ICT의 가치를 이끄는 사람들!!

131회

정보관리기술사 기출풀이 2교시

T 02.724.1831 E kpcitpe@kpc.or.kr H www.kpc.or.kr C cafe.naver.com/kpcitpe

국가기술자격 기술사 시험문제

정보처리기술사 제 131 회 제 2 교시

분야	정보처리	종목	정보관리기술사	수험번호		성명	

※ 다음 문제 중 4 문제를 선택하여 설명하시오. (각 25 점)

1. ISP(Information Strategy Planning)와 BPR(Business Process Reengineering)의 개념과 수행절차를 비교 설명하고, 기업에서 이 두가지가 상호 보완적으로 활용하기 좋은 방안을 설명하시오.

2. 데이터 시각화(Data Visualization)와 관련하여 다음을 설명하시오.
 가. 데이터 시각화의 개요
 나. 데이터 시각화의 원리 및 절차
 다. 데이터 시각화 유형
 라. 효과적인 데이터 시각화를 위한 효율화 방안

3. 인공지능의 개발 및 적용과정에서 윤리적으로 다루어져야 할 주요 내용과 인공지능을 효과적으로 관리하고 규제하기 위한 거버넌스 모형에 대하여 설명하시오.

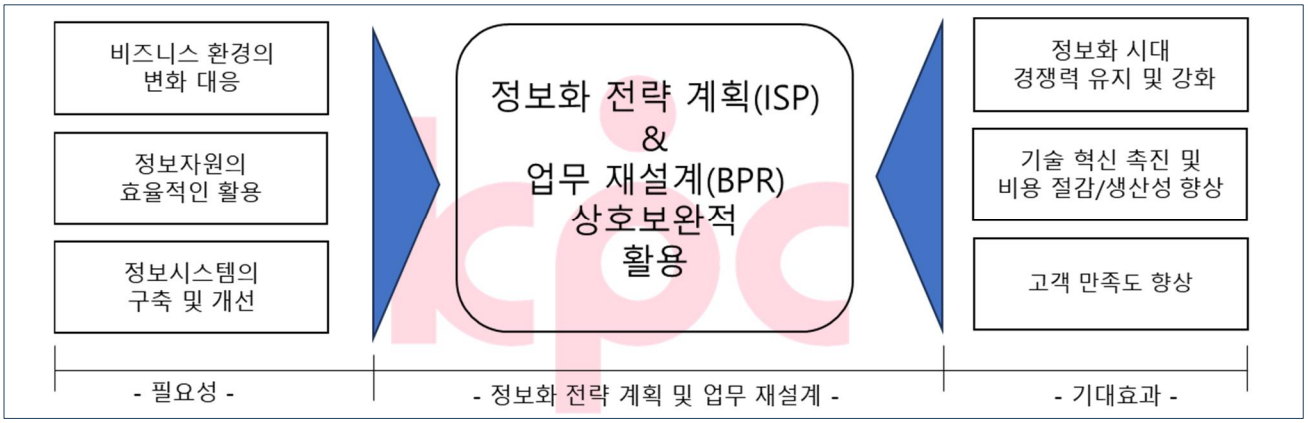
4. 제로 트러스트 보안(Zero Trust Security)모델의 보안원리, 핵심원칙, 적용분야를 트러스트 보안(Trust Security)모델과 비교하여 설명하시오.

5. 소켓(Socket) 통신과 관련하여 다음을 설명하시오.
 가. 소켓 통신 정의
 나. 소켓 통신 방식 개념도 및 유형
 다. TCP 소켓 및 Web 소켓 흐름 설명
 라. 소켓 통신 방식과 HTTP 통신 방식 비교

6. 아키텍처 스타일과 디자인 패턴에 대하여 다음을 설명하시오.
 가. 아키텍처 스타일과 디자인 패턴의 차이점
 나. 대표적인 아키텍처 스타일 3가지
 다. GoF(Gang of Four) 디자인 패턴의 유형을 구분하고, 유형별 대표적인 디자인 패턴 설명

문 제	1. ISP(Information Strategy Planning)와 BPR(Business Process Reengineering)의 개념과 수행절차를 비교 설명하고, 기업에서 이 두가지가 상호 보완적으로 활용하기 좋은 방안을 설명하시오.		
출 제 영 역	IT 경영	난 이 도	★★☆☆☆
출 제 배 경	ISP 와 BPR 의 기본적인 개념 및 수행절차 이해 및 활용 방안에 대한 의견 제시 확인		
출 제 빈 도	123 회 정보관리 ,127 회 정보관리, 129 회 정보관리		
참 고 자 료	- https://m.blog.naver.com/seaofu/130126749642 - https://casit.co.kr/173		
K e y w o r d	- 정보화 전략 계획, 업무프로세스 재설계, Zero-Base, 경영혁신기법, Blue Print(청사진)		
풀 이	류연춘 PPE (130 회 정보관리 / ycel@naver.com)		

1. 정보화 전략 계획 및 경영 혁신, ISP 와 BPR 필요성



- 기업의 경쟁력 강화를 위해 업무 프로세스의 문제점을 도출하고 경영 목표를 달성하기 위한 정보화 전략 수립(ISP) 및 업무 재설계(BPR)의 상호보완적 활용이 필요하며, 일반적으로 BPR 후 ISP 수행함.

1. ISP 와 BPR 의 개념 및 수행절차 비교 설명

가. ISP 와 BPR 의 개념 비교

ISP	BPR
조직의 경영전략 분석결과에 따른 정보화 비전, 목표 및 전략을 기반으로 한 운영/관리 체계와 정보 기술구조 수립 및 이행 계획 수립하는 활동	기업의 비용, 품질, 서비스 등 획기적인 향상을 이룰 수 있도록 프로세스를 근본적으로 제고하여 혁신적으로 재설계하는 경영혁신 활동

- BPR 은 업무혁신 위한 개선 과제를 도출하고 기존 업무를 재설계하는 것이고, ISP 는 업무 혁신 실현을 위해 정보화 비전 및 중장기 정보화 전략 계획을 수립하는 것으로 각각 활동에 대한 목적과 수행 절차 이해 필요.

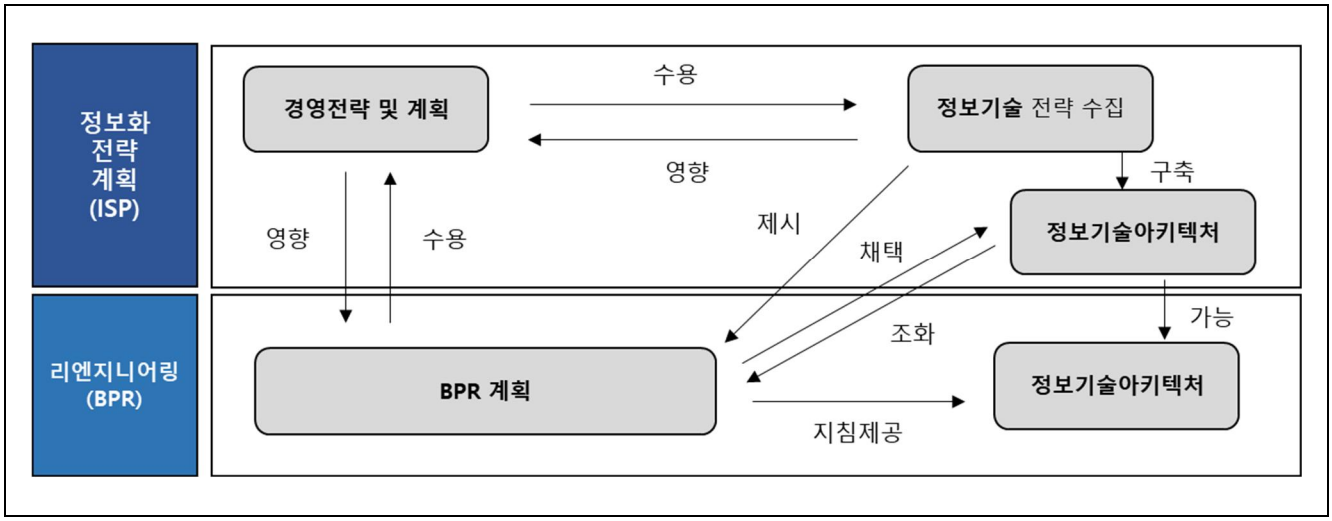
나. ISP 와 BPR 의 수행절차 비교

항목	ISP		BPR	
수행 절차	①환경분석->②현황분석(AS-IS)-> ③미래모델 설계(To-Be)->④통합이행계획		①현재 프로세스 이해->②고객니즈 파악 및 목표 설정->③혁신 프로세스 설계 ->④실행 과제 도출	
상세 수행 절차	①환경분석	- 경영 환경 분석 - 법령·제도 분석 - 정보기술(IT) 환경분석	①현재 프로세스 이해	- 현재 업무프로세스 현상 파악 - 핵심 이슈 도출
	②현황분석 (As-Is)	- 업무현황 분석 - 정보기술(IT)현황 분석 - 벤치마킹/차이분석 - 이슈 통합 및 개선 과제 도출	②고객 니즈 파악 및 목표설정	- 고객 니즈 파악 및 벤치마킹 통한 핵심 포인트 도출 - 목표치 설정/달성 위한핵심 프로세스 및 요소 도출
	③미래모델설계 (To-Be)	- To-Be 개선과제 상세화/ 업무프로세스 설계/ 정보 시스템 구조 설계/데이터 구조 설계/기술 및 보안 구조 설계	③혁신 프로세스 설계	- 업무 혁신 포인트 반영한 프로세스 구축 요건표 작성 - 혁신 프로세스 Mapping
	④통합이행계획	-통합 이행계획 수립 -총구축비 산출 -효과분석	④실행 과제 도출	- 혁신 프로세스 구현, 필요과제 도출 - 실행 과제별 세부추진 일정 수립 - 실제 구현 방안 및 점검

- 기업에서 ISP(정보화 전략 계획)과 BPR(업무재설계)를 상호보완적으로 활용하므로 경영 혁신 및 개선 수행

2. 기업에서 ISP 와 BPR 의 상호 보완적으로 활용하기 위한 방안 설명

가. ISP 와 BPR 의 상호 보완적 활용하기 위한 방안 도식화



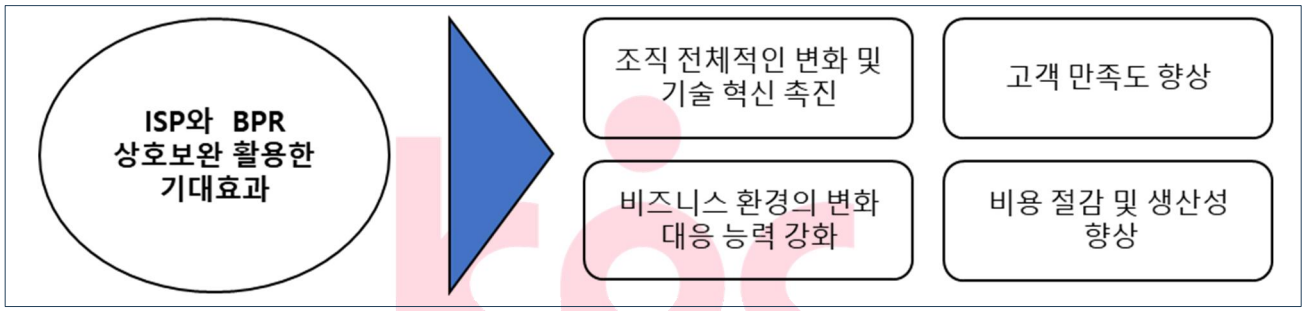
- 체계적인 업무 현황 분석과 혁신적인 중장기적 정보화 전략 계획 수립이 요구되고, ISP(정보화 전략 계획)과 BPR(업무 프로세스 재설계)를 상호 보완적으로 활용하여 경영 목표 달성을 위한 효과를 극대화 가능함.

