임 동기화, 오류 제어, 순서 제어 등을 지원하는 계층은 무엇인가?

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

77. 다음 설명에 해당하는 프로토콜은 무엇인가?

- 전송 계층의 프로토콜이다.
- 오버헤드가 적으며, 비연결형 서비스를 제공한다.
- 실시간 전송에 유리하고, 신뢰성보다는 속도가 중요하다.

- ① UDP ② TCP
 ③ RTCP ④ FTP

78. 다음 중 프로세스 스케줄링 기법에 대한 설명으로 가장 옳지 않은 것은?

- ① 비선점 스케줄링은 이미 할당된 CPU를 다른 프로세스가 강제로 빼앗아 사용할 수 없는 스케줄링 기법이다.
 ② 비선점 스케줄링은 프로세스 응답 시간의 예측이 용이하며, 일괄 처리 방식에 적합하다.
 ③ 선점 스케줄링은 주로 빠른 응답 시간을 요구하는 대화식 시분할 시스템에 사용된다.
 ④ 선점 스케줄링은 오버헤드를 최소화할 수 있다.

79. 사용자와 상호 작용 없이 여러 작업들이 정해진 일련의 순서에 따라 자동으로 수행되도록 지원하는 도구를 무엇이라 하는가?

- ① 배치 스케줄러(Batch Scheduler)
 ② 응용 프로그램(Application Program)
 ③ 패키지 소프트웨어(Package Software)
 ④ 프로시저(Procedure)

80. OSI의 전 계층에서 프로토콜 구조가 다른 네트워크 간의 연결을 위해 프로토콜 변환 기능을 수행하는 네트워크 장비는 무엇인가?

- ① 리피터(Repeater)
 ② 브리지(Bridge)
 ③ 라우터(Router)
 ④ 게이트웨이(Gateway)

제5과목 정보시스템 구축 관리

81. 다음 중 객체지향 방법론의 구성 요소에 해당하지 않는 것은?

- ① 객체(Object) ② 패키지(Package)
 ③ 클래스(Class) ④ 메시지(Message)

82. 다음 설명에 해당하는 소프트웨어 개발 방법론은 무엇인가?

- 기존의 시스템이나 소프트웨어를 구성하는 컴포넌트를 조합하여 하나의 새로운 애플리케이션을 만드는 방법론이다.
- 컴포넌트의 재사용(Reusability)이 가능하여 시간과 노력을 절감할 수 있다.

- ① 구조적 방법론
 ② 정보공학 방법론
 ③ Agile 방법론
 ④ CBD 방법론

83. 다음 중 소프트웨어 비용 산정에 대한 설명으로 옳지 않은 것은?

- ① 소프트웨어의 규모, 인력 등의 요소를 기반으로 개발에 필요한 비용을 예측하는 것이다.
 ② 소프트웨어 비용 산정 기법에는 상향식, 하향식, 혼합식 기법이 있다.
 ③ 소프트웨어 비용을 높게 산정할 경우 예산 낭비와 일의 효율성 저하를 초래할 수 있다.
 ④ 소프트웨어 비용 결정 요소에는 프로젝트 요소, 자원 요소, 생산성 요소가 있다.

84. 다음 중 하향식 비용 산정 기법으로만 올바르게 짝지어진 것은?

- ① 전문가 감정 기법, 델파이 기법
 ② 델파이 기법, 수학적 산정 기법
 ③ LOC 기법, 전문가 감정 기법
 ④ 델파이 기법, LOC 기법

85. LOC 기법에 의해 다음과 같은 조건이 제시되었을 때 개발에 필요한 프로그래머의 수는 몇 명인가?

- 총 라인 수 : 30,000
- 1인당 월평균 생산 코드 라인 수 : 600
- 개발 기간 : 10개월

- ① 3명 ② 4명
 ③ 5명 ④ 10명

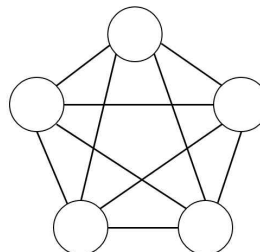
86. 다음 중 Putnam 모형에 대한 설명으로 가장 옳지 않은 것은?

- ① 소프트웨어 생명 주기의 전 과정 동안에 사용될 노력의 분포를 가정해 주는 모형이다.
 ② 대형 프로젝트의 노력 분포 산정에 이용되는 기법이다.
 ③ 자동화 추정 도구 ESTIMACS는 Putnam 모형을 기초로 하여 개발되었다.
 ④ 개발 기간이 늘어날수록 프로젝트 적용 인원의 노력이 감소한다.

87. 각종 컴퓨팅 자원을 중앙 컴퓨터에 두고 인터넷 기능을 갖는 단말기로 언제 어디서나 인터넷을 통해 컴퓨터 작업을 수행할 수 있는 환경으로, 키보드와 모니터, 마우스를 갖추고 통신 포트만 연결하면 업무 수행이 가능해지는 기술을 무엇이라 하는가?

- ① 클라우드 컴퓨팅(Cloud Computing)
 ② 그리드 컴퓨팅(Grid Computing)
 ③ 사물 인터넷(IoT)
 ④ 차세대 통신망(NGN)

88. 다음 그림과 같은 형태의 네트워크 토폴로지에 대한 설명으로 옳지 않은 것은?



- ① 모든 지점의 컴퓨터와 단말장치를 서로 연결한 형태로, 노드의 연결성이 높다.
 ② 많은 단말장치로부터 많은 양의 통신을 필요로 하는 경우에 유리하다.
 ③ 노드의 수가 5개이므로, 총 10개의 회선이 필요하다.
 ④ 보통 분산 처리 시스템에서 사용되며, 통신 회선의 총 경로가 가장 길다.

89. 다음 설명에 해당하는 기술은 무엇인가?

실시간으로 발생하는 많은 사건들 중 의미가 있는 것만을 추출할 수 있도록 사건 발생 조건을 정의하는 데이터 처리 방법으로, 대용량 데이터 스트림에 대한 요구에 실시간으로 대응하기 위해 개발되었다.

- ① 복잡 이벤트 처리(CEP)
- ② 소프트웨어 에스크로(Software Escrow)
- ③ 서비스형 소프트웨어(SaaS)
- ④ 딥 러닝(Deep Learning)

90. 여러 개의 하드디스크로 디스크 배열을 구성하여 디스크의 속도를 향상시키거나 데이터를 복구할 수 있도록 하는 기술은 무엇인가?

- | | |
|--------|----------|
| ① HA | ② MEMS |
| ③ RAID | ④ M-DISC |

91. 다음 괄호에 공통으로 들어갈 용어는 무엇인가?

()는 일련의 데이터를 정의하고 설명해 주는 데이터로, 컴퓨터에서는 데이터 사전의 내용, 스키마 등을 의미한다. ()는 여러 용도로 사용되나 주로 빠르게 검색하거나 내용을 간략하고 체계적으로 하기 위해 많이 사용된다.

- ① View
- ② Broad Data
- ③ Big Data
- ④ Meta Data

92. 다음 중 Secure SDLC에 대한 설명으로 가장 옳지 않은 것은?

- ① 보안상 안전한 소프트웨어를 개발하기 위해 SDLC에 보안 강화를 위한 프로세스를 포함한 것이다.
- ② 소프트웨어의 유지 보수 단계에서 보안 이슈의 해결에 소모되는 많은 비용을 최소화하기 위해 등장하였다.
- ③ 요구사항 분석, 설계, 구현, 테스트, 유지 보수 등 SDLC의 모든 단계에 걸쳐 수행되어야 할 보안 활동을 제시한다.
- ④ 설계 단계에서는 조직의 정보보호 관련 보안 정책을 참고하여 적용할 수 있는 항목들을 문서화한다.

93. 공개키 암호화(Public Key Encryption) 기법에 대한 설명으로 가장 옳지 않은 것은?

- ① 데이터를 암호화할 때 사용하는 공개키(Public Key)로 데이터를 암호화하고 복호화 한다.
- ② 비대칭 암호 기법이라고도 불린다.
- ③ RSA 기법이 대표적인 공개키 암호화 기법이다.
- ④ 암호화/복호화 속도가 느리지만, 키의 분배가 용이하고 관리해야 할 키의 개수가 적다.

94. 다음에서 설명하는 네트워크 공격 방법은 무엇인가?

여러 곳에 분산된 공격 지점에서 한 곳의 서버를 공격하는 것으로, 네트워크에서 취약점이 있는 호스트들을 탐색한 후 이들 호스트들에 공격용 툴을 설치하여 에이전트로 만든 후 공격에 이용한다.

- ① 스미싱 ② DDoS 공격
③ 무작위 대입 공격 ④ 트로이 목마

95. 공중 네트워크와 암호화 기술을 이용하여 인터넷망을 전용선의 사설망을 구축한 것처럼 이용하는 기술로, 이동 근무자가 지역적인 제한 없이 업무를 수행할 수 있게 도와주는 보안 솔루션을 무엇이라 하는가?

- ① ESM(Enterprise Security Management)
- ② IPS(Intrusion Prevention System)
- ③ VPN(Virtual Private Network)
- ④ NAC(Network Access Control)

96. 흐름 제어에서 확인 신호(ACK)를 받은 후에 다음 패킷을 전송하는 방식을 무엇이라 하는가?

- ① 정지 및 대기 ② 슬라이딩 윈도우
③ 다중 전송 ④ 적응적 ARQ

97. 코드 오류로 발생할 수 있는 보안 약점 중 널 포인터(Null Pointer) 역참조가 발생하지 않도록 프로그래머가 수행해야 하는 일로 가장 옳은 것은?

- ① 반환된 메모리에 접근할 수 없도록 주소를 저장하고 있는 포인터를 초기화한다.
- ② 변수 선언 시 할당된 메모리를 초기화한다.
- ③ 포인터를 이용하기 전에 널 값을 갖고 있는지 검사한다.
- ④ 자원 반환 코드가 누락되었는지 확인한다.

98. Public 메소드로부터 Private 배열이 반환될 경우 발생하는 문제점으로 가장 옳은 것은?

- ① 메소드로의 접근이 불가능해진다.
- ② 배열의 주소가 외부로 공개되어 외부에서 접근할 수 있게 된다.
- ③ 시스템의 내부 정보가 노출된다.
- ④ 동기화 오류가 발생한다.

99. 다음 설명에 해당하는 방법을 이용한 인증(Authentication) 유형은 무엇인가?

아이핀(i-PIN)은 인터넷에서 주민등록번호 대신 쓸 수 있도록 만든 사이버 주민등록번호로, 사용자에 대한 신원확인을 완료한 후에 본인확인기관에서 온라인으로 발행한다.

- ① 지식 기반 인증(Something You Know)
- ② 소유 기반 인증(Something You Have)
- ③ 생체 기반 인증(Something You Are)
- ④ 위치 기반 인증(Somewhere You Are)

100. 다음 중 세션ID의 관리 방법에 대한 설명으로 틀린 것은?

- ① 안전한 서버에서 최소 128비트의 길이로 생성한다.
- ② 세션ID가 노출되지 않도록 URL Rewrite 기능을 사용한다.
- ③ 예측이 불가능하도록 안전한 난수 알고리즘을 적용한다.
- ④ 로그인 시 로그인 전의 세션ID를 삭제하고 재발당한다.

정답 및 해설

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

- 1 애자일 모형(Agile Model)은 변화에 기민하고 유연하게 대응하는 것이 특징인 모형으로, 몇 개월이 아닌 몇 주 단위로 실행되는 소프트웨어를 제공합니다.

