

2회 집단지성 노란색(필수암기) / 파란색(수제비쌤들이 말한거) / 초록색(공부하는사람들이 말한거) / 회색(키워드 관련 내용)	
#SDLC(소프트웨어 생명주기) [폭포나반]	시스템의 요구분석부터 유지보수까지 전 과정을 모델링 한 것
폭포수 모델 (Waterfall Model)	가장 오래되고 가장 폭넓게 사용된 <b>고전적 생명 주기 모형</b> , 한 단계가 끝나야만 다음 단계로 넘어가는 <b>선형 순차적 모형</b> , 단계별 정의 및 산출물이 정확, 개발 중간에 <b>요구사항의 변경이 용이하지 않음</b> / 절차 : 타당성검토 → 계획 → 요구 분석 → 설계 → 구현(코딩) → 테스트(검사) → 유지보수 [분설구테유]
프로토타이핑 (Prototyping Model, 원형 모형)	<b>견본(시제)품을 만들어 피드백을 반영해 최종 결과물을 예측하는 모형</b> , 인터페이스 중점을 두어 개발, 개발 중간에 요구사항의 변경이 용이
나선형 모델 (Spiral Model, 점진적 모형)	폭포수 모형과 프로토타입 모형의 장점에 <b>위험 분석 기능</b> 을 추가한 모형, <b>점진적 개발 과정</b> 반복으로 요구사항 추가 가능, 정밀하고 유지보수 과정 필요 없음 절차: 계획 및 정의 → 위험 분석 → 개발 → 고객 평가 [계위개고]
반복적 모델 (Iteration Model)	병렬적으로 개발 후 통합하거나, 반복적으로 개발해 점차 완성시켜나가는 모델
# 소프트웨어 개발 방법론 [구정객컴애제]	소프트웨어의 개발 시작부터 전 개발 과정을 형상화한 방법론
구조적 방법론	전체 시스템을 나눠 개발하고 통합하는 분할-정복 방식의 방법론 / 나찌-슈나이더만 차트 사용
정보공학 방법론	정보 시스템 개발에 필요한 절차를 체계화한 방법론
객체지향 방법론	객체라는 단위로 시스템을 설계하는 방법론
컴포넌트 기반 방법론	컴포넌트를 조립해 작성하는 방법론
애자일 방법론(Agile Model)	<b>절차보다 사람이 중심이 되어</b> 변화에 유연하고 신속하게 효율적으로 개발 할 수 있는 <b>신속 적응적 경량 개발 방법론</b> 워터폴에 대비되는 방법론으로 최근 회사에서 각광받는 방법론 / 종류 : <b>XP(eXtreme Programming)</b> , <b>스크럼(Scrum)</b> , <b>칸반(Kanban)</b> , <b>크리스탈(Crystal)</b> , <b>린(LEAN)</b>
제품 계열 방법론	제품에 적용할 공통 기능을 정의하여 개발하는 방법론 (임베디드 소프트웨어 작성에 유용)
# XP(eXtreme Programming)	1~3주의 반복(Iteration) 주기를 갖는 애자일 방법론
▶ XP의 5가지 가치 [용단의 피존]	
용기(Courage)	
단순성(Simplicity)	
의사소통(Communication)	
피드백 (Feedback)	
존중(Respect)	
▶ XP의 12가지 기본원리	
짝 프로그래밍 (Pair Programming)	다른 사람과 페어로 개발하여 공동 책임을 지님
공동 코드 소유 (Collective Ownership)	시스템에 있는 코드는 누구나 언제든지 수정 가능
지속적인 통합 (CI; Continuos Integration)	여러 번 소프트웨어를 통합하고 빌드해야 함
계획 세우기 (Planning Process)	고객이 원하는 가치를 정의하고, 개발에 필요한 건 무엇이며, 어떤 곳에서 지연이 될 수 있는지 알려줘야 함
작은 릴리즈 (Small Release)	작은 시스템을 먼저 만들고, 짧은 단위로 업데이트
메타포어 (Metaphor)	공통 이름 체계를 통해 의사소통을 원활히
간단한 디자인 (Simple Design)	요구사항에 적합한 단순한 시스템을 설계
테스트 기반 개발 (TDD; Test Drive Develop)	테스트를 먼저 수행하고, 통과할 수 있는 코드를 작성
리팩토링 (Refactoring)	기능을 바꾸지 않으면서 중복제거, 단순화 등을 위해 코드를 재구성
40시간 작업 (40-Hour Work)	피곤으로 인한 실수가 없도록 주 40시간만 일하자
고객 상주 (On Site Customer)	개발자들의 질문에 즉각 대답해줄 수 있는 고객이 풀타임 상주해야 함
코드 표준 (Coding Standard)	코딩 표준을 두고 효과적으로 개발
# 스크럼 (Scrum)	매일 정해진 시간, 장소에 짧은 시간의 개발을 하는 팀을 위한 프로젝트 중심론
백로그	제품과 프로젝트에 대한 요구사항
스프린트	2~4주의 짧은 기간동안 반복적 수행으로 개발 품질 향상
스크럼 미팅	매일 15분 정도 미팅으로 to-do-list 수립
스크럼 마스터	프로젝트의 리더, 스크럼 수행시 문제를 인지 및 해결 하는 사람
스프린트 회고	스프린트 주기를 되돌아보며 정해놓은 규칙 준수 여부, 개선점 등을 확인 및 기록
번다운 차트	남아있는 백로그 대비 시간을 그래픽적으로 표현한 차트

# 디자인 패턴 [생구행]	
▶ 생성패턴 (Creational)    [생 빌프로 팩업싱] or [생싱추백 빌프]	
빌더(Builder)	작게 분리된 인스턴스를 건축 하듯이 조합하여 객체를 생성 , 객체의 생성 과정과 표현 방법을 분리 → 동일 객체 생성에서도 서로 다른 결과를 만들어 낼 수 있음
프로토타입(Prototype)	원본 객체를 복제하는 방법으로 객체를 생성
팩토리메서드(Factory Method)	생성할 객체의 클래스를 국한하지 않고 객체를 생성, 상위 클래스에서 인터페이스만 정의, 실제 생성은 서브 클래스가 담당
추상팩토리(Abstract Factory)	인터페이스를 통해 서로 연관, 의존하는 객체들의 그룹으로 생성하여 추상적으로 표현, 연관된 서브 클래스를 묶어 한 번에 교체하는 것이 가능
싱글톤(Singleton)	전역 변수 사용x, 한 클래스에 한 객체만 존재, 생성된 객체는 어디서든지 참조 가능, 여러 프로세스가 동시에 참조할 수는 없음
▶ 구조(Structural) 패턴    [구 브데 퍼플 프록 컴어] or [브프! 퍼플 컴 데어?] or [ABCDFFP]	
브리지(Bridge)	구현부에서 추상층을 분리 → 서로 독립적으로 확장할 수 있도록 구성한 패턴, 기능과 구현을 두 개의 별도 클래스로 구현
프록시(Proxy)	실제 객체에 대한 대리 객체를 사용 → 메모리 용량↓, 정보은닉
