

001 2020 1회 91번

소프트웨어 생명주기

1. 폭포수 모형
 - A. 이전 단계로 돌아갈 수 없다. 각 단계를 확실히 매듭짓고 그 결과를 철저히 검토하여 승인 과정 거친 후 다음 단계로 진행하는 개발 방법론
 - B. 소프트웨어공학에서 가장 오래됨, 가장 폭넓게, 전통적인 sw생명주기, 고전적 생명모형주기
 - C. 개발의 한 과정이 끝나야 다음단계로 진행 가능한 개발 방법론
 - D. 제품 일부가 될 매뉴얼 작성해야한다.
 - E. 2개 이상의 과정이 병행하여 수행되지 않는다
 - F. 각 단계가 끝난 후에는 다음 단계를 수행하기 위한 결과물이 명확하게 산출되어야 한다(단계적정의, 산출물 명확)
 - G. 순차적 접근방법
 - H. 모형 적용경험과 성공사례 많다.
2. 프로토타입 모형, 원형 모형
 - A. 사용자의 요구사항을 정확하게 하기 위해 실제 개발될 소프트웨어에 대한 견본품을 만들어 최종 결과물을 예측하는 모형이다
 - B. 시제품(프로토타입)은 사용자와 시스템 사이의 인터페이스 중점을 두어 개발한다
 - C. 시스템 일부 or 모형을 만드는 과정으로서 요구된 소프트웨어를 구현하는데, 이는 추후 구현단계에서 사용될 골격 코드가 된다.
 - D. Sw의 개발이 완료된 시점에서 오류가 발견되는 폭포수 모형의 단점 보완하기위한 모형
3. 나선형 모형, 점진적 모형, Spiral model
 - A. 보험이 제안
 - B. 폭포수 모형과 프로토타입 모형 장점에 위험 분석기능 추가한 모형
 - C. 대규모
 - D. 여러 번의 SW 개발 과정을 거쳐 점진적으로 완벽한 최종 SW를 개발
 - E. SW를 개발하면서 발생 할 수 있는 위험 관리 및 최소화가 목적
 - F. 점진적으로 개발 과정이 반복되므로 누락되거나 추가된 요구사항을 첨가할 수 있고, 정밀, 유지보수 과정이 필요없다.
 - G. 계획 및 정의 → 위험 분석 → 공학적 개발 → 고객 평가
4. 애자일 모형 Agile Model
 - A. 고객의 요구사항 변화에 유연하게 대응할 수 있도록 일정한 주기를 반복하면서 개발 과정 진행
 - B. 좋은 것을 빠르게, 낭비 없이 만들기 위해 고객과의 소통에 초점 맞춘 방법론
 - C. 스프린트(Sprint) 이터레이션 (Iteration) 과 같은 짧은 개발 주기 반복, 반복되는 주기마다 만들어지는 결과물에 대한 고객의 평가와 요구를 적극 수용
 - D. 각 개발주기에서는 고객의 요구사항에 우선순위를 부여하여 개발작업 진행
 - E. 소규모 프로젝트, 고도로 숙달된 개발자, 급변하는 요구사항에 적합
 - F. 스크럼(Scrum), XP(eXtreme Programming), 칸반(Kanban) , Lean, 크리스탈(Crystal), ASD, FDD, DSDM, DAD

003 2020 1회 11번x

XP(eXtreme Programming)의 5가지 핵심원칙

1. 의사소통 (Communication)
2. 단순성 (Simplicity)
3. 용기 (Courage)
4. 존중 (Respect)
5. 피드백(FeedBack)

005 2020 1회 18번**DBMS(데이터베이스 관리 시스템) 분석시(요구사항식별시) 고려사항**

1. 가용성
2. 성능
3. 기술 지원
4. 상호 호환성
5. 구축 비용

006 2020 1회. 13번**요구사항 정의 - 개발 프로세스**

도출 (Elicitation) 분석(Analysis) 명세(Specification) 확인(Validation)

1. 요구사항 도출(= 요구사항 수집, Requirement Elicitation)
 - 시스템, 사용자, 시스템 개발에 관련된 사람들이 서로 의견을 교환하여, 요구사항이 어디에 있는지, 어떻게 수집할 것인지 식별하고 이해하는 과정
 - 소프트웨어가 해결해야할 문제를 이해하는 첫 단계
 - 개발자와 고객 사이의 관계 만들어짐, 이해관계자(Stakeholder) 식별
 - 이해관계자 간의 의사소통 중요
 - 소프트웨어 개발 생명주기(SDLC, software Development Life Cycle)동안 지속적으로 반복
 - 도출 주요 기법 - 인터뷰, 설문, 브레인스토밍, 워크샵, 프로토타이핑, 유스케이스
2. 요구사항 분석(Requirement Analysis)
 - 요구사항 중, 명확하지 않거나 모호, 이해되지 않는 부분 발견 및 걸러내는 과정
 - 사용자 요구사항의 타당성 조사, 비용 일정에 대한 제약 설정
 - 서로 상충되는(내용 중복, 하나로 통합) 요구사항 있을 때 해결
 - 도출된 요구사항을 토대로 sw의 범위 파악, sw와 주변 환경이 상호작용하는 방법 이해
 - 요구사항 정의 문서화

009 2020. 1회 12번**UML(Unified Modeling Language)**

- 시스템 개발과정에서 의사소통이 원활하도록 표준화한 객체지향 모델링 언어
- UML 구성요소 : 사물(Things), 관계(Relationships), 다이어그램(diagram)
- 1. 사물(Things) : 모델을 구성하는 가장 중요한 기본 요소, 다이어그램 안에서 관계가 형성될수 있는 대상들

구조 사물 (Structural Things)	시스템의 개념적, 물리적 요소 표현 Class, Use case, Component, Node
행동 사물 (Behavioral Things)	시간과 공간에 따른 요소들의 행위 표현 Interaction(상호작용), State Machine(상태머신)
그룹 사물 (Grouping Things)	요소들을 그룹으로 묶어서 표현 Package
주해 사물 (Annotation Things)	부가적인 설명이나 제약조건 표현 Note

2. 관계(Relationships) : 사물과 사물 사이의 연관성 표현

연관관계 (Association)	<ul style="list-style-type: none"> - 2개 이상의 사물이 서로 관련됨 - 사물 사이를 실선으로, 방향성은 화살표로 표현, - 양방향의 경우, 화살표 생략 실선으로만 연결, - 연관에 참여하는 객체수(다중도)를 선위에 표현 		-
	1	1개의 객체가 연관	

	N	n개의 객체가 연관	
	0..1	연관된 객체가 없거나 1개만 존재	
	0.* 또는 *	연관된 객체가 없거나 다수	
	1..*	연관된 객체가 적어도 1개 이상	
	n..*	연관된 객체가 적어도 n개 이상	
	n..m	연관된 객체가 최소 n개에서 최대m개	
집합관계 (Aggregation)	<ul style="list-style-type: none"> 하나의 사물이 다른 사물에 포함되어있는 관계 포함하는 쪽(전체)과 포함되는 쪽(부분)은 서로 독립적 포함되는 쪽(part)에서 포함하는 쪽(whole)으로 빈 마름모 연결 		-
포함관계 (Composition)	<ul style="list-style-type: none"> 집합관계의 특수한 형태, 포함하는 사물의 변화가 포함되는 사물에게 영향을 미치는 관계를 표현 포함하는 쪽(전체)과 포함되는 쪽(부분)은 서로 독립적일 수 없음, 생명주기를 함께함. 포함되는 쪽(part)에서 포함하는 쪽(whole)으로 속이 채워진 마름모 연결 		
일반화 관계 (Generalization)	<ul style="list-style-type: none"> 하나의 사물이 다른 사물에 비해 더 일반적인지, 구체적인지 표현 일반적인 개념(상위, 부모) 구체적인 개념(하위, 자식) 구체적인 사물에서 일반적인 사물쪽으로 속이 빈 화살표 연결 		
의존 관계 (Dependency)	<ul style="list-style-type: none"> 사물 사이에 서로 연관은 있으나, 필요에 의해 서로에게 영향을 주는, 짧은 시간 동안만 연관을 유지하는 관계 하나의 사물과 다른 사물은 소유 관계는 아니지만, 사물의 변화가 다른 사물에도 영향을 미침 영향을 주는 사물(이용자)이 영향 받는 사물(제공자)쪽으로 점선 화살표 연결 		
실체화 관계 (Realization)	<ul style="list-style-type: none"> 사물이 할 수 있거나, 해야하는 기능(행위, 인터페이스)으로 서로를 그룹화 할 수 있는 관계 표현 사물에서 기능쪽으로 속이 빈 점선 화살표연결 		

3. 다이어그램(Diagram) : 사물과 관계를 도형으로 표현

- 여러 관점에서 시스템을 가시화한 뷰를 제공함으로써 의사소통에 도움을 줌
- 정적 모델링에서는 구조적 다이어그램을 사용, 동적 모델링에서는 행위 다이어그램 사용
 - 구조적(Structural) 다이어그램의 종류

