

제129회 정보관리기술사 해설집

2023.02.04

국가기술자격 기술사 시험문제

기술사 제 129 회

제 1 교시 (시험시간: 100 분)

분야	정보통신	자격 종목	정보관리기술사	수검 번호		성 명	
----	------	----------	---------	----------	--	--------	--

※ 다음 문제 중 10 문제를 선택하여 설명하시오. (각 10 점)

1. 디지털 플랫폼 정부의 특징, 구성요소, 기대효과
2. 정부의 인공지능 윤리기준(과학기술정보통신부 2020.12.23)에서 제시한 인공지능의 3대 기본원칙 및 10대 핵심요건
3. AOP(Aspect Oriented Programming)의 정의, 구성, 기대효과
4. 정보시스템 감리기준 고시(2021-4 호, 2021.1.19)에 의거한 현장감리의 활동, 작업내용을 기술하고 PMO(Project Management Office)와의 차이점을 설명
5. K-Means Clustering 과 DBSCAN(Density-Based Spatial Clustering of Applications with Noise) 개념, 구성요소, 장/단점
6. 데이터베이스 트랜잭션(Transaction)의 특징
7. 소프트웨어 결합도(Coupling)의 종류
8. 화이트박스 테스트(White Box Test)와 블랙박스 테스트(Black Box Test)
9. 객체지향 프로그래밍의 캡슐화(Encapsulation)
10. 코드형 인프라스트럭처

11. 정보보호 제품 신속 확인 제도
12. 비직교 다중접속(NOMA, Non-Orthogonal Multiple Access)
13. 트리정렬(Tree Sort)

01	디지털 플랫폼 정부		
문제	디지털 플랫폼 정부의 특징, 구성요소, 기대효과		
도메인	경영전략	난이도	중(상/중/하)
키워드	디지털 플랫폼 공급자, 공급자(생산자), 이용자(소비자), 생태계 전반		
출제배경	현 정부의 국정과제인 디지털 플랫폼 정부에 대한 출제		
참고문헌	디지털 플랫폼 정부의 개념과 특징(NIA, 2022.10.11)		
해설자	소원반 소민호 기술사(제 119회 정보관리기술사 / mhsope@naver.com)		

I. 디지털 플랫폼 정부의 개념 및 특징

가. 디지털 플랫폼 정부의 개념

개념도	개념
	<ul style="list-style-type: none"> - 모든 데이터가 하나로 연결되는 디지털 플랫폼 위에서, 국민, 기업, 정부가 함께 사회문제를 해결하고, 새로운 가치를 창출하는 정부 - 정부가 일방적으로 서비스를 제공하는 공급자에서, 민·관이 협업하고, 기업들이 혁신의 동반자가 되는 새로운 모델

나. 디지털 플랫폼 정부의 특징

구분	특징	설명
디지털 플랫폼	네트워크 효과	- 참여자 간 연결·소통을 통한 가치 창출 - 시장 효용 증대에 따른 규모·가치·사용자가 급격히 증대
	방대한 데이터 축적	- 모든 활동이 추적·수집·저장, 데이터기반 맞춤형 추천·광고 등 제공
	규모·범위의 경제	- 참여자 증가에 따른 추가비용이 거의 없이 사용자·데이터를 늘리고, 수집·축적된 데이터를 기반으로 인접산업으로 확장
	글로벌 특성	- 지리적 장벽이 없고 교역 비용 감소에 따른 글로벌 확장이 용이
디지털 플랫폼 공급자	민간 참여 및 양방향 서비스	- 민간의 자유로운 참여 를 통해 국민의 니즈에 맞는 서비스로의 혁신 가능
공급자(생산자)	혁신적 서비스	- 정부 제안에 따른 수동적 개발 대신, 보유 역량을 토대로 혁신적 서비스 창출 가능
이용자(소비자)	맞춤형 서비스	- 신청 없이 확인하면 서비스가 가능한 맞춤형 서비스 본격 시행
생태계 전반	창업 활성화	- 공공서비스 기반 스타트업 및 혁신기업 창업 활성화 기대

II. 디지털 플랫폼 정부의 구성요소

구성요소	역할	설명
디지털 플랫폼 공급자	생태계 활성화	- 정부가 디지털 플랫폼 구축, 운영, 제공, 플랫폼 운영방식에 대한 조정, 결정을 통해 플랫폼 생태계 활성화
공급자 (생산자)	서비스 개발	- 기업 및 개발자는 정부에서 제공하는 API, SDK, 클라우드, 공공데이터 등을 활용하여 정부 서비스를 제안, 개발, 공급, 혁신하는 역할 수행
이용자 (소비자)	서비스 활용	- 국민은 서비스 수혜자이자 데이터 및 의견 제공을 통한 디지털 플랫폼 정부 참여자로서 디지털 플랫폼 정부 기능 강화에 참여
생태계 전반	데이터 가치화	- 국민 편익, 정부 행정 효율화, 민간 혁신 등 국민, 정부, 민간이 모두에 가치 발생 - 운영상 수집되는 모든 데이터가 가치화 됨.

- 정부/공공 디지털 서비스라는 시장에 디지털 플랫폼, 디지털 서비스 기업, 디지털 서비스 이용자(국민)이 참여하는 생태계로 구성
- 정부는 생태계가 원활하게 작동하고 많은 이해관계자가 참여하도록 하기 위한 플랫폼을 구축해 서비스 제공

III. 디지털 플랫폼 정부의 기대효과

기대효과	설명
국민 맞춤형 대응성	- 정부가 보유하고 있는 다양한 데이터를 통해 개인맞춤형 대국민 서비스 제공이 가능. - 정부가 개인이 수혜 가능한 정부서비스를 제안하고 확인 받는 프로세스가 가능 - 인공지능, 빅데이터 등을 통해 민간 플랫폼 기업 수준의 맞춤형 서비스가 가능
서비스의 확장성	- 정부는 필수 디지털 인프라를 구축/제공, 플랫폼 생태계 구축 위한 민간 참여 장려 - 디지털 플랫폼이 잘 작동할 수 있도록 하는 규칙을 설정 - 정부는 민간이 디지털 플랫폼에 등록한 다양한 서비스를 활용하고 국민이 자유롭게 사용할 수 있도록 하는 촉매자의 역할을 담당 - 정부서비스는 자유로운 민간의 아이디어와 개발 능력에 따라 더욱 다양하고 혁신적인 방향으로 확장
지속가능한 혁신성	- 데이터를 기반으로 다양한 수요자와 공급자간 상호작용으로 서비스 지속적 발전 가능 - 데이터 분석을 통해 발 빠른 대응 및 정책적인 의사결정에도 객관적인 근거를 제시 - 데이터와 인공지능에 기반한 과학적인 의사결정 방식으로 전환이 가능

- 국민에게는 통합적·선제적·맞춤형 행정서비스를, 기업에게는 새로운 혁신의 기회를, 정부에게는 과학적으로 일할 수 있는 기반 마련

“끝”

02	인공지능 윤리기준		
문제	정부의 인공지능 윤리기준(과학기술정보통신부 2020.12.23)에서 제시한 인공지능의 3대 기본원칙 및 10대 핵심요건		
도메인	인공지능	난이도	중(상/중/하)
키워드	기본원칙 : 인간 존엄성 원칙, 사회의 공공선 원칙, 기술의 합목적성 원칙 핵심요건 : 인권보장, 프라이버시 보호, 다양성 존중, 침해 금지, 공공성, 연대성, 데이터 관리, 책임성, 안전성, 투명성		
출제배경	인공지능의 문제점 부각에 따라 정부에서 발표한 인공지능 윤리기준에 대한 출제		
참고문헌	사람이 중심이 되는 「인공지능(AI) 윤리기준」(과기정통부, 2020.12.23)		
해설자	소원반 소민호 기술사(제 119회 정보관리기술사 / mhsope@naver.com)		

I. 사람이 기준이 되는, 인공지능(AI) 윤리기준의 정의

- 인간성을 위한 인공지능 구현을 목표로 인공지능의 모든 개발과 활용 과정에서 지켜야 할 3대 기본원칙과 10대 핵심요건을 주된 내용으로 하는 과학기술정보통신부가 제안한 인공지능 윤리기준.

II. 인공지능(AI) 윤리기준의 3대 기본원칙 및 10대 핵심요건

가. 인공지능 윤리기준의 3대 기본원칙

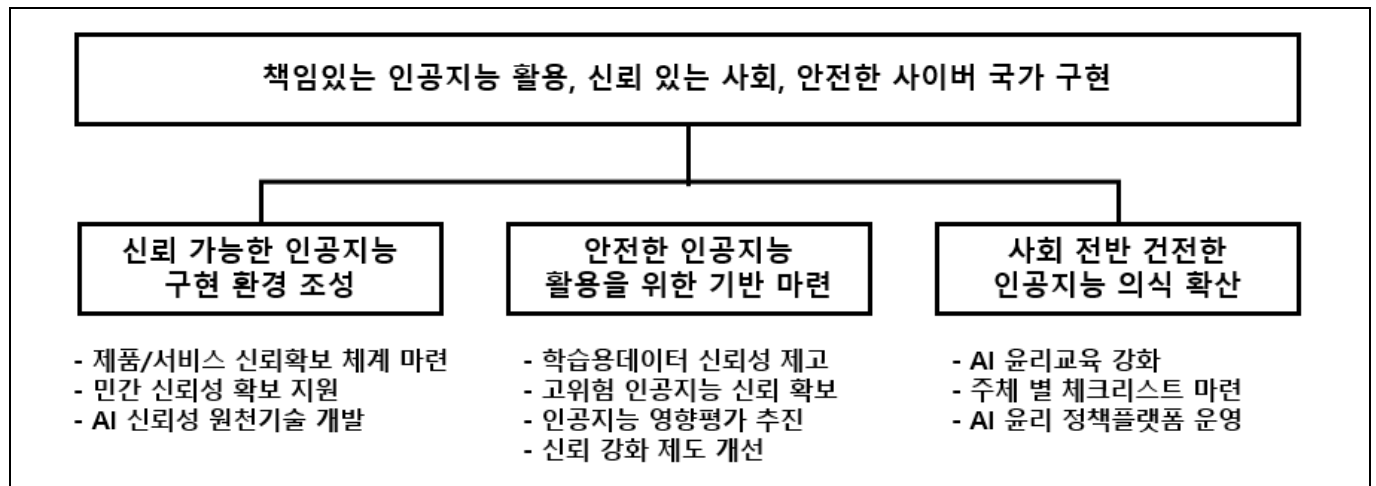
3대 기본원칙	설명
인간 존엄성 원칙	<ul style="list-style-type: none"> - 인간은 인간을 위해 개발된 기계제품과는 교환 불가능한 가치가 있음 - 인간의 생명, 정신적 및 신체적 건강에 해가 되지 않는 범위에서 개발 및 활용 - 인공지능 개발/활용은 안전성과 견고성을 갖추어 인간에게 해가 되지 않도록 함.
사회의 공공선 원칙	<ul style="list-style-type: none"> - 공동체로서 사회는 가능한 한 많은 사람의 안녕과 행복이라는 가치를 추구 - 인공지능은 사회적 약자와 취약 계층의 접근성을 보장하도록 개발 및 활용 - 공익 증진을 위한 인공지능은 사회, 국가, 글로벌 관점에서 보편적 복지를 향상
기술의 합목적성 원칙	<ul style="list-style-type: none"> - 인공지능 기술은 인류의 삶에 필요한 도구라는 목적과 의도에 부합되게 개발 및 활용되어야 하며 그 과정도 윤리적이어야 함. - 인류의 삶과 번영을 위한 인공지능 개발 및 활용을 장려하여 진흥

