1과목 소프트웨어 설계

001 소프트웨어 공학의 기본 원칙

- 현대적인 프로그래밍 기술을 계속적으로 적용해야 한다.
- 개발된 소프트웨어의 품질이 유지되도록 지속적으로 검증해야 한다.
- 소프트웨어 개발 관련 사항 및 결과에 대한 명확한 기록을 유지해야 한다.

002 폭포수 모형

- 이전 단계로 돌아갈 수 없다는 전제하에 각 단계를 확실히 매듭짓고 다음 단계를 진행하는 개발 방법론이다.
- 보헴이 제시한 고전적 생명 주기 모형이다.
- 요구사항을 반영하기 어렵다.

003 나선형 모형

- 나선을 따라 돌듯이 점진적으로 완벽한 최종 소프트웨어를 개발하는 것이다.
- '계획 수립 → 위험 분석 → 개발 및 검증 → 고객 평가' 과정이 반복적으로 수행된다.

004 애자일 모형의 주요 방법론

- 스크럼(Scrum)
- XP(eXtreme Programming)
- 기능 중심 개발(FDD; Feature Driven Development)
- 칸반(Kanban)
- Lean

005 애자일 개발 4가지 핵심 가치

- 프로세스와 도구보다는 개인과 상호작용에 더 가치를 둔다.
- 방대한 문서보다는 실행되는 SW에 더 가치를 둔다.
- 계약 협상보다는 고객과 협업에 더 가치를 둔다.
- 계획을 따르기 보다는 변화에 반응하는 것에 더 가치를 둔다.

006 XP의 핵심 가치

- 의사소통(Communication)
- 단순성(Simplicity)
- 용기(Courage)
- 존중(Respect)
- 피드백(Feedback)

007 주요 비기능 요구사항

- 성능 요구사항
- 보안 요구사항
- 품질 요구사항
- 제약사항
- 인터페이스 요구사항

008 요구사항 개발 프로세스

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

009 요구사항 분석

- 개발 대상에 대한 사용자의 요구사항을 이해하고 문서화(명세화)하는 활동을 의미한다.
- 소프트웨어 개발의 실제적인 첫 단계이다.
- 사용자 요구의 타당성을 조사하고 비용과 일정에 대한 제약을 설정한다
- 사용자의 요구를 정확하게 추출하여 목표를 정하고, 해결 방식을 결정한다.

010 자료 흐름도의 구성 요소

기호	표기법
프로세스(Process)	물품 확인
자료 흐름(Data Flow)	물품 코드 →
자료 저장소(Data Store)	물품대장
단말(Terminator)	공장

011 자료 사전의 표기 기호

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

012 HIPO

- 하향식 소프트웨어 개발을 위한 문서화 도구이다.
- 기호, 도표 등을 사용하므로 보기 쉽고 이해하기도 쉽다.
- 기능과 자료의 의존 관계를 동시에 표현할 수 있다.

013 UML

- 시스템 개발자와 고객 또는 개발자 상호 간의 의사소통이 원활하게 이루어지도록 표준화한 대표적인 객체지향 모델링 언어이다.
- 구성 요소 : 사물(Things), 관계(Relationships), 다이어그램(Diagram)

014 UML의 주요 관계

- 일반화(Generalization) 관계 : 하나의 사물이 다른 사물에 비해 더 일반적인지 구체적인지를 표현
- 의존(Dependency) 관계 : 필요에 의해 서로에게 영향을 주는 짧은 시간 동안만 연관을 유지하는 관계를 표현
- 실체화(Realization) 관계 : 사물이 할 수 있거나 해야 하는 기능으로 서로를 그룹화 할 수 있는 관계를 표현

015 구조적(Structural) 다이어그램의 종류

- 클래스 다이어그램(Class Diagram)
- 객체 다이어그램(Object Diagram)
- 컴포넌트 다이어그램(Component Diagram)
- 배치 다이어그램(Deployment Diagram)
- 복합체 구조 다이어그램(Composite Structure Diagram)
- 패키지 다이어그램(Package Diagram)

016 행위(Behavioral) 다이어그램의 종류

- 유스케이스 다이어그램(Use Case Diagram)
- 순차 다이어그램(Sequence Diagram)
- 커뮤니케이션 다이어그램(Communication Diagram)
- 상태 다이어그램(State Diagram)
- 활동 다이어그램(Activity Diagram)
- 상호작용 개요 다이어그램(Interaction Overview Diagram)
- 타이밍 다이어그램(Timing Diagram)

017 스테레오 타입

- UML에서 표현하는 기본 기능 외에 추가적인 기능을 표현하기 위해 사용한다.
- 길러멧(Guilemet)이라고 부르는 겹화살괄호(《 》) 사이에 표현할 형태를 기술한다.

018 유스케이스 다이어그램 - 액터(Actor)

- 시스템과 상호작용을 하는 모든 외부 요소로, 사람이나 외부 시스템을 의미한다.
- **주액터** : 시스템을 사용함으로써 이득을 얻는 대상으로, 주로 사람이 해당함
- **부액터** : 주액터의 목적 달성을 위해 시스템에 서비스를 제공하는 외부 시스템으로, 조직이나 기관 등이 될 수 있음

019 순차(Sequence) 다이어그램의 구성 요소

- 액터(Actor)
- 객체(Object)
- 생명선(Lifeline)
- 실행 상자(Active Box)
- 메시지(Message)

020 사용자 인터페이스의 특징

- 사용자의 편리성과 가독성을 높여준다.
- 작업 시간을 단축시킨다.
- 업무에 대한 이해도를 높여준다.
- 사용자 중심으로 설계되어 있다.

021 사용자 인터페이스의 구분

- CLI(Command Line Interface) : 명령과 출력이 텍스트 형태로 이뤄지는 인터페이스
- GUI(Graphical User Interface) : 아이콘이나 메뉴를 마우스로 선택하여 작업을 수행하는 그래픽 환경의 인터페이스
- NUI(Natural User Interface) : 사용자의 말이나 행동으로 기기를 조작하는 인터페이스

022 사용자 인터페이스의 기본 원칙

- **직관성** : 누구나 쉽게 이해하고 사용할 수 있어야 한다.
- **유효성** : 사용자의 목적을 정확하고 완벽하게 달성해야 한다.
- **학습성** : 누구나 쉽게 배우고 익힐 수 있어야 한다.

023 목업

- 와이어프레임보다 좀 더 실제 화면과 유사하게 만든 정적인 형태의 모형이다.
- 시각적으로만 구성 요소를 배치하는 것으로 실제로 구현되지는 않는다.

024 ISO/IEC 9126의 품질 특성

- 기능성(Functionality) : 요구사항을 정확하게 만족하는 기능을 제공하는지 여부를 나타냄
- 신뢰성(Reliability) : 요구된 기능을 오류 없이 수행할 수 있는 정도를 나타냄
- 사용성(Usability) : 사용자가 쉽게 배우고 사용할 수 있는 정도를 나타냄
- 이식성(Portability) : 다른 환경에서도 얼마나 쉽게 적용할 수 있는지 정도를 나타냄

025 소프트웨어 아키텍처의 설계 과정

설계 목표 설정 → 시스템 타입 결정 → 아키텍처 패턴 적용 → 서브시스템 구체화 → 검토

026 모듈화

- 기능의 분리가 가능하여 인터페이스가 단순해진다.
- 프로그램의 효율적인 관리가 가능하다.
- 오류의 파급 효과를 최소화할 수 있다.
- 모듈의 크기를 너무 작게 나누면 개수가 많아져 모듈간의 통합 비용이 많이 들고, 너무 크게 나누면 개수가 적어 통합 비용은 적게 들지만 모듈 하나의 개발 비용이 많이 든다.

027 추상화의 유형

- 과정 추상화
- 데이터(자료) 추상화
- 제어 추상화

028 정보 은닉

- 한 모듈 내부에 포함된 절차와 자료들의 정보가 감추어져 다른 모듈이 접근하거나 변경하지 못하도록 하는 기법이다.
- 모듈을 독립적으로 수행할 수 있다.
- 수정, 시험, 유지보수가 용이하다.
- 정보 은닉을 표기할 때 private의 의미는 은닉이다.

029 파이프 - 필터 패턴

- 시스템의 처리 결과물을 파이프를 통해 전달받아 처리한 후 그 결과물을 다시 파이프를 통해 다음 시스템으로 넘겨주는 패턴이다.
- 데이터 변환으로 인한 오버헤드가 발생한다.

030 MVC (Model-View-Controller) 패턴

- 모델(Model) : 서브시스템의 핵심 기능과 데이터를 보관함
- 뷰(View) : 사용자에게 정보를 표시함
- 컨트롤러(Controller) : 사용자로부터 입력된 변경 요청을 처리하기 위해 모델에게 명령을 보냄

031 메시지(Message)

- 객체에게 어떤 행위를 하도록 지시하는 명령 또는 요구사항이다.
- 객체들 간에 상호 작용을 하는 데 사용되는 수단이다.

032 초치기 클래스(Class)

- 공통된 속성과 연산(행위)을 갖는 객체의 집합이다.
- 클래스에 속한 각각의 객체를 인스턴스(Instance)라 한다.
- 객체지향 프로그램에서 데이터를 추상화하는 단위이다.

033 초치기 캡슐화(Encapsulation)

- 데이터와 데이터를 처리하는 함수를 하나로 묶는 것을 의미한다.
- 외부 모듈의 변경으로 인한 파급 효과가 적다.
- 인터페이스가 단순화된다.
- 재사용이 용이하다.

034 초치기 상속(Inheritance)

상위 클래스(부모 클래스)의 모든 속성과 연산을 하위 클래스(자식 클래스)가 물려받는 것이다.

035 초치기 다형성(Polymorphism)

- 오버로딩(Overloading) : 메소드의 이름은 같지만 인수를 받는 자료형과 개수를 달리하여 여러 기능을 정의할 수 있음
- 오버라이딩(Overriding) : 메소드의 이름은 같지만 메소드 안의 실행 코드를 달리하여 자식 클래스에서 재정의해서 사용할 수 있음

036 초치기 객체지향 분석 방법론 - Coad와 Yourdon 방법

- E-R 다이어그램을 사용하여 객체의 행위를 모델링한다.
- 객체 식별, 구조 식별, 주제 정의, 속성과 인스턴스 연결 정의, 연산과 메시지 연결 정의 등의 과정으로 구성하는 기법이다.

037 초치기 럼바우(Rumbaugh)의 분석 기법

- 객체(Object) 모델링 : 정보 모델링이라고도 하며, 객체들 간의 관계를 규정하여 객체 다이어그램으로 표시하는 것
- 동적(Dynamic) 모델링 : 상태 다이어그램을 이용하여 객체들 간의 동적인 행위를 표현하는 모델링
- 기능(Functional) 모델링 : 자료 흐름도를 이용하여 자료 흐름을 표현한 모델링

038 초치기 객체지향 설계 원칙(SOLID 원칙)

- 단일 책임 원칙(SRP; Single Responsibility Principle) : 객체는 단 하나의 책임만 가져야 한다는 원칙
- 개방-폐쇄 원칙(OCP; Open-Closed Principle) : 기존의 코드를 변경하지 않고 기능을 추가할 수 있도록 설계해야 한다는 원칙
- 리스코프 치환 원칙(LSP; Liskov Substitution Principle) : 자식 클래스는 최소한 자신의 부모 클래스에서 가능한 행위는 수행할 수 있어야 한다는 설계 원칙
- 인터페이스 분리 원칙(ISP; Interface Segregation Principle) : 자신이 사용하지 않는 인터페이스와 의존 관계를 맺거나 영향을 받지 않아야 한다는 원칙
- 의존 역전 원칙(DIP; Dependency Inversion Principle) : 각 객체들 간의 의존 관계가 성립될 때, 추상성이 낮은 클래스보다 추상성이 높은 클래스와 의존 관계를 맺어야 한다는 원칙

039 모듈(Module)

- 모듈화를 통해 분리된 시스템의 각 기능들이다.
- 단독으로 컴파일이 가능하다.
- 재사용 할 수 있다.
- 다른 모듈에서의 접근이 가능하다.

040 결합도의 정도(약함 → 강함)

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

041 결합도의 종류

- 자료(Data) 결합도 : 모듈 간의 인터페이스가 자료 요소로만 구성될 때의 결합도
- 스탬프(Stamp) 결합도 : 배열이나 레코드 등의 자료 구조가 전달될 때의 결합도
- 제어(Control) 결합도 : 제어 신호를 이용하여 통신하거나 제어 요소를 전달하는 결합도
- 외부(External) 결합도 : 데이터(변수)를 외부의 다른 모듈에서 참조할 때의 결합도
- 공통(Common) 결합도 : 공유되는 공통 데이터 영역을 여러 모듈이 사용할 때의 결합도
- 내용(Content) 결합도 : 다른 모듈의 내부 자료를 직접 참조하거나 수정할 때의 결합도

042 응집도의 정도(약함 → 강함)

우연적 응집도 → 논리적 응집도 → 시간적 응집도 → 절차적 응집도 → 교환적 응집도 → 순차적 응집도 → 기능적 응집도

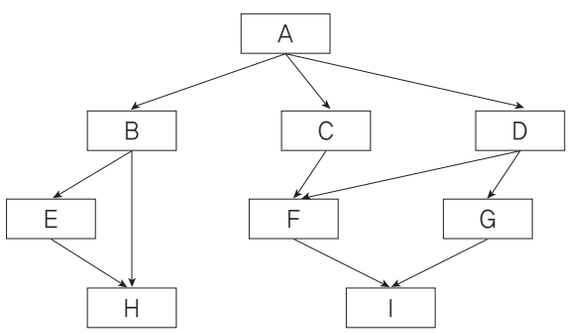
043 주요 응집도

- 절차적(Procedural) 응집도 : 모듈 안의 구성 요소들이 그 기능을 순차적으로 수행할 경우의 응집도
- 시간적(Temporal) 응집도 : 특정 시간에 처리되는 몇 개의 기능을 모아 하나의 모듈로 작성할 경우의 응집도
- 우연적(Coincidental) 응집도 : 각 구성 요소들이 서로 관련 없는 요소로만 구성된 경우의 응집도

044 팬인(Fan-In) / 팬아웃(Fan-Out)

- 팬인 : 어떤 모듈을 제어(호출)하는 모듈의 수
- 팬아웃 : 어떤 모듈에 의해 제어(호출)되는 모듈의 수

예제 다음의 시스템 구조도에서 각 모듈의 팬인(Fan-In)과 팬아웃(Fan-Out)을 구하시오.