[병행학습]

애자일 개발 12가지 실행 지침

- 유용한 소프트웨어를 빠르고, 지속적으로 제공하여 고객을 만족시킨다.
- 개발 막바지라도 요구사항 변경을 적극 수용한다.
- 몇 개월이 아닌 몇 주 단위로 실행되는 소프트웨어를 제공한다.
- 고객과 개발자가 프로젝트 기간에 함께 일한다.
- 개발에 대한 참여 의지가 확실한 사람들로 팀을 구성하고, 필요한 개발 환경과 지원을 제공하며, 일을 잘 끝낼 수 있도록 신뢰한다.
- 같은 사무실에서 얼굴을 맞대고 의견을 나눈다.
- 개발의 진척도를 확인하는 1차 기준은 작동하는 소프트웨어이다.
- 지속 가능한 개발을 장려하고 일정한 속도로 개발을 진행한다.
- 기술적 우수성과 좋은 설계에 지속적인 관심을 기울이면 민첩성이 향상된다.
- 단순화를 추구한다.
- 최상의 아키텍처, 명확한 요구사항, 최상의 설계는 자기 스스로 일을 주도하는 조직적인 팀으로부터 나온다.
- 더 효과적인 팀이 될 수 있는 방안을 정기적으로 깊이 고민하고 그에 따라 팀의 행동을 조정한다.

- 2 나선형 모형(Spiral Model)은 보헴(Boehm)이 제안한 것으로, 폭포수 모형과 프로토타입 모형의 장점에 위험 분석 기능을 추가한 모형입니다. 나머지 보기로 제시된 소프트웨어 생명주기 모델의 개념도 확인하고 넘어가세요.

[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 철저하게 검토하여 승인 과정을 거친 후에 다음 단계를 진행하는 개발 방법론
- 프로토타입 모형(Prototype Model, 원형 모형) : 사용자의 요구사항을 정확히 파악하기 위해 실제 개발될 소프트웨어에 대한 건본(시제)품(Prototype)을 만들어 최종 결과물을 예측하는 모형
- 애자일 모형(Agile Model) : 애자일은 ‘민첩한’, ‘기민한’이

라는 의미로, 고객의 요구사항 변화에 유연하게 대응할 수 있도록 일정한 주기를 반복하면서 개발하는 방법론

- 3 ④번의 내용은 스프린트 회고(Retrospective)에 대한 설명입니다. 스프린트 검토 회의(Sprint Review)에서는 부분 또는 전체 완성 제품이 요구사항에 잘 부합되는지 사용자가 포함된 참석자 앞에서 테스트를 수행합니다.

[병행학습]

스크럼 개발 프로세스

- 제품 백로그(Product Backlog) : 제품 개발에 필요한 모든 요구사항(User Story)을 우선순위에 따라 나열한 목록
- 스프린트 계획 회의(Sprint Planning Meeting) : 제품 백로그 중 이번 스프린트에서 수행할 작업을 대상으로 단기 일정을 수립하는 회의
- 스프린트(Sprint) : 실제 개발 작업을 진행하는 과정으로, 보통 2~4주 정도의 기간 내에서 진행
- 일일 스크럼 회의(Daily Scrum Meeting) : 모든 팀원이 매일 약속된 시간에 약 15분 정도의 짧은 시간동안 진행 상황을 점검하는 회의
- 스프린트 검토 회의(Sprint Review) : 부분 또는 전체 완성 제품이 요구사항에 잘 부합되는지 사용자가 포함된 참석자 앞에서 테스트를 수행하는 회의
- 스프린트 회고(Sprint Retrospective) : 스프린트 주기를 되돌아보며 정해놓은 규칙을 잘 준수했는지, 개선할 점은 없는지 등을 확인하고 기록하는 것

- 4 XP(eXtreme Programming)는 수시로 발생하는 고객의 요구사항에 유연하게 대응하기 위해 고객의 참여와 개발 과정의 반복을 극대화하여 개발 생산성을 향상시키는 방법으로, 고객의 요구 변화에 신속하게 대응하기 위해서는 릴리즈 기간을 가능한 짧게 반복해야 합니다.

[병행학습]

XP의 주요 실천 방법(Practice)

- Pair Programming(짝 프로그래밍) : 다른 사람과 함께 프로그래밍을 수행함으로써 개발에 대한 책임을 공동으로 나눠 갖는 환경을 조성함
- Test-Driven Development(테스트 주도 개발)
 - 개발자가 실제 코드를 작성하기 전에 테스트 케이스를 먼저 작성하므로 자신이 무엇을 해야 할지를 정확히 파악함
 - 테스트가 지속적으로 진행될 수 있도록 자동화된 테스트 도구(구조, 프레임워크)를 사용함
- Whole Team(전체 팀) : 개발에 참여하는 모든 구성원(고객

포함)들은 각자 자신의 역할이 있고 그 역할에 대한 책임을 가져야 함

- **Continuous Integration(계속적인 통합)** : 모듈 단위로 나눠서 개발된 코드들은 하나의 작업이 마무리될 때마다 지속적으로 통합함
- **Design Improvement(디자인 개선)** 또는 **Refactoring(리팩토링)** : 프로그램 기능의 변경 없이, 단순화, 유연성 강화 등을 통해 시스템을 재구성함
- **Small Releases(소규모 릴리즈)** : 릴리즈 기간을 짧게 반복함으로써 고객의 요구 변화에 신속히 대응함

- 5 포함(Composition) 관계는 집합 관계의 특수한 형태로, 포함하는 사물의 변화가 포함되는 사물에게 영향을 미치는 관계를 의미합니다. ①번은 집합(Aggregation) 관계에 대한 설명입니다.

[병행학습]

UML의 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 일반적이지 구체적이지를 표현하는 관계
- **의존(Dependency) 관계** : 연관 관계와 같이 사물 사이에 서로 연관은 있으나 필요에 의해 서로에게 영향을 주는 짧은 시간 동안만 연관을 유지하는 관계
- **실체화(Realization) 관계** : 사물이 할 수 있거나 해야 하는 기능(행위, 인터페이스)으로 서로를 그룹화 할 수 있는 관계

- 6 배치 다이어그램은 구조적 다이어그램의 종류 중 하나로, 결과물, 프로세스, 컴포넌트 등 물리적 요소들의 위치를 표현하고 노드와 의사소통(통신) 경로로 표현하는 것이 특징입니다.

[병행학습]

UML의 구조적(Structural) 다이어그램

- **클래스 다이어그램(Class Diagram)**
 - 클래스와 클래스가 가지는 속성, 클래스 사이의 관계를 표현한다.
 - 시스템의 구조를 파악하고 구조상의 문제점을 도출할 수 있다.
- **객체 다이어그램(Object Diagram)** : 클래스에 속한 사물(객체)들, 즉 인스턴스(Instance)를 특정 시점의 객체와 객체 사이의 관계로 표현한다.
- **컴포넌트 다이어그램(Component Diagram)**
 - 실제 구현 모듈인 컴포넌트 간의 관계나 컴포넌트 간의 인터페이스를 표현한다.
 - 구현 단계에서 사용되는 다이어그램이다.
- **배치 다이어그램(Deployment Diagram)**
 - 결과물, 프로세스, 컴포넌트 등 물리적 요소들의 위치를 표현한다.
 - 노드와 의사소통(통신) 경로로 표현한다.
 - 구현 단계에서 사용되는 다이어그램이다.
- **복합체 구조 다이어그램(Composite Structure Diagram)** : 클래스나 컴포넌트가 복합 구조를 갖는 경우 그 내부 구조를 표현함
- **패키지 다이어그램(Package Diagram)** : 유스케이스나 클래스 등의 모델 요소들을 그룹화한 패키지들의 관계를 표현함

- 7 사용자 인터페이스(UI)를 설계할 때 버튼이나 조작 방법 등은 새로운 방식 보다는 일관성 있고 표준화된 형태로 제공하여 사용자가 쉽게 습득하고 사용할 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 한 번 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

- 8 이클립스(Eclipse)는 개발에 필요한 환경을 모두 제공하는 통합 개발 환경(IDE) 도구입니다.

[병행학습]

UI 설계 도구의 종류

- **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

- 9 유스케이스(Use Case)는 사용자의 요구사항을 정리하고 기록하기 위한 도구로, 현재 진행 상태 등을 공유하는 데에는 적합하지 않습니다. 현재 진행 상태 등을 공유하는 데 적합한 도구는 와이어프레임(Wireframe)입니다.

- 10 데이터 추상화는 데이터의 세부적인 속성이나 용도를 정의하지 않고, 데이터 구조를 대표할 수 있는 표현으로 대체하는 방법입니다. ③번의 내용은 과정 추상화에 대한 설명입니다.

[병행학습]

추상화(Abstraction)

- 추상화는 문제의 전체적이고 포괄적인 개념을 설계한 후 차례로 세분화하여 구체화시켜 나가는 것이다.
- 인간이 복잡한 문제를 다룰 때 가장 기본적으로 사용하는 방법으로, 완전한 시스템을 구축하기 전에 그 시스템과 유사한 모델을 만들어서 여러 가지 요인들을 테스트할 수 있다.
- 추상화는 최소의 비용으로 실제 상황에 대처할 수 있고, 시스템의 구조 및 구성을 대략적으로 파악할 수 있게 해준다.

추상화의 유형

- 과정 추상화 : 자세한 수행 과정을 정의하지 않고, 전반적인 흐름만 파악할 수 있게 설계하는 방법
- 데이터 추상화 : 데이터의 세부적인 속성이나 용도를 정의하지 않고, 데이터 구조를 대표할 수 있는 표현으로 대체하는 방법
- 제어 추상화 : 이벤트 발생의 정확한 절차나 방법을 정의하지 않고, 대표할 수 있는 표현으로 대체하는 방법

- 11 UI 프로토타입을 사용할 경우 프로토타입에 사용자의 모든 요구사항을 반영하기 위해 반복적인 개선 및 보완 작업을 수행합니다. 이로 인해 작업 시간이 증가할 수 있고, 필요 이상의 자원이 소모될 수 있습니다.

[병행학습]

UI 프로토타입의 장·단점

장점	<ul style="list-style-type: none"> • 사용자를 설득하고 이해시키기 쉬움 • 요구사항과 기능의 불일치 등으로 인한 혼선을 예방할 수 있어 개발 시간을 줄일 수 있음 • 사전에 오류를 발견할 수 있음
단점	<ul style="list-style-type: none"> • 프로토타입에 사용자의 모든 요구사항을 반영하기 위한 반복적인 개선 및 보완 작업 때문에 작업 시간을 증가시킬 수 있고, 필요 이상으로 자원을 소모할 수 있음 • 부분적으로 프로토타이핑을 진행하다보면 중요한 작업이 생략될 수 있음

- 12 소프트웨어 아키텍처의 설계 과정은 설계 목표 설정, 시스템 타입 결정, 아키텍처 패턴 적용, 서브시스템 구체화, 검토 순입니다.

[병행학습]

소프트웨어 아키텍처의 설계 과정

- ① 설계 목표 설정 : 시스템의 개발 방향을 명확히 하기 위해 설계에 영향을 주는 비즈니스 목표, 우선순위 등의 요구사항을 분석하여 전체 시스템의 설계 목표를 설정함
- ② 시스템 타입 결정 : 시스템과 서브시스템의 타입을 결정하고, 설계 목표와 함께 고려하여 아키텍처 패턴을 선택함
- ③ 아키텍처 패턴 적용 : 아키텍처 패턴을 참조하여 시스템의 표준 아키텍처를 설계함
- ④ 서브시스템 구체화 : 서브시스템의 기능 및 서브시스템 간의 상호작용을 위한 동작과 인터페이스를 정의함
- ⑤ 검토 : 아키텍처가 설계 목표에 부합하는지, 요구사항이 잘 반영되었는지, 설계의 기본 원리를 만족하는지 등을 검토함

- 13 아키텍처 패턴은 시스템의 구조를 이해하는 것이 쉬워 개발에 참여하지 않은 사람도 손쉽게 유지보수를 수행할 수 있습니다.

[병행학습]

아키텍처 패턴

- 아키텍처를 설계할 때 참조할 수 있는 전형적인 해결 방식 또는 예제를 의미한다.
- 소프트웨어 시스템의 구조를 구성하기 위한 기본적인 윤곽을 제시한다.
- 서브시스템들과 그 역할을 정의하고 있으며, 서브시스템 사이의 관계와 여러 규칙·지침 등이 포함되어 있다.
- 아키텍처 스타일 또는 표준 아키텍처라고도 한다.
- 아키텍처 패턴의 장점
 - 시행착오를 줄여 개발 시간을 단축시키고, 고품질의 소프트웨어를 생산할 수 있다.
 - 검증된 구조로 개발하기 때문에 안정적인 개발이 가능하다.