나. 인공지능 윤리기준의 10대 핵심요건

기본원칙	핵심요건	설명
인간 존엄성 원칙	인권보장	- 모든 인간의 권리를 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	- 사용자의 다양성 , 대표성 반영, 개인 특성에 따른 편향과 차별을 최소화 하고, 모든 사람에게 공정하게 적용 - 사회적 약자, 취약 계층의 인공지능 기술 및 서비스에 대한 접근성 보장
사회의 공공선 원칙	침해금지	- 인공지능을 인간에게 직간접적인 해를 입히는 목적 활용 금지 - 인공지능의 위험과 부정적 결과에 대응 방안 을 마련
	공공성	- 개인적 행복추구 뿐만 아니라 사회적 공공성 증진 과 인류의 공동 이익을 위해 활용 - 인공지능은 긍정적 사회변화를 이끄는 방향으로 활용 - 인공지능의 순기능 극대화, 역기능 최소화위한 교육을 다방면으로 시행
	연대성	- 다양한 집단 간의 관계 연대성 유지 , 미래세대를 충분히 배려하여 인공지능을 활용 - 인공지능 전 주기에 걸쳐 다양한 주체들의 공정한 참여 기회 를 보장 - 윤리적 인공지능의 개발 및 활용에 국제사회 협력
기술의 합목적성 원칙	데이터 관리	- 개인정보 등 각각의 데이터를 목적 외 용도로 활용 금지 - 데이터 수집/활용 전 과정 데이터 편향성 최소화 , 데이터 품질/위험 관리
	책임성	- 인공지능 개발/활용과정에서 책임주체 설정함 으로써 발생할 수 있는 피해를 최소화 - 인공지능 설계 및 개발자, 서비스 제공자, 사용자 간의 책임소재를 명확화
	안전성	- 인공지능 개발/활용 전 과정에 걸쳐 잠재적 위험을 방지 하고 안전 을 보장 - 명백한 오류, 침해 발생 시 사용자가 작동을 제어 할 수 있는 기능 마련
	투명성	- 타 원칙과의 상충관계 고려 적합한 수준의 투명성과 설명 가능성 을 높임 - 제품/서비스 제공 시 인공지능의 위험 등의 유의사항 을 사전에 고지

- 인간성을 위한 인공지능 (AI for Humanity)을 위해 인공지능 개발에서 활용에 이르는 전 과정에서 고려되어야 할 기준으로 제시
- 인공지능의 책임있는 활용, 신뢰 있는 사회, 안전한 사이버국가를 위한 신뢰 확보 방안 필요

III. 인공지능의 신뢰 확보 방안



- 인공지능 신뢰 확보 방안을 통해 누구나 신뢰할 수 있는 인공지능, 모두가 누릴 수 있는 인공지능 구현

“끝”

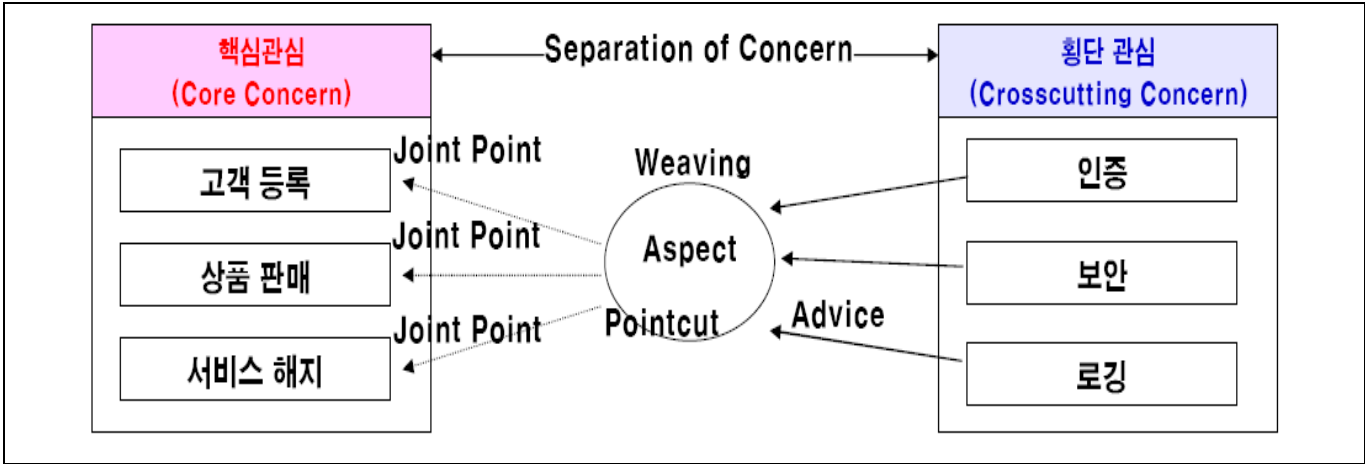
03	AOP(Aspect Oriented Programming)		
문제	AOP(Aspect Oriented Programming)의 정의, 구성, 기대효과		
도메인	소프트웨어공학	난이도	중(상/중/하)
키워드	관심사의 분리, 횡단 관심사, Weaving, Advice		
출제배경	소프트웨어공학 기본토픽 재확인		
참고문헌	ITPE 기술사회 자료		
해설자	강남평일야간반 전일 기술사(제 114회 정보관리기술사 /nikki6@hanmail.net)		

I. OOP 한계를 보완하기 위한 패러다임, AOP(Aspect Oriented Programming) 정의

- 관심사의 분리(Separation of Concern) 원칙에 기반하여, 시스템 구성을 핵심 관심사와 횡단 관심사로 분리하고 Weaving 을 통해 이를 엮어 프로그램을 구현하는 방법론

II. AOP의 구성도와 구성요소

가. AOP의 구성도



나. AOP의 구성요소

구분	구성요소	설명	사례
관심사	핵심관심사	- 단일 모듈이 가지는 주된 요구사항	- Business Logic
	횡단관심사	- 여러 개 모듈에 공통적 사용되는 부가적인 요구사항	- Logging, Security, Transaction, Error Handler
프로그래밍 요소	결합점 (Joint Point)	- Aspect를 플러그인 할 수 있는 애플리케이션 실행지점, Advice를 적용하는 지점	- 필드, 메소드의 호출
	교차점 (Point Cut)	- Advice가 어떤 Joint Point에 적용되는지를 정의 - 명시적 클래스/메소드 이름, 클래스와 메소드 이름의 패턴과 일치하는 정규식	- Weaving(결합) 지점

	Advice	- Aspect의 실제 구현체로 어플리케이션의 JointPoint에 삽입	- Before Advice, After, Around등
	직조 (Weaving)	- Advice를 핵심 로직 코드에 적용(컴파일, 클래스 로딩, 런타임 시점)	- 컴파일 시, 클래스 로딩 시, - 런타임 시 엮기
	Aspect	- 구현하고자 하는 횡단관심사의 기능	- 업무에 적용된 로깅

- Target: Advice 받는 클래스(커스텀 클래스, 써드파티 클래스)

III. AOP 기대효과

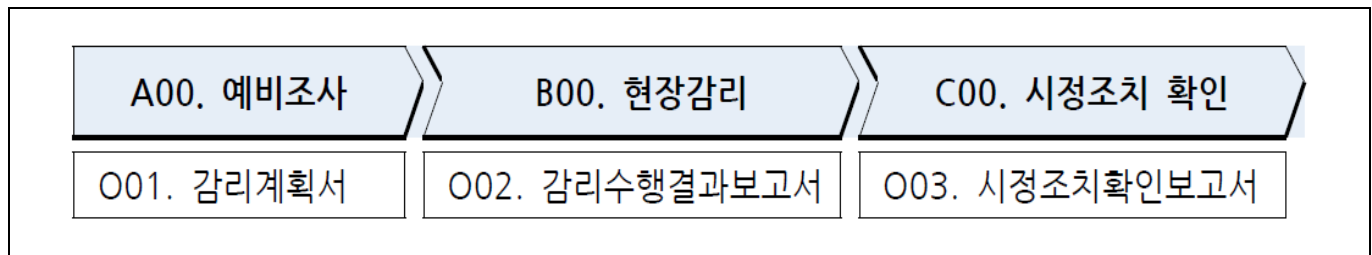
구성요소	설명	사례
고객 관점	Time-to-Market	- 필요한 부분(공통부분)만 일부 수정 시 전체적으로 적용 가능
	비용 절감	- 소스 작성 시간 및 자원 절감으로 인한 인력 누수 방지
개발자 관점	Clean Code	- 공통 관심사항을 핵심 관심사항에서 분리시켜 핵심 로직을 깔끔하게 유지
	Clean Architecture	- 각각의 모듈에 수정이 필요하면 다른 모듈의 수정 없이 해당 로직만 변경
	모듈화	- 공통 로직을 적용할 대상을 선택

- 하나의 클래스에 핵심과 횡단 관심사가 혼재되어 프로그램의 가독성 및 재 활용성에 비효율이 발생함에 따라 관심사를 분리하고 단순화하여 코드의 재활용과 유지보수의 효율성을 극대화

“끝”

04	정보시스템 감리		
문제	정보시스템 감리기준 고시(2021-4호, 2021.1.19)에 의거한 현장감리의 활동, 작업내용을 기술하고 PMO(Project Management Office)와의 차이점을 설명		
도메인	소프트웨어공학	난이도	중(상/중/하)
키워드	감리시작, 착수회의, 감리수행		
출제배경	기본토픽 이해 확인		
참고문헌	ITPE 기술사회 자료		
해설자	강남평일야간반 전일 기술사(제 114회 정보관리기술사 /nikki6@hanmail.net)		