퍼사드(Facade)	더 상위에 인터페이스를 구성함으로써 서브 클래스들의 기능을 간편하게 사용할 수 있도록 하는 패턴 서브 클래스들 사이의 통합 인터페이스를 제공하는 Wrapper 객체가 필요
플라이웨이트(Flyweight)	모두가 갖는 본질적 요소를 클래스화하여 공유 → 메모리 절약, 클래스의 경량화
컴포지트(Composite)	복합 객체와 단일 객체를 구분 없이 다루고자 할 때 사용, 복합 객체 안에 복합 객체가 포함되는 구조를 구현 가능
데코레이터(Decorator)	객체 간 결합을 통해 능동적으로 기능들을 확장할 수 있는 패턴 임의의 객체에 부가적 기능을 추가하기 위해 다른 객체들을 덧붙이는 방식으로 구현
어댑터(Adapter)	기존에 생성된 클래스를 재사용할 수 있도록 중간에서 맞춰주는 인터페이스 패턴/ 클래스 패턴, 인스턴스 패턴 두 가지 형태로 사용
▶ 행위(Behavioral) 패턴    [행미인이 템옵 스테 비커 스트 메체] or [행메케인 반중책 상옵템 전방]	
중재자(Mediator)	중재자를 만들어 중재자에게 모든 것을 요구하여 통신 빈도수를 낮추는 패턴 수많은 객체들 간 복잡한 상호작용을 캡슐화 → 객체로 정의하는 패턴, 객체 사이의 의존성 줄여 결합도를 감소시킬 수 있음
인터프리터(Interpreter)	언어의 다양한 해석을 맞는 클래스를 각각 작성하여 여러 형태의 언어 구문을 해석할 수 있게 만드는 패턴
이터레이터(Iterator)	컬렉션 구현 방법을 노출하지 않으면서 그 집합체 안에 들어있는 모든 항목에 접근할 수 있는 방법을 제공하는 패턴
템플릿메소드(Template Method)	작업 처리 일부분을 서브 클래스로 캡슐화해 전체 수행 구조는 바꾸지 않으면서 특정 단계 수행 내역을 바꾸는 패턴
옵저버(Observer)	한 객체의 상태가 바뀌면 그 객체에 의존하는 다른 객체에 연락이 가고, 자동으로 내용이 갱신되는 방법
상태(State)	객체 상태를 캡슐화하여 클래스화함으로써 그것을 참조하게 하는 방식, 상태에 따라 다르게 처리할 수 있도록 행위 내용 변경 가능
방문자(Visitor)	각 클래스에서 처리 기능을 분리, 별도 클래스 생성 / 클래스의 메서드가 각 클래스를 돌아다니며 특정 작업을 수행하도록 만드는 패턴
커맨드(Command)	요청을 객체 형태로 캡슐화 → 재이용하거나 취소할 수 있도록 요청에 필요 정보를 저장하거나 로그를 남기는 패턴 요청에 사용되는 각종 명령어들을 추상 클래스(인스턴스 생성 불가)와 구체 클래스로 분리, 단순화
전략(Strategy)	알고리즘 정의 후 각각 하나의 클래스로 캡슐화, 필요할 때 서로 교환해서 사용할 수 있게 하는 패턴, 행위 객체를 클래스로 캡슐화 해 동적으로 자유롭게 변환
메멘토(Memento)	객체를 이전 상태로 복구시켜야 하는 경우, '작업취소(undo)' 요청
책임연쇄(Chain of Responsibility)	한 객체가 처리하지 못하면 다음 객체로 넘어가는 형태의 패턴, 요청을 처리할 수 있는 각 객체들이 고리로 묶여 있어 요청이 해결될 때까지 고리를 따라 책임이 넘어감
# OSI 7계층 [물데네전세표응, 아파서 티내다 피나다]	
응용 (Application)	사용자 또는 애플리케이션이 네트워크에 접근할 수 있도록 도와주는 인터페이스를 담당하는 계층 / HTTP, FTP [데이터]
표현 (Presentation)	데이터의 변환, 압축, 암호화를 담당하는 계층 / JPEG, MPEG [데이터]
세션 (Session)	응용 프로그램 간의 연결을 성립하게 하고 연결이 안정되게 유지관리 하며, 작업 완료 후 연결을 끊는 역할을 담당하는 계층 / SSH, TLS [데이터]
전송 (Transport)	단말기 사이(종단과 종단 사이)에서 오류 수정과 흐름제어 혼잡제어를 수행하여 신뢰성 있고 세그먼트를 전달하는 계층 / TCP, UDP [세그먼트]
네트워크 (Network)	패킷이 최종 목적지에 도달하도록 경로를 정하여 최적으로 데이터를 전송하는 계층 / IP, ICMP[패킷]
데이터링크 (DataLink)	노드와 노드 사이의 데이터를 전송하며, 상위의 계층이 물리 계층을 정상적인 상태로 인식할 수 있게 도와주는 계층 / 이더넷[프레임]
물리 (Physical)	두 장비 간의 전송을 위한 연결이나 전달 등의 인터페이스의 기계적, 전기적, 절차적 특성을 정의하며 비트를 물리적인 매체를 통해 전송하는 계층 / RS-232C[비트]
# 트랜잭션 특성 [ACID ]	
원자성 (Atomicity)	연산 전체가 성공 또는 실패(All or Nothing) / 하나라도 실패할 경우 전체가 취소되어야하는 특성
일관성 (Consistency)	트랜잭션이 실행 성공 후 일관된 데이터베이스 상태를 보존해야 하는 특성
격리성 (Isolation)	트랜잭션 실행 중 생성하는 연산의 중간 결과를 다른 트랜잭션이 접근 불가한 특성
지속성/영속성 ( Durability)	성공이 완료된 트랜잭션의 결과는 계속적으로 데이터베이스에 저장하는 특성

# 요구공학(Requirements Engineering)	요구공학은 사용자의 요구가 반영된 시스템을 개발하기 위하여 사용자 요구사항에 대한 도출, 분석, 명세, 확인 및 검증하는 구조화된 활동, 요구공학 프로세스 : 요구공학 개발 단계 + 요구공학 관리단계
▶ 요구공학 개발 단계(CMM Level 3) [도분명확]	
도출	이해관계자 식별, 고객 분석
분석	분개할협분 분류→개념 모델링 생성→할당→협상→분석
명세	정형화된 형태로 명세 작성
확인	요구사항 이해를 확인하고 문서가 완전한지 검증
▶ 요구공학 관리 단계(CMM Level 2) [협기변화]	
협상	구현 가능한 기능 협상
기준선 설정	기준선(베이스라인) 설정
변경관리	형상통제 위원회를 운영하여 변경 관리
확인 및 검증	요구사항에 부합하는지 확인
#UML(Unified Modeling Language)	객체지향 소프트웨어 개발 과정에서 산출물을 명세화, 시각화, 문서화할 때 사용되는 모델링 기술과 방법론을 통합해서 만든 표준화된 범용 모델링 언어
# UML 다이어그램 종류	
▶ 구조적 [정적-클객컴배복패]	
클래스 다이어그램	클래스의 속성 및 연산과 클래스 간의 정적인 관계를 표현
객체 다이어그램	인스턴스를 특정 시점의 객체와 객체 사이의 관계로 표현
컴포넌트 다이어그램	시스템을 구성하는 물리적 컴포넌트와 그들 사이의 의존 관계 표현