해설

- **팬인(Fan-In)** : A는 0, B · C · D · E · G는 1, F · H · I는 2
- **팬아웃(Fan-Out)** : H · I는 0, C · E · F · G는 1, B · D는 2, A는 3

045 NS 차트

- 논리의 기술에 중점을 둔 도형을 이용한 표현 방법이다.
- 연속, 선택 및 다중 선택, 반복 등의 제어 논리 구조를 표현한다.
- GOTO나 화살표를 사용하지 않는다.
- 시각적으로 명확히 식별하는 데 적합하다.
- 이해하기 쉽고, 코드 변환이 용이하다.

046 재사용(Reuse)

- 이미 개발된 기능을 새로운 시스템이나 기능 개발에 사용할 수 있는 정도를 의미한다.
- **재사용 규모에 따른 분류** : 함수와 객체, 컴포넌트, 애플리케이션

047 효과적인 모듈 설계 방안

- 결합도는 줄이고 응집도는 높인다.
- 복잡도와 중복성을 줄인다.
- 일관성을 유지시킨다.
- 모듈의 기능은 지나치게 제한적이어서는 안 된다.
- 유지보수가 용이해야 한다.

048 주요 코드

- **순차 코드** : 일정 기준에 따라서 차례로 일련번호를 부여하는 방법
- **표의 숫자 코드** : 코드화 대상 항목의 중량, 면적, 용량 등의 물리적 수치를 적용시키는 방법

049 디자인 패턴(Design Pattern)

- 세부적인 구현 방안을 설계할 때 참조할 수 있는 전형적인 해결 방식 또는 예제를 의미한다.
- **디자인 패턴 유형** : 생성 패턴, 구조 패턴, 행위 패턴

050 생성 패턴(Creational Pattern)

- 추상 팩토리(Abstract Factory) : 서로 연관 · 의존하는 객체들의 그룹으로 생성하여 추상적으로 표현함
- 빌더(Builder) : 작게 분리된 인스턴스를 건축하듯이 조합하여 객체를 생성함
- 팩토리 메소드(Factory Method) : 객체 생성을 서브 클래스에서 처리하도록 분리하여 캡슐화한 패턴으로, 가상 생성자(Virtual Constructor) 패턴이라고도 함
- 프로토타입(Prototype) : 원본 객체를 복제하는 방법으로 객체를 생성함
- 싱글톤(Singleton) : 생성된 객체를 여러 프로세스가 동시에 참조할 수는 없음

051 구조 패턴(Structural Pattern)

- 어댑터(Adapter) : 인터페이스를 다른 클래스가 재사용할 수 있도록 변환함
- 브리지(Bridge) : 서로가 독립적으로 확장할 수 있도록 구성함
- 컴포지트(Composite) : 복합 객체와 단일 객체를 구분 없이 다루고자 할 때 사용함
- 데코레이터(Decorator) : 부가적인 기능을 추가하기 위해 다른 객체들을 덧붙이는 방식으로 구현함
- 퍼싸드(Facade) : 복잡한 서브 클래스들을 피해 더 상위에 인터페이스를 구성함
- 플라이웨이트(Flyweight) : 가능한 한 인스턴스를 공유해서 사용함으로써 메모리를 절약하는 패턴
- 프록시(Proxy) : 접근이 어려운 객체와 여기에 연결하려는 객체 사이에서 인터페이스 역할을 수행하는 패턴

052 행위 패턴 (Behavioral Pattern)

- 책임 연쇄(Chain of Responsibility) : 요청을 한 객체가 처리하지 못하면 다음 객체로 넘어가는 형태
- 커맨드(Command) : 재이용하거나 취소할 수 있도록 요청에 필요한 정보를 저장함
- 인터프리터(Interpreter) : 언어에 문법 표현을 정의함
- 반복자(Iterator) : 접근이 잦은 객체에 대해 동일한 인터페이스를 사용하도록 함

- 중재자(Mediator) : 복잡한 상호 작용을 캡슐화하여 객체로 정의함
- 메멘토(Memento) : 객체를 해당 시점의 상태로 돌릴 수 있는 기능을 제공, Ctrl+Z와 같은 되돌리기 기능을 개발할 때 주로 이용함
- 옵서버(Observer) : 객체에 상속되어 있는 다른 객체들에게 변화된 상태를 전달함
- 상태(State) : 객체의 상태에 따라 동일한 동작을 다르게 처리해야 할 때 사용함
- 전략(Strategy) : 동일한 계열의 알고리즘들을 상호 교환할 수 있게 정의함
- 템플릿 메소드(Template Method) : 하위 클래스에서 세부 처리를 구체화함
- 방문자(Visitor) : 처리 기능을 분리하여 별도의 클래스로 구성함

053 초치기 요구사항 검증 방법

- 동료검토(Peer Review) : 작성자가 명세서 내용을 직접 설명하면서 결함을 발견함
- 워크스루(Walk Through) : 미리 배포한 명세서를 사전 검토한 후 결함을 발견함
- 인스펙션(Inspection) : 작성자를 제외한 다른 검토 전문가들이 결함을 발견함
- 동료검토와 워크스루가 비공식적인 검토 방법인데 반해 인스펙션은 공식적인 검토 방법이다.

054 초치기 미들웨어(Middleware)

- 분산 컴퓨팅 환경에서 서로 다른 기종 간을 연결한다.
- 운영체제와 응용 프로그램 사이에서 다양한 서비스를 제공한다.
- 위치 투명성을 제공한다.
- 사용자가 미들웨어의 내부 동작을 확인하려면 별도의 응용 소프트웨어를 사용해야 한다.

055 초치기 미들웨어의 종류

- DB(DataBase)
- RPC(Remote Procedure Call)
- MOM(Message Oriented Middleware)
- TP(Transaction Processing)-Monitor
- ORB(Object Request Broker)
- WAS(Web Application Server)

2과목 소프트웨어 개발

056 자료 구조의 분류

- 선형 구조 : 배열, 선형 리스트, 스택, 큐, 데크
- 비선형 구조 : 트리, 그래프

057 스택(Stack)

- 리스트의 한쪽 끝으로만 자료의 삽입, 삭제 작업이 이루어지는 자료 구조이다.
- 가장 나중에 삽입된 자료가 가장 먼저 삭제되는 후입선출(LIFO) 방식으로 자료를 처리한다.

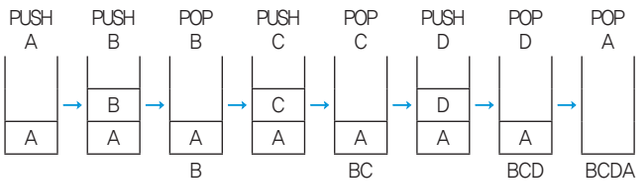
058 스택의 응용 분야

- 인터럽트의 처리
- 수식 계산 및 수식 표기법
- 서브루틴 호출 및 복귀 주소 저장

059 스택의 삽입(Push)과 삭제(Pop)

- PUSH : 스택에 자료를 입력하는 명령
- POP : 스택에서 자료를 출력하는 명령

예제 순서가 A, B, C, D로 정해진 입력 자료를 스택에 입력하였다가 B, C, D, A 순서로 출력하는 과정을 나열하시오.

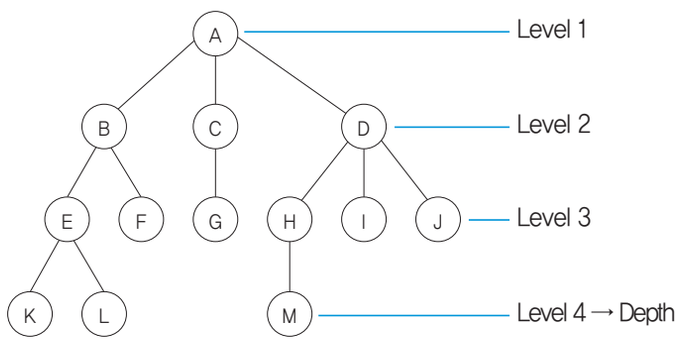


060 방향/무방향 그래프의 최대 간선 수

- 무방향 그래프의 최대 간선 수 : $n(n-1)/2$
- 방향 그래프의 최대 간선 수 : $n(n-1)$

061 트리(Tree)

정점(Node, 노드)과 선분(Branch, 가지)을 이용하여 사이클을 이루지 않도록 구성한 그래프(Graph)의 특수한 형태이다.



- 디그리(Degree, 차수) : 각 노드에서 뻗어 나온 가지의 수

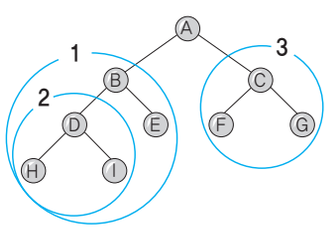
  예 A = 3, B = 2, C = 1, D = 3

- 단말 노드(Terminal Node) = 잎 노드(Leaf Node) : 자식이 하나도 없는 노드, 즉 디그리가 0인 노드

  예 K, L, F, G, M, I, J

062 이진 트리의 운행법

예 다음 트리를 Inorder, Preorder, Postorder 방법으로 운행했을 때 각 노드를 방문한 순서는?



Preorder 운행법의 방문 순서

① Preorder는 Root → Left → Right이므로 A13이 된다.
② 1은 B2E이므로 AB2E3이 된다.
③ 2는 DHI이므로 ABDHIE3이 된다.
④ 3은 CFG이므로 ABDHIECFG가 된다.
∴ 방문 순서 : ABDHIECFG

Inorder 운행법의 방문 순서

① Inorder는 Left → Root → Right이므로 1A3이 된다.
② 1은 2BE이므로 2BEA3이 된다.
③ 2는 HDI이므로 HDIBEA3이 된다.
④ 3은 FCG이므로 HDIBEAFCG가 된다.
∴ 방문 순서 : HDIBEAFCG

Postorder의 방문 순서

① Postorder는 Left → Right → Root이므로 13A가 된다.
② 1은 2EB이므로 2EB3A가 된다.
③ 2는 HID이므로 HIDEB3A가 된다.
④ 3은 FGC이므로 HIDEBFGCA가 된다.
∴ 방문 순서 : HIDEBFGCA

063 초치기 수식의 표기법(Infix → Postfix)

Infix로 표기된 수식에서 연산자를 해당 피연산자 두 개의 뒤(오른쪽)에 오도록 이동하면 Postfix가 된다.

X = A / B * (C + D) + E → X A B / C D + * E + =

① 연산 우선순위에 따라 괄호로 묶는다.
(X = (((A / B) * (C + D)) + E))
② 연산자를 해당 괄호의 뒤로 옮긴다.
X = (((A / B) * (C + D)) + E))
↓
(X (((AB) / (CD) +) * E) +) =
③ 괄호를 제거한다.
X A B / C D + * E + =

064 초치기 수식의 표기법(Infix → Prefix)

Infix로 표기된 수식에서 연산자를 해당 피연산자 두 개의 앞(왼쪽)에 오도록 이동하면 Prefix가 된다.

X = A / B * (C + D) + E → X + * / A B + C D E

① 연산 우선순위에 따라 괄호로 묶는다.
(X = (((A / B) * (C + D)) + E))
② 연산자를 해당 괄호의 앞으로 옮긴다.
(X = (((A / B) * (C + D)) + E))
↓
= (X + (* (/ (A B) + (C D)) E))
③ 괄호를 제거한다.
= X + * / A B + C D E

065 초치기 수식의 표기법(Postfix → Infix)

Postfix는 Infix 표기법에서 연산자를 해당 피연산자 2개의 뒤(오른쪽)로 이동한 것이므로 연산자를 다시 해당 피연산자 2개의 가운데로 옮기면 된다.

A B C − / D E F + * + → A / (B − C) +D * (E + F)

① 인접한 피연산자 2개와 오른쪽의 연산자를 괄호로 묶는다.
((A (B C −) /) (D (E F +) *) +)
② 연산자를 해당 피연산자의 가운데로 이동시킨다.
((A (B C −) /) (D (E F +) *) +)
↓
((A / (B − C)) + (D * (E + F)))
③ 필요 없는 괄호를 제거한다.
A / (B − C) + D * (E + F)

066 초치기 삽입 정렬(Insertion Sort)

예제 8, 5, 6, 2, 4를 삽입 정렬로 정렬하시오.

• **초기 상태 :** 8 5 6 2 4

• **1회전 :** 8 5 6 2 4 → 5 8 6 2 4

두 번째 값을 첫 번째 값과 비교하여 5를 첫 번째 자리에 삽입하고 8을 한 칸 뒤로 이동시킨다.

• **2회전 :** 5 8 6 2 4 → 5 6 8 2 4

세 번째 값을 첫 번째, 두 번째 값과 비교하여 6을 8자리에 삽입하고 8을 한 칸 뒤로 이동시킨다.

5 6 8 2 4 → 2 5 6 8 4

네 번째 값 2를 처음부터 비교하여 맨 처음에 삽입하고 나머지를 한 칸씩 뒤로 이동시킨다.

• **4회전 :** 2 5 6 8 4 → 2 4 5 6 8

다섯 번째 값 4를 처음부터 비교하여 5자리에 삽입하고 나머지를 한 칸씩 뒤로 이동시킨다.

067 초치기 선택 정렬(Selection Sort)

예제 8, 5, 6, 2, 4를 선택 정렬로 정렬하시오.

• **초기 상태 :** 8 5 6 2 4

• **1회전 :** 5 8 6 2 4 → 5 8 6 2 4 → 2 8 6 5 4 → 2 8 6 5 4

• **2회전 :** 2 6 8 5 4 → 2 5 8 6 4 → 2 4 8 6 5

• **3회전 :** 2 4 6 8 5 → 2 4 5 8 6

• **4회전 :** 2 4 5 6 8

068 초치기 버블 정렬(Bubble Sort)

예제 8, 5, 6, 2, 4를 버블 정렬로 정렬하시오.