나. ISP 와 BPR 의 상호 보완적 활용하기 위한 방안 상세 설명

구분	활용방안	설명
경영 혁신 측면	경영 혁신 기반 업무프로세스 재설계	- 경영 혁신을 위한 업무 프로세스 재설계 반영 - 제로 베이스 관점에서 도출한 현행 프로세스 이슈 및 개선사항을 경영 혁신 프로세스에 반영
정보전략계획 측면	정보 전략 계획 기반 업무 프로세스 재설계	- 중장기적 전사 정보 전략 계획에 대한 명확한 이해 기반 선행된 후 업무프로세스 재설계 수행으로 구체적 실행과제 도출
정보기술적용 아키텍처 활용 측면	정보기술적용 아키텍처기반 업무 프로세스 재설계	- 전사의 정보 기술 아키텍처 기반 업무 프로세스 재설계 적용

- 성공적으로 ISP 와 BPR 을 위해서 전사적 관심과 적극적인 참여, 최고 경영층의 확고한 의지가 매우 중요함.

3. 기업에서 ISP 와 BPR 의 상호 보완적으로 활용한 기대효과



- 정보화 비전 달성을 위해 EA(Enterprise Architecture) 기반 ISP 와 BPR 통해 전사의 실행 계획 수립 필요.

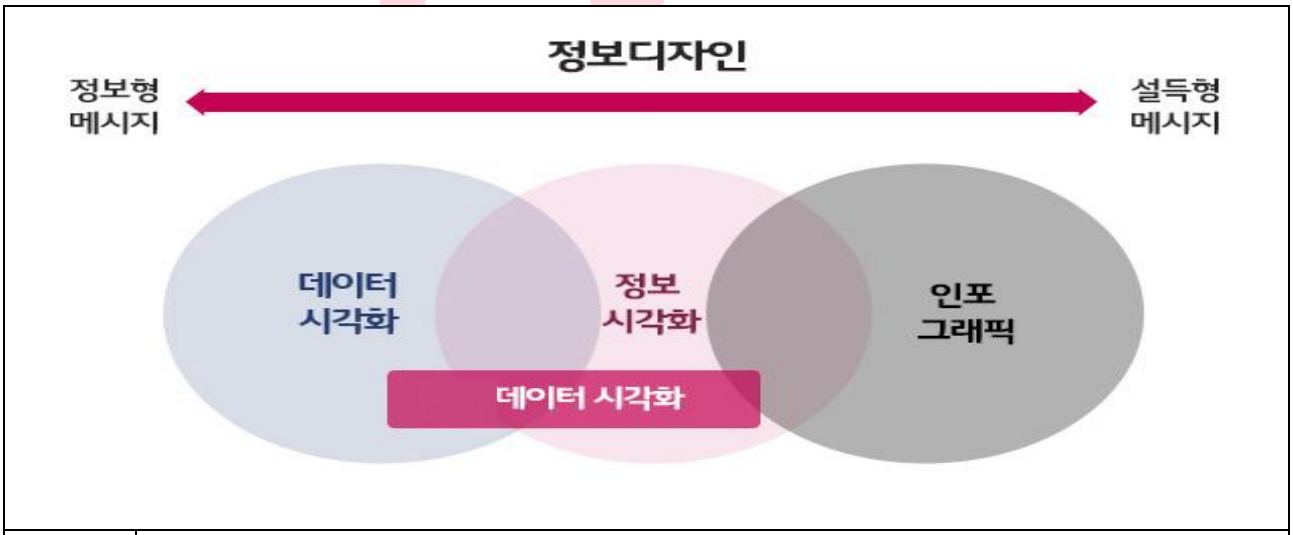
"끝"

기출풀이 의견

1. 도입부로 ISP과 BPR의 필요성을 먼저 언급하고, 2,3단락엔 문제에서 물어본 ISP와 BPR의 개념과 수행 절차의 비교, 상호보완적 활용방안을 핵심 키워드와 함께 풍부하게 작성해야 하며 결론부로 ISP, BPR의 기대효과로 마무리하시면 차별화되는 답안이 될 것 같습니다.

문 제	**2. 데이터 시각화(Data Visualization)와 관련하여 다음을 설명하시오.** **가. 데이터 시각화의 개요** **나. 데이터 시각화의 원리 및 절차** **다. 데이터 시각화 유형** **라. 효과적인 데이터 시각화를 위한 효율화 방안**
출 제 영 역	디지털 서비스 / 난 이 도 ★★★☆☆
출 제 배 경	데이터 시각화에 대한 전반적인 내용 이해 및 효율화 방안 의견 제시 확인
출 제 빈 도	102 회 컴시응, 126 회 정보관리
참 고 자 료	- https://modulabs.co.kr/blog/data-visualization/(데이터 시각화 사례) - https://paullydia.tistory.com/70 (데이터 시각화)
Keyword	- 정보구조화, 정보 시각화, 정보 시각적 표현
풀 이	류연춘 PPE (130 회 정보관리 / ycel@naver.com)

1. 직관적인 정보 전달 위한 데이터 시각화의 개요



개념	정확한 의사결정과 정보 전달을 위해서 주어진 데이터의 특성과 의미를 파악하여 분석 결과를 쉽게 이해할 수 있도록 시각적으로 표현하여 전달하는 과정 및 기법	
필요성	① 데이터 속 인사이트 도출 ② 올바르고 신속한 의사결정	- 원하는 조건과 목적에 따라 데이터 신속하게 가공 - 업무 효율성, 생산성 향상, 재무 성과

- 데이터 내에서 새로운 패턴이나 관계를 발견하도록 도와주는 데이터 시각화 원리 및 절차 이해 필요

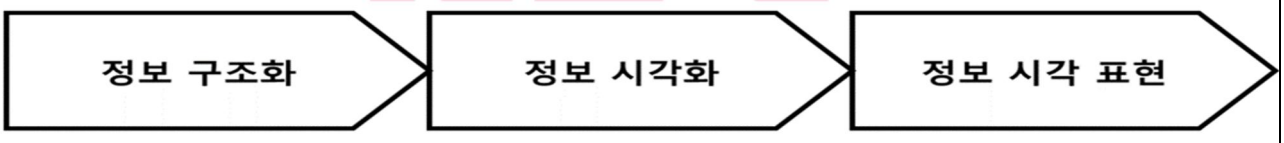
2. 데이터 시각화 원리 및 절차 설명

가. 데이터 시각화 원리

원리	핵심 원리	상세 설명
Abstraction - Figuration	추상화, 형상화	- 실제의 물체와 같은 것들일 수록 형상에 가깝고, 차트나 박스와 같이 단순화한 표현들은 추상에 가까움
Functionality - Decoration	기능성, 장식	- 장식에 집중하기보다 그 데이터 자체의 표현에 집중하였는가, 꾸미는데 더 집중하였는가를 표현
Density - Lightness	밀도 있는, 가벼운	- 자세히 살펴봄으로써 많은 데이터들을 얻을 수 있는지, 혹은 한눈에 보기에도 쉽게 데이터들을 파악할 수 있는지를 확인
Multidimensional - Unidimensional	다차원, 일차원	- 다른 다양한 종류의 데이터들을 표현하였는지, 혹은 한 두개의 데이터들을 표현하였는지를 나타냄
Originality - Familiarity	독창성, 친숙함	- 데이터 표현방식에 있어서 독창적으로 표현하였는지, 친숙하게 (막대나 꺾은 선) 표현하였는지 나타냄
Novelty - Redundancy	새로움, 반복	- 데이터를 표현하는 데 있어서, 한 번만 표현할 것인지 아니면 여러 번 반복적으로 표현할 것인지를 나타냄

- 데이터 시각화 원리를 활용하여 데이터 수집부터 시각화까지 지원하며, 시각화 절차대로 수행

나. 데이터 시각화의 절차



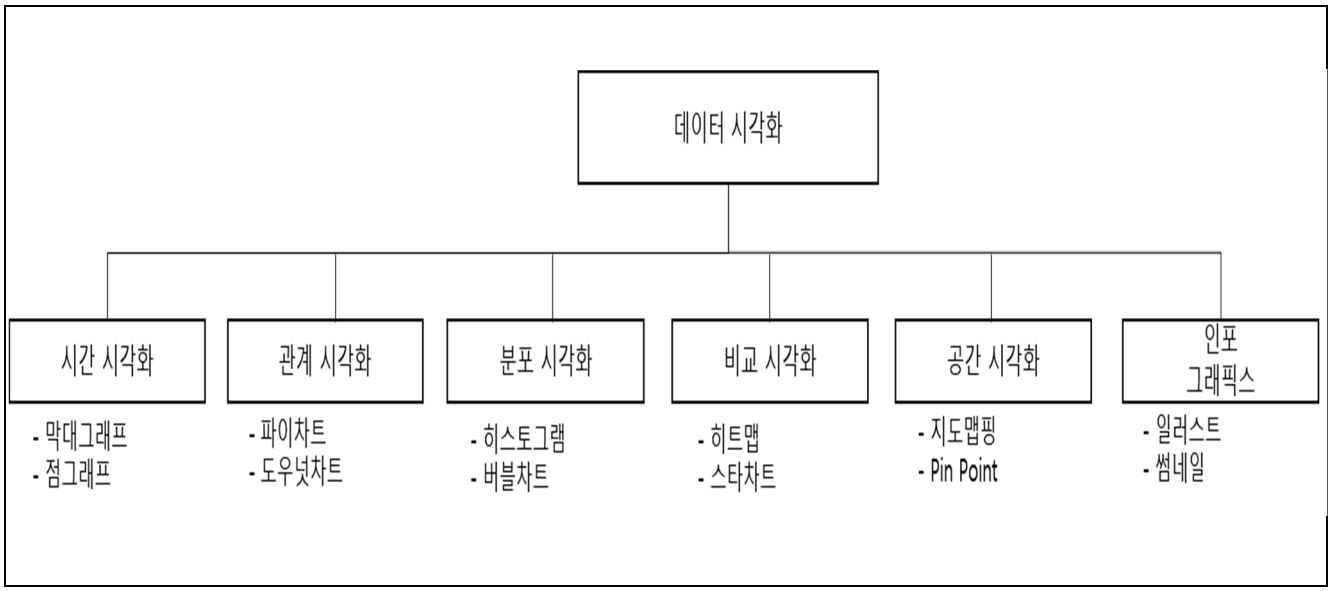
- 데이터 수집 및 탐색
- 데이터 분류 및 배열
- 시간, 분포, 관계 시각화
- 비교, 변수, 공간 시각화
- 디자인 기본원리 사용
- 인터랙션 디자인 활용

