수험생의 마음으로 만든 책! 시나공 시리즈

2020 시나공

시험에
나오는 것만
공부한다!

정보처리기사 필기

길벗알앤디 지음

길벗
독자의 1초까지 아껴주는
정성을 만나 보세요.

지은이
김정준 – 안양대학교 소프트웨어전공과 교수
길벗알앤디 – 강윤석, 김용갑, 김우경, 김유홍
길벗알앤디는 IT 서적을 기획하고 집필하는 출판 기획 전문 집단으로, 2003년부터 길벗출판사의 IT 수험서인 〈시험에 나오는 것만 공부한다!〉 시리즈를 기획부터 집필 및 편집까지 총괄하고 있다. 20여 년간 자격증 취득에 관한 교육, 연구, 집필에 몰두해 온 강윤석 실장을 중심으로 IT 자격증 시험의 분야별 전문가들이 모여 국내 IT 수험서의 수준을 한 단계 높이기 위한 다양한 연구와 집필 활동에 전념하고 있다.

정보처리기사 필기 – 시나공 시리즈 21
The Written Examination for Engineer Information Processing

초판 발행 · 2020년 7월 27일

발행인 · 이종원
발행처 · (주)도서출판 길벗
출판사 등록일 · 1990년 12월 24일
주소 · 서울시 마포구 월드컵로 10길 56(서교동)
주문 전화 · 02)332-0931 **팩스** · 02)323-0586
홈페이지 · www.gilbut.co.kr **이메일** · gilbut@gilbut.co.kr

기획 및 책임 편집 · 강윤석(kys@gilbut.co.kr), 김미정(kongkong@gilbut.co.kr)
디자인 · 윤석남 **제작** · 이준호, 손일순, 이진혁 **영업마케팅** · 임태호, 전선하
웹마케팅 · 차명환, 지하영 **영업관리** · 김명자 **독자지원** · 송혜란, 홍혜진

편집진행 및 교정 · 길벗알앤디(강윤석 · 김용갑 · 김우경 · 김종일, 김선길) **일러스트** · 윤석남
전산편집 · 예다움 **CTP 출력 및 인쇄** · 예림인쇄 **제본** · 예림바인딩

ⓒ 길벗알앤디, 2019
• 이 책은 저작권법에 따라 보호받는 저작물이므로 무단전재와 무단복제를 금합니다. 이 책의 전부 또는 일부를 이용하려면 반드시 사전에 저작권자와 (주)도서출판 길벗의 서면 동의를 받아야 합니다.
• 잘못된 책은 구입한 서점에서 바꿔 드립니다.

길벗 도서번호 100993

비매품

독자의 1초를 아껴주는 정성 길벗출판사
길벗 | IT실용, IT/일반 수험서, IT전문서, 경제실용서, 취미실용서, 건강실용서, 자녀교육서
더퀘스트 | 인문교양서, 비즈니스서
길벗이지톡 | 어학단행본, 어학수험서
길벗스쿨 | 국어학습서, 수학학습서, 유아학습서, 어학학습서, 어린이교양서, 교과서

페이스북 • www.facebook.com/gilbutzigy
커뮤니티 • http://cafe.naver.com/gilbutit

1 병행학습

001 객체지향 분석 6
002 자료 흐름도/자료 사전 8
003 CASE 11
004 HIPO 14
005 제품 소프트웨어 패키징 도구 16
006 유지보수 17
007 이진 트리의 운행법 19
008 알고리즘의 시간 복잡도/빅오 표기법 23
009 병행제어 25
010 교착상태 27
011 LAN(근거리 통신망) 30
012 비선점 스케줄링 33
013 선점 스케줄링 38
014 데이터 무결성 검사 도구 42

2 최신기출문제

2020년 1 · 2회 통합 정보처리기사 필기 44

3 최종점검 모의고사

1회 최종점검 모의고사 72
2회 최종점검 모의고사 83

4 최종점검 모의고사 해설

1회 최종점검 모의고사 해설 96
2회 최종점검 모의고사 해설 116

동영상 강의

동영상 강의가 제공되는 부분입니다.

방법1 스마트폰으로 QR코드를 스캔하세요.

방법2 시나공 카페의 [프리미엄 존] → [토막강의]에서 QR코드 번호를 입력하세요.