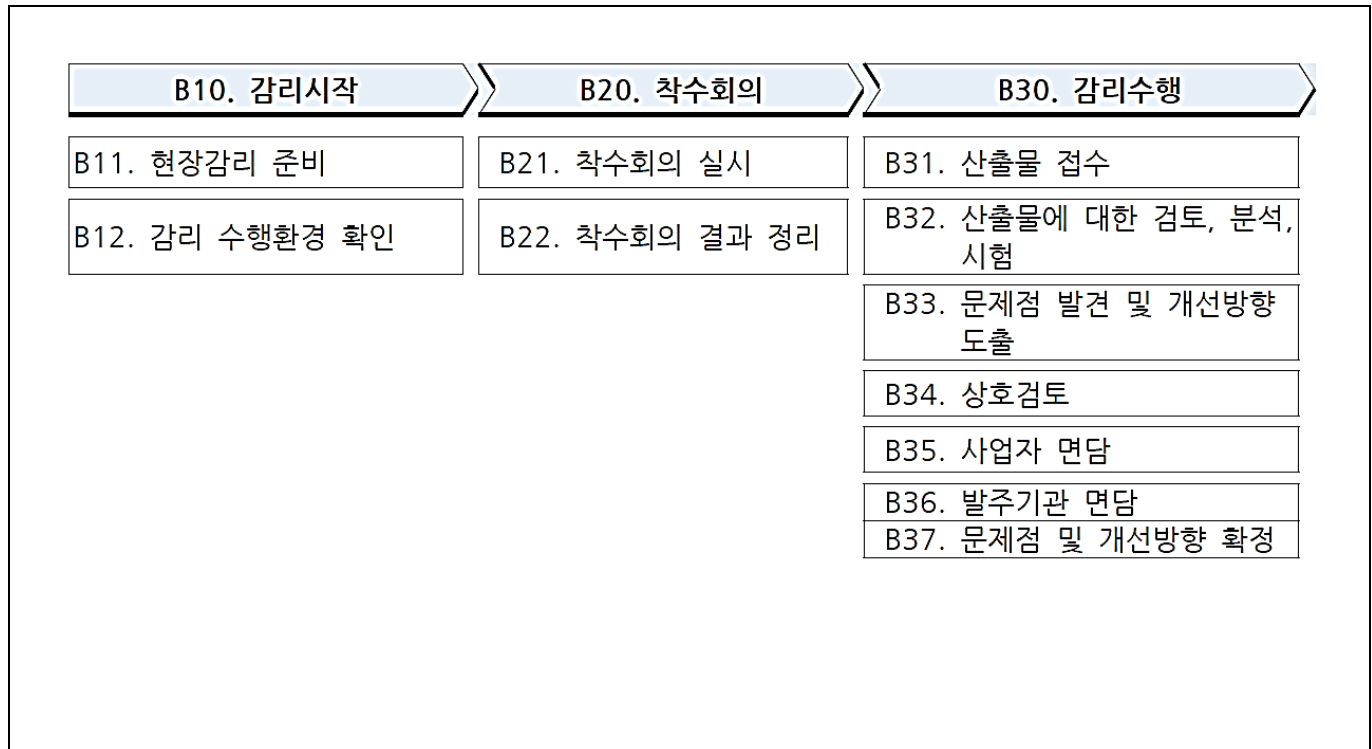
I. 공통감리의 두번째 단계, 현장감리의 개요



- 사전에 발주자 및 사업자에게 송부된 감리계획서를 바탕으로 직접 사업 현장에서 검토, 분석, 시험, 상호 검토 및 면담 등을 통해 문제점을 확인하고, 개선방향을 도출하여 감리수행결과보고서를 작성하는 활동

II. 현장감리의 활동 및 작업내용

가. 현장감리 활동



B40. 보고서(안) 작성 및 검토	B50. 종료회의	B60. 보고서 확정 및 통보
B41. 감리원별 보고서 초안 작성 및 검토	B51. 종료회의 준비	B61. 보고서(안) 이견사항 접수 및 검토
B42. 보고서 초안 취합	B52. 종료회의 실시	B62. 이견사항에 대한 처리 결과 공유
B43. 보고서(안) 검토		B63. 보고서 확정 및 통보

- 현장감리는 감리시작, 착수회의, 감리수행, 보고서 작성 및 검토, 종료회의, 감리보고서 확정 및 통보의 6가지 절차로 구성

나. 현장감리 단계별 상세 작업내용

절차	상세활동	작업 내용
B10. 감리시작	B11. 현장감리 준비	- 감리 착수회의 시간, 착수회의 참석자 확인 등 현장감리에서 필요한 각종 준비를 하는 단계
	B12. 감리 수행환경 확인	- 감리장소, 시스템 환경 등 사전 준비 상태 확인
B20. 착수회의	B21. 착수회의 실시	- 감리계획서 기준으로 감리 수행 절차 및 내용에 대해 설명
	B22. 착수회의 결과 정리	- 착수회의 중 사업자 및 발주자의 요청사항, 감리범위에 대한 변경사항, 점검항목에 대한 변경 사항이 발생한 경우 이를 감리계획서에 반영하여 현행화
B30. 감리수행	B31. 산출물 접수	- 제출된 산출물 목록을 기준으로 내용물의 누락여부를 확인
	B32. 산출물에 대한 검토, 분석, 시험	- 감리영역 별 점검항목을 기준으로 사업목적 및 요구사항에 부합하는지 분석, 구현된 기능에 대한 시험을 통하여 요구사항 반영 확인
	B33. 문제점 발견 및 개선방향 도출	- 개선 사항 발견 시, 객관적인 입장에서 이를 해결하기 위한 구체적인 개선방향을 도출
	B34. 상호검토	- 감리영역 간 일관성이 있는지 확인하기 위해 수시로 타 감리원 또는 총괄감리원과 의사소통
	B35. 사업자 면담	- 감리 수행 중 사업자와 면담을 실시하여 문제점 설명하고, 해결을 위한 구체적인 개선방향을 제시
	B36. 발주기관 면담	- 감리 수행 중 발주기관과 면담을 실시하여 문제점 설명하고, 해결을 위한 구체적인 개선방향을 제시
	B37. 문제점 및 개선방향 확정	- 전체적으로 일관성 있고 실행 가능한 수준으로 구체적이며, 사업의 목적에 맞는 문제점 및 개선방향을 확정
B40. 보고서(안) 작성 및 검토	B41. 감리원별 보고서 초안 작성 및 검토	- 발견된 문제점과 이를 개선하기 위한 개선 방향을 감리수행결과보고서 초안으로 작성
	B42. 보고서 초안 취합	- 총괄감리원이 감리원 별 감리수행 결과보고서초안과 감리계획서를 취합하고 종합의견을 작성하여 감리수행결과보고서(안)으로 함

	B43. 보고서(안) 검토	- 감리팀 내부 검토 및 사업자, 발주자 검토를 수행하여 감리수행결과보고서(안)을 확정.
B50. 종료회의	B51. 종료회의 준비	- 감리수행결과보고서(안)이 확정되고 발주자, 사업자가 검토하는 단계에 총괄감리원이 발주자와 협의를 통해 종료회의 준비
	B52. 종료회의 실시	- 감리수행결과보고서(안)과 현장감리 기간 중 해소되지 않은 이견사항에 대한 공개적이고 공식적인 토론을 통해 공감대를 형성하는 것
B60. 보고서 확정 및 통보	B61. 보고서(안) 이견사항 접수 및 검토	- 종료회의 시 논의된 이견사항 및 사업자 및 발주자의 추가적인 이견사항을 접수 받아, 해당 감리원과 총괄감리원의 협의를 거쳐 반영여부를 검토
	B62. 이견사항에 대한 처리 결과 공유	- 이견사항에 대한 반영여부 및 사유 등을 포함하여 사업자 및 발주자와 협의한
	B63. 보고서 확정 및 통보	- 감리수행결과보고서(안) 최종 확인하고 공문과 함께 발주자 및 사업자에게 통보

- 현장감리에서 나온 조치 내역에 대하여 시정 조치 수행 후 시정조치 확인을 통하여 최종 감리 진행

III. 현장 감리와 PMO의 비교

가. 제도 도입 측면의 비교

구분	현장감리	PMO
제도 도입 취지	- 사업시행자가 국가정보화사업을 수행함에 있어 발주기관의 요구대로 수행하였는지 점검하여 정보시스템의 품질을 보장	- 국가정보화사업에 대기업집단의 참여가 전면 제한됨에 따라 발주 기관과 중소기업의 사업관리 전문성 부족 문제를 보완
관점	- 제 3 자 관점 (독립적)	- 발주자 관점
투입시점	- 발주 후의 프로젝트 단계별 감리	- 발주 전과 프로젝트 중 단계별 지원
법적근거	- 전자정부법 제 57 조 1 항에 따른 의무사항	- 전자정부법 제 64 조 2 에따른 권고사항
기대효과	- 사업관리에 대한 점검 지원 - 위험요소의 대응방안 제시 - 정보시스템 및 산출물 품질 향상	- 잠재적인 프로젝트 위험 조기 식별 - 조직의 목적과 IT 연계하여 관리 - 효과적인 자원배분을 통한 비용절감

나. 수행 측면의 비교

구분	현장감리	PMO
수행역할	<ul style="list-style-type: none"> - 계약서와 실제 진행 내용의 차이에 대해서 확인하고, 원인을 파악해 적절한 조치권고 - 단계별 검증 및 테스트 - 산출물 검증 및 추적 	<ul style="list-style-type: none"> - 발주 전: RFP작성, 프로젝트 원칙수립, 초기위험 도출, 발주처 선정지원 - 발주 후: 발주지원조직운영(PMO), 사업관리계획수립, 업무표준 프로세스 수립
수행조직	- 감리법인	- 컨설팅업체, 회계법인, 대형 SI
주요산출물	<ul style="list-style-type: none"> - 감리계획서 - 감리수행결과보고서 - 시정조치확인보고서 	<ul style="list-style-type: none"> - 요구사항 정의서(RFP) - 사업자 선정 기준서 - 사업자 관리 계획서 - 아키텍처 정의서 - 영역별 관리 계획서
사업관리측면	- 직접 사업관리업무 수행하지 않음.	- 사업관리 수행

- 감리는 기술적 측면의 평가 성격이며 PMO 는 프로젝트 전 과정에 개입하는 관리적 성격이 강함

“끝”

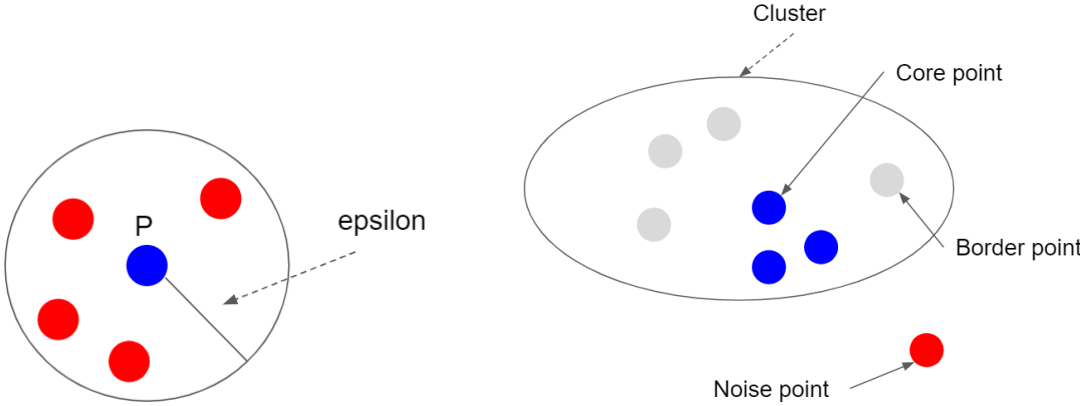
05	인공지능 학습		
문제	K-means Clustering과 DBSCAN(Density-Based Spatial Clustering of Application with Noise) 개념, 구성요소, 장/단점		
도메인	인공지능	난이도	중(상/중/하)
키워드	Clustering, 비지도학습, 군집화, 거리기반		
출제배경	인공지능 비지도학습(군집) 기본 알고리즘 이해		
참고문헌	ITPE 기술사회 자료집		
해설자	정상 기술사(제 12X회 정보관리기술사 / jeongsang_pe@naver.com)		