Class Diagram 클래스 다이어그램	<ul style="list-style-type: none">- 클래스와 클래스가 가지는 속성, 클래스 사이의 관계 표현- 시스템의 구조 파악, 구조상의 문제점 도출	
Object Diagram 객체 다이어그램	<ul style="list-style-type: none">- 클래스에 속한 사물(객체)들 = 인스턴스(instance)를 특정 시점의 객체와 객체 사이의 관계로 표현	
Component Diagram 컴포넌트 다이어그램	<ul style="list-style-type: none">- 실제 구현 모듈인 컴포넌트 간의 관계나 컴포넌트 간의 인터페이스를 표현	구현 단계에서 사용
Deployment Diagram 배치 다이어그램	<ul style="list-style-type: none">- 물리적요소(결과물, 프로세스, 컴포넌트)들의 위치를 표현- 노드와 의사소통(통신)경로로 표현	
Composite Structure Diagram 복합체 구조 다이어그램	<ul style="list-style-type: none">- 클래스나 컴포넌트가 복합 구조를 갖는 경우, 그 내부 구조를 표현	
Package Diagram 패키지 다이어그램	<ul style="list-style-type: none">- 유스케이스나 클래스 등의 모델 요소들을 그룹화한 패키지들의 관계를 표현	

■ 행위(Behavioral) 다이어그램의 종류

Use Case Diagram 유스케이스 다이어그램	<ul style="list-style-type: none"> - 사용자의 요구를 분석하는 것으로, 기능 모델링 작업에 사용 - 사용자(Actor)과 사용사례(use case)로 구성 - 사용 사례 간에는 여러 형태의 관계로 이뤄짐
Sequence Diagram 시퀀스 다이어그램	<ul style="list-style-type: none"> - 상호 작용하는 시스템이나 객체들이 주고받는 메시지 표현 생명선, 실행, 메시지
Communication Diagram 커뮤니케이션 다이어그램	<ul style="list-style-type: none"> - 동작에 참여하는 객체들이 주고받는 메시지를 표현하는데, 메시지 뿐만 아니라 객체들 간의 연관까지 표현
State Diagram 상태 다이어그램	<ul style="list-style-type: none"> - 하나의 객체가 자신이 속한 클래스의 상태 변화 or 다른 객체와의 상호 작용에 따라 상태가 어떻게 변화하는지 표현
Active Diagram 활동 다이어그램	<ul style="list-style-type: none"> - 시스템이 어떤 기능을 수행하는지 객체의 처리 로직이나 조건에 따른 처리의 흐름을 순서에 따라 표현한다
Interaction Overview Diagram 상호작용 개요 다이어그램	<ul style="list-style-type: none"> - 상호작용 다이어그램 간의 제어흐름 표현
Timing Diagram 타이밍 다이어그램	<ul style="list-style-type: none"> - 객체 상태 변화와 시간 제약을 명시적으로 표현

010 2020 1회, 10번

사용자인터페이스(UI) 설계 원칙, 기본원칙

1. 직관성 : 누구나 쉽게 이해하고, 사용
2. 유효성 : 사용자의 목적을 정확하고, 완벽하게 달성
3. 학습성 : 누구나 쉽게 배우고 익힐 수 있어야함
4. 유연성 : 사용자의 요구사항을 최대한 수용하고 실수 최소화

014 2020 1회 30번

품질 요구사항 - ISO/IEC 9126

- 소프트웨어의 품질 특성과 평가를 위한 표준 지침

Functionality 기능성	Sw가 사용자의 요구사항을 정확하게 만족하는 기능을 제공하는지에 대한 여부	① 적절성/적합성 ② 정밀성/정확성 ③ 상호 운용성 ④ 보안성 ⑤ 준수성	Suitability Accuracy Interoperability Security Compliance
Reliability 신뢰성	Sw가 요구된 기능을 정확하고 일관되게 오류없이 수행할 수 있는정도	① 성숙성 ② 고장 허용성 ③ 회복성	Maturity Fault Tolerance Recoverability
Usability 사용성	사용자가 정확하게 이해하고, 사용하며 향후 다시 사용하고 싶은 정도	① 이해성 ② 학습성 ③ 운용성 ④ 친밀성	Understandability Learnability Operability Attractiveness
Efficiency 효율성	사용자가 요구하는 기능을 할당된 시간동안 한정된 자원으로 얼마나 처리할 수 있는지에 대한 정도	① 시간 효율성 ② 자원 효율성	Time Behavior Resource Behavior
Maintainability 유지보수성	환경변화 or 새로운 요구사항이 발생했을때, sw를 개선 및 확장할 수 있는정도	① 분석성 ② 변경성 ③ 안정성 ④ 시험성	Analyzability Changeability Stability Testability

Portability 이식성	Sw가 다른 환경에서도 얼마나 쉽게 적용할수 있는지에 대한 정도	① 적용성 ② 설치성 ③ 대체성 ④ 공존성	Adaptability Installability Replaceability Co-existence
--------------------	--	----------------------------------	--

022 2020 1회. 3번

객체지향의 주요 구성요소와 개념 - 객체, 클래스, 캡슐화, 상속, 다형성

1. 객체(Object) : 데이터와 함수를 캡슐화한 하나의 소프트웨어 모듈
2. 클래스(Class) : 공통된 속성과 연산을 갖는 객체의 집합 / 데이터를 추상화 하는 단위
 - A. 인스턴스(Instance) : 클래스에 속한 각각의 객체
 - B. 슈퍼클래스(Super Class) : 특정 클래스의 상위(부모) 클래스
 - C. 서브클래스(Sub Class) : 특정 클래스의 하위(자식) 클래스
3. 캡슐화(Encapsulation) : 데이터와 함수를 하나로 묶는 것
4. 상속(Inheritance) : 상위 클래스(부모클래스)의 모든 속성과 연산을 하위 클래스(자식클래스)가 물려받는것
5. 다형성(Polymorphism) : 메시지에 의해 객체(클래스)가 연산을 수행할 때, 하나의 메시지에 대해 각각의 객체(클래스)가 가지고 있는 고유한 방법(특성)으로 응답할 수 있는 능력을 의미

객체지향 방법론의 구성요소 - 객체, 클래스, 메시지

객체지향 방법론의 기본원칙 - 정보은닉, 추상화, 캡슐화, 상속성, 다형성

023 2020 1회 64번

1. 결합도 (Coupling)

- ① 모듈 간에 상호 의존하는 정도
- ② 두 모듈 사이의 연관관계
- ③ 결합도가 약할수록 품질이 높고, 강할수록 품질이 낮다

자료 결합도 → 스탬프 결합도 → 제어 결합도 → 외부 결합도 → 공통 결합도 → 내용 결합도

Data stamp control external common content
(약 강)

자료 결합도 Data Coupling	
스탬프 결합도 stamp Coupling	
제어 결합도 control Coupling	
외부 결합도 external Coupling	
공통 결합도 common Coupling	
내용 결합도 content Coupling	

2. 응집도 (Cohension)

기능적 응집도→순차적 응집도→교환적 응집도→절차적 응집도→시간적 응집도→논리적 응집도→우연적 응집도
(강 약)

기능적 응집도 Functional	모듈 내부의 모든 기능요소들이 단일문제와 연관되어 수행될 경우의 응집도
-----------------------	---

순차적 응집도 Sequencetial	모듈 내 하나의 활동으로부터 나온 출력 데이터를 그 다음 활동의 입력 데이터로 사용할 경우의 응집도
교환(통신)적 응집도 Communication	동일한 입력과 출력을 사용하여, 서로 다른 기능을 수행하는 구성요소들이 모였을 경우의 응집도
절차적 응집도 Procedural	모듈이 다수의 관련 기능을 가질 때, 모듈 안의 구성요소들이 그 기능을 순차적으로 수행할 경우의 응집도
시간적 응집도 Temporal	특정 시간에 처리되는 몇 개의 기능을 모아 하나의 모듈로 작성할 경우의 응집도
논리적 응집도 Logical	유사한 성격을 갖거나 특정 형태로 분류되는 처리요소들로, 하나의 모듈이 형성되는 경우의 응집도
우연적 응집도 Coincidental	모듈 내부의 각 구성 요소들이 서로 관련 없는 요소로만 구성된 경우의 응집도

- 모듈의 독립성은 결합도와 응집도에 의해 측정 122 서버프로그램 구현
- 독립성을 높이면 / 모듈의 결합도1를 약하게, 응집도2를 강하게, 모듈크기 작게

024 2020 1회 15번

공통모듈 명세기법

1. Correctness 정확성 : 시스템 구현시 해당 기능이 필요하다는 것을 알 수 있도록 정확히 작성
2. Clarity 명확성 : 해당 기능을 이해할 때, 중의적으로 해석되지 않도록 명확하게 작성
3. Completeness 완전성 : 시스템 구현을 위해, 필요한 모든 것 기술
4. Consistency 일관성 : 공통 기능들 간 상호 충돌이 발생하지 않도록 작성
5. Traceability 추적성 : 기능에 대한 요구사항의 출처, 관련 시스템 등의 관계를 파악할 수 있도록 작성

효과적인 모듈 설계 방안

1. 결합도는 줄이고, 응집도는 높여서, 모듈의 독립성과 재사용성을 높인다.
2. 복잡도와 중복성을 줄이고, 일관성 유지
3. 모듈 제어 영역안에서 그 모듈의 영향 영역을 유지시킨다.
4. 모듈의 기능은 예측 가능해야하며, 지나치게 제한적이면 안된다.
5. 유지보수가 용이해야함
6. 모듈 크기는 시스템의 전반적인 기능과 구조를 이해하기 쉬운 크기로 분해한다.
7. 하나의 입구, 하나의 출구를 갖도록함
8. 인덱스 번호나 기능 코드들이 전반적인 처리 논리 구조에 예기치 못한 영향을 끼치지 않도록 모듈 인터페이스를 설계해야한다.

