정보처리기사 실기 대비 이론문제

버전: v0.9

인프런/주말코딩
Youtube/@weekendcode

목차

정보처리기사 실기 준비

1. 요구사항 확인

1. 소프트웨어 재공학 과정을 순서대로 쓰시오
- 분석 → 구성 → 역공학 → 이식

2. XP의 5가지 핵심가치를 쓰시오
(피존의 용기는 단순하게 소통한다)
- **피드백, 존중, 용기, 단순성, 소통**
  - 피드백: 지속적인 테스트와 반복적 결함 수정
  - 존중: 모든 프로젝트 관리자는 팀원의 기여 존중
  - 용기: 고객 요구사항 변화에 능동적인 대처
  - 단순성: 부가적 기능, 사용않는 알고리즘 배제
  - 소통: 개발자, 관리자, 고객 간의 원활한 소통

3. 스크럼마스터에 대해 설명하시오
- 업무 배분만 하고 일은 강요하지 않음
- 팀을 스스로 조직하고 관리하도록 지원
- 개발 과정에서 스크럼의 원칙과 가치 지원
- 개발 과정 장애 요소를 찾아 제거

4. 현행 시스템 파악절차 순서
- 1단계: 시스템 구성 파악 → 시스템 기능 파악
  → 시스템 인터페이스 현황 파악
- 2단계: 아키텍처 파악 → 소프트웨어 구성 파악
- 3단계: 시스템 하드웨어 현황 파악
  → 네트워크 구성 파악

5. 애플리케이션 개발 단계에서 요구사항을 분석한 뒤 요구사항을 명세하는 명세 3종류는?
- 시스템 정의서, 시스템 요구사항 명세서, 소프트웨어 요구사항 명세서

6-1. 구조적 다이어그램 종류 작성

클래스	시스템을 구성하는 클래스 사이 관계
패키지	클래스나 유스케이스 등 여러 모델 요소를 그룹화하여 패키지 구성하고, 그 패키지 사이의 관계를 표현
복합체 구조	복합 구조의 클래스와 컴포넌트 내부 구조를 표현
객체	객체 정보를 보여준다. (럼바우 객체지향 분석기법에서 활용됨)
컴포넌트	컴포넌트 구조 사이의 관계를 표현
배치	소프트웨어, 하드웨어, 네트워크를 포함한 실행 시스템의 물리 구조를 표현

* 기출: 컴포넌트 사이의 종속성을 표현하고 결과물, 프로세스, 컴포넌트 등 물리적 요소들의 위치를 표현하는 UML 다이어그램은? **배치 다이어그램**

6-2. 럼바우 객체지향 분석(순서대로)
- 객체 모델링(Object Modeling)
: 정보 모델링이라고도 하며, 시스템에서 요구하는 객체를 찾고 객체들 간의 관계를 정의. 가장 중요하고 선행되어야함. 결과물로 객체 다이어그램이 나온다
- 동적 모델링(Dynamic Modeling)
: 시간의 흐름에 따라 객체들 사이의 제어 흐름, 동작 순서 등의 동적인 행위를 표현, 결과물로 상태 다이어그램이 나온다.
- 기능 모델링(Functional Modeling)
: 프로세스들의 자료 흐름을 중심으로 처리 과정 표현, 결과물로 DFD(자료흐름도)가 나온다.

7. 행위 다이어그램의 종류

유스케이스	사용자 관점에서 시스템 행위 표현
활동	업무 처리 과정이나 연산이 수행되는 과정을 표현
콜라보레이션	순차 다이어그램과 같으며, 모델링 공간에 제약이 없어 구조적인 면 중시
상태머신	객체의 생명주기 표현
순차	시간 흐름에 따른 객체 사이의 상호작용
통신	객체 사이의 관계를 중심으로 상호작용 표현
상호작용 개요	상호작용 다이어그램 사이 제어흐름 표현
타이밍	객체 상태 변화와 시간 제약을 명시적으로 표현

8. UML에서 표현하는 기본 기능 외에 추가적인 기능을 표현하는 것은?
- 스테레오 타입이라고 부르며 ≪≫로 표기한다.
  - 종류: include, extend, interface, exception, constructor

9. 유스케이스 다이어그램 정의와 구성요소**(기출)**
- 정의: 사용자와 다른 외부 시스템들이 향후 개발될 시스템을 이용해 수행할 수 있는 기능을 '사용자의 관점'에서 기술한 것

– 구성요소: 시스템(네모로 표기) / 액터 / 유스케이스(타원형으로 표기) / 관계(화살표)

10. **디자인 패턴**(시험 시작하자마자 찾아서 풀기)

– **생성**패턴: 추상팩토리, 빌더, 팩토리메서드, 프로토타입, 싱글턴 → 객체를 만들 때 생성

– **구조**패턴: 어댑터, 브릿지, 컴포지트, 데코레이터, 퍼사드, 플라이웨이트, 프록시 → 구조를 짤 때

– **행위**패턴: 스테이트, 스트레티지, 템플릿메서드, 메멘토, 옵서버, 커맨드, 인터프리터, 책임연쇄, 비지터(visitor), 이터레이터, 미디에이터 → 객체들이 어떻게 움직일까

11. 비용산정 모형 분류

– 하향식 산정방법

예) 델파이 기법: 전문가의 경험적 지식을 통한 문제 해결 및 미래예측을 위한 기법

– 상향식 산정방법

(1) LoC(코드 라인 수): 원시 코드 라인 수의 낙관치, 중간치, 비관치를 측정하여 예측치를 구해 비용산정

(2) COCOMO: 보헴이 제안한 모형으로 프로그램 규모에 따라 비용산정

(3) Man Month: 한 사람이 1개월 간 할 수 있는 일의 양을 기준으로 비용 산정

12. 소프트웨어 아키텍처 4+1뷰

뷰	설명
유스케이스 뷰	아키텍처를 도출하고 설계하는 작업을 주도하는 뷰
논리 뷰	설계 모델의 추상화, 주요 설계 패키지와 서브 시스템, 클래스 식별 뷰
프로세스 뷰	성능이나 가용성 같은 비기능적 요구사항을 고려하는 뷰
구현 뷰	개발 환경 안에서 정적인 소프트웨어 모듈의 구성을 표현하는 뷰
배포 뷰	물리적인 노드의 구성과 상호연결 관계를 배포 다이어그램으로 표현하는 뷰

13. 플랫폼 성능 특성 분석 항목 3가지

- 응답시간, 가용성, 사용률

14. CMMi 5단계

* CMMi: 카네기 멜런 대학의 소프트웨어 공학 전문 연구소의 지침

- 초기단계→관리단계→정의단계→정량적 관리 단계→ 최적화 단계

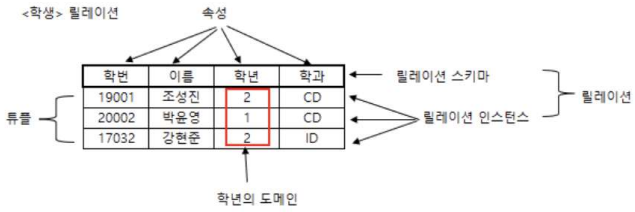
15. 소프트웨어 아키텍처의 평가 및 설계에 사용되는 다양한 방법론과 기술

- SAAM(Software Architecture Analysis Method)
- ATAM(Architecture Trade-off Analysis Method)
- CBAM(Cost Benefit Analysis Method)
- ADR(Active Design Review)
- ARID(Active Review for Intermediate Designs)

2. 데이터 입출력 구현

1. 데이터베이스의 구조와 제약 조건에 관한 전반적인 명세를 기술한 스키마의 종류는?

- 외부 스키마: 사용자나 응용 프로그래머가 각 개인의 입장에서 필요로 하는 데이터베이스의 논리적 구조를 정의한 것
- 개념 스키마: 데이터베이스의 전체적인 논리적 구조, 사용자들이 필요로 하는 데이터를 종합한 조직 전체의 데이터베이스로, 하나만 존재함.
- 내부 스키마: 물리적 저장장치 입장에서 본 데이터베이스 구조, 실제로 저장될 레코드의 형식이나 저장 데이터 항목의 표현방법과 내부 레코드 물리적 순서 등을 나타냄



2. 튜플, 차수, 속성(애트리뷰트)를 설명하라**(기출)**

- 튜플: 릴레이션을 구성하는 각각의 행으로, 튜플의 수를 **카디널리티** 또는 기수라고 한다. 중복이 허용될 수 있으며 셀 수 있는 수량의 순서 있는 열거이다.(=레코드), (튜플의 수=카디널리티)
- 카디널리티: 특정 데이터 집합의 유니크한 값의 개수로, 전체 행에 대한 특정 컬럼의 중복 수치를 나타내는 지표이다.
- 차수: 한 릴레이션에 들어있는 애트리뷰트 수를 차수라고 한다.
- 속성: 하나의 릴레이션은 현실세계의 어떤 개체를 표현하는데, 이 개체에 저장하고 싶은 속성

정보를 뜻한다.(속성의 수=디그리=차수)

- 릴레이션 인스턴스: 어느 한 시점에서 릴레이션의 내용(상태). 저장된 데이터 전체를 의미한다. 릴레이션 또는 릴레이션 외연(Extension)이라고도 한다.

3. 일반 집합 연산자

- 합집합(∪) : 두 릴레이션의 튜플의 합집합
- 교집합(∩) : 두 릴레이션의 튜플의 교집합
- 차집합(-) : 두 릴레이션의 튜플의 차집합
- 교차곱(×) : 두 릴레이션의 튜플들의 순서쌍

3. 관계대수와 관계해석

- Select: 선택조건을 만족하는 튜플의 부분집합을 구해서 새로운 릴레이션을 만든다. 행(가로)에 해당하는 튜플을 구하는 것으로 수평연산
  표기형식: σ (조건)(릴레이션이름)
- Project: 속성 리스트에 제시된 속성 값만을 추출해 새로운 릴레이션을 만드는 연산. 열에 해당하는 속성을 추출하므로 수직연산이며 속성만 추출하므로 '중복을 제거함'
  표기형식: π (속성리스트)(릴레이션이름)
- Join: 공통 속성을 중심으로 두 개 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 연산
  표기형식:(릴레이션속성)▷◁(키속성r-릴레이션R의속성)=(키속성s-릴레이션S의속성)S
- Division: 릴레이션의 행(가로)에 해당하는 튜플을 구하는 것으로 수평연산. Division은 X⊂Y(부분집합)인 두 개의 릴레이션 R(X)와 S(Y)가 있을 때, R의 속성이 S의 속성값을 모두 가진 튜플에서 S가 가진 속성을 제외한 속성만을 구하는 연산
  표기형식: R[속성r ÷ 속성s]S

4. 관계형 데이터베이스의 제약조건에 대해서 설명하라 (무결성)

개체무결성	기본 테이블의 기본키를 구성하는 어떤 속성도 Null값이나 중복값을 가질 수 없다.
참조무결성	외래키 값은 Null이거나 참조 릴레이션의 기본키 값과 동일해야한다. 즉, 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다는 규정
도메인무결성	주어진 속성 값이 정의된 도메인에 속한 값이어야한다는 규정
사용자정의 무결성	속성 값들이 사용자가 정의한 제약조건에 만족되어야 한다는 규정
NULL무결성	릴레이션의 특정 속성 값이 Null이 될 수 없도록 하는 규정
관계무결성	릴레이션에 어느 한 튜플의 삽입 가능 여부 또는 한 릴레이션과 다른 릴레이션의 튜플들 사이에 관계에 적절성 여부를 지정한 규정

* **도메인이란? (기출)** 하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자들의 집합(예, 성별 애트리뷰트의 도메인은 "남"과 "여"이다.)

5. 정규형의 과정 설명(도부이결다조)

1) **도**메인이 원자값 (비정규→1정규형)
2) **부**분적 함수 종속 제거(1정규형→2정규형)
3) **이**행적 함수 종속 제거(2정규형→3정규형)
4) **결**정자이면서 후보키가 아닌 것 제거 (3정규형→BCNF정규형)
5) **다**치 종속 제거(BCNF정규형→4정규형)
6) **조**인 종속성 이용(4정규형→5정규형)

6. 반정규화의 정의와 방법을 설명하시오.

- 정의: 시스템의 성능을 향상하고 개발 및 운영의 편의성을 높이기 위해 정규화된 데이터 모델을 의도적으로 통합, 중복, 분리하여 정규화 원칙을 위배하는 행위
- 반정규화를 수행하면 시스템의 성능이 향상되고 관리 효율성은 증가하지만, 데이터의 일관성 및 정합성이 저하될 수 있다.
- 방법: 테이블 통합, 테이블 분할, 중복 테이블 추가, 중복 속성 추가

7. 시스템 카탈로그에 대해서 설명하시오.

- 시스템 그 자체에 관련이 있는 다양한 객체에 관한 정보를 포함하는 시스템 데이터베이스
- 시스템 카탈로그 내의 각 테이블은 사용자를 포함하여 DBMS에서 지원하는 모든 데이터 객체에 대한 정의나 명세에 관한 정보를 유지

관리하는 시스템 테이블이다.
- 카탈로그들이 생성되면 데이터 사전에 저장되기 때문에 좁은 의미로는 카탈로그를 데이터 사전이라고도 한다.

8. 트랜잭션의 특성을 제시 및 약술하라.(기출 ACID)
- 원자성(Atomicity): 트랜잭션의 연산은 데이터베이스에 모두 반영되거나 전혀 반영되지 않아야 한다. (Commit, Rollback의 성질)
- 일관성(Consistency): 트랜잭션이 그 실행을 성공적으로 완료하면, 언제나 일관성 있는 데이터베이스 상태로 변환된다.
- 독립성, 격리성(Isolation): 둘 이상의 트랜잭션이 동시에 병행되어 실행되는 경우 어느 하나의 트랜잭션 실행 중에는 다른 트랜잭션의 연산이 끼어 들 수 없다.
- 지속성, 영속성(Durability): 성공적으로 완료된 트랜잭션의 결과는 시스템이 고장 나더라도 영구적으로 반영되어야 한다.