배치 다이어그램	컴포넌트 사이의 종속성 표현, 물리적 요소들의 위치 표현
복합체 구조 다이어그램	클래스나 컴포넌트가 복합 구조를 갖는 경우, 그 내부 구조 표현
▶ 행위적 [동적-유시커상활타]	
유스케이스 다이어그램	시스템이 제공하는 기능과 관련 외부 요소를 사용자 관점에서 표현
시퀀스 다이어그램	객체 간 동적 상호작용을 시간적 개념 중심으로 메시지 흐름으로 표현
커뮤니케이션 다이어그램	동작에 참여하는 객체들이 주고받는 메시지와 객체 간 연관 표현
상태 다이어그램	하나의 객체가 자신이 속한 클래스의 상태 변화 혹은 다른 객체와의 상호작용에 따라 어떻게 변화하는지 표현
활동 다이어그램	시스템이 어떤 기능을 수행하는지, 객체의 처리 로직이나 조건에 따른 처리 흐름을 순서대로 표현
타이밍 다이어그램	객체 상태 변화와 시간 제약을 명시적으로 표현
# UML 관계 [연집포일의실]	
연관 관계(Association Relationship)	2개 이상의 사물이 서로 관련 되어 있는 관계, 사물사이를 실선(-)으로 표현 / 방향성은 화살표(→)로 표현 , 양방향일 경우 화살표 생략하고 실선으로만 연결
집합 관계(Aggregation Relationship)	하나의 사물이 다른 사물에 포함 되어 있는 관계, 포함하는 쪽과 포함되는 쪽은 서로 독립적, 포함되는 쪽에서 포함하는 쪽으로 속이 빈 마름모 화살표(-◇)로 연결하여 표현
포함 관계(Composition Relationship)	포함하는 사물의 변화가 포함되는 사물에게 영향을 미치는 관계, 포함하는 쪽과 포함되는 쪽은 서로 독립이 될 수 없고 생명주기를 함께 함, 포함되는 쪽에서 포함하는 쪽으로 속이 채워진 마름모(-◆)
일반화 관계(Generalization Relationship)	하나의 사물이 다른 사물에 비해 더 일반적인지 구체적인지 표현, 상위(일반적인 개념) 하위(구체적인 개념), 하위 사물에서 상위 사물인 쪽으로 속이 빈 화살표(-▷) 연결
의존관계(Dependency Relationship)	사물 사이에 서로 연관은 있으나 필요에 의해 서로에게 영향을 주는 짧은 시간 동안만 연관을 유지 하는 관계, 소유 관계는 아니지만 사물의 변화가 다른 사물에도 영향을 미침, 영향을 주는 사물(이용자)이 영향을 받는 사물(제공자)쪽으로 점선 화살표(-->) 연결 점선 화살표로 연결
실체화 관계(Realization Relationship)	사물이 할 수 있거나 해야 하는 기능(행위,인터페이스)로 서로를 그룹화 할 수 있는 관계 사물에서 기능쪽으로 속이 빈 점선 화살표(--▷)로 연결

# 관계 대수 연산자	일반집합 연산자 / 순수 관계 연산자
▶ 일반 집합 연산자 [합교차카]	
합집합(Union)	$R \cup S$ (전체)
교집합(Intersection)	$R \cap S$ (공통)
차집합(Difference)	$R - S$ (r에만 존재하고 s에는 없음)
카티션 프로덕트(CARTESIAN Product)	$R \times S$ (r과 s에 속한 모든 튜플 연결)
▶ 순수 관계 연산자 [셀프조디]	
선택트(Select)	$\sigma$ 조건 (R) (R에서 조건을 만족하는 튜플 반환) $\Rightarrow$ 가로
프로젝트(Project)	$\pi$ 속성리스트(R) (R에서 주어진 속성들로만 구성된 튜플 반환) $\Rightarrow$ 세로
조인(Join)	$R \bowtie S$ (공통 속성으로 R과 S의 튜플 연결하여 반환)
디비전(Division)	$R \div S$ (S의 모든 튜플과 관련된 R의 튜플 반환)
# 정규화 [원본이 결다조]	데이터의 중복성을 제거하여 이상 현상을 방지하고, 데이터의 일관성과 정확성을 유지하기 위한 과정
1NF (1차 정규화)	도메인이 원자값
2NF (2차 정규화)	부분 함수 종속 제거
3NF (3차 정규화)	이행 함수 종속 제거
BCNF (보이스-코드 정규화)	결정자 (a $\rightarrow$ b 에서 a) 가 후보키가 아닌 것 제거
4NF (4차 정규화)	다치 종속 제거
5NF (5차 정규화)	조인 종속 제거
# EAI	기업에서 운영되는 서로 다른 플랫폼 및 애플리케이션 간 연계를 돕는 솔루션
▶ EAI 구성 요소	
어댑터(Adapter)	다양한 패키지 애플리케이션 및 기업에서 자체적으로 개발한 애플리케이션을 연결하는 EAI의 핵심 장치로 데이터 입출력 도구
브로커(Broker)	시스템 상호 간 데이터가 전송될 때, 데이터 포맷과 코드를 변환하는 솔루션
메시지 큐(Message Queue)	비동기 메시지를 사용하는 다른 응용프로그램 사이에서 데이터를 송수신하는 기술
▶ EAI 유형 [포허메하]	
point to point(포인트투포인트)	1:1 단순 통합 방식
hub&spoke(허브앤스포크)	단일한 접점의 허브 시스템을 통하여 데이터 전송, 중앙 집중 방식
message bus(메시지 버스)	앱 사이 미들웨어를 두어 연계하는 미들웨어 통합 방식(대용량o)
hybrid(하이브리드)	그룹 내는 허브 앤 스포크, 그룹 간에는 메시지 버스 방식으로 통합
# 웹 서비스	
UDDI	WSDL을 기록하고 보관하기 위한 저장소
WSDL	웹 서비스의 상세 내용을 기술된 XML 형태로 기술한 언어
SOAP	HTTP, HTTPS, SMTP 등을 사용하여 XML 기반의 메시지를 네트워크 상태에서 교환하는 프로토콜
# 인터페이스 관련	
JSON	키-값' 쌍으로 이루어진 데이터를 전달하기 위해, 인간이 읽을 수 있는 텍스트를 사용한 포맷 (AJAX에서 많이 사용)
AJAX	서버-클라 간 비동기적으로 데이터를 교환하기 위한 기술
XML	HTML의 단점을 보완하여, 특수한 목적을 갖는 마크업 언어
REST	HTTP URI로 자원을 표시하고, HTTP 메서드를 통해 해당 자원에 대한 삽입, 삭제, 갱신 등의 명령을 적용하는 아키텍처
IPSec	네트워크 계층(3계층)에서 사용하는 보안 프로토콜, 네트워크 계층(3계층)에서 사용하는 보안 프로토콜. - AH(인증) 프로토콜: MAC를 통해 인증 제공 - ESP(암호화) 프로토콜 : MAC+암호화를 통해 인증+기밀성 제공 - IKE(키관리) 프로토콜 : Key를 주고받는 알고리즘
SSL/TLS	전송계층(4계층)과 응용계층(7계층) 사이에서 안전한 데이터 전송을 보장하는 보안 프로토콜
S-HTTP	클라/서버 간 메시지를 암호화하는 보안 기술 (HTTP를 사용한 경우에만 가능)

