

ICT의 가치를
이끄는 사람들!!

125회

정보관리기술사 기출풀이 3교시

국가기술자격 기술사 시험문제

정보처리기술사 제 125 회

제 3 교시

분야	정보처리	종목	정보관리기술사	수험 번호		성명	
----	------	----	---------	-------	--	----	--

※ 다음 문제 중 4 문제를 선택하여 설명하시오. (각 25 점)

1. 배열을 이용하여 큐(Queue)를 구성할 때의 단점과 해결 방법을 설명하시오.

2. IT 프로젝트의 위험도가 크거나 대규모 사업은 상주감리 제도를 적용할 수 있다. 상주감리 업무 수행 범위와 절차를 설명하고, PMO(Project Management Office)와의 연관관계를 설명하시오.

3. QoS(Quality of Service) 방식인 DiffServ 와 IntServ 를 설명하시오.

4. 인공지능(AI) 기술이 확산됨에 따라 개인정보 침해 등 다양한 사회적 문제 발생 우려가 높아졌다. 이를 대응하기 위하여, 최근 개인정보보호위원회에서 인공지능(AI) 개인정보보호 자율점검표를 확정하였다. 인공지능(AI) 개인정보보호 자율점검표의 원칙과 총괄흐름도를 설명하시오.

5. JASON Web Token 에 대하여 다음을 설명하시오
 - 가. JASON Web Token의 개념
 - 나. JASON Web Token의 구성과 인증 메커니즘
 - 다. JASON Web Token의 장·단점과 활용방안

6. IT 투자평가 기법에 대하여 다음 물음에 답하시오.
 - 가. IT 투자 평가의 정량적 및 정성적 평가 항목 및 내용을 설명하시오
 - 나. 매출증대의 재무적 목표 달성을 위해 고객에게 배송시간 절감과 만족도 증가를 제공하려는 기업의 BSC(Balanced Score Card) 전략맵을 작성하시오.

문제	3-1 배열을 이용하여 큐(Queue)를 구성할 때의 단점과 해결 방법을 설명하시오.		
출제영역	알고리즘	난이도	★★☆☆☆
출제배경	선형 자료구조 중 큐에 대한 이해		
출제번호	<ul style="list-style-type: none"> - 기출 1회(114), KPC 6회(모의_2014.04, 합숙_2018.01, 합숙_2016.07, 합숙_2014.07, 합숙_2012.02, 모의_2013.01) 		
참고자료	<ul style="list-style-type: none"> - https://mr-son.tistory.com/49 - https://archivingstudy.tistory.com/18 - https://monzeac.tistory.com/30 		
Keyword	enqueue, dequeue, isEmpty, isFull, Peek, front, rear, 선형, 원형, 배열, 연결리스트, 우선 순위 큐		
풀이	서광석(123 회 정보관리기술사)		

1. 선형 자료구조, Queue에 대한 개요

구분	설명	
개념	<ul style="list-style-type: none"> - Front, Rear에 대한 위치 정보와 enqueue, dequeue 기능을 통해 먼저 들어온 데이터가 먼저 나오는 FIFO(First In First Out) 방식의 선형 자료구조 	
개념도		
유형	Linear Queue	<ul style="list-style-type: none"> - 크기가 제한된 공간으로 FIFO가 구현된 자료구조 - 주로 배열이나 연결리스트로 구현
	Circular Queue	<ul style="list-style-type: none"> - 저장 공간의 처음과 끝이 원형으로 연결된 자료구조 - Front와 rear의 위치가 원 형태에서 지속 이동
	Priority Queue	<ul style="list-style-type: none"> - 데이터의 우선순위가 높은 데이터부터 꺼내도록 만들어진 큐 - 데이터 삽입 시 우선순위에 따라 정렬 후 삽입

- 큐는 데이터가 저장 공간에 연속적으로 저장되어 있는 선형 자료구조로 배열이나 연결리스트로 구현

2. 배열을 이용한 큐 구현 시 단점 설명

가. 배열을 이용한 큐 구현시 동작 메커니즘 설명

주요동작	동작 코드	동작 설명
초기화	<pre>#define MAX 3 int front = 0; int rear = 0; char queue[MAX];</pre>	<ul style="list-style-type: none"> - 초기 배열 크기와 front, rear의 위치 인덱스를 지정 - front와 rear 값 초기화 상태 - front = rear로 empty 상태

큐 빈 공간 체크	<pre>int IsFull(void){ if(MAX == rear) return true; else return false; }</pre>	- 큐에 빈 공간이 있는지 체크 - 레코드 삽입 전에 수행 - rear의 위치가 최대 크기이면 full로 인식
레코드 삽입	<pre>void enQueue(int value){ if(IsFull() == false) queue[rear++] = value; }</pre>	- 큐가 가득 찬 상태인지 체크 후 enQueue 동작을 수행 - rear는 +1 된 위치의 인덱스 지정
레코드 존재 체크	<pre>int IsEmpty(void){ if(front == rear) return true; else return false; }</pre>	- 큐에 데이터가 있는지 체크하는 동작으로 레코드 인출 전에 수행 - front와 rear가 같으면 레코드가 없는 것으로 인식
레코드 인출	<pre>int deQueue(){ if(IsEmpty() == false) return queue[front++]; }</pre>	- 큐의 공백여부 체크 후 레코드 인출 동작을 수행 - front는 front + 1 인덱스 위치

- 배열 큐의 구현 시 초기화 시 저장공간의 크기가 고정되는 제약 사항이 존재

나. 배열을 이용한 큐 구현 시 단점 설명

단점	개념도	단점 설명									
크기의 제약	<p>레코드 개수 > n 일 경우</p> <table border="1"> <tr> <td>[0]</td> <td>[1]</td> <td>...</td> <td>[n-1]</td> </tr> <tr> <td>1</td> <td>2</td> <td>...</td> <td>n</td> </tr> </table> <p>- 초기화 시 큐 Size = n</p>	[0]	[1]	...	[n-1]	1	2	...	n	<ul style="list-style-type: none"> - 초기 지정한 큐 사이즈 이상의 레코드 활용 불가 - 초기에 큐 사이즈 고려 필요 	
[0]	[1]	...	[n-1]								
1	2	...	n								
공간 낭비 발생	<table border="1"> <tr> <td>[0]</td> <td>[1]</td> <td>[2]</td> </tr> <tr> <td>재사용 불가</td> <td>B</td> <td>C</td> </tr> <tr> <td>↑ Front</td> <td>↑ Rear</td> <td></td> </tr> </table>	[0]	[1]	[2]	재사용 불가	B	C	↑ Front	↑ Rear		<ul style="list-style-type: none"> - 인출이 발생된 데이터가 있던 공간은 재사용이 불가
[0]	[1]	[2]									
재사용 불가	B	C									
↑ Front	↑ Rear										
추가 비용의 발생	<p>레코드 재배치</p> <table border="1"> <tr> <td></td> <td>B</td> <td>C</td> </tr> <tr> <td>← ↑ Front</td> <td>↑ Rear</td> <td></td> </tr> </table>		B	C	← ↑ Front	↑ Rear		<ul style="list-style-type: none"> - 공간 재사용 위한 레코드 이동 시 이동 비용이 추가적으로 발생 			
	B	C									
← ↑ Front	↑ Rear										

- 배열을 이용한 큐 구현 시 저장공간의 낭비/이동비용 등의 단점이 존재하나 원형 큐, 연결리스트 사용으로 해결이 가능

3. 배열을 이용한 큐 구현 시 단점의 해결방안 설명

가. 배열을 이용한 큐 구현 시 단점 해결 방안, 연결리스트 큐의 설명

구 분	설 명	
개념	레코드 값과 다음 Node 주소로 구성된 Node를 생성, 연결, 삭제 동작의 단일 연결리스트로 큐를 구현하는 방식	
개념도	<p style="text-align: center;">head</p> <p style="text-align: center;">tail</p>	
선형 큐 단점 해결 사항	크기제약 없음	<ul style="list-style-type: none"> - 초기 공간 크기에 대한 FIX 없음. 데이터 추가 시 사용할 공간 생성
	추가비용 없음	<ul style="list-style-type: none"> - 데이터에 대한 이동 없음. 인출 시 해당 노드는 메모리 Free.
주요 동작	삽입	<ul style="list-style-type: none"> - 추가 Node를 생성하여 기존 tail 노드에 연결하고 tail 을 변경
	인출	<ul style="list-style-type: none"> - head 노드를 삭제하고 연결 된 노드를 head로 변경

- 선형 큐의 공간 크기/낭비/비용의 제약 단점을 해결하나, 지속적인 포인터 메모리 공간의 사용 및 해제가 필요.