9. 접근 통제 기술의 종류를 제시하고 약술하라.
- 임의 접근통제(DAC): 데이터에 접근하는 사용자의 신원에 따라 접근 권한을 부여
- 강제 접근통제(MAC): 주체와 객체 등급을 비교하여 접근 권한을 부여
- 역할기반 접근통제(RBAC): 사용자의 역할에 따라 접근 권한을 부여

10. 파티션의 정의와 종류를 약술하시오.
- 정의: 데이터베이스에서 파티션은 대용량의 테이블이나 인덱스를 작은 논리적 단위인 파티션으로 나누는 것을 말한다.

(1) 범위 분할: 지정한 열의 기준으로 분할함
(2) 해시 분할: 해시 함수를 적용한 결과값에 따라 데이터를 분할함
(3) 조합 분할: 범위 분할로 분할한 다음 해시 함수를 적용하여 다시 분할하는 방식

11. 분산 데이터베이스의 목표는 무엇인가?
- 위치 투명성: 액세스하려는 데이터베이스의 실제 위치를 알 필요 없이 단지 데이터베이스의 논리적인 명칭만으로 액세스할 수 있다.
- 중복 투명성: 동일 데이터가 여러 곳에 중복되어 있더라도 사용자는 마치 하나의 데이터만 존재하는 것처럼 사용하고, 시스템은 자동으로 여러 자료에 대한 작업을 수행한다.
- 병행 투명성: 분산 데이터베이스와 관련된 다수의 트랜잭션들이 동시에 실현되더라도 그 트랜잭션의 결과는 영향을 받지 않는다.
- 장애 투명성: 트랜잭션, DBMS, 네트워크, 컴퓨터 장애에도 불구하고 트랜잭션을 정확하게 처리한다.

12. 데이터베이스에서 클러스터의 개념을 서술하라
- 데이터 저장시 데이터 액세스의 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적인 방법

13. 데이터 모델 구성 3요소
- 구조: 논리적으로 표현된 개체 타입들 간의 관계로서 데이터 구조 및 정적 성질을 표현
- 연산: 데이터베이스에 저장된 실제 데이터를 처리하는 작업에 대한 명세로 데이터베이스를 조작하는 도구
- 제약조건: 데이터베이스에 저장될 수 있는 실제 데이터의 논리적인 제약 조건

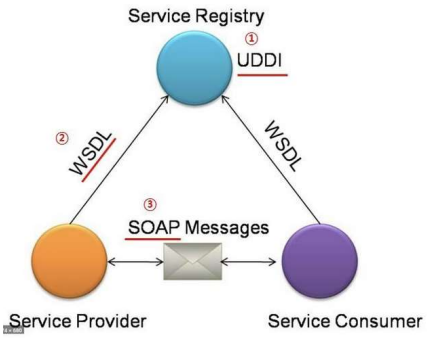
14. 함수적 종속성(Functional Dependency) 1-109
* 속성 Y가 속성 X에 함수적으로 종속된다는 말은 속성 X를 이용하여 속성 Y를 식별할 수 있다는 말이다.
- 완전함수종속
- 부분함수종속

15. SQL DDL, DML, DCL 종류
- DDL: CREATE, ALTER, DROP, RENAME, TRUNCATE
- DML: SELECT, INSERT, UPDATE, DELETE
- DCL: GRANT, REVOKE
- TCL: COMMIT, ROLLBACK, SAVEPOINT

학생 테이블에 대한 권한 부여:
GRANT SELECT, INSERT ON student to kim, lee;
학생 테이블 권한 회수:
REVOKE SELECT, INSERT ON student from lee;

3. 통합 구현

1. 웹 서비스는 네트워크에 분산된 정보를 서비스 형태로 개방하여 표준화된 방식으로 공유하는 기술로 서비스 지향 아키텍처(SOA) 개념을 실현하는 대표적 기술이다. ①, ②, ③ 용어를 약술하시오.



1. 웹 서비스는 네트워크에 분산된 정보를 서비스 형태로 개방하여 표준화된 방식으로 공유하는 기술로 서비스 지향 아키텍처(SOA) 개념을 실현하는 대표적 기술이다. ①, ②, ③ 용어를 약술하시오. **(기출)**

① **UDDI** : 웹 서비스에 대한 정보인 WSDL을 등록하고 검색하기 위한 저장소로, 공개적으로 접근/검색이 가능한 레지스트리

② **WSDL** : 웹 서비스명, 제공 위치, 메시지 포맷, 프로토콜 정보 등 웹 서비스에 대한 상세 정보가 기술된 XML로 구현되어있는 언어

③ **SOAP** : HTTP, HTTPS, SMTP 등을 이용하여 XML 기반의 메시지를 네트워크 상태에서 교환하는 프로토콜

2. 연계(인터페이스) 구성 3단계를 쓰시오
- 송신시스템, 중계 시스템, 수신 시스템

3. 연계 테스트 진행 순서
- 연계 케이스 작성 → 연계 테스트 환경 구축 → 연계 테스트 수행 → 연계 테스트 수행 결과 검증

4. 연계 방식의 종류 중 **직접 연계 종류**를 제시하라.
- DB Link: 한 데이터베이스에서 네트워크상의 다른 데이터베이스에 접속하기 위한 설정을 해주는 오라클 객체이다.
- DB Connection Pool: 수신 시스템 WAS에서 송신 시스템 DB로 연결되는 Connection Pool을 생성하고 프로그램 소스에서 WAS에 설정된 Connection Pool명을 참고하여 구현한다.
- JDBC: 수신 시스템의 Batch, Online 프로그램에서 JDBC 드라이버를 이용하여 송신 시스템의 DB와 연결을 생성한다.
- 화면 링크: 웹 애플리케이션 화면에서 하이퍼링크를 사용한다. (예, <a href="url">)
- API/Open API: 송신시스템의 DB와 연결하여 데이터를 제공하는 인터페이스 프로그램

5. 연계 방식의 종류 중 **간접연계 종류**를 제시하라. **(기출)**
- EAI: 실제 송수신 처리와 진행 현황을 모니터링 및 통제하는 EAI 서버, 송수신 시스템에 설치되는 어댑터(Client)를 말한다. 기업 내 각종 애플리케이션 및 플랫폼 간의 정보 전달, 연계, 통합 등 상호 연동이 가능하게 해주는 솔루션

– 구축유형

: Point to Point, Hub & Spoke, Message Bus, Hybrid (구축유형 기출)

(1) Point-to-Point: 중간에 미들웨어를 두지 않고 각 애플리케이션 간 Point to Point 형태 연결

(2) Hub & Spoke: 단일 접점이 허브 시스템을 통해 데이터를 전송하는 중앙 집중적 방식

(3) Message Bus: 애플리케이션 사이 미들웨어를 두어서 처리하는 방식

(4) Hybrid: (2), (3) 혼합방식으로 데이터 병목현상을 최소화하는 방식

- Web Service/ESB: 웹 서비스가 설명된 WSDL과 SOAP 프로토콜을 이용한 시스템을 연계한다. 미들웨어인 ESB에서 서비스 간 연동을 위한 변환 처리로 다중 플랫폼을 지원한다.
- Socket: 소켓을 생성하여 포트를 할당하고, 클라이언트의 요청을 연결하여 통신하는 네트워크 프로그램의 기반 기술이다.

6. 객체지향 설계원칙 (설계 5대 원칙)

- **단일 책임 원칙(Single Responsibility Principle)**

: 모든 클래스는 각각 하나의 책임만 가져야한다. 클래스는 그 책임을 완전히 캡슐화해야한다.

ex) 계산 클래스는 오직 사칙연산 기능만 책임진다.

- **개방-폐쇄 원칙(Open Closed Principle)**

: 확장에는 열려있고, 수정에는 닫혀있어야 한다. 기존의 코드를 변경하지 않으면서 기능을 추가할 수 있도록 설계가 되어야한다.

ex) 캐릭터를 설계할 때 각각의 캐릭터가 움직임이 다른 경우 하위 클래스에게 움직임을 구현 위임

- **리스코프 치환 원칙(Liskov Subsititution Principle)**

: 자식 클래스는 언제나 부모의 클래스를 대체할 수 있다.

- **인터페이스 분리 원칙(Interface Segregation Principle)**

: 한 클래스는 자신이 사용하지 않는 인터페이스는 구현하지 말아야 한다. 일반적인 인터페이스보다 여러 개의 구체적인 인터페이스가 낫다.

- **의존 역전 원칙(Dependency Inversion Principle)**

: 의존 관계를 맺을 때 구체적인 클래스보다

인터페이스나 추상 클래스와 관계를 맺을 것.

4. 서버 프로그램 구현

(본 단원에 디자인 패턴은 별도 첨부로 암기할 것)

1. 개발 언어의 선정 기준 5가지
- 적정성, 효율성, 이식성, 친밀성, 범용성

2. 소프트웨어 아키텍처의 설계 과정
- 설계 목표 설정 → 시스템 타입 결정 → 아키텍처 패턴 적용 → 서브 시스템 구체화 → 검토

3. 소프트웨어 아키텍처의 설계 과정
(1) 설계 목표 설정
(2) 시스템 타입 결정
(3) 아키텍처 패턴 적용
(4) 서브 시스템 구체화
(5) 검토

4. 아키텍처 패턴
- 레이어 패턴: 시스템을 계층으로 구분하여 구성하는 고전적인 방법의 패턴
- 클라이언트-서버: 하나의 서버 컴포넌트와 다수의 클라이언트 컴포넌트로 구성되는 패턴
- 파이프-필터 패턴: 데이터 스트림 절차의 각 단계를 필터로 캡슐화하여 파이프를 통해 전송하는 패턴
- 모델-뷰-컨트롤러 패턴: 서브 시스템을 모델, 뷰, 컨트롤러로 구조화하는 패턴
- 기타: 마스터-슬레이브 패턴, 브로커 패턴, 피어-투-피어 패턴, 이벤트-버스 패턴, 블랙보드 패턴

5. 객체 지향 분석의 방법론
- 럼바우 방법: 모든 소프트웨어 구성 요소를 그래픽 표기법을 이용하여 모델링하는 기법 (**객**체모델링, **동**적모델링, **기**능모델링 순으로 이뤄진다.)
- 부치 방법: 미시적 개발프로세스와 거시적 개발 프로세스를 모두 사용함
- 자콥스 방법: 유스케이스를 강조하여 사용
- Coad와 Yourdon 방법: E-R 다이어그램을 사용하여 객체의 행위를 모델링함
- Wirfs-Brock 방법: 분석과 설계 간의 구분이 없으며 고객 명세서를 평가해서 설계 작업까지 연속적으로 수행한다.

6. 객체지향 설계원칙 5가지
- 단일 책임 원칙: 객체는 단 하나의 책임만 가져야한다.
- 개방-폐쇄 원칙: 기존의 코드를 변경하지 않고, 기능을 추가할 수 있도록 설계해야한다.
- 리스코프 치환 원칙: 자식 클래스는 최소한 부모 클래스의 기능은 수행할 수 있어야한다.
- 인터페이스 분리 법칙: 자신이 사용하지 않는 인터페이스와 의존 관계를 맺거나 영향을 받지 않아야 한다.
- 의존 역전 원칙: 의존 관계 성립시, 추상성이 높은 클래스와 의존 관계를 맺어야한다.

7. 소프트웨어 아키텍처 설계 기본 원리
- 모듈화: 소프트웨어 성능 향상, 시스템의 수정 및 재사용, 유지 관리 등이 용이하도록 시스템의 기능들을 모듈 단위로 나누는 것
- 추상화: 문제의 전체적이고 포괄적인 개념을 설계한 후, 차례로 세분화하여 구체화시켜 나가는 것
- 단계적 분해: 문제를 상위의 중요 개념으로부터 하위의 개념으로 구체화시키는 분할 기법
- 정보 은닉: 한 모듈 내부에 포함된 절차와 자료들의 정보가 감추어져 다른 모듈이 접근하거나 변경하지 못하도록 하는 기법

8. 모듈의 독립성은 결합도와 응집도에 의해 측정된다. 결합도와 응집도를 설명하고, 모듈의 독립성을 높이기 위한 조건을 제시하라.
- 모듈의 독립성을 높이기 위해서는 결합도는 약하게, 응집도는 높게 해야한다.

9. **모듈 간 통신 방식을 구현하기 위해 대표적인 프로그래밍 인터페이스 집합인 IPC**의 대표 메소드는?
* Inter-Process Communication: 프로세스 간 통신
- Shared memory: 공유 가능한 메모리를 구성하여 다수의 프로세스가 통신하는 방식
- Socket: 네트워크 소켓을 이용하여 네트워크를 경유하는 프로세스 간에 통신하는 방식
- Semaphores: 공유 자원에 대한 접근 제어를 통해 통신하는 방식
- Pipes & Named Pipes: “Pipe”라고 불리는 선입선출 형태로 구성된 메모리를 여러 프로세스가 공유하여 통신하는 방식
- Message Queueing: 메시지가 발생하면 이를 전달하는 방식으로 통신하는 방식

10. 서버 개발 과정은 DTO/VO, SQL, DAO, Service,

Controller를 각각 구현하는 과정이다. 각각에 대해서 설명하라.

- DTO/VO 구현: 데이터 교환을 위해 사용할 객체 만드는 과정
- SQL 구현: 데이터 삽입/변경/삭제 등의 작업을 수행할 SQL문을 생성하는 과정
- DAO 구현: 데이터베이스에 접근하고, SQL을 활용하여 데이터를 실제로 조작하는 코드를 구현하는 과정
- Service 구현: 사용자의 요청에 응답하기 위한 로직을 구현하는 과정
- Controller 구현: 사용자의 요청에 적절한 서비스를 호출하여, 그 결과를 사용자에게 반환하는 코드를 구현하는 과정

11. 소프트웨어 재사용을 위한 공통 모듈화
- 분할과 정복(Divide & Conquer): 복잡한 문제를 분해해서 모듈 단위로 해결
- 자료추상화: 함수 내의 자료 구조, 표현 내역을 숨기고 인터페이스를 통해 접근
- 모듈 독립성: 각 모듈의 결합도를 줄이고 응집도를 높인다.
- 정보 은닉: 중요한 정보나 노출 가능성이 큰 모듈을 은닉한다.
- 비용과 모듈: 모듈 수가 많아지면, 비용도 증가한다.