025 2020 1회, 2번

코드 주요기능

- 식별, 분류, 배열

코드의 종류 (Code)

순차 코드 (Sequence Code)	자료의 발생 순서, 크기 순서 등 일정 기준을 따라 최초의 자료로부터 차례로 일련번호를 부여하는 방법 = 순서 코드, 일련번호 코드
블록 코드 (Block Code)	코드화 대상 항목 중에서 공통성이 있는 것 끼리 블록으로 구분, 각 블록 내에서 일련번호를 부여하는 방법 = 구분코드 (1001~1100 총무부
10진 코드 (Decimal Code)	코드화 대상 항목을 0~9까지 10진 분할 하고, 다시 그 각각에 대해 10진 분할하는 방법을 필요한 만큼 반복하는 방법 = 도서 분류식 코드
그룹 분류 코드 (Group Classification Code)	코드화 대상 항목을 일정 기준에 따라 대분류, 중분류, 소분류 등으로 구분하고 각 그룹 안에서 일련번호 부여 (1-01-001 본사-총무부-인사계)

연상 코드 (Mnemocnic Code)	코드화 대상 항목의 명칭이나 약호와 관계있는 숫자나 문자, 기호를 이용하여 코드 부여하는 방법 (TV-40, : 40인치 티비)
표의 숫자코드 (Significant Digit Code)	코드화 대상 항목의 성질(길이, 넓이, 부피, 지름, 높이)의 물리적 수치를 그대로 코드에 적용시키는 방법 = 유효 숫자 코드 (120-300-60 같은 책상 사이즈)
합성 코드 (Combined Code)	필요한 기능을 하나의 코드로 수행하기 어려운 경우, 2개 이상을 코드를 조합하여 만드는 방법 (KE-711, 대한항공 711기)

026 2020 1회 7번**디자인 패턴(Design Pattern) – GoF(Gang of four) 디자인 패턴**

1. 생성패턴(Creational Pattern) – 객체 생성과 관련된 패턴
 - ① Abstract Factory 추상팩토리
 - ② Builder 빌더
 - ③ Factory Method 팩토리 메소드 – 객체 생성을 서브 클래스에서 처리하도록 분리한 캡슐화한 패턴, 상위 클래스에서 인터페이스만 정의하고 실제생성은 서브 클래스가 담당 = virtual-constructor
 - ④ Prototype 프로토타입
 - ⑤ Singleton 싱글톤
2. Structural Pattern 구조 패턴
 - ① Adapter 어댑터
 - ② Bridge 브리지
 - ③ Composite 컴포지트
 - ④ Decorator 데코레이터
 - ⑤ Façade 퍼싸드
 - ⑥ Flyweight 플라이웨이트
 - ⑦ Proxy 프록시
3. Behavioral Pattern 행위 패턴
 - ① Chain of Responsibility 책임 연쇄
 - ② Command 커맨드
 - ③ Interpreter 인터프리터
 - ④ Iterator 반복자
 - ⑤ Mediator 중재자
 - ⑥ Memento 메멘토
 - ⑦ Observer 옵서버
 - ⑧ State 상태
 - ⑨ Strategy 전략
 - ⑩ Template Method 템플릿 메소드
 - ⑪ Visitor 방문자

028 2020 1회,1번**인터페이스 요구사항 검증 방법**

1. 요구사항 검토(requirements review)
 - A. 동료검토(Peer Review) : 요구사항 명세서 작성자가 명세서 내용을 직접 설명, 동료가 이를 들으면서 결함을 발견하는 형태의 검토방법
 - B. 워크스루(Walk Through) : 검토 회의 전, 요구사항 명세서를 미리 배포, 사전 검토후 짧은 검토회의를 통해 결함을 발견하는 검토방법
 - C. 인스펙션(Inspection) : 요구사항 명세서 작성자를 제외한 다른 검토 전문가들이 요구사항 명세서를 확인하면서 결함을 발견하는 형태의 검토방법

2. 프로토타이핑(Prototyping) : 사용자의 요구사항을 정확히 파악하기 위해, 실제 설계될 소프트웨어에 대한 견본품을 만들어 최종 결과물 예측
3. 테스트 설계 : 요구사항을 테스트할 수 있도록 작성되어야 하며, 이를 위해 테스트 케이스(Test Case)를 생성하여 이후에 요구사항이 현실적으로 테스트 가능한지 검토
4. CASE(Computer Aided Software Engineering) 도구 활용 : 일관성 분석(consistency Analysis)을 통해 요구사항 변경사항의 추적 및 분석, 관리하고, 표준 준수 여부를 확인
 - A. 주요 기능 : 그래픽 지원, sw 생명주기 전 단계의 연결, 다양한 sw 개발 모형 지원
 - B. Sw 개발 과정의 일부 or 전체를 자동화하기 위한 도구
 - C. 표준화된 개발 환경 구축 및 문서 자동화 기능 제공
 - D. 작업 과정 및 데이터 공유를 통해 작업자가 의사소통(communication)증대

033 2020 1회 9번

미들웨어 middleware 종류

1. DB (Date Base)
 - 데이터베이스 벤더에서 제공하는 클라이언트에서 원격의 DB와 연결하기 위한 MW, 2-tier 아키텍처
2. RPC (Remote Procedure Call)
 - 원격 프로시저 호출은 응용 프로그램의 프로시저를 사용하여, 원격 프로시저를 마치 로컬 프로시저 처럼 호출하는 방식의 MW
3. MOM (Message Oriented Middelware)
 - 메시지 지향 미들웨어는 메시지 기반의 비동기형 메시지를 전달하는 방식의 MW
 - 이기종 분산 데이터 시스템의 데이터 동기를 위해 많이 사용
4. TP-Monitor (Transaction Processing Monitor)
 - 트랜잭션 처리 모니터는 온라인 트랜잭션 업무에서 트랜잭션을 처리 및 감시하는 MW
 - 사용자 수가 증가해도 빠른 응답 속도를 유지해야하는 업무에 사용(항공기, 철도예약)
5. ORB (Object Request Broker)
 - 객체 요청 브로커는 객체 지향 미들웨어로, 코바(cobba) 표준 스펙을 구현한 mw
6. WAS (Web Application Server)
 - 웹 애플리케이션 서버는 동적인 콘텐츠를 처리하기 위해 사용되는 mw
 - 웹 환경 구현
 - HTTP 세션처리를 위한 웹서버 기능과 미션-크리티컬한 기업 업무까지 JAVA, EJB 컴포넌트 기반으로 구현 가능

041 2020 1회 39번

Sw 패키징 고려사항

- 패키징시, 사용자에게 배포되는 sw이므로 보안을 고려한다.
- 사용자 편의성을 위한 복잡성 및 비효율성 문제를 고려한다
- 제품 sw 종류에 적합한 암호화 알고리즘을 적용
- 추가로 다양한 이기종 연동 고려
- 제품 sw 종류에 적합한 암호화 알고리즘을 고려
- 개발자가 아닌 사용자를 중심으로 진행
- 소스 코드는 향후 관리를 고려하여 모듈화하여 패키징한다.

043 2020 1회 32번

DRM 디지털 저작권 관리의 기술요소

- ① Encryption 암호화
- ② Key Management 키 관리
- ③ Packager 암호화 파일 생성
- ④ Identification 식별기술 / 콘텐츠 식별체계표현
- ⑤ Right Expresssion 저작권 표현
- ⑥ Policy management 정책관리
- ⑦ Tamper Resistance 크랙 방지
- ⑧ Authentication 인증
- ⑨ -----라이선스 발급 및 관리

046 2020 1회 34번

소프트웨어 형상 관리

- 형상관리(SCM, software Configuration Management) : sw의 개발 과정에서 sw의 변경사항을 관리하기위해 개발된 일련의 활동
- 형상 관리를 통해, 이전 리버전이나 버전에 대한 정보에 접근 가능하여 배포본 관리 유용
- 불필요한 사용자의 소스 수정 제한
- 동일한 프로젝트에 대해 여러 개발자가 동시 개발 가능

형상관리 도구의 주요기능

1. 저장소 Repository
2. 가져오기 import
3. 체크아웃
4. 체크인
5. 커밋 commit
6. 동기화 update

051 2020 1회. 22번

테스트 기법에 따른 애플리케이션 테스트

1. 화이트박스 테스트(White Box test)	2. 블랙박스 테스트 (Black Box Test)
<ol style="list-style-type: none"> ① 모듈의 원시코드를 오픈시킨 상태에서 원시코드의 논리적인 모든 경로를 테스트하여 tc를 설계하는 방법 ② 설계된 절차에 초점을 둔 구조적 테스트 ③ 프로시저 설계의 제어구조를 사용하여 tc 설계, ④ 테스트 과정에 초기에 적용 ⑤ 모듈 안의 작동을 직접 관찰 ⑥ 원시 코드(source code, 모듈)의 모든 문장을 한 번 이상 실행함으로써 수행 ⑦ 프로그램의 제어 구조에 따라 분기점 부분들을 수행함으로써, 논리적 경로 제어 	<ol style="list-style-type: none"> ① 소프트웨어가 수행할 특정 기능을 알기 위해, 각 기능이 완전히 작동되는 것을 입증하는 테스트 ② = 기능 테스트 ③ 사용자 요구사항 명세를 보면서 테스트 하는 것 ④ 구현된 기능을 테스트함 ⑤ 소프트웨어 인터페이스에서 실시되는 테스트 ⑥ 부정확, 누락된 기능, 인터페이스 오류, 자료 구조나 외부 DB 접근에 따른 오류, 행위나 성능 오류, 초기화와 종료 오류 등을 발견하기 위해 사용 ⑦ 테스트 과정 후반부에 적용
Statement Coverage (문장 검증기준) Branch Coverage (분기 검증 기준) Condition Coverage (조건 검증 시준) Branch/Condition Coverage (분기/조건 기준)	Equivalence Partitioning Testing (동치/동등 분할검사) Boundary Value Analysis (경계값 분석) Cause-Effect Graphing Testing (원인-효과 그래프 검사) Error Guessing(오류 예측 검사) Comparison Testing(비교검사)