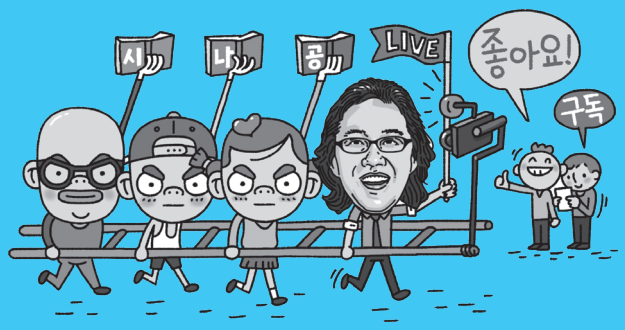
방법3 유튜브 검색 창에 "시나공" +QR 코드 번호를 입력하세요.

예 시나공440201

m·e·m·o

병행학습

001 객체지향 분석
002 자료 흐름도/자료 사전
003 CASE
004 HIPO
005 제품 소프트웨어 패키징 도구
006 유지보수
007 이진 트리의 운행법
008 알고리즘의 시간 복잡도/빅오 표기법
009 병행제어
010 교착상태
011 LAN(근거리 통신망)
012 비선점 스케줄링
013 선점 스케줄링
014 데이터 무결성 검사 도구



병행학습
001
LIVE

객체지향 분석

전문가의 조언

객체지향 분석의 특징을 파악하고, 각 분석 방법론의 개념과 럼바우 방법론에서 사용되는 세 가지 모델링의 특징을 찾아낼 수 있도록 정리하세요.

1 객체지향 분석의 개념

객체지향 분석(OOA; Object Oriented Analysis)은 사용자의 요구사항을 분석하여 요구된 문제와 관련된 모든 클래스(객체), 이와 연관된 속성과 연산, 그들 간의 관계 등을 정의하여 모델링하는 작업이다.

- 소프트웨어를 개발하기 위한 비즈니스(업무)를 객체와 속성, 클래스와 멤버, 전체와 부분 등으로 나누어서 분석한다.
- 분석가에게 주요한 모델링 구성 요소인 클래스, 객체, 속성, 연산들을 표현해서 문제를 모형화할 수 있게 해준다.
- 객체는 클래스로부터 인스턴스화되고, 이 클래스를 식별하는 것이 객체지향 분석의 주요한 목적이다.

2 객체지향 분석의 방법론

객체지향 분석을 위한 여러 방법론이 제시되었으며 각 방법론은 다음과 같다.

- Rumbaugh(럼바우) 방법 : 가장 일반적으로 사용되는 방법으로 분석 활동을 객체 모델, 동적 모델, 기능 모델로 나누어 수행하는 방법이다.
- Booch(부치) 방법 : 미시적(Micro) 개발 프로세스와 거시적(Macro) 개발 프로세스를 모두 사용하는 분석 방법으로, 클래스와 객체들을 분석 및 식별하고 클래스의 속성과 연산을 정의한다.
- Jacobson 방법 : Use Case*를 강조하여 사용하는 분석 방법이다.
- Coad와 Yourdon 방법 : E-R 다이어그램을 사용하여 객체의 행위를 모델링하며, 객체 식별, 구조 식별, 주제 정의, 속성과 인스턴스 연결 정의, 연산과 메시지 연결 정의 등의 과정으로 구성하는 기법이다.
- Wirfs-Brock 방법 : 분석과 설계 간의 구분이 없고, 고객 명세서를 평가해서 설계 작업까지 연속적으로 수행하는 기법이다.

Use Case(사용 사례)
사용자, 외부 시스템, 다른 요소들이 시스템과 상호 작용하는 방법을 기술한 설명

3 럼바우(Rumbaugh)의 분석 기법

럼바우의 분석 기법은 모든 소프트웨어 구성 요소를 그래픽 표기법을 이용하여 모델링하는 기법으로, 객체 모델링 기법(OMT, Object-Modeling Technique)이라고도 한다. 분석 활동은 객체 모델링, 동적 모델링, 기능 모델링을 통해 이루어진다.

- **객체 모델링(Object Modeling)**
  - 정보 모델링이라고도 하며, 시스템에서 요구되는 객체를 찾아내어 속성과 연산 식별 및 객체들 간의 관계를 규정하여 객체 다이어그램*으로 표시하는 것이다.
  - 분석 활동의 세 가지 모델 중 가장 중요하며 선행되어야 할 모델링이다.

객체 다이어그램
소프트웨어를 구성하는 객체와 객체 간의 관계를 표현하는 그래픽 표기법

- 동적 모델링(Dynamic Modeling)
  - 상태 다이어그램*(상태도)을 이용하여 시간의 흐름에 따른 객체들 간의 제어 흐름, 상호 작용, 동작 순서 등의 동적인 행위를 표현하는 모델링이다.
  - 동적 모델링에서는 객체나 클래스의 상태, 사건을 중심으로 다룬다.