- 이해관계자들이 공통된 아키텍처를 공유할 수 있어 의사소통이 간편해진다.
- 시스템의 구조를 이해하는 것이 쉬워 개발에 참여하지 않은 사람도 손쉽게 유지보수를 수행할 수 있다.
- 시스템의 특성을 개발 전에 예측하는 것이 가능해진다.

- 14 객체가 반응할 수 있는 메시지(Message)의 집합을 행위라고 하며, 객체는 메시지를 통해 행위의 특징을 나타낼 수 있습니다.

[병행학습]

객체(Object)의 특성

- 객체는 독립적으로 식별 가능한 이름을 가지고 있다.
- 객체가 가질 수 있는 조건을 상태(State)라고 하는데, 일반적으로 상태는 시간에 따라 변한다.
- 객체와 객체는 상호 연관성에 의한 관계가 형성된다.
- 객체가 반응할 수 있는 메시지(Message)의 집합을 행위라고 하며, 객체는 메시지를 통해 행위의 특징을 나타낼 수 있다.
- 객체는 일정한 기억장소를 가지고 있다.

- 15 모듈이 파라미터(매개변수)나 인수로 데이터를 주고받는 것을 데이터 결합도(Data Coupling)라고 합니다.

[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 때의 결합도
 - 스탬프(검인) 결합도(Stamp Coupling) : 모듈 간의 인터페이스로 배열이나 레코드 등의 자료 구조가 전달될 때의 결합도
 - 제어 결합도(Control Coupling) : 한 모듈에서 다른 모듈로 논리적인 흐름을 제어하는 데 사용하는 제어 요소(Function code, Switch, Tag, Flag)가 전달될 때의 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

- 16 우연적 응집도(Coincidental Cohesion)는 모듈 내부의 각 구성 요소들이 서로 관련 없는 요소로만 구성된 경우의 응집도입니다. ②번의 내용은 순차적 응집도(Sequential Cohesion)에 대한 설명입니다.

[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
 - 기능적 응집도(Functional Cohesion) : 모듈 내부의 모든 기능 요소들이 단일 문제와 연관되어 수행될 경우의 응집도

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

17 미들웨어(Middleware)는 미들(Middle)과 소프트웨어(Software)의 합성어로, 운영체제와 해당 운영체제에서 실행되는 응용 프로그램 사이에서 운영체제가 제공하는 서비스 이외에 추가적인 서비스를 제공하는 소프트웨어입니다. 나머지 보기로 제시된 용어의 개념도 확인하고 넘어가세요.

[병행학습]

- **ORM(Object-Relational Mapping)** : 객체지향 프로그래밍의 객체(Object)와 관계형(Relational) 데이터베이스의 데이터를 연결(Mapping)하는 기술을 의미함
- **객체지향 프로그래밍(OOP; Object Oriented Programming)** : 현실 세계의 개체(Entity)를 기계의 부품처럼 하나의 객체(Object)로 만들어, 기계적인 부품들을 조립하여 제품을 만들듯이 소프트웨어를 개발할 때에도 객체들을 조립해서 작성할 수 있는 기법을 의미함
- **인터페이스(Interface)** : 서로 다른 두 시스템이나 소프트웨어 등을 서로 이어주는 부분 또는 접속 장치를 의미함

18 대량의 송·수신 데이터를 처리할 때 사용하는 방식은 배치 방식입니다.

[병행학습]

인터페이스 처리 유형

- **실시간 방식** : 사용자가 요청한 내용을 바로 처리해야 할 때 사용하는 방식
- **지연 처리 방식** : 데이터를 매긴 단위로 처리할 경우 비용이 많이 발생할 때 사용하는 방식
- **배치 방식** : 대량의 데이터를 처리할 때 사용하는 방식

19 RPC(Remote Procedure Call)는 응용 프로그램의 프로시저를 사용하여 원격 프로시저를 마치 로컬 프로시저처럼 호출하는 방식의 미들웨어입니다. ①번은 TP-Monitor(Transaction Processing Monitor)에 대한 설명입니다.

[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

20 아키텍처 패턴이 전체 시스템의 구조를 설계하기 위한 참조 모델이라면, 디자인 패턴은 서브시스템에 속하는 컴포넌트들과 그 관계를 설계하기 위한 참조 모델이기 때문에 아키텍처 패턴보다 하위 수준의 설계에 사용됩니다.

[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.

21 연결 리스트(Linked List)는 선형 리스트(Linear List)의 한 종류로, 포인터(Pointer)를 이용하여 서로 연결시킨 형태의 자료 구조입니다.

[병행학습]

자료 구조의 종류

배열(Array)	동일한 자료형의 데이터들이 같은 크기로 나열되어 순서를 갖고 있는 집합
선형 리스트(Linear List)	일정한 순서에 의해 나열된 자료 구조로, 배열을 이용하는 연속 리스트(Contiguous List)와 포인터를 이용하는 연결 리스트(Linked List)로 구분됨
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 ①번은 저장된 데이터(Stored Data), ③번은 운영 데이터(Operational Data), ④번은 공용 데이터(Shared Data)에 대한 설명입니다.

- 23 ①번은 설치 매뉴얼이 아닌 사용자 매뉴얼에 포함되어야 하는 내용입니다.

[병행학습]

소프트웨어 설치 매뉴얼 작성 내용 및 방법

- 소프트웨어 설치 매뉴얼은 사용자가 설치 과정을 이해하기 쉽도록 설치 화면을 누락 없이 캡처하고 순서대로 상세히 설명한다.
- **설치 화면 및 UI** : 설치 실행과 메인 화면 및 안내창에 대한 내용을 기술함
- **설치 이상 메시지 설명** : 설치 방법이나 설치 환경이 잘못된 경우 표시될 수 있는 메시지에 대해 설명함
- **설치 완료 및 결과** : 설치 완료 화면을 수록하여 설치가 정상적으로 마무리되었음을 사용자에게 최종적으로 알림
- **FAQ** : 설치 과정에서 사용자가 직면할 수 있는 문제 상황에 대비할 수 있도록, 설치 시 발생할 수 있는 다양한 상황을 FAQ로 정리하여 수록함
- **설치 시 점검 사항**
 - 설치 전 사용자의 설치 환경에 따라 점검해야 할 사항들이 무엇인지 설명
 - 설치에 필요한 사용자 계정 및 설치 권한에 대해 확인할 수 있도록 설명
 - 설치 과정에서 오류가 발생할 경우 점검할 수 있는 사항들에 대해 설명
- **Network 환경 및 보안**
 - 네트워크 오류로 인해 설치 시 문제가 발생하지 않도록 사전에 필요한 네트워크 연결 상태를 점검하도록 안내함
 - 보안이나 방화벽으로 인해 설치 시 문제가 발생하지 않도록 관련된 내용을 안내함
- **고객 지원 방법(Customer Support)** : 설치와 관련하여 기술적인 지원이나 소프트웨어에 대한 서비스를 원할 경우 국가, 웹 사이트, 전화번호, 이메일 등 문의할 수 있는 연락처를 안내함
- **준수 정보 & 제한 보증(Compliance Information & Limited Warranty)**
 - Serial 보존, 불법 등록 사용 금지 등에 대한 준수 사항을 안내함
 - 저작권자 소유권 정보, SW 허가권 정보, 통신 규격, 개발 언어, 연동 프로그램, 문서 효력, 지적 소유권 정보 등과 관련된 내용을 안내함

- 24 인수 테스트는 개발된 소프트웨어를 사용자가 직접 수행하는 테스트입니다.

[병행학습]

인수 테스트(Acceptance Test)

- 개발한 소프트웨어가 사용자의 요구사항을 충족하는지에 중점을 두고 테스트하는 방법이다.
- 개발한 소프트웨어를 사용자가 직접 테스트한다.
- 테스트에 문제가 없으면 사용자는 소프트웨어를 인수하게 되고, 프로젝트는 종료된다.
- 인수 테스트의 종류
 - 사용자 인수 테스트 : 사용자가 시스템 사용의 적절성 여부를 확인
 - 운영상의 인수 테스트 : 시스템 관리자가 시스템 인수 시 수행하는 테스트 기법으로, 백업/복원 시스템, 재난 복구, 사용자 관리, 정기 점검 등을 확인
 - 계약 인수 테스트 : 계약상의 인수/검수 조건을 준수하는지 여부를 확인
 - 규정 인수 테스트 : 소프트웨어가 정부 지침, 법규, 규정 등 규정에 맞게 개발되었는지 확인

- 알파 테스트 : 개발자의 장소에서 사용자가 개발자 앞에서 행하는 테스트 기법
- 베타 테스트 : 선정된 최종 사용자가 여러 명의 사용자 앞에서 행하는 테스트 기법

- 25 테스트 시나리오는 시스템별, 모듈별, 항목별 등과 같이 여러 개의 시나리오로 분리하여 작성해야 합니다.

[병행학습]

테스트 시나리오 작성 시 유의 사항

- 시스템별, 모듈별, 항목별 등과 같이 여러 개의 시나리오로 분리하여 작성해야 한다.
- 사용자의 요구사항과 설계 문서 등을 토대로 작성해야 한다.
- 각각의 테스트 항목은 식별자 번호, 순서 번호, 테스트 데이터, 테스트 케이스, 예상 결과, 확인 등을 포함해서 작성해야 한다.
- 유스케이스(Use Case) 간 업무 흐름이 정상적인지를 테스트할 수 있도록 작성해야 한다.
- 개발된 모듈 또는 프로그램 간의 연계가 정상적으로 동작하는지 테스트할 수 있도록 작성해야 한다.

- 26 결함 관리의 처리 순서는 '기록 → 검토 → 수정 → 재확인'입니다.

[병행학습]

결함 관리 프로세스

- ① **결함 관리 계획** : 전체 프로세스에 대한 결함 관리 일정, 인력, 업무 프로세스 등을 확보하여 계획을 수립
- ② **결함 기록** : 발견된 결함을 결함 관리 DB에 등록
- ③ **결함 검토** : 등록된 결함을 검토하고 수정할 개발자에게 전달
- ④ **결함 수정** : 개발자가 결함 수정
- ⑤ **결함 재확인** : 개발자가 수정한 내용을 확인하고 다시 테스트를 수행
- ⑥ **결함 상태 추적 및 모니터링 활동** : 결함 관리 상태를 한 눈에 확인할 수 있도록 대시보드나 게시판 형태로 제공
- ⑦ **최종 결함 분석 및 보고서 작성** : 발견된 결함에 대한 정보와 이해관계자들의 의견이 반영된 보고서를 작성하고 결함 관리를 종료

- 27 애플리케이션이 과부하 상태에서 어떻게 작동하는지 확인하는 테스트는 스트레스 테스트(Stress Test)입니다.

[병행학습]

성능 테스트 도구

- 애플리케이션의 성능을 테스트하기 위해 애플리케이션에 부하나 스트레스를 가하면서 애플리케이션의 성능 측정 지표를 점검하는 도구이다.
- **부하(Load) 테스트** : 애플리케이션에 일정 시간 동안 부하를 가하면서 반응을 측정하는 테스트
- **스트레스(Stress) 테스트** : 부하 테스트를 확장한 테스트로, 애플리케이션이 과부하 상태에서 어떻게 작동하는지 테스트
- 종류
 - JMeter : HTTP, FTP 등 다양한 프로토콜을 지원하는 부하 테스트 도구
 - LoadUI : 서버 모니터링, Drag&Drop 등 사용자의 편리성이 강화된 부하 테스트 도구
 - OpenSTA : HTTP, HTTPS 프로토콜에 대한 부하 테스트 및 생산품 모니터링 도구

- 28 연계 테스트는 송·수신용 연계 응용 프로그램의 단위 테스트

를 먼저 수행한 후 연계 테스트 케이스에 따라 데이터 추출, 데이터 송·수신, 데이터 반영 과정 등의 연계 테스트를 수행해야 합니다.