I. 중심기반 군집화 K-평균 알고리즘(K-Means)

구분	설명	
개념	- n개의 데이터를 K개의 군집으로 분류하기 위해 거리 기반으로 반복적으로 계산해 나가는 Clustering 알고리즘	
개념도		
구성요소	K값	- 클러스팅하여 묶을 클러스터의 개수
	Centroid	- 클러스터링을 할 중심 값
장점	짧은 계산 시간	- 간단한 알고리즘에 빠른 연산 가능
	탐색적 방법	- 탐색적 방법을 사용하여 대용량 데이터에 적합
	데이터 다양화	- 다양한 데이터의 적용 가능
단점	가중치, 거리 정의	- 적절한 가중치와 거리의 정의가 필요
	초기 클러스터링 수	- 초기 클러스터링 수 결정의 어려움
	결과 해석	- 사전에 주어진 목적이 없어 결과 해석이 어려움

- k-means의 노이즈(outlier)에 민감한 단점을 보완한 K-medoid, DBSCAN 알고리즘이 있음

II. 밀도기반 군집화 밀도기반 클러스터링(DBSCAN)

구분	설명	
개념	- 임의의 클러스터 중심을 이동시키며 중심으로부터 정해진 반경 거리 내에 최소 데이터 포인트 개수를 확인하며 밀도 기반으로 군집화를 수행하는 알고리즘	
개념도	 <p>- Epsilon 내에 중심점 P기준으로 군집판단기준(mPTS)을 계산하여 군집 판단</p>	
구성요소	- Core Point	- 거리 ϵ (Epsilon)이내에 데이터가 m 개 이상 존재하여 한의 군집으로 인정되는 데이터 집합
	- Border Point	- 군집의 중심이 되는 core point는 되지 못하지만, core point로 하는 군집에는 속하는 데이터
	- Connected	- core point 와 core point가 반경내에 겹칠 경우 연결되어 있다고 보고 하나의 군집으로 정의
	- Noise Point	- 어떤 점의 중심으로도 조건을 만족시키지 못하는 데이터
장점	클러스터 개수 미정의	- 클러스터 개수 정의 없이 클러스터 진행
	밀도 기반 클러스터링	- 데이터들의 밀도를 계산해서 클러스터링 진행
	비선형 경계 클러스터링	- 비선형 경계에 있는 클러스터링 가능
단점	데이터 순서 영향	- 데이터 사용 순서에 따라 클러스터링 차이가 있음
	차원의 저주	- 고차원 데이터에 대한 적절한 엡실론을 찾기 어려움
	밀도 높은 탐색 저하	- 고차원에서 밀도가 높은 지역을 찾기 어려움

- Epsilon(Eps), minPts 초기 파라미터 값에 따라서 군집 및 속도 성능이 크게 차이남

“끝”

06	트랜잭션		
문제	데이터베이스 트랜잭션(Transaction)의 특징		
도메인	데이터베이스	난이도	하(상/중/하)
키워드	논리적 작업단위, 원자성, 일관성, 격리성, 영속성,		
출제배경	데이터베이스 트랜잭션에 대한 개념 숙지 확인		
참고문헌	ITPE 기술사회 자료집		
해설자	정상 기술사(제 12X회 정보관리기술사 / jeongsang_pe@naver.com)		

I. 데이터베이스 처리 논리적 작업단위, 트랜잭션의 개요

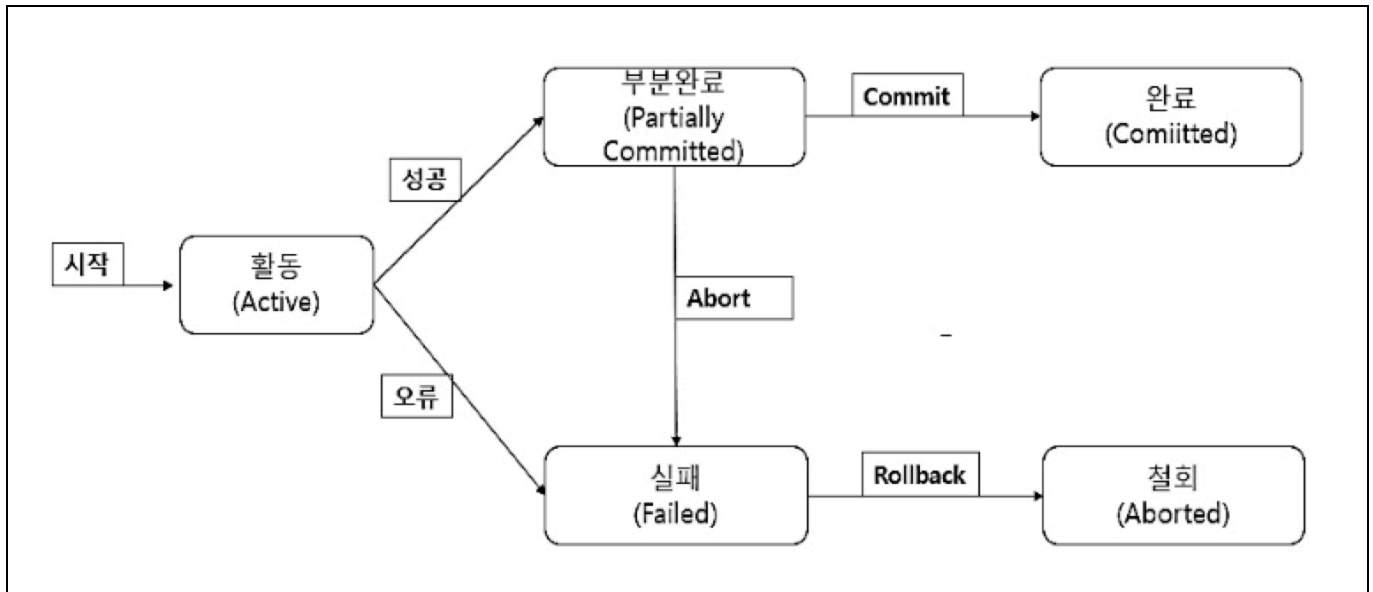
- 데이터의 정확한 일관성과 무결성을 보장하기 위해 완전히 종료해야 하는 데이터베이스 처리의 논리적 작업 단위(Logical Unit of Work)

II. 데이터베이스 트랜잭션 특징 설명

특징	개념도	설명
원자성 (Atomicity)	<p>Transaction</p> <p>연산1 연산n</p> <p>Commit</p> <p>Rollback</p>	<ul style="list-style-type: none"> - 분해가 불가능한 수행 단위로 완전히 수행, 수행되지 않음 - 구현기법: All or Nothing, Commit or Rollback - 기능: 회복
일관성 (Consistency)	<p>Database</p> <p>일관된 상태</p> <p>Committed Transaction</p> <p>일관된 상태</p>	<ul style="list-style-type: none"> - 트랜잭션이 성공적으로 완료되면 언제나 모순이 없는 상태 유지 - 무결성 제약조건, 사용자가 요구하는 논리적 요건의 충족 - 구현기법: 도메인 무결성, 릴레이션 무결성 - 기능: 무결성 제약조건, 동시성 제어
격리성 (Isolation)	<p>Transaction 의한 갱신 중</p> <p>Data</p> <p>접근 제어</p>	<ul style="list-style-type: none"> - 트랜잭션 실행 중에는 다른 트랜잭션 접근 불가 - 구현기법: isolation level - 기능: 동시성 제어
영속성 (Durability)	<p>Commit 정보 기록</p> <p>DB</p> <p>장애 시 회복</p>	<ul style="list-style-type: none"> - 트랜잭션이 실행이 성공적으로 완료하면 그 결과는 영속적임 - 구현기법: Archive, 로그, Redo/Undo 기반 회복 - 기능: 회복

- 데이터베이스는 4가지 특징으로 기본 동작 진행

III. 데이터베이스 트랜잭션 상태 전이도



- 다중 프로그램 환경에서 트랜잭션의 ACID 조건을 만족 시키기 위해서는 트랜잭션이 동시에 수행될 경우 발생하는 문제점을 해결 위한 동시성 제어가 필요

“끝”

07	결합도		
문제	소프트웨어 결합도(Coupling)의 종류		
도메인	소프트웨어 공학	난이도	하(상/중/하)
키워드	자료, 스템프, 제어, 외부, 공통, 내용		
출제배경	소프트웨어 모듈화에 대한 이해 확인		
참고문헌	ITPE 기술사회 자료집		
해설자	정상 기술사(제 12X회 정보관리기술사 / jeongsang_pe@naver.com)		

I. 외부 모듈과의 연관도, 소프트웨어 결합도의 개념

- 모듈내부가 아닌 외부의 모듈과의 연관도(모듈 간의 상호연관도)
- 소프트웨어 구조에서 모듈 간의 관련성을 측정하는 척도

II. 소프트웨어 결합도의 종류 상세설명

종류	설명	결합도
자료	- 모듈들이 간단히 변수를 파라미터로 교환	<div>↑</div> <div>낮음</div> <div>높음</div>
스탬프	- 모듈 사이에 자료구조 교환	
제어	- 제어용 신호, 정보를 주고받아 제어하는 경우	
외부	- 모듈들이 소프트웨어의 외부환경과 연관되는 경우	
공통	- 많은 모듈들이 전역변수를 참조할 때 발생	
내용	- 한 모듈이 다른 모듈의 내부 자료나 제어정보를 사용	

- 결합도의 종류에 따라 결합도 강도가 정해지고, 모듈 간 영향 강도가 달라짐

III. 소프트웨어 모듈의 응집도

단계	설명	응집도
우연적	- 아무 관련성 없는 작업을 한 모듈에서 모음	<div>↓</div> <div>낮음</div> <div>높음</div>
논리적	- 유사한 성격의 작업들을 모음	
시간적	- 같은 시간대에 처리되어야 하는 것들을 모음	
절차적	- 모듈 진행 요소들이 서로 관계되어지고 순서대로 진행	
통신적	- 동일한 입/출력 자료를 이용하여 서로 다른 기능을 수행하는 기능	
순차적	- 작업의 결과가 다른 모듈의 입력자료로 사용	
기능적	- 하나의 기능만 수행하는 모듈	

- 소프트웨어 모듈화에는 결합도 외 응집도의 요소도 고려하여야 함

“끝”

08	화이트박스 테스트(White Box Test)와 블랙박스 테스트(Black Box Test)		
문제	화이트박스 테스트(White Box Test)와 블랙박스 테스트(Black Box Test)		
도메인	소프트웨어공학	난이도	하(상/중/하)
키워드	화제루, 불동경의상 유분페오		
출제배경	빈출 토픽으로 SW 테스트 중요도 증가에 따른 기본 지식 확인을 위한 출제		
참고문헌	ITPE기술사회 자료		
해설자	서경석 기술사(제119회 정보관리기술사 / akslemlf@naver.com)		

I. 개발자 관점의 내부구조 테스트 방식, 화이트박스 테스트의 개요

가. 화이트박스 테스트(WhiteBox Text)의 개념

- 개발자가 소프트웨어 또는 컴포넌트 등의 내부 Logic에 대한 테스트를 수행하기 위하여 설계 단계에서 요구된 사항을 확인하여 테스트 하는 유형