12. MVC 패턴에 대해서 설명하라
- Model-View-Controller의 줄임말로, 애플리케이션을 세 가지 주요 구성 요소인 모델(Model), 뷰(View), 컨트롤러(Controller)로 분리하여 개발의 효율성과 유지보수성을 높인다.
- Model: 데이터와 비즈니스 로직을 담당. 데이터를 관리하고 비즈니스 핵심 로직을 처리하고, 데이터를 가공하며 애플리케이션의 상태를 유지
- View: 사용자 인터페이스 담당. 데이터 표시나 버튼, 텍스트 필드 등을 포함
- Controller: 사용자의 입력을 처리하고 모델을 업데이트하며 뷰를 업데이트함. 모델과 뷰 사이에서 연결하는 역할

13. 일정 스케줄마다 돌아가면서 일괄처리하는 프로그램은?
- 배치 프로그램

- **결합도(Coupling)**: 모듈 간에 상호 의존하는 정도 또는 두 모듈 사이의 연관관계 (결합도가 강한 순서, **내공외제스자**)

내용 결합도(Content) → 공통 결합도(Common) → 외부 결합도(External) → 제어 결합도(Control) → 스탬프 결합도(Stamp) → 자료 결합도(Data)

1. 내용: 한 **모듈이 다른 모듈의 내부 기능 및 내부 자료를 직접 참조**하거나 수정
2. 공통: 공유되는 **공통 데이터 영역을 여러 모듈이 사용**할 때의 결합도(전역변수 쓸 때)
3. 외부: 어떤 모듈에서 **선언한 데이터(변수)를 외부의 다른 모듈에서 참조**할 때 결합도
4. 제어: 어떤 모듈이 다른 모듈 내부의 논리적인 흐름을 제어하기 위해 **제어요소(flag)를 전달**하는 결합도 (제어요소를 전달 (if 등))
5. 스탬프: 모듈 간의 인터페이스로 배열이나 **오브젝트, 포인터** 등 동일한 자료 구조가 전달될 때의 결합도
6. 자료: 모듈 간의 **인터페이스가 자료 요소**로만 구성되는 결합도 func a(x)를 a(10) 호출

- **응집도(Cohesion)**: 모듈의 내부 요소들이 서로 연관되어 있는 정도 (응집도가 강한 순서, **기순교절시논우**):

기능적 응집도(Functional) → 순차적 응집도(Sequential) → 교환적 응집도(Communication) → 절차적 응집도(Procedural) → 시간적 응집도(Temporal)→ 논리적 응집도(Logical) → 우연적 응집도(Coincidental)

1. 기능적: 모듈 내부 기능 요소들이 단일 문제와 연관되어 수행
2. 순차적: 모듈 내 하나 활동으로 나온 **출력 데이터를 그 다음 활동 입력 데이터**로 사용
3. 교환적: **동일한 입력과 출력을 사용해 서로 다른 기능을 수행**하는 구성 요소가 모임
4. 절차적: 모듈 안의 구성요소들이 **기능을 순차적으로 수행**할 경우 응집도
5. 시간적: **특정 시간에 처리**되는 몇 개의 기능을 모아 하나의 모듈로 작성할 경우의 응집도
6. 논리적: **유사한 성격을 갖거나 특정 형태로 분류되는 처리 요소**들로 하나의 모듈을 형성
7. 우연적: 서로 **관련 없는 요소**들로 하나의 모듈 구성

* 개별 결합도와 응집도에 대한 세부 설명 기출

5. 인터페이스 구현

1. AJAX의 특징
- 비동기 통신, 부분 업데이트, 다양한 데이터 형식 지원, 빠른 응답시간, 서버와 클라이언트 분리

2. 정보보안의 3요소 **(기출)**
- 기밀성: 인가된 사용자만 정보 자산에 접근할 수 있다.
- 무결성: 시스템 내 정보는 오직 인가된 사용자가 인가된 방법으로만 수정할 수 있다.
- 가용성: 사용자가 필요할 때 데이터에 접근할 수 있는 능력

3. 시큐어 코딩 가이드: 입력 데이터 검증 및 표현, 보안 기능, 시간 및 상태, 에러 처리, 코드 오류, 캡슐화, API 오용

4. 인터페이스 보안 기능을 적용해야 하는 3영역
- 데이터베이스 영역
- 애플리케이션 영역
- 네트워크 영역

6. 화면 설계

1. 기능적 요구사항/비기능적 요구사항
- 기능적: 시스템 입출력으로 포함되어야 할 사항, 동기화 등의 기타 요구사항, 제품을 구현하기 위해 소프트웨어가 가져야할 기능적 속성
- 비기능적: 품질에 관한 요구사항(사용성, 신뢰성, 이식성, 표준, 윤리적 등), 제품 품질 기준 등의 만족을 위해 소프트웨어가 가져야할 특성

2. UI의 기본 원칙을 제시하고 약술하시오**(기출)**
- **직관성**: 누구나 쉽게 이해하고 사용할 수 있어야 함
- **유효성**: 사용자의 목적을 정확하고 완벽하게 달성해야 함
- **학습성**: 누구나 쉽게 배우고 익힐 수 있어야 함
- **유연성**: 사용자의 요구사항을 최대한 수용하고 실수를 최소화해야 함

3. 메뉴를 기반으로 작업할 수 있는 환경은?
- MUI(Menu User Interface)

7. 애플리케이션 테스트 관리

1. 소스코드 품질 분석 도구 종류인 정적분석과 동적분석 기법을 간략히 정의하시오.
- 정적분석: 프로그램을 실행하지 않고 명세서나 소스코드를 대상으로 분석하는 도구
  - 종류: 워크스루, 인스펙션, 코드 검사

(1) 코드 워크스루: 코드를 작성한 프로그래머가 테스터나 다른 프로그래머한테 형식을 갖춰 발표

(2) 인스펙션: 개발팀에서 작성한 결과물을 분석하여 개발 표준 위배 여부 판단, 잘못 작성된 부분을 수정하는 작업

- 동적분석: 프로그램을 실행하여 메모리 누수현황, 발생한 스레드 결함 등을 분석하기 위한 도구
  - 종류: 블랙박스 테스트, 화이트박스 테스트

2. 화이트박스 테스트의 정의에 대해 약술하고 테스트 기법 3가지를 쓰시오.
- 정의: 모듈의 원시 코드를 오픈한 상태에서 원시 코드의 모든 문장을 한 번 이상 테스트하여 테스트 케이스를 설계하는 방법

– 종류:

(1) 기초 경로 검사(Base Path Test)

(2) 조건검사, 루프검사, 데이터 흐름 검사

3. 블랙박스 테스트의 정의에 대해 약술하고 테스트 기법 3가지를 쓰시오.
- 정의: 소프트웨어가 수행할 특정 기능을 알기 위해서 각 기능이 완전히 작동되는 것을 입증하는 테스트로, 기능 테스트라고도 한다.

– 종류 (방법별로 키워드 암기)

(1) 동치분할 검사: **입력 자료에 초점**을 맞춰 테스트 케이스를 만들고 검사하는 방법

(2) 경계값 분석: **입력 조건의 중간값보다 경계값에서 오류가 발생될 확률**이 높으므로 입력 조건의 경계값으로 테스트

(3) 원인효과 그래프 검사: **입력 자료 간의 관계와 출력에 영향을 미치는 상황**을 체계적으로 분석 후 효용성이 높은 테스트 케이스를 선정해서 테스트

(4) 오류 예측 검사: **과거 경험이나 테스터의 감각**으로 테스트 → 보충적 검사기법

(5) 비교 검사: **여러 버전의 프로그램에 동일한 자료**를 제공해 동일한 결과가 출력되는지 테스트

4. 개발 단계에 따른 테스트 분류와 그에 대한 설명. **(기출: 애플리케이션 테스트 레벨의 유형, V모델)**

① 단위 테스트: 코딩 직후 소프트웨어 설계 최소 단위인 모듈이나 컴포넌트에 초점을 맞춰 테스트

② 통합 테스트: 단위 테스트가 완료된 모듈을

결합하여 하나의 시스템으로 완성시키는 과정의 테스트

③ 시스템 테스트: 개발된 소프트웨어가 해당 컴퓨터 시스템에서 완벽히 수행되는지 점검하는 테스트

④ 인수 테스트: 개발한 소프트웨어가 사용자의 요구사항을 충족하는지 중점을 두고 테스트 (사용자가 직접 테스트한다.)

4. 통합 테스트 수행 방법 비교

구분	상향식	하향식
모듈	드라이버	스텁
방법	가장 하부의 모듈부터 통합해 가면서 진행	가장 상부의 모듈부터 통합해 가면서 진행
수행	하위 → 상위	상위 → 하위
장점	– 장애 위치 확인 용이 – 모든 모듈이 개발 준비되어 있지 않아도 가능	– 장애 위치 확인 용이 – 초기 프로토타입 가능
단점	– 초기 프로토타입 불가 – 중요한 모듈들이 마지막에 테스트될 가능성	– 많은 스텁 필요 – 낮은 수준 모듈은 부적절한 테스트 가능성
예시	클라이언트만 구현된 상태. 가상의 서버가 드라이버	서버만 구현된 상태. 가상의 클라가 스텁

5. 테스트 하네스 도구의 정의와 구성 요소를 설명

- 정의: 테스트가 실행될 환경을 시뮬레이션 하여 컴포넌트와 모듈이 정상적으로 테스트되도록 하는 도구

– 구성요소

테스트 드라이버	테스트 대상의 하위 모듈을 호출하고, 파라미터 전달하고, 모듈 테스트 수행 후의 결과를 도출하는 도구
테스트 스텁	제어 모듈이 호출하는 타 모듈의 기능을 단순히 수행하는 도구로, 일시적으로 필요한 조건만을 가지고 있는 테스트용 모듈
테스트 슈트	테스트 대상 컴포넌트나 모듈, 시스템에 사용되는 테스트 케이스 집합
테스트 케이스	사용자의 요구사항을 정확하게 준수했는지 확인하기 위한 입력값, 실행조건, 기대결과 등으로 만들어진 테스트 항목의 명세서
테스트 스크립트	자동화된 테스트 실행 절차에 대한 명세서
목(mock) 오브젝트	사전에 사용자의 행위를 조건부로 입력해두면, 그 상황에 맞는 예정된 행위를 수행하는 객체

6. 애플리케이션 성능 측정 지표 4가지와 특징 설명

① 처리량: 일정 시간 내에 애플리케이션이 처리하는 일의 양(Throughput)

② 응답시간: 애플리케이션 요청을 전달한 시간부터 응답이 도착할 때까지 걸린 시간(Response Time)

③ 경과시간: 애플리케이션에 작업을 의뢰한 시간부터 처리가 완료될 때까지 걸린 시간(Turn Around Time)

④ 자원 사용률: 애플리케이션이 의뢰한 작업을 처리하는 동안 CPU 사용량, 메모리 사용량, 네트워크 사용량 등 자원 사용률(Resource Usage)

7. 소프트웨어 테스트의 산출물 4가지

① 테스트 계획서 ② 테스트 케이스

③ 테스트 시나리오 ④ 테스트 결과서

8. 테스트 오라클이란?

- 정의: 테스트 결과가 올바른지 판단하기 위해 사전에 정의된 참 값을 대입하여 비교하는 기법

– 종류

① 참 오라클: 모든 입력값에 적합한 결과를 생성하여 발생한 오류를 모두 검출할 수 있는 오라클

② 일관성 검사 오라클: 애플리케이션 변경이 있을 때, 수행 전과 수행 후의 결과값이 같은지 확인하는 오라클

③ 샘플링 오라클: 일반적인 업무에 사용하는 테스트 오라클로, 특정한 몇 개의 테스트 케이스의 입력값에 대해서만 기대하는 결과를 제공하는 오라클

④ 휴리스틱(추정) 오라클: 임의의 입력값에 대해 올바른 결과를 제공하고, 나머지 값에 대해서는 추정으로 처리하는 오라클 (샘플링 개선방법)

9. 화이트박스 테스트의 검증 기준 4가지는?

- 문장, 분기, 조건, 분기/조건

10. 여러 테스트 케이스의 집합으로 테스트 케이스의 동작 순서를 정의한 기술문서는?

- 테스트 시나리오

11. 소프트웨어 개발 단계와 애플리케이션 테스트를 연결한 모델인 **V-모델**의 순서는?

- 요구사항 → 분석 → 설계 → 구현 → 단위 테스트 → 통합 테스트 → 시스템 테스트 → 인수 테스트

12. 소프트웨어 테스트의 원리

- 결함 집중(파레토 법칙): 결과 80%가 전체 원인 20%에서 발생
- 살충제 패러독스: 같은 테스트 케이스를 반복하면 결함을 발견할 수 없다.
- 오류-부재의 궤변: 사용자의 요구사항을 만족하지 못하는 오류를 제거했어도, 해당 애플리케이션의 품질이 높다고 말할 수 없다.