[병행학습]

연계 테스트 프로세스

① 연계 테스트 케이스 작성

- 연계 시스템 간의 데이터 및 프로세스의 흐름을 분석하여 필요한 테스트 항목을 도출하는 과정이다.
- 송·수신용 연계 응용 프로그램의 단위 테스트 케이스와 연계 테스트 케이스를 각각 작성한다.

② 연계 테스트 환경 구축

- 테스트의 일정, 방법, 절차, 소요 시간 등을 송·수신 기관과의 협의를 통해 결정한다.
- 연계 서버 또는 송·수신용 어댑터(Adapter) 설치, 연계를 위한 IP(Internet Protocol) 및 포트(Port) 허용 신청, 연계를 위한 DB 계정 및 테이블과 데이터 생성 등의 테스트 환경을 구축한다.

③ 연계 테스트 수행

- 연계 응용 프로그램을 실행하여 연계 테스트 케이스의 시험 항목 및 처리 절차 등을 실제로 진행한다.
- 송·수신용 연계 응용 프로그램의 단위 테스트를 먼저 수행한 후 연계 테스트 케이스에 따라 데이터 추출, 데이터 송·수신, 데이터 반영 과정 등의 연계 테스트를 수행한다.

④ 연계 테스트 수행 결과 검증

- 연계 테스트 케이스의 시험 항목 및 처리 절차를 수행한 결과가 예상 결과와 동일한지를 확인한다.
- 연계 테스트 수행 결과는 테스트 케이스 항목별 검증 방법을 이용하여 검증한다.

29 애플리케이션의 성능 관리를 위해 접속자, 자원 현황, 트랜잭션 수행 내역, 장애 진단 등 다양한 모니터링 기능을 제공하는 소프트웨어는 APM(Application Performance Management/Monitoring)입니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- ESB(Enterprise Service Bus) : 애플리케이션 간 연계, 데이터 변환, 웹 서비스 지원 등 표준 기반의 인터페이스를 제공하는 솔루션
- EAI(Enterprise Application Integration) : 기업 내 각종 애플리케이션 및 플랫폼 간의 정보 전달, 연계, 통합 등 상호연동이 가능하게 해주는 솔루션
- 옵티마이저(Optimizer) : SQL문을 수행할 가장 효율적이고 빠른 방법을 선택해주는 DBMS의 핵심 엔진

30 SQL은 관계형 데이터베이스(RDB)를 지원하는 언어로, 데이터 정의어(DDL), 데이터 조작어(DML), 데이터 제어어(DCL)로 구분되며, 데이터의 보안, 무결성, 회복, 병행 수행 제어 등을 정의하는 데 사용되는 SQL 언어는 데이터 제어어(DCL)입니다.

[병행학습]

SQL의 종류

- 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

31 클리어링 하우스(Clearing House)는 디지털 저작권 라이선

스의 중개 및 발급을 수행하는 곳으로, 디지털 저작물의 이용 내역을 근거로 저작권료의 정산 및 분배가 수행됩니다. 'Clearing'에는 '결제', '청산'이라는 의미도 있으므로 클리어링 하우스란 '결제가 이루어지는 곳'으로 해석할 수도 있습니다.

[병행학습]

디지털 저작권 관리(DRM) 관련 용어

- 클리어링 하우스(Clearing House) : 저작권에 대한 사용권한, 라이선스 발급, 사용량에 따른 결제 관리 등을 수행하는 곳
- 콘텐츠 제공자(Contents Provider) : 콘텐츠를 제공하는 저작권자
- 패키지(Packager) : 콘텐츠를 메타 데이터와 함께 배포 가능한 형태로 묶어 암호화하는 프로그램
- 콘텐츠 분배자(Contents Distributor) : 암호화된 콘텐츠를 유통하는 곳이나 사람
- 콘텐츠 소비자(Customer) : 콘텐츠를 구매해서 사용하는 주체
- DRM 컨트롤러(DRM Controller) : 배포된 콘텐츠의 이용권한을 통제하는 프로그램
- 보안 컨테이너(Security Container) : 콘텐츠 원본을 안전하게 유통하기 위한 전자적 보안 장치

32 ③번의 내용은 공유 폴더 방식에 대한 설명입니다.

[병행학습]

소프트웨어 버전 관리 도구

- 공유 폴더 방식
 - 버전 관리 자료가 로컬 컴퓨터의 공유 폴더에 저장되어 관리되는 방식이다.
 - 종류 : SCCS, RCS, PVCS, QVCS 등
- 클라이언트/서버 방식
 - 버전 관리 자료가 중앙 시스템(서버)에 저장되어 관리되는 방식이다.
 - 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 방식이다.
 - 종류 : Git, GNU arch, DVC, Bazaar, Mercurial, TeamWare, Bitkeeper, Plastic SCM 등

33 ③번의 내용은 화이트박스(White Box) 테스트에 대한 설명입니다.

[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 등

블랙박스 테스트(Black Box Test)

- 소프트웨어가 수행할 특정 기능을 알기 위해서 각 기능이 완전히 작동되는 것을 입증하는 테스트로, 기능 테스트라고

도 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

- 34 테스트 케이스 설계자가 절차적 설계의 논리적 복잡성을 측정할 수 있게 해주고, 결과가 실행 경로의 기초를 정의하는 지침으로 사용되는 테스트는 기초 경로 검사입니다.

[병행학습]

화이트 박스 테스트의 종류

• 기초 경로 검사

- 대표적인 화이트박스 테스트 기법으로, 테스트 케이스 설계자가 절차적 설계의 논리적 복잡성을 측정할 수 있게 해준다.
- 테스트 측정 결과는 실행 경로의 기초를 정의하는 데 지침으로 사용된다.

• 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

- 35 하향식 통합 테스트를 수행할 때 주요 제어 모듈의 종속 모듈들은 스텝(Stub)로 대체합니다.

[병행학습]

하향식 통합 테스트(Top Down Integration Test)

- 프로그램의 상위 모듈에서 하위 모듈 방향으로 통합하면서 테스트하는 기법이다.
- 주요 제어 모듈을 기준으로 하여 아래 단계로 이동하면서 통합하는데, 이때 깊이 우선 통합법이나 넓이 우선 통합법을 사용한다.
- 테스트 초기부터 사용자에게 시스템 구조를 보여줄 수 있다.
- 상위 모듈에서는 테스트 케이스를 사용하기 어렵다.
- 하향식 통합 테스트 수행 절차

- ① 주요 제어 모듈은 작성된 프로그램을 사용하고, 주요 제

어 모듈의 종속 모듈들은 스텝(Stub)으로 대체한다.

- ② 깊이 우선 또는 넓이 우선 등의 통합 방식에 따라 하위 모듈인 스텝들이 한 번에 하나씩 실제 모듈로 교체한다.
- ③ 모듈이 통합될 때마다 테스트를 실시한다.
- ④ 새로운 오류가 발생하지 않음을 보장하기 위해 회귀 테스트를 실시한다.

- 36 모듈 연계 방법에는 EAI와 ESB 두 가지가 있으며, 애플리케이션 중심의 모듈 연계 방법은 EAI, 서비스 중심의 모듈 연계 방법은 ESB입니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- IDE(Integrated Development Environment) : 개발에 필요한 환경, 즉 편집기(Editor), 컴파일러(Compiler), 디버거(Debugger) 등의 다양한 툴을 하나의 인터페이스로 통합하여 제공하는 것
- 미들웨어(Middleware) : 미들(Middle)과 소프트웨어(Software)의 합성어로, 운영체제와 해당 운영체제에서 실행되는 응용 프로그램 사이에서 운영체제가 제공하는 서비스 이외에 추가적인 서비스를 제공하는 소프트웨어

- 37 소프트웨어 패키징은 사용자를 중심으로 작업이 진행되므로 사용자의 편의성 및 실행 환경을 우선적으로 고려해야 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.

- 38 살충제 패러독스(Pesticide Paradox)는 살충제를 지속적으로 뿌리면 벌레가 내성이 생겨서 죽지 않는 현상을 의미하는 것으로, 애플리케이션 테스트에서는 동일한 테스트를 반복하면 더 이상 결함이 발견되지 않는 현상을 가리킵니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- 결함 집중(Defect Clustering) : 대부분의 결함이 소수의 특정 모듈에 집중해서 발생하는 현상
- 파레토 법칙(Pareto Principle) : 테스트로 발견된 80%의 오류는 20%의 모듈에서 발견된다는 법칙
- 오류-부재의 궤변(Absence of Errors Fallacy) : 결함을 모두 제거해도 사용자의 요구사항을 만족시키지 못하면 해당 소프트웨어는 품질이 높다고 말할 수 없음

- 39 IPSec, SSL, S-HTTP는 네트워크 영역에 보안 기능을 적용

하기 위해 사용하는 암호화 방식입니다. 데이터베이스 영역에서 사용되는 암호화 방식에는 DES, RSA 등이 있습니다.

[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.

- 40 통합 개발 환경(IDE; Integrated Development Environment)은 개발에 필요한 환경, 즉 편집기(Editor), 컴파일러(Compiler), 디버거(Debugger) 등의 다양한 툴을 하나의 인터페이스로 통합하여 제공하는 것을 의미합니다.

[병행학습]

통합 개발 환경(IDE)

- 개발에 필요한 환경, 즉 편집기(Editor), 컴파일러(Compiler), 디버거(Debugger) 등의 다양한 툴을 하나의 인터페이스로 통합하여 제공하는 것을 의미한다.
- 통합 개발 환경 도구는 통합 개발 환경을 제공하는 소프트웨어를 의미한다.
- 통합 개발 환경 도구는 코드의 자동 생성 및 컴파일 가능하고 추가 기능을 위한 도구들을 다운로드하여 추가할 수 있다.
- 통합 개발 환경 도구는 코드를 실행하거나 테스트할 때 오류가 발생한 부분을 시각화하므로 수정이 용이하다.
- 통합 개발 환경 도구는 외부의 다양한 서비스와 연동하여 개발에 편의를 제공하고 필요한 정보를 공유할 수 있다.
- 통합 개발 환경을 지원하는 도구는 플랫폼, 운영체제, 언어별로 다양하게 존재한다.
- **통합 개발 환경 지원 도구** : 이클립스(Eclipse), 비주얼 스튜디오(Visual Studio), 엑스 코드(Scode), IDEA 등

- 41 트랜잭션의 인터페이스와 테이블을 설계하는 단계는 데이터 모델링(논리적 설계) 단계입니다. 데이터 구조화(물리적 설계) 단계에서는 논리적 구조로 표현된 데이터를 디스크 등의 물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 작업을 수행합니다.

[병행학습]

논리적 설계(데이터 모델링)

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

- 트랜잭션의 인터페이스를 설계한다.
- 관계형 데이터베이스라면 테이블을 설계하는 단계이다.

물리적 설계(데이터 구조화)

- 논리적 설계 단계에서 논리적 구조로 표현된 데이터를 디스크 등의 물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 과정이다.
- 물리적 설계 단계에서는 다양한 데이터베이스 응용에 대해 처리 성능을 얻기 위해 데이터베이스 파일의 저장 구조 및 액세스 경로를 결정한다.
- 저장 레코드의 형식, 순서, 접근 경로와 같은 정보를 사용하여 데이터가 컴퓨터에 저장되는 방법을 묘사한다.

- 42 개체 어커런스(Occurrence)는 개체 인스턴스를 부르는 다른 말입니다.

[병행학습]

개체의 구성 요소

- **속성** : 개체가 가지고 있는 특성
- **개체 타입** : 속성으로만 기술된 개체의 정의
- **개체 인스턴스** : 개체를 구성하고 있는 각 속성들이 값을 가져 하나의 개체를 나타내는 것으로 개체 어커런스(Occurrence)라고도 함
- **개체 세트** : 개체 인스턴스의 집합

- 43 하나의 릴레이션에는 동일한 튜플이 존재할 수 없습니다.