나. 화이트박스 테스트 기법

기법	설명	사례															
제어구조 시험 (Control Structure Testing)	- McCabe 에 의해 제안된 대표적 White Box Test 기법 - 프로그램의 처리 흐름을 제어하는 방법 및 수행 제어를 위해 사용되는 문장의 구조	- 순차형(순차 구조, Sequence 형) - 선택형(분기구조, If Then Else 형) - 반복형(반복구조, Do While 형)															
루프 시험 (Loop Testing)	- 프로그램 루프 구조에 국한해서 실시하는 기법 - 루프 시험의 대상 결함 : 초기화 결함, 인덱싱 및 증가의 결함, 루프의 경계선에서 나타나는 경계 오류 - 루프의 유형 : 단순루프, 중첩루프, 연결루프, 비구조적 루프	- for, while 등 - go to 등															
구문 커버리지 (Statement Coverage)	- 프로그램을 구성하는 모든 구문들이 최소한 한번은 실행될 수 있는 입력 데이터를 테스트 데이터로 선정 - 프로그램 내 모든 구문의 테스트를 보장	- 소스코드 if(a>0 or b>0) call join - 테스트케이스 a = 3, b = 4															
결정 커버리지 (Decision Coverage)	- 프로그램 내의 전체 결정문이 적어도 한번은 참과 거짓의 결과를 수행하는 테스트 케이스 생성 - 모든 분기문을 테스트하는 방법	<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>A OR B</th></tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	A	B	A OR B	1	1	1	1	0	1	0	1	1	0	0	0
A	B	A OR B															
1	1	1															
1	0	1															
0	1	1															
0	0	0															
조건 커버리지 (Condition Coverage)	- 결정 명령문 내의 각 조건이 적어도 한 번은 참과 거짓의 결과가 되도록 수행하는 테스트 케이스 - 모든 조건을 커버하는 방법	<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>A OR B</th></tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	A	B	A OR B	1	1	1	1	0	1	0	1	1	0	0	0
A	B	A OR B															
1	1	1															
1	0	1															
0	1	1															
0	0	0															

조건/결정 커버리지 (Condition/Decision Coverage)	- 전체 조건식뿐만 아니라 개별 조건식도 참 한번, 거짓 한번 결과가 되도록 수행하는 테스트 케이스	<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>A OR B</th></tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	A	B	A OR B	1	1	1	1	0	1	0	1	1	0	0	0
A	B	A OR B															
1	1	1															
1	0	1															
0	1	1															
0	0	0															
변경조건/결정 커버리지 (MC/DC)	- 각 개별 조건식이 다른 개별 조건식에 영향을 받지 않고 전체 조건식에 독립적으로 영향을 주도록 하는 테스트 케이스	<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>A OR B</th></tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	A	B	A OR B	1	1	1	1	0	1	0	1	1	0	0	0
A	B	A OR B															
1	1	1															
1	0	1															
0	1	1															
0	0	0															
다중조건/결정 커버리지(Multiple Condition Coverage)	- 결정 포인트 내에 있는 모든 개별식 조건의 모든 조합을 고려한 커버리지	<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>A OR B</th></tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	A	B	A OR B	1	1	1	1	0	1	0	1	1	0	0	0
A	B	A OR B															
1	1	1															
1	0	1															
0	1	1															
0	0	0															

- 구문, 로직 기반 화이트박스 테스트 진행과 병행하여 요구 명세서 적합 판단 위한 블랙박스 테스트 필요

II. 요구 명세서 기반 기능 위주의 테스트 방식, 블랙박스 테스트(Black Box Test) 개요

가. 블랙박스 테스트 개념

- 테스트 대상의 내부 구조를 참조하지 않고 테스트 베이스 개발자, 테스터, 사용자 경험 바탕으로 기능적, 비기능적 테스트 케이스 도출하고 선택하는 기법

나. 블랙박스 테스트의 기법 설명

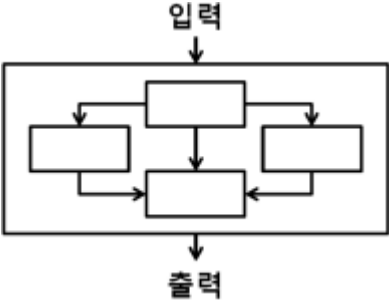
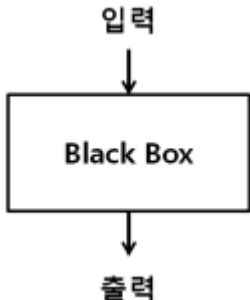
기법	개념도	상세 설명
동등 클래스 분할 기법 (Equivalence Class Partitioning)	<p>- 등가 분할 된 대표 값을 이용 테스트 케이스 도출</p>	<ul style="list-style-type: none"> - 프로그램의 입력 도메인을 등가영역들로 분할 후 각 영역별로 대표되는 값들을 선정하여 테스트케이스를 설계하는 방법 - 예) 입력데이터 x값이 0 ~ 100 사이여야 한다면 TC를 ($x < 0$), ($x = 50$), ($x > 100$)으로 분할하여 적용
경계값 분석 (Boundary Value Analysis)	<p>- 결함은 경계 값 근처에서 많이 발생 이용</p>	<ul style="list-style-type: none"> - 입력 영역의 분할 클래스의 경계값으로 테스트 케이스를 설계하는 방법 - 등가 분할 기법의 확장, 등가분할 된 경계의 유효한 값, 경계의 가장 가까운 유효하지 않는 값 선택 테스트 진행 - 예) x값이 0~100사이여야 한다면 TC를 ($x=0$), ($x=100$), ($x=-0.01$), ($x=100.1$)로 정의

의사결정 테이블 테스트 (Decision Table Testing)	<table><tr><th>상태</th><th>Case 1</th><th>Case 2</th><th>Case 3</th><th>Case 4</th></tr><tr><td colspan="5">테스트조건</td></tr><tr><td>유효한 지폐</td><td>N</td><td>Y</td><td>-</td><td>-</td></tr><tr><td>유효한 카드</td><td>-</td><td>-</td><td>N</td><td>Y</td></tr><tr><td>유효한 암호</td><td>-</td><td>-</td><td>Y</td><td>N</td></tr><tr><td colspan="5">예상결과</td></tr><tr><td>지폐 거부</td><td>Y</td><td>N</td><td>N</td><td>N</td></tr><tr><td>카드 거부</td><td>N</td><td>N</td><td>Y</td><td>Y</td></tr><tr><td>허용</td><td>N</td><td>N</td><td>N</td><td>N</td></tr></table> <p>- 조건과 결과를 참/거짓으로 표현</p>	상태	Case 1	Case 2	Case 3	Case 4	테스트조건					유효한 지폐	N	Y	-	-	유효한 카드	-	-	N	Y	유효한 암호	-	-	Y	N	예상결과					지폐 거부	Y	N	N	N	카드 거부	N	N	Y	Y	허용	N	N	N	N	<ul style="list-style-type: none">- 주요한 의사결정 요소들을 표(결정테이블)로 만들고, 요소들간의 결합에 의한 테스트 케이스 설계- 각 의사결정 요소들의 조합을 통해 다양한 형태의 테스트 시나리오를 도출																																			
상태	Case 1	Case 2	Case 3	Case 4																																																																														
테스트조건																																																																																		
유효한 지폐	N	Y	-	-																																																																														
유효한 카드	-	-	N	Y																																																																														
유효한 암호	-	-	Y	N																																																																														
예상결과																																																																																		
지폐 거부	Y	N	N	N																																																																														
카드 거부	N	N	Y	Y																																																																														
허용	N	N	N	N																																																																														
상태 전이 테스팅 (State transition Testing)	<table><tr><th>START STATE</th><td>OFF</td><td>ON</td><td>OFF</td></tr><tr><th>INPUT</th><td>SWITCH ON</td><td>SWITCH OFF</td><td>SWITCH ON</td></tr><tr><th>OUTPUT</th><td>LIGHT ON</td><td>LIGHT OFF</td><td>SWITCH OFF</td></tr><tr><th>FINISH STATE</th><td>ON</td><td>OFF</td><td>OFF</td></tr></table> <p>Test 1 Test 2 Test 3</p>	START STATE	OFF	ON	OFF	INPUT	SWITCH ON	SWITCH OFF	SWITCH ON	OUTPUT	LIGHT ON	LIGHT OFF	SWITCH OFF	FINISH STATE	ON	OFF	OFF	<ul style="list-style-type: none">- 상태전이 다이어그램 통해 이벤트, 액션, 활동, 상태 변화로 발생하는 관계, 동작 파악하여 테스트- 임베디드SW 테스트시 적용																																																																
START STATE	OFF	ON	OFF																																																																															
INPUT	SWITCH ON	SWITCH OFF	SWITCH ON																																																																															
OUTPUT	LIGHT ON	LIGHT OFF	SWITCH OFF																																																																															
FINISH STATE	ON	OFF	OFF																																																																															
유스케이스 테스팅 (Use Case Testing)	<p>- usecase명세서 이용 테스트 케이스 설계</p>	<ul style="list-style-type: none">- 유즈케이스를 통해 도출되는 비즈니스 시나리오(기본 흐름, 대체 흐름)를 기반으로 테스트를 명세화하여 테스트- 컴포넌트/단위 레벨 유즈케이스 테스트- 시스템 레벨 유즈케이스 테스트																																																																																
분류 트리 기법 (Classification Tree Method)		<ul style="list-style-type: none">- SW의 일부 또는 전체를 트리 구조로 분석 및 표현하여 테스트 케이스를 설계하는 기법- 트리구조를 시각화 하여 테스트 케이스를 설계하므로 불필요한 중복 및 누락을 회피																																																																																
페어와이즈 테스팅 (Paireise Testing)	<table><tr><th colspan="5">8 Test Cases</th></tr><tr><th>Test ID</th><th>X</th><th>Y</th><th>Z</th><th></th></tr><tr><td>T1</td><td>A</td><td>C</td><td>E</td><td></td></tr><tr><td>T2</td><td>A</td><td>D</td><td>E</td><td></td></tr><tr><td>T3</td><td>A</td><td>C</td><td>F</td><td></td></tr><tr><td>T4</td><td>A</td><td>D</td><td>F</td><td></td></tr><tr><td>T5</td><td>B</td><td>C</td><td>E</td><td></td></tr><tr><td>T6</td><td>B</td><td>D</td><td>E</td><td></td></tr><tr><td>T7</td><td>B</td><td>C</td><td>F</td><td></td></tr><tr><td>T8</td><td>B</td><td>D</td><td>F</td><td></td></tr></table> <table><tr><th colspan="5">Pairwise testing 4 Test Cases</th></tr><tr><th>Test ID</th><th>X</th><th>Y</th><th>Z</th><th></th></tr><tr><td>T1</td><td>A</td><td>C</td><td>E</td><td></td></tr><tr><td>T4</td><td>A</td><td>D</td><td>F</td><td></td></tr><tr><td>T6</td><td>B</td><td>D</td><td>E</td><td></td></tr><tr><td>T7</td><td>B</td><td>C</td><td>F</td><td></td></tr></table>	8 Test Cases					Test ID	X	Y	Z		T1	A	C	E		T2	A	D	E		T3	A	C	F		T4	A	D	F		T5	B	C	E		T6	B	D	E		T7	B	C	F		T8	B	D	F		Pairwise testing 4 Test Cases					Test ID	X	Y	Z		T1	A	C	E		T4	A	D	F		T6	B	D	E		T7	B	C	F		<ul style="list-style-type: none">- 대부분 결합이 2 개의 요소(Pair)의 상호작용에 기인한다는 것에 착안하여, 각 값들이 다른 파라미터의 값과 최소한번씩은 조합을 이루도록 구성하는 테스트 기법- 모든 조합을 포함하지 않으므로 결함을 찾지 못하는 경우 발생 가능
8 Test Cases																																																																																		
Test ID	X	Y	Z																																																																															
T1	A	C	E																																																																															
T2	A	D	E																																																																															
T3	A	C	F																																																																															
T4	A	D	F																																																																															
T5	B	C	E																																																																															
T6	B	D	E																																																																															
T7	B	C	F																																																																															
T8	B	D	F																																																																															
Pairwise testing 4 Test Cases																																																																																		
Test ID	X	Y	Z																																																																															
T1	A	C	E																																																																															
T4	A	D	F																																																																															
T6	B	D	E																																																																															
T7	B	C	F																																																																															
원인-결과 그래프 (Cause Effect Graph)	<p>Fishbone Diagram</p>	<ul style="list-style-type: none">- 입력 데이터간 관계가 출력에 영향을 미치는 상황을 체계적으로 분석하여 테스트 케이스 설계 및 테스트- 원인,결과에 근거한 테스트 케이스 생성하며 시스템 외부 동작만 고려- 원인: 시스템 내부 입력 상태- 결과: 시스템 변환 또는 원인 조합으로 인한 출력 상태																																																																																