8. SQL 응용

1. ALTER: 테이블 컬럼 추가, 컬럼 수정, 컬럼 삭제

ALTER TABLE 테이블명 ADD 컬럼 데이터타입 [제약조건];

ALTER TABLE 테이블명 MODIFY 컬럼 데이터타입 [제약조건];

ALTER TABLE 테이블명 DROP 컬럼명;

2. DROP: 테이블 삭제

DROP TABLE 테이블명 [CASCADE|RESTRICT];

CASCADE: 참조하는 테이블까지 함께 제거

RESTRICT: 다른 테이블이 삭제할 테이블을 참조 중이면 제거하지 않음

3. TRUNCATE: 테이블 내 데이터 삭제

TRUNCATE TABLE 테이블명;

4. SELECT

```
SELECT [DISTINCT] 속성명, 속성명,
FROM 테이블이름
WHERE 조건
        (BETWEEN A AND B)
        (LIKE 패턴)
        (IS NULL)
GROUP BY 속성명
HAVING 그룹조건
ORDER BY 속성 [ASC|DESC];
* 서브쿼리 : 쿼리 안에 쿼리
```

5. INSERT (튜플 행 삽입)

INSERT INTO 테이블명 (속성명)
VALUES (데이터)

6. UPDATE (튜플 데이터 수정)

UPDATE 테이블명
SET 속성명 = 데이터
WHERE 조건;

7. DELETE (튜플 삭제)

DELETE FROM 테이블명
WHERE 조건;
예) DELETE FROM 학생 WHERE 이름='민수';

8. 권한 부여 / 권한 회수

GRANT 권한 ON 테이블 TO 사용자
REVOKE 권한 ON 테이블 FROM 사용자

9. JOIN

• INNER JOIN(내부 조인)
: 양쪽 테이블에 모두 있는 것이 표기
SELECT 열
FROM 테이블 INNER JOIN 두 번째 테이블
ON 조인조건
WHERE 검색조건

• OUTER JOIN(외부 조인)
: INNER와 달리 '한 쪽에만 데이터가 있어도' 출력
SELECT 열
FROM 테이블
<LEFT/RIGHT/FULL> OUTER JOIN 두번째 테이블
ON 조인조건
WHERE 검색조건
- LEFT : 왼쪽 테이블의 모든 값이 출력
- RIGHT : 오른쪽 테이블의 모든 값이 출력
- FULL : 왼쪽 또는 오른쪽 모든 값이 출력

• CROSS JOIN(상호 조인)
: 한 쪽 테이블의 모든 행과 다른 쪽 테이블의 모든 행을 조인시키는 기능
SELECT *
FROM 첫 번째 테이블
CROSS JOIN 두 번째 테이블

• SELF JOIN(자체 조인)
: 자기 자신과 조인함 (1개 테이블 이용)
SELECT *
FROM 테이블 별칭A
INNER JOIN 테이블 별칭B ON 조인조건
WHERE 검색 조건

9. 소프트웨어 보안 구축

1. 소프트웨어 개발 보안 요소에 대해서 제시하라
• 기밀성: 시스템 내의 정보와 자원은 인가된 사용자에게만 접근이 허용됨
• 무결성: 시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있음
• 가용성: 인가받은 사용자는 시스템 내의 정보와 자원을 언제라도 사용할 수 있음
• 인증: 시스템 내 정보와 자원을 사용하려는 사용자가 합법적인 사용자인지 확인하는 행위
• 부인 방지: 데이터 송수신한 자가 송수신 사실을 부인할 수 없도록 송수신 증거를 제공

2. 암호 알고리즘의 종류

• 암호화 방식에 따른 분류
(1) 스트림 암호화 기법(LFSR, RC4)
(2) 블록 암호화 방식(DES, SEED, AES, ARIA)
• 공개키 암호화
(1) 공개키는 사용자에게 공개하고, 복호화할 때 비밀키는 관리자가 비밀리에 관리
(2) 대표적으로 RSA 기법이 있음
• 방향에 따른 알고리즘
(1) 단방향: HASH 알고리즘
가. SHA 시리즈: 93년 NSA가 설계하고 NIST에 의해 발표됨.
1) SHA-1: NSA에서 미 정부 표준 지정
나. MD5: 91년 R.Rivest가 MD4를 대체하기 위해 고안한 암호화 해시함수 (블록크기 512비트, 키는 128비트임)
다. N-NASH: 89년 일본의 NTT에서 발표한 암호화 해시함수. 블록/키가 모두 128비트
라. SNEFRU: 90년 R.C.Merkle가 발표한 해시 함수, 32비트 프로세스에서 구현을 용이하게 할 목적으로 개발됨
(2) 양방향 **(전체 기출)**
가. SEED: KISA가 개발한 블록 암호화 알고리즘. 블록크기는 128비트이며 키에 따라 128, 256으로 분류
나. **ARIA**: 04년 국가정보원과 산학연합회가 개발한 블록 암호화 알고리즘
다. **DES**: 1975년 NBS에서 발표한 개인키 암호화 알고리즘. DES를 3번 적용한 3DES도 있음
라. **AES**: 01년 NIST에서 발표한 개인키 암호화 알고리즘. DES의 한계를 느낀 NIST에서 공모한 후 발표
마. **RSA**: 1978년 MIT의 라이베스트, 샤미르, 애들먼에 의해 제안된 공개키 암호화 알고리즘. 큰 숫자를 소인수분해하기 어렵다는 것에 기반하여 제작됨

3. 다음 보안 솔루션에 대해 약술하라.
• VPN(Virtual Private Network)
: 인터넷 등 통신 사업자의 공중 네트워크와 암호화 기술을 이용하여 사용자가 마치 자신의 전용 회선을 사용하는 것처럼 해주는 보안 솔루션
• NAC(Network Access Control)
: 네트워크에 접속하는 내부 PC의 MAC 주소를 IP 관리 시스템에 등록한 후 일관된 보안 관리 기능을 제공하는 보안 솔루션
• ESM(Enterprise Security Management)
: 방화벽, 침입 탐지 시스템, 가상 사설망 등 보안 솔루션을 하나로 모은 통합 보안 관리 시스템

4. 서비스 공격 유형
• Ping of Death: Ping 명령을 전송할 때 패킷의 크기를 인터넷 프로토콜 허용 범위 이상으로 전송하여 공격 대상의 네트워크를 마비시키는 서비스 거부 공격 방법
• SMURFING (스머핑)
: IP나 ICMP의 특성을 악용하여 엄청난 양의 데이터를 한 사이트에 집중적으로 보냄으로써 네트워크를 불능 상태로 만드는 공격 방법
- 해결: 각 네트워크 라우터에서 브로드캐스트 주소를 사용할 수 없게 미리 설정해놓음
• SYN Flooding
: TCP의 3-way-handshake 과정을 의도적으로 중단시켜 공격 대상지인 서버가 대기 상태에 놓여 정상적인 서비스를 수행하지 못하게 하는 공격 방법
- 해결: SYN Flooding에 대비하여 'SYN' 수신 대기 시간을 줄이거나 침입 차단 시스템 도입
• TearDrop
: 데이터 송수신 과정에서 패킷의 크기가 클 때 분할순서를 알기위해 Fragment Offset를 함께 보내는데 이 값을 변경시켜 패킷을 재조립할 때 오류로 인한 과부화를 발생시켜 시스템이 다운되는 공격
- 해결: Fragment Offset이 잘못되면 해당 패킷 폐기
• LAND attack
: 패킷 전송시 송신 IP 주소와 수신 IP 주소를 모두 공격 대상의 IP 주소로 하여 공격대상에게 전송하는 것으로, 이 패킷을 받는 공격 대상은 자기 자신에 대해 무한히 응답한다.
- 해결: 송/수신 IP 주소의 적절성을 검사
• UDP flooding
: 대량의 UDP 패킷을 이용하여 대상 호스트의 네트워크 자원을 소모시키는 공격
• Ping flooding
: 대상 시스템에 ICMP 패킷을 계속해서 보내서 시스템이 요청에 응답하느라 다른 일을 못하도록 하는 공격
• DDoS 공격
: 여러 곳 분산된 공격 지점에서 한 곳의 서버에 대해 분산 서비스 공격을 수행
- 해결: 대역폭 용량과 서버용량 증설, 비정상

트래픽 파악, 방화벽 배포 등

- 피싱(Phishing)
: 진짜 웹 사이트와 거의 동일하게 꾸며진 가짜 웹 사이트를 통해 개인정보를 탈취
- 파밍(Pharming)
: 도메인을 탈취하거나, 악성코드를 통해 DNS의 이름을 속여 진짜 웹사이트로 오인하게 만들어 개인정보 탈취
- 스미싱(Smishing)
: 네트워크상에서 다른 상대방들의 패킷 교환을 엿들음
- 큐싱(Qshing)
: 사용자 인증 등이 필요한 것처럼 속여 QR코드를 통해 악성 앱을 내려받게함
- 랜섬웨어
: 인터넷 사용자의 컴퓨터에 침입해 내부 문서 파일 등 암호화해 사용자가 열지 못하게 하는 방법
- 키 로거
: 컴퓨터 사용자의 키보드 움직임을 탐지해 개인정보를 몰래 빼감
- XSS
: 게시판의 글에 원본과 함께 악성코드를 삽입하여 글을 읽으면 악성코드를 실행하여 클라이언트 공격
- 제로데이 공격 (Zero-Day Attack)
: 조사된 정보를 바탕으로 정보 시스템, 웹 애플리케이션 등의 알려지지 않은 취약점 및 보안 시스템에서 탐지되지 않는 악성코드를 감염시킴
- 무작위 대입공격(Brute-force)
: 패스워드에 사용될만한 문자열의 범위를 정하고, 그 범위 내에서 생성 가능한 패스워드를 활용하는 공격 방법
- APT 공격
: 특정 타깃을 목표로 하여 다양한 수단을 통한 지속적이고 지능적인 맞춤형 공격 기법으로 특수목적의 조직이 하나의 표적에 대해 다양한 IT 기술을 이용하여, 지속적으로 정보를 수집하고, 취약점을 분석하여 피해를 주는 공격 기법
- 스피어 피싱
: 일반적인 이메일로 위장한 메일을 지속적으로 발송하여 링크나 첨부된 파일을 클릭하게 유도한 후 개인정보 탈취
- 백도어
: 프로그램이나 손상된 시스템에 허가되지 않은 접근을 할 수 있도록 정상적인 보안 절차를 우회하는 악성 소프트웨어
- Rainbow Table Attack
: 패스워드 크래킹 기법으로, 패스워드 별로 해시값을 미리 생성해놓은 테이블을 사용하여 Reduction 함수의 반복 수행을 통해 일치하는 해시값으로 패스워드 탈취
- CSRF
: 불특정 다수를 대상으로 로그인된 사용자가 자신의 의지와 상관없이 공격자의 의도에 따라 행위를 하게 하는 공격
- TOCTOU
: 병렬 시스템을 사용할 때 두 시점 사이의 타이밍을 노리는 공격 또는 그런 공격을 가능하게 하는 버그 유형
- 웜(Worm)
: 감염된 컴퓨터에서 자가 복제해 다른 컴퓨터로 복사본을 확산시키는 악성 프로그램
- 트로이 목마
: 정상적인 프로그램으로 가장하여 사용자가 직접 설치하게 유도 후에 백도어를 만들어 공격자가 침입하게 만드는 악성 프로그램. 감염된 이후에 스스로 복제하는 능력은 없지만 시스템 내부 정보를 공격자의 컴퓨터로 빼돌리게 한다.

5. 보안 솔루션 유형

- 방화벽: 내부 보안 정책을 만족하는 트래픽만이 방화벽을 통과할 수 있다.
- 웹 방화벽: 클라이언트가 보낸 요청을 검사하여 악의적인 요청과 침입을 검사하고 차단함
- IDS(Intrusion Detection System): 침입 공격에 대하여 외부 침입을 탐지하는 것을 목표로 하는 보안 솔루션
- IPS(Intrusion Prevention System): 침입 공격에 대하여 방지하는 것을 목표로 하는 보안 솔루션. 침입을 탐지하면 이에 대한 대처까지 수행.
- DMZ: 보안 조치가 취해진 네트워크 영역
- NAC(Network Access Control): 사용자 컴퓨터 및 네트워크 단말기가 네트워크 접근하기 전에 보안 정책 준수 여부를 검사하여 네트워크 접근을 통제하는 보안 솔루션
- DLP(Data Loss Prevention): 기업 데이터 유출을 방지하는 것을 목표로 하는 보안 솔루션
- ESM(Enterprise Security Management): 방화벽, 침입 탐지 시스템, 가상 사설망 등 보안 솔루션을 하나로 모은 통합 보안 관리 시스템

- VPN(Virtual Private Network): 안전하지 않은 공용 네트워크를 이용하여 사설 네트워크를 구성하는 기술

6. 서버인증-사용자 인증기법
- 지식 기반 인증: 사용자가 기억하는 지식(ID/PW)
- 소유 기반 인증: 소지하고 있는 물품, 공인인증서
- 생체 기반 인증: 사용자의 고유한 생체 정보 이용

7. 서버 접근 통제
- DAC(임의적 접근 통제): 정보 소유자가 보안 레벨을 결정하고 이에 대한 접근제어를 설정하는 방식
- MAC(강제적 접근 통제): 중앙에서 정보를 수집하고 분류하여 보안 레벨을 결정하고 정책적으로 접근제어를 수행
- RBAC(역할 기반 접근 통제): 사람이 아닌 직책에 대해 권한을 부여하여 효율적인 권한 관리

10. 프로그래밍 언어 활용

※ 주의: 출력값에서 개행이 잘못된 경우 부분점수 없음

1. 프로그래밍별 참/거짓값 (대소문자 주의)

언어	코드	출력결과
C	printf("%d", 5<3);	0
	printf("%d", 5>3);	1
Java	System.out.println(5<3);	false
	System.out.println(5>3);	true
Python	print(5<3)	False
	print(5>3)	True

2. C언어 이중포인터: 포인터의 개념에 다시 포인터

- 포인터는 접근했을 때 그 주소가 가지고 있는 값을 반환하게 되어 있는데, 그것을 다시 포인터로 활용하면 그 값이 다시 주소로 활용되는 것

```
#include <stdio.h>
int main() {
 int *array[3];
 int a = 12, b = 24, c = 36;
 array[0] = &a;
 array[1] = &b;
 array[2] = &c;
 printf("%d", *array[1] + **array+1);

 // 24(1번 인덱스) + 12(배열의 첫 요소의 값) + 1 = 37
 // 주소의 이름은 array[0]의 주소를 의미한다.
 // array[0]의 주소의 값은 a의 메모리 주소이며,
    그 값(8진수)을 출력하는게 아니라, 그 값을 다시 주소로 찾아간 이후에 그 내부의 값을 꺼내옴
}
```

3. JAVA 관계연산자/비트연산자

```
public static void main(String[] args) {
 int w=3, x=4, y=3, z=5;
 if((w == 2 | w == y & !(y > z)&(1 == x^y != z))
{
 w = x + y;
 if(7 == x^y!= w) {
  System.out.println(w);
 } else {
  System.out.println(x);
 }
} else {
 w = y + z;
 if(7 == y^z != w){
  System.out.println(w);
} else {
  System.out.println(z);
  }
 }
}
```

- 우선순위: 괄호 > 관계 > 비트

첫 번째 if문 결과 T&T&T = T

(1) w==2 | w==y → F | T → 0000 | 1111 → 1111 → T

(2) !(y>z) → !F → T

(3) 1 == x^y!=z → F^T → 0000^1111→ 1111→T (Exclusive OR)

(4) w = x+y; // w=7

두 번째 if문 결과 T

(1) 7 == x^y!=w → F^T → 0000^1111 → T

w값은 7

4. JAVA언어의 관련 용어

(1) instance변수: 객체마다 다른 메모리 공간

(2) static 변수: 클래스변수, 공유변수라 부르며 객체들이 공통된 값을 공유한다. (클래스명. 변수명)

(3) println: 개행한다. print: 개행하지 않는다.