나. 배열을 이용한 큐 구현 시 단점 해결 방안, 원형 큐의 개념

구 분	설 명
개념	큐에서 입력과 나오는 곳을 원형으로 연결하고, 모듈러 연산을 통해 선형 큐의 단점을 해결하는 1차원 배열 형태의 큐 구현 방식
개념도	<div style="display: flex; align-items: center;"> <div style="flex: 1;"> <p style="margin-left: 10px;">- 선형 큐</p> </div> <div style="margin: 0 20px;"> <p>배열 시작과 끝을 연결</p> </div> <div style="flex: 1;"> <p style="margin-left: 10px;">- 원형 큐</p> </div> </div>

선형 큐 단점 해결 사항	공간낭비 방지	- 배열로 처음과 끝을 연결한 구조. 인출된 공간은 재사용이 가능함
	추가비용 없음	- Front 와 rear 가 원 형태로 이동, 데이터의 위치 이동이 필요 없음.
주요 동작	삽입	- 기존 rear + 1 한 후 해당 index에서 레코드 삽입
	인출	- 기존 front에서 front+1 한 곳의 레코드를 인출

- 배열로 구현한 선형 큐의 단점을 해결하나 empty 와 Full 구분 어려움의 단점 존재하여 빈 공간을 두어 해결 가능.

4. 연결리스트 구현 큐와 배열로 구현한 원형 큐의 장단점 비교

구 분	연결리스트 구현 선형 큐	배열로 구현한 원형 큐
장점	<ul style="list-style-type: none"> - 초기 큐 Size에 대한 제약 불필요 - 데이터의 삽입과 삭제가 용이함 	<ul style="list-style-type: none"> - 이미 인출되어 비어 있는 공간 재사용 가능 - 선형 큐의 공간제약과 비용 문제를 해결
단점	<ul style="list-style-type: none"> - 검색 성능이 좋지 않음 - 포인터로 인한 저장 공간의 낭비가 발생 	<ul style="list-style-type: none"> - 레코드의 전체 크기에 대한 제약 발생 가능 - 초기 형태와 데이터가 full 상태의 front 와 rear index 가 동일함

- 너비우선 탐색, 프로세스 스케줄링의 FCFS, 캐시 구현 등에서 큐를 활용하며, QoS 의 WFQ 등 우선순위 필요시에는 주로 트리 형태의 Heap 으로 구현된 우선순위 큐를 사용.

"끝"

기출풀이 의견

1. 큐에 대한 기본적인 이해를 묻는 문제로 배열로 구현된 선형 큐의 공간제약, 추가비용발생의 단점을 사례 기반 제시하고 해결방안으로 원형 큐, 연결리스트 큐 등으로 기술이 필요해 보입니다.

※ 참조 1. 원형 큐의 빈공간과 Full 구분 해결 방안

구 분	설 명
해결 방안 개념도	<p>- 초기 : front = rear</p> <p>- 꽉찬 큐 : front = rear + 1</p>
해결 원리 설명	<ul style="list-style-type: none"> - 큐에서 하나의 레코드 삽입 공간은 항상 비워서 초기와 full 상태를 구분 - 초기 : front = rear , Full : front = rear+1

※ 참조 2. 큐의 유형과 구현방식 및 활용사례

구 분	설 명
유형	Linear Queue <ul style="list-style-type: none"> - 크기가 제한된 공간으로 FIFO 가 구현된 자료구조 - 빈 공간 필요 시 데이터 이동에 대한 비용 발생
	Circular Queue <ul style="list-style-type: none"> - 저장 공간의 처음과 끝이 원형으로 연결된 자료구조 - Front 와 rear 의 위치가 원 형태에서 지속 이동
	Priority Queue <ul style="list-style-type: none"> - 데이터의 우선순위가 높은 데이터부터 꺼내도록 만들어진 큐 - 데이터 삽입 시 우선순위에 따라 정렬 후 삽입
구현방식	배열 <ul style="list-style-type: none"> - 사용 공간의 크기를 선언 후 사용하는 구현 방식 - 주로 1차 배열 형태로 구현
	연결 리스트 <ul style="list-style-type: none"> - 사용 공간의 크기 제약 없이 선형 형태 큐 관리 방식 - 포인트 기반의 연결 방식
	힙(Heap) <ul style="list-style-type: none"> - 우선 순위 큐 구현시 사용 방식 - 내부 이진트리 사용, 시간 복잡도 $O(N\log N)$
활용 사례	<ul style="list-style-type: none"> - 프로세스 관리(FCFS) - 너비 우선 탐색(BFS, Breadth-First Search) 구현 - 캐시(Cache) 구현 - 우선순위가 같은 작업 예약 (프린터의 인쇄 대기열) - 동영상 스트리밍 서비스에서 컨텐츠 버퍼링

3-2 IT 프로젝트의 위험도가 크거나 대규모 사업은 상주감리 제도를 제 적용할 수 있다. 상주감리 업무 수행 범위와 절차를 설명하고, PMO(Project Management Office)와의 연관관계를 설명하시오..

출 제 영 역	소프트웨어공학	난 이 도	★★★☆☆
출 제 배 경	중소 SW 기업 활성화 정책으로 도입된 상주감리와 PMO에 대한 이해		
출 제 빈 도	기출 9회(108회, 118회, 114회, 104회 등), KPC 31회(모의_2021.05, 모의_2016.10 등)		
참 고 자 료	<ul style="list-style-type: none"> - 정보시스템_감리_수행_가이드(2.1) - PMO_도입운영가이드 		
Key word	전자정부법 57조, 64조, 대기업참여제한, 상주감리계획서, 사업관리, 품질강화		
풀 이	서광석(123회 정보관리기술사)		

1. 프로젝트 위험 최소화, 품질 강화를 위한 상주 감리의 개요

구분	설명
개념	"전자정부법 57조 행정기관등의 정보시스템 감리" 법령에 따라 이해관계로부터 독립된 자가 제3자적 관점에서 점검 및 문제점을 개선하도록 하는 제도
추진배경	<div style="display: flex; align-items: center; justify-content: space-between;"> <div style="flex: 1;"> <div style="border: 1px solid orange; padding: 10px; margin-bottom: 10px;"> <ul style="list-style-type: none"> • 대기업 공공 입찰 참여제한 • 중소기업 참여 확대 • 발주 기관 사업 역량 문제 </div> <div style="text-align: center; margin: 0 auto; width: 40px;"> </div> <div style="border: 1px solid orange; padding: 10px; margin-top: 10px;"> <ul style="list-style-type: none"> • 발주기관 사업 역량 보완 • 정보시스템 품질 제고 • 산출물 검토 및 기술적 자문 </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 20px;"> 도입 배경 도입 효과 및 역할 </div> </div>
자격요건	<ul style="list-style-type: none"> - 사업비 20억 이상인 사업감리 경력 3회 이상인 수석감리원 - PM/QA 경력 3년 이상인 수석감리원 - 기타 발주자가 인정한 수석감리원

- 상주감리는 추가 감리의 한 유형으로 위험도가 크거나 대규모 사업의 경우, 기술 지원이나 의사결정 및 의사소통의 활성 지원의 역할을 수행으로 발주기관 업무를 지원함.

2. 상주감리의 업무 수행 범위와 절차 설명

가. 상주감리의 업무 수행 범위 설명

감리기준	업무 범위	정보화사업 표준점검항목
제 10 조의 2 제 1 항 제 1 호	- 사업수행계획서, 상세공정표(WBS), 개발방법론의 공정·산출물 조정 내역 검토	- 일정관리체계 수립, 관리여부 - 품질관리체계 수립, 관리여부
제 10 조의 2 제 1 항 제 2 호	- 과업 범위(요구사항) 구체화, 과업변경 영향·타당성 검토	- 범위관리체계, 변경관리체계 수립, 관리 여부 - 범위 변경관리 수행 여부 - 사용자요구사항 반영 여부
제 10 조의 2 제 1 항 제 3 호	- 상세공정표에 따른 계획 대비 실적 점검 및 이행 상태 확인	- 진척관리 수행 여부
제 10 조의 2 제 1 항 제 4 호	- 산출물에 대한 품질 검토	- 품질관리 수행 여부
제 10 조의 2 제 1 항 제 5 호	- 위험요소 사전 파악 및 개선방향 제시	- 위험관리체계 수립, 관리여부 • 위험관리 수행 여부
제 10 조의 2 제 1 항 제 6 호	- 쟁점사항에 대한 기술검토 지원 및 의견조율	- 위험관리 수행 여부
제 10 조의 2 제 1 항 제 7 호	- 발주자의 의사결정지원 및 자문	- 의사소통관리체계 수립, 관리 여부 - 의사소통관리 수행 여부
제 10 조의 2 제 1 항 제 8 호	- 그 밖에 사업을 성공적으로 수행하기 위해 필요한 지원 등	- 자원관리체계 수립, 관리 여부 - 인력관리 수행 여부
제 3 조 제 2 항단서	- 요구사항정의서의 과업내용 반영여부	- 사용자요구사항 반영 여부

- 상주 감리 수행 시 요구정의 단계 감리는 생략하고, 상주 감리원이 과업내용 반영 여부 직접 점검 필요함.