오류예측 기법 (Error Guessing)	예) 입력값 없이 Return 친다. 문법에 어긋난 입력을 시험한다 등	<ul style="list-style-type: none"> - 각 시험 기법들이 놓치기 쉬운 오류들을 감각과 경험으로 찾아 검증 - Ad-hoc Testing이라고도 하며 직관과 경험에 의한 특정 형태의 결함 예측 및 해당 결함을 드러내는 테스트 케이스 설계 기법.
-----------------------------------	--	--

- 화이트박스 테스트와 블랙박스 테스트의 중첩 부분 테스트를 위한 그레이박스 테스트도 활용 가능

III. 화이트박스 테스트와 블랙박스 테스트의 비교

구 분	화이트박스 테스트(구조기반)	블랙박스 테스트(명세기반)
개념도		
개념	- 컴포넌트 혹은 소프트웨어의 내부 구조 분석에 바탕을 두고 테스트케이스를 도출하여 테스트 하는 기법	- 프로그램 내부 구조를 고려하지 않고, 요구 사항 명세서나 사양서에서 테스트 케이스를 추출하여 테스트하는 기법
특징	- 내부로직 위주 설계, 알고리즘 위주, 구조 기반	- 기능/데이터 위주, 명세 기반
관점	- 사용자 관점	- 개발자 관점
대상	- Loop, Decision 결함, 비 수행 구문	- 시작/종료/인터페이스 결함
장점	- 내부 로직에 대한 테스트를 수행함으로써 상세한 테스트 가능	- 시스템의 기능과 명세로 테스트 케이스를 도출하여 테스트케이스가 명확
단점	- 테스트케이스 도출 한계	- 내부 로직 테스트가 없어 테스트 범위가 구조기반보다 제한적
기법	- 구문 커버리지, 결정 커버리지, 조건/결정 커버리지, 루프테스트, 제어구조 시험	- 동등분할, 경계값분석, 의사결정테이블, 상태전이, Pairwise 분석 오류 예측, 원인결과 그래프
활용	- 단위 테스트 위주, 알파 테스트	- 대부분의 테스트에 적용, 베타 테스트

“끝”

09	객체지향 프로그래밍의 캡슐화(Encapsulation)		
문제	객체지향 프로그래밍의 캡슐화(Encapsulation)		
도메인	소프트웨어공학	난이도	하(상/중/하)
키워드	속성, method, public, private		
출제배경	객체지향 프로그램 설계 기본 원리에 대한 기초 지식 확인을 위한 출제		
참고문헌	ITPE기술사회 자료		
해설자	서경석 기술사(제119회 정보관리기술사 / akslemlf@naver.com)		

I. 정보 은닉을 위한 설계 원리, 객체지향 프로그래밍의 캡슐화의 개요

가. 캡슐화(Encapsulation)의 정의

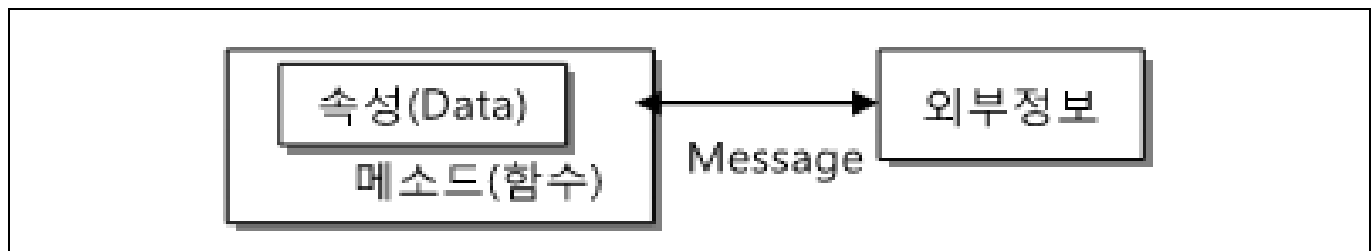
- 객체의 속성(Data Fields)과 행위(메소드, Methods)를 하나로 묶고, 실제 구현 내용 일부를 일부에 감추어 은닉하는 객체지향의 특성

나. 캡슐화의 필요성

접근 방지	외부에서 특정 객체 데이터 및 함수의 직접 접근을 방지 및 변경 방지 통한 유지보수나 확장 시 오류 범위 최소화
독립성 확보	객체내 정보 손상, 오용 방지하고, 데이터 변경 시 다른 객체에 영향 최소화
재사용성 증가	객체 모듈화 통한 새로운 시스템 내 모듈처럼 사용 가능

II. 캡슐화의 개념도와 접근제어자

가. 캡슐화의 개념도



나. 캡슐화의 접근제어자

구분	세부	설명
Class	Default	- 동일 패키지의 클래스(class)에만 인스턴스(객체)를 생성 가능
	Public	- 다른 패키지에서 인스턴스(객체)를 생성 가능
Method	Private	- 동일한 클래스 내 접근 가능, this 사용 시 외부 접근 불가, 상속 불가
	Default	- 접근 제어자가 없는 형태로 동일 패키지 내에서만 접근 가능
	Protected	- 동일 패키지 내 사용 가능, 다른 패키지인 경우 상속받은 클래스 접근 가능
	Public	- 모든 객체 접근 가능

- 자신 내부 모듈 은닉 및 다른 모듈 내부 작업 직접 개입 불가하도록 적절한 접근 제어자 통한 설계 필요

III. 캡슐화의 구현 예시

```
public class member {

    private String id;           "member 클래스 내 모든 변수는 private으로 선언하여
    private String pw;           "member 클래스 내부에서만 접근 가능
    private int age;

    //getter                     "각각의 값 사용을 위해 get 사용
    public String getId() {
        return id;
    }
    public String getPw() {
        return pw;
    }
    public int getAge() {
        return age;
    }

    //setter                     "각각의 값 설정을 위해 set 사용
    public void setId(String id) {
        this.id = id;
    }
    public void setPw(String pw) {
        this.pw = pw;
    }
    public void setAge(int age) {
        this.age = age;
    }
}
```

“끝”

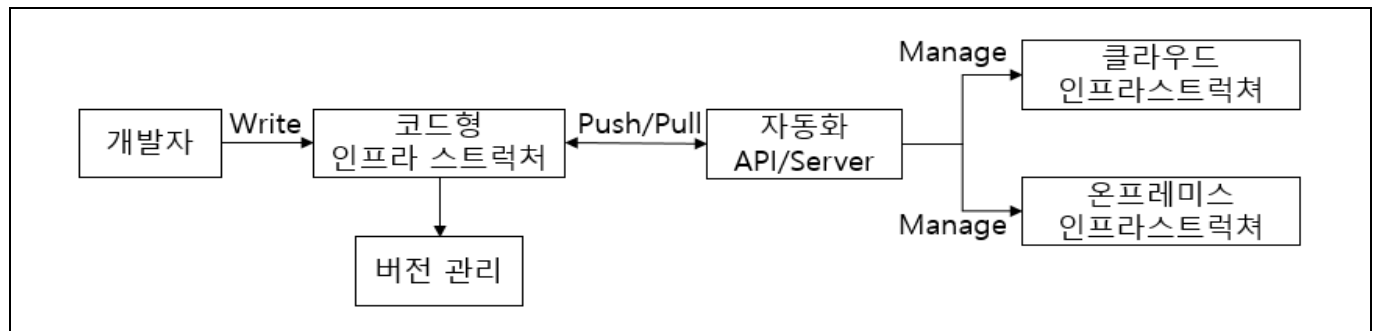
10	코드형 인프라스트럭처		
문제	코드형 인프라스트럭처		
도메인	디지털서비스	난이도	하(상/중/하)
키워드	Bootstrap, Configuration, Orchestration		
출제배경	IaC 활용도 증가 추세에 따른 연관 지식 확인을 위한 출제		
참고문헌	ITPE기술사회 자료		
해설자	서경석 기술사(제119회 정보관리기술사 / akslemlf@naver.com)		

I. 코드로 인프라를 제어하는, 코드형 인프라스트럭처의 정의

- 인프라 구성 및 제어를 위해 소프트웨어를 프로그래밍하는 것처럼 스크립트를 사용하여 인프라 오케스트레이션을 통한 인프라 제어 기술

II. 코드형 인프라스트럭처의 Workflow 및 구성 기술

가. 코드형 인프라스트럭처의 Workflow



- 인프라를 코드로 통제하며 시스템의 가독성이 높아지고, TDD처럼 코드화된 인프라 테스트하며 먹등성 제공

나. 코드형 인프라스트럭처의 구성 기술

구분	구성 기술	설명
Bootstrap	Vagrant	- 가상 머신 리소스 및 초기 스크립트 수행
	Docker	- 어플리케이션 구동 필요 설정 및 파일 관리
	CLI	- 기 구성 클라우드 인프라를 코드 형태로 Export
Configuration	Chef	- 레시피, playbook 등을 통한 먹등성 제공
	Puppet	- agent 방식과 SSH 기반 non-agent 방식 제공
	Ansible	- 운영 환경 필요 설정 관련 사항 파일로 관리
Orchestration	Mcollective	- 각 노드의 요청, 응답 관리
	SaltStack	- 마스터에서 ZeroMQ를 사용, 다수 노드 관리
	Jenkins+Fabric	- Fabric 실행 결과를 Jenkins에 연동 관리 가능

- 오류 발생시 빠르게 수정하여 개선된 코드를 통해 지속적으로 인프라 관리 가능

III. 코드형 인프라스트럭처의 장점

측면	세부	기대 효과
비율/효율성	작업 시간 단축	- 반복 작업에 대한 자동화 제공
	재작업 감소	- 인프라 구성 이전 테스트 수행 가능
에러 관리	휴먼 에러 감소	- 맥등성 관점 잦은 실수 방지
	빠른 복원 가능	- 구성 실패 시 이전 버전 인프라로 원복 가능

- 코드형 인프라스트럭처를 통해 개발환경 및 배포 프로세스를 자동화로 DevOps의 기술 측면 요구사항 만족
- “끝”

11	정보보호 제품 신속 확인 제도		
문제	정보보호 제품 신속 확인 제도		
도메인	정보보안	난이도	중(상/중/하)
키워드	신속확인 대상 검토, 신속확인, 사후관리		
출제배경	기존 인증제도 평가기준이 없는 신기술의 공공시장 진입 제도 확인		
참고문헌	정보보호산업 진흥 포털(https://www.ksecurity.or.kr)		
해설자	이상용 기술사(제 124회 정보관리기술사 / orangeday77@gmail.com)		