5. Python 기출

```
a=100    # 이진수: 1100100
result=0
for i in range(1, 3):    #1, 2까지만
 result= a >> i
 result= result+1
print(result)
```

- 첫 번째 시프트: 110010 → 50 (+1)
- 두 번째 시프트: 11001 → 25 (+1)
- 마지막 result는 26이다.

5. Python 자료형 종류와 메서드

- List
  - append(): 리스트 끝에 요소를 추가한다.
  - clear(): 리스트의 모든 요소를 삭제한다.
  - copy(): 리스트의 복사본을 돌려준다.
  - count(): 특정 값을 가진 요소의 개수를

반환한다.
- extend(): 다른 리스트의 요소를 현재 리스트 끝에 추가한다.
- index(): 특정 값을 가진 첫 번째 요소를 인덱스를 반환한다.
- insert(index, element): 리스트의 지정된 위치에 요소를 삽입합니다. 예) list.insert(1, 'a')는 리스트의 두 번째 위치에 'a'를 삽입합니다.
- pop([index]): 지정된 인덱스의 요소를 제거하고 그 요소를 반환합니다. 인덱스를 지정하지 않으면 마지막 요소를 제거하고 반환합니다. 예) list.pop(1)은 리스트의 두 번째 요소를 제거하고 그 값을 반환합니다.
- remove(value): 리스트에서 첫 번째로 일치하는 값을 제거합니다. 예) list.remove('a')는 리스트에서 첫 번째로 등장하는 'a'를 제거합니다.
- reverse(): 리스트의 요소 순서를 반대로 뒤집습니다.
- sort(): 리스트의 요소들을 정렬합니다. key는 정렬 기준을 지정하는 함수이고, reverse가 True이면 내림차순으로 정렬합니다. 예) list.sort()는 오름차순으로 정렬하고, list.sort(reverse=True)는 내림차순으로 정렬합니다.

• Dictionary
- clear(): 딕셔너리의 모든 요소를 제거합니다.
- copy(): 딕셔너리의 복사본을 반환합니다.
- fromkeys(): 특정 키와 값으로 만들어서 딕셔너리를 반환
- get(): 지정된 키의 값을 반환합니다. 키가 없으면 default 값을 반환합니다. 예를 들어, dict.get('a', 0)은 키 'a'의 값을 반환하고, 키가 없으면 0을 반환합니다.
- items(): 딕셔너리의 (키, 값) 쌍을 담은 뷰 객체를 반환합니다. 예를 들어, dict.items()는 dict_items([('a', 1), ('b', 2)])을 반환합니다.
- keys(): 딕셔너리의 키를 담은 뷰 객체를 반환합니다. 예를 들어, dict.keys()는 dict_keys(['a', 'b'])를 반환합니다.
- pop(key, default=None): 지정된 키의 값을 반환하고 그 키-값 쌍을 딕셔너리에서 제거합니다. 키가 없으면 default 값을 반환합니다.
- popitem(): 임의의 키-값 쌍을 제거하고 반환합니다. 딕셔너리가 비어 있으면 KeyError가 발생합니다. 예) dict.popitem()은 임의의 키-값 쌍을 반환하고 딕셔너리에서 제거합니다.
- setdefault(): 특정 키의 값을 리턴한다. 키가 없는 경우 특정 값을 삽입한다.
- update(): 특정 키-값을 가진 딕셔너리를 추가
- values(): 딕셔너리의 값을 담은 뷰 객체를 반환합니다. 예를 들어, dict.values()는 dict_values([1, 2])를 반환합니다.

• Tuple: 튜플은 변경할 수 없는 시퀀스형 자료구조로, 주로 데이터를 읽기 전용으로 사용할 때 유용합니다. 튜플은 몇 가지 메서드만을 제공합니다.
- count(value): 튜플에서 지정된 값의 개수를 반환합니다. 예를 들어, (1, 2, 2, 3).count(2)는 2를 반환합니다.
- index(value, [start, [stop]]): 튜플에서 지정된 값의 첫 번째 인덱스를 반환합니다. 시작 및 끝 인덱스를 선택적으로 지정할 수 있습니다. 예를 들어, (1, 2, 3).index(2)는 1을 반환합니다.

• Set: Python의 set 자료형은 고유한 값들의 집합을 나타내는 데이터 구조입니다. 중복되지 않는 요소들의 모임을 표현합니다. set은 변경 가능
- add(element): 집합에 요소를 추가합니다. 예를 들어, set.add(1)은 집합에 1을 추가합니다.
- remove(element): 집합에서 지정된 요소를 제거합니다. 요소가 집합에 없으면 KeyError가 발생합니다. 예를 들어, set.remove(1)은 집합에서 1을 제거합니다.

6. Python 문법 주의할 것들

• for문 횟수
- range(0, 3): 0, 1, 2
- range(6): 0, 1, 2, 3, 4, 5
- for n in range(0, 3):
  print(n, end=', ') # 0, 1, 2가 나온다.

• 슬라이싱
- str = Hello World
- str[0:3]: Hel (맨 뒤에 있는 것은 -1하자)
- str[-3:]: rld (마이너스는 숫자 그대로)
- str[8:]: rld (그 숫자부터 뒤 끝)

– a=[0, 10, 20, 30, 40, 50, 60, 70, 80, 90]

– print(a[:7:2]) # 시작부터 6까지 2개씩 건너뛰자

– 답: [0, 20, 40, 60] # 리스트가 되어야함

```
a = 100
result = 0
for i in range(1, 3):
    result = a >> i  # 쉬프트연산자
    print('{0:08b}'.format(result))
    result = result + 1
print(result)


>>
00110010
00011001
26
```

- 파이썬의 print는 기본적으로 개행이 들어간다.
- continue: 반복문의 처음으로 돌아가서 다음 횟수로 넘어가라.

7. Java의 static 개념

```
class Person {
  static int person_count = 0;
  public int age = 0;
  public String name;

  Person(String param name) {
    this.name = param_name;
    person_count++;
    age++;
  }

public void print_info(){
  System.out.println("인구 :" + person_count);
  System.out.println(name + ":" + age);
  }
}
```

– 위와 같은 코드가 있을 때 다음 출력값은?

```
public static void main(String[] args){
  Person p1 = new Person("홍길동");
  p1.print_info();

  Person p2 = new Person("김길동");
  p2.print_info();
}
```

person_count는 공용공간이다.

인구: 1

홍길동: 1

인구 2 → 이 부분은 공용공간이라 여러 객체가 공유해서 사용한다.

김길동 1

8. C언어의 구조체 문제

```
#include <stdio.h>

struct student {
  char name[12];
  int score1, score2, total, final_total;
};

struct student students[3] = {{"Data1", 53, 88, 0, 0}, {"Data2", 45, 91, 0, 0}, {"Data3", 22, 75, 0, 0}};

int main() {
  struct student* ptr;
  ptr = &students[0];
  (ptr + 1) -> total = (ptr + 1) -> score1 + (ptr + 2) -> score2;
  (ptr + 1) -> final_total = (ptr + 1) -> total + ptr -> score1 + ptr -> score2;
  printf("%d\n", (ptr + 1) -> total + (ptr + 1) -> final_total);

  return 0;
}
```

해설: struct student라는 구조체를 정의하고, 세 명의 학생 데이터를 초기화한 후 포인터를 사용하여 구조체의 멤버를 조작하는 프로그램입니다.

각 학생 데이터에는 이름, 두 개의 점수, 총합 및 최종 총합이 포함됩니다. 이 코드는 특정 학생의 total과 final_total 값을 계산하고 출력합니다.

출력값: 381

11. 응용 SW 기초 기술 활용

운영체제, 데이터베이스, 네트워크 등이 출제

1. 운영체제의 목적을 약술하시오 **(처반사신)**
(영어로도 외울 것)
- 처리 능력(Throughput)
  : 일정 시간 내에 시스템이 처리하는 일의 양을 늘리기 위함
- 반환 시간(Turn Around Time)
  : 시스템에 작업을 의뢰한 시간부터 처리가 완료될 때까지 걸린 시간을 줄이기 위함
- 사용 가능도(Availability)
  : 시스템을 사용할 필요가 있을 때 즉시 사용 가능한 정도를 증대하기 위함
- 신뢰도(Reliability)
  : 시스템이 주어진 문제를 정확하게 해결하는 정도를 올리기 위함

2. 운영체제 운영방식
- 일괄 처리 시스템(Batch Processing System)
- 다중프로그래밍시스템(Multi-Programming)
- 시분할시스템(Time Sharing System)
- 다중처리시스템(Multi-Processing)
- 실시간처리시스템(Real Time Processing)
- 다중모드시스템(Multi-Mode)
- 분산처리시스템(Distributed Processing)

3. UNIX의 구성요소
- 커널(Kernel): 하드웨어를 보호하고, 프로그램과 하드웨어 간의 인터페이스 역할을 한다. 프로세스 관리, 기억장치 관리, 파일 관리, 입출력 관리, 데이터 전송 등을 수행한다.
- 쉘(Shell): 사용자의 명령어를 인식하여 프로그램을 호출하고 명령을 수행하는 명령어 해석기. 시스템과 사용자 간의 인터페이스
- 유틸리티 프로그램: 일반 사용자가 작성한 응용 프로그램을 처리하는 데 사용함. 예시로 에디터, 컴파일러, 인터프리터, 디버거가 있다.

4. 가상메모리 기법의 페이징 기법에서 페이지 부재가 계속 발생되어 프로세스가 수행되는 시간보다 페이지 교체에 소비되는 시간이 더 많이 드는 현상을 일컫는 용어는?
- 스래싱(Thrashing)

5-1. 2개의 프로세스가 서로 다른 프로세스가 점유하고 있는 자원을 요구하며 무한정 기다리는 현상인 교착상태(DeadLock)이 발생하기 위한 4가지 조건은? **(기출, 상점비환)**
- 상호배제(Mutual Exclusion)
  : 한 번에 한 개의 프로세스만이 공유 자원을 사용할 수 있어야한다.
- 점유와 대기(Hold and Wait)
  : 이미 자원을 가진 프로세스가 다른 자원의 할당을 요구한다.
- 비선점(Non-Preemption)
  : 프로세스에 할당된 자원은 사용이 끝날 때까지 강제로 빼앗을 수 없다.
- 환형 대기(Circular Wait)
  : 이미 자원을 가진 프로세스가 앞이나 뒤의 프로세스 자원을 요구한다.

5-2. 교착상태의 해결방법 **(예발회회)**
- 예방(Prevention): 교착 상태 발생의 4가지 조건 중에서 어느 하나를 제거함
- 회피(Avoidance): 교착 상태가 발생할 가능성을 배제하지 않고, 교착 상태가 발생하면 적절하게 빠져나감. **은행원 알고리즘**이 사용됨
- 발견(Detection): 시스템에 교착 상태가 발생했는지 점검하여 교착 상태에 있는 프로세스와 자원을 발견하는 것
- 회복(Recovery): 교착 상태를 일으킨 프로세스를 종료하거나 교착 상태의 프로세스에 할당된 자원을 선점하여 프로세스나 자원을 회복함

6. Dijkstra가 제안한 상호 배제 알고리즘으로 각 프로세스가 임계구역에 대해 각각의 프로세스들이 접근하기 위하여 사용되는 P와 V 연산을 통해 프로세스 사이 동기를 유지하고 상호 배제의 원리를 보장하는 알고리즘은?
- 세마포어(Semaphore)

7. 무결성 제약조건

개체무결성	기본 테이블의 기본키를 구성하는 어떤 속성도 Null값이나 중복값을 가질 수 없다.
참조무결성	외래키 값은 Null이거나 참조 릴레이션의 기본키 값과 동일해야한다. 즉, 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다는 규정
도메인무결성	주어진 속성 값이 정의된 도메인에 속한 값이어야한다는 규정
고유무결성	특정 속성에 대해 고유한 값을 가지도록 조건이 주어진 경우, 그 속성값은 모두 달라야하는 제약 조건
NULL무결성	릴레이션의 특정 속성 값이 Null이 될 수 없도록 하는 규정
키 무결성	한 릴레이션(테이블)에는 최소한 하나의 키가 존재해야 하는 제약조건

8. 관계 데이터베이스의 논리적 설계 과정에서 하나의 릴레이션에 많은 속성이 있어 데이터의 종속과 중복으로 여러 가지 문제가 발생한다. 이 현상은 릴레이션을 처리하는 데 발생하는 여러 문제로 **삽입, 삭제, 갱신**이 있는데 이것은 무엇인가? **(기출)**

- 이상(Anomaly)

(1) 삽입 이상: 자료를 삽입할 때 의도하지 않은 자료까지 삽입해야만 자료를 테이블에 추가가 가능한 현상

(2) 삭제 이상: 어떤 정보를 삭제하면, 유용한 다른 정보까지 삭제되어버리는 현상

(3) 갱신 이상: 중복된 데이터 중 일부만 수정되어 데이터 모순이 일어나는 현상

9-1. 데이터베이스 회복 중 회복기법

- 즉시갱신 회복기법
  : 트랜잭션 수행 도중에도 변경 내용을 즉시 데이터베이스에 기록
- 지연갱신 회복기법
  : 커밋이 발생하기 전까지 데이터베이스에 기록하지 않음. 변경 내용을 로그 파일에만 저장
- 검사시점 회복기법
  : 체크포인트 회복 기법. 장애 발생시 검사점 이전에 처리된 트랜잭션은 회복에서 제외하고 검사점 이후에 처리된 트랜잭션은 회복 작업수행
- 그림자 페이징
  : 트랜잭션이 실행되는 메모리상의 현재 페이지테이블과 하드디스크의 그림자 페이지테이블을 이용한다.
- 미디어 회복기법
  : 디스크와 같은 비휘발성 저장 장치 대비 회복 기법으로 백업, 미러링, RAID 등을 이용해 별도의 물리적인 저장장치에 덤프

9-2. 데이터베이스 회복 중 REDO, UNDO

- REDO(재수행)
- UNDO(취소)

10. **병행제어**란 동시에 여러 개의 트랜잭션을 병행 수행할 때, 동시에 실행되는 트랜잭션들이 데이터베이스에 일관성을 파괴하지 않도록 트랜잭션 간의 상호 작용을 제어하는 것이다. 여기서 한꺼번에 로킹할 수 있는 객체의 크기에 대해서 약술하시오.