052 2020 1회, 37번**개발 단계에 따른 애플리케이션 테스트****1. 인수테스트**

- A. 개발한 sw가 사용자의 요구사항을 충족하는지에 중점을 두고 테스트
- B. 사용자가 직접 테스트
- C. 문제가 없으면 사용자 sw 인수, 프로젝트 종료
- D. 종류

사용자 인수 테스트	- 사용자가 시스템 사용의 적절성 여부를 확인
운영상의 인수 테스트	- 시스템 관리자가 시스템 인수 시 수행하는 테스트 기법 - 백업/복원 시스템, 재난 복구, 사용자 관리, 정기 점검 등 확인
계약 인수 테스트	- 계약상의 인수/검수 조건을 준수하는지 확인
규정 인수 테스트	- Sw가 정부 지침, 법규, 규정 등 규정에 맞게 개발되었는지 확인
알파 테스트	- 개발자의 장소에서 사용자가 개발자 앞에서 행하는 테스트 기법 - 테스트는 통제된 환경에서 이뤄짐, - 오류와 사용상의 문제점을 사용자와 개발자가 함께 확인 및 기록
베타 테스트	- 선정된 최종 사용자가 여러명의 사용장 앞에서 행하는 테스트 기법 - 실업무를 가지고 사용자가 직접 테스트하는 것, - 개발자에 의해 제어되지 않은 상태에서 테스트가 행해지며, 발견된 오류와 사용상의 문제점을 기록하고 개발자에게 주기적으로 보고한다

053 2020 1회 38번**통합테스트****1. 하향식 통합 테스트 (Top Down Integration Test)**

- A. 깊이 우선 통합법, 넓이 우선 통합법 사용
- B. 테스트 초기부터 사용자에게 시스템 구조 보여줄 수 있다.
- C. 상위 모듈에서는 tc를 사용하기 어렵다.
- D. 상위 모듈에서 하위 모듈 방향으로 통합하면서 테스트하는 기법
- E. 절차
 - ① 주요 제어 모듈은 작성된 프로그램을 사용하고, 주요 제어 모듈의 종속 모듈들은 스텝(stub)로 대체한다
 - ② 깊이우선 or 넓이 우선 등의 통합방식에 따라, 하위 모듈인 스텝들이 한 번에 하나씩 실제모듈로 교체된다.
 - ③ 모듈이 통합될 때마다 테스트 실시
 - ④ 새로운 오류가 발생하지 않음을 보증하기 위해 회귀 테스트 실시

2. 상향식 통합 테스트 (Bottom Up Integration Test)

- A. 하위 모듈에서 상위 모듈 방향으로 통합하면서 테스트 하는 방법
- B. 가장 하위 단계의 모듈부터 통합 및 테스트가 수행되므로 스텝(stub)은 필요하지 않다.
- C. 하나의 주요 제어 모듈과 관련된 종속 모듈의 그룹인 클러스터(cluster)은 필요하다.
- D. 절차
 - ① 하위 모듈을 클러스터로 결합
 - ② 상위 모듈에서 데이터의 입출력을 확인하기 위해, 더미 모듈인 드라이버 작성
 - ③ 통합된 클러스터 단위로 테스트
 - ④ 테스트가 완료되면 클러스터는 프로그램 구조의 상위로 이동하여 결합하고, 드라이버는 실제 모듈로 대체

059 2020 1회. 36번**소스코드 품질 분석 도구**

1. 정적분석 도구 - pmd, cppcheck, SonarQube, checkstyle, ccm, cobertura
2. 동적분석 도구 - Avalanche, Valgrind

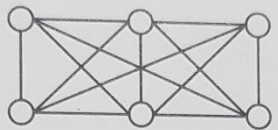
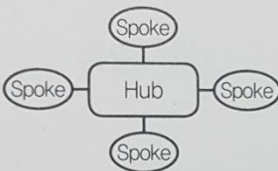
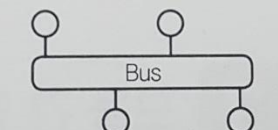
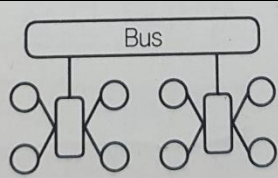
클린코드 작성원칙

가독성	누구든지 쉽게 읽도록 이해하기 쉬운 용어, 들여쓰기 기능 활용
단순성	코드 간단하게 작성, 한번에 한가지 처리하도록 코드 작성, 클래스/메소드/함수 등을 최소 단위로 분리
의존성 배제	코드가 다른 모듈에 미치는 영향 최소화 코드 변경시 다른 부분에 영향 없도록 작성
중복성 최소화	코드중복 최소화 중복된 코드 삭제, 공통된 코드 사용
추상화	상위클래스/메소드/함수에서는 간략히 애플리케이션 특성을 나타내고, 상세 내용은 하위클래스/메소드/함수에서 구현

061 2020 1회 25번**EAI(Enterprise Application Integration)**

- 기업 내 각종 애플리케이션 및 플랫폼 간의 정보 전달, 연계, 통합 등 상호 연동이 가능하게 해주는 솔루션
- 비즈니스 간 통합 및 연계성을 증대시켜 효율성 및 각 시스템 간의 확정성(Determinacy)을 높여준다

구축유형

Point-to-Point	<ul style="list-style-type: none"> - 가장 기본적인 애플리케이션 통합 방식 - 애플리케이션을 1:1 로 연결 - 변경 및 재사용이 어렵다. 	
Hub & Spoke	<ul style="list-style-type: none"> - 단일 점점인 허브 시스템을 통해 데이터를 전송하는 중앙 집중형 방식 - 확장 및 유지보수 용이 - 허브 장애 발생시 시스템 전체에 영향을 미침 	
Message Bus (ESB 방식)	<ul style="list-style-type: none"> - 애플리케이션 사이에 mw를 두어 처리하는 방식 - 확장성이 뛰어남 - 대용량 처리 가능 	
Hybrid	Hub & Spoke + Message Bus 혼합 방식 <ul style="list-style-type: none"> - 그룹 내에서는 Hub & Spoke 방식을, 그룹 간에는 Message Bus 방식 사용 - 필요한 경우, 한 가지 방식으로 EAI 방식 구현 가능 - 데이터 병목 현상 최소화 	

064 2020 1회 86번**JSON(JavaScript object Notation)**

- 속성-값 쌍으로 이뤄진 데이터 객체를 전달하기 위해 사람이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷
- 비동기 처리에 사용되는 AJAX에서 XML을 대체하여 사용되고 있음.

AJAX (Asynchronous JavaScript and XML)

- 웹 페이지에서 자바 스크립트를 사용하여 xml로 데이터를 교환 및 제어함으로써 이용자가 웹 페이지와 자유롭게 상호 작용할 수 있도록 하는 기술

- 비동기 통신기술사용

XML(eXtensible Markup Language)

- 특수한 목적을 갖는 마크업 언어를 만드는데 사용하는 다목적 마크업 언어

066 2020 1회 27

인터페이스 보안 기능 적용

네트워크 영역	<ul style="list-style-type: none"> - 인터페이스 송*수신 간 스니핑(sniffing)등을 이용한 데이터 탈취 및 변조 위협을 방지하기 위해 네트워크 트래픽에 대한 암호화를 설정한다. - 암호화는 인터페이스 아키텍처에 따라 IPsec, SSL, S-HTTP
애플리케이션 영역	<ul style="list-style-type: none"> - 소프트웨어 개발 보안 가이드를 참조하여 애플리케이션 코드 상의 보안 취약점을 보완하는 방향으로 애플리케이션 보안 기능을 적용한다.
데이터베이스 영역	<ul style="list-style-type: none"> - DB, 스키마, 엔티티의 접근 권한과 프로시저((Procedure), 트리거(Trigger) 등 DB 동작 객체의 보안 취약점에 보안 기능을 적용한다 - 개인 정보나 업무상 민감한 데이터의 경우, 데이터 자체의 보안방안(암호화나 익명화) 고려

068 2020 1회 24

인터페이스 구현 검증 도구

xUnit	Java(Junit), C++(Cppunit), .Net(Nunit)등 다양한 언어를 지원하는 단위 테스트 프레임워크
STAF	서비스 호출 및 컴포넌트 제사용 등의 환경을 지원하는 테스트 프레임워크
FitNesse	웹 기반 tc 설계, 실행, 결과 확인 등을 지원하는 테스트 프레임워크
NTAF	FitNesse의 장점(협업기능)과 STAF의 장점(재사용 및 확장성)을 통합한 NHN(Naver)의 테스트 자동화 프레임워크
Selenium	다양한 브라우저 및 개발언어를 지원하는 웹 애플리케이션 테스트 프레임워크
Water	Ruby를 사용하는 애플리케이션 테스트 프레임워크

070 2020 1회 46번

데이터베이스 설계

논리적설계(데이터 모델링, logical design) 단계

- 논리적 DB구조로 매핑(Mapping)
- 트랜잭션 인터페이스 설계
- 스키마 평가 및 정제
- DBMS에 맞는 논리적 자료 구조로 변환시키는 과정.