• **초기 상태 :** 8 5 6 2 4

• **1회전 :** 5 8 6 2 4 → 5 6 8 2 4 → 5 6 2 8 4 → 5 6 2 4 8

• **2회전 :** 5 6 2 4 8 → 5 2 6 4 8 → 5 2 4 6 8

• **3회전 :** 2 5 4 6 8 → 2 4 5 6 8

• **4회전 :** 2 4 5 6 8

069 초치기 이분 검색(이진 검색)

- 검색할 데이터가 정렬되어 있어야 한다.
- 비교 횟수를 거듭할 때마다 검색 대상이 되는 데이터의 수가 절반으로 줄어든다.
- 탐색 효율이 좋다.
- 탐색 시간이 적게 소요된다.
- 중간 레코드 번호(M) : (F+L) / 2 (단, F : 첫 번째 레코드 번호, L : 마지막 레코드 번호)

070 초치기 주요 해싱 함수

- 제산법(Division) : 레코드 키 값(K)을 해시표(Hash Table)의 크기보다 큰 수 중에서 가장 작은 소수(Prime, Q)로 나눈 나머지를 홈 주소로 삼는 방식
- 제곱법(Mid-Square) : 레코드 키 값(K)을 제곱한 후 그 중간 부분의 값을 홈 주소로 삼는 방식
- 폴딩법(Folding) : 레코드 키 값(K)을 여러 부분으로 나눈 후 각 부분의 값을 더하거나 XOR한 값을 홈 주소로 삼는 방식
- 숫자 분석법(Digit Analysis) : 키 값을 이루는 숫자의 분포를 분석하여 비교적 고른 자리를 필요한 만큼 선택해서 홈 주소로 삼는 방식

071 스키마 3계층

- 외부 스키마 : 사용자나 응용 프로그래머가 각 개인의 입장에서 필요로 하는 데이터베이스의 논리적 구조를 정의한 것
- 개념 스키마 : 데이터베이스의 전체적인 논리적 구조로서, 개체 간의 관계와 제약 조건을 나타내고, 데이터베이스의 접근 권한, 보안 및 무결성 규칙에 관한 명세를 정의함
- 내부 스키마 : 물리적 저장장치의 입장에서 본 데이터베이스 구조로서, 실제로 데이터베이스에 저장될 레코드의 형식을 정의하고 저장 데이터 항목의 표현 방법, 내부 레코드의 물리적 순서 등을 나타냄

072 빌드 자동화 도구

- Ant : 아파치 소프트웨어 재단에서 개발한 소프트웨어
- Maven : Ant의 대안으로 개발한 소프트웨어
- Jenkins : JAVA 기반의 오픈 소스 형태의 빌드 자동화 도구
- Gradle : Groovy를 기반으로 한 오픈 소스 형태의 빌드 자동화 도구

073 소프트웨어 패키징

- 모듈별로 생성한 실행 파일들을 묶어 배포용 설치 파일을 만드는 것이다.
- 개발자가 아니라 사용자를 중심으로 진행한다.

074 소프트웨어 패키징 시 고려사항

- 내부 콘텐츠에 대한 암호화 및 보안을 고려한다.
- 다른 여러 콘텐츠 및 단말기 간 DRM(디지털 저작권 관리) 연동을 고려한다.

075 DRM(디지털 저작권 관리)의 구성 요소

- 클리어링 하우스(Clearing House) : 저작권에 대한 사용 권한, 라이선스 발급, 사용량에 따른 결제 관리 등을 수행하는 곳
- 콘텐츠 제공자(Contents Provider) : 콘텐츠를 제공하는 저작권자
- 패키저(Packager) : 콘텐츠를 메타 데이터와 함께 배포 가능한 형태로 묶어 암호화하는 프로그램
- 콘텐츠 분배자(Contents Distributor) : 암호화된 콘텐츠를 유통하는 곳이나 사람
- DRM 컨트롤러(DRM Controller) : 배포된 콘텐츠의 이용 권한을 통제하는 프로그램

076 DRM(디지털 저작권 관리)의 기술 요소

- 콘텐츠 암호화 및 키 관리
- 콘텐츠 식별체계 표현
- 라이선스 발급 및 관리
- 정책 관리 기술
- 크랙 방지 기술

077 소프트웨어 설치 매뉴얼

- 설치 매뉴얼은 사용자를 기준으로 작성한다.
- 기본 사항
  - 소프트웨어 개요
  - 설치 관련 파일
  - 프로그램 삭제 등

078 초치기 형상 관리(SCM)

- 소프트웨어의 개발 과정에서 소프트웨어의 변경 사항을 관리하기 위해 개발된 일련의 활동이다.
- 목적 : 개발 비용 감소, 방해 요인 최소화
- 관리 항목 : 소스 코드, 프로젝트 분석서, 운영 및 설치 지침서 등
- 형상 관리 도구 : Git, CVS, Subversion 등

079 초치기 소프트웨어의 버전 등록 관련 주요 기능

- 체크아웃(Check-Out) : 프로그램을 수정하기 위해 저장소에서 파일을 받아옴
- 체크인(Check-In) : 체크아웃 한 파일의 수정을 완료한 후 저장소의 파일을 새로운 버전으로 갱신함
- 커밋(Commit) : 체크인을 수행할 때 이전에 갱신된 내용이 있는 경우에는 충돌(Conflict)을 알리고 diff 도구를 이용해 수정한 후 갱신을 완료함

080 초치기 파레토 법칙(Pareto Principle)

소프트웨어 테스트에서 오류의 80%는 전체 모듈의 20% 내에서 발견된다는 법칙이다.

081 초치기 화이트박스 테스트 (White Box Test)

- 모듈의 원시 코드를 오픈시킨 상태에서 원시 코드의 논리적인 모든 경로를 테스트하여 테스트 케이스를 설계하는 방법이다.
- 프로그램의 제어 구조에 따라 선택, 반복 등의 분기점 부분들을 수행함으로써 논리적 경로를 제어한다.

082 초치기 화이트박스 테스트의 종류

- 기초 경로 검사
- 제어 구조 검사
  - 조건 검사
  - 루프 검사
  - 데이터 흐름 검사

※ 기초 경로(Base Path = Basis Path) : 수행 가능한 모든 경로를 의미함

083 초치기 블랙박스 테스트 종류

- 동치 분할 검사
- 경계값 분석
- 원인-효과 그래프 검사
- 오류 예측 검사
- 비교 검사

084 초치기 단위 테스트(Unit Test)

코딩 직후 소프트웨어 설계의 최소 단위인 모듈이나 컴포넌트에 초점을 맞춰 테스트하는 것이다.

085 초치기 단위 테스트로 발견 가능한 오류

- 알고리즘 오류에 따른 원치 않는 결과
- 탈출구가 없는 반복문의 사용
- 틀린 계산 수식에 의한 잘못된 결과

086 초치기 인수 테스트의 종류

- 알파 테스트 : 개발자의 장소에서 사용자가 개발자 앞에서 행하는 테스트 기법
- 베타 테스트 : 선정된 최종 사용자가 여러 명의 사용자 앞에서 행하는 테스트 기법으로, 필드 테스팅(Field Testing)이라고도 불림

087 초치기 통합 테스트 (Integration Test)

- 하향식 통합 테스트 : 프로그램의 상위 모듈에서 하위 모듈 방향으로 통합하면서 테스트하는 기법으로, 깊이 우선 통합법이나 넓이 우선 통합법을 사용함
- 상향식 통합 테스트 : 프로그램의 하위 모듈에서 상위 모듈 방향으로 통합하면서 테스트하는 기법

088 초치기 테스트 드라이버(Test Driver)

- 테스트 대상의 하위 모듈을 호출하는 도구이다.
- 매개 변수(Parameter)를 전달하고, 모듈 테스트 수행 후의 결과를 도출한다.
- 상향식 통합 테스트에 사용된다.

089 초치기 테스트 스텁(Test Stub)

- 일시적으로 필요한 조건만을 가지고 있는 시험용 모듈이다.
- 하향식 통합 테스트에 사용된다.

090 초치기 테스트 오라클(Test Oracle)

- 테스트 결과가 올바른지 판단하기 위해 사전에 정의된 참 값을 대입하여 비교하는 기법 및 활동이다.
- 종류 : 참 오라클, 샘플링 오라클, 추정 오라클, 일관성 검사 오라클

091 초치기 주요 최악의 시간 복잡도

O(1)	입력값(n)에 관계 없이 일정하게 문제 해결에 하나의 단계만을 거침 예 스택의 삽입(Push), 삭제(Pop)
$O(n\log_2 n)$	문제 해결에 필요한 단계가 $n(\log_2 n)$번만큼 수행됨 예 힙 정렬(Heap Sort), 2-Way 합병 정렬(Merge Sort)

092 초치기 클린 코드 작성 원칙

- 가독성 : 누구든지 코드를 쉽게 읽을 수 있도록 작성함
- 단순성 : 코드를 간단하게 작성함
- 의존성 배제 : 코드가 다른 모듈에 미치는 영향을 최소화함
- 중복성 최소화 : 코드의 중복을 최소화함
- 추상화 : 상위 클래스/메소드/함수에서는 간략하게 애플리케이션의 특성을 나타내고, 상세 내용은 하위 클래스/메소드/함수에서 구현함

093 초치기 외계인 코드(Alien Code)

아주 오래되거나 참고문서 또는 개발자가 없어 유지 보수 작업이 어려운 코드를 의미한다.

094 소스 코드 품질 분석 도구 - 정적 분석 도구

- 하드웨어 또는 소프트웨어적인 방법으로 코드 분석이 가능하다.
- **종류** : pmd, checkstyle, cppcheck 등

095 EAI의 구축 유형

- Point-to-Point : 애플리케이션을 1:1로 연결함
- Hub & Spoke : 단일 접점인 허브 시스템을 통해 데이터를 전송하는 중앙 집중형 방식
- Message Bus(ESB 방식) : 애플리케이션 사이에 미들웨어를 두어 처리하는 방식
- Hybrid : Hub & Spoke와 Message Bus의 혼합 방식

096 JSON (JavaScript Object Notation)

속성-값 쌍(Attribute-Value Pairs)으로 이루어진 데이터 객체를 전달하기 위해 사람이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷이다.

097 AJAX(Asynchronous JavaScript and XML)

- 자바 스크립트(JavaScript) 등을 이용한 비동기 통신 기술이다.
- 클라이언트와 서버 간에 XML 데이터를 교환 및 제어한다.

098 인터페이스 보안 기능 적용 - 네트워크 영역

- 네트워크 트래픽에 대한 암호화를 설정한다.
- 암호화는 IPSec, SSL, S-HTTP 등의 다양한 방식으로 적용한다.

099 tripwire

크래커가 침입하여 백도어를 만들어 놓거나, 설정 파일을 변경했을 때 분석하는 데이터 무결성 검사 도구 중 하나이다.

100 인터페이스 구현 검증 도구

- xUnit : JUnit, CppUnit, NUnit, HttpUnit 등 다양한 언어에 적용되는 단위 테스트 프레임워크
- STAF : 서비스 호출 및 컴포넌트 재사용 등 다양한 환경을 지원하는 테스트 프레임워크
- FitNesse : 웹 기반 테스트 케이스 설계, 실행, 결과 확인 등을 지원하는 테스트 프레임워크
- NTAF : FitNesse와 STAF의 장점을 통합한 NHN(Naver)의 테스트 자동화 프레임워크
- watir : Ruby를 사용하는 애플리케이션 테스트 프레임워크

3과목 데이터베이스 구축

101 개념적 설계 (정보 모델링, 개념화)

- 정보의 구조를 얻기 위하여 현실 세계에 대한 인식을 추상적 개념으로 표현하는 과정이다.
- 개념 스키마 모델링과 트랜잭션 모델링을 병행 수행한다.

102 논리적 설계(데이터 모델링)

- 자료를 특정 DBMS가 지원하는 논리적 자료 구조로 변환(mapping)시키는 과정이다.
- 트랜잭션의 인터페이스를 설계한다.
- 개념 스키마를 평가 및 정제한다.

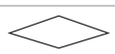
103 물리적 설계

- 논리적 구조로 표현된 데이터를 물리적 구조의 데이터로 변환하는 과정이다.
- 데이터베이스 파일의 저장 구조 및 액세스 경로를 결정한다.
- 저장 레코드의 형식, 순서, 접근 경로, 조회가 집중되는 레코드와 같은 정보를 사용한다.

104 데이터 모델에 표시할 요소

- 구조(Structure) : 논리적으로 표현된 개체 타입들 간의 관계로서 데이터 구조 및 정적 성질을 표현함
- 연산(Operation) : 데이터베이스에 저장된 실제 데이터를 처리하는 작업에 대한 명세로서 데이터베이스를 조작하는 기본 도구
- 제약 조건(Constraint) : 데이터베이스에 저장될 수 있는 실제 데이터의 논리적인 제약 조건

105 E-R 다이어그램

기호	기호 이름	의미
	사각형	개체(Entity) 타입
	마름모	관계(Relationship) 타입
	타원	속성(Attribute)
	이중 타원	다중값 속성(복합 속성)
	선, 링크	개체 타입과 속성을 연결

106 튜플(Tuple)

- 릴레이션을 구성하는 각각의 행을 말한다.
- 튜플의 수 = 카디널리티(Cardinality)

107 속성(Attribute)

- 데이터베이스를 구성하는 가장 작은 논리적 단위이다.
- 속성의 수 = 디그리(Degree) = 차수

108 도메인(Domain)

하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자(Atomic)값들의 집합이다.

109 릴레이션의 특징

- 한 릴레이션에는 똑같은 튜플이 포함될 수 없으므로 릴레이션에 포함된 튜플들은 모두 상이하다.
- 한 릴레이션에 포함된 튜플 사이에는 순서가 없다.
- 속성의 유일한 식별을 위해 속성의 명칭은 유일해야 한다.
- 속성의 값은 논리적으로 더 이상 쪼갤 수 없는 원자값만을 저장한다.

110 후보키(Candidate Key)

- 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용하는 속성들의 부분집합, 즉 기본키로 사용할 수 있는 속성들을 말한다.
- 릴레이션에 있는 모든 튜플에 대해서 유일성과 최소성을 만족시켜야 한다.

111 기본키(Primary Key)

- 후보키 중에서 특별히 선정된 주키(Main Key)로 중복된 값을 가질 수 없다.
- NULL 값을 가질 수 없다.

112 대체키(Alternate Key)

- 후보키가 둘 이상일 때 기본키를 제외한 나머지 후보키를 의미한다.
- 보조키라고도 한다.

113 슈퍼키(Super Key)

- 한 릴레이션 내에 있는 속성들의 집합으로 구성된 키이다.
- 릴레이션을 구성하는 모든 튜플에 대해 유일성은 만족시키지만, 최소성은 만족시키지 못한다.

114 외래키(Foreign Key)

- 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미한다.
- 한 릴레이션에 속한 속성 A와 참조 릴레이션의 기본키인 B가 동일한 도메인 상에서 정의되었을 때의 속성 A를 외래키라고 한다.