- **로킹**(어떤 단위를 액세스하기 전에 잠금을 요청해서 Lock이 허락되어야만 액세스 가능) 단위

: 병행 제어에서 한꺼번에 로킹할 수 있는 객체의 크기

: 로킹 단위가 크면 로크 수가 작아 관리하기 쉽지만, 병행성 수준은 낮아진다.

: 로킹 단위가 작으면, 로크 수가 많아 관리하기 복잡해 오버헤드가 증가하지만, 병행성 수준이 높아진다.

11. 서브네팅에 대해서 서술하세요

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

12-1. IPv6의 특징을 약술하라. **(기출)**

- 특징

(1) 128비트의 확장된 주소 공간

(2) 인증성, 기밀성, 데이터 무결성 지원

(3) IPv4 확장헤더를 추가하여 네트워크 기능 확장

(4) 시스템 관리, 주소 자동설정으로 시스템 관리 시간과 비용 줄여줌

- 주소체계: Unicast, Anycast, Multicast
  - Unicast: 단일송신/단일수신(1:1)
  - Anycast: 단일송신/가장 가까운 단일수신(1:1)
  - Multicast: 단일송신/다중수신 (1대多)

12-2. IPv4 특징과 종류

- 8bit씩 4개 필드로 구성되어 있음 (총 32bit)

• 유니캐스트, 멀티캐스트, 브로드캐스트가 있음
• 헤더옵션이 40byte로 제한되어있음

13. DNS란?
• 문자로 된 도메인 네임을 컴퓨터가 이해할 수 있는 IP 주소로 변환하는 역할을 하는 시스템을 DNS(Domain Name System)이라고 한다.

14. OSI 7계층에 대해 서술하시오 **(물데네전세표응)**
① 하위 계층: 물리→데이터링크→네트워크 계층
② 상위 계층: 전송→세션→표현→응용 계층

• **물리계층(Physical Layer)**
: 전송에 필요한 두 장치 간의 실제 접속과 절단 등 기계적, 전기적, 절차적 특성에 관한 규칙을 정의한다.
표준) RS-232C, X.21
장비) **리피터**(거리가 증가할수록 감쇠하는 디지털 신호의 장거리 전송을 위해 수신한 신호를 재생하거나 출력 전압을 높여 전송),
허브(여러 대의 컴퓨터를 연결하여 네트워크로 보내거나 하나의 네트워크로 수신된 정보를 여러 컴퓨터에 송신하는 장비)

• **데이터 링크 계층(Data Link Layer)**
: 2개의 인접한 개방 시스템 간의 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 시스템 간 연결 설정과 유지 및 종료를 담당
: 송/수신 측 속도 차이 해결 위한 **흐름 제어 기능**
: 프레임의 시작, 끝을 구별하기 위해 **프레임 동기화 기능**
: 오류 검출과 회복을 위한 **오류 제어 기능**
: 프레임 순서적 전송을 위한 **순서 제어 기능**
표준) HDLC, LAPB, MAC, LLC, PPP 등
장비) 랜카드, 브리지, 스위치(브리지와 같이 LAN과 LAN을 연결하여 훨씬 더 큰 LAN을 만드는 장치)

• **네트워크 계층(Network Layer, 망 계층)**
: 개방 시스템 간의 네트워크 연결을 관리하는 기능과 데이터의 교환 및 중계 기능을 한다.
: 네트워크 연결을 설정/유지/해제하는 기능
: 경로설정, 데이터 교환 및 중계, 트래픽 제어, 패킷 정보 전송을 수행
표준) X.25, IP
장비) 라우터(브리지와 같이 LAN과 LAN의 연결 기능에 데이터 전송의 최적 경로를 선택하는 기능이 추가된 장치

• **전송 계층(Transport Layer)**
: 논리적 안정과 균일한 데이터 전송 서비스를 제공함으로써 종단 시스템(End-to-End) 간에 투명한 데이터 전송을 가능케 한다.
: 상위 3계층의 인터페이스를 담당
: 종단 시스템 간의 전송 연결 설정, 데이터 전송, 연결 해제 기능
: 주소 설정, 다중화, 오류제어, 흐름제어
표준) TCP, UDP
* TCP: 두 개의 호스트를 연결하고 데이터스트림을 교환하게 해주는 네트워크프로토콜. 데이터와 패킷이 보내진 순서대로 전달하는 것을 보장
* UDP: IP를 사용하는 네트워크 내에서 컴퓨터 간 메시지를 교환하고 제한된 서비스만을 제공하는 통신프로토콜. 도착한 데이터의 손상 정보를 알 수 있는 checksum제공. 비연결형으로 고속전송
장비) 게이트웨이

• **세션 계층(Session Layer)**
: 송/수신 간의 관련성을 유지하고 대화 제어를 담당
: 대화 구성 및 동기 제어
: 동기점은 오류가 있는 데이터의 회복을 위해 사용하는 것으로, 소동기점과 대동기점이 있다.
예제 프로토콜) NetBIOS, RPC, PPTP 등
* NetBIOS: 세션 계층에서 네트워크 애플리케이션 간의 통신을 돕기 위해 사용합니다.
* RPC: 네트워크 상에서 원격 프로시저를 호출하는 기술로, 클라이언트와 서버 간의 세션 설정하여 관리한다.

• **표현 계층(Presentation Layer)**
: 응용 계층으로부터 받은 데이터를 세션 계층에 보내기 전에 통신의 적당한 형태로 변환하고, 세션 계층에서 받은 데이터는 응용 계층에 맞게 변환한다.
표준) JPEG, MPEG, SMB, AFP

• **응용 계층(Application Layer)**
: 사용자가 OSI 환경에 접근할 수 있도록 서비스를 제공한다. 정보 교환, 전자사서함, 파일 전송, 가상 터미널 등을 제공한다.
표준) HTTP, FTP, DHCP, DNS

15. TCP/IP 프로토콜의 계층에 대해서 서술하시오
• 응용 계층(Application Layer)
: OSI 7계층의 응용, 표현, 세션 계층
: 응용 프로그램 간의 데이터 송수신 제공
종류) TELNET, FTP, DNS, SNMP, HTTP

• 전송 계층(Transport Layer)
: OSI 7계층의 전송 계층
: 호스트 간의 신뢰성 있는 통신제공
종류) TCP, UDP, RTCP

• 인터넷 계층(Internet Layer)
: OSI 7계층의 네트워크 계층
: 데이터 전송을 위한 주소 지정, 경로 설정을 제공함
종류) IP, ICMP, IGMP, ARP, RARP

* ICMP: 인터넷 프로토콜의 비신뢰적인 특성을 보완하기 위한 프로토콜. IP 패킷을 처리할 때 발생하는 에러 발생 원인을 알려주거나 네트워크 상태를 진단해주는 기능을 제공
* IGMP: 인터넷 그룹 관리 프로토콜은 호스트 컴퓨터와 인접 라우터가 멀티캐스트 그룹 멤버십을 구성하는 데 사용하는 통신 프로토콜
* ARP: 호스트의 IP 주소를 호스트와 연결된 네트워크 접속 장치의 물리적 주소(MAC 주소)로 바꿈 (IP to MAC)
* RARP: ARP와 반대로 물리적 주소(MAC 주소)를 IP 주소로 변환하는 기능을 함 (MAC to IP)

• 네트워크 액세스 계층
: OSI 7계층의 데이터 링크, 물리 계층
: 실제 데이터(프레임)을 송수신하는 역할
종류) Ethernet, IEEE 802, X.25, ARQ, HDLC 등
* X.25: 패킷 교환망을 통한 DTE와 DCE 간의 인터페이스를 제공하는 프로토콜
* RS-232C: 공중 전화 교환망을 통한 DTE와 DCE 간의 인터페이스를 제공하는 프로토콜
* HDLC: 비트 위주의 데이터 링크 제어 프로토콜

16. 스위치의 분류를 작성하시오

분류	내용
L2 스위치	- OSI 2계층에 속하는 장비 - MAC 주소를 기반으로 프레임 전송 - 일반적으로 부르는 스위치
L3 스위치	- OSI 3계층에 속하는 장비 - L2 스위치에 라우터 기능이 추가된 것으로 IP 주소를 기반으로 패킷 전송 - 서로 다른 네트워크 간의 연결이 가능
L4 스위치	- OSI 4계층에 속하는 장비 - 로드 밸런서가 달린 L3 스위치로, TCP/UDP를 기반으로 서버의 부하가 적은 곳에 로드 밸런싱 기능을 제공
L7 스위치	- OSI 7계층에 속하는 장비 - IP주소, TCP/UDP 포트 정보에 패킷 내용까지 참조하여 세밀하게 로드밸런싱 제공

17. 데이터베이스 조건에 만족하는 튜플을 찾거나 순서대로 정렬할 때 기준이 되는 속성인 키들에 대해 약술하시오
- **기본키:** 후보키 중에서 특별히 선정된 주키, 중복된 값을 가질 수 없으며 한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성, 기본키는 NULL을 가질 수 없다.
- **후보키:** 기본키로 사용할 수 있는 속성들. 모든 릴레이션에는 반드시 하나 이상의 후보키가 존재하며 후보키는 유일성과 최소성을 만족시켜야 한다.
- **슈퍼키:** 한 릴레이션 내에 있는 속성들의 집합으로 구성된 키로 릴레이션을 구성하는 모든 튜플에 대해 유일성은 만족하지만 최소성은 만족시키지 못함
- **대체키:** 후보키가 둘 이상일 때 기본키를 제외한 나머지 후보키
- **외래키:** 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합

18. **교환 회선방식**의 종류에 대해 나열하라
(1) 교환회선: 회선교환, 축적교환으로 나뉨
- 회선교환-물리적 전용선을 활용하여 데이터 전달 경로가 정해진 후 동일 경로로만 전달

(2) 축적교환: 메시지 교환, 패킷 교환으로 나뉨
- 메시지-하나의 메시지 단위로 저장-전달 방식에 의해 데이터를 교환
- 패킷-메시지를 일정한 길이의 전송 단위인 패킷으로 나누어 전송하는 방식, 실패한 패킷은 재전송이 가능하고 현재 컴퓨터 네트워크에서 주로 사용하는 방식이다.

(3) 패킷교환: 가상회선, 데이터그램으로 나뉨

- **가상회선**-데이터를 전송하기 전에 논리적 연결이 설정되는데 이를 가상회선이라 한다. (회선교환 방식 + 데이터그램 방식)의 장점을 결합한 통신 방식. 처음 패킷을 최적 경로로 고정하고 그 다음 패킷을 나눠 고속 전송, 통신기술에는 ATM이 있으며 정해진 시간 안에 다량의 데이터를 연속으로 보낼 때 좋다.
- **데이터그램**-데이터를 전송하기 전에 논리적 연결을 하지 않고, 패킷이 독립적으로 전송. 패킷교환 방식으로 동작하면서 IP 주소를 사용하는 인터넷을 의미. 가상회선 방식과 달리 특정 교환기가 고장나도 그 경로를 피해서 전송할 수 있기 때문에 신뢰성가 가능하다.

19. 프로토콜의 기본 3요소는?

- 구문: 데이터의 형식이나 부호화 및 신호 레벨을 규정한 것
- 의미: 전송의 조작이나 오류 제어를 위한 제어 정보에 대한 규정
- 타이밍: 접속되어 있는 개체 간의 통신 속도의 조정이나 메시지의 순서 제어 등을 규정

20. IP 주소와 서브넷마스크가 주어졌을 때 네트워크 주소와 해당 네트워크 주소와 브로드캐스트 주소를 제외한 호스트 개수

* 서브넷: 부분 네트워크 / 마스크: AND 연산

예) 10101101 (8비트)에서 가장 마지막 4개를 0으로 만들고 싶다. (하나라도 0이면 0)

```
    10101101
AND 11110000  <- 마스크 기능
------------
    10100000
```

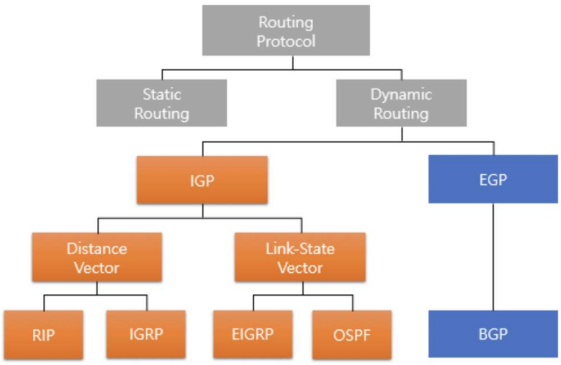
살리고 싶은 것은 1, 없애고 싶은 것은 0으로 AND 연산해버린다.

서브넷 마스킹하려는 곳을 전부 1로 채운다.

```
네트워크주소: 11010011 10101000 01010011 00000000
서브넷마스크: 11111111 11111111 11111111 00000000
-------------------------------------------------
              211      168      83       0
```

마스킹하면 가장 마지막의 비트만 바뀔 수 있다.
이 방식으로 해당 클래스를 네트워크로 구성할 수 있다.

21. 라우팅 프로토콜의 종류와 예시



구 분	특 징
내부 라우팅 프로토콜(IGP) 내부게이트웨이 프로토콜 (자치시스템 내)	- RIP: **거리벡터 알고리즘**. 홉수만 본다. (Hop count) - OSPF: **링크 상태 변화**시에만 라우팅 정보를 전송, 홉수/대역폭/지연시간 등을 고려하며 다익스트라 알고리즘을 기반으로 한다.
외부 라우팅 프로토콜(EGP) 외부게이트 프로토콜 (자치시스템 간)	- EGP: 다른 그룹과 라우팅 정보를 교환하는 프로토콜 - BGP: 유일한 EGP의 프로토콜로 AS(Autonomous System)라는 집단끼리 상호작용하는 프로토콜.