절차	상세절차	내용
정보 구조화	데이터 수집 및 탐색	빅데이터 고유 특성 훼손되지 않으면서 데이터를 수집 탐색 필요
	데이터 분류 및 배열	의도한 정보 디자인을 위한 데이터가 도출되었는지 확인
정보 시각화	시간, 분포, 관계 시각화	막대/누적그래프, 파이/도우넛 차트, 스캐터/버블/히스토그램
	비교, 변수, 공간 시각화	히트맵/체로노프 페이스/평행 좌표계/다차원 척도법, 지도맵핑
정보 시각 표현	디자인 기본원리 사용	구분, 순서, 색채 사용과 인지 등의 기본 원리를 사용하여 시각화
	인터랙션 디자인 활용	다양한 유형의 시각 자료를 적용하고 테스트하여 지속적 확인

- 데이터 시각화는 특성이 정확히 파악될 수 있도록 데이터, 목적, 유형에 맞춰서 이해를 높을 수 있도록 지원함

3. 데이터 시각화 유형 설명

가. 데이터 시각화 유형 분류



- 데이터 통계 분석 결과 등을 쉽고 명확하게 전달하기 위해 이미지나 도표 등을 이용하여 시각적으로 표현.

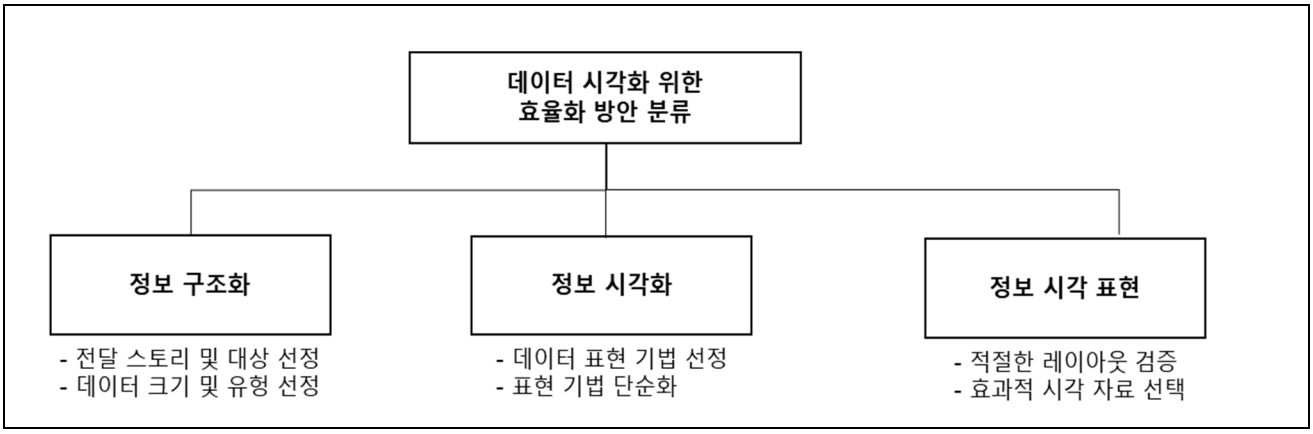
나. 데이터 시각화 유형 상세설명

유형	주요 시각화 기법	상세 설명
시간시각화	막대그래프, 점그래프, 간트차트	- 시간 데이터 기반 트렌드 경향성을 표시하는 시각화 기법 **예)** 연인들 이별시기 분석한 라인차트
분포시각화	파이차트, 도우넛차트, 트리맵	- 분포 데이터 기반 가능한 선택/결과 표시하는 시각화 기법 **예)** 전세계 빌리언 달러 지출 분포를 알 수 있는 트리맵차트
관계 시각화	스캐터플롯, 버블차트, 히스토그램	- 통계학에서 데이터 간 관계를 표시하는 시각화 기법 **예)** 미국의 주별 살인, 범죄 발생 빈도(4 가지 변수 간 관계 파악)
공간시각화	지도매핑, Pin Point	- 지도를 이용하여 직관성을 극대화한 시각화 기법 **예)** 대한민국 행정구역별 인구현황 지도 차트
비교시각화	히트맵, 스타차트, 다차원 척도법	- 각 데이터 간 유사성 관계도 확인해 볼 수 있는 시각화 기법 **예)** 2008~2009 시즌 기준 최고의 NBA 선수 기록
인포그래픽스	일러스트, 썸네일, 벡터	- 복잡한 정보를 쉽고 빠르게 전달하기 위한 시각화 기법 **예)** 유튜브 안내 썸네일 이미지

- 데이터의 의미를 명확하게 전달하기 위해 시각/분포/관계/공간/비교 등 시각화 유형에 따른 기법을 적용 시 효과적으로 데이터 시각화를 위한 효율화 방안을 고려해야 함.

4. 효과적인 데이터 시각화를 위한 효율화 방안

가. 효과적인 데이터 시각화를 위한 효율화 방안 분류



- 신속한 의사 결정 지원 및 데이터 의미 전달을 위해 정보 구조화/시각화/시각적 표현 방법 순으로 시각화 수행.

나. 효과적인 데이터 시각화를 위한 효율화 방안 상세 설명

절차	효율화 방안	상세 설명
정보 구조화	데이터 크기/유형 선정	시각화 위한 데이터 크기 및 유형 명확한 기준에 따른 선정
	대상 및 주요 스토리 선정	시각화 보고서를 전달할 대상 및 스토리 선정
정보 시각화	데이터 표현 기법 선정	데이터에 따른 표현 기법 추출 및 선정
	보고서 단순화	보고서 스토리에 맞는 시각화 도구로 보고서 단순화 수행
정보 시각 표현	Layout 검증	목적에 맞는 보고서 Layout 최종 검토
	효과적 시각 자료 선택	목적에 따른 시각적 자료 선택 통한 효율성 증대

- 비즈니스 환경에서 전문성을 갖춘 큐레이션을 통해서 데이터 분석 및 결과 활용 확대.

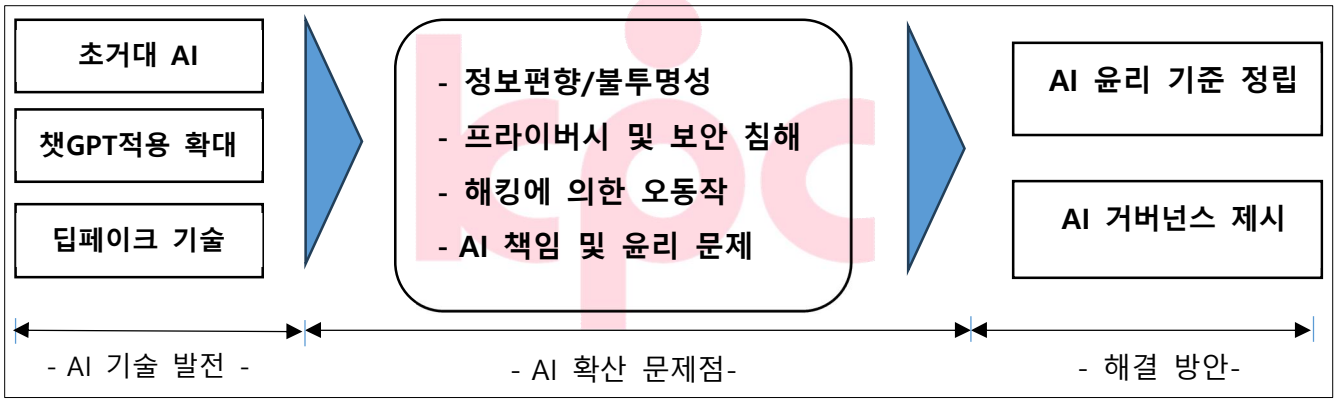
"끝"

기출풀이 의견

2. 데이터 시각화는 익숙한 토픽이지만 물어본 내용이 많으므로 시간 안배 및 분량 조절이 필요합니다. 또한, 물어본 것에 집중에서 키워드 중심으로 임팩트 있게 답안을 작성하시고, 단락 간 간글로 논리적으로 연결해 주시면 고득점을 받을 수 있을 것 같습니다.

문 제	3. 인공지능의 개발 및 적용과정에서 윤리적으로 다루어져야 할 주요 내용과 인공지능을 효과적으로 관리하고 규제하기 위한 거버넌스 모형에 대하여 설명하시오.		
출 제 영 역	인공지능	난 이 도	★★★★☆
출 제 배 경	인공지능 기술 발전에 따른 문제점 해결하기 위한 인공지능 윤리 기준과 규제하기 위한 거버넌스 모형 이해 확인		
출 제 빈 도	122 회 컴시응, 124 회 관리,		
참 고 자 료	- 인공지능 윤리(AI Ethics)란 무엇인가? (개념/사례/가이드라인) - - Char: FISCON2020 - 국내외 인공지능 정책 동향 및 시사점 (tistory.com)		
Keyword	- AI 윤리, 인간의 존엄성, 사회의 공공선, 기술 합목적성, AI 거버넌스, 가이드라인, R&R, 프로세스, 검증		
풀 이	류연춘 PPE (130 회 정보관리 / ycel@naver.com)		