076 2020 1회 58번

피터첸 표기법 (Peter Chen Notation)

개체타입 Entity	사각형	
관계타입 Relation	마름모	
속성타입 Attribute	타원	
다중값 속성 (복합속성)	이중타원	
기본키 속성	밑줄타원	
복합 속성	복수타원	

1:1, 1:N, N:M 등의 개체 간 관계에 대한 대응수를 선위에 기술	관계	
개체타입과 속성 연결	선링크	

078 2020 1회. 48**관계형 데이터베이스의 Relation 구조**

- 릴레이션 : 테이블들을 표(table)형태로 표현한 것,
 - 릴레이션 스키마(구조 나타냄)와 릴레이션 인스턴스(실제값)으로 구성
- 튜플 Tuple
 - 릴레이션을 구성하는 각각의 행
 - 속성의 모임으로 구성
 - 파일구조에서 레코드와 같은 의미
 - 튜플의 수 = 카디널리티 cardinality, 기수, 대응수
- 속성 attribute, 애트리뷰트
 - DB를 구성하는 가장 작은 논리적 단위
 - 파일 구조상의 데이터 항목 or 데이터 필드에 해당
 - 개체의 특징 기술
 - 속성의 수 = 디그리 (degree), 차수
- 도메인 Domain
 - 하나의 애트리뷰트(속성)가 취할 수 있는 같은 타입의 원자(Atomic)값들의 집합

릴레이션의 특징

- 한 릴레이션에는 똑같은 튜플이 포함 될 수 없으므로, 릴레이션에 포함된 튜플들은 모두 상이함
- 한 릴레이션에 포함된 튜플 사이에는 순서가 없다
- 튜플의 삽입, 삭제 등의 작업으로 인해 릴레이션은 시간에 따라 변한다.
- 릴레이션 스키마를 구성하는 속성들 간의 순서는 중요하지 않다.
- 속성의 유일한 식별을 위해, 속성의 명칭은 유일해야하지만, 속성을 구성하는 값은 동일한 값이 있을 수 있다.
- 릴레이션을 구성하는 튜플을 유일하게 식별하기 위해, 속성들의 부분집합을 키(Key)로 설정한다.
- 속성의 값은 논리적으로 더 이상 쪼갤 수 없는 원자값만을 저장한다.

079 2020 1회 59번**관계형 DB의 제약조건 - key**

제약조건 : DB에 저장되는 데이터의 정확성을 보장하기 위해 키를 이용하여 입력되는 데이터에 제한을 주는 것, 개체 무결성 제약, 참조 무결성 제약

- 후보키 (Candidate Key)
 - 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용하는 속성들의 부분집합.
 - 기본키로 사용할 수 있는 속성
 - 하나의 릴레이션 내에서는 중복된 튜플이 있을 수 없으므로, 모든 릴레이션에는 반드시 하나 이상의 후보키 존재.
 - 후보키는 릴레이션에 있는 모든 튜플에 대해서 유일성과 최소성을 만족시켜야한다.
 - 유일성 : 하나의 키값, 하나의 튜플만을 유일하게 식별할 수 있다.
 - 최소성 : 모든 레코드들을 유일하게 식별하는데 꼭 필요한 속성으로 구성
- 기본키 (Primary Key)
- 대체키 (Alternate Key)
- 슈퍼키 (Super Key)
- 외래키 (Foreign Key)

080 2020 1회 43번

관계형 DB의 제약조건 - 무결성

1. 개체 무결성 (Entity Integrity, 실체 무결성)
 - A. 기본 테이블의 기본키를 구성하는 어떤 속성도 Null 값, 중복값을 가질 수 없다,
2. 도메인 무결성(Domain Integrity, 영역무결성)
 - A. 주어진 속성 값이 정의된 도메인에 속한 값이어야한다는 규정.
3. 참조 무결성 (Referential Integrity)
 - A. 외래값은 Null 값이거나 참조 릴레이션의 기본키 값과 동일해야한다.
 - B. 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다.
 - C. 외래키와 참조하려는 테이블의 기본키는 도메인과 속성 개수와 같아야한다.
4. 사용자 정의 무결성 (User-Defined Integrity)
 - A. 속성 값들을 사용자가 정의한 제약조건에 만족해야한다는 규정

081 2020 1회 50번

관계대수 및 관계해석

관계대수

1. 순수 관계 연산자 : select, project, join, division
 - A. Select : 릴레이션에 존재하는 튜플 중에서 선택조건을 만족하는 튜플의 부분집합을 구하여 새로운 릴레이션을 만드는 연산 σ
 - B. Project : 주어진 릴레이션에서 속성 리스트(Attribute list)에 제시된 속성 값만을 추출하여 새로운 릴레이션을 만드는 연산,, 중복발생시 제거. π
 - C. Join : 공통 속성을 중심으로 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 연산 \bowtie
 - D. Division : $X \supset Y$ 인 두개의 릴레이션 $R(X)$, $S(Y)$ 가 있을 때, R 의 속성이 S 의 속성 값을 모두 가진 튜플에서 S 가 가진속성을 제외한 속성만 구하는 연산 $R \div [속성r \div 속성s] S$
2. 일반 집합 연산자 : UNION, INTERSECTION, DIFFERENCE, CARTESIAN PRODUCT

082 2020 1회. 47번

1. 이행적 종속 관계(transtive Dependency) : $A \rightarrow B$ 이고 $B \rightarrow C$ 이면, $A \rightarrow C$ 를 만족하는 관계

2. 정규화 과정 : 두부이결다줘, 도부이결다줘

비정규 릴레이션

도메인이 원자값

1NF

부분적 함수 종속 제거 / 완전 함수적 종속관계 만족

2NF

이행적 함수 종속 제거

3NF

결정자이면서 후보키가 아닌 것 제거

BCNF(보이스/코드)

다치 종속 제거

4NF

조인 종속성 이용

5NF

3. 정규화 목적

- ① 데이터 구조의 안정성 및 무결성 유지
- ② 어떤 릴레이션이라도 DB내에서 표현 가능
- ③ 효과적인 검색 알고리즘 생성

- ④ 데이터 중복을 배제하여, 이상발생 방지 및 자료 저장 공간의 최소화 가능
- ⑤ 데이터 삽입 시, 릴레이션 재구성할 필요성을 줄인다.
- ⑥ 데이터 모형의 단순화 가능
- ⑦ 속성의 배열 상태 검증 가능
- ⑧ 개체와 속성의 누락 여부 확인 가능
- ⑨ 자료 검색과 추출의 효율성 추구

4. 이상 (Anomaly)

A. 개념 : 릴레이션 조작시 데이터들이 불필요하게 중복되어 예기치 않게 발생하는 곤란한 현상.

083 2020 1회 29번

반정규화 중복테이블

- 집계테이블의 추가
- 진행테이블의 추가
- 특정 부분만을 포함하는 테이블의 추가

086 2020 2회 35

파티셔닝 테이블(Partitioning Table)

1. 대용량 데이터의 테이블을 작은 논리적 단위인 파티션(Partition)으로 나눈 테이블
2. 파티션 키를 잘못구성할시, 성능저하등의 역효과 가져올수 있다.
3. 파티셔닝 방식에 따라
 - A. 범위 분할 Range Partitioning
 - B. 해시 분할 Hash Partitioning
 - C. 조합 분할 Composite Partitioning

087 2020 1회 51번

트랜잭션의 특성 - 데이터의 무결성(integrity) 보장위해

Atomicity 원자성	<ul style="list-style-type: none"> - 트랜잭션의 연산은 DB에 모두 반영되도록 완료(commit)되든지, 전혀 반영되지 않도록 복구(Rollback)해야한다. - 트랜잭션 내의 모든 명령은 반드시 완벽히 수행, 모두가 완벽히 수행되지 않고 어느 하나라도 오류발생시, 트랜잭션 전부가 취소되어야함.
Consistency 일관성	<ul style="list-style-type: none"> - 트랜잭션이 실행을 성공적으로 완료하면, 언제나 일관성 있는 DB 상태로 변환한다. - 시스템이 가지고 있는 고정요소는 트랜잭션 수행 전과 수행 완료 후의 상태가 같아야한다.
Isolation 독립성, 격리성, 순차성	<ul style="list-style-type: none"> - 둘 이상의 트랜잭션이 동시병행 수행될 경우, 어느 하나의 트랜잭션 실행 중에 다른 트랜잭션의 연산이 끼어들 수 없다. - 수행중인 트랜잭션은 완료될 때까지 다른 트랜잭션에서 수행결과를 참조 할 수 없다.
Durability 영속성, 지속성	<ul style="list-style-type: none"> - 성공적으로 완료된 트랜잭션의 결과는 시스템이 고장나더라도 영구적으로 반영되어야한다.

089 2020 1회 44번

뷰 view 설계

뷰 특징

- ① 사용자에게 접근이 허용된 자료만 제한적으로 보여주기 위해, 하나 이상의 기본 테이블로 유도된, 이름을 가지는 가상테이블,
- ② 임시적인 작업(데이터 보정작업, 처리과정 시험)을 위한 용도로 활용
- ③ 뷰는 기본테이블과 같은 형태의 구조 사용, 조작도 거의 같다.

- ④ 가상 테이블이기 때문에, 저장장치 내에 물리적으로 존재(구현)되어있지 않다.
- ⑤ 데이터의 논리적 독립성 제공
- ⑥ 필요한 데이터만 뷰로 정의해서 처리할 수 있기 때문에, 관리 용이, 명령문 간단
- ⑦ 뷰를 통해서만 데이터에 접근하게 되면, 뷰에 나타나지 않는 데이터를 안전하게 보호하는 효율적인 기법으로 사용할 수 있다.
- ⑧ 기본 테이블의 기본키를 포함한 속성(열)집합으로 뷰를 구성해야만 삽입, 삭제, 갱신 연산 가능.
- ⑨ 일단 정의된 뷰는 다른 뷰의 정의에 기초될 수 있다.
- ⑩ 뷰가 정의된 기본테이블이나, 뷰를 삭제하면, 그 테이블이나 뷰를 기초로 정의된 다른 뷰도 자동삭제
- ⑪ 조인문 사용 최소화로 사용상의 편의성 최대화,
- ⑫ Create, drop.