115 무결성

- 개체 무결성 : 기본 테이블의 기본키를 구성하는 어떤 속성도 Null 값이나 중복값을 가질 수 없다는 규정
- 참조 무결성 : 외래키 값은 Null이거나 참조 릴레이션의 기본키 값과 동일해야 함. 즉 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다는 규정

116 관계대수

- 관계형 데이터베이스에서 원하는 정보와 그 정보를 검색하기 위해서 어떻게 유도하는가를 기술하는 절차적인 언어이다.
- 질의에 대한 해를 구하기 위해 수행해야 할 연산의 순서를 명시한다.

117 순수 관계 연산자 - Select

- 릴레이션에 존재하는 튜플 중에서 선택 조건을 만족하는 튜플의 부분집합을 구하여 새로운 릴레이션을 만드는 연산이다.
- 기호 : 시그마(σ)

118 순수 관계 연산자 - Project

- 주어진 릴레이션에서 속성 리스트에 제시된 속성 값만을 추출하여 새로운 릴레이션을 만드는 연산이다.
- 기호 : 파이(π)

119 순수 관계 연산자 - Join

- 공통 속성을 중심으로 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 연산이다.
- 기호 : $\bowtie$

120 순수 관계 연산자 - Division

- X⊃Y인 두 개의 릴레이션 R(X)와 S(Y)가 있을 때, R의 속성이 S의 속성값을 모두 가진 튜플에서 S가 가진 속성을 제외한 속성만을 구하는 연산이다.
- 기호 : ÷

121 일반 집합 연산자 - 교차곱 (CARTESIAN PRODUCT)

- 두 릴레이션에 있는 튜플들의 순서쌍을 구하는 연산이다.
- 교차곱의 디그리는 두 릴레이션의 디그리를 더한 것과 같다.
- 교차곱의 카디널리티는 두 릴레이션의 카디널리티를 곱한 것과 같다.

122 관계해석

- 관계 데이터 모델의 제안자인 코드(Codd)가 수학의 Predicate Calculus(술어 해석)에 기반을 두고 관계 데이터베이스를 위해 제안했다.
- 주요 기호

기호	구성 요소	설명
∀	전칭 정량자	가능한 모든 튜플에 대하여
∃	존재 전량자	하나라도 일치하는 튜플이 있음(There Exists)

123 정규화(Normalization)

- 함수적 종속성 등의 종속성 이론을 이용하여 잘못 설계된 관계형 스키마를 더 작은 속성의 세트로 쪼개어 바람직한 스키마로 만들어 가는 과정이다.
- 논리적 설계 단계에서 수행한다.
- 데이터 중복을 배제하여 이상(Anomaly)의 발생 방지한다.
- 자료 저장 공간의 최소화가 가능하다.

124 이상(Anomaly)

- 정규화를 거치지 않으면 데이터베이스 내에 데이터들이 불필요하게 중복되어 릴레이션 조작 시 예기치 못한 곤란한 현상이 발생하는 것을 의미한다.
- 종류 : 삽입 이상, 삭제 이상, 갱신 이상

125 정규화 과정

비정규 릴레이션
↓ 도메인이 원자값
1NF
↓ 부분적 함수 종속 제거
2NF
↓ 이행적 함수 종속 제거
3NF
↓ 결정자이면서 후보키가 아닌 것 제거
BCNF
↓ 다치 종속 제거
4NF
↓ 조인 종속성 이용
5NF

126 함수적 종속 (Functional Dependency)

- 데이터들이 어떤 기준값에 의해 종속되는 것을 의미한다.
- '학번'에 따라 '이름'이 결정될 때 '이름'을 '학번'에 함수종속적이라고 하며 '학번 → 이름'과 같이 쓴다.

127 이행적 종속 관계

A → B이고 B → C일 때 A → C를 만족하는 관계를 의미한다.

128 반정규화(Denormalization)

- 시스템의 성능 향상, 개발 및 운영의 편의성 등을 위해 정규화된 데이터 모델을 통합, 중복, 분리하는 과정으로, 의도적으로 정규화 원칙을 위배하는 행위이다.
- 방법 : 테이블 통합, 테이블 분할, 중복 테이블 추가, 중복 속성 추가 등

129 시스템 카탈로그 (System Catalog)

- 시스템 그 자체에 관련이 있는 다양한 객체에 관한 정보를 포함하는 시스템 데이터베이스이다.
- 사용자가 시스템 카탈로그 내용을 검색할 수는 있지만 갱신할 수는 없다.

130 트랜잭션(Transaction) 정의

- 데이터베이스의 상태를 변환시키는 하나의 논리적 기능을 수행하기 위한 작업의 단위
- 한꺼번에 모두 수행되어야 할 일련의 연산

131 트랜잭션의 상태

- 활동(Active) : 트랜잭션이 실행 중인 상태
- 실패(Failed) : 트랜잭션 실행 중 오류가 발생하여 중단된 상태
- 철회(Aborted) : 트랜잭션이 비정상적으로 종료되어 Rollback 연산을 수행한 상태
- 부분 완료(Partially Committed) : 트랜잭션의 마지막 연산까지 완료했지만, Commit 연산이 실행되기 직전의 상태
- 완료(Committed) : 트랜잭션이 성공적으로 종료되어 Commit 연산까지 수행한 상태

132 트랜잭션의 특성

- Atomicity(원자성) : 트랜잭션의 연산은 데이터베이스에 모두 반영되도록 완료(Commit)되든지 아니면 전혀 반영되지 않도록 복구(Rollback)되어야 함
- Consistency(일관성) : 트랜잭션이 그 실행을 성공적으로 완료하면 언제나 일관성 있는 데이터베이스 상태로 변환함
- Isolation(독립성) : 둘 이상의 트랜잭션이 동시에 병행 실행되는 경우 어느 하나의 트랜잭션 실행 중에 다른 트랜잭션의 연산이 끼어들 수 없음
- Durability(영속성) : 성공적으로 완료된 트랜잭션의 결과는 시스템이 고장나더라도 영구적으로 반영되어야 함

133 인덱스(Index)

- 데이터 레코드를 빠르게 접근하기 위해 〈키 값, 포인터〉 쌍으로 구성되는 데이터 구조이다.
- 데이터 정의어(DDL)를 이용하여 사용자가 생성, 변경, 제거할 수 있다.

134 뷰(View)

- 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블이다.
- 뷰는 가상 테이블이기 때문에 물리적으로 구현되어 있지 않다.
- 뷰로 구성된 내용에 대한 삽입, 삭제, 갱신 연산에 제약이 따른다.
- 뷰를 정의할 때는 CREATE문, 제거할 때는 DROP문을 사용한다.
- 독립적인 인덱스를 가질 수 없다.

135 초치기 파티션의 종류

- 범위 분할(Range Partitioning) : 지정한 열의 값을 기준으로 범위를 지정하여 분할함

  예 일별, 월별, 분기별 등

- 해시 분할(Hash Partitioning) : 해시 함수를 적용한 결과값에 따라 데이터를 분할함
- 조합 분할(Composite Partitioning) : 범위 분할로 분할한 다음 해시 함수를 적용하여 다시 분할하는 방식
- 목록 분할(List Partitioning) : 지정한 열 값에 대한 목록을 만들어 이를 기준으로 분할함
- 라운드 로빈 분할(Round Robin Partitioning) : 레코드를 균일하게 분배하는 방식

136 초치기 분산 데이터베이스

- 논리적으로는 하나의 시스템에 속하지만 물리적으로는 네트워크를 통해 연결된 여러 개의 컴퓨터 사이트(Site)에 분산되어 있는 데이터베이스를 말한다.
- 데이터베이스 설계 및 소프트웨어 개발이 어렵다.
- 분산 데이터베이스의 구성 요소 : 분산 처리기, 분산 데이터베이스, 통신 네트워크

137 초치기 분산 데이터베이스의 목표

- 위치 투명성(Location Transparency) : 액세스하려는 데이터베이스의 실제 위치를 알 필요 없이 단지 데이터베이스의 논리적인 명칭만으로 액세스할 수 있음
- 중복 투명성(Replication Transparency) : 동일 데이터가 여러 곳에 중복되어 있더라도 사용자는 마치 하나의 데이터만 존재하는 것처럼 사용하고, 시스템은 자동으로 여러 자료에 대한 작업을 수행함
- 병행 투명성(Concurrency Transparency) : 분산 데이터베이스와 관련된 다수의 트랜잭션들이 동시에 실현되더라도 그 트랜잭션의 결과는 영향을 받지 않음
- 장애 투명성(Failure Transparency) : 트랜잭션, DBMS, 네트워크, 컴퓨터 장애에도 불구하고 트랜잭션을 정확하게 처리함

138 초치기 암호화·복호화 과정

- 암호화(Encryption) 과정 : 암호화되지 않은 평문을 정보보호를 위해 암호문으로 바꾸는 과정
- 복호화(Decryption) 과정 : 암호문을 원래의 평문으로 바꾸는 과정

139 초치기 접근통제 기술

- 임의 접근통제(DAC; Discretionary Access Control) : 데이터에 접근하는 사용자의 신원에 따라 접근 권한을 부여하는 방식
- 강제 접근통제(MAC; Mandatory Access Control) : 주체와 객체의 등급을 비교하여 접근 권한을 부여하는 방식
- 역할기반 접근통제(RBAC; Role Based Access Control) : 사용자의 역할에 따라 접근 권한을 부여하는 방식

140 초치기 벨 라파듈라 모델 (Bell-LaPadula Model)

- 군대의 보안 레벨처럼 정보의 기밀성에 따라 상하 관계가 구분된 정보를 보호하기 위해 사용하는 접근제어 모델
- 보안 취급자의 등급을 기준으로 읽기 권한과 쓰기 권한이 제한된다.

141 초치기 DAS (Direct Attached Storage)

- 서버와 저장장치를 전용 케이블로 직접 연결하는 방식이다.
- 일반 가정에서 컴퓨터에 외장하드를 연결하는 것이 여기에 해당된다.

142 초치기 SAN(Storage Area Network)

- DAS의 빠른 처리와 NAS의 파일 공유 장점을 혼합한 방식이다.
- 서버와 저장장치를 연결하는 전용 네트워크를 별도로 구성한다.

143 초치기 DDL(데이터 정의어)

- 스키마, 도메인, 테이블, 뷰, 인덱스를 정의하거나 변경 또는 삭제할 때 사용하는 언어이다.
- CREATE : 스키마, 도메인, 테이블, 뷰, 인덱스를 정의함
- ALTER : TABLE에 대한 정의를 변경하는 데 사용함
- DROP : 스키마, 도메인, 테이블, 뷰, 인덱스를 삭제함

144 초치기 DML(데이터 조작어)

- 데이터베이스 사용자가 응용 프로그램이나 질의어를 통하여 저장된 데이터를 실질적으로 처리하는 데 사용되는 언어이다.
- SELECT : 테이블에서 조건에 맞는 튜플을 검색함
- INSERT : 테이블에 새로운 튜플을 삽입함
- DELETE : 테이블에서 조건에 맞는 튜플을 삭제함
- UPDATE : 테이블에서 조건에 맞는 튜플의 내용을 변경함

145 초치기 DCL(데이터 제어어)

- 데이터의 보안, 무결성, 회복, 병행 수행 제어 등을 정의하는 데 사용되는 언어이다.
- COMMIT : 명령에 의해 수행된 결과를 실제 물리적 디스크로 저장하고, 데이터베이스 조작 작업이 정상적으로 완료되었음을 관리자에게 알려줌
- ROLLBACK : 데이터베이스 조작 작업이 비정상적으로 종료되었을 때 원래의 상태로 복구함
- GRANT : 데이터베이스 사용자에게 사용 권한을 부여함
- REVOKE : 데이터베이스 사용자의 사용 권한을 취소함

146 초치기 CREATE TABLE

- 테이블을 정의하는 명령문이다.
- 표기 형식

```
CREATE TABLE 테이블명
        (속성명 데이터_타입 [DEFAULT 기본값] [NOT NULL], …
        [, PRIMARY KEY(기본키_속성명, …)]
        [, UNIQUE(대체키_속성명, …)]
        [, FOREIGN KEY(외래키_속성명, …)]
              [REFERENCES 참조테이블(기본키_속성명, …)]
              [ON DELETE 옵션]
              [ON UPDATE 옵션]
        [, CONSTRAINT 제약조건명] [CHECK (조건식)]);
```

147 초치기 ALTER TABLE

- 테이블에 대한 정의를 변경하는 명령문이다.
- 표기 형식

```
ALTER TABLE 테이블명 ADD 속성명 데이터_타입 [DEFAULT '기본값'];
ALTER TABLE 테이블명 ALTER 속성명 [SET DEFAULT '기본값'];
ALTER TABLE 테이블명 DROP COLUMN 속성명 [CASCADE];
```

148 초치기 DROP TABLE

- 기본 테이블을 제거하는 명령문이다.
- 표기 형식

```
DROP TABLE 테이블명 [CASCADE | RESTRICT];
```

- CASCADE : 제거할 요소를 참조하는 다른 모든 개체를 함께 제거함
- RESTRICT : 다른 개체가 제거할 요소를 참조중일 때는 제거를 취소함

149 초치기 삽입문(INSERT INTO~)

- 기본 테이블에 새로운 튜플을 삽입할 때 사용한다.
- 표기 형식

```
INSERT INTO 테이블명([속성명1, 속성명2,…])
VALUES (데이터1, 데이터2,… );
```

150 초치기 삭제문(DELETE FROM~)

- 기본 테이블에 있는 튜플들 중에서 특정 튜플(행)을 삭제할 때 사용한다.
- 표기 형식

```
DELETE
FROM 테이블명
[WHERE 조건];
```

151 초치기 갱신문(UPDATE~ SET~)

- 기본 테이블에 있는 튜플들 중에서 특정 튜플의 내용을 변경할 때 사용한다.
- 표기 형식

```
UPDATE 테이블명
SET 속성명 = 데이터[, 속성명=데이터, …]
[WHERE 조건];
```

152 초치기 데이터 조작문의 네 가지 유형

- SELECT(검색) : SELECT~ FROM~ WHERE~
- INSERT(삽입) : INSERT INTO~ VALUES~
- DELETE(삭제) : DELETE~ FROM~ WHERE~
- UPDATE(변경) : UPDATE~ SET~ WHERE~

153 초치기 Select문

```
SELECT [PREDICATE] [테이블명.]속성명1, [테이블명.]속성명2,…
FROM 테이블명1, 테이블명2,…
[WHERE 조건]
[GROUP BY 속성명1, 속성명2,…]
[HAVING 조건]
[ORDER BY 속성명 [ASC | DESC]];
```

- SELECT절
  - Predicate : 불러올 튜플 수를 제한할 명령어
    - ▶ DISTINCT : 중복된 튜플이 있으면 그 중 첫 번째 한 개만 검색
  - 속성명 : 검색하여 불러올 속성(열) 및 수식들
- FROM절 : 질의에 의해 검색될 데이터들을 포함하는 테이블명
- WHERE절 : 검색할 조건
- GROUP BY절 : 특정 속성을 기준으로 그룹화하여 검색할 때 그룹화 할 속성
- HAVING절 : 그룹에 대한 조건
- ORDER BY절
  - 속성명 : 정렬의 기준이 되는 속성명
  - [ASC | DESC] : 정렬 방식(ASC는 오름차순, DESC 또는 생략하면 내림차순)

154 초치기 조건 연산자 - LIKE

- 대표 문자를 이용해 지정된 속성의 값이 문자 패턴과 일치하는 튜플을 검색하기 위해 사용된다.
- 대표 문자
  - % : 모든 문자를 대표함
  - _ : 문자 하나를 대표함
  - # : 숫자 하나를 대표함

155 조건 연산자 - BETWEEN

지정된 속성이 두 숫자 사이의 값을 가지는 튜플을 검색하기 위해 사용된다.