# 추상 클래스 (Abstract Class)	유사 클래스들의 공통된 특징을 정의하고, 하나 이상의 추상 메서드와 일반 필드, 메서드를 포함하는 클래스
# 오버라이딩(Overriding)	부모 클래스로부터 상속받은 메서드 멤버를 재정의 하는 것
# 오버로딩(Overloading)	한 클래스 내에서 메서드를 중복해서 생성하는 것
# 상속(Inheritance)	객체 지향 개념에서 이미 정의되어 있는 상위 클래스의 메소드 등 모든 속성을 하위 클래스가 물려받는 것
# 캡슐화(Encapsulation)	객체의 속성과 행위를 하나로 묶고, 실제 구현 내용 일부를 외부에 감추어 은닉한다.
# 인터페이스	자바에서 다형성을 극대화하기 위한 문법으로, 추상 메서드와 상수만을 멤버로 갖는다 / interface로 정의하고 implements로 가져다 씀
# 병행 제어 미보장 시 문제점 [갱신모연]	갱신손실(Lost Update) / 현황파악오류(Dirty Read) / 모순성(Inconsistency) / 연쇄복귀(Cascading Rollback)
# 회복기법 [회로체크]	
로그기반 회복 기법 (지연 / 즉각 갱신)	- 지연 갱신 회복 기법 : 트랜잭션 완료 전에는 로그에만 기록. 장애 발생 시 로그를 폐기 - 즉각 갱신 회복 기법 : 트랜잭션 갱신 결과를 바로 DB에 반영. 장애 발생 시 로그를 참고하여 되돌림
체크 포인트 회복 기법	체크포인트 이전으로 복원
그림자 페이징 기법	트랜잭션 수행 시 복제본을 생성해 이를 이용해 복구
# 형상관리 [식통감기]	소프트웨어 개발 전체 과정에서 발생하는 모든 항목의 변경 사항 관리 활동
형상식별	형상 관리 대상을 정의, 식별하는 활동
형상통제	형상 항목의 버전 관리를 위해서 변경 여부와 변경 활동을 통제
형상감사	소프트웨어 베이스라인의 무결성 평가
형상기록	소프트웨어 형상 및 변경관리에 대한 각종 수행결과 기록
# 형상관리 도구 유형 [공글분]	
공유 폴더 방식	개발이 완료된 파일을 공유 폴더에 복사 / RCC, SCCS
클라이언트/서버 방식	중앙에 버전관리 시스템이 동작 / CVS, SVN
분산 저장소 방식	로컬 저장소/원격 저장소 분산 / GIT
# 응집도 (낮음-> 높은 순)	[우논시절 통순기] ↑ : 모듈 내부 구성요소간 연관 정도
우연적 응집도(Coincidental Cohesion)	모듈 내부의 각 구성요소들이 연관이 없을 경우
논리적 응집도 (Logical Cohesion)	유사한 성격을 갖거나 특정 형태로 분류되는 처리 요소들이 한 모듈에서 처리
시간적 응집도 (Temporal Cohesion)	특정 시간에 처리되어야 하는 활동들을 한 모듈에서 처리할 경우
절차적 응집도 (Procedural Cohesion)	모듈이 다수의 기능을 가질 때 모듈안의 구성요소들이 그 기능을 순차적 수행
통신적 응집도(Communication Cohesion)	동일한 입력과 출력을 사용하여 다른 기능을 수행하는 활동들이 모여 있음
순차적 응집도(Sequential Cohesion)	모듈 내에서 한 활동으로부터 나온 출력 값을 다른 활동이 사용할 경우
기능적 응집도 (Functional Cohesion)	모듈 내부의 모든 기능이 단일한 목적을 위해 수행되는 경우
# 결합도 (높음 -> 낮은 순)	[내공외제 스자] ↓ : 모듈과 모듈간 연관도 또는 모듈간의 상호의존성
내용 결합도 (Content Coupling)	다른 모듈 내부에 있는 지역 변수나 기능을 다른 모듈에서 사용하는 경우
공통 결합도 (Common Coupling)	모듈 밖에서 선언된 전역 변수를 참조하고 갱신하는 등 상호 작용
외부 결합도 (External Coupling)	두 개의 모듈이 외부에서 도입된 데이터 포맷, 통신 프로토콜, 또는 디바이스 인터페이스를 공유
제어결합도 (Control Coupling)	단순 처리할 대상인 값과 어떻게 처리를 해야 한다는 제어 요소가 전달되는 경우
스탬프 결합도 (Stamp Coupling)	모듈 간의 인터페이스로 배열이나 오브젝트, 스트럭처 등이 전달되는 경우
자료 결합도 (Data Coupling)	모듈 간의 인터페이스로 전달되는 파라미터를 통해서만 모듈 간의 상호작용함
# 정보보안 3요소 (기무가)	
기밀성(Confidentiality)	인가되지 않은 개인 혹은 시스템 접근에 따른 정보 공개 및 노출을 차단하는 특성
무결성(Integrity)	정당한 방법을 따르지 않고서는 데이터가 변경될 수 없으며, 데이터의 정확성 및 완전성과 고의/악의로 변경되거나 훼손 또는 파괴되지 않음을 보장하는 특성
가용성(Availability)	권한을 가진 사용자나 애플리케이션 원하는 서비스를 지속해서 사용할 수 있도록 보장하는 특성



# CMMI (Capability Maturity Model Integration)	시스템과 소프트웨어 개발에 대한 능력 및 성숙도에 대한 평가와 프로세스 개선 활동에 광범위한 적응성을 제공하는 지속적인 품질 개선 모델 국제 품질 인증심사로 정보시스템을 구축하는 기업의 능력수준
# 시큐어코딩(Secure Coding)	소프트웨어 개발 과정에서 개발자의 실수, 논리적 오류 등으로 인해 소스 코드 등에 내포될 수 있는 잠재적인 보안 취약점을 최소화하고, 안전한 소프트웨어를 개발하기 위한 일련의 보안 활동
# 전자서명(Electronic signature)	서명자를 확인하고 서명자가 전자문서에서 서명하였음을 나타내는 데 이용하기 위하여 전자문서에 첨부하거나 논리적으로 결합된 전자적 형태의 정보
# DOS (서비스 거부 공격)	시스템을 악의적으로 공격해 시스템 자원을 부족하게 만드는 공격
SYN 플러딩	3-way-handshake 과정을 의도적으로 중단 (SYN 패킷만 보냄), TCP 프로토콜의 구조적 문제를 이용한 공격
UDP 플러딩	대량의 UDP 패킷을 만들어 임의의 포트 번호로 전송하여 응답 메시지를 생성하게 하여 지속적으로 자원을 고갈시키는 공격
스머프(Smurf) / 스머핑 (smurfing)	출발지 주소를 공격 대상의 IP로 설정, 네트워크 전체에 ICMP Echo 패킷을 직접 브로드캐스팅하여 타겟시스템을 마비
죽음의 핑(PoD,Ping of Death)	큰 사이즈의 패킷을 의도적으로 목표시스템으로 발생시켜 시스템이 서비스를 할 수 없는 상태
랜더택 (Land Attack)	출발지 IP 포트를 목적지 IP 포트와 동일하게 하여 동작 마비
티어 드롭 (Tear Drop)	IP패킷의 재조합 과정에서 잘못된 Fragment Offset 정보로 인해 수신 시스템이 문제를 발생하도록 만드는 DoS공격
봉크(Bonk) / 보잉크(Boink)	시스템의 패킷 재전송과 재조립이 과부하를 유발하게 하는 공격기법 (봉크 : 같은 시퀀스 번호 / 보잉크 : 시퀀스 번호에 빈 공간 생성)
# DDOS(분산 서비스 공격)	여러 대의 공격자를 분산 배치, 동시에 공격
Trinoo	UDP 플러딩 공격 도구
TFN	UDP 플러딩, SYN 플러딩, 스머프 등등 여러 DDos 공격이 가능한 도구
Stacheldraht	DDos의 에이전트 역할을 하는 도구
# DRDOS(분산 반사 서비스 거부 공격)	출발지 IP를 공격대상 IP로 위조하여 다수의 반사서버로 요청 정보를 전송, 반사 서버로부터 다량의 응답을 받아 서비스 거부 되는 공격
# 세션 하이재킹	TCP의 세션 관리 취약점을 이용한 공격기법
# 애플리케이션 공격	
HTTP GET 플러딩	과도한 GET 메시지를 이동하여 웹 서버의 과부하를 유발시키는 공격, 웹 서버 자원 소진시키는 공격
slowloris	HTTP GET 메서드를 사용하여 헤더의 최종 끝을 알리는 개행 문자열을 전송하지 않아 연결상태를 지속시켜 자원을 소진시키는 공격(/r/n/r/n>/r/n)
RUDY	요청 헤더의 Content-length를 비정상적으로 크게 설정하여 메시지 바디부분을 매우 소량으로 보내 계속 연결상태를 유지시키는 공격
Slow Http Read Dos	TCP 윈도우 크기와 데이터 처리율을 감소시킨 상태에서 다수 HTTP 패킷을 지속적으로 전송하여 대상 웹 서버의 연결 상태 장시간 지속, 연결 자원을 소진시키는 서비스 거부 공격
Hulk Dos	공격자가 공격대상의 URL을 계속 변경하면서(=차단 정책 우회) 다량의 GET 요청을 보내는 공격
Hash Dos	웹 서버의 해시 테이블에 해시 충돌을 일으켜 자원을 소모시키는 공격
# 네트워크 공격	
스니핑(Sniffing)	공격 대상에게 직접 공격하지 않고 데이터만 몰래 들여다보는 수동적 공격 기법