12. 제품 소프트웨어 패키징

1. DRM의 정의와 기술적 요구사항을 약술하라
- 정의: 디지털 콘텐츠의 생성부터 실제 사용자까지 유통되는 과정에 걸쳐 콘텐츠를 안전하게 관리 및 보호하고 허가된 사용자만이 접근할 수 있도록 제한하는 기술
- 기술적 요구사항:

(1) 지속적 보호 (2) 이용 편리성 (3) 유연성
(4) 통합의 용이성

2. 디지털 콘텐츠의 사용권한 유형
- 렌더 퍼미션: 사용자에게 콘텐츠가 표현되고 이용되는 권리 형태를 정의
- 트랜스포트 퍼미션: 사용자들 간의 권리 교환가능
- 데리버티브 퍼미션: 콘텐츠 추출 변형이 가능

3. 버전관리 도구
- CVS: 동시 버전 시스템, 여러 개발자가 협력하여 작업할 수 있으며 GNU 일반 공중 사용 허가서하에 배포됨. 서버와 클라이언트 구조의 중앙 통제 방식으로 다수의 인원이 동시에 범용적인 운영체제로 접근가능한 버전 관리 도구.
- RCS: 파일 수정을 한 사람만으로 제한하는 도구. 파일 수정 중엔 파일을 잠금처리하며 다수의 사람들이 동시에 수정할 수 없음
- SVN: CVS보다 속도 개선, 2000년부터 콜랩넷에서 개발, 아파치 최상위 프로젝트. 소스코드를 올리면서 로그를 남기고 상황에 따라 롤백이 가능한 중앙 집 중형 버전 관리 시스템
- Bitkeeper: SVN과 비슷하나 대규모 프로젝트에서 빠른 속도를 내도록 개발
- Git: Linux 커널 개발 이용하기 위해 개발. 로컬 저장소에서 소스코드를 관리한 후 작업이 완료되면 중앙 저장소에 Push하는 방법으로 프로그램을 관리하는 분산 버전 관리 시스템

4. 빌드 자동화 도구의 처리 절차
- 종속성 다운로드 → 컴파일 → 패키징 → 테스트 → 배포

디자인 패턴 정리

목적	디자인 패턴	설명
생성	빌더	Builder 생성 단계를 캡슐화하여 구축 공정을 동일하게
	프로토타입	Prototype 기존 객체를 복제해서 새 객체를 생성 가능
	팩토리메서드	Factory Method 상위 클래스에서 객체 생성 인터페이스 정의, 하위 생성
	추상 팩토리	Abstract Factory 생성군을 하나로 모아두고, 팩토리 중에 선택
	싱글턴	Singleton 유일한 하나의 인스턴스를 보장하는 패턴
구조	브릿지	Bridge 추상과 구현을 분리해서 결합도를 낮춘 패턴
	데코레이터	Decorator 소스를 변경하기 않고 기능을 확장하는 패턴
	퍼사드	Facade 하나의 인터페이스 통해 느슨한 결합 제공
	플라이웨이트	Flyweight 대량의 작은 객체들을 공유하는 패턴
	프록시	Proxy 대리인이 대신 그 일을 처리하는 패턴. 실체 객체를 호출하면 중간에 가로채서 다른 동작을 수행.
	컴포지트	Composite 개별 객체와 복합 객체를 클라이언트에서 동일사용
	어댑터	Adapter 인터페이스 때문에 사용 못하는 클래스를 함께 사용
행위	인터프리터	Interpreter 언어 규칙 클래스를 이용하는 패턴
	템플릿메서드	Template Method 알고리즘 골격의 구조를 정의한 패턴
	책임의 고리	Chain of Responsibility 객체끼리 연결 고리를 만들어 내부적으로 전달하는 패턴
	커맨드	Command 요청 자체를 캡슐화해 파라미터로 넘기는 패턴
	이터레이터	Iterator 내부 표현은 보여주지 않고 순회하는 패턴
	미디에이터	Mediator 객체 간 상호작용을 캡슐화한 패턴
	메멘토	Memento 상태 값을 미리 저장해두었다가 복구하는 패턴
	옵저버	Observer 상태가 변할 때 의존자에게 알리고 자동 업데이트
	스테이트	State 객체 내부 상태에 따라서 행위를 변경하는 패턴
	스트레티지	Strategy 여러 알고리즘을 캡슐화해 대체가 가능하게 함
	비지터	visitor 오퍼레이션을 별도의 클래스에 새롭게 정의한 패턴

시험 들어가기 전 외울 것

[관계 데이터 모델 구성요소]

릴레이션	행과 열로 구성된 테이블
튜플	릴레이션의 행(Row)에 해당하는 요소
속성	릴레이션의 열(Column)에 해당하는 요소
카디널리티	튜플(Row)의 수
차수	애트리뷰트(Column)의 수

[접근통제 유형]

DAC	임의접근통제	신분(소유자 등) 기준
MAC	강제접근통제	정책, 보안의 등급
RBAC	역할기반접근통제	신분X, 개인역할O

[테스트 종류 정리]

정적검사	워크스루, 인스펙션, 동료검토
동적검사	화이트박스 테스트 / 블랙박스 테스트

[정적 검사 중 "기술리뷰"]

워크스루	명세서 배포하여 사전검토 회의 진행
인스펙션	엄격. 작성자를 제외한 다른 전문가들
기술리뷰	정의된 계획, 명세를 준수하는지 검토
관리리뷰	진행상황 전반적 검토, 통제/의사결정

[동적 검사1 - 화이트박스 테스트]

❍ 원시코드를 오픈하여 내부 로직을 보면서 수행하는 테스트

기초경로검사	Base Path Testing. 테스트 설계자가 절차적 설계 논리적 복잡성 측정할 수 있게
제어구조검사	① 조건검사: 논리적 조건을 테스트 ② 반복검사: 반복구조 초점 ③ 데이터흐름검사: 사용자의 값을 변수에 넣을 때 변수 정의, 위치 어떻게 했는지 테스트

❍ 화이트박스 테스트의 구체적인 예시

구문커버리지	프로그램 내 모든 명령문을 적어도 한 번 수행
결정커버리지	각 분기의 결정 포인트의 전체 조건식을 최소 한 번은 T/F 모두 수행
조건커버리지	각 분기의 결정 포인트 내의 개별 조건식을 최소 한 번은 T/F 모두 수행
다중조건커버리지	결정 조건 내의 모든 개별 조건식을 모든 가능한 조합을 100% 보장하는 커버리지
제어흐름테스트	제어 구조를 그래프 형태로 나타내어 로직 테스트
루프 테스트	프로그램의 반복 구조에 초점을 맞춰 실시하는 테스트

[동적 검사2 - 블랙박스 테스트]

❍ 특정 기능을 알기 위해서 각 기능을 테스트 (기능테스트)

동치분할검사	정상/비정상 입력자료 개수 맞춰서 테스트
경계값분석	입력 조건의 경계값을 테스트
원인효과 그래프검사	여러 입출력 데이터 분석 후 상황을 분석하여 효율성 높은 테스트 케이스 선정
오류예측검사	과거의 경험이나 확인자의 감각으로
비교검사	동일 테스트케이스 여러 버전 프로그램에

[요구사항 분석 중 구조적 분석기법]

❍ 주요 구조적 분석 기법 도구

자료흐름도	DFD. 자료 흐름과 변환 과정을 도형으로 기술
자료 사전	DD. 자료흐름도 자료를 더 자세히 정의, 기록
소단위 명세서	Mini-Spec. 데이터 흐름도에서 있는 처리항목을 1~2페이지 소규모로 정리

[데이터베이스 이상현상]

❍ 데이터의 중복성으로 인해 릴레이션 조작할 때 발생하는 비합리적인 현상

삽입 이상	정보 저장시 불필요한 세부정보까지 입력해야됨
삭제 이상	정보 삭제시 원치 않은 다른정보까지 삭제됨
갱신 이상	중복 데이터 중 특정 부분만 수정되어서 중복된 값이 모순을 일으키는 경우

[소프트웨어 생명주기 모델]

폭포수 모델	각 단계를 확실히 마무리 지은 후 다음
프로토타이핑	요구사항 프로토타입 구현, 피드백 반영
나선형 모델	위험 최소화. 점진적으로 완벽한 시스템

[병행 제어 기법의 종류]

로킹	트랜잭션의 순차적 진행을 보장하는 기법
낙관적 검증	일단 트랜잭션 수행. 종료시 검증
타임스탬프순서	트랜잭션 실행하기전 타임스탬프로 수행

[소프트웨어 개발 보안의 3대 요소]

기밀성	인가되지 않은 개인 혹은 시스템 접근에 따른 정보 공개 및 노출을 차단하는 특성
무결성	정당한 방법을 따르지 않고서는 데이터가 변경될 수 없음. 데이터 정확성/완전성 보장
가용성	권한을 가진 사용자나 애플리케이션이 원하는 서비스를 지속해서 사용할 수 있도록 보장

[스키마의 유형, 데이터베이스 설계순서]

❍ 스키마 유형

외부	사용자, 응용프로그래머가 필요로 하는 논리적 구조
개념	개체간 관계, 접근 권한, 보안 정책 등 정의
내부	물리적 구조를 정의한 스키마. 저장 장치의 관점에서 봄

❍ 데이터베이스 설계순서

요구조건 분석→개념적설계→논리적설계→물리적설계→구현

[UI 설계 원칙]

직관성	누구나 쉽게 이해하고, 쉽게 사용할 수 있어야됨
유효성	정확하고 완벽하게 사용자의 목표 달성해야됨
학습성	초보, 숙련자가 쉽게 배우고 사용할 수 있어야됨
유연성	사용자의 요구사항을 최대한 포용하고 실수 방지

SQL과 JOIN 정리

쿼리 종류	SQL 키워드	사용 예제와 의미	
DDL (Definition)	CREATE	CREATE TABLE table_name (column1 datatype, column2 datatype, ...);	테이블 생성
	ALTER	ALTER TABLE table_name ADD column_name datatype; ALTER TABLE table_name MODIFY column_name datatype; ALTER TABLE table_name DROP COLUMN column_name;	데이터베이스 객체 수정 (테이블이나 컬럼 등)
	DROP	DROP TABLE table_name;	테이블 없애기
	TRUNCATE	TRUNCATE TABLE table_name;	데이터 삭제, 구조는 유지
DML (Manipulation)	SELECT	SELECT column1, column2, ... FROM table_name WHERE condition;	데이터 조회
	INSERT	INSERT INTO table_name (column1, column2, ...) VALUES (value1, value2, ...);	데이터 삽입
	UPDATE	UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;	데이터 갱신
	DELETE	DELETE FROM table_name WHERE condition;	데이터 삭제
DCL (Control)	GRANT	GRANT SELECT, .. ON TABLE TO USER	유저에게 권한 부여
	REVOKE	REVOKE SELECT, .. ON TABLE FROM USER	주어진 권한 회수

조인 종류	정의	예시
내부 조인 (Inner Join)	두 테이블에서 일치하는 조건이 있는 행만 반환합니다.	SELECT * FROM A INNER JOIN B ON A.id = B.id;
왼쪽 외부 조인 (Left Outer Join)	왼쪽 테이블의 모든 행과 오른쪽 테이블에서 일치하는 행을 반환합니다. 일치하지 않는 오른쪽 테이블의 행은 NULL 로 채워집니다.	SELECT * FROM A LEFT OUTER JOIN B ON A.id = B.id;
오른쪽 외부 조인 (Right Outer Join)	오른쪽 테이블의 모든 행과 왼쪽 테이블에서 일치하는 행을 반환합니다. 일치하지 않는 왼쪽 테이블의 행은 NULL 로 채워집니다.	SELECT * FROM A RIGHT OUTER JOIN B ON A.id = B.id;
완전 외부 조인 (Full Outer Join)	양쪽 테이블의 모든 행을 반환합니다. 일치하지 않는 행은 NULL 로 채워집니다.	SELECT * FROM A FULL OUTER JOIN B ON A.id = B.id;
자연 조인 (Natural Join)	두 테이블에서 동일한 이름의 열을 기준으로 일치하는 행을 결합합니다. ON 절이 필요 없으며, 동일한 이름의 열만 기준으로 자동 결합됩니다.	SELECT * FROM A NATURAL JOIN B;
세타 조인 (Theta Join)	지정된 조건(Theta 조건)에 따라 두 테이블을 결합합니다. ON 절을 사용하여 원하는 조건을 설정할 수 있습니다.	SELECT * FROM A JOIN B ON A.id < B.id;
교차 조인 (Cross Join)	두 테이블의 모든 행의 조합을 반환합니다. Cartesian Product 라고 부르며, 조인 조건 없이 두 테이블의 모든 조합을 생성합니다.	SELECT * FROM A CROSS JOIN B;
자기 조인 (Self Join)	동일한 테이블을 두 번 사용하여 자기 자신과 조인합니다. 주로 별칭(Alias)을 사용하여 두 테이블을 구분합니다.	SELECT * FROM A AS A1 JOIN A AS A2 ON A1.parent_id = A2.id;

네트워크 & 프로토콜 정리

계층	계층명	장비, 프로토콜	계층설명
1	물리 계층	케이블, 허브, 리피터 / RS-232, V.35	전기적, 기계적 신호 전송을 담당합니다.
2	데이터 링크 계층	스위치, 브리지 / Ethernet, PPP, FDDI	프레임 단위로 데이터 전송, MAC 주소를 이용한 통신을 관리합니다.
3	네트워크 계층	라우터 / IP, ICMP, IPsec	패킷 단위로 데이터를 전송하며, 라우팅을 통해 목적지까지 데이터를 전달합니다.
4	전송 계층	TCP, UDP	데이터 전송의 신뢰성 및 흐름 제어, 오류 검출 및 복구를 담당합니다.
5	세션 계층	NetBIOS, PPTP, RTP	세션 설정, 관리, 종료를 담당하며, 데이터 교환을 동기화합니다.
6	표현 계층	SSL, TLS, JPEG, MPEG	데이터의 형식, 인코딩, 암호화를 담당하여 데이터의 표현을 처리합니다.
7	응용 계층	HTTP, FTP, SMTP, DNS	사용자와 가장 가까운 계층으로, 응용 프로그램 간의 데이터 교환을 지원합니다.

차이점	TCP	UDP
연결방식	연결형	비연결형
패킷교환방식	가상회선 방식	데이터그램 방식
통신방식	1:1 통신	1:1, 1:N, N:N
신뢰성	높다 (3-way-handshake)	낮다

프로토콜	설명
ARP	ARP는 IP 주소를 물리적 네트워크 주소(MAC 주소)로 변환하는 데 사용됩니다. 이는 동일한 네트워크 내의 장치들이 서로 통신할 수 있도록 도와줍니다.
RARP	RARP는 MAC 주소를 IP 주소로 변환하는 데 사용됩니다. 주로 디스크 없는 워크스테이션이 부팅 시 자신의 IP 주소를 알기 위해 사용됩니다.
ICMP	네트워크 장치들 간의 진단 및 오류 메시지를 전달하는 데 사용됩니다. 이는 IP 네트워크의 상태를 모니터링하고, 문제를 감지하는 데 도움을 줍니다.
IGMP	멀티캐스트 그룹 멤버십을 관리하는 데 사용됩니다. 이를 통해 멀티캐스트 라우터는 네트워크 상의 호스트가 어떤 멀티캐스트 그룹에 가입 또는 탈퇴했는지를 알 수 있습니다.