[병행학습]

릴레이션의 특징

- 한 릴레이션에는 똑같은 튜플이 포함될 수 없으므로 릴레이션에 포함된 튜플들은 모두 상이하다.
- 한 릴레이션에 포함된 튜플 사이에는 순서가 없다.
- 튜플들의 삽입, 삭제 등의 작업으로 인해 릴레이션은 시간에 따라 변한다.
- 릴레이션 스키마를 구성하는 속성들 간의 순서는 중요하지 않다.
- 속성의 유일한 식별을 위해 속성의 명칭은 유일해야 하지만, 속성을 구성하는 값은 동일한 값이 있을 수 있다.
- 릴레이션을 구성하는 튜플을 유일하게 식별하기 위해 속성들의 부분집합을 키(Key)로 설정한다.
- 속성의 값은 논리적으로 더 이상 쪼갤 수 없는 원자값만을 저장한다.

- 44 외래키(Foreign Key)는 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미합니다.

[병행학습]

키(Key)의 종류

- **후보키(Candidate Key)** : 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용하는 속성들의 부분집합, 즉 기본키로 사용할 수 있는 속성들
- **기본키(Primary Key)** : 후보키 중에서 특별히 선정된 주키(Main Key)로, 한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성
- **대체키(Alternate Key)** : 대체키는 후보키가 둘 이상일 때 기본키를 제외한 나머지 후보키
- **슈퍼키(Super Key)** : 한 릴레이션 내에 있는 속성들의 집합으로 구성된 키로서 릴레이션을 구성하는 모든 튜플들 중 슈퍼키로 구성된 속성의 집합과 동일한 값을 나타내지 않음
- **외래키(Foreign Key)** : 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미하며, 릴레이션 간에 참조 관계를 표현하는데 중요한 도구

- 45 공통 속성을 중심으로 두 개의 릴레이션을 하나로 합쳐서 새로

2회 대비

운 릴레이션을 만드는 연산은 Join으로, 기호는 \bowtie 를 사용합니다.

[병행학습]

Join

- 공통 속성을 중심으로 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 연산이다.
- Join의 결과로 만들어진 릴레이션의 차수는 조인된 두 릴레이션의 차수를 합한 것과 같다.
- Join의 결과는 Cartesian Product(교차곱)를 수행한 다음 Select를 수행한 것과 같다.
- 연산자의 기호는 \bowtie 를 사용한다.
- 표기 형식
 - $R \bowtie_{\text{키속성}r} \text{키속성} S$
 - 키 속성 r은 릴레이션 R의 속성이고, 키 속성 s는 릴레이션 S의 속성이다.

- 46 BCNF는 결정자가 모두 후보키인 정규형으로, 제 3정규형에서 모든 결정자가 후보키가 되도록 분해하면 BCNF가 됩니다.

[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? ≡ 도부이걸다조
도메인이 원자값
부분적 함수 종속 제거
이행적 함수 종속 제거
결정자이면서 후보키가 아닌 것 제거
다치 종속
조인 종속성 이용

- 47 시스템의 성능 향상, 개발 및 운영의 편의성 등의 목적으로 정규화된 데이터 모델을 통합, 중복, 분리하는 과정을 반정규화(Denormalization)라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- **정규화(Normalization)** : 함수적 종속성 등의 종속성 이론을 이용하여 잘못 설계된 관계형 스키마를 더 작은 속성의 세트로 쪼개어 바람직한 스키마로 만들어 가는 과정
- **조인(Join)** : 2개의 테이블에 대해 연관된 튜플들을 결합하여, 하나의 새로운 릴레이션을 반환하는 것
- **분할(Partitioning)** : 반정규화를 위해 테이블을 수직 또는 수평으로 분할하는 것

- 48 인덱스(Index)는 <키 값과 포인터> 쌍으로 구성되는 데이터

구조로, 인덱스 설계 시에는 인덱스가 가리키는 데이터와는 별개로 추가적인 저장 공간이 필요하다는 것을 염두에 두어야 합니다.

[병행학습]

인덱스 설계 시 고려사항

- 새로 추가되는 인덱스는 기존 액세스 경로에 영향을 미칠 수 있다.
- 인덱스를 지나치게 많이 만들면 오버헤드가 발생한다.
- 넓은 범위를 인덱스로 처리하면 많은 오버헤드가 발생한다.
- 인덱스를 만들면 추가적인 저장 공간이 필요하다.
- 인덱스와 테이블 데이터의 저장 공간이 분리되도록 설계한다.

- 49 사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블을 뷰(View)라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- **인덱스(Index)** : 데이터 레코드를 빠르게 접근하기 위해 <키 값, 포인터> 쌍으로 구성되는 데이터 구조
- **클러스터(Cluster)** : 데이터 저장 시 데이터 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법
- **파티션(Partition)** : 대용량의 테이블이나 인덱스를 작은 논리적 단위인 파티션으로 나누는 것

- 50 클러스터(Cluster)는 데이터 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법입니다.

[병행학습]

클러스터(Cluster)

- 데이터 저장 시 데이터 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법이다.
- 클러스터링키로 지정된 컬럼 값의 순서대로 저장되고, 여러 개의 테이블이 하나의 클러스터에 저장된다.
- 클러스터링 된 테이블은 데이터 조회 속도는 향상시키지만 데이터 입력, 수정, 삭제에 대한 성능은 저하시킨다.
- 클러스터는 데이터의 분포도가 넓을수록 유리하다.
- 데이터 분포도가 넓은 테이블을 클러스터링 하면 저장 공간을 절약할 수 있다.
- 클러스터링된 테이블은 클러스터링 열을 공유하므로 저장 공간이 줄어든다.
- 대용량을 처리하는 트랜잭션은 전체 테이블을 스캔하는 일이 자주 발생하므로 클러스터링을 하지 않는 것이 좋다.
- 처리 범위가 넓은 경우에는 단일 테이블 클러스터링을, 조인이 많이 발생하는 경우에는 다중 테이블 클러스터링을 사용한다.
- 파티셔닝된 테이블에는 클러스터링을 할 수 없다.
- 클러스터링을 하면 비슷한 데이터가 동일한 데이터 블록에 저장되기 때문에 디스크 I/O가 줄어든다.
- 클러스터링된 테이블에 클러스터드 인덱스를 생성하면 접근 성능이 향상된다.

- 51 범위 분할로 분할한 다음 해시 함수를 적용하여 다시 분할하는 방식을 조합 분할(Composite Partitioning)이라고 합니다.

[병행학습]

파티션의 종류

- 범위 분할(Range Partitioning)
 - 지정한 열의 값을 기준으로 분할한다.
 - 예) 일별, 월별, 분기별 등

- 해시 분할(Hash Partitioning)
 - 해시 함수를 적용한 결과 값에 따라 데이터를 분할한다.
 - 특정 파티션에 데이터가 집중되는 범위 분할의 단점을 보완한 것으로, 데이터를 고르게 분산할 때 유용하다.
 - 특정 데이터가 어디에 있는지 판단할 수 없다.
 - 고객번호, 주민번호 등과 같이 데이터가 고른 컬럼에 효과적이다.
- 조합 분할(Composite Partitioning)
 - 범위 분할로 분할한 다음 해시 함수를 적용하여 다시 분할하는 방식이다.
 - 범위 분할한 파티션이 너무 커서 관리가 어려울 때 유용하다.

52 분산 데이터베이스를 이용할 경우 DBMS가 수행할 기능이 복잡하고, 소프트웨어 개발 비용 및 처리 비용이 증가한다는 단점이 있습니다.

[병행학습]

분산 데이터베이스의 장·단점

장점	<ul style="list-style-type: none"> • 지역 자치성이 높음 • 자료의 공유성이 향상됨 • 분산 제어가 가능함 • 시스템 성능이 향상됨 • 중앙 컴퓨터의 장애가 전체 시스템에 영향을 끼치지 않음 • 효율성과 융통성이 높음 • 신뢰성 및 가용성이 높음 • 점진적 시스템 용량 확장이 용이함
단점	<ul style="list-style-type: none"> • DBMS가 수행할 기능이 복잡함 • 데이터베이스 설계가 어려움 • 소프트웨어 개발 비용이 증가함 • 처리 비용이 증가함 • 잠재적 오류가 증가함

53 '가입방식' 속성을 기준으로 그룹화하고 내림차순으로 정렬한 후, 그룹별로 튜플의 개수를 구하는 COUNT 함수의 결과를 출력합니다.

- **SELECT COUNT(*)** : 튜플의 개수를 표시한다.
- **FROM 회원** : <회원> 테이블을 대상으로 검색한다.
- **GROUP BY 가입방식** : '가입방식'을 기준으로 그룹을 지정한다.
- **ORDER BY 가입방식 DESC** : '가입방식' 속성의 값을 기준으로 내림차순 정렬한다.

54 GRANT문의 기본 형식은 'GRANT 권한_리스트 ON 개체 TO 사용자 [WITH GRANT OPTION];'이지만, 부여받을 권한을 다른 사용자에게 다시 부여할 수 있는 권한에 대한 언급이 없으므로 '[WITH GRANT OPTION]'을 생략하고 작성하면 됩니다.

- **GRANT SELECT** : 조회(SELECT) 권한을 부여한다.
- **ON EMPLIST** : <EMPLIST> 테이블에 대한 권한을 부여한다.
- **TO guest03** : 'guest03'이라는 사용자에게 부여한다.

55 참조 대상이 되는 튜플이 삭제될 때 취해야 할 사항을 지정하는 옵션이 'ON DELETE SET NULL'이므로, <제조처> 테이블에서 튜플이 삭제되면 관련된 모든 튜플의 '제조사' 속성의 값을 NULL로 변경합니다.

- 1 CREATE TABLE 상품(
- 2 상품명 VARCHAR(10) PRIMARY KEY,
- 3 단가 INT,
- 4 재고량 INT,
- 5 제조사 VARCHAR(20),
- 6 FOREIGN KEY(제조사) REFERENCES 제조처(이름)
- 7 ON DELETE SET NULL);

- 1 <상품> 테이블을 생성한다.
- 2 '상품명' 속성은 최대 문자 10자로, 기본키로 정의한다.
- 3 '단가' 속성은 정수이다.
- 4 '재고량' 속성은 정수이다.
- 5 '제조사' 속성은 최대 문자 20자이다.
- 6 '제조사' 속성은 <제조처> 테이블의 '이름' 속성을 참조하는 외래키이다.
- 7 <제조처> 테이블에서 튜플이 삭제되면 관련된 모든 튜플의 '제조사' 속성의 값을 NULL로 변경한다.

56 RETURN 명령어가 반드시 사용되어야 하는 절차형 SQL은 사용자 정의 함수입니다.

[병행학습]

트리거(Trigger)

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

57 문제의 그림에서 '방문객'과 '상담사'의 '상담' 관계는 다 대일의 관계로, 한 사람의 상담사가 다수의 방문객을 상담할 수 있음을 나타냅니다.

[병행학습]

피터 첸 표기법

사각형(□)	개체(Entity) 타입
마름모(◇)	관계(Relationship) 타입
타원(○)	속성(Attribute)
이중 타원(◎)	다중값 속성(복합 속성)
밑줄 타원(⏟○)	기본키 속성
복수 타원(⊖○)	복합 속성
관계(□ ^N —◇—□ ^M)	1:1, 1:N, N:M 등의 개체 간 관계에 대한 대응수를 선 위에 기술함
선, 링크(——)	개체 타입과 속성을 연결

58 Java 언어로 다양한 종류의 데이터베이스에 접속하고 SQL문을 수행할 때 사용되는 표준 API는 JDBC(Java DataBase Connectivity)입니다. 나머지 보기로 제시된 DBMS 접속 기술들의 개념도 확인하고 넘어가세요.

[병행학습]

- **ODBC(Open DataBase Connectivity)** : ODBC는 데이터베이스에 접근하기 위한 표준 개방형 API로, 개발 언어에

관계없이 사용할 수 있음

- **MyBatis** : JDBC 코드를 단순화하여 사용할 수 있는 SQL Mapping 기반 오픈 소스 접속 프레임워크
- **ORM(Object - Relational Mapping)** : 객체지향 프로그래밍의 객체(Object)와 관계형 데이터베이스(Relational Database)의 데이터를 연결(Mapping)하는 기술

- 59 쿼리문의 성능을 최적화하기 위해 IN보다 EXISTS를 사용하는 것이 좋습니다.