예 생일이 '01/09/69'에서 '10/22/73' 사이인 자료만 검색

→ WHERE 생일 BETWEEN #01/09/69# AND #10/22/73#

156 그룹 함수

- GROUP BY절에 지정된 그룹별로 속성의 값을 집계할 때 사용된다.
- COUNT/SUM/AVG/MAX/MIN(속성명) : 그룹별 튜플 수/합계/평균/최대값/최소값을 구하는 함수

157 집합 연산자의 종류

- UNION : 두 조회 결과를 통합하여 모두 출력하되, 중복된 행은 한 번만 출력함
- UNION ALL : 두 조회 결과를 통합하여 모두 출력하되, 중복된 행도 그대로 출력함
- INTERSECT : 두 조회 결과 중 공통된 행만 출력함
- EXCEPT : 첫 번째 조회 결과에서 두 번째 조회 결과를 제외한 행을 출력함

158 트리거(Trigger)

데이터의 삽입(Insert), 갱신(Update), 삭제(Delete) 등의 이벤트(Event)가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL이다.

4과목 프로그래밍 언어 활용

159 C/JAVA의 자료형

종류	C	JAVA
문자	char(1Byte)	char(1Byte)
정수	int(4Byte)	int(4Byte)
	long(8Byte)	long(8Byte)
논리		boolean(1Byte)

160 C언어의 구조체

- 자료의 종류가 다른 변수의 모임이다.
- 예약어 struct를 이용해 정의한다.

161 Python의 시퀀스 자료형

- 리스트(List) : 필요에 따라 개수를 늘리거나 줄일 수 있음
- 튜플(Tuple) : 요소의 추가, 삭제, 변경은 불가능함
- range : 연속된 숫자를 생성함

162 변수명 작성 규칙

- 영문자, 숫자, _(under bar)를 사용할 수 있다.
- 첫 글자는 숫자는 올 수 없다.
- 공백이나 *, +, -, / 등의 특수문자를 사용할 수 없다.
- 대 · 소문자를 구분한다.
- 예약어를 변수명으로 사용할 수 없다.

163 가비지 콜렉터 (Garbage Collector)

선언만 하고 사용하지 않는 변수들이 점유한 메모리 공간을 강제로 해제하여 다른 프로그램들이 사용할 수 있도록 하는 것이다.

164 산술 연산자

연산자	의미	비고
%	나머지	정수만 연산할 수 있으며, 실수를 사용하면 오류가 발생함
++	증가	• **전치** : 변수 앞에 증감 연산자가 오는 형태로 먼저 변수의 값을 증감시킨 후 변수를 연산에 사용함(++a, --a)
--	감소	• **후치** : 변수 뒤에 증감 연산자가 오는 형태로 먼저 변수를 연산에 사용한 후 변수의 값을 증감시킴(a++, a--)

165 비트 연산자

- & (and) : 모든 비트가 1일 때만 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

166 논리 연산자

- ! (not) : 부정
- && (and) : 모두 참이면 참
- || (or) : 하나라도 참이면 참

167 조건 연산자

조건에 따라 서로 다른 수식을 수행한다.

예 mx = a 〈 b ? b : a;

a가 b보다 작으면 mx에 b를 저장하고 그렇지 않으면 mx에 a를 저장한다.

168 연산자 우선순위

대분류	중분류	연산자	결합 규칙	우선 순위
단항 연산자	단항 연산자	! ~ ++ -- sizeof	←	높음 ↑
이항 연산자	산술 연산자	* / %	→	
		+ -		
	시프트 연산자	〈〈 〉〉		
	관계 연산자	〈 〈= 〉= 〉		
		== !=		
	비트 연산자	& ^ |		
	논리 연산자	&& ||		
삼항 연산자	조건 연산자	? :	→	
대입 연산자	대입 연산자	= += -= *= /= %= 〈〈= 〉〉= 등	←	↓
순서 연산자	순서 연산자	,	→	낮음

169 주요 서식 문자열

- %d : 정수형 10진수를 입 · 출력하기 위해 지정함
- %c : 문자를 입 · 출력하기 위해 지정함
- %s : 문자열을 입 · 출력하기 위해 지정함

170 printf( ) 함수

인수로 주어진 값을 화면에 출력하는 함수이다.

예 printf("%d, %c", a, b);

a의 값을 정수로 출력하고 쉼표(,)와 공백 한 칸을 띄운 후, b의 값을 문자로 출력한다.

171 JAVA의 출력 함수

- printf( )

  예 System.out.printf("%d", r);
  r의 값을 10진수 정수로 출력한다.

- print( )

  예 System.out.print(r + s);
  r과 s를 더한 값을 출력한다.

- println( )

  예 System.out.println(r + "은(는) 소수");
  r의 값과 **은(는) 소수**를 출력한 후, 커서를 다음 줄의 처음으로 옮긴다.

172 단순 if문

- 조건이 한 개일 때 사용하는 제어문이다.
- 조건이 참일 때만 실행하는 경우

예

```
if (a > b)
    printf("Gilbut");
```

a가 b보다 크면 Gilbut을 출력하고, 아니면 if문을 벗어난다.

- 조건이 참일 때와 거짓일 때 실행할 문장이 다른 경우

예

```
if (a > b)
    printf("참");
else
    printf("거짓");
```

a가 b보다 크면 **참**을 출력하고, 아니면 **거짓**을 출력한다.

173 switch문

- 조건에 따라 분기할 곳이 여러 곳인 경우 간단하게 처리할 수 있는 제어문이다.
- break문이 생략되면 수식과 레이블이 일치할 때 실행할 문장부터 break문 또는 switch문이 종료될 때까지 모든 문장이 실행된다.

예

```
switch(a) {
case 1:
    printf("바나나");
    break;
case 2:
    printf("딸기");
    break;
default:
    printf("없음");
}
```

- a가 1이면 바나나를 출력하고 switch문을 탈출한다.
- a가 2면 딸기를 출력하고 switch문을 탈출한다.
- a가 1이나 2가 아니면 없음을 출력하고 switch문을 탈출한다.

174 for문

초기값, 최종값, 증가값을 지정하는 수식을 이용해 정해진 횟수를 반복하는 제어문이다.

예

```
for (i = 1; i <= 10 ; i++)
    sum = sum + i;
```

반복 변수 i가 1부터 1씩 증가하면서 10보다 작거나 같은 동안 sum에 i의 값을 누적시킨다.

175 초치기 while문

조건이 참인 동안 실행할 문장을 반복 수행하는 제어문이다.

예

```
while (i <= 10)
    i = i + 1;
```

i가 10보다 작거나 같은 동안 i의 값을 1씩 누적시킨다.

176 초치기 do~while문

- 조건이 참인 동안 정해진 문장을 반복 수행하다가 조건이 거짓이면 반복문을 벗어난다.
- 실행할 문장을 무조건 한 번 실행한 다음 조건을 판단하여 탈출 여부를 결정한다.

예

```
do
    i = i + 1;
while (i <= 10);
```

i가 10보다 작거나 같은 동안 i의 값을 1씩 누적시킨다.

177 초치기 1차원 배열

변수들을 일직선상의 개념으로 조합한 배열이다.

예 char a[3] = {'A', 'B', 'C'};

3개의 요소를 갖는 문자형 배열 a를 선언한다.

	a[0]	a[1]	a[2]
배열 a	'A'	'B'	'C'

178 초치기 2차원 배열

변수들을 평면, 즉 행과 열로 조합한 배열이다.

예 int b[2][3] = {{11, 22, 33}, {44, 55, 66}};

2개의 행과 3개의 열을 갖는 정수형 배열 b를 선언한다.

	a[0][0]	a[0][1]	a[0][2]
배열 b	11	22	33
	44	55	66
	a[1][0]	a[1][1]	a[1][2]

179 초치기 배열 형태의 문자열 변수

- C언어에서는 큰따옴표(" ")로 묶인 글자는 글자 수에 관계없이 문자열로 처리된다.
- 배열에 문자열을 저장하면 문자열의 끝을 알리기 위한 널 문자('\0')가 문자열 끝에 자동으로 삽입된다.

예 char a[5] = "love";

5개의 요소를 갖는 문자형 배열 a를 선언하고, "love"로 초기화한다.

배열 a	'l'	'o'	'v'	'e'	'\0'
	a[0]	a[1]	a[2]	a[3]	a[4]

180 초치기 포인터와 포인터 변수

- 포인터 변수를 선언할 때는 자료의 형을 먼저 쓰고 변수명 앞에 간접 연산자 *를 붙인다(예 int *a;).
- 포인터 변수에 주소를 저장하기 위해 변수의 주소를 알아낼 때는 변수 앞에 번지 연산자 &를 붙인다(예 a = &b;).
- 실행문에서 포인터 변수에 간접 연산자 *를 붙이면 해당 포인터 변수가 가리키는 곳의 값을 말한다(예 c = *a;).

예제 다음 C언어로 구현된 프로그램의 출력 결과를 확인하시오.

```
main( )
{
 int a = 50; ❶
 int *b; ❷
 b = &a; ❸
 *b = *b+20; ❹
 printf("%d, %d", a, *b); ❺
}
```

❶ 정수형 변수 a를 선언하고 50으로 초기화한다.
❷ 정수형 변수가 저장된 곳의 주소를 기억할 포인터 변수 b를 선언한다.
❸ 정수형 변수 a의 주소를 포인터 변수 b에 기억시킨다. b에는 a의 주소가 저장된다.
❹ b가 가리키는 곳의 값에 20을 더한다. b가 가리키는 곳이 a이므로 결국 a의 값도 바뀌는 것이다.
❺ 결과 **70, 70**

181 초치기 포인터와 배열

- 배열을 포인터 변수에 저장한 후 포인터를 이용해 배열의 요소에 접근할 수 있다.
- 배열 위치를 나타내는 첨자를 생략하고 배열의 대표명만 지정하면 배열의 첫 번째 요소의 주소를 지정하는 것과 같다.

예

```
❶ int a[5], *b
❷ b = a;
❸ b = &a[0];
```

❶ 5개의 요소를 갖는 정수형 배열 a와 정수형 포인터 변수 b를 선언한다.
❷ 배열의 대표명을 적었으므로 a 배열의 시작 주소인 a[0]의 주소를 b에 저장한다.
❸ a 배열의 첫 번째 요소인 a[0]의 주소(&)를 b에 저장한다.

	a[0]	a[1]	a[2]	a[3]	a[4]	← 배열 표기 방법
배열 a	첫 번째	두 번째	세 번째	네 번째	다섯 번째	
	*(a+0)	*(a+1)	*(a+2)	*(a+3)	*(a+4)	← 포인터 표기 방법

182 초치기 Python의 input( ) 함수

- 키보드로 입력받아 변수에 저장하는 함수이다.
- 입력되는 값은 문자열로 취급되어 저장된다.

예 a = input('입력하세요.')
– **입력하세요.**가 출력되고 커서가 깜빡거리며 입력을 기다린다.
– 키보드로 값을 입력하면 변수 a에 저장된다.

183 초치기 Python의 print( ) 함수

인수로 주어진 값을 화면에 출력하는 함수이다.

예 print(82, 24, sep = '–', end = ',')
82와 24 사이에 분리문자 '–'가 출력되고, 마지막에 종료문자 ','가 출력된다.
결과 **82–24,**

184 초치기 입력 값의 형변환

- input( ) 함수는 입력되는 값을 무조건 문자열로 저장하므로 숫자로 사용하기 위해서는 형을 변환해야 합니다.
- 변환할 데이터가 1개일 때

예 a = int(input( ))
input( )으로 입력받은 값을 정수로 변환하여 변수 a에 저장한다.

- 변환할 데이터가 2개 이상일 때

예 a, b = map(int, input( ).split( ))
input( ).split( )으로 입력받은 2개의 값을 정수로 변환하여 변수 a, b에 저장한다.

185 초치기 Python의 리스트(List)

- 리스트는 필요에 따라 개수를 늘이거나 줄일 수 있기 때문에 리스트를 선언할 때 크기를 적지 않는다.
- 배열과 달리 하나의 리스트에 정수, 실수, 문자열 등 다양한 자료형을 섞어서 저장할 수 있다.
- Python에서 리스트의 위치는 0부터 시작한다.

예1 방법1 : a = [10, 'mike', 23.45]
방법2 : a = list([10, 'mike', 23.45])

결과	리스트 a	a[0]	a[1]	a[2]
		10	'mike'	23.45

예2 a[0] = 1 → a[0]에 1을 저장한다.

결과	리스트 a	a[0]	a[1]	a[2]
		1	'mike'	23.45

186 초치기 Python의 딕셔너리(Dictionary)

- 연관된 값을 묶어서 저장하는 용도로 딕셔너리를 사용한다.
- 리스트가 저장된 요소에 접근하기 위한 키로 위치값인 0, 1, 2 등의 숫자를 사용했다면, 딕셔너리에서는 사용자가 원하는 키를 직접 지정한 후 사용한다.
- 딕셔너리에 접근할 때는 딕셔너리 뒤에 대괄호([ ])를 사용하며, 대괄호([ ]) 안에 키를 지정한다.

예1 방법1 : a = {'이름':'홍길동', '나이':25, '주소':'서울'}
방법2 : a = dict({'이름':'홍길동', '나이':25, '주소':'서울'})

결과	리스트 a	a['이름']	a['나이']	a['주소']
		'홍길동'	25	'서울'

예2 a['이름'] = '이순신' → 딕셔너리 a의 '이름' 위치에 '이순신'을 저장한다.