나. 상주감리의 업무 수행 절차 설명

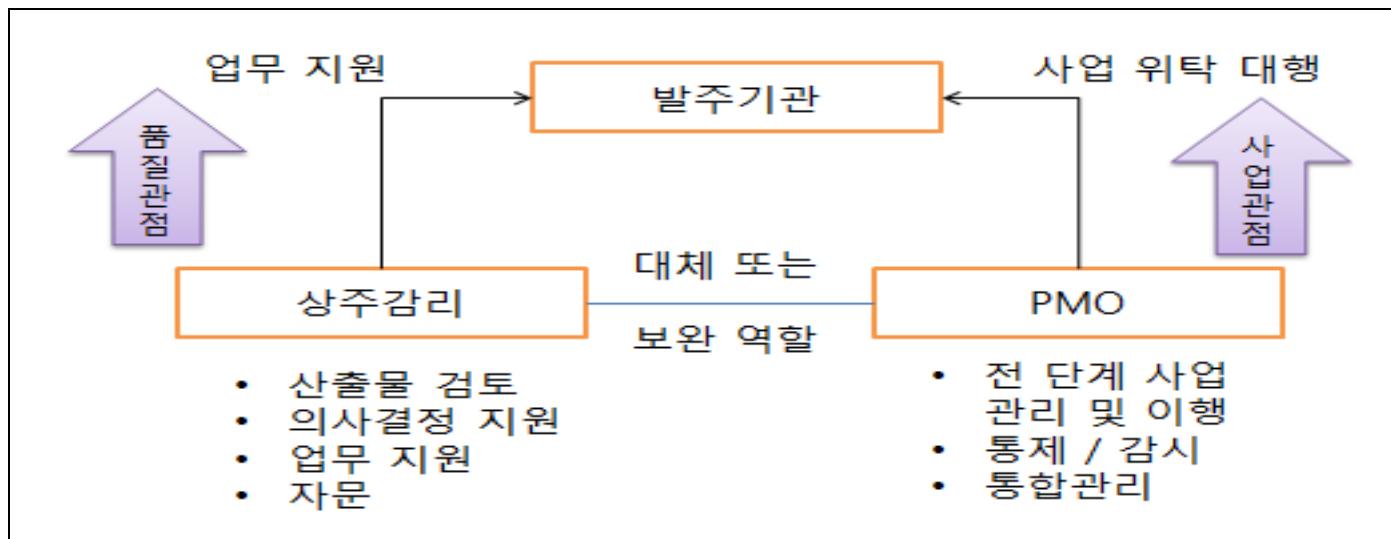
절차	수행 사항		주요 산출물
A00. 계획 수립	<ul style="list-style-type: none"> - 투입기간, 투입인력, 업무 범위에 대한 계획 수립 - 감리 사업 수행계획서의 일부로 포함 또는 별도 작성 		<ul style="list-style-type: none"> - 상주감리 계획서
B 0 0 . 상 주 감 리 수 행	B10. 검토수행	<ul style="list-style-type: none"> - 점검항목에 대한 상시 점검 및 결과를 사업자/발주자에게 공유 - 상주 감리 점검 항목 매핑표 활용 	<ul style="list-style-type: none"> - 감리수행 결과보고서 - 시정조치 확인보고서
	B20. 정기보고	<ul style="list-style-type: none"> - 계획서의 일정에 따라 정기적인 보고를 수행 - 투입 감리원, 기간, 수행활동 작성 	<ul style="list-style-type: none"> - 정기 보고서

C00. 완료보고	- 상주감리의 종료시 완료에 대한 보고를 수행 - 완료보고 형태는 발주자와 협의	- 완료보고서
-----------	---	---------

- 상주감리와 PMO 제도는 발주기관의 사업을 지원하고 위험 최소화 목적으로 도입됨.

3. 상주감리와 PMO의 연관관계 설명

가. 상주감리와 PMO 제도의 연관도 설명



- PMO 도입 시 상주감리와의 병행 통한 보완 효과 미흡 예상 시 상주감리가 불필요 할 수 있음.

나. 상주감리와 PMO 제도의 연관 관계 역할 설명

구 분	수행주체	역할	연관 역할 설명
대체측면	상주감리	사업지원	<ul style="list-style-type: none"> - 사업지원, 변경검토, 자원, 인력관리 검토 등 사업관리 측면 지원 - PMO의 전문적인 사업관리 영역의 일부 공통적 역할 수행
	PMO	품질관리	<ul style="list-style-type: none"> - QA 조직과 QC 프로세스를 통해 품질 계획 및 관리 수행 - PMO 도입시 5 억 미만, 5 개월 미만의 경우 의무 감리 제외
보완측면	PMO	사업이행	<ul style="list-style-type: none"> - 사업관리의 전문성과 경험을 가진 PMO로 구성 - 사업 특성에 대한 이해와 도메인 업무 지식 보유 - 위험의 사전 진단과 이슈의 적시 조치 등 정량적 공정관리 지원
	상주감리	품질점검	<ul style="list-style-type: none"> - 고난이도, 고신뢰성 요구 프로젝트의 전문적 품질 점검 수행 - 제3자 관점 의사소통 및 중재자 역할 수행 - PMO 대비 기술적 측면의 자문 및 점검으로 위험 완화

- 상주 감리와 PMO 제도는 고난이도, 고신뢰가 필요한 과제에서 병행 수행 시 보다 높은 효과 가능

4. 상주감리와 PMO 제도의 비교

구분	상주감리	PMO
추진배경	대기업 참여 제한 ➔ 3 단계 감리 불충분 ➔ 상주감리 추가 허용	대기업 참여제한 ➔ 사업자 관리 역량 부족 ➔ PMO 제도 도입
목 적	- 제 3 자적 입장의 전문가에 의한 기술적 측면의 감사와 평가	- 발주자 입장의 전문가에 의한 프로젝트 관리, 이슈 중재 및 컨설팅
자격 요건	- 사업비 20 억 이상인 사업감리 경력 3 회 이상인 수석감리원 - PM/QA 경력 3 년 이상인 수석감리원 - 기타 발주자가 인정한 수석감리원	- 전자정부사업의 수행책임자 역할 3 회 이상 수행자(1 명 이상) - 전자정부 사업관리 위탁용역을 3 회이상 수행한자(2 명 이상)
투입 시점	- 프로젝트 착수단계부터 종료단계까지	- 프로젝트 계획 단계부터 사후관리 단계까지
역 할	- 품질 보증 및 단계별 감사 - 기능 및 과업이행여부 점검 및 조치확인	- 일정, 인력, 위험 등 사업관리 모니터링 및 지원 - 발주기관 요구사항 가시화 및 의사결정 지원
제약사항	- 단계별 감리시 상주 감리원 참여 금지	- 특별 제한 없음
주요산출물	- 감리계획서 - 감리수행결과보고서 - 시정조치확인보고서	- PMO 수행계획서 - PMO 검토보고서 - PMO 수행결과보고서

- 발주기관의 사업역량 확보 위한 교육 강화 및 PMO 와 상주감리 제도의 보완(병행)으로 전문성 제고하여
사업성과의 가시성 확보가 가능함.