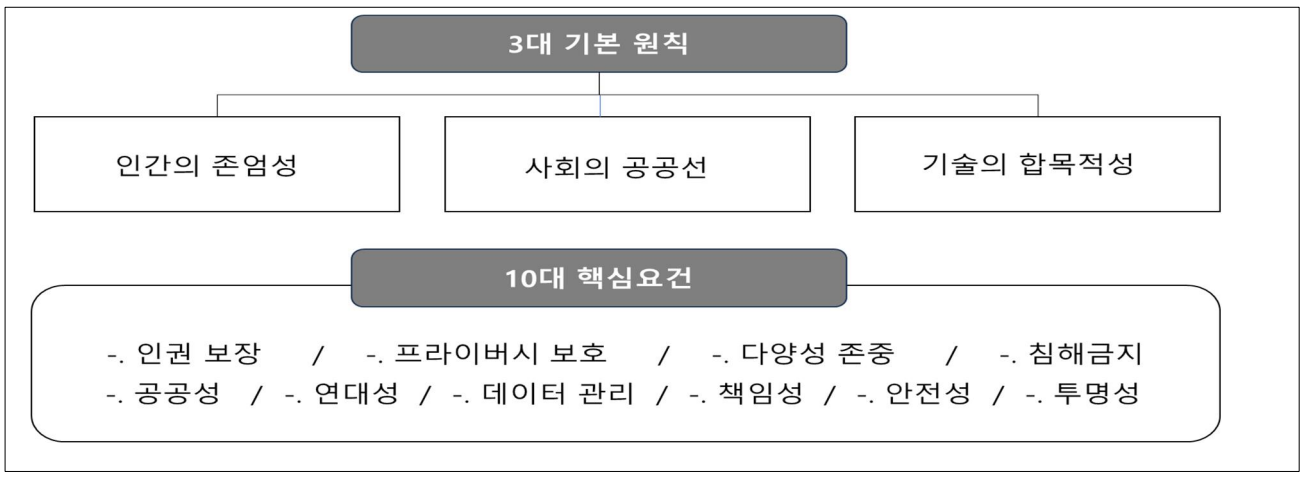
1. 인공지능 기술 발전에 따른 문제점



- 인공지능 기술 발전과 확산에 따라 발생하는 문제점들을 인식하고, 올바른 활용 및 신뢰할 수 있는 인공지능을 위해 AI 윤리 기준 정립 및 거버넌스 제시하며, 개발/적용과정에서의 윤리적으로 다뤄져야 할 내용 이해 필요

2. 인공지능의 개발 및 적용과정에서 윤리적으로 다루어져야 할 주요 내용

가. 인공지능 윤리기준의 3 대 기본원칙 및 10 대 핵심 요건



- 윤리기준이 지향하는 최고 가치를 인간성으로 설정하고, 인공지능 개발 및 활용 과정에서 3 대 원칙을 실천하고 이행할 수 있도록 전 과정에서 10 대 핵심 요건이 충족되어야 함.

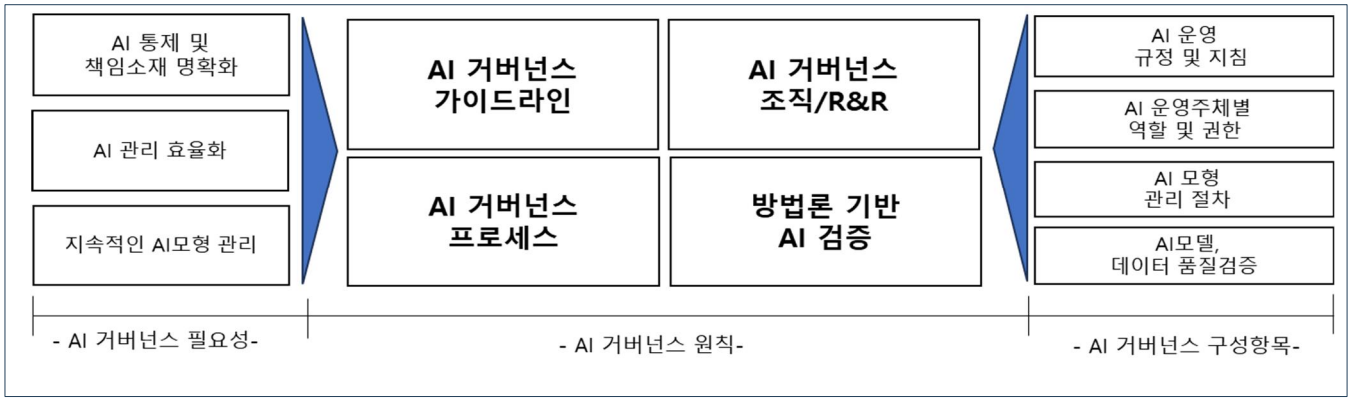
나. 인공지능 윤리 기준의 3 대 원칙 및 10 대 핵심 요건 상세 설명

3 대 원칙	10 대 핵심 요건	상세 설명
인간 존엄성 원칙	인권 보장	- 모든 인간의 권리 존중, 민주적 가치와 국제인권법 등에 명시된 권리 보장 - 인공지능의 개발과 활용은 인간의 권리와 자유를 침해하지 않아야 함
	프라이버시 보호	- 인공지능을 개발/활용 전 과정에서 개인의 프라이버시를 보호 - 인공지능 전 생애주기에 걸쳐 개인 정보의 오용을 최소화
	다양성 존중	- 개인 특성 따른 편향과 차별을 최소화하고, 모든 사람에게 공정하게 적용 - 사회적 약자, 취약 계층의 인공지능 기술 및 서비스에 대한 접근성 보장
	침해금지	- 인공지능을 인간에게 직간접적인 해를 입히는 목적 활용 금지 - 인공지능의 위험과 부정적 결과에 대응 방안을 마련
	안전성	- 인공지능 개발/활용 전 과정에 걸쳐 잠재적 위험을 방지하고 안전을 보장 - 명백한 오류, 침해 발생 시 사용자가 작동을 제어할 수 있는 기능 마련
사회 공공선 원칙	공공성	- 개인적 행복추구 외 사회적 공공성 증진과 인류의 공동 이익을 위해 활용 - 인공지능은 긍정적 사회변화를 이끄는 방향으로 활용 - 인공지능의 순기능 극대화, 역기능 최소화위한 교육을 다방면으로 시행
	연대성	- 다양한 집단 간 관계 연대성 유지, 미래세대 충분히 배려한 AI 활용 - 인공지능 전 주기에 걸쳐 다양한 주체들의 공정한 참여 기회를 보장 - 윤리적 인공지능의 개발 및 활용에 국제사회 협력
	데이터 관리	- 개인정보 등 각각의 데이터를 목적 외 용도로 활용 금지 - 데이터 수집/활용 전 과정 데이터 편향성 최소화, 데이터 품질/위험 관리
기술 합목적성 원칙	책임성	- 인공지능 개발/활용과정에서 책임주체 설정,발생할 수 있는 피해를 최소화 - 인공지능 설계 및 개발자, 서비스 제공자, 사용자 간의 책임소재를 명확화
	투명성	- 타 원칙과의 상충관계 고려 적합한 수준의 투명성과 설명 가능성을 높임 - 제품/서비스 제공 시 인공지능의 위험 등의 유의사항을 사전에 고지

- AI 윤리 기준 3 대 원칙과 10 대 핵심 요건 기반, 효과적 관리 및 규제하기 위한 AI 거버넌스 모형 제시 필요.

3. 인공지능을 효과적으로 관리하고 규제하기 위한 거버넌스 모형 설명

가. 인공지능 거버넌스 모형



- AI 개발 및 적용 등 영향에 대한 원칙과 절차를 정립하고 인간에게 유익하고 안전하게 사용되도록 체계 수립

나. 인공지능 거버넌스 모형 상세 설명

원칙	핵심요소	설명
AI 거버넌스 가이드라인	- 정부의 AI 관련 법/제도 - 회사 내부 규정	-AI 운영에 필요한 기본적인 사항의 규정 및 지침 -AI와 관련된 내부 정책에 대한 종합적 방향성 제시
AI 거버넌스 조직/R&R	- AI 위원회/협의체 - AI 전담조직/운영조직	-AI 운영주체별 역할 및 권한 -AI 생애주기를 관리 위해 AI 위원회, 전담조직 신설
AI 거버넌스 프로세스	- AI 모델 기획/설계, 개발 - AI 성능 평가/검증, 운영	-AI 모형 생성/운영/과정에서의 모든 사항을 관리 절차 -단계별 AI 프로세스 상세 설계 및 역할 지정
AI 모델 및 데이터 검증	- AI 모델검증(모델성능, 편향성 등) - AI 데이터 품질검증	-AI 모형과 관련하여 발생되는 이슈관리를 위한 검증 -AI 모델성능, 편향성 기준 검증, AI 데이터 품질 검증

-기업들이 인공지능 생태계 발전과 고객 신뢰를 위해 AI를 관리하는 AI 거버넌스 구축에 관심을 기울어야 함.

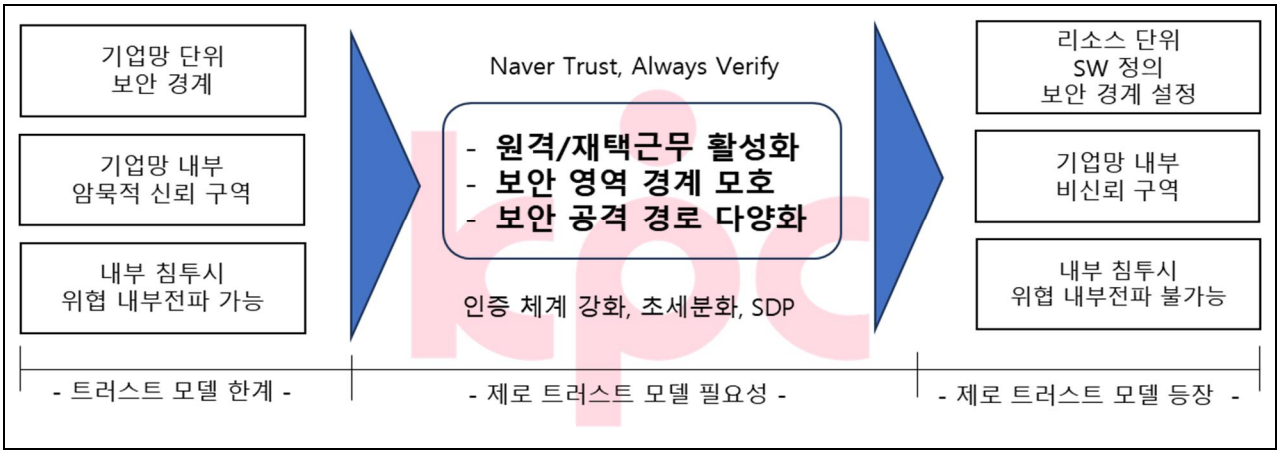
"끝"

기출풀이 의견

3. 챗GPT, 생성형 AI, 초거대 AI 등 인공지능 기술이 발전함에 따라 발생하는 문제점을 인식하고 이를 해결하기 위한 방안으로 AI윤리 기준을 이해하고 AI를 관리하기 위한 거버넌스 정립이 필요하다는 맥락으로 답안을 작성해 가시면 좋을 것 같습니다.