결과	리스트 a	a['이름']	a['나이']	a['주소']
		'이순신'	25	'서울'

187 초치기 Python의 Range

연속된 숫자를 생성하는 것으로, 리스트, 반복문 등에서 많이 사용된다.

예1 a = list(range(5))
0에서 4까지 연속된 숫자를 리스트 a로 저장한다.

리스트 a	0	1	2	3	4

예2 b = list(range(4, 9))
4에서 8까지 연속된 숫자를 리스트 b로 저장한다.

리스트 b	4	5	6	7	8

예3 c = list(range(1, 15, 3))
1에서 14까지 3씩 증가하는 숫자들을 리스트 c로 저장한다.

리스트 c	1	4	7	10	13

188 초치기 Python의 슬라이스

문자열이나 리스트와 같은 순차형 객체에서 일부를 잘라(slicing) 반환하는 기능이다.

예 a = ['a', 'b', 'c', 'd', 'e'] 일때
a[1:3] → ['b', 'c']
a[0:5:2] → ['a', 'c', 'e']
a[3:] → ['d', 'e']
a[:3] → ['a', 'b', 'c']
a[::3] → ['a', 'd']

189 초치기 Python의 for문

- range를 이용하는 방식

예

```
for i in range(1, 11):
    sum = sum + i
```

i에 1부터 10까지 순서대로 저장하며 sum에 i의 값을 누적시키는 실행문을 반복 수행한다.

• 리스트(List)를 이용하는 방식

예

```
a = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]
for i in a:
    sum = sum + i
```

a 리스트에 저장된 10개의 요소를 i에 순서대로 저장하며 sum에 i의 값을 누적시키는 실행문을 반복 수행한다.

190 초치기 Python의 while문

예

```
while i <= 10:
    i = i + 1
```

i가 10보다 작거나 같은 동안 i의 값을 1씩 누적시킨다.

191 초치기 Python의 클래스

클래스를 사용하려면 클래스 이름을 정하고 객체 생성을 위한 속성과 메소드(함수)를 정의한 후, 객체를 선언하면 된다.

예제 다음 Python으로 구현된 프로그램의 출력 결과를 확인하시오.

```
    class Cls:
        x = 10
④       def add(self, a):
⑤           return a + self.x
①   a = Cls( )
②   a.x = 5
③⑥ print(a.add(5))
```

① Cls 클래스의 객체 a를 생성한다.
② 객체 a의 변수 x에 5를 저장한다. (a.x = 5)
③ 5를 인수로 a 객체의 add( ) 메소드를 호출한 후 돌려받은 값을 출력한다.
④ add( ) 메소드의 시작점이다. ③번에서 전달받은 5를 a가 받는다.
⑤ a와 객체의 변수 x를 더한 값 10을 메소드를 호출했던 ⑥번으로 반환한다.
⑥ ⑤번에서 돌려받은 값 10을 출력한다.

결과 **10**

192 초치기 Python의 클래스 없는 메소드

C언어의 사용자 정의 함수와 같이 클래스 없이 메소드만 단독으로 사용할 수 있다.

예제 다음 Python으로 구현된 프로그램의 출력 결과를 확인하시오.

```
③ def calc(x, y):
④     x *= y
⑤     return x
① a, b = 3, 4
② a = calc(a, b)
⑥ print(a, b)
```

① 변수 a와 b에 3과 4를 저장한다.
② a, b 즉 3과 4를 인수로 하여 calc 메소드를 호출한 결과를 a에 저장한다.
③ calc( ) 메소드의 시작점이다. ②번에서 전달받은 3과 4를 x와 y가 받는다.
④ x = x * y이므로 x는 12가 된다.
⑤ x의 값을 반환한다. x의 값 12를 ②번의 a에 저장한다.
⑥ 결과 **12, 4**

193 초치기 스크립트 언어의 종류

- 자바스크립트 : 웹 페이지의 동작을 제어하는 데 사용되는 클라이언트용 스크립트 언어
- PHP : Linux, Unix, Windows 운영체제에서 사용 가능한 서버용 스크립트 언어
- 파이썬(Python) : 귀도 반 로섬이 발표한 대화형 인터프리터 언어
- 쉘 스크립트 : 쉘에서 사용되는 명령어들의 조합으로 구성된 스크립트 언어
- Basic : 절차지향 기능을 지원하는 대화형 인터프리터 언어

194 초치기 쉘 스크립트에서 사용되는 제어문

- 선택형 : if, case
- 반복형 : for, while, until

195 초치기 라이브러리

- 표준 라이브러리 : 프로그래밍 언어에 기본적으로 포함되어 있는 라이브러리로, 여러 종류의 모듈이나 패키지로 구성됨
- 외부 라이브러리 : 개발자들이 필요한 기능들을 만들어 인터넷 등에 공유해 놓은 것으로, 외부 라이브러리를 다운받아 설치한 후 사용함

196 초치기 C언어의 stdlib.h

- 자료형 변환, 난수 발생, 메모리 할당에 사용되는 기능들을 제공한다.
- 주요 함수 : atoi, atof, srand, rand, malloc, free 등

197 초치기 UNIX의 특징

- 대부분 C 언어로 작성되어 있어 이식성이 높다.
- 장치와 프로세스 간의 호환성이 높다.
- 다중 사용자(Multi-User), 다중 작업(Multi-Tasking)을 지원한다.
- 트리 구조의 파일 시스템을 갖는다.

198 초치기 UNIX - 커널(Kernel)의 기능

- 프로세스(CPU 스케줄링) 관리
- 기억장치 관리
- 파일 시스템 관리
- 입 · 출력 관리

199 초치기 UNIX - 쉘(Shell)

- 사용자의 명령어를 인식하여 프로그램을 호출하고 명령을 수행하는 명령어 해석기이다.
- 시스템과 사용자 간의 인터페이스를 담당한다.

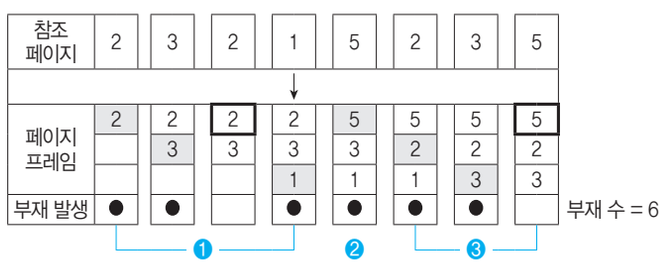
200 초치기 기억장치의 배치 전략

- 최초 적합(First Fit) : 첫 번째 분할 영역에 배치
- 최적 적합(Best Fit) : 단편화를 가장 작게 남기는 분할 영역에 배치
- 최악 적합(Worst Fit) : 단편화를 가장 많이 남기는 분할 영역에 배치

201 초치기 페이지 교체 알고리즘 - FIFO

각 페이지가 주기억장치에 적재될 때마다 그때의 시간을 기억시켜 가장 먼저 들어와서 가장 오래 있었던 페이지를 교체하는 기법이다.

예 다음의 참조 페이지를 세 개의 페이지 프레임을 가진 기억장치에서 FIFO 알고리즘을 사용하여 교체했을 때 페이지 부재의 수는? (단, 초기 페이지 프레임은 모두 비어있는 상태이다.)



❶ 참조 페이지를 각 페이지 프레임에 차례로 적재시키되 이미 적재된 페이지는 해당 위치의 페이지 프레임을 사용한다.
❷ 사용할 페이지 프레임이 없을 경우 가장 먼저 들어와서 오래 있었던 페이지 2를 제거한 후 5를 적재한다.
❸ 그 다음에 적재된 페이지 3을 제거한 후 2를 적재하며, 같은 방법으로 나머지 참조 페이지를 수행한다.

202 초치기 스래싱(Thrashing)

프로세스의 처리 시간보다 페이지 교체에 소요되는 시간이 더 많아지는 현상이다.

203 초치기 프로세스 상태 및 상태 전이

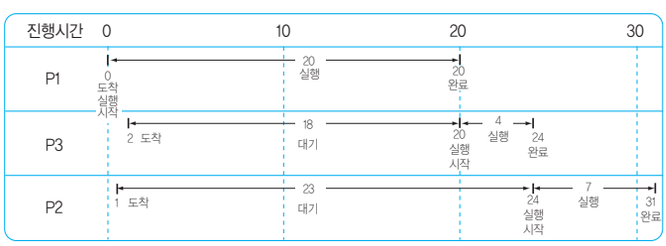
- 프로세스의 상태 : 제출(Submit), 접수(Hold), 준비(Ready), 실행(Run), 대기(Wait, 보류, 블록), 종료(Terminated, Exit)
- 프로세스 상태 전이 : Dispatch, Wake-Up, Spooling

204 초치기 스케줄링 - SJF

실행 시간이 가장 짧은 프로세스에게 먼저 CPU를 할당하는 기법이다.

예제 다음과 같은 프로세스들이 차례로 준비상태 큐에 들어왔다고 가정할 때, SJF 기법을 이용하여 평균 실행 시간, 평균 대기 시간, 평균 반환 시간을 구하시오(제출 시간이 있을 경우).

프로세스 번호	P1	P2	P3
실행 시간	20	7	4
도착 시간	0	1	2



- 평균 실행 시간 : (20+4+7)/3 = 10.3
- 평균 대기 기간 : (0+18+23)/3 = 13.6
- 평균 반환 시간 : (20+22+30)/3 = 24

205 초치기 스케줄링 - HRN

예 HRN 방식으로 스케줄링할 경우, 입력된 작업이 다음과 같을 때 우선순위가 가장 높은 작업은?

작업	대기시간	서비스시간
A	5	5
B	10	6
C	15	7
D	20	8

- HRN 방식의 우선순위 계산식은 $\frac{\text{대기 시간} + \text{서비스 시간}}{\text{서비스 시간}}$ 이므로 이 식에 대입해서 풀면 된다.
- A 작업은 $\frac{5+5}{5} = 2$
- B 작업은 $\frac{10+6}{6} = 2.6$
- C 작업은 $\frac{15+7}{7} = 3.1$
- D 작업은 $\frac{20+8}{8} = 3.5$이다.
- 결과가 큰 값이 우선순위가 높다.

206 초치기 UNIX의 주요 명령어

- fork : 새로운 프로세스를 생성함
- uname : 시스템 정보를 표시함
- wait : 상위 프로세스가 하위 프로세스 종료 등의 event를 기다림
- chmod : 파일의 보호 모드를 설정함
- ls : 현재 디렉터리 내의 파일 목록을 확인함
- cat : 파일 내용을 화면에 표시함
- chown : 소유자를 변경함

207 초치기 인터넷 주소 체계 - IPv4

- 8비트씩 4부분, 총 32비트로 구성되어 있다.
- A 클래스에서 E 클래스까지 총 5단계로 구성되어 있다.

208 초치기 인터넷 주소 체계 - IPv6

- 16비트씩 8부분, 총 128비트로 구성되어 있다.
- 각 부분을 16진수로 표현하고, 콜론(:)으로 구분한다.
- 주소의 확장성, 융통성, 연동성이 뛰어나고, 품질 보장이 용이하다.
- IPv6 주소 체계 : 유니캐스트(Unicast), 멀티캐스트(Multicast), 애니캐스트(Anycast)

209 초치기 OSI 7계층 - 데이터 링크 계층 (Data Link Layer)

- 인접한 개방 시스템들 간에 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 한다.
- 기능 : 흐름 제어, 프레임 동기화, 오류 제어, 순서 제어 등
- 프로토콜 : HDLC, LAPB, PPP, LLC 등

210 초치기 OSI 7계층 - 네트워크 계층 (Network Layer)

- 개방 시스템들 간의 네트워크 연결을 관리하고 데이터를 교환 및 중계한다.
- 기능 : 경로 설정, 트래픽 제어, 패킷 정보 전송 등

211 초치기 OSI 7계층 - 전송 계층 (Transport Layer)

- 종단 시스템(End-to-End) 간에 투명한 데이터 전송을 가능하게 한다.
- 기능 : 전송 연결 설정, 데이터 전송, 연결 해제 기능, 주소 설정, 다중화, 오류 제어, 흐름 제어 등

212 초치기 OSI 7계층 - 세션 계층 (Session Layer)

- 송 · 수신 측 간의 관련성을 유지하고 대화 제어를 담당한다.
- 대화의 생성, 관리, 종료를 위해 토큰을 사용한다.
- 기능 : 대화 구성 및 동기 제어, 데이터 교환 관리 등

213 초치기 네트워크 관련 주요 장비

- 리피터(Repeater) : 원래의 신호 형태로 재생하여 다시 전송함
- 브리지(Bridge) : LAN과 LAN을 연결하거나 LAN 안에서의 컴퓨터 그룹을 연결함
- 라우터(Router) : 데이터 전송의 최적 경로를 선택함
- 스위치(Switch) : LAN과 LAN을 연결하여 훨씬 더 큰 LAN을 만드는 장치
- 브라우터(Brouter) : 브리지와 라우터의 기능을 모두 수행하는 장치로, 브리지 기능은 내부 네트워크를 분리하는 용도로 사용하고 라우터 기능은 외부 네트워크에 연결하는 용도로 사용함

214 초치기 TCP/IP 프로토콜 - MQTT

발행-구독 기반의 메시징 프로토콜로, IoT 환경에서 자주 사용된다.

215 초치기 TCP/IP 프로토콜 - TCP

- 신뢰성 있는 연결형 서비스를 제공한다.
- 양방향 연결형 서비스를 제공한다.
- 기능 : 패킷의 다중화, 순서 제어, 오류 제어, 흐름 제어, 스트림 전송 등

216 초치기 TCP/IP 프로토콜 - UDP

- 비연결형 서비스를 제공한다.
- 흐름 제어나 순서 제어가 없어 전송 속도가 빠르다.
- 실시간 전송에 유리하다.

217 초치기 TCP/IP 프로토콜 - ARP

호스트의 IP 주소를 호스트와 연결된 네트워크 접속 장치의 물리적 주소(MAC Address)로 바꾼다.

218 초치기 CSMA/CD

IEEE 802.3 LAN에서 사용되는 전송 매체 접속 제어(MAC) 방식이다

5과목 정보시스템 구축 관리

219 초치기 구조적 방법론

- 정형화된 분석 절차에 따라 사용자 요구사항을 파악하여 문서화하는 처리(Precess) 중심의 방법론이다.
- 복잡한 문제를 다루기 위해 분할과 정복(Divide and Conquer) 원리를 적용한다.