[병행학습]

쿼리 성능 최적화

- 실행 계획에 표시된 연산 순서, 조인 방식, 테이블 조회 방법 등을 참고하여 SQL문이 더 빠르고 효율적으로 작동하도록 SQL 코드와 인덱스를 재구성하는 것을 의미한다.
- **SQL 코드 재구성**
 - WHERE 절을 추가하여 일부 레코드만 조회하게 함으로써 조회에 들어가는 비용을 줄인다.
 - WHERE 절에 연산자가 포함되면 INDEX를 활용하지 못하므로 가능한 한 연산자 사용을 자제한다.
 - 서브 쿼리에 특정 데이터가 존재하는지 확인할 때는 IN보다 EXISTS를 활용한다.
 - 옵티마이저의 실행 계획이 잘못되었다고 판단되는 경우 힌트를 활용하여 실행 계획의 액세스 경로 및 조인 순서를 변경한다.
- **인덱스 재구성**
 - SQL 코드에서 조회되는 속성과 조건들을 고려하여 인덱스를 구성한다.
 - 실행 계획을 참고하여 인덱스를 추가하거나 기존 인덱스의 열 순서를 변경한다.
 - 인덱스의 추가 및 변경은 해당 테이블을 참조하는 다른 SQL문에도 영향을 줄 수 있으므로 신중히 결정한다.
 - 단일 인덱스로 쓰거나 수정 없이 읽기만으로 사용되는 테이블의 경우 IOT(Index-Organized Table)로 구성하는 것을 고려한다.
 - 불필요한 인덱스를 제거한다.

- 60 데이터 전환 계획서의 항목 중 비상 계획은 데이터 전환 당일에 구성하게 될 종합상황실에 대한 계획을 수립하고 데이터 전환 시 종합상황실과 전환 관리 그룹과의 실시간 정보 공유 방법에 대한 계획을 수립하는 것을 의미합니다. ④번은 데이터 복구 대책에 대한 설명입니다.

[병행학습]

데이터 전환

- 운영 중인 기존 정보 시스템에 축적되어 있는 데이터를 추출(Extraction)하여 새로 개발할 정보 시스템에서 운영 가능하도록 변환(Transformation)한 후, 적재>Loading)하는 일련의 과정을 말한다.
- 데이터 전환을 ETL(Extraction, Transformation, Load), 즉 추출, 변환, 적재 과정이라고 한다.
- 데이터 전환을 데이터 이행(Data Migration) 또는 데이터 이관이라고도 한다.
- 데이터 전환을 성공적으로 수행하기 위해 데이터 전환이 필요한 대상을 분석하여 데이터 전환 작업에 필요한 모든 계획을 기록하는 데이터 전환 계획서를 작성하고 이를 토대로 진행한다.
- 데이터 전환 계획서의 주요 항목
 - **데이터 전환 개요** : 데이터 전환 목표, 주요 성공 요인, 전제조건 및 제약조건
 - **데이터 전환 대상 및 범위**
 - **데이터 전환 환경 구성** : 원천 시스템 구성도, 목적 시스템

구성도, 전환 단계별 DISK 사용량

- **데이터 전환 조직 및 역할** : 데이터 전환 조직도, 조직별 역할
- **데이터 전환 일정**
- **데이터 전환 방안** : 데이터 전환 규칙, 데이터 전환 절차, 데이터 전환 방법, 데이터 전환 설계 등
- **데이터 정비 방안** : 데이터 정비 대상 및 방법, 데이터 정비 일정 및 조직
- **비상 계획** : 종합상황실 및 의사소통 체계
- **데이터 복구 대책** : 백업 및 복구 방안

- 61 네트워크 트래픽의 포화를 방지하기 위해 응답 속도를 제한하는 기능은 대역폭 제한(Bandwidth Throttling)입니다.

[병행학습]

웹 서버(Web Server)의 기능

- **HTTP/HTTPS** : 지원 브라우저로부터 요청을 받아 응답할 때 사용되는 프로토콜
- **통신 기록(Communication Log)** : 처리한 요청들을 로그 파일로 기록하는 기능
- **정적 파일 관리(Managing Static Files)** : HTML, CSS, 이미지 등의 정적 파일들을 저장하고 관리하는 기능
- **대역폭 제한(Bandwidth Throttling)** : 네트워크 트래픽의 포화를 방지하기 위해 응답 속도를 제한하는 기능
- **가상 호스팅(Virtual Hosting)** : 하나의 서버로 여러 개의 도메인 이름을 연결하는 기능
- **인증(Authentication)** : 사용자가 합법적인 사용자인지를 확인하는 기능

- 62 프레임워크는 추상화가 아니라 캡슐화를 통해 모듈화를 강화하고 설계 및 구현의 변경에 따른 영향을 최소화함으로써 소프트웨어의 품질을 향상시킵니다.

[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

- 63 boolean을 기본 자료형으로 갖고 있는 프로그래밍 언어는 JAVA입니다. C 언어에서 사용하기 위해서는 라이브러리 파일을 불러와야 합니다.

[병행학습]

C/JAVA의 기본 자료형

- **C 언어**
 - 문자 : char
 - 정수 : short, int, long, long long
 - 실수 : float, double, long double
- **JAVA**
 - 문자 : char
 - 정수 : byte, short, int, long
 - 실수 : float double
 - 논리 : boolean

- 64 메모리가 아닌 CPU 내부의 레지스터에 기억영역을 할당받는 변수는 레지스터 변수입니다.

[병행학습]

기억 클래스의 종류

- **자동 변수(Automatic Variable)** : 함수나 코드의 범위를 한정하는 블록 내에서 선언되는 변수
- **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)** : 메모리가 아닌 CPU 내부의 레지스터에 기억영역을 할당받는 변수

- 65 사용된 코드의 의미는 다음과 같습니다.

```
#include <stdio.h>
main() {
    ❶ int a = 3, b = 5;
    ❷ int *c = &a;
    ❸ *c += ++b;
    ❹ printf("%d", a + b + *c);
}
```

- 정수형 변수 a와 b를 선언하고, 각각 3과 5로 초기화한다.
- 포인터 변수 c를 선언하고, 정수형 변수 a의 주소로 초기화한다.
- c가 가리키는 곳에 증가한 b의 값을 누적한다. c가 가리키는 곳이 a이므로 a의 값이 변경된다.
 $*c += ++b \rightarrow *c = *c + ++b \rightarrow *c = 3 + 6$
- 정수형 변수 a, b와 포인터 변수 c가 가리키는 곳의 값을 합하여 출력한다. $9 + 6 + 9 = 24$

- 66 scanf() 함수에서 5개의 문자를 입력받기 위해서는 문자를 입력받는 서식지정자 '%s' 또는 '%c'를 사용하여 '%5s' 또는 '%5c'로 작성해야 합니다. 또한 배열명 앞에는 '&'를 붙이지 않아야 합니다.

- 67 코드에 사용된 Random 클래스를 포함하고 있는 패키지는 java.util입니다.

```
❶ import java.util.*;
public class Test {
    public static void main(String[] args) {
        ❷ Random rand = new Random();
        ❸ int a[] = new int[6];
        ❹ for(int i = 0; i < 6; i++)
        ❺ a[i] = rand.nextInt(10);
    }
}
```

- Random 클래스가 정의되어 있는 라이브러리이다.
- Random 클래스의 객체 변수 rand를 선언한다.
- 6개의 요소를 갖는 정수형 배열 a를 선언한다.
- 반복 변수 i가 0에서 시작하여 1씩 증가하면서 6보다 작은 동안 ❺번을 반복 수행한다.
- 객체 변수 rand의 메소드 nextInt()를 수행하여 0~9 사이의 정수 난수를 a[i]에 저장한다.

- 68 반복문에서 이후의 문장을 실행하지 않고 제어를 반복문의 처음으로 옮기는 명령어(예약어)는 continue입니다. break는 반복문의 처음이 아닌 반복문을 벗어나는 명령어이며, next와 release는 존재하지 않는 명령어입니다.

- 69 ③번의 내용은 MOM(Message Oriented Middleware)에 대

한 설명입니다.

[병행학습]

운영체제의 기능

- 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.

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

참조 페이지	0	1	2	0	3	1	0	2
페이지 프레임	0	0	0	0	0	0	0	0
		1	1	1	3	3	3	2
			2	2	2	1	1	1
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 발생합니다. LRU는 최근에 가장 오랫동안 사용하지 않은 페이지를 교체하는 기법이므로 참조 페이지 3을 참조할 때에는 1을 제거한 후 3을 가져오게 됩니다. 이러한 과정으로 모든 페이지에 대한 요구를 처리하고 나면 총 페이지 결함 발생 횟수는 6회입니다.

- 72 최적 적합(Best Fit)은 프로그램이나 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 단편화를 가장 작게 남기는 영역에 배치시키는 방법이므로, 사용중이 아닌 영역 중 9K의 작업이 들어갈 수 있는 가장 작은 영역은 10K의 공간이 있는 영역C입니다.

[병행학습]

배치(Placement) 전략

- 배치 전략은 새로 반입되는 프로그램이나 데이터를 주기억 장치의 어디에 위치시킬 것인지를 결정하는 전략이다.
- 최초 적합(First Fit)** : 프로그램이나 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 첫 번째 분할 영역에 배치시키는 방법
- 최적 적합(Best Fit)** : 프로그램이나 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 단편화를 가장 작게 남기는 분할 영역에 배치시키는 방법
- 최악 적합(Worst Fit)** : 프로그램이나 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 단편화를 가장 많이 남기는 분할 영역에 배치시키는 방법

- 73 프로세스의 처리 시간보다 페이지 교체에 소요되는 시간이 더 많아지는 현상을 스래싱(Thrashing)이라고 합니다.

[병행학습]

스래싱(Thrashing)

- 프로세스의 처리 시간보다 페이지 교체에 소요되는 시간이 더 많아지는 현상이다.
- 다중 프로그래밍 시스템이나 가상기억장치를 사용하는 시스템에서 하나의 프로세스 수행 과정에서 자주 페이지 부재가 발생함으로써 나타나는 현상으로, 전체 시스템의 성능이 저하된다.
- 다중 프로그래밍의 정도가 높아짐에 따라 CPU의 이용률은 어느 특정 시점까지는 높아지지만, 다중 프로그래밍의 정도가 더욱 커지면 스래싱이 나타나고, CPU의 이용률은 급격히 감소하게 된다.
- 스래싱 현상 방지 방법
 - 다중 프로그래밍의 정도를 적정 수준으로 유지한다.
 - 페이지 부재 빈도(Page Fault Frequency)를 조절하여 사용한다.
 - 워킹 셋을 유지한다.
 - 부족한 자원을 증설하고, 일부 프로세스를 중단시킨다.
 - CPU 성능에 대한 자료의 지속적 관리 및 분석으로 임계치를 예상하여 운영한다.

- 74 프로세스(Process)는 일반적으로 프로세서(처리기, CPU)에 의해 처리되는 사용자 프로그램, 시스템 프로그램, 즉 실행 중인 프로그램을 의미하며, 작업(Job), 태스크(Task)라고도 합니다. ②번은 프로시저(Procedure)에 대한 설명입니다.

[병행학습]

프로세스의 정의

- PCB를 가진 프로그램
- 실기억장치에 저장된 프로그램
- 프로세서가 할당되는 실행체로서, 디스패치가 가능한 단위
- 프로시저가 활동 중인 것
- 비동기적 행위를 일으키는 주체
- 지정된 결과를 얻기 위한 일련의 계통적 동작

- 목적 또는 결과에 따라 발생하는 사건들의 과정
- 운영체제가 관리하는 실행 단위

- 75 D 클래스 주소는 멀티캐스팅(Multicasting)으로 사용하기 위해 예약되어 있습니다.

[병행학습]

IP 주소(Internet Protocol Address)

- 인터넷에 연결된 모든 컴퓨터 자원을 구분하기 위한 고유한 주소이다.
- 숫자로 8비트씩 4부분, 총 32비트로 구성되어 있다.
- IP 주소는 네트워크 부분의 길이에 따라 다음과 같이 A 클래스에서 E 클래스까지 총 5단계로 구성되어 있다.
- A Class** : 국가나 대형 통신망에 사용(0~127로 시작)
- B Class** : 중대형 통신망에 사용(128~191로 시작)
- C Class** : 소규모 통신망에 사용(192~223으로 시작)
- D Class** : 멀티캐스트용으로 사용(224~239로 시작)
- E Class** : 실험적 주소이며 공용되지 않음

- 76 두 개의 인접한 개방 시스템들 간에 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 하는 계층은 데이터 링크 계층(Data Link Layer)입니다.