문 제	4. 제로 트러스트 보안(Zero Trust Security)모델의 보안원리, 핵심원칙, 적용분야를 트러스트 보안(Trust Security)모델과 비교하여 설명하시오.		
출 제 영 역	보안	난 이 도	★★★☆☆
출 제 배 경	최근 제로 트러스트 가이드 1.0 발표로 인해 기존 보안 모델과 차별화된 원칙 이해 확인		
출 제 빈 도	122 회 컴시응		
참 고 자 료	- 제로 트러스트 가이드 1.0 - KISA 한국인터넷진흥원>지식플랫폼>법령·가이드라인>가이드라인>융합보안		
Keyword	- 선인증 후연결, 트러스트 보안 모델 한계, 무신뢰 기반, 과립형 경계모델, SDP, Micro-Segmentation, 강화된 인증 PDP, PEP, 성숙도 모델		
풀 이	류연춘 PPE (130 회 정보관리 / ycel@naver.com)		

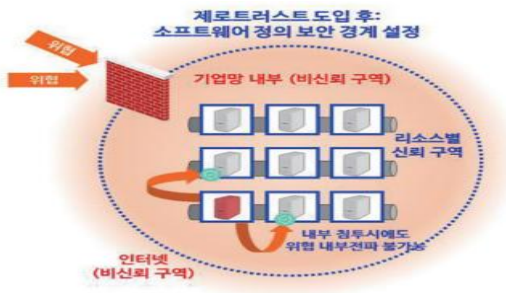
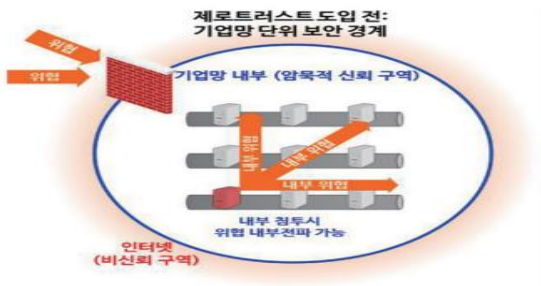
1. 보안 패러다임 변화 대응, 제로 트러스트 보안 모델 필요성



- 외부 침입자만 경계하면서 내부 있는 것들에 대해는 의심도 하지 않는 기본 정보보안이 아닌, 내부에 있는 것들에 대해서도 경계를 필요하며, 보안 패러다임 변화에 대응하기 위해 제로 트러스트 보안 모델 필요

2. 제로 트러스트 보안 모델과 트러스트 모델 보안원리 비교 설명

가. 제로 트러스트 보안 모델과 트러스트 모델 보안원리 개념 비교

제로트러스트 모델	트러스트 모델
	
(선인증 후접속) 기업의 네트워크 내·외부 모든 접속 요구는 신뢰할 수 없다는 가정으로 구축된 보안 모델	**(선접속 후인증)**기업의 네트워크 내부 접속 요구는 어느 정도 신뢰할 수 있다는 가정으로 구축된 보안 모델

- Never Trust, Always verify(절대 믿지 말고, 계속 검증하라) 기본원칙 기반 선인증 후접속 보안 모델

나. 제로 트러스트 보안 모델과 트러스트 모델 보안원리 상세 비교

비교항목	제로트러스트 모델	트러스트 모델
기본원리	- 명시적 신뢰 확인 후 리소스 접근 허용 (선인증 후접속)	- 시스템 접속 허용 후 신뢰 확인 (선접속 후인증)
신뢰수준	- 네트워크 내·외부 어디든 존재가능하며, 모든 접속 요구는 신뢰할 수 없다는가정	- 네트워크 내부 접속 요구는 어느 정도 신뢰할 수 있다는 가정
보안 경계	- 보호해야 할 모든 데이터와 컴퓨팅 서비스 각각의 자원으로 분리·보관	- 신뢰하는 자원과 신뢰하지 않은 자원 사이 보안 경계 설정
접속권한	- 정해진 권한만큼 활동 가능, 인근 자원에 대한 추가 접속 요구 시 지속적 인증으로 침투 제한	- 내부자 공모 또는 권한탈취 후 침투, 권한 상승 및 횡적이동을 통한 데이터 유출
검증/모니터링	- 자산/데이터 단위 및 Traffic 대상 로그기반 신뢰성 지속적 검증, 제어	- 외부 접속에 의한 Traffic 대상 시스템 보안 상태는 정기적 감시

- 무신뢰 기반 과립형 경계모델로 인증 관리 강화, 마이크로 세그먼테이션, SW 정의 경계 기반 3 대 원칙 적용

3. 제로 트러스트 보안 모델과 트러스트 모델 핵심 원칙 및 적용 분야 비교 설명

가. 제로 트러스트 보안 모델과 트러스트 모델 핵심 원칙 비교 설명

핵심원칙	제로트러스트 모델	트러스트 모델
인증 관리	- 다양한 인증 정보 활용한 다중인증 등 지속적 인증 통한 인증 체계 강화	- 내부 네트워크 신뢰 기반으로 필요한 최소한의 권한만 부여
	- 선 인증 후 접속	- 선 접속 후 인증
세분화	- 서버, 컴퓨팅 서비스 등 중심 작은 단위 분리, 마이크로 세그먼테이션	- 내부 네트워크의 세그먼트 간 통신 허용, 역할에 따른 서로 다른 권한 부여
	- 개별적으로 자원 배치 후 지속적으로 신뢰성 검증 및 제어	- 내부 네트워크 상 자원 간 횡적으로 자유롭게 이동 가능
경계관리	- 소프트웨어 기반으로 정의한 경계를 보안 경계로 설정	- 물리적으로 구분한 네트워크 중심으로 보안 경계 설정
	- 논리적인 보안 경계를 구분하여, 네트워크를 동적 구성	- 물리적인 보안 경계를 구분하여, 네트워크를 정적 구성

- 보호 대상 자원에 대한 접근 요구에 대해 접속을 허락할지 말지를 결정하는 과정으로 제로 트러스트 기본 철학을 바탕으로 모바일, 클라우드, 금융 분양 등 다양한 분야에 적용하여 활용

나. 제로 트러스트 보안 모델과 트러스트 모델 적용 분야 비교 설명

적용 분야	제로트러스트 모델	트러스트 모델
모바일 분야	- 멀티팩터 인증 적용 - IoT 기기 관리 및 인증 강화 - Software Defined Perimeter(SDP)	- ID/PW, OTP, 생체인증 등 하나의 인증 수단 사용 - 접속 및 사용자 데이터 보호
금융 분야	- 지속적 실시간 신원 검증 - 세분화된 금융망 설계로 필요한자원 선택적 접근	- 접근 권한 획득 후 인증 1 회성 - 기존 방화벽과 보안 솔루션 등 활용한 보안 체계 운용/관리
NW 서비스 분야	- 중앙집중적 정책/통제 관리 - 마이크로 세그먼테이션	- 신뢰할 수 있는 NW 구성과 통신 구축, 침입 탐지 및 방지 - 분산화 기반 정책 관리
클라우드 서비스 분야	- 클라우드 사용자 인증 후 접속 - 클라우드 NW 접속 후에도 다른 자원 접속 시 지속적인 인증 획득 필요	- 클라우드 접속 후 사용자 인증 - 클라우드 사용자 인증 후 다른 자원 접속 시 인증 불필요
위협 탐지 분야	- 모든 상태 모니터링 및 로그 기록 - 신뢰성 지속 검증 및 제어	- 제한된 모니터링 - 규칙 및 시나리오 기반 대응
원격 지원 분야	- BOYD 환경 안정성 향상 - 자원 분류 및 NW 세분화	- NW 분리 일부 적용 - NW 분리 일부 적용

- 제로 트러스트 보안 모델 성공적 도입 및 적용을 위해 보안기술, 기업 문화, 정책, 규제 환경 고려

4. 제로트러스트 보안 모델 도입 시 고려사항

고려사항	고려사항 설명
보안기술	- 제로트러스트에 관한 핵심 기능을 갖춘 기업의 기술과 솔루션을 지속적으로 모니터링함으로써, 기존 기술을 대체·보완하는 방법을 파악
기업 문화	- 변화에 대응하고 새로운 기술을 받아들이는데 적극적인 문화를 가진 기업이 유리, 결정권자가 제로트러스트의 필요성 인지 및 적극적인 노력 중요
정책	- 다양한 형태의 기기가 활용되는 환경 등에서 접근제어 정책과 신뢰도 평가 알고리즘을 설정하는 것은 매우 어려울 수 있음을 이해하고, 신중히 접근
규제 환경	- 기업이나 기관에서는 국가적 표준이나 지침 등 규제 환경을 고려한 제로 트러스트 도입 계획을 수립하는 것이 중요

- 제로트러스트를 적용해서 해결 또는 개선하고자 하는 영역(범위)을 명확히 정의하는 것이 중요하고, Identity 부터 Data 까지 CIS 의 제로트러스트 성숙도 모델 바탕으로 제로트러스트를 적용해 나감. **"끝"**

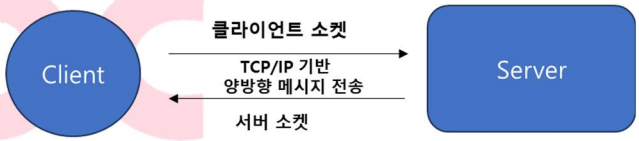
기출풀이 의견

4. 최근 부각된 제로 트러스트 모델의 필요성을 도입부로 설명하고, 이후 물어본 것에 대해 집중하여 작성하되 비교 문제이니 토픽에 맞는 비교 항목들로 풍부하게 작성하시고, 제로 트러스트 모델을 도입할 때 고려사항으로 마무리해 주시면 좋을 것 같습니다.