220 초치기 정보공학 방법론

- 정보 시스템의 개발을 위해 정형화된 기법들을 상호 연관성 있게 통합 및 적용하는 자료(Data) 중심의 방법론이다.
- 데이터베이스 설계를 위한 데이터 모델링으로 개체 관계도(ERD; Entity-Relationship Diagram)를 사용한다.

221 초치기 컴포넌트 기반(CBD) 방법론

- 기존의 시스템이나 소프트웨어를 구성하는 컴포넌트를 조합하여 하나의 새로운 애플리케이션을 만드는 방법론이다.
- 분석 단계에서 사용자 요구사항 정의서가 산출된다.

222 초치기 소프트웨어 재사용의 이점

- 개발 시간과 비용 단축
- 소프트웨어 품질 향상
- 소프트웨어 개발의 생산성 향상
- 시스템 명세, 설계, 코드 등 문서 공유

223 소프트웨어 재사용 방법

- 합성 중심 : 전자 칩과 같은 소프트웨어 부품, 즉 블록(모듈)을 만들어서 끼워 맞추어 소프트웨어를 완성시키는 방법
- 생성 중심 : 추상화 형태로 쓰여진 명세를 구체화하여 프로그램을 만드는 방법

224 소프트웨어 재공학의 이점

- 위험 부담 감소
- 개발 시간 단축
- 개발 비용 절감
- 시스템 명세의 오류 억제

225 소프트웨어 재공학의 주요 활동

- 분석(Analysis) : 기존 소프트웨어의 명세서를 확인하여 소프트웨어의 동작을 이해하고, 재공학할 대상을 선정하는 활동
- 재구성(Restructuring) : 기존 소프트웨어의 구조를 향상시키기 위하여 코드를 재구성하는 활동
- 역공학(Reverse Engineering) : 기존 소프트웨어를 분석하여 소프트웨어 개발 과정과 데이터 처리 과정을 설명하는 분석 및 설계 정보를 재발견하거나 다시 만들어 내는 활동
- 이식(Migration) : 기존 소프트웨어를 다른 운영체제나 하드웨어 환경에서 사용할 수 있도록 변환하는 활동

226 CASE(Computer Aided Software Engineering)

소프트웨어 개발 과정 전체 또는 일부를 컴퓨터와 전용 소프트웨어 도구를 사용하여 자동화하는 것이다.

227 CASE의 원천 기술

- 구조적 기법
- 프로토타이핑
- 자동 프로그래밍
- 정보 저장소
- 분산처리

228 CASE의 주요 기능

- 소프트웨어 생명 주기 전 단계의 연결
- 다양한 소프트웨어 개발 모형 지원
- 모델들의 모순 검사 및 오류 검증
- 그래픽 지원
- 자료 흐름도 작성 등

229 비용 산정 기법 - LOC 기법

소프트웨어 각 기능의 원시 코드 라인 수의 비관치, 낙관치, 기대치를 측정하여 예측치를 구하고 이를 이용하여 비용을 산정하는 기법이다.

230 LOC 기법의 산정 공식

- 노력(인월)
  - 개발 기간 × 투입 인원
  - LOC / 1인당 월평균 생산 코드 라인 수
- 개발 비용 : 노력(인월) × 단위 비용(1인당 월평균 인건비)
- 개발 기간 : 노력(인월) / 투입 인원
- 생산성 : LOC / 노력(인월)

231 수학적 산정 기법의 종류

- COCOMO(COnstructive COst MOdel) 모형
- Putnam 모형
- 기능 점수(Function Point) 모형

232 비용 산정 기법 - Putnam 모형

- 소프트웨어 생명 주기의 전 과정 동안에 사용될 노력의 분포를 가정해 주는 모형이다.
- Rayleigh-Norden 곡선의 노력 분포도를 기초로 한다.

233 비용 산정 기법 - COCOMO

보헴(Boehm)이 제안한 것으로, 원시 프로그램의 규모(LOC)에 의한 비용 산정 기법이다.

234 COCOMO의 소프트웨어 개발 유형

- 조직형(Organic Mode) : 기관 내부에서 개발된 중·소 규모의 소프트웨어로 일괄 자료 처리나 과학기술 계산용, 비즈니스 자료 처리용으로 5만(50KDSI) 라인 이하의 소프트웨어를 개발하는 유형
- 반분리형(Semi-Detached Mode) : 조직형과 내장형의 중간형으로 트랜잭션 처리 시스템이나 운영체제, 데이터베이스 관리 시스템 등의 30만(300KDSI) 라인 이하의 소프트웨어를 개발하는 유형
- 내장형(Embedded Mode) : 내장형은 초대형 규모의 트랜잭션 처리 시스템이나 운영체제 등의 30만(300KDSI) 라인 이상의 소프트웨어를 개발하는 유형

235 자동화 추정 도구 - SLIM

Rayleigh-Norden 곡선과 Putnam 예측 모델을 기초로 하여 개발된 자동화 추정 도구이다.

236 기능 점수(FP) 모형 - 가중치 증대 요인

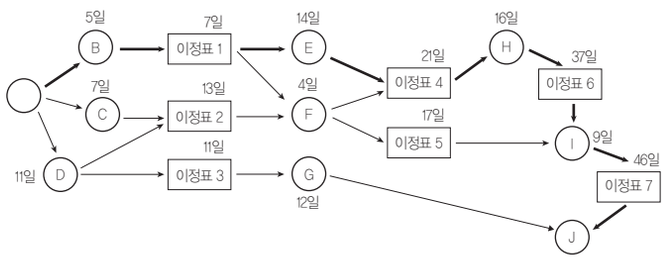
- 자료 입력(입력 양식)
- 정보 출력(출력 보고서)
- 명령어(사용자 질의수)
- 데이터 파일
- 필요한 외부 루틴과의 인터페이스

237 PERT (프로그램 평가 및 검토 기술)

- 각 작업별로 낙관적인 경우, 가능성이 있는 경우, 비관적인 경우로 나누어 각 단계별 종료 시기를 결정하는 방법이다.
- 결정 경로, 작업에 대한 경계 시간, 작업 간의 상호 관련성 등을 알 수 있다.

238 임계 경로 소요 기일

임계 경로는 최장 경로(굵은 선)를 의미한다.



239 간트 차트(Gantt Chart)

- 프로젝트의 작업 일정을 막대 도표를 이용하여 표시하는 프로젝트 일정표이다.
- 수평 막대의 길이는 각 작업(Task)의 기간을 나타낸다.

240 소프트웨어 프로젝트 관리

주어진 기간 내에 최소의 비용으로 사용자를 만족시키는 시스템을 개발하기 위한 전반적인 활동을 의미한다.

241 위험 관리(Risk Analysis)

프로젝트 추진 과정에서 예상되는 각종 돌발 상황(위험)을 미리 예상하고 이에 대한 적절한 대책을 수립하는 일련의 활동을 의미한다.

242 ISO/IEC 12207

- 기본 생명 주기 프로세스 : 획득, 공급, 개발, 운영, 유지보수 프로세스
- 지원 생명 주기 프로세스 : 품질 보증, 검증, 확인, 활동 검토, 감사, 문서화, 형상 관리, 문제 해결 프로세스
- 조직 생명 주기 프로세스 : 관리, 기반 구조, 훈련, 개선 프로세스

243 CMMI의 소프트웨어 프로세스 성숙도 5단계

- 초기(Initial)
- 관리(Managed)
- 정의(Defined)
- 정량적 관리(Quantitatively Managed)
- 최적화(Optimizing)

244 SPICE(소프트웨어 처리 개선 및 능력 평가 기준)

소프트웨어의 품질 및 생산성 향상을 위해 소프트웨어 프로세스를 평가 및 개선하는 국제 표준이다.

245 SPICE의 프로세스 수행 능력 단계

- Level 0 – 불완전(Incomplete)
- Level 1 – 수행(Performed)
- Level 2 – 관리(Managed)
- Level 3 – 확립(Established)
- Level 4 – 예측(Predictable)
- Level 5 – 최적화(Optimizing)

246 소프트웨어 개발 방법론 테일러링

프로젝트 상황 및 특성에 맞도록 정의된 소프트웨어 개발 방법론의 절차, 사용기법 등을 수정 및 보완하는 작업이다.

247 소프트웨어 개발 방법론 테일러링 고려사항

내부적 기준

- 목표 환경 : 시스템의 개발 환경과 유형
- 요구사항 : 개발, 운영, 유지보수 등
- 프로젝트 규모 : 비용, 인력, 기간 등
- 보유 기술 : 프로세스, 개발 방법론, 산출물, 구성원의 능력 등

외부적 기준

- 법적 제약사항 : 프로젝트별로 적용될 IT Compliance
- 표준 품질 기준 : 금융, 제도 등 분야별 표준 품질 기준

248 소프트웨어 개발 프레임워크

- 소프트웨어 개발에 공통적으로 사용되는 구성 요소와 아키텍처를 일반화하여 손쉽게 구현할 수 있도록 여러 가지 기능들을 제공해주는 반제품 형태의 소프트웨어 시스템이다.
- 선행 사업자의 기술에 의존하지 않은 표준화된 개발 기반으로 인해 사업자 종속성이 해소된다.
- 개발자가 관리하고 통제해야 하는 객체들의 제어를 프레임워크에 넘김으로써 생산성을 향상시킨다.

249 SDN(Software Defined Networking)

네트워크를 컴퓨터처럼 모델링하여 여러 사용자가 각각의 소프트웨어들로 네트워킹을 가상화하여 제어하고 관리하는 네트워크이다.

250 SDS (Software-Defined Storage)

물리적인 데이터 스토리지(Data Storage)를 가상화하여 여러 스토리지를 하나처럼 관리하는 기술이다.

251 SDDC(Software Defined Data Center)

데이터 센터의 모든 자원을 가상화하여 인력의 개입없이 소프트웨어 조작만으로 관리 및 제어되는 데이터 센터이다.

252 메시 네트워크 (Mesh Network)

수십에서 수천 개의 디바이스를 그물망(Mesh)과 같이 유기적으로 연결하여 모든 구간을 동일한 무선망처럼 구성하여 사용자가 안정적인 네트워크를 사용할 수 있게 한다.

253 피코넷(PICONET)

여러 개의 독립된 통신장치가 블루투스 기술이나 UWB (Ultra Wide Band) 통신 기술을 사용하여 통신망을 형성하는 무선 네트워크 기술이다.

254 클라우드 기반 HSM

- 클라우드를 기반으로 암호화 키의 생성 · 저장 · 처리 등의 작업을 수행하는 보안기기를 가리키는 용어이다.
- 암호화 키 생성이 하드웨어적으로 구현되기 때문에 소프트웨어적으로 구현된 암호 기술이 가지는 보안 취약점을 무시할 수 있다.

255 파스-타(PaaS-TA)

소프트웨어 개발 환경을 제공하기 위해 개발한 개방형 클라우드 컴퓨팅 플랫폼이다.

256 징(Zing)

10cm 이내 거리에서 3.5Gbps 속도의 데이터 전송이 가능한 초고속 근접무선통신(NFC)이다.

257 SSO(Single Sign On)

한 번의 로그인으로 개인이 가입한 모든 사이트를 이용할 수 있게 해주는 시스템이다.

258 스마트 그리드(Smart Grid)

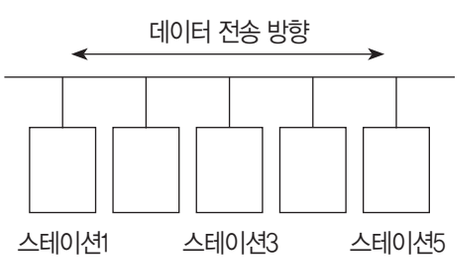
전력선을 기반으로 모든 통신, 정보, 관련 애플리케이션 인프라를 하나의 시스템으로 통합하여 관리함으로써 효율적인 에너지 관리가 가능하다.

259 WDM(Wavelength Division Multiplexing)

파장이 다른 광선끼리는 서로 간섭을 일으키지 않는 성질을 이용하여 여러 대의 단말기가 동시에 통신 회선을 사용할 수 있도록 하는 기술이다.

260 버스형(Bus)

한 개의 통신 회선에 여러 대의 단말장치가 연결된 형태이다.



261 VLAN(Virtual Local Area Network)

LAN의 물리적인 배치와 상관없이 논리적으로 분리하는 기술로, 접속된 장비들의 성능 및 보안성을 향상시킬 수 있다.

262 WPA (Wi-Fi Protected Access)

Wi-Fi에서 제정한 무선 랜(WLAN) 인증 및 암호화 관련 표준이다.

263 CSMA/CA

무선 랜에서 데이터 전송 시 매체가 비어있음을 확인한 뒤 충돌을 피하기 위해 일정한 시간을 기다린 후 데이터를 전송하는 방법이다.

264 LAN의 표준 규격 - 802.11e

802.11의 부가 기능 표준으로, QoS 기능이 지원되도록 하기 위해 매체 접근 제어(MAC) 계층에 해당하는 부분을 수정하였다.

265 RIP(Routing Information Protocol)

- IGP에 속하는 라우팅 프로토콜로, 거리 벡터 라우팅 프로토콜이라고도 불린다.
- 최대 홉(Hop) 수를 15로 제한한다.

266 OSPF(Open Shortest Path First protocol)

라우팅 정보에 노드 간의 거리 정보, 링크 상태 정보를 실시간으로 반영하여 최단 경로로 라우팅을 지원하는 프로토콜

267 흐름 제어 - 정지-대기 (Stop-and-Wait)

- 수신 측의 확인 신호(ACK)를 받은 후에 다음 패킷을 전송하는 방식이다.
- 한 번에 하나의 패킷만을 전송할 수 있다.

268 도커(Docker)

컨테이너 기술을 자동화하여 쉽게 사용할 수 있게 하는 오픈소스 프로젝트이다.

269 스크래피(Scrapy)

Python 기반의 웹 크롤링 프레임워크로, 코드 재사용성을 높이는 데 도움이 되며, 대규모의 크롤링 프로젝트에 적합하다.

270 텐서플로(TensorFlow)

구글의 구글 브레인(Google Brain) 팀이 만든, 다양한 작업에 대해 데이터 흐름 프로그래밍을 위한 오픈소스 소프트웨어 라이브러리이다.

271 서비스 지향 아키텍처(SOA) 기반 애플리케이션 구성 계층

- 표현(Presentation) 계층
- 업무 프로세스(Biz-Process) 계층
- 서비스 중간(Service Intermediary) 계층
- 애플리케이션(Application) 계층
- 데이터 저장(Persistency) 계층

272 매시업(Mashup)

웹에서 제공하는 정보 및 서비스를 이용하여 새로운 소프트웨어나 서비스, 데이터베이스 등을 만드는 기술이다.