I. 사이버 위협 신속 대응, 정보보호제품 신속확인 제도의 개요

가. 정보보호제품 신속확인 제도의 정의

- 신기술, 융·복합 제품에 대해 최소한의 절차와 인증 기준으로 보안 제품을 평가한 뒤 평가 기준이 마련될 때까지 공공부문에 제품을 적용할 수 있도록 하는 제도

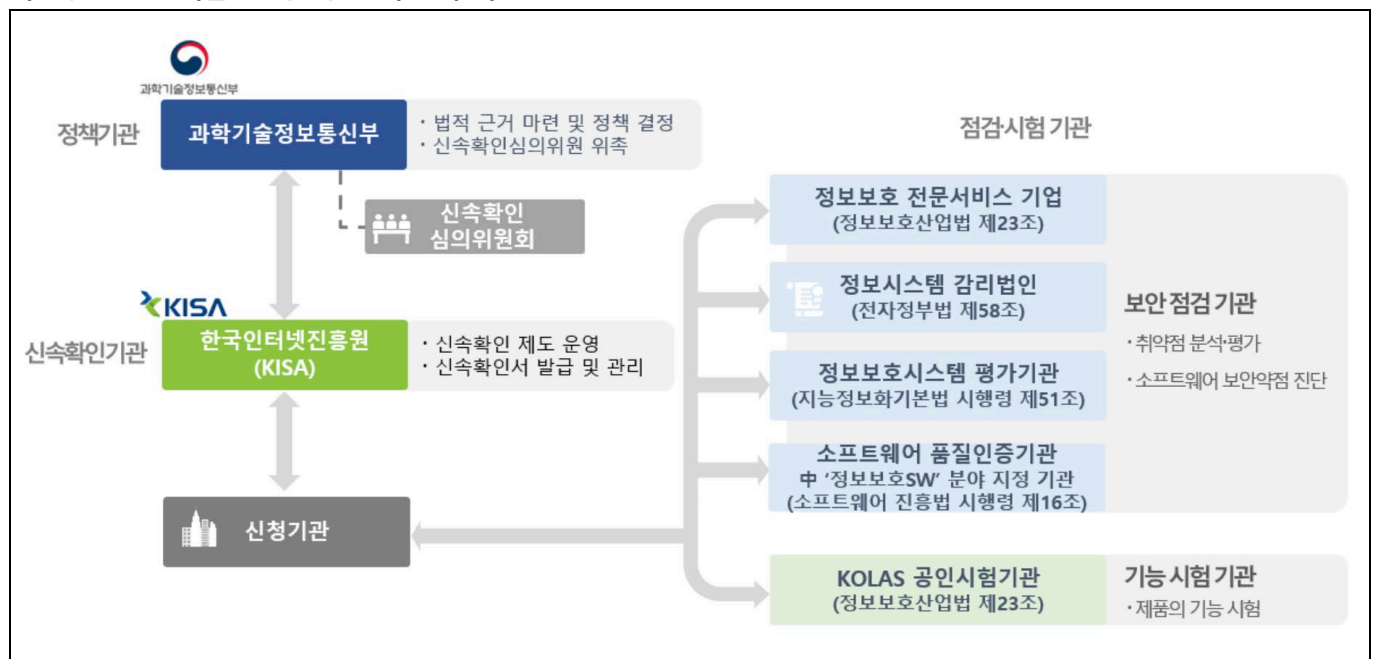
나. 정보보호제품 신속확인 목적 및 법적 근거

신기술 신속 진입	기존 인증제도에서 평가기준이 없는 신기술 및 융·복합제품의 공공시장 진입
국가 역량 강화	정부·공공기관의 신규 보안 위협 대응 등 국가 사이버 위협 대응역량 강화
법적 근거	과학기술정보통신부 '정보보호 시스템 평가·인증 등에 관한 고시'

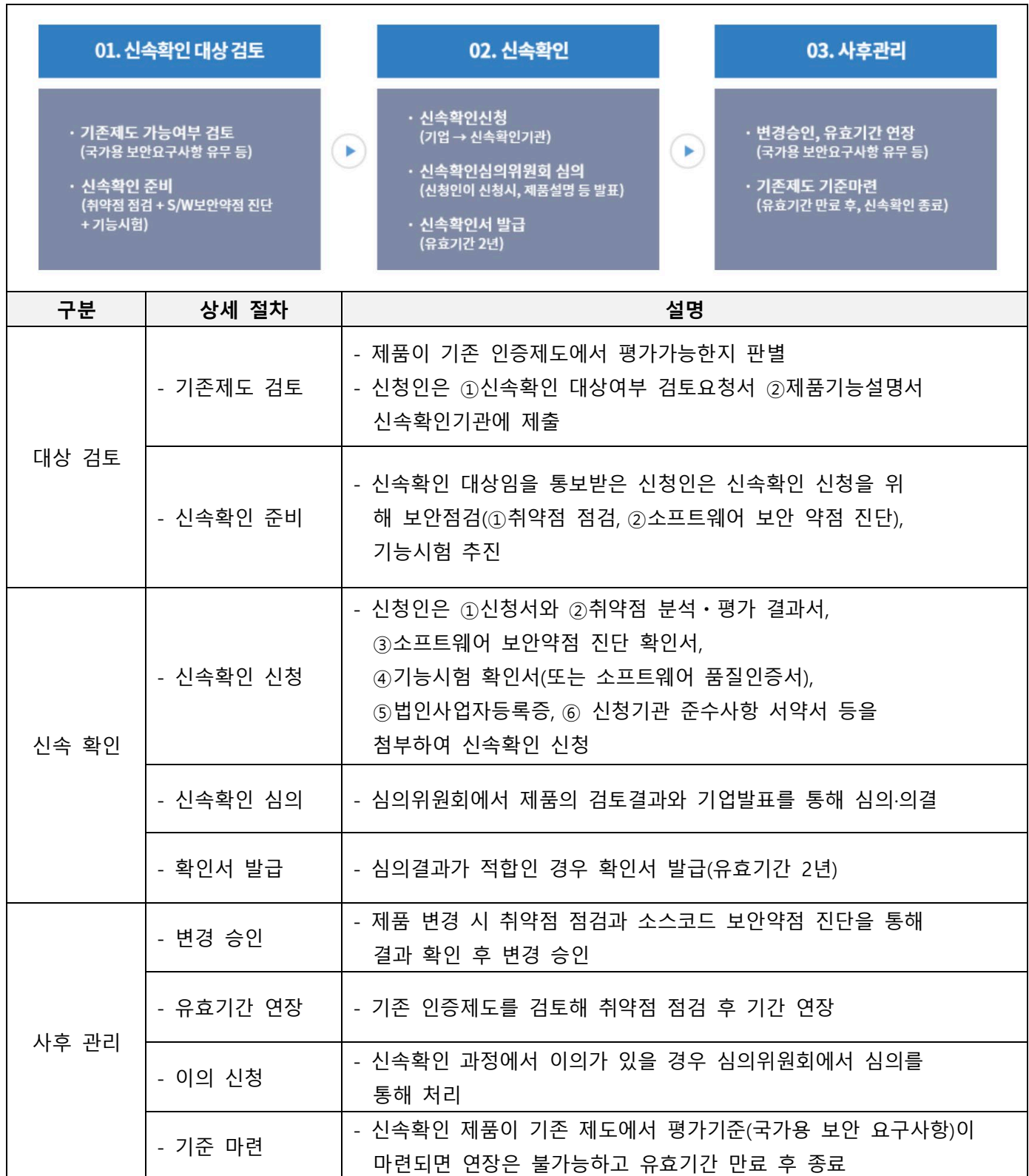
- CC인증의 국가정보원 국가용보안요구사항에서 규정한 20여개 제품의 까다로운 인증 등의 불편을 보완하기 위해 신기술, 융·복합제품은 인증 체계 마련

II. 정보보호 제품 신속 확인 제도 주체와 절차

가. 정보보호 제품 신속 확인 제도 주체



나. 정보보호 제품 신속 확인 제도 절차



- 신속확인서를 발급받은 정보보호제품은 '나', '다' 그룹에 편성된 공공분야에 보안적합성 검증 없이 도입 가능

III. 보안 적합성 검증제도 개편과 정보보호제품 신속확인제도 도입 요건

구분	도입 요건	검증필제품목록	도입 요건 미 지정	취약점
가	기존과 동일 (국내CC, 보안기능확인서, 성능평가(3종), 검증필 암호 모듈)	검증필제품목록	검증필제품목록 외 제품 도입시 보안적합성 검증 신청	국정원 취약점 대응체계 준수
나	기존제도 '권고' 또는 국제 CC 성능평가 전 제품	국제CC, 성능평가 전 제품, 신속확인제도	국제CC, 성능평가 전 제품, 신속확인제도가 아닌 제품은 보안적합성 검증 신청	
다	자율	자율	도입기관 필요 시	

- 보안적합성검증 제도란 국가 정보통신망의 보안수준을 제고하기 위해 국가·공공기관이 도입하는 정보보호시스템·네트워크 장비 등 보안기능이 탑재된 IT 제품의 안전성을 검증하는 제도

번호	신속확인제도 도입 요건
1	- '가'그룹 기관의 경우 국가정보원의 '보안적합성 검증' 받으면 신속확인 제품 도입 가능
2	- 세부사항은 국가사이버안보센터(ncsc.go.kr) 홈페이지에 게시된 ' 공공분야 IT보안제품 新 보안적합성 검증체계 개선' 참고

“끝”

12	다중접속		
문제	비직교 다중접속(NOMA, Non-Orthogonal Multiple Access)		
도메인	네트워크	난이도	중(상/중/하)
키워드	비직교성, 다중접속, 주파수/시간/코드 공유, 주파수 향상 기술, 5G 셀 용량 증대, 순차적 간섭 제거(Successive Interference Cancellation: SIC)		
출제배경	5G/6G 핵심기술 이해 확인		
참고문헌	주간기술동향(5G and Beyond를 위한 비직교 다중 접속 기술, 신수용)		
해설자	이상용 기술사(제 124회 정보관리기술사 / orangeday77@gmail.com)		

I. 5G 주파수 효율 향상 기술, NOMA (Non-Orthogonal Multiple Access)의 개요

가. NOMA (Non-Orthogonal Multiple Access)의 정의

- 셀의 주파수 용량 향상을 위해 동일한 시간, 주파수, 공간 자원상에 다수의 사용자들을 위한 신호를 동시에 전송하여 주파수 효율을 향상시키는 기술