장점	단점
<ul style="list-style-type: none"> - 논리적 데이터 독립성 제공 - 동일 데이터에 대해 동시에 여러 사용자의 상이한 응용이나 요구를 지원해준다. - 사용자의 데이터 관리 간단해짐(사용자 데이터 관리 용이) - 접근제어를 통한 자동보안이 제공됨 - 데이터 보관 용이 	<ul style="list-style-type: none"> - 독립적인 인덱스 가질수 없다. - 뷰의 정의 변경불가 - 뷰로 구성된 내용에 대한 삽입, 삭제, 갱신, 연산에 제약 따름

093 분산 DB 설계

분산 DB의 목표

1. Location Transparency, 위치 투명성 : 액세스하려는 DB의 실제 위치를 알 필요 없이, 단지 DB의 논리적인 명칭만으로 액세스 할 수 있다.
2. Replication Transparency 중복 투명성 : 동일 데이터가 여러 곳에 중복되어 있더라도, 사용자는 마치 하나의 데이터만 존재하는 것처럼 사용하고, 시스템은 자동으로 여러 자료에 대한 작업을 수행한다.
3. Concurrency Transparency 병행 투명성 : 분산 DB와 관련된 다수의 트랜잭션들이 동시에 실행되더라도 그 트랜잭션의 결과는 영향을 받지 않는다
4. Failure Transparency 장애 투명성 : 트랜잭션, DBMS, 네트워크, 컴퓨터 장애에도 불구하고 트랜잭션을 정확히 처리한다. 데이터 무결성이 보장된다.

099 2020 1회 42번

논리 데이터 모델의 물리 데이터 모델 변화

1. 테이블(Table)
 - A. 데이터를 정의하는 DB의 가장 기본적인 오브젝트(객체)
 - B. 컬럼(행, Column), 로우(열, row)로 구성,
 - C. 컬럼에는 지정된 유형에 따라 데이터가 저장된다.
 - D. 테이블 구성요소

row 로우, 열	튜플, 인스턴스, 어커런스
Column 컬럼, 행,	각 속성 항목에 대한 값을 저장
Primary Key 기본키	<ul style="list-style-type: none"> - 후보키에서 선택한 주키(main key) - 한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성
Foreign Key 외래키	<ul style="list-style-type: none"> - 다른 릴레이션의 기본키를 참조하는 속성 or 속성들의 집합 - 한 릴레이션에 속한 속성 a와 참조 릴레이션의 기본키인 b가 동일한 도메인 상에서 정의되어 있을 때, 속성 a를 외래키라함.

101 2020 1회 41번

SQL의 개념

- Structured Query Language
 - 국제표준 데이터베이스 언어.
 - 관계형 데이터베이스(RDB) 지원하는 언어
 - 관계대수와 관계해석을 기초로 한 혼합 데이터언어
 - 질의어지만, 데이터 구조의 정의, 데이터조작, 데이터 제어기능 갖춘.
1. DDL (Date Define Language, 데이터 정의어)
 - A. Schema, domain, table, view, index 정의 및 변경, 삭제
 - B. 논리적 데이터구조와 물리적 데이터구조의 사상 정의
 - C. DB관리자나, DB 설계자가 사용
 - D. DDL 3가지 유형
 - i. CREATE : Schema, domain, table, view, index 정의
 - ii. ALTER : table에 대한 정의변경하는데 사용
 - iii. DROP : Schema, domain, table, view, index 삭제
 2. DML (Date Mainpulation Language, 데이터 조작어)
 - A. DB 사용자가 응용프로그램이나, 질의어를 통하여 저장된 데이터를 실질적으로 처리하는데 사용되는 언어.
 - B. DB 사용자와 DB 관리 시스템 간의 인터페이스 제공
 - C. DML 4가지 유형
 - i. SELECT : 테이블에서 조건에 맞는 튜플 검색
 - ii. INSERT : 테이블에 새로운 튜플 삽입
 - iii. DELETE : 테이블에서 조건에 맞는 튜플 삭제
 - iv. UPDATE : 테이블에서 조건에 맞는 튜플내용 변경
 3. DCL (Date control language, 데이터 제어어)
 - A. 데이터의 보안, 무결성, 회복, 병행 수행 제어를 정의하는데 사용되는 언어
 - B. DB 관리자가 데이터 관리 목적으로 사용
 - C. DCL 4가지 종류
 - i. COMMIT : 명령에 대한 수행된 결과를 실제 물리적 디스크로 저장, DB 조작 작업이 정상적으로 완료되었음을 관리자에게 알림
 - ii. ROLLBACK : DB 조작 작업이 비정상적으로 종료되었을 때, 원래상태로 복구
 - iii. GRANT : DB 사용자에게 사용권한 부여
 - iv. REVOKE : DB 사용자의 사용권한 취소

102 2020 1회 54번

참조 무결성의 CASCADE 법칙

- 참조 무결성 제약이 설정된 기본 테이블의 어떤 데이터(부모테이블)를 삭제할 경우, 그 데이터와 밀접하게 연관된 다른 테이블(자식테이블)의 데이터도 자동으로 삭제되는 옵션

106 2020 2회 56번

SQL의 집계함수(Aggregaation function)

Count, Sum, Avg, Max, Min, Stddev, Variance, Rollup, cube,

108 53번 프로시저 : 절차형 SQL을 활용하여 특정 기능을 수행하는 일종의 트랜잭션 언어.

109 53번 트리거 : DB 시스템에서 데이터의 삽입, 갱신, 삭제 등의 이벤트가 발생할 때마다 관련작업이 자동으로 수행되는 절차형 SQL

124 2020 2회 68**배치프로그램 필수요소**

대용량 데이터	대량의 데이터를 가져오거나, 전달, 계산하는 등의 처리 가능해야함
자동화	심각한 오류 발생 상황 제외하고는 사용자의 개입 없이 수행되어야함
견고성	잘못된 데이터나 데이터 중복등의 상황으로 중단되는 일 없이 수행되어야함
안정성/신뢰성	오류가 발생하면, 오류의 발생위치, 시간 등 추적가능할 수 있어야함 어떤문제가 생겼는지, 언제 발생했는지 추적
성능	다른 응용 프로그램의 수행을 방해하지 않아야하고, 지정된 시간 내에 처리가 완료 되어야함.

배치(replacement) 전략 = 영역크기- 할당크기 / 공백인지 사용중인지 확인 [144]

1. First fit : 빈 영역 중에서 첫번째 분할 영역에 배치시키는 방법
2. Best fit : 단편화를 가장 작게 남기는 분할 영역에 배치시키는 방법
3. Worst fit : 단편화를 가장 많이 남기는 분할 영역에 배치시키는 방법

128 2020 1회 62번**비트연산자 (& ^ | ~ << >>)**

&	And	모든 비트가 1일때만 1
^	Xor	모든 비트가 같으면 0, 하나라도 다르면 1
	Or	모든 비트 중 한 비트라도 1이면 1
~	Not	각 비트의 부정, 0이면 1, 1이면 0
<<	왼쪽 시프트	비트를 왼쪽으로 이동
>>	오른쪽 시프트	비트를 오른쪽으로 이동

135 2020 1회 76번**스크립트언어**

1. 서버용 스크립트 언어 : ASP, JSP, PHP, 파이썬
2. 클라이언트용 스크립트 언어 : 자바 스크립트.

141 2020 2회 67**운영체제 OS Operating System**

- 컴퓨터 시스템의 자원들을 효율적으로 관리하며, 사용자가 컴퓨터를 편리하고 효과적으로 사용할 수 있도록 환경을 제공하는 여러 프로그램의 모임
- 사용자와 하드웨어 간의 인터페이스
- 동작하는 시스템 sw의 일종, 다른 응용 프로그램이 유용한 작업을 할 수 있도록 환경 제공,

기능

- 프로세서(처리기, processor), 기억장치(주기억장치, 보조기억장치) 입출력장치, 파일 및 정보 등의 자원관리
- 자원의 스케줄링 기능 제공
- 시스템의 각종 하드웨어와 네트워크 관리 및 제어
- 데이터 관리, 데이터 및 자원의 공유 기능 제공
- 시스템의 오류 검사 및 복구,
- 다중 사용자와 다중 응용 프로그램 환경하에서 자원의 현재 상태를 파악하고 자원 분배를 위한 스케줄링을 담당한다.
- Cpu, 메모리 공간, 기억장치, 입출력 장치 등의 자원 관리
- 입출력 장치와 사용자 프로그램 제어

- 이미 할당된 cpu를 다른 프로세스가 강제로 빼앗아 사용할 수 없는 스케줄링 기법
- 프로세스가 cpu를 할당 받으면, 해당 프로세스가 완료될 때 까지 cpu 사용
- 모든 프로세스에 대한 요구 공정하게 처리
- 프로세스 응답 시간 예측이 용이. 일괄 처리 방식에 적합
- 중요한 작업(짧은 작업)이 중요하지 않은 작업(긴 작업)을 기다리는 경우 발생
- 종류 : FCFS, SJF, HRN, 우선순위, 기한부 알고리즘

HRN

대기 시간이 긴 프로세스의 경우, 우선 순위가 높아진다.