네트워크 스캐너 / 스니퍼	네트워크 하드웨어 및 소프트웨어 구성의 취약점을 파악을 위해 공격자가 취약점을 탐색하는 공격 도구
사전 크래킹 =사전 대입 공격 (Dictionary Attak)	ID와 PW가 될 가능성이 있는 단어를 파일로 만들어 놓고 이 파일의 단어를 대입하여 크랙하는 공격기법
무차별 크래킹 =무차별 대입 공격 (Brutal Force Attack)	PW로 사용될 수 있는 문자열의 범위를 정하고, 그 범위 내에서 생성 가능한 모든 패스워드를 생성하여 이를 하나씩 대입, 일치 여부를 확인 (일반적으로 사전대입 공격 실패 후 무차별 공격을 진행함)
패스워드 하이브리드 공격(Hybrid Attack)	사전 대입 공격과 무작위 대입공격을 혼합한 방식, 사전(Dictionary) 파일에 있는 문자열에 문자, 숫자 등을 추가로 무작위 대입하여 패스워드 일치 여부를 확인
레인보우 테이블 공격(Rainbow Table Attack)	패스워드별로 해시 값을 미리 생성하여, 크래킹하고자 하는 해시 값을 테이블에 검색해서 역으로 패스워드를 찾는 공격 기법
스푸핑(Spoofing)	승인받은 사용자인 것처럼 시스템에 접근하거나 네트워크 상에서 허가된 주소로 가장하여 접근 제어를 우회하는 공격 기법
IP 스푸핑	침입자가 인증된 컴퓨터 시스템인것처럼 속여서 타깃 시스템의 정보를 빼내기 위해서 본인의 패킷 헤더를 인증된 호스트의 IP어드레스로 위조하여 타깃에 전송하는 공격기법
ARP 스푸핑	공격자가 특정 호스트의 MAC 주소를 자신의 MAC 주소로 위조한 ARP Reply를 만들어 희생자에게 지속적으로 전송하여 특정 호스트의 MAC 정보를 공격자의 MAC으로 변경
ICMP Redirect	3계층에서 스니핑 시스템을 네트워크에 존재하는 또 다른 라우터라고 알림으로써 패킷의 흐름을 바꾸는 공격기법
트로이 목마	악성 루틴이 숨어있는 프로그램으로 겉보기에는 정상적, 실행하면 악성 코드 실행

# 버퍼 오버플로우 공격	할당된 메모리의 버퍼 크기를 초과하는 데이터를 입력해 프로세스 흐름을 변경시키는 공격 기법
스택 버퍼 오버플로우 공격	스택 영역(지역변수, 매개변수가 저장되는 영역)의 버퍼에 오버플로우를 일으켜서 복귀주소를 바꾸는 공격 기법
힙 버퍼 오버플로우 공격	힙 영역(사용자가 직접 관리 가능한 메모리 영역)의 버퍼에 오버플로우를 일으켜서 데이터를 오염시키는 공격 기법
# 보안 관련 용어	
스피어피싱(Spear Phishing)	발송 메일의 본문 링크나 첨부된 파일을 클릭하도록 유도하여 사용자의 개인정보 탈취하는 공격 기법
스미싱(Smishing)	문자메시지를 이용하여 신뢰할 수 있는 사람 또는 기업이 보낸 것처럼 가장하여 개인 비밀정보를 요구하거나 휴대폰 소액 결제를 유도하는 피싱 공격 (SMS+피싱의 합성어)
큐싱 (Qshing)	QR 코드를 통해 악성 앱을 내려받도록 유도하고 금융 정보등을 빼내는 피싱 공격
봇넷(Botnet)	악성 프로그램에 감염되어 악의적인 의도로 사용될 수 있는 다수의 컴퓨터가 네트워크로 연결된 형태
APT 공격(Advanced Persistent Threat)	특정 대상을 목표로 한 지능적/지속적인 공격, 특정 기업이나 조직 네트워크 대상으로 보안을 위협, 무력화하여 정보를 수집한 다음 파괴
공급망 공격(Supply Chain Attack)	SW개발사의 네트워크에 침투하여 소스코드를 수정하여 악의적인 코드를 삽입하는 공격 기법
제로데이 공격	보안 취약점이 널리 공표되기 전에 해당 취약점을 악용하여 이루어지는 보안 기법 (대응책이 나오기 전에 공격이 이루어짐)
웜	네트워크를 통해 연속적으로 자신을 복사하여 시스템의 부하를 높여 시스템을 다운시키는 바이러스의 일종
악성 봇	스스로 실행 X , 해커의 명령에 의해 원격에서 제어 또는 실행 가능 (DDos 등에 악용)
사이버 킬체인	사이버 공격을 프로세스상으로 분석하여 각 공격단계에서 조직에 가해지는 위협요소들을 파악하여 보안 시스템을 확보
랜섬웨어(Ransomware)	사용자 컴퓨터에 잠입해 파일을 암호화하여 사용자가 열지 못하게 하는 프로그램
이블 트윈 공격	무선 WiFi 피싱 기법, 합법적인 WiFi 제공자처럼 위장해 정보를 탈취하는 무선 네트워크 기법
# 그 외 공격 기법	
XSS (크로스 사이트 스크립트 공격)	검증되지 않은 외부 입력 데이터가 포함된 웹페이지를 열때 부적절한 스크립트가 실행되는 공격
CSRF (사이트 간 요청 위조)	사용자가 자신의 의도와는 무관하게 공격자가 의도한 행위를 특정 웹사이트에 요청하게 하는 기법
SQL Injection	외부에서 SQL 쿼리를 삽입하여 시스템 권한을 획득하거나 데이터를 훔치는 공격기법
레이스 컨디션	공유 자원에 대해 여러개의 프로세스가 동시에 접근하기 위해 경쟁하는 상태
디지털 트윈	물리적인 사물과 컴퓨터에 동일하게 표현되는 가상 모델
스누핑(Snooping)	스니핑과 유사한 단어로, 네트워크상의 정보를 엿탐하여 불법적으로 얻는 행위
Pharming	합법적으로 소유하고 있던 사용자의 도메인을 탈취하여 진짜 사이트로 오인하여 접속시켜 개인정보를 훔치는것
난독화	역공학을 통한 공격을 막기 위해 코드를 식별하기 어렵게 하는 작업
키로거 공격	사용자의 키보드 움직임을 탐지해 정보를 빼가는 것
사회공학	비기술적인 해킹 중 사람과 상호작용을 하여 신뢰를 얻어서 보안을 통과하여 정보를 빼앗는 것
클론 피싱 (Clone Phishing)	링크나 첨부파일이 포함된, 과거 전달된 적법한 이메일에 거의 동일하거나 복제된 이메일을 만드는 데 사용되는 수신자 주소와 내용이 포함되어 있는 경우
웨일링 (Whaling)	여러 피싱 공격이 고위 경영 간부와 비즈니스 내 기타 고위직을 대상으로 이루어짐. 웨일링이라는 용어는 이러한 종류의 공격
워터링홀(Watering Hole)	특정 대상을 타겟으로 하는 사회공학적 해킹 기법으로, 대상이 자주 방문하는 웹 사이트 같은 곳을 미리 감염시킨 후, 대상이 방문하면 공격 대상의 컴퓨터에 추가로 악성코드를 설치하여 공격하는 기법
스턱스넷(stuxnet)	산업자동화제어시스템만을 감염시켜 오작동을 일으키는 웜 바이러스
드라이브 바이 다운로드	웹에 악성 스크립트 설치하고 사용자 동의없이 의도된서버(말웨어)로 연결하는 보안용어
킬 스위치	분실한 정보기기를 원격으로 조작해 개인 데이터를 삭제하고 사용을 막는 일종의 자폭기능
# 접근 통제 방법 [덱맥알백]	
임의적 접근 통제 (DAC : Discretionary Access Control)	시스템 객체에 대한 접근을 사용자/ 그룹의 신분을 기준으로 제한하는 방법 사용자나 그룹이 객체의 소유자라면 다른 주체에 대해 이 객체에 대한 접근 권한을 설정할 수 있음
강제적 접근 통제 (MAC : Mandatory Access Control)	미리 정해진 정책과 보안 등급에 의거하여 주체에게 허용된 접근 권한과 객체에게 부여된 허용 등급을 비교하여 접근을 통제하는 모델 높은 보안을 요구하는 정보는 낮은 보안 수준의 주체가 접근할 수 없으며 소유자라고 할 지라도 정책에 어긋나면 객체에 접근할수 없으므로 강력한 보안을 제공
역할기반 접근 통제 (RBAC : Role Based Access Control)	중앙관리자가 주체와 객체의 상호관계를 통제하며 조직내에서 맡은 역할에 기초하여 자원에 대한 접근허용 여부를 결정하는 방법

# 대칭키 암호 방식 (=비밀키 암호화 / 양방향)		같은 키를 이용하여 메시지를 암호·복호화 하는 것
	DES	1975년 미국 NBS에서 발표한 미국표준(NIST) 블록 암호 알고리즘으로, <b>블록 크기 64bit , 키 길이는 56bit</b> 인 페이스텔 구조, 평문을 64bit로 나눠 각 블록에 치환과 전치를 16Round 반복하여 암호화
	3DES	암호화키 2개를 사용하여 암호화(K1) → 복호화(K2) → 암호화(K1) 순으로 암호화
	AES	2001년 DES의 한계를 느낀 미국 표준 기술 연구소(NIST)에서 발표한 고급 암호화 표준이라 불리는 블록 암호 알고리즘으로 (전사 공격 가능) <b>블록 크기는 128비트</b> 이며, <b>키 길이에 따라 128, 192, 256으로 분류하는 알고리즘</b> / 128bit 평문을 128bit로 암호화
	SEED	1999 한국인터넷 진흥원(KISA)이 개발한 3DES보다 신뢰성이 우수한블록 암호화 알고리즘 , 128bit 평문을 128bit 암호 블록으로 암호화 블록 크기는 128bit이며, 키 길이에 따라 128bit, 256bit로 분류