[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

- 77 데이터 전송 전에 연결을 설정하지 않는 비연결형 서비스를 제공하며, TCP에 비해 상대적으로 단순한 헤더 구조를 가지고 있어 오버헤드가 적은 전송 계층의 프로토콜은 UDP(User Datagram Protocol)입니다.

[병행학습]

UDP(User Datagram Protocol)

- 데이터 전송 전에 연결을 설정하지 않는 비연결형 서비스를 제공한다.
- TCP에 비해 상대적으로 단순한 헤더 구조를 가지므로, 오버헤드가 적다.
- 고속의 안정성 있는 전송 매체를 사용하여 빠른 속도를 필요로 하는 경우, 동시에 여러 사용자에게 데이터를 전달할 경우, 정기적으로 반복해서 전송할 경우에 사용한다.
- 실시간 전송에 유리하며, 신뢰성보다는 속도가 중요시되는 네트워크에서 사용된다.
- UDP 헤더에는 Source Port Number, Destination Port Number, Length, Checksum 등이 포함된다.

- 78 선점 스케줄링(Preemptive Scheduling)은 선점으로 인해 많은 오버헤드(Overhead)를 초래합니다.

[병행학습]

비선점(Non-Preemptive) 스케줄링

- 이미 할당된 CPU를 다른 프로세스가 강제로 빼앗아 사용할 수 없는 스케줄링 기법이다.
- 프로세스가 CPU를 할당받으면 해당 프로세스가 완료될 때까지 CPU를 사용한다.
- 모든 프로세스에 대한 요구를 공정하게 처리할 수 있다.
- 프로세스 응답 시간의 예측이 용이하며, 일괄 처리 방식에 적합하다.
- 중요한 작업(짧은 작업)이 중요하지 않은 작업(긴 작업)을 기다리는 경우가 발생할 수 있다.
- 비선점 스케줄링의 종류에는 FCFS, SJF, 우선순위, HRN, 기한부 등의 알고리즘이 있다.

선점(Preemptive) 스케줄링

- 하나의 프로세스가 CPU를 할당받아 실행하고 있을 때 우선순위가 높은 다른 프로세스가 CPU를 강제로 빼앗아 사용할 수 있는 스케줄링 기법이다.
- 우선순위가 높은 프로세스를 빠르게 처리할 수 있다.
- 주로 빠른 응답 시간을 요구하는 대화식 시분할 시스템에 사용된다.
- 많은 오버헤드(Overhead)를 초래한다.
- 선점이 가능하도록 일정 시간 배당에 대한 인터럽트용 타이머 클럭(Clock)이 필요하다.
- 선점 스케줄링의 종류에는 Round Robin, SRT, 선점 우선순위, 다단계 큐, 다단계 피드백 큐 등의 알고리즘이 있다.

- 79 일괄 처리(Batch Processing) 작업이 설정된 주기에 맞춰 자동으로 수행되도록 지원해주는 도구를 배치 스케줄러(Batch Scheduler)라고 합니다.

[병행학습]

배치 스케줄러(Batch Scheduler)

- 일괄 처리(Batch Processing) 작업이 설정된 주기에 맞춰 자동으로 수행되도록 지원해주는 도구이다.
- 배치 스케줄러는 특정 업무(Job)를 원하는 시간에 처리할 수 있도록 지원한다는 특성 때문에 잡 스케줄러(Job Scheduler)라고도 불린다.
- 주로 사용되는 배치 스케줄러에는 스프링 배치, Quartz 등이 있다.
- 스프링 배치(Spring Batch)**
 - Spring Source사와 Accenture사가 2007년 공동 개발한 오픈 소스 프레임워크이다.
 - 스프링 프레임워크의 특성을 그대로 가져와 스프링이 가지고 있는 다양한 기능들을 모두 사용할 수 있다.
 - 데이터베이스나 파일의 데이터를 교환하는데 필요한 컴포넌트들을 제공한다.
- Quartz**
 - 스프링 프레임워크로 개발되는 응용 프로그램들의 일괄 처리를 위한 다양한 기능을 제공하는 오픈 소스 라이브러리이다.
 - 수행할 작업과 수행 시간을 관리하는 요소들을 분리하여 일괄 처리 작업에 유연성을 제공한다.

- 80 프로토콜 구조가 다른 네트워크 간의 연결을 위해 프로토콜 변환 기능을 수행하는 장비는 게이트웨이(Gateway)입니다.

[병행학습]

네트워크 관련 장비

- 허브(Hub)** : 한 사무실이나 가까운 거리의 컴퓨터들을 연결하는 장치로, 각 회선을 통합적으로 관리하며, 신호 증폭 기능을 하는 리피터의 역할도 포함함
- 리피터(Repeater)** : 전송되는 신호가 전송 선로의 특성 및 외부 충격 등의 요인으로 인해 원래의 형태와 다르게 왜곡되거나 약해질 경우 원래의 신호 형태로 재생하여 다시 전송하는 역할을 수행함
- 브리지(Bridge)** : LAN과 LAN을 연결하거나 LAN 안에서의 컴퓨터 그룹(세그먼트)을 연결하는 기능을 수행함
- 스위치(Switch)** : 브리지와 같이 LAN과 LAN을 연결하여 훨씬 더 큰 LAN을 만드는 장치
- 라우터(Router)** : 브리지와 같이 LAN과 LAN의 연결 기능에 데이터 전송의 최적 경로를 선택할 수 있는 기능이 추가된 것으로, 서로 다른 LAN이나 LAN과 WAN의 연결도 수행함
- 게이트웨이(Gateway)** : 전 계층(1~7계층)의 프로토콜 구조가 다른 네트워크의 연결을 수행함

- 81 객체지향 방법론의 구성 요소에는 객체(Object), 클래스(Class), 메시지(Message) 등이 있습니다. 패키지(Package)는 JAVA에서 관련 클래스나 인터페이스 등을 하나로 모아둔 것을 가리킵니다.

[병행학습]

객체지향 방법론

- 현실 세계의 개체(Entity)를 기계의 부품처럼 하나의 객체(Object)로 만들어, 소프트웨어를 개발할 때 기계의 부품을 조립하듯이 객체들을 조립해서 필요한 소프트웨어를 구현하는 방법론이다.
- 구조적 기법의 문제점으로 인한 소프트웨어 위기의 해결책

으로 채택되었다.

- 객체지향 방법론의 구성 요소에는 객체(Object), 클래스(Class), 메시지(Message) 등이 있다.
- 객체지향 방법론의 기본 원칙에는 캡슐화(Encapsulation), 정보 은닉(Information Hiding), 추상화(Abstraction), 상속성(Inheritance), 다형성(Polymorphism) 등이 있다.
- **객체지향 방법론의 절차** : 요구 분석 단계 → 설계 단계 → 구현 단계 → 테스트 및 검증 단계 → 인도 단계

82 기존의 시스템이나 소프트웨어를 구성하는 컴포넌트를 조합하여 하나의 새로운 애플리케이션을 만드는 방법론을 컴포넌트 기반(CBD; Component Based Design) 방법론이라고 합니다. 나머지 보기로 제시된 방법론들의 개념도 정리해 두세요.

- **구조적 방법론** : 정형화된 분석 절차에 따라 사용자 요구사항을 파악하여 문서화하는 처리(Precess) 중심의 방법론
- **정보공학 방법론** : 정보 시스템의 개발을 위해 계획, 분석, 설계, 구축에 정형화된 기법들을 상호 연관성 있게 통합 및 적용하는 자료(Data) 중심의 방법론
- **애자일(Agile) 방법론** : 애자일은 ‘민첩한’, ‘기민한’이라는 의미로, 애자일 방법론은 고객의 요구사항 변화에 유연하게 대응할 수 있도록 일정한 주기를 반복하면서 개발 과정을 진행하는 방법론

83 소프트웨어 비용 산정 기법에는 하향식 비용 산정 기법과 상향식 비용 산정 기법이 있습니다. 혼합식 비용 산정 기법은 존재하지 않습니다.

[병행학습]

소프트웨어 비용 산정

- 소프트웨어의 개발 규모를 소요되는 인원, 자원, 기간 등으로 확인하여 실행 가능한 계획을 수립하기 위해 필요한 비용을 산정하는 것이다.
- 소프트웨어 비용 산정을 너무 높게 산정할 경우 예산 낭비와 일의 효율성 저하를 초래할 수 있고, 너무 낮게 산정할 경우 개발자의 부담이 가중되고 품질문제가 발생할 수 있다.
- 소프트웨어 비용 산정 기법에는 하향식 비용 산정 기법과 상향식 비용 산정 기법이 있다.

84 하향식 비용 산정 기법에는 전문가 감정 기법, 델파이 기법 등이 있습니다.

[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 간파할 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 = 30000/600 = 50
 • 투입인원 = 노력/개발 기간 = 50/10개월 = 5명
 ※ 단순히 '30,000라인 / (10개월 * 600라인) = 5'로 계산해도 됩니다.

86 Putnam 모형을 기초로 개발된 자동화 추정 도구는 SLIM입니다. ESTIMACS는 FP 모형을 기초로 개발되었습니다.

[병행학습]

Putnam 모형

- 소프트웨어 생명 주기의 전 과정 동안에 사용될 노력의 분포를 가정해 주는 모형이다.
- 푸트남(Putnam)이 제안한 것으로 생명 주기 예측 모형이라고도 한다.
- 시간에 따른 함수로 표현되는 Rayleigh-Norden 곡선의 노력 분포도를 기초로 한다.
- 대형 프로젝트의 노력 분포 산정에 이용되는 기법이다.
- 개발 기간이 늘어날수록 프로젝트 적용 인원의 노력이 감소한다.

87 각종 컴퓨팅 자원을 중앙 컴퓨터에 두고 인터넷 기능을 갖는 단말기로 언제 어디서나 인터넷을 통해 컴퓨터 작업을 수행할 수 있는 환경을 클라우드 컴퓨팅 환경이라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- **그리드 컴퓨팅(Grid Computing)** : 인터넷에 연결된 수많은 컴퓨터를 격자 구조로 모두 연결하여 하나의 컴퓨터처럼 사용할 수 있게 하는 것
- **사물 인터넷(IoT)** : 정보 통신 기술을 기반으로 실세계(Physical World)와 가상 세계(Virtual World)의 다양한 사물들을 인터넷으로 서로 연결하여 진보된 서비스를 제공하기 위한 서비스 기반 기술
- **차세대 통신망(NGN)** : 유선망뿐만 아니라 이동 사용자를 목표로 하며, 이동통신에서 제공하는 완전한 이동성(Full Mobility) 제공을 목표로 개발되고 있는 차세대 통신망

88 그림은 망형(Mesh)으로, 보통 공중 데이터 통신망에서 사용 됩니다. 분산 처리 시스템에서 사용되는 네트워크 구성 방식은 계층형(Tree)입니다.