"끝"

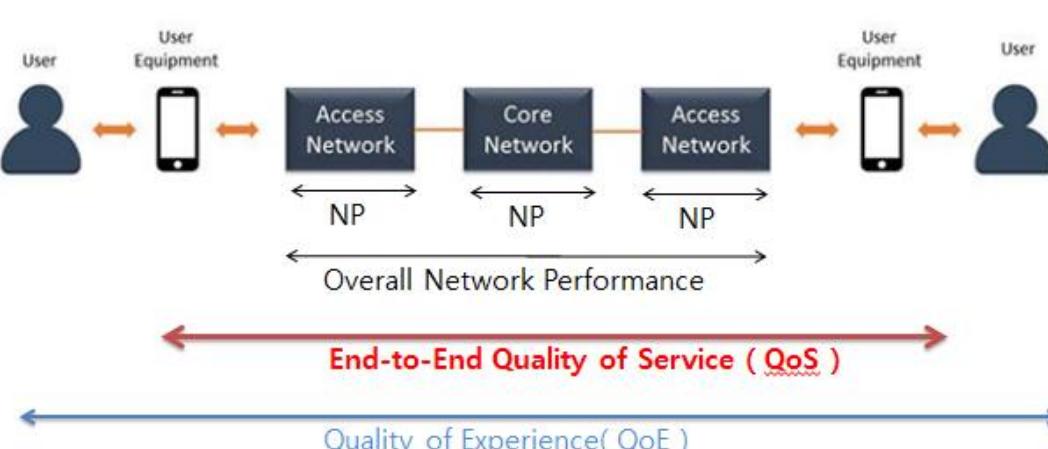
기출풀이 의견

- 상주감리와 PMO 제도에 대한 도입배경을 이해하고 각각의 차이점과 보완을 통한 시너지 효과에
대한 이해가 필요함.

문 제 3-3 QoS(Quality of Service) 방식인 DiffServ와 IntServ를 설명하시오.

출 제 영 역	네트워크	난 이 도	★★★☆☆
출 제 배 경	네트워크 서비스에 대한 품질 관리 및 보장 기술의 이해		
출 제 빈 도	기출 1회(81 회), KPC4 회(합숙(2011, 2012, 2020), 모의(2019))		
참 고 자 료	<ul style="list-style-type: none"> - 도리의 디지털라이프 - 2020년 KPC 합숙 풀이집 		
Key word	- QoE, QoS, NP, RSVP, DSCP, PHB, PATH, RESV, 분류, 미터, 마커, shaping, droping		
풀 이	서광석(123 회 정보관리기술사)		

1. 서비스 제공 품질 제고, QoS 의 개요

구 분	설 명								
개 념	<ul style="list-style-type: none"> - 한정된 네트워크 망의 대역폭을 효율적으로 사용, 네트워크 트래픽을 정책별로 제어하여 인터넷 종단간 서비스 품질을 향상시키는 기술 								
개 념 도	 <p>The diagram illustrates the network architecture and the scope of QoS and QoE. Two users are connected via their User Equipment (phones) to an Access Network, which then connects to a Core Network. The Core Network is shown with three Network Performance (NP) points. A horizontal double-headed arrow labeled "Overall Network Performance" spans the Core Network. Below the Core Network, a red double-headed arrow labeled "End-to-End Quality of Service (QoS)" spans the entire path from one user to the other. Below the Core Network, a blue double-headed arrow labeled "Quality of Experience (QoE)" spans the path from each user to their respective User Equipment.</p>								
측 정 지 표	<table border="1"> <tr> <td>대 역 폭 (Bandwidth)</td> <td> <ul style="list-style-type: none"> - 특정 Application이 할당된 NW 자원량에 따른 속도 - Queuing, Shapping, Policing으로 제어 </td></tr> <tr> <td>지 연 (Delay)</td> <td> <ul style="list-style-type: none"> - 종단간(End to End) 송수신 시간 지연 - Queuing으로 제어 </td></tr> <tr> <td>지 터 (Jitter)</td> <td> <ul style="list-style-type: none"> - 원래의 신호로부터 왜곡되는 정도 - Queuing으로 제어 </td></tr> <tr> <td>패 킷 손 실 (Packet Loss)</td> <td> <ul style="list-style-type: none"> - 전송 과정에서 충돌, 간섭 등으로 인한 패킷 drop - Queuing, RED, WRED 등으로 제어 </td></tr> </table>	대 역 폭 (Bandwidth)	<ul style="list-style-type: none"> - 특정 Application이 할당된 NW 자원량에 따른 속도 - Queuing, Shapping, Policing으로 제어 	지 연 (Delay)	<ul style="list-style-type: none"> - 종단간(End to End) 송수신 시간 지연 - Queuing으로 제어 	지 터 (Jitter)	<ul style="list-style-type: none"> - 원래의 신호로부터 왜곡되는 정도 - Queuing으로 제어 	패 킷 손 실 (Packet Loss)	<ul style="list-style-type: none"> - 전송 과정에서 충돌, 간섭 등으로 인한 패킷 drop - Queuing, RED, WRED 등으로 제어
대 역 폭 (Bandwidth)	<ul style="list-style-type: none"> - 특정 Application이 할당된 NW 자원량에 따른 속도 - Queuing, Shapping, Policing으로 제어 								
지 연 (Delay)	<ul style="list-style-type: none"> - 종단간(End to End) 송수신 시간 지연 - Queuing으로 제어 								
지 터 (Jitter)	<ul style="list-style-type: none"> - 원래의 신호로부터 왜곡되는 정도 - Queuing으로 제어 								
패 킷 손 실 (Packet Loss)	<ul style="list-style-type: none"> - 전송 과정에서 충돌, 간섭 등으로 인한 패킷 drop - Queuing, RED, WRED 등으로 제어 								

관리 기술	Queue 관리	- FIFO, CBQ, Priority Queue, Fair Queue, Weighted Fair Queue
	Traffic Shaping	- Leaky Bucket, Token Bucket, Hybrid
	사전 패킷 폐기	- RED, WRED, Tail Drop
	QoS 보장 기술	- IntServ, DiffServ, MPLS, CDN, DQSP(Dynamic QoS Support Protocol)

- QoS를 보장하기 위한 구현 모델로 예약 기반 IntServ와 차별화 기반 DiffServ 모델이 존재

2. 서비스 등급에 따른 QoS 보장 모델, DiffServ의 설명

가. DiffServ의 개념 설명

구 분	설 명
개념	패킷 헤더의 DS(Diff-Service) 필드의 DSCP를 마킹하는 서비스 클래스 별 PHB(Per Hop Behavior) 우선 순위 기반 QoS 보장 서비스 모델
개념도	
특징	- IntServ 모델의 현실적 한계를 극복

- Edge와 Core 노드별 Classifier 및 Conditioner, PHB 수행 통해 우선순위에 따른 차별화 서비스 제공

나. Diffserv의 동작 절차

구 분	설 명
Edge Node 동작	<ol style="list-style-type: none"> 입력 트래픽을 비슷한 성질로 분류하여 정의된 DSCP 마킹 할당 받은 임계 값 초과시 Shapping, Dropping으로 처리 절차대로 수행되는 상태 확인/측정
Core Node 동작	<ol style="list-style-type: none"> 입력 트래픽 분류, 해당 DSCP 기반 Forwarding 모든 Core Node에서 동일한 절차 수행

- DiffServ는 IntServ의 종단간 예약에 대한 현실적 한계를 해결하나 망 자원 범위라는 한계 존재

3. 흐름기반 QoS 보장 모델, IntServ 설명

가. IntServ의 개념 설명

구 분	설 명
개념	개별 종단간 트래픽 흐름단위로 경로상의 라우터 자원을 미리 예약(RSVP)함으로써 종단간 QoS를 보장하는 모델
개념도	
특징	<ul style="list-style-type: none"> - 종단 간 시그널링을 통해 트래픽 특성과 QoS 정보 전달 및 필요 대역폭 할당

- RSVP는 호스트가 라우터에 APP이 선택한 클래스와 트래픽 특성, QoS 요구사항 전달하는 프로토콜

나. IntServ의 동작 설명

구 분	설 명
구성도	

- 전송계층의 QoS 보장 모델로 망이 가지고 있는 자원 범위 내에서는 서비스 요청시 On-Demand 제공 가능하나 트래픽 제어의 복잡성이 존재함.