	ARIA	2004년 국가정보원과 산학연합회가 개발한 블록 암호 알고리즘으로, 학계(Academy), 연구기관(Research Institute), 정부(Agency)의 앞글자를 따서 만들었으며 <b>블록 크기는 128비트</b> 이며, <b>키 길이에 따라 128, 192, 256으로 분류하는 알고리즘 / 공공사용을 목적으로 함</b>
	IDEA	DES를 대체하기 위해 스위스 연방기술 기관에서 개발한 블록 암호화 알고리즘 128bit 의 키를 사용하여 64bit 평문을 8라운드에 거쳐 64bit 암호문을 만듦
	RC4	<b>스트림 암호</b> , TLS,WEP 등에서 사용, Octet(2진부호모음, 8bit) 단위 기반 암호화
	LF5R	시프트 레지스터의 일종으로 레지스터에 입력되는 값이 이전 상태 값들의 선형 함수로 계산되는 구조로 되어있는 <b>스트림</b> 암호화 알고리즘
# 비대칭 키 암호 방식(=공개키 암호화 / 양방향)		메시지를 암호화 하는 키와 복호화 하는 키가 다름
	디피-헬만	최초의 공개 키 알고리즘, 유한 필드 내에서 이산 대수의 계산이 어려운 문제를 기본 원리로 하고있음 공개 키 암호 방식 개념을 이용하여 두 사용자 간의 공통의 암호화 키를 안전하게 공유할 방법 제시, 최초의 비밀키 교환 프로토콜
	RSA	1977년 3명의 MIT 수학교수가 개발, <b>소인수 분해하는 알고리즘</b>
	DSA	<b>이산대수의 어려움을 안정성의 바탕으로 개발</b>
	ECC	RSA암호방식의 대안, 이산대수의 문제에 기초한 공개키 암호화 알고리즘 RSA의 문제점인 속도와 안전성 해결을 위해 타원 기반 구조체의 안정성과 효율성을 기반으로 생성, RSA보다 키의 비트 수를 적게 하면서 동일한 성능 제공
	ElGamal	이산대수의 계산이 어려운 문제를 기본 원리로 하고 있으며, RSA와 유사하게 전자서명과 데이터 암호·복호화에 함께 사용 가능
# 해시 암호 방식 (일반형)		임의의 길이 정보를 입력 받아 고정된 길이의 암호문(해시값)을 출력하는 알고리즘
	MD5	MD4를 개선한 암호화 알고리즘, 프로그램이나 파일의 무결성 검사에 사용 512bit 입력 메시지 블록에 대해 차례로 동작, 128bit의 해시값 생성하는 해시 알고리즘
	SHA	1993년 NSA, 미국 국가 표준으로 지정
	SHA-1	NSA에서 미 정부 표준으로 지정, DSA에서 사용 ,160 비트의 해시값 생성하는 해시 알고리즘
	SHA256/384/512	SHA 알고리즘의 한 종류로서 256비트 해시값을 생성하는 함수 ,AES의 키 길이에 대응하도록 출력 길이를 늘린 해시 알고리즘
	HAS-160	국내 표준 디지털 서명 알고리즘(KCDSA)를 위해 개발된 알고리즘, MD5와 SHA-1의 장점을 취하여 개발
	HAVAL	메시지를 1024bits블록으로 나누고 128,160,192,224,256 비트인 다이제스트를 출력하는 해시 알고리즘
# 라우팅 프로토콜		송수신측 간의 전송 경로 중 최적 패킷 교환 경로를 설정하는 기능
	RIP (Routing Information Protocol)	인접한 라우터와 라우팅 정보를 교환하는 거리벡터라우팅. 소규모 동종 네트워크 안에서 효율적. 최대 15홉
	OSPF (Open Shortest Path First)	홉수 제한 X. 대규모 네트워크에서 많이 사용. 라우팅 정보에 변화가 있을 때 변화 정도만 네트워크 내 모든 라우터에게 알리는 링크상태 라우터
	BGP (Border Gateway Protocol)	AS number가 서로 다른 네트워크 간에 라우팅 정보를 주고받을 때 이용될 수 있는 Exterior Gateway Protocol
# BCP(비즈니스 연속성 계획) 관련 주요 용어		
	BIA(Business Impact Analysis)	장애나 재해로 인해 운영상의 주요 손실을 볼 것을 가정하여 시간 흐름에 따른 영향도 및 손실평가를 조사하는 BCP를 구축하기 위한 비즈니스 영향 분석
	RTO(Recovery Time Object)	업무중단 시점부터 업무가 복구되어 다시 가동될 때까지의 시간, 재해시 복구 목표의 시간의 선정
	RPO(Recovery Point Object)	업무 중단 시점부터 데이터가 복구되어 다시 정상가동될 때 데이터의 손실 허용 시점, 재해시 복구 목표 지점의 선정
	DRP(Disaster Recovery Plan)	재난으로 장기간에 걸쳐 시설의 운영이 불가능한 경우를 대비한 재난 복구 계획
	DRS(Disaster Recovery System)	재해복구계획의 원활한 수행을 지원하기 위하여 평상시에 확보하여 두는 인적, 물적 자원 및 이들에 대한 지속적인 관리체계가 통합된 재해복구센터
# IDS		네트워크에 대한 공격을 실시간으로 탐지하는 시스템
# IPS		네트워크에 대한 공격을 실시간으로 차단 및 능동적으로 대응하는 시스템



#블랙박스 [동경결상 유분페원비]	내부구조나 동작을 모르는 상태에서 검사하는 테스트 방식/ 소프트웨어가 수행할 특정 기능을 알기 위해 각 기능이 완전히 작동되는것을 입증하는 테스트
동등 분할 테스트 = 동치 분할 테스트 (Equivalence Partitioning Testing)	입력 데이터의 영역을 유사한 도메인별로 유효값/무효 값을 그룹핑하여 대푯값 테스트 케이스를 도출하여 테스트하는 기법
경곷값 분석 테스트 = 한곷값 테스트 (Boundary Value Analysis Testing)	등가 분할 후 경계값 부분을 포함하여 테스트 케이스를 설계하여 테스트 하는 기법
결정 테이블 테스트 (Decision Table Testing)	요구 사항의 논리와 발생조건을 테이블 형태로 나열하여, 조건과 행위를 모두 조합하여 테스트 하는 기법
상태 전이 테스트 (State Transition testing)	테스트 대상이나 객체 상태를 구분하고 이벤트에 의해 어느 한 상태에서 다른 상태로 전이되는 경우의 수를 수행하는 테스트 기법
유스케이스 테스트 (Use Case Testing)	시스템이 실제 사용되는 유스케이스로 모델링 되어 있을 때, 프로세스 흐름을 기반으로 테스트 케이스를 명세화해 수행하는 테스트 기법
분류 트리 테스트 (Classification Tree Method )	SW의 일부 또는 전체를 트리 구조로 분석 및 표현하여 테스트 케이스를 설계하여 테스트 하는 기법
페어 와이즈 테스트 (Pairwise Testing)	테스트 데이터 값을 최소한 한 번씩 조합하는 방식
원인결과 그래프 테스트 (CauseEffect Graphing Testing)	그래프를 활용해 입력 데이터 간의 관계 및 출력에 미치는 영향을 분석하여 효용성이 높은 테스트 케이스를 선정하여 테스트 하는 기법
비교 테스트 (Comparison Testing)	여러 버전의 프로그램에 같은 입력값을 넣어 동일한 결과가 나오는지 비교해 보는 테스트 기법
#화이트박스 [구결조 조변다 기제데]	내부구조와 동작을 검사하는 테스트 방식
구문 커버리지 = 문장 커버리지 (Statement Coverage)	프로그램 내의 모든 명령문을 적어도 한 번 수행하는 커버리지
결정 커버리지 = 선택 커버리지 (Decision Coverage) = 분기 커버리지 (Branch Coverage)	결정 포인트 내의 전체 조건식이 적어도 한 번은 참과 거짓의 결과를 수행하는 테스트 커버리지
조건 커버리지 (Condition Coverage)	결정 포인트 내의 각 개별 조건식이 적어도 한 번은 참과 거짓 결과가 되도록 수행하는 테스트 커버리지
조건/결정 커버리지 (Condition/Decision Coverage)	전체 조건뿐 아니라 개별 조건식도 참 한 번, 거짓 한 번 결과가 되도록 수행하는 테스트 커버리지
변경 조건/결정 커버리지 (Modified Condition/Decision Coverage)	개별 조건식이 다른 개별 조건식의 영향을 받지 않고 전체 조건식에 독립적으로 영향을 주도록 함으로써 조건/결정 커버리지를 향상시킨 커버리지
다중 조건 커버리지 (Multiple Condition Coverage)	결정 조건 내 모든 개별 조건식의 모든 가능한 조합을 100% 보장하는 커버리지
기본 경로 커버리지 = 경로 커버리지 (Base Path Coverage)	수행 가능한 모든 경로 테스트하는 기법, 맥케이브 순환 복잡도 기반으로 커버리지 계산
제어 흐름 테스트 (Control Flow Testing)	프로그램 제어 구조를 그래프 형태로 나타내어 내부로직을 테스트하는 기법