문 제	**5. 소켓(Socket) 통신과 관련하여 다음을 설명하시오.** **가. 소켓 통신 정의** **나. 소켓 통신 방식 개념도 및 유형** **다. TCP 소켓 및 Web 소켓 흐름 설명** **라. 소켓 통신 방식과 HTTP 통신 방식 비교**		
출 제 영 역	네트워크	난 이 도	★★★★☆
출 제 배 경	소켓통신과 HTTP 통신방식 대한 기본적 이해 확인		
출 제 빈 도	미출제		
참 고 자 료	- [CS] 그림으로 알아보는 네트워크 - 소켓 프로그래밍과 Handshaking (velog.io) - Network: 소켓(Socket)과 웹소켓 (oopy.io)		
K e y w o r d	- 양방향성, 실시간, 신뢰적, TCP, UDP, Web, read, write, accept		
풀 이	류연춘 PPE (130 회 정보관리 / ycel@naver.com)		

1. 클라이언트와 서버 간 데이터 송수신, 소켓 통신 정의

가. 소켓 통신의 정의

정의	개념도
TCP/IP 기반 네트워크 통신에서 소켓을 통해 Client-Server 간 데이터를 주고받는 양방향 연결 지향성 통신	

-소켓 통신은 클라이언트와 서버간, 혹은 컴퓨터 장치 간 데이터를 교환(SWAP)하는 데 활용됨

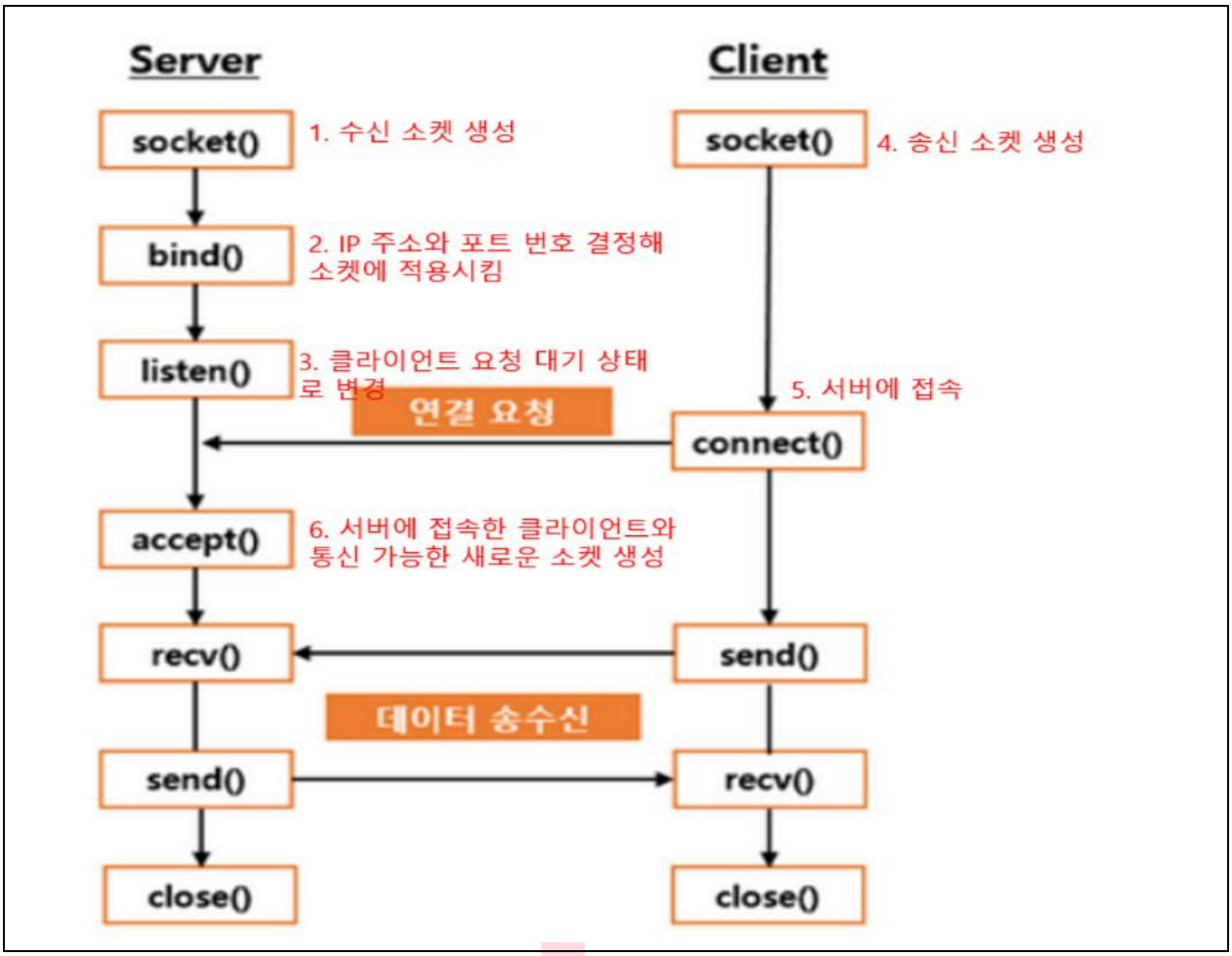
나. 소켓 통신의 특징

특징	설 명
- Server-Client 구조	- TCP/UDP 기반 동작하므로 **Server-Client 통신 구조**
- 양방향 통신	- 실시간 스트리밍이나 채팅에 주로 사용
- 프로그램 언어나 OS 에 종속	- 프로그래밍 언어마다 소켓 API 를 구현한 라이브러리가 다름

- web 환경에서 서버와 클라이언트가 port 를 통해 실시간으로 데이터를 주고받는 상황이 필요한 경우 사용

2. 소켓 통신 방식 개념도 및 유형

가. 소켓 통신 방식 개념도



- 소켓 통신 방식은 TCP, UDP 통신으로 분류할 수 있으며, 특성과 용도에 따라 선택적으로 적용

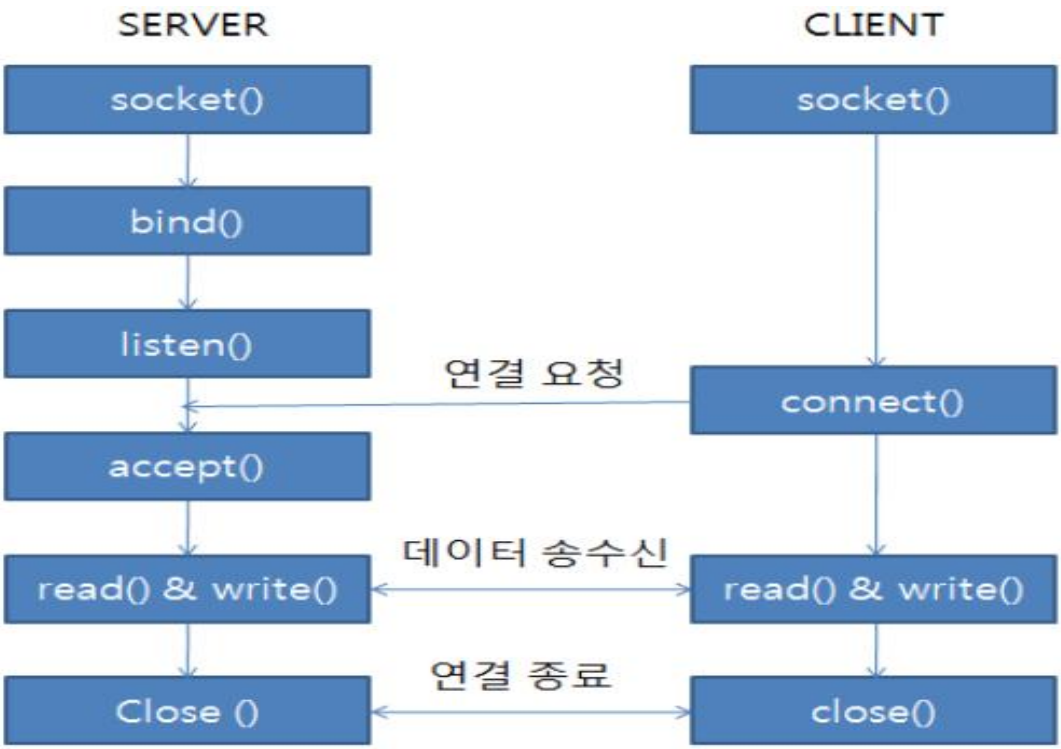
나. 소켓통신방식 유형

유형	개념	연결방식
TCP 통신	양방향 데이터 전송, 실시간으로 데이터를 신속하게 전송하기 위한 연결기반 통신 기술	① 연결기반(connection) ② 연결 후 통신(전화기) ③ 1:1 통신 방식
UDP 통신	단방향 데이터 전송, 대용량 데이터를 신속하게 전송하기 위한 비연결 기반 통신 기술	① 비연결기반(connectionless) ② 연결없이 통신(소포) ③ 1:1, 1:n, n:n 통신방식

- 시스템 구축 시 구축 시스템 특성을 고려하여 TCP 소켓과 web 소켓의 기술을 활용함.

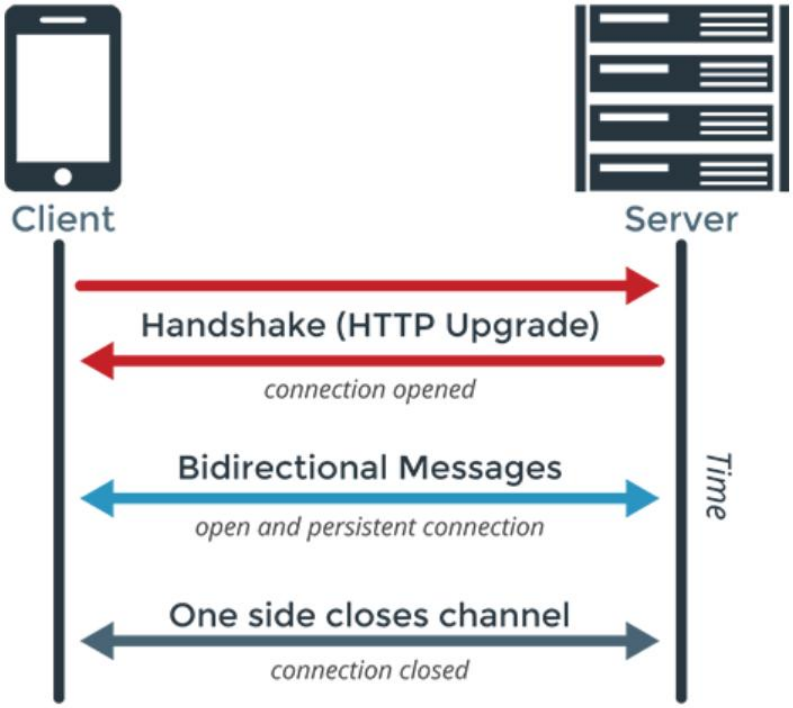
3. TCP 소켓 및 Web 소켓 흐름 설명

가. TCP 소켓 흐름 설명

<table>
<tr><th>구분</th><th colspan="3">설명</th></tr>
<tr><td>흐름도</td><td colspan="3"> 
</td></tr>
<tr><td rowspan="8">흐름
상세
설명</td><td rowspan="4">서버
측</td><td>① socket 생성</td><td>socket() 함수 이용 소켓 생성</td></tr>
<tr><td>② ip, port 설정</td><td>bind()함수 이용 ip 와 Port 설정</td></tr>
<tr><td>③ 클라이언트 대기 결정</td><td>listen()함수 이용 Client 접근 요청 수신 대기열 요청</td></tr>
<tr><td>④ 클라이언트 대기 연결</td><td>accept()함수로 Client 와 연결 대기</td></tr>
<tr><td rowspan="4">클라이언트
측</td><td>① socket open</td><td>socket() 함수 이용 소켓 오픈</td></tr>
<tr><td>② ip, port 통신</td><td>connect() 함수 이용 통신할 서버 설정된 ip 와 port 에 통신 시도</td></tr>
<tr><td>③ socket Descriptor 반환</td><td>서버가 accept() 함수 이용 클라이언트의 socket descriptor 반환</td></tr>
<tr><td>④ 서버와 통신 연결</td><td>클라이언트와 서버 간 서로 read(), write()를 하며 통신</td></tr>
</table>

- 서버 측에서 소켓을 생성하고 클라이언트 측과 서로 read(), write()를 수행하며 반복적으로 통신과정 거침.