4. DiffServ 와 IntServ 의 비교

항 목	IntServ	DiffServ
모 델	- 흐름 기반 모델	- 등급 기반 모델
QoS 보장	- 흐름당(Flow 별) 종단간 QoS	- 군집당(Class 별)
QoS 보장 범위	- 단대단	- 도메인내(Edge-to-Edge)
자원 예약	- 응용에 의해 제어	- 에지 노드에서 제어
자원 관리	- 분배형	- 집중형
신호 제어 형식	- RSVP 프로토콜	- IP 패킷 헤더 내 DSCP
확장성	- 제약받음	- 확장가능
구현성	- 복잡	- 단순
망 규모	- 소규모(LAN)	- 대규모(WAN, MAN)
관련 기술	- RSVP	- PHB(Per Hop Behavior)
장점	- 단순한 망 설정	- 확장성 용이
단점	- 확장성 문제, 라우터 기능 복잡	- 종단간 QoS 를 미보장 - 노드 분석 시간 필요

- 규모가 큰 코어에서는 DiffServ 망을 사용하고 규모가 작은 송신자와 수신자 측의 에지 망에서는 IntServ 망을 사용하는 식의 Hybrid 모델 존재

"끝"

기출풀이 의견

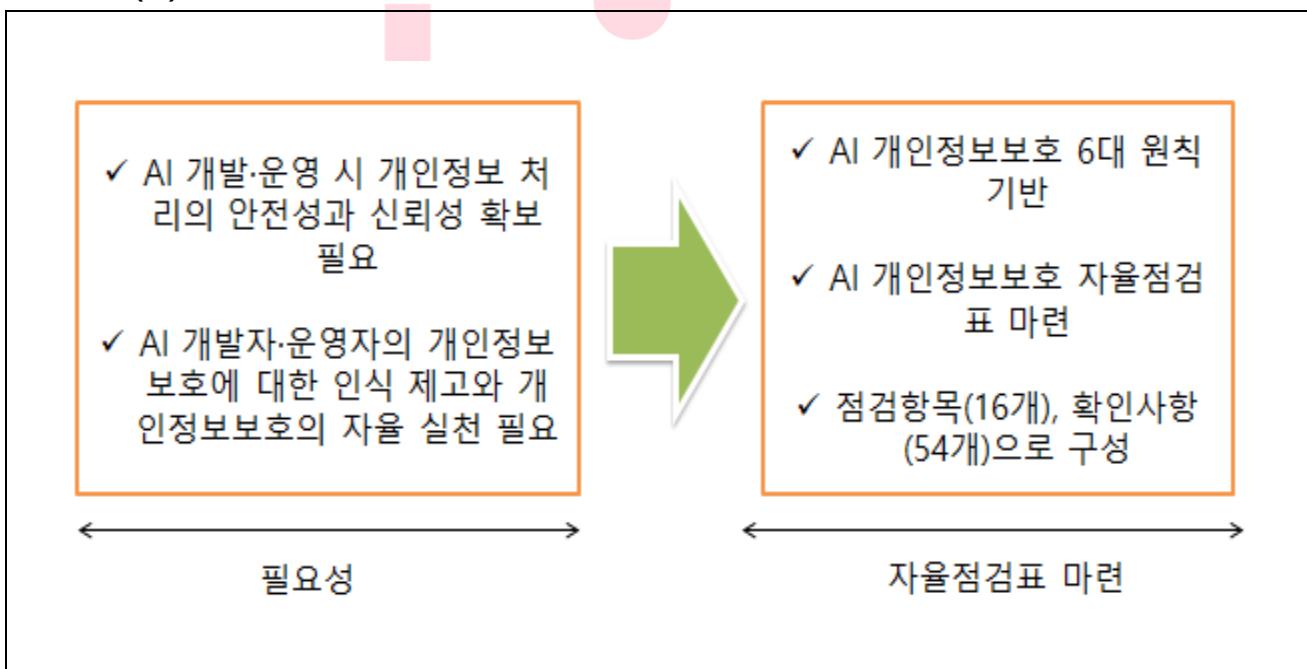
- QoS를 보장하기 위한 IntServ 모델과 DiffServ의 주요 동작 원리와 차이의 이해가 필요하며, 병합적인 Hybrid 방식의 차별화 단락 제시 고려

문

제 3-4 인공지능(AI) 기술이 확산됨에 따라 개인정보 침해 등 다양한 사회적 문제 발생 우려가 높아졌다. 이를 대응하기 위하여, 최근 개인정보보호위원회에서 인공지능(AI)개인정보 보호 자율점검표를 확정하였다. 인공지능(AI) 개인정보보호 자율점검표의 원칙과 총괄흐름도를 설명하시오.

출 제 영 역	AI	난 이 도	★★★☆☆
출 제 배 경	인공지능에서의 개인정보보호 처리의 중요성 증대와 자율점검표 마련		
출 제 빈 도	미출제		
참 고 자 료	- AI_개인정보보호_자율점검표(개발자·운영자용)(2021.5.31)		
Key word	6 대 원칙, 단계별 점검, 상시 점검, 총괄 흐름도, 개인정보보호법, Privacy by Design, AI 윤리기준		
풀 이	서광석(123 회 정보관리기술사)		

1. 인공지능(AI) 개인정보보호 자율점검표 마련 추진 배경



- 「개인정보 보호법」상 보호 원칙(§3)을 기본으로, 자율적 보호활동을 위한 「개인정보보호 중심 설계」(Privacy by Design) 원칙, 윤리적 이슈 대응을 위한 「AI 윤리기준」(2012, 과학기술정보통신부)을 반영하여, AI 관련 개인정보보호 6 대 원칙을 도출함

2. 인공지능(AI) 개인정보보호 자율점검표의 원칙 설명

가. 인공지능(AI) 개인정보보호 자율점검표의 6 대 원칙 도출 기반 설명



- 현행 법령의 의무 준수사항과 자율적 보호활동 및 윤리적 기준을 통해 6 대 원칙을 도출함.

나. 인공지능(AI) 개인정보보호 자율점검표의 6 대 원칙의 상세 설명

원 칙	설 명
적 법 성	- 개인정보의 수집·이용·제공 등 처리의 근거는 적법·명확해야 한다.
안 전 성	- 개인정보를 안전하게 처리하고 관리한다.
투 명 성	- 개인정보 처리 내역을 정보주체가 알기 쉽게 공개한다
참 여 성	- 개인정보 처리에 대한 소통체계를 갖추고 정보주체의 권리를 보장한다
책 임 성	- 개인정보 처리에 대한 관리 책임을 명확히 한다
공 정 성	- 개인정보를 수집 목적에 맞게 처리하여 사회적 차별·편향 등 발생을 최소화한다

- 인공지능(AI) 개인정보보호 자율점검표의 총괄흐름도는 단계별 점검과 상시 점검 사항으로 나누어 구성되었으며, 점검 사항 별 상세 확인 사항이 마련됨.

3. 인공지능(AI) 개인정보보호 자율점검표의 총괄 흐름도 설명

가. 단계별 점검 사항 설명

총괄흐름	단계	점검 사항 설명
1 단계	기획 / 설계	<ul style="list-style-type: none"> - 기획·설계 단계부터 PbD 원칙 적용 - 개인정보 영향평가 수행
2 단계	개인 정보 수집	<ul style="list-style-type: none"> - 적법한 개인정보의 수집 동의 방법 이행 - 동의 이외의 개인정보 수집 근거 확인 - 정보주체 이외 개인정보 수집에 대한 수집 출처 및 처리 목적 고지
3 단계	개인 정보 이용 · 제공	<ul style="list-style-type: none"> - 수집 목적 내 개인정보 이용 및 제 3 자 제공, 목적 외의 경우 별도 근거 확인 - 가명정보를 처리하는 경우 허용된 목적 및 기준 준수
4 단계	개인 정보 보관 · 파기	<ul style="list-style-type: none"> - 암호화, 접근통제 등의 조치를 통해 안전하게 보관·관리 - 개인정보가 불필요하게 되었을 때에는 자체 없이 파기

- 단계별 점검은 총 4 단계, 9 개 점검항목과 33 개 확인사항으로 구성

나. 상시 점검 사항 설명

총괄흐름	단계	점검 사항 설명
5 단계	AI 서비스 관리 · 감독	<ul style="list-style-type: none"> - 개인정보취급자에 대한 관리·감독 및 정기교육 실시 - 개인정보 처리업무 위탁 시 관리사항 문서화, 수탁자 교육 및 관리·감독 수행
6 단계	AI 서비스 이용자 보호 및 피해구제	<ul style="list-style-type: none"> - 개인정보처리방침의 투명한 공개 - 이용자의 권리행사 요구에 대한 절차 마련 및 이행 - 개인정보 유출 시 정보주체 통지, 관계기관 신고, 피해구제 지원 등 절차 마련
7 단계	개인 정보 자율보호 활동	<ul style="list-style-type: none"> - 자율적인 보호활동 적극 수행
8 단계	AI 윤리 점검	<ul style="list-style-type: none"> - 윤리적 이슈 발생여부 지속 점검·개선