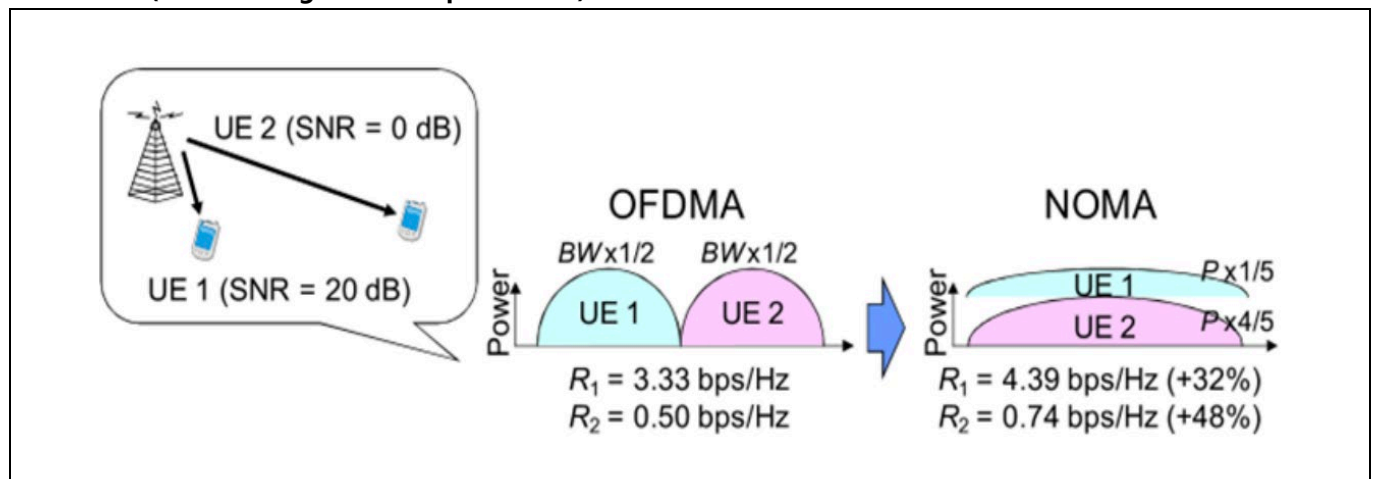
나. NOMA (Non-Orthogonal Multiple Access)의 특징

비직교성	- 직교성을 의도적으로 위배함으로써 다수의 신호를 중첩시키고 순차적으로 간섭 제거
동일 시간, 동일 주파수 사용	- 전력분배를 이용하여 신호를 전력영역에서 다중화하여 동일시간 동일주파수 대역 사용 가능

- 기존 직교 다중 접속(OFDMA) 방식이 갖고 있던 주파수 자원 할당 관점에서의 직교성(orthogonality property)을 깨고 같은 주파수 자원 상에 두 대 이상의 단말 데이터를 동시에 중첩 할당하여 자원 효율 높임

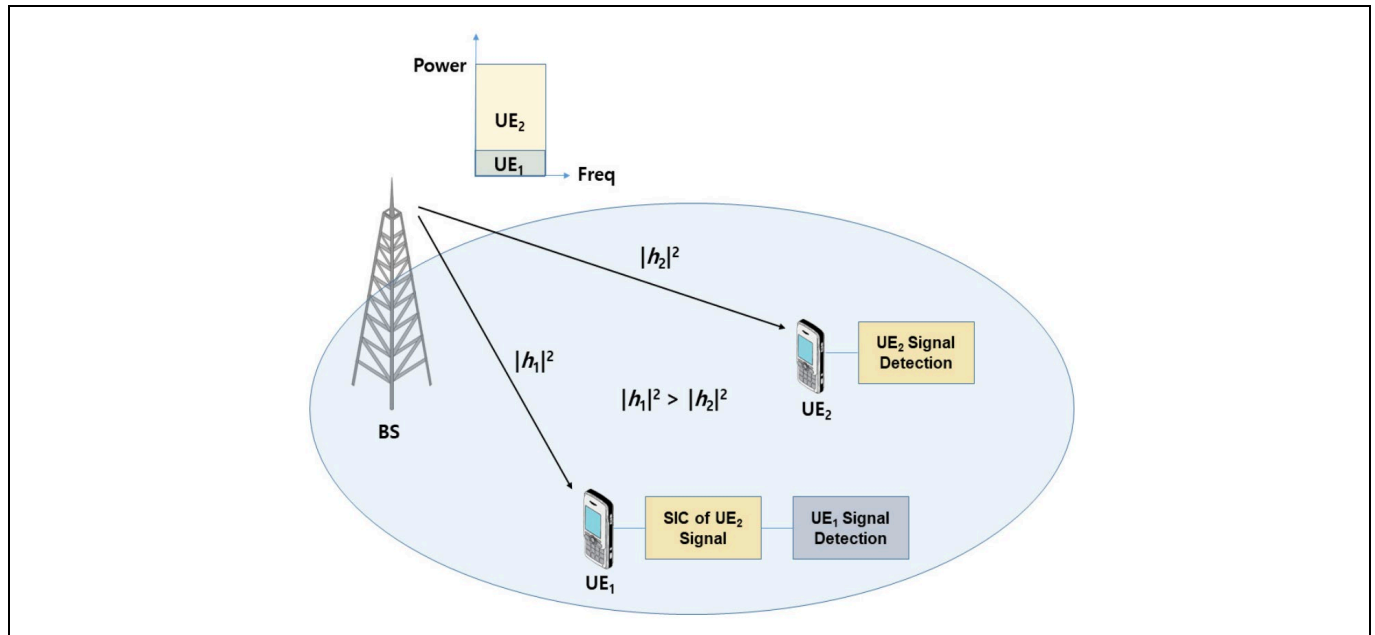
II. NOMA (Non-Orthogonal Multiple Access)의 개념도와 원리 및 핵심기술

가. NOMA (Non-Orthogonal Multiple Access)의 개념도



- NOMA는 기존의 다중 접속 방식에서의 직교성을 깨고, 동일한 주파수 대역에 둘 이상의 사용자들이 전송을 가능하게 함으로써 주파수 자원 효율성을 높이는 기술
- NOMA는 기존의 직교 다중 접속(Orthogonal Multiple Access: OMA)기술보다 우수한 주파수 효율 가지게 됨

나. NOMA (Non-Orthogonal Multiple Access)의 원리 및 핵심기술



원리	<ul style="list-style-type: none"> - 기존 다중접속 기법은 시간/주파수 영역에서 다중화 되지만, NOMA는 전력 영역의 다중화 기법을 활용 - 전력 분배를 이용하여 신호를 전력 영역에서 다중화시키기 때문에 다수의 사용자가 동일한 시간에 동일한 주파수 영역을 사용하는 것이 가능 	
구분	핵심 기술	설명
기지국	- 중첩코딩	- UE1과 UE2에는 각각의 사용자의 채널이득을 기반으로 기지국의 송신전력을 할당하고 두 사용자의 신호가 중첩코딩되어 전송
수신 단말	- 순차적 간섭 제거 (Successive Interference Cancellation: SIC)	- 신호의 세기가 큰 셀 경계 단말의 간섭신호를 먼저 복호화하여 제거 후 자신의 신호를 복호화

III. NOMA의 활용 연구

사례	설명
MIMO-NOMA	- 동시 사용자 수 대비 더 많은 안테나가 기지국에 장착
협력적 통신	- 다수의 릴레이의 도움을 통해 소스와 목적지 간의 통신 연결을 수행하는 협력통신
인지 무선 NW	- CRN(Cognitive Radio Network)
Massive IoT	- 임의 접속 절차 없이 활용 가능하므로 액세스 지연과 신호 오버 헤드 감소

“끝”

13	정렬		
문제	트리정렬(Tree Sort)		
도메인	알고리즘	난이도	하(상/중/하)
키워드	O(log ₂ N), 이진트리, 중위 순회		
출제배경	이진 탐색트리를 활용한 정렬 알고리즘의 지식 확인		
참고문헌	geeksforgeeks.org		
해설자	이상용 기술사(제 124회 정보관리기술사 / orangeday77@gmail.com)		

I. $O(\log_2 N)$ 의 시간 복잡도, 트리정렬(Tree Sort)의 개요

가. 트리정렬(Tree Sort)의 정의

- 이진 탐색 트리 형태의 구성을 띄면서 데이터가 삽입/삭제될 경우 트리탐색 과정을 거쳐 재구성한 후, 키 값을 삽입 후 중위 순회를 수행하는 정렬 알고리즘

나. 이진 탐색 트리 원리

	(규칙1) 모든 원소의 키는 유일해야 함
	(규칙1) 루트보다 작은 값은 왼쪽에 위치
	(규칙1) 루트보다 큰 값은 오른쪽에 위치

- 이진 탐색 트리로 삽입된 1, 3, 6, 8, 10 값을 중위순회 방법으로 오름차순 정렬 진행함

II. 트리정렬(Tree Sort) 매커니즘

개념도	순서	매커니즘 설명
	1	<ul style="list-style-type: none"> - 정렬될 첫 값이 루트 노드가 됨 - 예시 값: [8, 3, 10, 1, 6]
	2	정렬할 원소N개를 차례대로 트리에 삽입하여 이진탐색트리 구성 <ul style="list-style-type: none"> - 다음 값부터는 기존 노드 값과 비교 - 추가 노드 값 < 기존 노드 값 -> 왼쪽 자식 노드로 추가 - 추가 노드 값 >= 기존 노드 값 -> 오른쪽 자식 노드로 추가
	3	- 원소N개 까지 2단계 반복
	4	- 이진 탐색 트리를 중위 순회하여 원소 저장 - 예시결과: [1, 3, 6, 8, 10] (오름차순)

III. 트리정렬(Tree Sort) 코드

코드 예시	
<pre># Class containing left and # right child of current # node and key value class Node: def __init__(self, item = 0): self.key = item self.left, self.right = None, None # Root of BST root = None def insert(key): global root root = insertRec(root, key) # insert a new key in BST def insertRec(root, key): # If the tree is empty, return a new node if (root == None): root = Node(key) return root if (key < root.key): root.left = insertRec(root.left, key) elif (key > root.key): root.right = insertRec(root.right, key) return root</pre>	<pre># inorder traversal of BST def inorderRec(root): if (root != None): inorderRec(root.left) print(root.key, end = " ") inorderRec(root.right) def treeins(arr): for i in range(len(arr)): insert(arr[i]) # Driver Code def main(): arr = [5, 4, 7, 2, 11] treeins(arr) inorderRec(root) # output 2 4 5 7 11</pre>
알고리즘 설명	
<ul style="list-style-type: none"> - 단계1: Take the elements input in an array. - 단계2: Create a Binary search tree by inserting data items from the array into the binary search tree. - 단계3: Perform in-order traversal on the tree to get the elements in sorted order. 	

“끝”



ITPE 기술사회

제129회 정보처리기술사 기출문제 해설집

대 상	정보관리기술사, 컴퓨터시스템응용기술사, 정보통신기술사, 정보시스템감리사 시험
발행일	2023년 02월 04일
집 필	강정배PE, 소민호PE, 전일PE, 정상PE, 석PE, 이상용PE, 장건환PE
출 판	ITPE(Information Technology Professional Engineer)
주 소	ITPE 대치점 서울시 강남구 선릉로 86길 17 선릉엠티빌딩 7층 ITPE 선릉점 서울시 강남구 선릉로 86길 15, 3층 IT교육센터 아이티피이 ITPE 강남점 서울시 강남구 테헤란로 52길 21 파라다이스벤처타워 3층 303호 ITPE 영등포점 서울시 영등포구 당산동2가 하나비즈타워 7층 ITPE
연락처	070-4077-1267 / itpe@itpe.co.kr

본 저작물은 [ITPE\(아이티피이\)](http://itpe.co.kr)에 저작권이 있습니다.

저작권자의 허락없이 **본 저작물을 불법적인 복제 및 유통, 배포**하는 경우
법적인 처벌을 받을 수 있습니다.