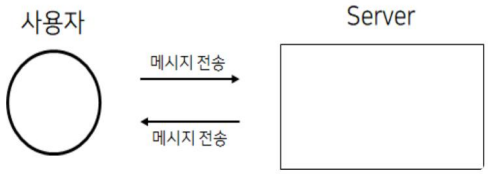
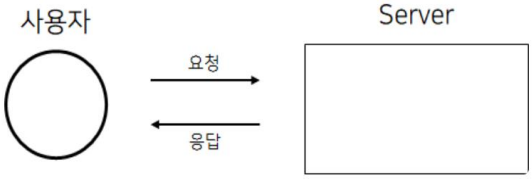
나. 웹 소켓 흐름 설명

<table>
<tr><th>구분</th><th colspan="3">설명</th></tr>
<tr><td>흐름도</td><td colspan="3"> 
</td></tr>
<tr><td rowspan="8">흐름
상세
설명</td><td rowspan="4">서버
측</td><td>① Server socket 생성</td><td>연결 대상 정보가 들어있지 않은 Socket 생성</td></tr>
<tr><td>② Server socket 바인딩</td><td>socket 과 Port 바인딩</td></tr>
<tr><td>③ Client 연결 요청 대기</td><td>Port 와 바인딩 작업 마친 후 Client 요청 대기</td></tr>
<tr><td>④ Client 연결 수행</td><td>최종 요청 accept 하면서 다른 연결 처리 위해 listen 이나 close</td></tr>
<tr><td rowspan="4">클라이언트
측</td><td>① Client socket 생성</td><td>연결 대상 정보가 들어있지 않은 Socket 생성</td></tr>
<tr><td>② 연결 요청</td><td>IP, Port 연결하고 싶은 대상 선정</td></tr>
<tr><td>③ 데이터 송수신</td><td>요청 결과 수신(Send, Receive)</td></tr>
<tr><td>④ socket 종료</td><td>데이터 송수신 불필요시 소켓 Close</td></tr>
</table>

- 웹소켓 통신은 채팅 어플리케이션, SNS, 구글 Docs, LOL 같은 멀티플레이 게임, 화상회의 등 많은 분야에서 많이 활용되며, 통신 용도와 목적에 따라 소켓 통신 방식과 HTTP 통신 방식으로 구분하여 사용함.

4. 소켓 통신 방식과 HTTP 통신 방식 비교

가. 소켓 통신 방식과 HTTP 통신 방식 개념 비교

구분	소켓 통신 방식	HTTP 통신 방식
개념	클라이언트와 서버 양쪽에서 서로에게 데이터 전달을 하는 방식의 양방향 통신	클라이언트가 요청이 있을 때 서버가 응답하는 방식의 단방향 통신
연결 방식		

- 사용자와 서버 간 통신 방향과 연결 방식에 따라 소켓 통신과 HTTP 통신 방식으로 구분하여 사용됨.

나. 소켓 통신 방식과 HTTP 통신 방식 상세 비교

비교 항목	소켓 통신 방식	HTTP 통신 방식
통신 방향성	양방향 (Client <-> Server) 통신	단방향 (Client -> Server) 통신
프로토콜	TCP, UDP 등 다양한 프로토콜 가능	HTTP, HTTPS
연결방식	직접 연결	Request-Responds 구조
데이터 형식	Byte Stream or Message	JSON, HTML, XML
통신 속도	HTTP 방식에 비해 빠름	소켓 방식에 비해 느림
보안	추가적 보안 설정 필요(방화벽, IDS, IPS 등)	HTTPS 기반 암호화 통신 지원
활용분야	스트리밍 서비스, 채팅, 게임, P2P 통신 등	Web 서비스, Web Application, API 통신 등

- 소켓 통신 방식은 직접적으로 제어를 허용하며 실시간 통신에 주로 사용되고, HTTP 통신 방식은 웹 서비스와 데이터 교환에 적용하여 사용하는 게 효과적임.

"끝"

기출풀이 의견

5. 소켓통신이라는 토픽에 대해 물어본 것이 많기 때문에 시간내 답안을 작성하기 위해 분량과 시간 조절이 관건입니다. 또한, 답인 있는 토픽 문제이므로 정확한 키워드 중심으로 균형 있게 답안을 작성하시면 고득점을 받을 수 있습니다.

문 제	**6. 아키텍처 스타일과 디자인 패턴에 대하여 다음을 설명하시오.** **가. 아키텍처 스타일과 디자인 패턴의 차이점** **나. 대표적인 아키텍처 스타일 3가지** **다. GoF(Gang of Four) 디자인 패턴의 유형을 구분하고, 유형별 대표적인 디자인 패턴 설명**
출 제 영 역	소프트웨어공학 / 난 이 도 ★★☆☆☆
출 제 배 경	아키텍처 스타일과 디자인 패턴은 빈출된 두 토픽 간 명확한 차이 이해 확인
출 제 빈 도	104 회 정보관리,111 회 컴시응, 116 회 정보관리,117 회 컴시응, 120 회 정보관리
참 고 자 료	- [도서]GoF 의 디자인 패턴 재사용성을 지닌 객체지향 소프트웨어의 핵심요소 - [Design Pattern] GoF(Gang of Four) 디자인 패턴 - HERSTORY (4z7l.github.io)
Keyword	- 생성패턴, 구조패턴, 행위패턴, Layered, Client-Sever, Master-Slave, Pipes and Filters, BlackBoard, Peer-to-Peer, MVC, Interpreter
풀 이	류연춘 PPE (130 회 정보관리 / ycel@naver.com)

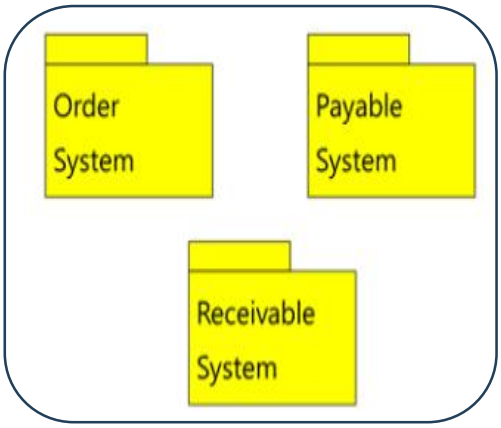
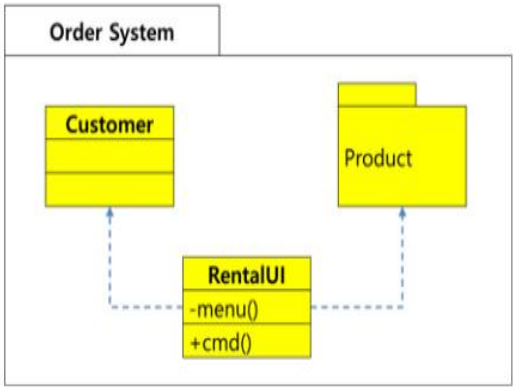
1. 아키텍처 스타일과 디자인 패턴의 차이점 설명

가. 아키텍처 스타일과 디자인 패턴의 개념 차이점 설명

아키텍처 스타일	디자인 패턴
아키텍처 설계 시 반복해서 발생하는 문제를 해결하고, 만족시켜야 할 품질 속성을 달성할 수 있는 방법을 정리한 SW 아키텍처 설계 유형 모음	소프트웨어 상세 설계 시 일반적으로 발생하는 문제 해결하기 위한 경험적인 솔루션 기반 Best Practice 모음집

- 소프트웨어 아키텍처와 디자인 패턴은 소프트웨어 설계에 있어 적용 시점 및 대상에 있어 차이점이 있음

나. 아키텍처 스타일과 디자인 패턴의 상세 차이점 설명

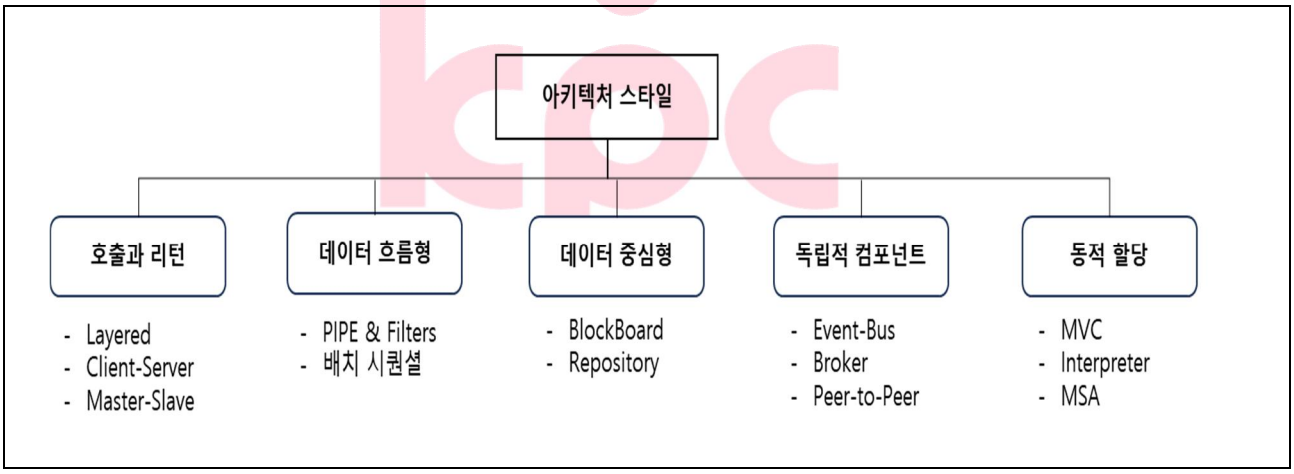
비교항목	아키텍처 스타일	디자인 패턴
적용 수준		
적용시점	소프트웨어 기본 설계 단계	소프트웨어 상세 설계 단계
관점	조직 전 이해 관계자의 경험 중심	개발자의 경험 중심