- 상시 점검은 총 4 단계, 7 개 점검항목과 21 개 확인사항으로 구성됨.
- 자율점검표는 사전점검용 업무처리 지침으로 활용 가능하고, AI 관련 산업현장에서 개인정보보호책임자 (CPO)나 업무담당자 등을 대상으로 하는 교육용 교재로 활용할 수 있음

"끝"

기출풀이 의견

4. 인공지능의 개인정보보호에 대한 자율점검표에 대한 학습이 기반되어야 선택 가능한 문제로 보이며, 6대원칙 및 총괄흐름도(단계별, 상시)에 따른 점검 항목에 대한 이해가 필요

3-5 JSON Web Token에 대하여 다음을 설명하시오

- 문제
제
가. JSON Web Token의 개념
나. JSON Web Token의 구성과 인증 메커니즘
다. JSON Web Token의 장·단점과 활용방안

출제영역	디지털서비스	난이도	★★★☆☆
출제배경	MSA 환경과 API 기반 시스템의 복잡도 증가와 기존 세션 인증의 확장성 문제 대응으로 JWT 사용되어, 이에 대한 기본 이해 요구		
출제번호	기출 0회(0), KPC 2회(2020, 2021)		
참고자료	<ul style="list-style-type: none"> - https://velopert.com/2389 - https://code-machina.github.io/2019/09/01/Security-On-JSON-Web-Token.html 		
Keyword	<ul style="list-style-type: none"> - 3 단 구조, Self-contained, claim, 서명, 확장성 해결, Stateless, 위변조 방지, MSA 		
풀이	서광석(123 회 정보관리기술사)		

1. 토큰 자체 정보 보유 Claim 기반, JSON Web Token의 개념

가. JSON Web Token의 개념 설명

구분	설명
개념	<ul style="list-style-type: none"> - 헤더, 페이로드, 서명 구조의 Self-Contained 특징의 JSON 포맷을 이용하여 사용자에 대한 속성을 저장하는 Claim 기반의 Web Token으로 RFC 7519 공개 표준
등장배경	<p>등장배경</p> <p>전통적 세션 인증 웹 App. 구조</p> <p>JWT 토큰 인증 웹 App. 구조</p> <p>HTTP Stateless 대응</p> <p>Scale Out 문제 해결</p>

- 사용자의 내부 로컬 스토리지나 쿠키 및 Static 변수에 토큰 정보를 저장 관리

나. JSON Web Token의 특징 설명

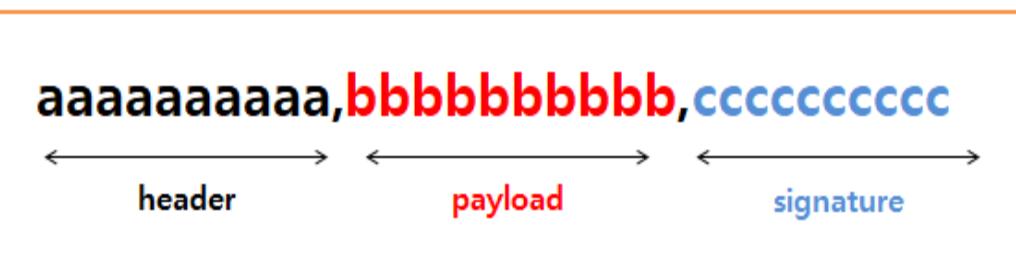
구분	특징	설명
구조적 측면	자가수용적(Self-Contained)	<ul style="list-style-type: none"> - 필요한 모든 정보를 자체적 보유 - 토큰 자체에 기본정보, 전달 정보, 서명 정보 저장
	Claim(메시지) 정의	<ul style="list-style-type: none"> - Claim JSON 문자열(헤더, 페이로드, 서명)을 BASE64 인코딩을 통해 하나의 문자열로 변환

	변조방지	- HMAC, 서명으로 헤더와 페이로드의 위변조 방지 - HS, RS, ES(256, 384, 512) 등 기본 12 개 알고리즘 사용
	전달 용이	- 자가 수용적 특성 기반 개체간 전달 용이 - HTTP 헤더 또는 URL 파라미터 전달 가능
세션 처리 문제 해결 측면	HTTP Stateless 해결	- 토큰 자체보유 및 세션 유지 불필요 - HTTP의 요청마다 인증 필요한 특성 해결
	확장성 대응	- Scale Out 시 세션 유지 및 관리 처리 로직 불필요 - Client 단 토큰 보유로 MSA 환경 등에서 확장 용이
	별도 저장소 불필요	- 추가적인 세션 관리 위한 DB 불필요 - Client의 로컬 스토리지 및 쿠키에 저장 가능
	웹/앱 간 처리 로직 상이 문제 해결	- 세션 인증 체계에서의 멀티 디바이스 접근시 로직 상이 - 동일 인증 과정 및 토큰 생성과 JWT로 인증 단순화

- JSON Web Token은 3 영역(header, payload, signature)로 구성되며 header에서 암호화 알고리즘 지정

2. JSON Web Token의 구성과 인증 매커니즘 설명

가. JSON Web Token의 구성 설명

구 분	설명		
구 성 도	<p style="text-align: center;">aaaaaaaaaaa,bbbbbbbbbbb,cccccccccc</p>  <p style="text-align: center;">← → ← → ← →</p> <p style="text-align: center;">header payload signature</p>		
	<ul style="list-style-type: none"> - URL-Safe 한 BASE64url 인코딩 사용 		
구 성	설명	예시	
H e a d e r	<ul style="list-style-type: none"> - 토큰의 타입과 해시 암호화 알고리즘으로 구성 - TYP : 토큰의 타입, "JWT" 지정 - ALG : 사용할 알고리즘 지정 	<pre>{ "typ": "JWT", "alg": "HS256" }</pre>	
P a y l o a d	<ul style="list-style-type: none"> - 토큰에 담을 Claim(registered / private / public) 정보 포함 - name / value 의 한 쌍 - registered 는 토큰 사용 위해 이름 정의된 정보 - private 는 서버와 클라이언트간 협의된 정보 - public 는 URI 형식 공개 정보 	<pre>{ "iss": "xxx.com", "exp": "1485270000000", "https://xxx.com/": true, "userId": "11028373727102", "username": "xxx" }</pre>	

Signature	<ul style="list-style-type: none"> - 헤더의 인코딩 값과, 정보의 인코딩 값을 합친 후 주어진 비밀키로 해싱하여 생성 	HMACSHA256(base64UrlEncode(header) + "." + base64UrlEncode(payload), secret)
	<ul style="list-style-type: none"> - ‘.’을 구분자로 3 가지의 문자열로 구성되고 header와 payload의 인코딩 문자열을 ‘.’으로 합친 후 해싱하여 signature로 사용 	

나. JSON Web Token의 인증 매커니즘 설명

구 분	설명
인증 매커니즘	<pre> graph TD User((User)) -- "1. 로그인" --> AuthServer[인증 서버] AuthServer -- "2. 사용자 확인" --> MemberDB([회원 DB]) MemberDB --> AuthServer AuthServer -- "3. JWT 발급" --> User User -- "4. 응답(JWT 전달)" --> User User -- "5. API 호출(+JWT)" --> AppServer[응용서버] AppServer -- "6. JWT 검증" --> AppServer AppServer -- "7. API 결과 리턴" --> User </pre>
상세 절차	<ol style="list-style-type: none"> 유저 로그인 정보를 인증 서버로 전송 유저 DB에 저장된 사용자인지 확인 자격 증명 유효성 체크 후 JWT 생성 로그인 결과와 함께 JWT를 전달 요청 API와 함께 JWT를 응용서버로 전달 JWT의 위/변조, 만료 시간 등 유효성을 검증 호출된 API의 결과를 리턴

- 별도 세션 및 인증 저장소의 불필요와 수평 스케일의 용이의 장점과 암호화 관리 측면의 단점이 존재
- JWT의 장단점을 파악하고 인증이나 데이터 전달 등의 활용분야별 옵션 및 협의 반영 필요

3. JSON Web Token의 장단점 및 활용방안 설명

가. JSON Web Token의 장단점 설명

구 분	장단점	설명
장 점	별도 인증 저장소 불필요	<ul style="list-style-type: none"> - 세션 기반 인증에서의 인증 저장소 사용 불필요 - Client 단 저장소 활용 또는 토큰 자체 저장
	Scale Out 용이	<ul style="list-style-type: none"> - 멀티 디바이스 적용 상태에서의 확장이 수월 - 토큰 기반 API 서버 별 Validate 수행 가능
	MSA 환경 대응 용이	<ul style="list-style-type: none"> - 인증서버 통해 생성된 토큰으로 Micro 서비스 별 API 호출로 세션 처리시의 로직 복잡도 문제 해결
	토큰 유효성 설정 및 REST 사용 가능	<ul style="list-style-type: none"> - 토큰 만료시간, 알고리즘 설정 가능 - REST를 통한 API 서비스 호출이 가능