SJF를 보완하기 위한 방식

긴 작업과 짧은 작업 간의 지나친 불평등을 해소 할 수 있다.

151 운영체제 기본 명령어

1. UNIZ/LINUX 기본 명령어

Cat	파일 내용을 화면에 표시
Chdir	현재 사용할 디렉터리의 위치 변경
Chmod	파일의 보호모드 설정하여 파일의 사용허가 지정
Chown	소유자 변경
Cp	파일 복사
Exec	새로운 프로세스 수행
Find	파일 찾을
Fork	새로운 프로세스 생성
Fsck	파일 시스템 검사 및 보수
Getpid	자신의 프로세스 아이디 얻는다
Getppid	부모 프로세스 아이디 얻는다
Is	현재 디렉터리 내의 파일 목록 확인
Mount/unmount	파일 시스템을 마운팅 한다/ 마운팅 해제
Rm	파일삭제
Wait	fork후에 exec에 의해 실행되는 프로세스의 상위 프로세스가 하위 프로세스 종료 등의 이벤트 기다림

152 2020 1회 61

IPv6

- ① IPv4의 주소 부족 문제를 해결하기 위해 개발
- ② 128비트의 긴 주소 사용(16비트씩 8부분), IPv4보다 자료전송속도가 빠르고, 보안성 강화
- ③ 인증성/기밀성/데이터 무결성의 지원으로 보안문제를 해결할 수 있다.
- ④ IPv4와 호환성이 뛰어나다.
- ⑤ 주소의 확장성, 융통성, 연동성이 뛰어나며, 실시간 흐름제어로 향상된 멀티미디어 기능을 지원한다.
- ⑥ 멀티미디어의 실시간 처리 가능
- ⑦ Traffic Class, Flow Label을 이용하여 등급별, 서비스별로 패킷을 구분할 수 있어 품질 보장 용이.
- ⑧ IPv6 확장 헤더를 통해 네트워크 기능 확장이 용이하다.
- ⑨ 각 부분을 16진수로 표현하고 콜론(:)으로 구분한다.
- ⑩ 자동으로 네트워크 환경구상이 가능.
- ⑪ 3가지 주소 체계
 - A. Unicast 유니캐스트 : 단일 송신자와 단일 수신자 간의 통신 1:1
 - B. Multicast 멀티캐스트 : 단일 송신자와 다중 수신자 간의 통신 1:다
 - C. Anycast 애니캐스트 : 단일 송신자와 가장 가까이 있는 단일 수신자 간의 통신 1:1

153 2020 1회 68

OSI 7계층

물리 계층 Physical Layer	<ul style="list-style-type: none"> - 전송에 필요한 두 장치 간의 실제접속, 절단 등 기계적, 전기적, 기능적, 절차적 특성에 대한 규칙 정의 - 물리적 전송 매체와 전송 신호 방식을 정의 - RS-232C, X.21 - 관련 장비 : 리피터, 허브
데이터링크 계층 Data Link Layer	<ul style="list-style-type: none"> - 두 개의 인접한 개방 시스템들 간에 신뢰성 있고, 효율적인 정보 전송을 할 수 있도록 함 - 송신측과 수신측의 속도 차이를 해결하기 위한 흐름 제어 기능 - 프레임의 시작과 끝을 구분하기 위한 프레임의 동기화 기능 - 오류의 검출과 회복을 위한 오류제어기능 - 프레임의 순서적 전송을 위한 순서제어기능 - HDLC, LAPB, LLC, MAC, LAPD, PPP 등의 표준 - 관련장비 : 랜카드, 브리지, 스위치
네트워크 계층 Network Layer, 망 계층	<ul style="list-style-type: none"> - 개방 시스템들 간의 네트워크 연결을 관리하는 기능과 데이터의 교환 및 중계 기능을 한다. - 네트워크 연결을 설정, 유지, 해제하는 기능 - 경로설정(Routing), 데이터 교환 및 중계, 트래픽 제어, 패킷정보 전송 수행 - X.25, IP등의 표준 - 관련 장비 : 라우터
전송 계층 Transport layer	<ul style="list-style-type: none"> - 논리적 안정과 균일한 데이터 전송 서비스를 제공함으로써 종단 시스템(END TO END) 간에 투명한 데이터 전송 가능하게 한다. - 상 하위 3계층 간의 인터페이스 담당 - 종단시스템 간의 전송 연결 설정, 데이터 전송, 연결 해제 기능 - 주소 설정, 다중화(분할 및 재조립), 오류 제어, 흐름제어 - TCP, UDP 등의 표준 - 관련 장비 : 게이트웨이
세션 계층 Session layer	<ul style="list-style-type: none"> - 송 수신 측 간의 관련성을 유지하고 대화제어 담당 - 대화(회화) 구성 및 동기 제어, 데이터 교환 관리 기능 - 송수신 측 간의 대화 동기를 위해, 저송하는 정보의 일정한 부분에 체크점을 두어 정보의 수신 상태 체크, 이때의 체크점을 동기점이라 한다 - 동기점은 오류가 있는 데이터의 회복을 위해 사용하는 것으로 종류에는 소동기점, 대동기점이 있다.
표현 계층 Presentation layer	<ul style="list-style-type: none"> - 응용 계층으로부터 받은 데이터를 세션계층에 보내기 전에 통신에 적당한 형태로 변환하고, 세션계층에서 받은 데이터는 응용 계층에 맞게 변환하는 기능 - 서로 다른 데이터 표현 형태를 갖는 시스템 간의 상호 접속을 위해 필요한 계층 - 코드 변환, 데이터 암호, 데이터 압축, 구문 검색, 정보 형식(포맷) 변환 문맥관리 기능을 한다
응용 계층 Application Layer	<ul style="list-style-type: none"> - 사용자(응용 프로그램)가 OSI 환경에 접근할 수 있도록 서브슬르 제공 - 응용 프레스 간의 정보교환, 전자 사서함, 파일 전송, 가상 터미널 등의 서비스 제공

156 2020 1회 63**TCP 프로토콜**

- ① OSI 7계층의 전송 계층에 해당
- ② 신뢰성 있는 연결형 서비스 제공(연결 지향형 전달서비스)
- ③ 패킷의 다중화, 순서 제어, 오류 제어, 흐름 제어기능 제공
- ④ 스트림(Stream) 전송 기능 제공
- ⑤ TCP 헤더는 Source/Destination port number, Sequence number, Acknowledgement number, Checksum, 등이 포함됨

Ip 프로토콜**TCP/IP 주요 프로토콜**

응용계층 프로토콜	응용 프로그램 간의 데이터 송수신 제공	<ul style="list-style-type: none"> ① FTP ② SMTP ③ TELNET ④ SNMP ⑤ DNS ⑥ HTTP
전송계층 프로토콜	호스트들 간의 신뢰성 있는 통신 제공	<ul style="list-style-type: none"> ① TCP ② UDP ③ RTP
인터넷계층 프로토콜	데이터 전송을 위한 주소 지정, 경로설정 제공	<ul style="list-style-type: none"> ① IP ② ICMP ③ IGMP ④ ARP (IP주소를 MAC주소로 변환) ⑤ RARP
네트워크 액세스 프로토콜	실제 데이터(프레임)을 송수신하는 역할	<ul style="list-style-type: none"> ① Ethernet (IEEE 802.3) ② IEEE 802 -> mac 기능 지원 802.11e ③ HDLC ④ X.25 ⑤ RS-232C

161 2020 1회 81**수학적 산정 기법****1. Putnam 모형 (푸트남 모형)**

- A. Rayleigh – Norden 곡선의 노력 분포도 이용한 프로젝트 비용 산정방법,
- B. = 생명 주기 예측 모형
- C. 자동화 추정 도구 : SLIM

2. cocomo 모형**A. Cocomo 소프트웨어 개발 유형**

- ① 조직형 Organic Model – 5만, 50KDSI 이하, 중소규모의 소프트웨어 (일괄자료처리,과학기술계산용)
- ② 반분리형 (Semi-Detached Model) – 30만, 300KDSI 이하, 트랜잭션 처리시스템, 운영체제, DB관리시스템
- ③ 내장형 (Embedded Model) – 30만, 300KDSI 이상, 신호기 제어 시스템, 미사일 유도 시스템, 실시간 처리 시스템

B. Cocomo 모형의 종류

- ① 기본형 (Basic)

② 중간형 (Intermediate)

③ 발전형 (Detailed)

3. 기능점수 모형 (Functional Point)

A. 알브레히트가 제안

B. 가중치

① 자료 입력/입력 양식

② 정보 출력/출력 보고서

③ 명령어 / 사용자 질의수

④ 데이터 파일

⑤ 필요한 외부 루틴과의 인터페이스

163 2020 1회 85번

소프트웨어 개발 표준

1. CMMI (Capability Maturity Model Intergration)의 단계

- 초기, 관리, 정의, 정량적 관리(Quantitatively managed), 최적화(optimizing)

2. ISO/IEC 12207

3. SPICE – SW 개발 표준 중 SW 품질 및 생산성 향상을 위해 SW 프로세스를 평가 및 개선하는 국제 표준

164 2020 1회 97번

소프트웨어 개발 방법론 테일러링

1. 내부기준 – 목표환경, 요구사항, 프로젝트 규모(비용, 기간, 인력), 보유기술(프로세스, 개발방법론)