구 분	특 징
내부 라우팅 프로토콜(IGP) (자치 시스템 내)	- RIP: 거리벡터 알고리즘. 홉수만 본다. (Hop count) - OSPF: 링크 상태 변화시에만 라우팅 정보를 전송, 홉수/대역폭/지연시간 등을 고려하며 다익스트라 알고리즘을 기반으로 한다.
외부 라우팅 프로토콜(EGP) (자치 시스템 간)	- EGP: 초기 인터넷 라우트 프로토콜. 다른 그룹과 라우팅 정보를 교환하는 프로토콜 - BGP: EGP의 후속 프로토콜로, AS(Autonomous System)라는 집단끼리 상호작용하는 프로토콜. (경로 벡터 라우팅 프로토콜 형태)

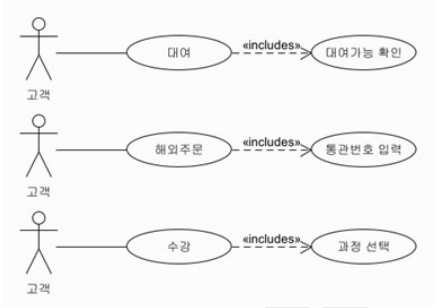
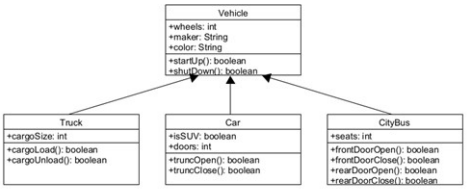
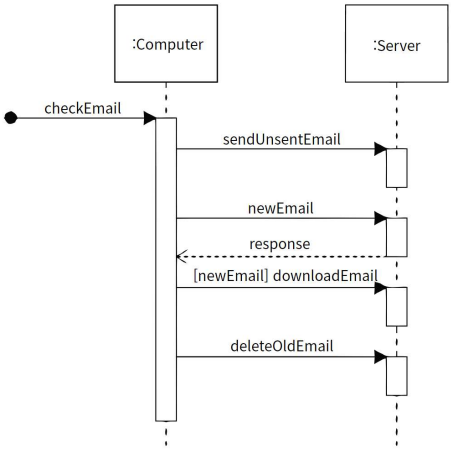
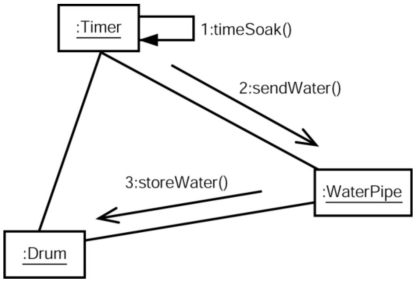
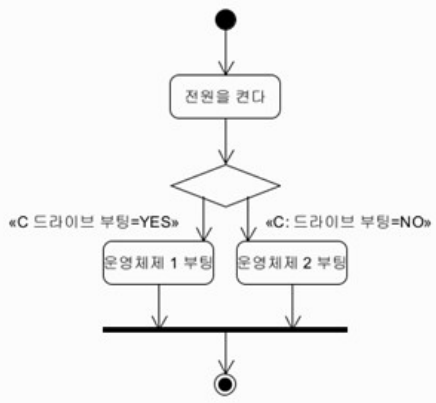
신기술 & 전산관련 용어 정리

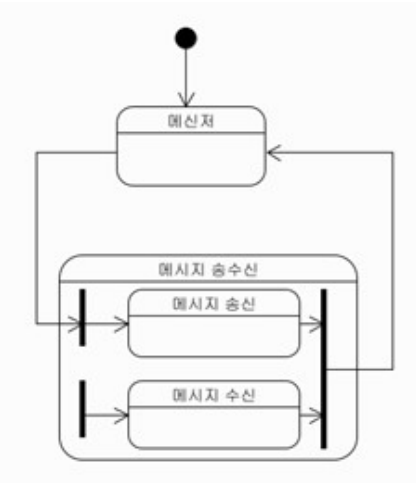
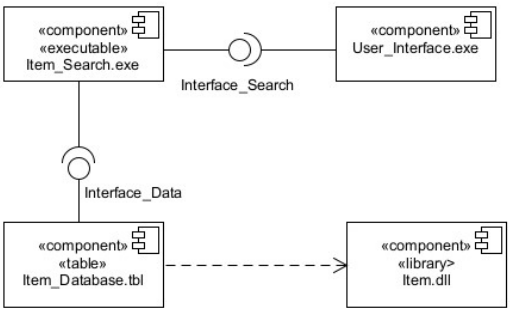
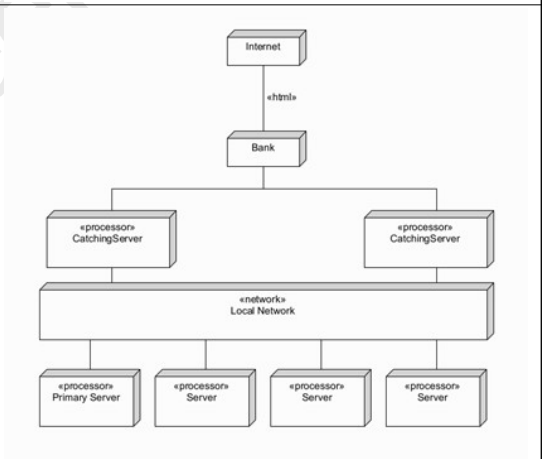
용어	정의	용어	정의
LLM	대량의 텍스트 데이터를 학습하여 자연어 처리를 수행하는 인공지능 모델.	OCR	이미지에서 텍스트를 인식하는 기술
VPN	공용 네트워크를 통해 안전하게 연결하기 위한 가상 사설망.	인덱스	검색 연산의 최적화를 위해 킷값과 포인터의 쌍으로 구성되는 데이터 구조
CI/CD	지속적인 통합 및 배포를 통해 소프트웨어 개발 주기를 단축하는 방법론.	파티셔닝	테이블을 작은 논리적인 단위인 파티션으로 나누는 것
NLP	자연어를 처리하고 이해하는 인공지능 기술.	샤딩	데이터베이스를 수평적으로 분할하여 데이터의 부분 집합을 포함하게 하는 것
RPA	반복적인 작업을 자동화하는 소프트웨어.	DTO	프로세스 사이에서 데이터를 전송하는 객체
NoSQL	비관계형 데이터베이스 시스템	DAO	특정 타입의 데이터베이스의 추상 인터페이스를 제공하는 객체
엣지 컴퓨팅	데이터 소스 근처에서 처리를 수행하는 분산 컴퓨팅	IPC	프로세스 간 통신 (세마포어, 공유메모리, 소켓, 파이프)
프록시 서버	클라이언트와 다른 서버 사이에서 중계 역할을 하는 서버	유니캐스트	단 하나의 수신자에게 1:1 정보 전송
SSO	하나의 인증으로 여러 서비스에 접근할 수 있게 하는 인증 시스템	멀티캐스트	같은 내용의 데이터를 여러명의 특정한 일부 그룹 수신자에게 전송
ORM	객체 지향 프로그래밍 언어와 관계형 데이터베이스 사이의 데이터 변환 기술	브로드캐스트	하나의 송신자가 같은 서브네트워크 모든 수신자에게 전송
VLAN	물리적 위치와 무관하게 논리적으로 분할된 네트워크	DNS	도메인 이름을 IP 주소와 맵핑한 시스템
포트	네트워크 통신에서 특정 프로세스를 식별하는 논리적 단위	형상관리	소프트웨어 변경 사항을 관리하기 위해 개발된 활동
스위치	네트워크 내에서 데이터 패킷을 전달하는 장치	GIT	분산 버전관리 시스템
라우터	네트워크 간 데이터 패킷을 전달하는 장치	SVN	아파치에서 발표한 버전관리 도구
패리티 비트	패리티 비트는 데이터 전송 중 오류 검출을 위해 사용되는 단순한 오류 검출 코드입니다. 주로 비트 단위의 데이터 전송에서 활용.	해밍 코드	해밍 코드는 데이터 전송에서 발생할 수 있는 단일 비트 오류를 검출하고 수정할 수 있는 오류 검출 및 정정 코드 (원리) 코드워드 생성: 원본 데이터 비트에 추가적인 패리티 비트를 삽입하여 코드워드를 생성합니다.

용어	정의	용어	정의
RAID	RAID는 여러 개의 하드 디스크를 묶어 데이터 중복과 성능 향상을 동시에 달성하기 위한 기술	RAID 0	데이터 스트라이핑을 통해 성능 향상을 꾀하지만, 데이터 보호 기능이 없음.
RAID 1	데이터 미러링을 통해 데이터 복제 및 보호를 제공함. (2개를 쓰지만, 1개만큼 용량만 씀)	RAID 5	데이터와 패리티 정보를 블록 레벨에서 분산 저장하여 데이터 보호와 읽기 성능 향상을 동시에 제공 (HDD 3개 이상일 때, 패리티영역 존재)
RAID 6	RAID 5와 유사하나, 이중 패리티를 사용하여 두 개의 디스크 장애까지 견딜 수 있음. (HDD 4개 이상, 패리티 영역, 2개의 하드 장애까지 견딜 수 있음)	RAID 10	RAID 0과 RAID 1을 결합하여 스트라이핑과 미러링을 동시에 제공함. (HDD 4개 이상 적용 가능. 장착된 HDD의 절반 만큼으로 용량 적용)
CSMA/CA	네트워크 환경에서 충돌을 피하기 위한 미디어 액세스 제어 방법으로 주로 무선네트워크에서 사용	인터럽트 Interrupt	프로그램 실행 중 오류 등 예기치 못한 상황에서 실행 중인 작업을 중단하고 예기치 못한 상황을 처리한 이후에 다시 작업 중인 것으로 돌아가는 것. (예를 들어 프로그램 실행 중 프린터 하드웨어를 실행하면 프린터를 즉시 실행시킨다.)
테스트 오라클	테스트시 기대되는 출력이나 동작을 정의하고, 테스트 실행결과를 이를 기준으로 비교하는 기법	회선 교환	통신 방법으로, 데이터 전송 전에 송수신 측 사이에서 물리적 통신 경로를 설정하고 설정된 통신 경로에 따라 연속 스트림 데이터로 전송
가상 메모리	운영체제의 메모리 할당 방법으로 주기억장치보다 프로세스의 요청 메모리가 클 때 보조기억장치의 일부를 사용하는 저장장치	데브 옵스 (DevOps)	개발과 운영의 합성어로, 소프트웨어와 운영을 통합하여 효율성/협력/안전성을 높이는 방법론
가상회선 방식	패킷 스위칭 네트워크에서 사용되며, 데이터 전송을 위해 가상의 경로를 설정하고 데이터를 전송	데이터그램 패킷교환	데이터를 전송하기 전에 논리적 연결이 설정되지 않고, 패킷이 독립적으로 전송되는 방식
UML	현실 세계를 추상화하거나 모델링하여 일반화하는 방법에 사용되는 모델링 언어	SQL Injection	SQL 쿼리를 재구성하여 파라미터로 넣어 데이터베이스를 조작하는 해킹 기법
SDLC	소프트웨어의 개발 타당성 검토부터 폐기까지의 전 과정을 생명 주기로 모형화하여 틀을 제공하는 모델	RARP	MAC주소를 IP주소로 변환하기 위해 사용
ARP	IP주소를 MAC주소로 변환하기 위해 사용	트리거	데이터베이스 테이블에 수정/삭제/삽입 등 이벤트 발생시 자동으로 DBMS에서 실행되도록 정의된 프로그램
허니팟 honey pot	침입자를 속이는 침입탐지기법으로, 공격을 당하는 것처럼 보여 크래커를 추적한다.	OLAP	대규모 데이터 세트를 다차원적으로 분석하고 집계하는 기능을 제공하는 시스템이나 프로그램
TLS	인터넷에서 정보를 암호화하여 송수신하는 프로토콜이다.	맵리듀스	구글에서 빅데이터를 처리하기 위해서 만든 소프트웨어 프레임워크로, 분산처리를 목적으로 한다.
패키지	여러 특정 기능과 관련된 모듈들을 하나의 상위로 묶은 것. 혹은 폴더.	솔트 (salt)	암호화 혹은 단방향 해시함수에서 추가로 입력하여 사용되는 임의의 문자열
교착상태 DeadLock	두 개 이상의 프로세스가 서로의 자원을 요구하며 무한정 대기하는 현상	리팩토링	기능은 수정하지 않지만, 복잡한 소스코드를 효율적으로 재정리하여 가독성/유지보수성을 올리는 행위

시험에 나올만한 다이어그램

❍ 「소프트웨어 요구사항」의 다이어그램의 종류, 사례 도표를 제공해드립니다. (1~2문제 정도 밖에 안 나옵니다.)

이름	영문 이름 (~diagram)	정의와 설명	다이어그램 예제
유스케이스	UseCase	사용자가 시스템과 상호작용하는 것을 작성	
클래스	Class	프로그램의 클래스의 관계를 도식화하여 작성	
순차	Sequence	시간 순서대로 배열된 프로세스나 객체의 상호 작용을 도식화하여 작성	
통신	Communication	객체나 부품 간의 상호작용을 시퀀스 메시지로 모델링한 것	
활동	Activity	선택, 반복, 동시성을 지원하는 단계별 활동이나 작업의 흐름을 도표로 작성	

상태	State	시스템이나 소프트웨어가 가질 수 있는 여러가지 상태와 전이관계를 도표로 작성	
컴포넌트	Component	시스템이나 소프트웨어 내부의 구성 요소를 서로 연결하여 더 큰 시스템을 형성하는 방법을 묘사	
배치	Deployment	하드웨어와 소프트웨어 간의 물리적인 관계를 표시. 각 요소를 Node(노드)라고 부르며 연결선을 커넥션이라고 한다.	
패키지	Package	전체 시스템의 구성 요소를 패키지라는 단위로 그룹화하여 작성