단점	토큰 Modify 제한	- DB 단의 사용자 정보 조작시 토큰의 직접 수정 불가
	토큰 Size 증가	- 관리 데이터 필드 증가시 토큰의 Size 도 증가
	트래픽 영향도 존재	- 거의 모든 요청에 대해 토큰 전송으로 데이터 트래픽의 순간적 증가 등의 영향이 존재
	Payload 암호화 필요	- Payload 자체는 인코딩 된 상태로 암호화 필요 - JWE로 암호화하거나 중요 데이터 추가 관리 필요

- JWT는 세션 인증처럼 상태를 추가적으로 저장하지 않아 임의 삭제가 불가하므로 만료시간 설정이 꼭 필요

나. JSON Web Token의 활용방안 설명

구분	활용방안	설명
인증 및 권한 관리 용이 측면	Oauth 권한 부여	- OAuth 2.0 액세스 토큰 대신 서명된 JWT 토큰을 OpenID Connect 제공자에 전송 가능
	인증, 권한 관리	- 가장 일반적인 사용자 인증시 사용 - 로그인 자격 검증 및 권한에 대한 유효성 검증
정보 교류 및 안전성 측면	서명 기반 안전성 확보	- 전달된 데이터의 위변조 검증이 가능 - 서명을 기반으로 부인방지 목적 활용 가능
	컨텐츠 수준 제한	- 컨텐츠에 대한 추가 보호 수준 추가 - 사용자의 수준(권한, 보안요건) 제한시 활용

- 모던 웹 환경, MSA 환경에서 인증 및 권한 관리 위한 JWT의 활용이 증대 중

- 해싱 알고리즘 변경 및 토큰 하이재킹 등의 보안 취약점이 존재하므로 이에 대한 대응이 필요함

4. JSON Web Token에 대한 보안 취약점 및 대응방안 설명

취약점	취약점 설명	대응 방안
None 해싱 알고리즘	- 공격자가 None으로 해싱 알고리즘 변경 - None 알고리즘에 대해 유효한 토큰으로 인식하는 라이브러리 존재	- JWT 사용시 HMAC-256 등의 알고리즘 명시적 선언 - 주의 라이브러리 : node-jsonwebtoken, pyjwt, namshi/jose, php-jwt, jsjwt
토큰 하이재킹	- 전달 중간에서 토큰을 가로채거나 도난시 시스템에 대한 접근 권한 탈취 및 악용	- 사용자 컨텍스트를 토큰에 생성하고 유효성 검증 시 추가적인 검증을 수행 - HttpOnly, Secure, SameSite, cookie, prefixes와 같은 보안 설정 - 무작위 문자열의 SHA256 같은 해시 설정

- JWT의 활용의 보안 취약점은 주기적으로 대응이 필요하며 JWT 구현시 취약점 별 대응방안을 수립하고 시큐어 코딩을 활용한 보안 강화 활동 필요

“끝”

기출풀이 의견

5. MSA 환경에서 확장성 대응과 Stateless 특성 해결에 대한 JWT의 활용 증가 배경을 이해하고 제시된 질문인 JWT 구조 및 동작 절차에 대해 작성하고 장단점 및 활용방안을 작성하고 추가적으로 JWT 사용에서의 보안 취약점과 대응방안을 차별화로 제시

3-6 IT 투자평가 기법에 대하여 다음 물음에 답하시오.

가. IT 투자 평가의 정량적 및 정성적 평가 항목 및 내용을 설명하시오
 나. 매출증대의 재무적 목표 달성을 위해 고객에게 배송시간 절감과 만족도 증가를 제공하려는 기업의 BSC(Balanced Score Card) 전략맵을 작성하시오.

출 제 영 역	IT 경영	난 이 도	★★☆☆☆
출 제 배 경	IT 투자에 대한 비용 대비 효과에 대한 정성적, 정량적 측정 기법의 이해 빈출 토픽에 대한 이해 필요		
출 제 빈 도	기출 3 회(84, 119, 122), KPC 합숙 2 회(2014,2018), KPC 모의 4 회 (2011, 2012, 2015, 2020)		
	- 122 회 기출 풀이 해설지 -		
참 고 자 료	https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=best999&logNo=221620052735		
Key word	- 정보화 효과 극대화, BC, ROI, IRR, NPV, PP, BCG Matrix, Value Chain, EVA, IO		
풀 이	서광석(123 회 정보관리기술사)		

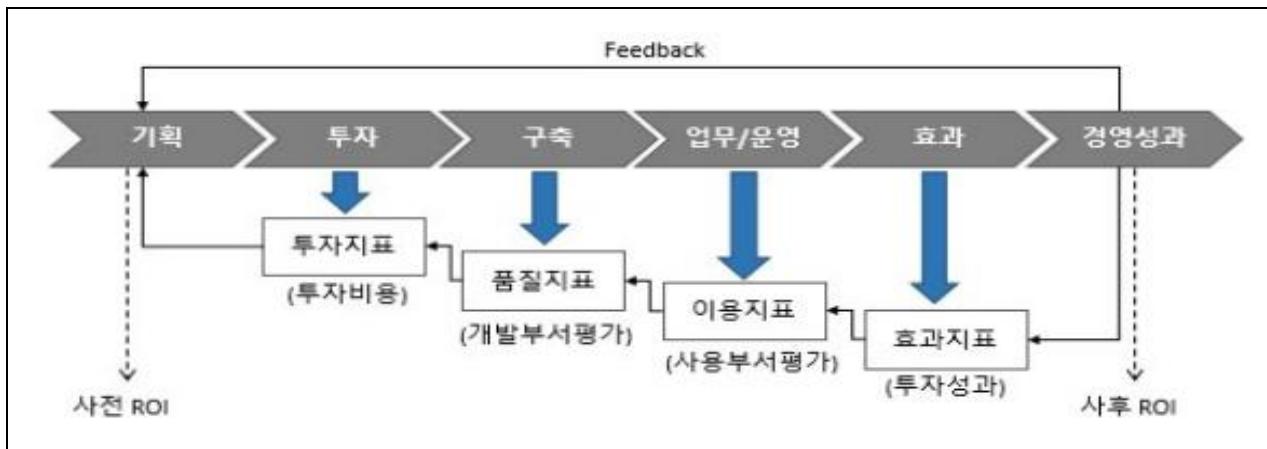
1. IT 투자평가의 개요

가. IT 투자평가의 개념 및 필요성

구 분	설명
개념	- 조직의 투자에 대한 목표 달성을 기여도를 투자/품질/이용/효과 측면에서 정량적, 정성적으로 타당성을 평가하는 활동
필요성	<div style="display: flex; align-items: center; justify-content: space-around;"> <div style="border: 1px solid black; padding: 10px; width: 30%;"> <ul style="list-style-type: none"> ✓ 투자 의사결정 최적화 ✓ 무형효과의 정량화 및 화폐가치 산출 ✓ 투자 위험 및 불확실성 감소 확보 ✓ 비용 대비 효과로 투자 타당성과 합리화 입증 </div> <div style="text-align: center; width: 40%;"> <pre> graph TD A[정보화 효과 극대화] <--> B[필요성] A <--> C[목적] A <--> D[목적 도달 방안] </pre> </div> <div style="border: 1px solid black; padding: 10px; width: 30%;"> <ul style="list-style-type: none"> ✓ 체계적인 정보화 투자관리 체계 실현 ✓ 객관적, 합리적인 정보화 성과 평가 시스템 구축 </div> </div>