데이터 흐름 테스트 (Data Flow Testing)	제어흐름 그래프에 데이터 사용현황을 추가한 그래프를 통해 테스트하는 기법
# 성능 테스트	
부하 테스트 (Load Testing)	부하를 계속 증가시키면서 시스템의 임계점 find
스트레스 테스트 (Stress Testing)	임계점 이상의 부하를 가하여 비정상적인 상황에 처리를 하는 테스트
스파이크 테스트 (Spike Testing)	짧은 시간에 사용자가 몰릴때 시스템 반응 측정 시스템
내구성 테스트 (Endurance Testing)	오랜시간동안 시스템에 높은 부하를 가하여 시스템 반응 테스트
# IPv4	32bit / [유멀브] 유니캐스트, 멀티캐스트, 브로드캐스트
# IPv6	128bit / [유멀애] 유니캐스트, 멀티캐스트, 애니캐스트
# IPv6 특징 [확인실패 플플이단해]	IP주소 확장 / 인증 및 보안 기능 / 실시간 패킷 추적 가능 / Plug & Play 지원 / 이식성 / 단순 헤더
# IPv4 -> IPv6 전환기술 [듀터주]	
듀얼스택 (Dual Stack)	IP 계층에 두 가지(IPv4, IPv6)의 프로토콜 이 모두 탑재되어 있고 통신 상대방에 따라 해당 IP스택을 선택하는 방법
터널링 (Tunneling)	IPv6 망에서 인접한 IPv4 망을 거쳐 다른 IPv6 망으로 통신할 때 IPv4 망에 터널을 만들고 IPv4에서 사용하는 프로토콜로 캡슐화하여 전송 하는 방법
주소변환 (Address Transiation)	IPv4 망과 IPv6 망 사이에 주소변환기(IPv4-IPv6 게이트웨이)를 사용하여 서로 다른 네트워크 상의 패킷을 변환시키는 방법

# TCP	[흐름 제어, 혼잡 제어, 오류 제어, 재조립 / 신뢰성, 연결 지향]
# UDP	비신뢰성, 비연결성
# 선점 스케줄링 알고리즘 [SMMR]	
SRT(Shortest Remaining time first)	가장 짧은 시간이 소요되는 프로세스를 먼저 수행하고, 남은 처리 시간이 더 짧다고 판단되는 프로세스가 준비 큐에 생기면 언제라도 프로세스 점유가 되는 방식
다단계 큐(MultiLevel Queue)	작업들을 여러 종류의 그룹으로 분할, 여러 개의 큐를 이용하여 상위 단계 작업에 의한 하위 단계 작업이 선점당함
다단계 피드백 큐(MultiLevel Fedback Queue)	새로운 프로세스는 높은 우선순위를 가지고 프로세스의 실행시간이 길어질수록 점점 낮은 큐로 이동하고 마지막 단계는 RR방식을 적용하는 방식 ( FCFS + RR)
라운드 로빈(RR, Round Robin)	균등한 CPU 시간을 할당, 프로세스가 할당된 시간 내에 처리완료를 못하면 준비 큐 리스트의 가장 뒤로 보내지고, CPU는 대기중인 다음 프로세스로 넘어감
# 비선점 스케줄링 알고리즘 [우기HFS]	
우선순위	주요/긴급 프로세스에 대한 우선처리, 동일순위는 FIFO, 에이징기법
기한부	
HRN	응답률 높은 것 선택, 기아현상 최소화 기법, 응답률=(대기시간+서비스시간)/서비스시간
FCFS(FIFO)	도착한 순서대로 처리
SJF(Shortest Job First)	프로세스가 도착하는 시점에 따라 그 당시 가장 작은 서비스 시간을 갖는 프로세스가 종료 시 까지 자원을 점유하는 방식
#블랙박스 [동경결상 유분폐원비]	내부구조나 동작을 모르는 상태에서 검사하는 테스트 방식/ 소프트웨어가 수행할 특정 기능을 알기 위해 각 기능이 완전히 작동되는것을 입증하는 테스트
동등 분할 테스트 = 동치 분할 테스트 (Equivalence Partitioning Testing)	입력 데이터의 영역을 유사한 도메인별로 유효값/무효 값을 그룹핑하여 대푯값 테스트 케이스를 도출하여 테스트하는 기법
경계값 분석 테스트 = 한겻값 테스트 (Boundary Value Analysis Testing)	등가 분할 후 경계값 부분을 포함하여 테스트 케이스를 설계하여 테스트 하는 기법
결정 테이블 테스트(Decision Table Testing)	요구 사항의 논리와 발생조건을 테이블 형태로 나열하여, 조건과 행위를 모두 조합하여 테스트 하는 기법
상태 전이 테스트(State Transition testing)	테스트 대상이나 객체 상태를 구분하고 이벤트에 의해 어느 한 상태에서 다른 상태로 전이되는 경우의 수를 수행하는 테스트 기법
유스케이스 테스트(Use Case Testing)	시스템이 실제 사용되는 유스케이스로 모델링 되어 있을 때, 프로세스 흐름을 기반으로 테스트 케이스를 명세화해 수행하는 테스트 기법
분류 트리 테스트 (Classification Tree Method Testing)	SW의 일부 또는 전체를 트리 구조로 분석 및 표현하여 테스트 케이스를 설계하여 테스트 하는 기법
페어 와이즈 테스트(Pairwise Testing)	테스트 데이터 값을 최소한 한 번씩 조합하는 방식
원인결과 그래프 테스트 (CauseEffect Graphing Testing)	그래프를 활용해 입력 데이터 간의 관계 및 출력에 미치는 영향을 분석하여 효용성이 높은 테스트 케이스를 선정하여 테스트 하는 기법
비교 테스트(Comparison Testing)	여러 버전의 프로그램에 같은 입력값을 넣어 동일한 결과가 나오는지 비교해 보는 테스트 기법
#화이트박스 [구결조 조변다 기제데]	내부구조와 동작을 검사하는 테스트 방식
구문 커버리지 = 문장 커버리지 (Statement Coverage)	프로그램 내의 모든 명령문을 적어도 한 번 수행하는 커버리지
결정 커버리지 = 선택 커버리지 (Decision Coverage) = 분기 커버리지(Branch Coverage)	결정 포인트 내의 전체 조건식이 적어도 한 번은 참과 거짓의 결과를 수행하는 테스트 커버리지
조건 커버리지(Condition Coverage)	결정 포인트 내의 각 개별 조건식이 적어도 한 번은 참과 거짓 결과가 되도록 수행하는 테스트 커버리지
조건/결정 커버리지 (Condition/Decision Coverage)	전체 조건뿐 아니라 개별 조건식도 참 한 번, 거짓 한 번 결과가 되도록 수행하는 테스트 커버리지
변경 조건/결정 커버리지 (Modified Condition/Decision Coverage)	개별 조건식이 다른 개별 조건식의 영향을 받지 않고 전체 조건식에 독립적으로 영향을 주도록 함으로써 조건/결정 커버리지를 향상시킨 커버리지
다중 조건 커버리지 (Multiple Condition Coverage)	결정 조건 내 모든 개별 조건식의 모든 가능한 조합을 100% 보장하는 커버리지
기본 경로 커버리지 = 경로 커버리지 (Base Path Coverage)	수행 가능한 모든 경로 테스트하는 기법, 맥케이브 순환 복잡도 기반으로 커버리지 계산
제어 흐름 테스트 (Control Flow Testing)	프로그램 제어 구조를 그래프 형태로 나타내어 내부로직을 테스트하는 기법
데이터 흐름 테스트 (Data Flow Testing)	제어흐름 그래프에 데이터 사용현황을 추가한 그래프를 통해 테스트하는 기법

# 통합 테스트 [단통사인]	
단위 테스트	사용자 요구사항에 대한 <b>단위 모듈, 서브루틴</b> 등을 테스트하는 단계
통합 테스트	단위 테스트를 통과한 모듈 사이의 인터페이스, <b>통합된 컴포넌트 간의 상호작용을 검증</b> 하는 테스트 단계, 인터페이스에 대한 테스트
시스템 테스트	<b>통합된 단위 시스템의 기능이 시스템에서 정상적으로 수행되는지를 검증하는 테스트 단계</b>
인수 테스트	계약상의 <b>요구사항이 만족되었는지 확인</b> 하기 위한 테스트 단계
# 클라우드 컴퓨팅	인터넷 기반 구동 컴퓨팅 기술. 응용프로그램을 필요에 따라 불러 사용하고 데이터를 손쉽게 저장 추출 가능
IaaS (Infrastructure as a Service)	웹상에서 구글, 아마존, MS 등에서 제공하는 환경의 네트워크, 보안, 데이터 저장소 등 서비스를 포함한 다양한 인프라를 임대하여 이용할 수 있는 서비스
PaaS (Platform as a Service)	운영체제가 이미 구성된 상태에서 사용자는 데이터와 애플리케이션만 직접 관리할 수 있는 서비스
SaaS (Software as a Service)	인프라, 운영체제뿐만 아니라 사용가능한 SW까지 갖추어져 웹상 로그인만으로 다양한 SW를 사용한 만큼 비용을 지불해가며 사용할 수 있는 서비스
AlaaS (AI as a Service)	AI 서비스 자체를 API 형태로 클라우드에서 제공하는 서비스를 사용자가 활용하고, 사용한 만큼 지불하는 서비스를 제공하는 플랫폼.