[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 실시간으로 발생하는 많은 사건들 중 의미가 있는 것만을 추출할 수 있도록 사건 발생 조건을 정의하는 데이터 처리 방법을

복잡 이벤트 처리(CEP; Complex Event Processing)라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- **소프트웨어 에스크로(Software Escrow)** : 소프트웨어 개발자의 지식재산권을 보호하고 사용자는 저렴한 비용으로 소프트웨어를 안정적으로 사용 및 유지보수 받을 수 있도록 소스 프로그램과 기술 정보 등을 제3의 기관에 보관하는 것
- **서비스형 소프트웨어(SaaS)** : 소프트웨어의 여러 기능 중에서 사용자가 필요로 하는 서비스만 이용할 수 있도록 한 소프트웨어
- **딥 러닝(Deep Learning)** : 인간의 두뇌를 모델로 만들어진 인공 신경망(ANN; Artificial Neural Network)을 기반으로 하는 기계 학습 기술

90 디스크의 속도 향상을 위해 여러 개의 하드디스크로 디스크 배열을 구성하여 파일을 구성하고 있는 데이터 블록들을 서로 다른 디스크들에 분산 저장하는 기술을 RAID라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- **고가용성(HA; High Availability)** : 긴 시간동안 안정적인 서비스 운영을 위해 장애 발생 시 즉시 다른 시스템으로 대체 가능한 환경을 구축하는 메커니즘
- **멤스(MEMS; Micro-Electro Mechanical Systems)** : 초정밀 반도체 제조 기술을 바탕으로 센서, 액추에이터(Actuator) 등 기계 구조를 다양한 기술로 미세 가공하여 전기기계적 동작을 할 수 있도록 한 초미세 장치
- **엠디스크(M-DISC, Millennial DISC)** : 한 번의 기록만으로 자료를 영구 보관할 수 있는 광 저장 장치

91 일련의 데이터를 정의하고 설명해 주는 데이터를 메타 데이터(Meta Data)라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- **뷰(View)** : 사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블
- **브로드 데이터(Broad Data)** : 다양한 채널에서 소비자와 상호 작용을 통해 생성된, 기업 마케팅에 있어 효율적이고 다양한 데이터이며, 이전에 사용하지 않거나 알지 못했던 새로운 데이터나, 기존 데이터에 새로운 가치가 더해진 데이터
- **빅데이터(Big Data)** : 기존의 관리 방법이나 분석 체계로는 처리하기 어려운 막대한 양의 정형 또는 비정형 데이터 집합

92 Secure SDLC에서 조직의 정보보호 관련 보안 정책을 참고하여 소프트웨어 개발에 적용할 수 있는 보안 정책 항목들의 출처, 요구 수준, 세부 내용 등을 문서화하는 단계는 요구사항 분석 단계입니다.

[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 보안대

책, 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

93 공개키 암호화 기법은 암호화 할 때는 공개키(Public Key)를, 복호화 할 때는 비밀키(Secret Key)를 사용합니다.

[병행학습]

공개키 암호화(Public Key Encryption) 기법

- 공개키 암호화 기법은 데이터를 암호화할 때 사용하는 공개키(Public Key)는 데이터베이스 사용자에게 공개하고, 복호화할 때의 비밀키(Secret Key)는 관리자가 비밀리에 관리한다.
- 공개키 암호화 기법은 비대칭 암호 기법이라고도 하며, 대표적으로는 RSA(Rivest Shamir Adleman) 기법이 있다.
- **장점** : 키의 분배가 용이하고, 관리해야 할 키의 개수가 적음
- **단점** : 암호화/복호화 속도가 느리며, 알고리즘이 복잡하고, 개인키 암호화 기법보다 파일의 크기가 큼

94 여러 곳에 분산된 공격 지점에서 한 곳의 서버에 대해 분산 서비스 공격을 수행하는 것을 DDoS(Distributed Denial of Service, 분산 서비스 거부) 공격이라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

• 스미싱(Smishing)

- 문자 메시지(SMS)를 이용해 사용자의 개인 신용 정보를 빼내는 수법이다.
- 초기에는 문자 메시지를 이용해 개인 비밀정보나 소셜 결제를 유도하는 형태로 시작되었다.
- 현재는 각종 행사 안내, 경품 안내 등의 문자 메시지에 링크를 걸어 안드로이드 앱 설치 파일인 apk 파일을 설치하도록 유도하여 사용자 정보를 빼가는 수법으로 발전하고 있다.

• **무작위 대입 공격(Brute Force Attack)** : 암호화된 문서의 암호키를 찾아내기 위해 적용 가능한 모든 값을 대입하여 공격하는 방식

• **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

95 가상 사설 네트워크로서 인터넷 등 통신 사업자의 공중 네트워크와 암호화 기술을 이용하여 사용자가 마치 자신의 전용 회선

을 사용하는 것처럼 해주는 보안 솔루션을 VPN(Virtual Private Network, 가상 사설 통신망)이라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- **ESM(Enterprise Security Management)** : 다양한 장비에서 발생하는 로그 및 보안 이벤트를 통합하여 관리하는 보안 솔루션
- **침입 방지 시스템(IPS; Intrusion Prevention System)** : 방화벽과 침입 탐지 시스템을 결합한 것으로, 비정상적인 트래픽을 능동적으로 차단하고 격리하는 등의 방어 조치를 취하는 보안 솔루션
- **NAC(Network Access Control)** : 네트워크에 접속하는 내부 PC의 MAC 주소를 IP 관리 시스템에 등록한 후 일관된 보안 관리 기능을 제공하는 보안 솔루션

- 96 네트워크 내의 원활한 흐름을 위해 송·수신 측 사이에 전송되는 패킷의 양이나 속도를 규제하는 흐름 제어(Flow Control) 방식에는 수신 측의 확인 신호(ACK)를 받은 후에 다음 패킷을 전송하는 정지-대기(Stop-and-Wait) 방식과 수신 통지를 이용하여 송신 데이터의 양을 조절하는 슬라이딩 윈도우(Sliding Window) 방식이 있습니다.

[병행학습]

흐름 제어(Flow Control)

- 네트워크 내의 원활한 흐름을 위해 송·수신 측 사이에 전송되는 패킷의 양이나 속도를 규제하는 기능이다.
- 송신 측과 수신 측 간의 처리 속도 또는 버퍼 크기의 차이로 인해 생길 수 있는 수신 측 버퍼의 오버플로(Overflow)를 방지하기 위한 기능이다.
- **정지-대기(Stop-and-Wait)**
 - 수신 측의 확인 신호(ACK)를 받은 후에 다음 패킷을 전송하는 방식이다.
 - 한 번에 하나의 패킷만을 전송할 수 있다.
- **슬라이딩 윈도우(Sliding Window)**
 - 확인 신호, 즉 수신 통지를 이용하여 송신 데이터의 양을 조절하는 방식이다.
 - 수신 측의 확인 신호를 받지 않더라도 미리 정해진 패킷의 수만큼 연속적으로 전송하는 방식으로, 한 번에 여러 개의 패킷을 전송할 수 있어 전송 효율이 좋다.
 - 송신 측은 수신 측으로부터 확인 신호(ACK) 없이도 보낼 수 있는 패킷의 최대치를 미리 약속받는데, 이 패킷의 최대치가 윈도우 크기(Window Size)를 의미한다.
 - 윈도우 크기(Window Size)는 상황에 따라 변한다. 즉, 수신 측으로부터 이전에 송신한 패킷에 대한 긍정 수신 응답(ACK)이 전달된 경우 윈도우 크기는 증가하고, 수신 측으로부터 이전에 송신한 패킷에 대한 부정 수신 응답(NAK)이 전달된 경우 윈도우 크기는 감소한다.

- 97 널 포인터(Null Pointer) 역참조를 방지하기 위해서는 널이 될 수 있는 포인터를 이용하기 전에 널 값을 갖고 있는지 검사해야 합니다.

[병행학습]

널 포인터(Null Pointer) 역참조

- 널 포인터가 가리키는 메모리에 어떠한 값을 저장할 때 발생하는 보안 약점이다.
- 많은 라이브러리 함수들이 오류가 발생할 경우 널 값을 반환하는데, 이 반환값을 포인터로 참조하는 경우 발생한다.
- 대부분의 운영체제에서 널 포인터는 메모리의 첫 주소를 가리키며, 해당 주소를 참조할 경우 소프트웨어가 비정상적으로 종료될 수 있다.
- 공격자가 널 포인터 역참조로 발생하는 예외 상황을 악용할 수 있다.

- 널이 될 수 있는 포인터를 이용하기 전에 널 값을 갖고 있는지 검사함으로써 방지할 수 있다.

- 98 Public 메소드로부터 Private 배열이 반환될 경우 배열의 주소가 외부로 공개되어 외부에서 Private 배열에 접근할 수 있게 됩니다.

[병행학습]

Public 메소드로부터 반환된 Private 배열

- 선언된 클래스 내에서만 접근이 가능한 Private 배열을 모든 클래스에서 접근이 가능한 Public 메소드에서 반환할 때 발생하는 보안 약점이다.
- Public 메소드가 Private 배열을 반환하면 배열의 주소가 외부로 공개되어 외부에서 접근할 수 있게 된다.
- Private 배열을 별도의 메소드를 통해 조작하거나, 동일한 형태의 복제본으로 반환받은 후 값을 전달하는 방식으로 방지할 수 있다.

- 99 고정된 패스워드, 패스 프레이즈, 아이핀 등과 같이 사용자가 기억하고 있는 정보를 기반으로 인증을 수행하는 것을 지식 기반 인증(Something You Know)이라고 합니다.

[병행학습]

인증의 주요 유형

• 지식 기반 인증(Something You Know)

- 사용자가 기억하고 있는 정보를 기반으로 인증을 수행하는 것이다.
- 지식 기반 인증은 사용자의 기억을 기반으로 하므로 관리 비용이 저렴하다.
- 사용자가 인증 정보를 기억하지 못하면 본인이라도 인증받지 못한다.
- 고정된 패스워드(Password) : 사용자가 알고 있는 비밀 번호를 접속할 때 마다 반복해서 입력한다.
- 패스 프레이즈(Passphrase) : 'iloveyou'와 같이 일반 패스워드보다 길이가 길고 기억하기 쉬운 문장을 활용하여 비밀번호를 구성하는 방법이다.
- 아이핀(i-PIN) : 인터넷에서 주민등록번호 대신 쓸 수 있도록 만든 사이버 주민등록번호로, 사용자에 대한 신원확인을 완료한 후에 본인확인기관에서 온라인으로 발행함

• 소유 기반 인증(Something You Have)

- 사용자가 소유하고 있는 것을 기반으로 인증을 수행하는 것이다.
- 소유 기반 인증은 소유물이 쉽게 도용될 수 있으므로 지식 기반 인증 방식이나 생체 기반 인증 방식과 함께 사용된다.
- 신분증 : 사용자의 사진이 포함된 주민등록증, 운전면허증, 여권 등을 사용하여 사용자의 신분을 확인함
- 메모리 카드(토큰) : 마그네틱 선에 보안 코드를 저장해서 사용하는 것으로, 카드 리더기를 통해서만 읽을 수 있음(예 : 일반 은행 입출금 카드)
- 스마트 카드 : 마이크로프로세서, 카드 운영체제, 메모리 등으로 구성되어 사용자의 정보뿐만 아니라 특정 업무를 처리할 수 있는 기능이 내장되어 있음(예 : IC칩이 내장된 카드)
- OTP(One Time Password) : 사용자가 패스워드를 요청할 때마다 암호 알고리즘을 통해 새롭게 생성된 패스워드를 사용하는 것으로, 한 번 사용된 패스워드는 폐기됨

• 생체 기반 인증(Something You Are)

- 사용자의 고유한 생체 정보를 기반으로 인증을 수행하는 것이다.
- 생체 기반 인증은 사용이 쉽고 도난의 위험도 적으며 위조가 어렵다.

2회 대비

- 생체 인증 대상 : 지문, 홍채/망막, 얼굴, 음성, 정맥 등
- **위치 기반 인증(Somewhere You Are)**
 - 인증을 시도하는 위치의 적절성을 확인한다.
 - 예) 콜백, GPS나 IP 주소를 이용한 위치 기반 인증

100 세션ID가 노출되지 않도록 하려면 URL Rewrite 기능을 사용하지 않아야 합니다. URL Rewrite 기능은 URL에 사용자의 세션ID를 포함하는 기능으로, 외부에 사용자의 세션ID가 노출될 수 있습니다.

[병행학습]**세션ID의 관리 방법**

- 세션ID는 안전한 서버에서 최소 128비트의 길이로 생성한다.
- 세션ID의 예측이 불가능하도록 안전한 난수 알고리즘을 적용한다.
- 세션ID가 노출되지 않도록 URL Rewrite 기능을 사용하지 않는 방향으로 설계한다.
- 로그인 시 로그인 전의 세션ID를 삭제하고 재할당한다.
- 장기간 접속하고 있는 세션ID는 주기적으로 재할당되도록 설계한다.