- IT 투자 대비 효과 극대화 위한 평가 시스템 구축시 단계별 주요 지표별 점검이 필요

나. IT 투자 평가의 주요 지표 설명



- 정보화 투자 성과 평가를 위해 정량적, 정성적 평가 항목 이해 및 적용이 필요함

2. IT 투자 평가의 정량적 및 정성적 평가 항목 및 내용 설명

가. IT 투자 평가의 정량적 평가 항목 및 내용 설명

평 가 항 목	설 명	판단기준
비 용 편 익 비 율 (Benefit-cost ratio)	<ul style="list-style-type: none"> - 편익과 비용의 할인된 금액의 비율 - 미래에 발생될 비용과 편익을 현재가치로 환산 - 비용편익비율이 1 보다 크면 투자에 대한 경제적 타당성이 있다고 판단 	$B / C \geq 1$
N P V (Net Present Value)	<ul style="list-style-type: none"> - 순 현재가치 - 일정 할인율에 따라 모든 비용과 편익을 현재가치로 할인 후, 편익에서 비용을 차감한 값 - 순 현재가치가 0 보다 크면 경제적 타당성이 있다고 판단 	$NPV \geq 0$
내 부 수 익 율 (Internal Rate of Return)	<ul style="list-style-type: none"> - 정확한 미래 순이익 규모 평가 - 순현재가치가 0 이 되도록 하는 할인율 - 내부수익률이 사회적 할인율보다 크면 경제성이 높은 것으로 판단 	$IRR \geq r$ $NPV = 0$ 시점
회 수 기 간 법 (Payback Period)	<ul style="list-style-type: none"> - 자본 회수기간 - 자본예산 프로젝트가 초기 투자비용을 회수하는데 소요되는 시간 - 모든 다른 요인이 동일할 경우 투자 회수 기간이 짧은 투자가 더 좋은 투자를 의미 	$순이익 = 0$ 시점

- 정량적(재무적) 평가 항목으로는 추가적으로 EVA, TCO, EVS, TEI, TVO, IP 기법 존재
- 수요, 비용, 할인율 등 주요 모수 변화에 대한 민감도 분석의 병행 수행으로 각종 추정치의 오차를 보완

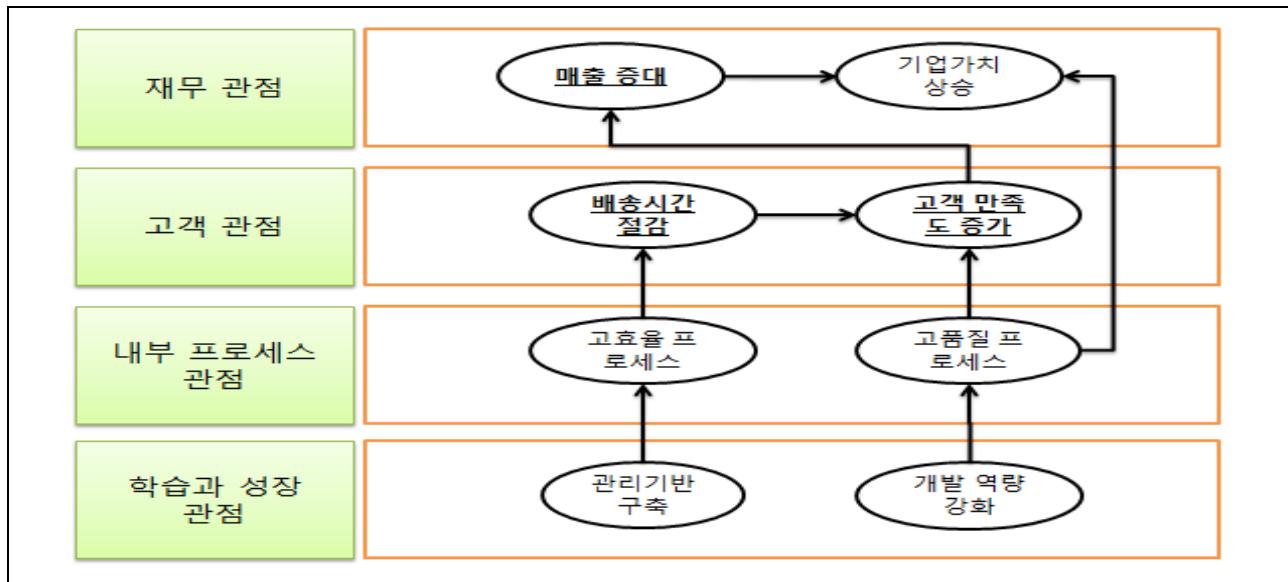
나. IT 투자 평가의 정성적 평가 항목 및 내용 설명

평가 항목	설명	분석요소 / 판단 기준
Value Chain	<ul style="list-style-type: none"> - 기업의 전반적인 생산 활동을 주 활동과 부(지원) 활동으로 나누어 분석하는 기법 	<ul style="list-style-type: none"> - 핵심 활동의 강점, 약점, 차별화 요인, 원가 동인
		<ul style="list-style-type: none"> - 상대적 가치 점수 기반 우선 순위로 결정
BCG Matrix	<ul style="list-style-type: none"> - 시장의 성장률과 그 산업 내 기업의 경쟁적인 위치를 확인하여 각기 다른 전략 사업 단위와 비교할 수 있도록 하는 포트폴리오 관리 기법 	<ul style="list-style-type: none"> - 시장 점유율, 시장 성장률
		<ul style="list-style-type: none"> - Star, Cash Cow, Dogs, ?(Question Marks)
I O (Information Orientation)	<ul style="list-style-type: none"> - 인적 중심 정보화 효과성에 기반을 둔 IT 전략을 평가할 수 있는 효과적인 프레임워크 	<ul style="list-style-type: none"> - 정보 행동 및 가치(IBV), 정보 관리 실행(IMP), 정보 기술 실행(ITP)
		<ul style="list-style-type: none"> - IO = IBV + IMP + ITP - 상대적 점수화로 결정
I P M (IT Portfolio Management)	<ul style="list-style-type: none"> - IT 관련 모든 활동의 내재적 가치와 미래의 효익 기반한 가치 측정 평가 방식 	<ul style="list-style-type: none"> - 금액, 일정, 예산, 성과, 위험
		<ul style="list-style-type: none"> - IT 자원, 투자, 프로젝트로 구분 - 평가 결과 우선순위로 결정

- 정성적 평가 항목으로 IO, IPM, IE 등의 기업이 추가 존재하며 정량적 지표의 보완 방안으로도 사용함.

3. 매출증대의 재무적 목표 달성을 위한 기업의 BSC(Balanced Score Card) 전략 맵 설명

가. CSF 기반 BSC 전략 맵 작성



- 핵심 성공요인의 도출된 KPI는 대표성, 측정 및 명확성, 통제 가능성 기반한 KPI를 선정 필요

나. CSF 기반 BSC 전략 맵의 상세 설명

관점	CSF	주요 KPI
재무 관점	매출증대	- 매출액, 영업 이익률, 당기순손실
	기업가치상승	- 경제적 부가가치(EVA), 브랜드 인지도
고객 관점	배송시간 절감	- 고객만족도향상도, 배송 확인률, 배송 소요 시간
	고객 만족도 증가	- 고객만족도 설문조사, 클레임 처리율, 정부경영평가
내부프로세스관점	고효율 프로세스	- Leadtime, 납기준수율, 배송유형별 활용률
	고품질 프로세스	- 장애율, 가용도, 불량률, 반품률
학습과 성장 관점	관리기반구축	- 부가가치 업무 처리 시간, BSC 활용률
	개발역량강화	- 최신 트랜드반영률, 교육실시 목표달성을, 교육훈련비

- BSC 활용 시 목표 달성을 위한 KPI와 CSF를 전략 맵을 작성하고 가중치 및 목표치를 설정
- CSF와 KPI에 대한 도출 시 전략 별로 구체성 여부와 정량적 측정 가능여부 등 확인 필요

4. BSC 의 CSF와 KPI 도출 시 검토사항

구 분	검토사항
CSF (Critical Success Factor)	<ul style="list-style-type: none"> - 각 세부 전략 별로 구체성, 중요성, 근본요소인지 체크 - 전략과의 미상관 요인의 포함되었는지 체크 - 회사의 장기적 경쟁우위 확보에 기여하는 요소인지 체크 - 도출시 최고경영진, 업계의 전문가, 핵심 구성원의 검토를 받았는지 체크
KPI (Key Performance Index)	<ul style="list-style-type: none"> - 사업전략 및 핵심성공요인과의 관련성, 대표성 체크 - 최고경영자, 이해관계자(주주, 고객 등)의 니즈를 반영여부 - 효과적으로 측정 가능한 명확한 지표 여부 - 통제 가능한 지표 여부

"끝"



기출풀이 의견

6. IT 투자 평과 관리 지표 및 기법은 빈출 토픽으로 품질, 이용, 효과 지표와 정성적 정량적 지표에 대한 숙지가 필요, 추가적으로 BSC에 대한 전략 맵은 제시된 매출증대, 고객만족, 배송시간 절감에 목표하여 주요 KPI 와 같이 작성하여 차별화 포인트로 활용