특징	- 중요한 아키텍처 설명 - 밀접한 설계 결정 사항들 패키지화 - 재사용성을 허용하는 알려진 속성 중심 데이터 중심	- 단순화, 이해 용이성 - 분석과 설계에 대한 추상적인 관점 제공 - 빠른 설계와 유지보수가 용이하며 시스템 안정성 우수
주요 유형	- 분산스타일 - 파이프와 필터 스타일 - 데이터 중심 스타일 - 규칙기반 스타일 - 블랙보드 스타일 - 레이어 스타일	- Adaptor Pattern - Factory Pattern - Command Pattern - Interpreter Pattern - Façade Pattern - Builder Pattern
문제해결 측면	- 접근 방법에 집중 - 경량화 된 가이드라인	- 특정 컨텍스트에서 검증된 구조적 해결 방안 제시

- 아키텍처 스타일을 이용해 소프트웨어 전반의 아키텍처 기본 설계를 마친 후 상세설계 시점에서 디자인 패턴을 활용하여 패키지 내부의 클래스 간의 관계를 설정, 아키텍처 스타일 특성 이해하고 선택 필요

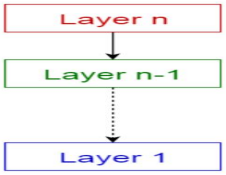
2. 대표적인 아키텍처 스타일 3 가지 설명

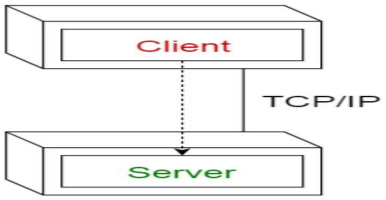
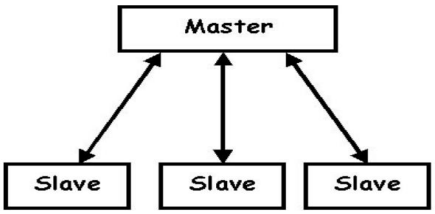
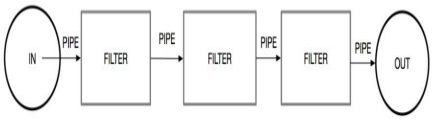
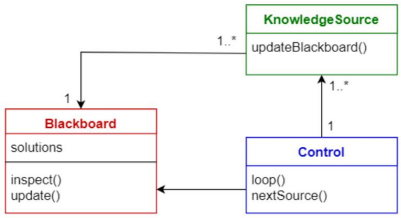
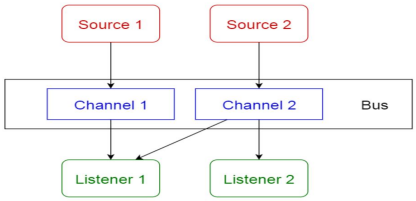
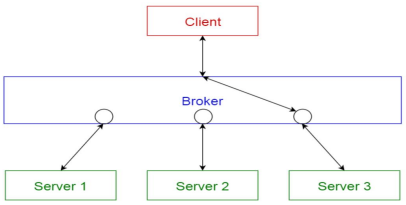
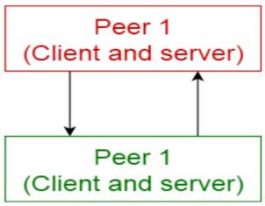
가. 대표적인 아키텍처 스타일 3 가지 분류

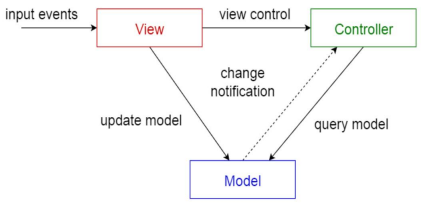
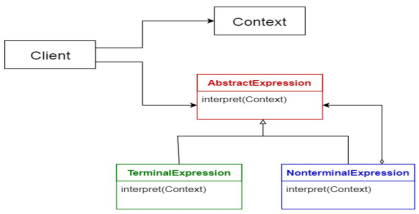


- 아키텍처 스타일은 시스템 요구사항 반영 및 기본적인 설계의 구조를 정립하고 시스템 특성 반영하여 분류함.

나. 대표적인 아키텍처 스타일 3 가지 상세 설명

아키텍처 스타일	개념도	설명
Layered		- SW 를 관심사별로 여러 개 계층으로 분리한 아키텍처 - 추상화된 인터페이스로만 소통, 단방향 의존성

Client-Sever		- 서버는 client 서브시스템에 서비스를 제공 - 클라이언트: 사용자 입력을 받아 범위 체크 및 데이터베이스 데이터 수집 - 서버: 트랜잭션 수행 및 데이터의 일관성 보장
Master-Slave		- Master-Slave 두 부분으로 구성한 모델 - Slave DB 는 Mater DB 와 Synchronization
Pipes and Filters		- 서브시스템이 입력 데이터를 받아 처리하고 결과를 다른 시스템에 보내는 작업이 반복 - 서브 시스템을 필터라고 하고 서브시스템 사이의 관계는 파이프
BlackBoard		- 블랙보드: 솔루션 객체 포함한 구조화된 전역 메모리 - 지식 소스: 자체 표현을 가진 특수 모듈 - 제어 컴포넌트: 모듈 선택, 설정 및 실행 담당
Event-Bus		- 이벤트를 처리하며 이벤트 소스/리스너/채널 /버스 4 가지 주요 컴포넌트들을 갖음. - 안드로이드 개발, 알림 서비스로 활용
Brocker		- 분리된 컴포넌트들로 이루어진 분산 시스템에서 사용 - 브로커 컴포넌트는 컴포넌트 간 통신 조정하는 역할
Peer-to-Peer		- 각 컴포넌트를 피어 (peers)라고 부름 - 피어는 클라이언트로서 피어에게 서비스를 요청, 서버로서 각 피어에게 서비스 제공

MVC		- 모델, View, 제어로 분리 - Data 를 여러 개의 View 로 표현하거나, View 와 Controller 가 Data 보다 자주 변경 이유
Interpreter		- 특정 언어로 작성된 프로그램을 해석하는 컴포넌트를 설계할 때 사용 - SQL 과 같은 데이터베이스 쿼리언어, 통신 프로토콜 정의하기 위한 언어

- SW 개발 생산성 향상을 위해 아키텍처 관점에서 아키텍처 스타일을 선택하고, 개발자 관점에서 디자인 패턴선택

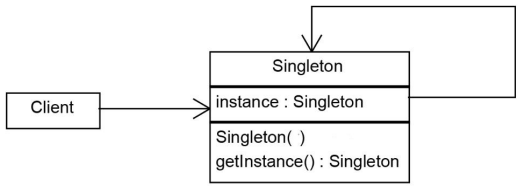
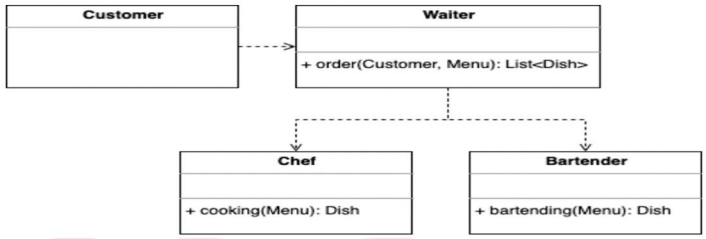
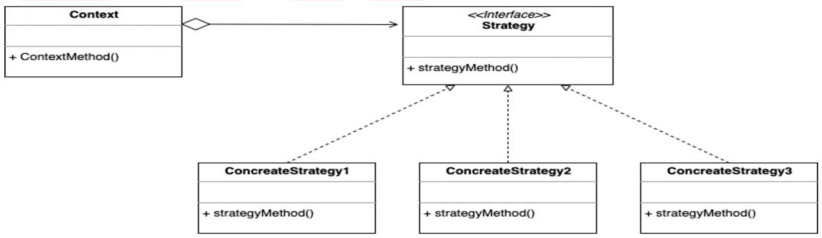
3. GoF 디자인 패턴의 유형 구분과 유형별 대표적인 디자인 패턴 설명

가. GoF 디자인 패턴의 유형 구분 설명

패턴 유형	개념	범위
생성 패턴	- 객체의 생성 방식 결정 - 클래스 정의, 객체 생성 방식 구조화, 캡슐화	**(클래스)** Factory Method **(객체)** Abstract Factory **(객체)** Singleton **(객체)** Prototype **(객체)** Builder
구조 패턴	- 객체를 조직화하는 일반적인 방법제시 - 클래스라이브러리 통합 시 유용	**(클래스)** Adapter **(객체)** Bridge, Composite, **(객체)** Decorator, Façade, **(객체)** Flyweight, Proxy
행위 패턴	- 객체의 행위를 조직화, 관리, 연합하고 객체나 클래스 연동에 대한 위협제시	**(클래스)** Interpreter **(객체)** Command, Iterator, **(객체)** Mediator, Memento, **(객체)** Observer, State, **(객체)** Strategy, Visitor

- GoF 디자인 패턴은 언어에 독립적이며 모든 프로그래밍 언어에 적용하여 설계부터 개발까지 용이성을 제공하며, 객체의 생성과 객체의 형태, 객체의 처리 방식에 따라 3 가지로 구분하고 패턴 정의, 개발.

나. 유형별 대표적인 디자인 패턴 설명

<table>
<tr><th>유형</th><th>디자인
패턴</th><th>설명 및 다이어그램</th></tr>
<tr><td rowspan="2">생성패턴</td><td rowspan="2">Singleton</td><td>전역변수를 사용하지 않고, 객체를 하나만 생성하도록 하며, 생성된 객체를 어디에서도 참조할 수 있도록 하는 패턴</td></tr>
<tr><td> 
</td></tr>
<tr><td rowspan="2">구조패턴</td><td rowspan="2">Facade</td><td>일련의 저수준 인터페이스들을 하나의 고수준 인터페이스로 묶어주는 패턴</td></tr>
<tr><td> 
</td></tr>
<tr><td rowspan="2">행위패턴</td><td rowspan="2">Strategy</td><td>행위를 클래스로 캡슐화해 동적으로 행위를 자유롭게
바꿀 수 있게 해주는 패턴</td></tr>
<tr><td> 
</td></tr>
</table>

-다양한 아키텍처 구조의 효율적인 설계, 개발을 위해 GoF 디자인패턴을 적용하여 높은 성능 향상하고, 분산 환경에서는 효과적인 이벤트 처리 위해 옵저버 패턴을 활용한 디자인을 적용하여 활용

"끝"

기출풀이 의견

6. 아키텍처 스타일과 디자인 패턴은 고전 토픽으로 누구나 접근이 가능하므로 정확한 키워드를 작성해야 차별화된 답안이 가능합니다. 또한, 비교 항목은 일반적인 것보다 토픽에 맞는 특징적인 항목을 도출하여 작성해 주시면 더욱 좋을 것 같습니다.