2. 외부기준 – 법적 제약사항, 표준 품질 기준

165 2020 1회 88번

소프트웨어 개발 프레임워크

1. 기대효과 – 품질보증, 개발 용이성, 변경 용이성

166 2020 1회, 90번

네트워크 관련 신기술

1. PICONET 피코넷 – 블루투스기술이나 UWB 통신 기술을 사용하여 통신망 형성하는 무선 네트워크기술

2. WI-SUN 와이선 – 장거리 무선 통신을 필요로 하는 사물 인터넷(IoT) 서비스를 위한 저전력 장거리(LPWA) 통신기술

3. NFC, 근거리 무선통신 – 고주파(HF) 이용한 근거리 무선 통신

4. Mech Network, 메시 네트워크 – 기존 무선 랜의 한계 극복, 대규모 디바이스의 네트워크 생성에 최적화, 차세대 이동통신, 홈네트워킹, 공공안전 등의 특수목적에 위한 새로운 방식의 네트워크 기술,

169 2020 1회. 95번

경로제어(라우팅) 프로토콜 (Routing Protocol)

1. IGP [Interior Gateway Protocol, 내부 게이트웨이 프로토콜] : 하나의 자율시스템(AS)내의 라우팅에 사용되는 프로토콜

A. RIP(Routing Information Protocol) : 소규모, 최대 홉수 15로 제한, 거리 벡터 라우팅 프로토콜, 최단 경로탐색에는 Bellman-ford 알고리즘 사용

B. OSPF(Open Shortest Path First Protocol) : 대규모

2. EGP, Exterior Gateway Protocol, 외부 게이트웨이 프로토콜

3. BDP, Border Gateway Protocol

170 2020 2회 81

SW 관련 신기술

1. 디지털 트윈 (digital Twin)

A. 현실속 사물(모델)을 소프트웨어 가상화한 모델

- B. 실제 자산의 특성에 대한 정확한 정보 얻음
- C. 자산 최적화, 돌발사고 최소화, 생산성 증가 등 설계부터 제조, 서비스에 이르는 모든 과정의 효율성을 향상 시킬 수 있다.

176 2020 1회 92번**하둡(Hadoop)**

- 오픈 소스를 기반으로 한 분산 컴퓨팅 플랫폼
- 일반 pc급 컴퓨터들로 가상화된 대형 스토리지 형성
- 그 안에 보관된 거대한 데이터 세트를 병렬로 처리할 수 있도록 개발된 자바 sw 프레임워크
- 구글. 야후에서 사용
- 다양한 소스를 통해 생성된 빅데이터를 효율적으로 저장 및 처리

179 2020 1회 100번**보안요소 / 정보보안의 3요소 (차례대로 3개)**

기밀성	- 시스템 내의 정보와 자원은 인가된 사용자에게만 접근허용 - 정보가 전송 중에 노출되더라도 데이터를 읽을 수 없습니다
무결성	- 시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있다
가용성	- 인가 받은 사용자는 언제라도 사용가능
인증	- 시스템 내의 정보와 자원을 사용하려는 사용자가 합법적인 사용자인지를 확인하는 모든 행위 - 패스워드 인증용 카드, 지문검사가 해당됨
부인 방지	- 데이터를 송수신한 자가 송수신 사실을 부인할 수 없도록 송수신 증거를 제공

186 2020 1회.80**JAVA 언어 접근 지정자(접근 제한자)**

- Public, Protected, Default, Private

188 2020 1회 93

공개키 암호화 기법(Public key Encryption) – RSA

개인키 암호화 기법(Private key Encryption) – 블록 암호화 방식 : DES. SEED. AES. ARIA

- 스트림 암호화 방식 : LFSR, RC4

189 2020 1회 84번**서비스 공격 유형****1. 서비스 거부 DDos, Denial of Service**

- A. 서비스 자원을 고갈 시킬 목적으로 다수의 공격자나 시스템에서 대량의 데이터를 한곳의 서버에 집중적으로 전송하여 표적 서버의 정상적인 기능을 방해하는 것
- B. Trin00, TFN(Tribe Flooding Network), Stacheldraht, TFN2k

2. Ping of Death 죽음의 핑

- A. Ping 명령을 전송할 때, 패킷의 크기를 인터넷 프로토콜 허용버머위 이상으로 전송하여 공격 대상의 네트워크를 마비시키는 서비스 공격 방법

3. Smurfing 스머핑 :

- A. Ip나 ICMP의 특성을 악용하여 특정 사이트에 집중적으로 데이터를 보냄으로써 네트워크 불능 상태로 만드는 공격방법

정보보안 침해공격 관련 용어

1. 키로거 공격 (Key logger Attack) : 컴퓨터 사용자의 키보드 움직임을 감지해, id, 패스워드 등 개인의 중요한 정보를 몰래 빼돌리는 해킹 공격
2. 랜섬웨어 (Ransomware) : 인터넷 사용자의 컴퓨터에 침입해 내부 문서나 파일을 암호화해 사용자가 열지 못하게 하는 프로그램, 암호 해독용 프로그램의 전달을 조건으로 사용자에게 돈 요구

기타 정리 = 책에 없던 부분

1. 데이터 흐름도(DFD)=자유흐름도의 구성요소

Process, data flow, data store, terminator

2. 소프트웨어 설계시 구축된 플랫폼의 성능특성 분석에 사용되는 측정항목

- 응답시간(response time), 가용성(availability), 사용률(utilization)

3.UML 확장 모델에서 스테레오 타입 객체 표현 - <<>>

4. 자료사전

- 자료 생략 ()
- 자료 반복 { }
- 자료 정의 =
- 자료 연결 +
- 자료 선택 [|]
- 자료 설명 **

5. 린바우 객체지향 분석 절차

객체모형 -> 동적 모형 -> 기능모형

6. 객체지향 기법에서 클래스들 사이의 부분-전체 관계, 부분의 관계로 설명되는 연관성 = 집단화

7. HIPO(Hierarchy input process output)

- Hipo 차트 종류에는 가시적 도표, 총제적 도표, 세부적 도표가 있다.
- 기능과 자료의 의존 관계를 동시에 표현할수 있다.
- 보기 쉽고 이해쉬움
- 응집 응집도

8. E-R 다이어그램을 사용하여 객체의 행위를 모델링하며, 객체식별, 구조식별, 주체 정의, 속성 정의, 서비스 정의 등의 과정으로 구성되는 것 = coad and Yourdon 방법

9. 정렬된 N개의 데이터를 처리하는데 o의 시간이 소요되는 정렬 알고리즘 - 합병정렬

10. Pareto법칙 - sw 테스트에서 오류의 80%는 전체 모듈의 20%내에서 발견된다는 법칙

11. 외계인 코드(alien code) - 아주 오래되거나 참고문서or개발자가 없어 유지보수 작업이 어려운 프로그램

12. 병행제어의 로킹(locking)단위

a. 로킹 단위 개념

- DB, 파일, 레코드
- 한꺼번에 로킹 할 수 있는 객체의 크기
- 로킹의 대상이 되는 객체의 크기

b. 로킹 단위가 작아지면 DB 공유도가 증가, 로킹 오버헤드 증가

13. 은행가 알고리즘(banker's algorithm) 의 교착 상태의 해결방안 - avoidance

14. 교착 상태 발생의 필요충분조건 - 상호배제, 점유와 대기, 환형대기

15. 스택가드 - 메모리상에서 프로그램의 복귀 주소와 변수사이에 특정 값을 저장해두었다가 그값이 변경되었을 경우, 오버플로우 상태로 가정하여 프로그램 실행을 중단하는 기술

16. 백도어 탐지 방법 - 무결성검사, 로그분석, setUID 파일 검사

17. tripwire : 크래커가 침입하여 백도어를 만들어 놓거나, 설정 파일을 변경했을 때 분석하는 도구

18. 컴포넌트 설계시, '협약에 의한 설계' contract

19. 객체지향 sw 설계시, 디자인 패턴 구성 요고 - 문제 및 배경 ,사례, 샘플코드

20. LSP(Liskov Substition Prinple) : 객체지향 설계원칙 중, 서브타입은 어디에서나 자신의 기반 타입으로 교체할 수 있다

21. sw 공학의 기본 원칙

- 품질 높은 sw 상품 개발
- 지속적인 검증 시행
- 결과에 대한 명확한 기록유지

22. iso/iec 12119 : 패키지 sw의 일반적인 제품 품질 요구사항 및 테스트를 위한 국제표준

23. 알고리즘 설계 기법

- Divide and conquer, Greedy, Backtracking

24. 소프트웨어 재공학이 소프트웨어 재개발에 비해 갖는 장점

- 위험부담 감소, 비용절감, 시스템 명세의 오류억제

25. 즉각 갱신 기법 : DB 로그를 필요로 하는 회복기법

26. 버퍼 오버플로 : 메모리를 다루는데 오류가 발생하여 잘못된 동작을 하는 프로그램 취약점.

27. 파이썬의 변수작성 규칙 설명

- 첫 자리에 숫자 사용 불가
- 영문 대문자/소문자, 숫자, 밑줄(_) 사용 가능
- 이미 사용되고 있는 예약어 사용 불가

28. mashup – 웹에서 제공하는 정보 및 서비스를 이용하여 새로운 sw나 서비스, DB를 만드는 기술

29. Date Mining – 빅데이터 분석 기술 중 대량의 데이터를 분석하여 데이터 속에 내재되어 있는 변수 사이의 상호관례를 규명하여 일정한 패턴을 찾아내는 기법

30. 합성 중심 : 전자칩과 같은 소프트웨어 부품, 즉 블록(모듈)을 만들어서 끼워 맞추는 방법으로, 소프트웨어를 완성시키는 재사용 방법이다.

31. SDLC에 통합한 sw 개발 보안 생명 주기 방법론 = save touchpoint