273 디지털 트윈(Digital Twin)

현실 속의 사물을 소프트웨어로 가상화한 모델이다.

274 서비스형 블록체인(BaaS)

블록체인(Blockchain) 앱의 개발 환경을 클라우드 기반으로 제공하는 서비스이다.

275 TCP 래퍼(TCP Wrapper)

외부 컴퓨터의 접속 인가 여부를 점검하여 접속을 허용 및 거부하는 보안용 도구이다.

276 DPI(Deep Packet Inspection)

OSI 7 Layer 전 계층의 프로토콜과 패킷 내부의 콘텐츠를 파악하여 침입 시도, 해킹 등을 탐지하고, 트래픽을 조정하기 위한 패킷 분석 기술이다.

277 허니팟(Honeypot)

비정상적인 접근의 탐지를 위해 의도적으로 설치해 둔 시스템이다.

278 OWASP(오픈 웹 애플리케이션 보안 프로젝트)

웹 정보 노출이나 악성 코드, 스크립트, 보안이 취약한 부분을 연구하는 비영리 단체이다.

279 고가용성 솔루션(HACMP)

긴 시간 동안 안정적인 서비스 운영을 위해 장애 발생 시 즉시 다른 시스템으로 대체 가능한 환경을 구축하는 메커니즘을 의미한다.

280 앤 스크린(N-Screen)

N개의 서로 다른 단말기에서 동일한 콘텐츠를 자유롭게 이용할 수 있는 서비스를 말한다.

281 Secure OS

- 기존의 운영체제(OS)에 보안 기능을 갖춘 커널을 이식하여 시스템 자원을 보호하는 운영체제이다.
- Secure OS의 보안 기능 : 식별 및 인증, 임의적/강제적 접근통제, 객체 재사용 보호, 완전한 조정, 신뢰 경로, 감사 및 감사기록 축소 등

282 하둡(Hadoop)

- 오픈 소스를 기반으로 한 분산 컴퓨팅 플랫폼이다.
- 하둡과 관계형 데이터베이스(RDB) 간 대용량 데이터를 전송할 때 스쿱(Sqoop)이라는 도구를 이용한다.

283 맵리듀스(MapReduce)

대용량 데이터를 분산 처리하기 위한 목적으로 개발된 프로그래밍 모델이다.

284 데이터 마이닝(Data Mining)

데이터 웨어하우스에 저장된 대량의 데이터 집합에서 사용자의 요구에 따라 유용하고 가능성 있는 정보를 발견하기 위한 기법이다.

285 OLAP(Online Analytical Processing)

- 다차원으로 이루어진 데이터로부터 통계적인 요약 정보를 분석하여 의사결정에 활용하는 방식을 말한다.
- OLAP 연산 : Roll-up, Drill-down, Drill-through, Drill-across, Pivoting, Slicing, Dicing

286 회복(Recovery)

트랜잭션들을 수행하는 도중 장애가 발생하여 데이터베이스가 손상되었을 때 손상되기 이전의 정상 상태로 복구하는 작업이다.

287 즉각 갱신 기법 (Immediate Update)

- 트랜잭션이 데이터를 갱신하면 트랜잭션이 부분 완료되기 전이라도 즉시 실제 데이터베이스에 반영하는 방법이다.
- 장애가 발생하여 회복 작업할 경우를 대비하여 갱신된 내용들은 Log에 보관시킨다.

288 로킹 단위 (Locking Granularity)

- 병행제어에서 한꺼번에 로킹할 수 있는 객체의 크기를 의미한다.
- 로킹 단위가 크면 로크 수가 작아 관리하기 쉽지만 병행성 수준이 낮아진다.
- 로킹 단위가 작으면 로크 수가 많아 관리하기 복잡해 오버헤드가 증가하지만 병행성 수준이 높아진다.

289 타임 스탬프 순서 (Time Stamp Ordering)

직렬성 순서를 결정하기 위해 트랜잭션 간의 처리 순서를 미리 선택하는 기법들 중에서 가장 보편적인 방법이다.

290 교착상태 발생의 필요 충분 조건 4가지

- 상호 배제(Mutual Exclusion)
- 점유와 대기(Hold and Wait)
- 비선점(Non-preemption)
- 환형 대기(Circular Wait)

291 회피 기법(Avoidance)

- 교착상태가 발생할 가능성을 배제하지 않고 교착상태가 발생하면 적절히 피해 나가는 방법으로, 주로 은행원 알고리즘(Banker's Algorithm)이 사용된다.
- 은행원 알고리즘(Banker's Algorithm) : E. J. Dijkstra가 제안한 것으로, 은행에서 모든 고객의 요구가 충족되도록 현금을 할당하는 데서 유래한 기법

292 Seven Touchpoints

소프트웨어 보안의 모범사례를 SDLC에 통합한 방법론이다.

293 보안 3대 요소

- 기밀성 : 시스템 내의 정보와 자원은 인가된 사용자에게만 접근이 허용되며, 정보가 전송 중에 노출되더라도 데이터를 읽을 수 없음
- 무결성 : 시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있음
- 가용성 : 인가받은 사용자는 언제라도 사용할 수 있음

294 세션 하이재킹 (Session Hijacking)

- 서버에 접속하고 있는 클라이언트들의 세션 정보를 가로채는 공격기법으로, 세션 가로채기라고도 한다.
- 탐지 방법 : 비동기화 상태 탐지, ACK Storm 탐지, 패킷의 유실 탐지, 예상치 못한 접속의 리셋 탐지 등

295 SQL 삽입(SQL Injection)

웹 응용 프로그램에 SQL을 삽입하여 내부 데이터베이스(DB) 서버의 데이터를 유출 및 변조하고, 관리자 인증을 우회하는 보안 약점이다.

296 경로 조작 및 자원 삽입

데이터 입출력 경로를 조작하여 서버 자원을 수정 · 삭제할 수 있는 보안 약점이다.

297 크로스사이트 스크립팅(XSS; Cross Site Scripting)

웹페이지에 악의적인 스크립트를 삽입하여 방문자들의 정보를 탈취하거나, 비정상적인 기능 수행을 유발하는 보안 약점이다.

298 메모리 버퍼 오버플로

할당된 메모리의 범위를 넘어선 위치에서 자료를 읽거나 쓰려고 할 때 발생하는 보안 약점이다.

299 하드코드된 비밀번호

- 소스코드 유출 시 내부에 하드코드된 패스워드를 이용하여 관리자 권한을 탈취할 수 있다.

※ 하드코드 : 데이터를 코드 내부에 직접 입력하여 프로그래밍하는 방식

300 스택 가드(Stack Guard)

메모리상에서 프로그램의 복귀 주소와 변수 사이에 특정 값을 저장한 후 그 값이 변경되었을 경우 오버플로우 상태로 판단하여 프로그램 실행을 중단함으로써 잘못된 복귀 주소의 호출을 막는 기술이다.

301 접근 지정자(접근 제어자)

- 프로그래밍 언어에서 특정 개체를 선언할 때 외부로부터의 접근을 제한하기 위해 사용되는 예약어이다.
- 종류 : Public, Protected, Default, Private

302 개인키 암호화(Private Key Encryption) 기법

- 동일한 키로 데이터를 암호화하고 복호화한다.
- 알고리즘이 단순하므로 암호화/복호화 속도가 빠르다.

303 개인키 암호화 기법의 종류

- 블록 암호화 방식
  - 한 번에 하나의 데이터 블록을 암호화하는 방식
  - 종류 : DES, SEED, AES, ARIA, IDEA 등
- 스트림 암호화 방식
  - 평문과 동일한 길이의 스트림을 생성하여 비트/바이트/워드 단위로 암호화하는 방식
  - 종류 : LFSR, RC4 등

304 해시(Hash)

- 임의의 길이의 입력 데이터나 메시지를 고정된 길이의 값이나 키로 변환하는 것을 의미한다.
- 복호화가 거의 불가능한 일방향 함수에 해당한다.
- 해시 함수의 종류 : SHA 시리즈, MD4, MD5, N-NASH, SNEFRU 등

305 공개키 암호화 (Public Key Encryption) 기법

- 데이터를 암호화할 때 사용하는 공개키(Public Key)는 데이터베이스 사용자에게 공개하고, 복호화할 때의 비밀키(Secret Key)는 관리자가 비밀리에 관리한다.
- 암호화 대상이 n개일 때 사용되는 키의 개수는 2n이다.
- 대표적으로 RSA(Rivest Shamir Adleman)가 있다.

306 양방향 알고리즘 종류

- DES(Data ncryption Standard) : 1975년 미국 NBS에서 발표한 64비트의 개인키 암호화 알고리즘
- AES(Advanced Encryption Standard) : DES에 한계를 느낀 미국 표준 기술 연구소(NIST)가 2001년 발표한 개인키 암호화 알고리즘
- RSA(Rivest Shamir Adleman) : 큰 숫자를 소인수분해 하기 어렵다는 것에 기반하여 만들어진 공개키 암호화 알고리즘

307 솔트(Salt)

암호화를 수행하기에 앞서 원문에 무작위의 값을 덧붙이는 과정이다.

308 죽음의 핑(Ping of Death)

Ping 명령을 전송할 때 ICMP 패킷의 크기를 인터넷 프로토콜 허용 범위 이상으로 전송하여 공격 대상의 네트워크를 마비시키는 서비스 거부 공격 방법이다.

309 SMURFING(스머핑)

IP나 ICMP의 특성을 악용하여 엄청난 양의 데이터를 한 사이트에 집중적으로 보냄으로써 네트워크를 불능 상태로 만드는 공격 방법이다.

310 DDoS(Distributed Denial of Service, 분산 서비스 거부) 공격

- 여러 곳에 분산된 공격 지점에서 한 곳의 서버에 대해 분산 서비스 공격을 수행하는 것이다.
- 분산 서비스 공격용 툴의 종류 : Trin00, TFN(Tribe Flooding Network), TFN2K, Stacheldraht 등

311 피싱(Phishing)

이메일이나 메신저 등을 통해 공기관이나 금융 기관을 사칭하여 개인 정보를 빼내는 기법이다.

312 Ping Flood

매우 많은 ICMP 메시지를 보내 이에 대한 응답(Respond)으로 시스템 자원을 모두 사용하게 해 시스템이 정상적으로 동작하지 못하도록 하는 공격 방법이다.

313 스위치 재밍 (Switch Jamming)

위조된 매체 접근 제어(MAC) 주소를 지속해서 네트워크로 흘려보내, 스위치 MAC 주소 테이블의 저장 기능을 혼란시켜 더미 허브(Dummy Hub)처럼 작동하게 하는 공격이다.

314 블루투스(Bluetooth) 관련 공격

- 블루버그(BlueBug) : 블루투스 장비 사이의 취약한 연결 관리를 악용한 공격
- 블루스나프(BlueSnarf) : 블루투스의 취약점을 활용하여 장비의 파일에 접근하는 공격
- 블루프린팅(BluePrinting) : 공격 대상이 될 블루투스 장비를 검색하는 활동
- 블루재킹(BlueJacking) : 블루투스를 이용해 스팸처럼 메시지를 익명으로 퍼뜨리는 공격

315 웜(Worm)

네트워크를 통해 연속적으로 자신을 복제하여 시스템의 부하를 높임으로써 결국 시스템을 다운시키는 바이러스의 일종이다.

316 키로거 공격 (Key Logger Attack)

컴퓨터 사용자의 키보드 움직임을 탐지해 ID, 패스워드, 계좌번호, 카드번호 등과 같은 개인의 중요한 정보를 몰래 빼가는 해킹 공격이다.

317 랜섬웨어(Ransomware)

인터넷 사용자의 컴퓨터에 잠입해 내부 문서나 파일 등을 암호화해 사용자가 열지 못하게 하는 프로그램으로, 암호 해독용 프로그램의 전달을 조건으로 사용자에게 돈을 요구하기도 한다.

318 백도어 (Back Door, Trap Door)

- 시스템 설계자가 액세스 편의를 위해 시스템 보안을 제거하여 만들어놓은 비밀 통로이다.
- **백도어 탐지 방법** : 무결성 검사, 열린 포트 확인, 로그 분석, SetUID 파일 검사 등

319 인증(認證, Authentication)

- 로그인을 요청한 사용자의 정보를 확인하고 접근 권한을 검증하는 보안 절차이다.
- **인증의 유형**
  - 지식 기반 인증 : 고정된 패스워드, 아이핀 등
  - 소유 기반 인증 : 신분증, 토큰, 스마트카드 등
  - 행위 기반 인증 : 서명, 움직임, 음성 등

320 관리적/물리적/기술적 보안

- **관리적 보안** : 정보보호 정책, 정보보호 조직, 정보자산 분류, 정보보호 교육 및 훈련, 인적 보안, 업무 연속성 관리 등의 정의
- **물리적 보안** : 건물 및 사무실 출입 통제 지침, 전산실 관리 지침, 정보 시스템 보호 설치 및 관리지침, 재해 복구 센터 운영 등의 정의
- **기술적 보안** : 사용자 인증, 접근 제어, PC, 서버, 네트워크, 응용 프로그램, 데이터(DB) 등의 보안지침 정의

321 커널 로그의 종류

- wtmp : 성공한 로그인/로그아웃과 시스템의 시작/종료 시간에 대한 로그
- utmp : 현재 로그인한 사용자의 상태에 대한 로그
- btmp : 실패한 로그인에 대한 로그
- lastlog : 마지막으로 성공한 로그인에 대한 로그

322 침입 탐지 시스템(IDS; Intrusion Detection System)

- 컴퓨터 시스템의 비정상적인 사용, 오용, 남용 등을 실시간으로 탐지하는 시스템이다.
- **오용 탐지(Misuse Detection)** : 미리 입력해 둔 공격 패턴이 감지되면 이를 알려줌
- **이상 탐지(Anomaly Detection)** : 평균적인 시스템의 상태를 기준으로 비정상적인 행위나 자원의 사용이 감지되면 이를 알려줌

323 VPN(Virtual Private Network, 가상 사설 통신망)

공중 네트워크와 암호화 기술을 이용하여 사용자가 마치 자신의 전용 회선을 사용하는 것처럼 해주는 보안 솔루션이다.

324 SSH(Secure SHell, 시큐어 셸)

- 다른 컴퓨터에 로그인, 원격 명령 실행, 파일 복사 등을 수행할 수 있도록 다양한 기능을 지원하는 프로토콜 또는 이를 이용한 응용 프로그램이다.
- 기본적으로는 22번 포트를 사용한다.