# 블루투스 공격기법 필기기술 연계	
블루버그(BlueBug)	블루투스 <b>장비 사이의 취약한 연결</b> 관리를 악용한 공격
블루스나프(BlueSnarf)	블루투스의 <b>취약점을 활용해 장비의 파일에 접근하는 공격</b> 으로 OPP를 사용하여 정보를 열람
블루재킹(BlueJacking)	블루투스를 이용해 <b>스팸처럼 명함을 익명으로 퍼뜨리는 것</b>
블루프린팅(BluePrinting)	블루투스 <b>공격 장치의 검색 활동</b> 을 의미
# 빅데이터(R)	
# iptables	리눅스 계열의 OS에서 방화벽 관리를 위한 명령어 (보안쌔이 나올거같다고 하는데 뭐가 나올지 모르겠어요 )
# MQTT	IoT 장치, 텔레메트리 장치 등에서 최적화되어 사용할 수 있도록 개발된 프로토콜로, 브로커를 사용한 Publish/Subscribe 방식의 라이트 메시징을 전송하는 프로토콜 저전력 센서, 스위치, 밸브 등의 기기에 대한 표준적 인터넷환경을 지원하고, 프로토콜 리소스 점유 최소화, 한정된 자원 시스템을 지원하는 특징이 있음.
# 피코넷(piconet)	여러개의 독립된 통신장치가 블루투스나 UWB통신기술을 사용하는 무선네트워크 기술
# 서비스 지향 아키텍처 (SOA , Service Oriented Architecture)	* 프로세스 수행을 지원하는 정보 시스템의 구현을 위해 가장 선진화된 소프트웨어 아키텍처로 서비스라고 정의되는 분할된 애플리케이션 조각들을 Loosely-coupled 하게 연결해 하나의 완성된 Application을 구현하기 위한 아키텍처 * 웹 서비스나 REST 서비스로 구현된 소프트웨어 모듈들을 BPEL로 구현한 프로세스이며 오케스트레이션(Orchestration) 함으로써, 프로세스 변경에 따른 정보시스템의 재구성을 신속유연하게 할 수 있는 소프트웨어 아키텍처
# 소프트웨어 정의 데이터센터 (SDDC , Software Defined Data Center )	모든 하드웨어가 가상화되어 가상 자원의 풀을 구성하고 데이터센터 전체를 운영하는 소프트웨어가 필요한 기능 및 규모에 따라 동적 자원을 할당, 관리 하는 역할을 수행하는 데이터 센
# CVE (Common Vulnerabilities and Exposures)	*공개적으로 알려진 컴퓨터 보안 <b>결함</b> 목록, 보안 취약점을 데이터베이스로 관리하기 위한 표기 방법, 소프트웨어의 보안약점을 사전식으로 분류해놓은 목록 * 미국 비 영리회사인 MITRE 사에서 공개적으로 알려진 소프트웨어의 보안 취약점을 표준화한 식별자 목록
# CWE (Common <b>Weakness</b> Enumeration)	소프트웨어 <b>취약점</b> 및 취약점에 대한 범주 시스템으로, 소프트웨어 결함을 이해하고 이러한 결함을 식별 수정 방지하는데 사용할 수 있는 자동화된 도구를 작성하는 시스템
# 온라인 분석처리 (OLAP , Online Analytical Processing)	데이터 웨어하우스나 데이터마트 같은 시스템과 상호 연관되는 정보 시스템, 데이터 웨어하우스의 데이터를 전략적인 정보로 변환시켜서 의사결정을 지원하는 역할
# 자료사전의 기호	
=	정의(is composed of)
+	구성(and, along with)
{ }	반복(iteration)
[ ]	선택(selection)
( )	생략가능(optional)
* *	주석(comment)

# 프로그래밍	
- switch 문에서 break 없이 출제	
- 나머지 연산 문제 응용(약수, 배수, 최대공약수, 최소공배수, PrimeNumber)	[2022년 제2회 기사 실기 출제예상 문제] (6과목) 프로그래밍 활용 - 5 (PrimeNumber)
- 알고리즘(스택, 큐, 리스트, 힙, 정렬, 검색)	
- C언어 문자열 처리 함수 (str계열 함수 : strcmp, strchr, strstr 등)	
- C언어 포인터 문제	
- 자바 디자인 패턴(팩토리 패턴)	
- 파이썬 리스트 함수	
C++ : 2중 for문	
상속 (자바 : 상속관련 abstract 사용)	
배열	
슬라이싱 (python : 3중 슬라이싱)	
# 데이터베이스	
SELECT 명령어	셀프웨구해오 1. SELECT [ALL   DISTINCT] 컬럼명 [AS 별칭] 2. FROM 테이블 3. WHERE 조건 4. GROUP BY 그룹명 5. HAVING 그룹조건 6. ORDER BY 컬럼명 [ASC   DESC];
ratio_to_report	
cube	다차원 집계 산출  SELECT 기숙사, 학년, SUM(점수) FROM 점수 GROUP BY CUBE(기숙사, 학년);
rollup	소계 산출  SELECT 기숙사, 학년, SUM(점수) FROM 점수 GROUP BY ROLLUP(기숙사, 학년);
내부 조인	양 테이블의 공통 컬럼의 값이 같을 때 사용 inner join
외부 조인	- 왼쪽 외부 조인 : 왼쪽 테이블 전체 + 오른쪽 테이블 동일 데이터 LEFT OUTER JOIN - 오른쪽 외부 조인 : 오른쪽 테이블 전체 + 왼쪽 테이블 동일 데이터 RIGHT OUTER JOIN - 완전 외부 조인 : 양쪽 모든 데이터 추출 FULL OUTER JOIN
집계함수	단일값 리턴, "SELECT" 절에 사용 NULL은 없는 데이터로 취급함  1. COUNT : 해당하는 튜플의 수 ex. COUNT(DISTINCT 컬럼) = 중복 제거한 컬럼 수 ex. COUNT(학과) = 해당 학과에 해당하는 튜플의 수 (학생 등) 2. SUM : 합계 3. AVG : 평균 4. MAX : 최댓 5. MIN : 최솟값 6. STDDEV : 표준편차 7. VARIAN : 분산