사건	하나의 객체로부터 다른 객체에 자극을 주어 객체의 상태를 변화시키는 것
상태	특정 시점의 객체에 대한 속성값

- 기능 모델링(Functional Modeling)
  - 자료 흐름도(DFD)를 이용하여 다수의 프로세스들 간의 자료 흐름을 중심으로 처리 과정을 표현한 모델링이다.
  - 어떤 데이터를 입력하여 어떤 결과를 구할 것인지를 표현하는 것이다.

상태 다이어그램
객체의 상태가 시간에 따라 어떻게 변하는지를 표현하는 그래픽 표기법

예상문제

예상문제은행 병행학습 001

2020년 1 · 2회 통합 정보처리기사 필기 기출문제

1. 객체지향 분석 방법론 중 E-R 다이어그램을 사용하여 객체의 행위를 모델링하며, 객체 식별, 구조 식별, 주체 정의, 속성 및 관계 정의, 서비스 정의 등의 과정으로 구성되는 것은?

① Coad와 Yourdon 방법 ② Booch 방법
③ Jacobson 방법 ④ Wirfs-Brocks 방법

'객체 모형, 동적 모형, 기능 모형' 하면 Rumbaugh 방법, '미시적/거시적 개발 프로세스' 하면 Booch 방법, 'E-R 다이어그램' 하면 Coad와 Yourdon 방법! 꼭 기억해 두세요.

2. 럼바우(Rumbaugh)의 객체지향 분석 절차를 가장 바르게 나열한 것은?

① 객체 모형 → 동적 모형 → 기능 모형
② 객체 모형 → 기능 모형 → 동적 모형
③ 기능 모형 → 동적 모형 → 객체 모형
④ 기능 모형 → 객체 모형 → 동적 모형

럼바우의 객체지향 분석 절차는 앞글자만 따서 '객 · 동 · 기'로 기억해 두세요.

3. 럼바우 분석 기법에서 정보 모델링이라고도 하며, 시스템에서 요구되는 객체를 찾아내어 속성과 연산 식별 및 객체들 간의 관계를 규정하여 객체 다이어그램으로 표시하는 모델링은?

① 동적 모델링 ② 객체 모델링
③ 기능 모델링 ④ 정적 모델링

문제를 잘 보세요. '객체를 찾아 ~ 객체들 간의 관계 ~ 객체 다이어그램!' 이것과 관련된 모델링은 무엇일까요?

4. 객체지향 분석 기법의 하나로 객체 모형, 동적 모형, 기능 모형의 3개 모형을 생성하는 방법은?

① Wirfs-block Method ② Rumbaugh Method
③ Booch Method ④ Jacobson Method

객체지향 분석에서 일반적으로 가장 많이 사용되는 것은 럼바우 기법입니다. 또한 '럼바우 기법' 하면 세 가지 모델을 기억해야 합니다.

5. 럼바우의 분석 기법에서 다음 설명에 해당하는 것은?

- 자료 흐름도를 이용하여 다수의 프로세스들 간의 자료 흐름을 중심으로 처리 과정을 표현한 모델링이다.
- 어떤 데이터를 입력하여 어떤 결과를 구할 것인지를 표현하는 것이다.

① 기능 모델링 ② 동적 모델링
③ 객체 모델링 ④ 정적 모델링

객체 모델링은 '객체', 동적 모델링은 '객체의 흐름, 상태, 행위', 기능 모델링은 '자료 흐름, 처리 과정'과 연결시켜서 기억해 두세요.

6. Rumbaugh의 모델링에서 상태도와 자료 흐름도는 각각 어느 모델링과 관련이 있는가?

① 상태도 – 기능 모델링, 자료 흐름도 – 동적 모델링
② 상태도 – 객체 모델링, 자료 흐름도 – 기능 모델링
③ 상태도 – 객체 모델링, 자료 흐름도 – 동적 모델링
④ 상태도 – 동적 모델링, 자료 흐름도 – 기능 모델링

객체 모델링은 객체도, 동적 모델링은 상태도, 기능 모델링은 자료 흐름도를 사용합니다.

▶ 정답 : 1. ① 2. ① 3. ② 4. ② 5. ① 6. ④

병행학습
002
LIVE

자료 흐름도/자료 사전

전문가의 조언

자료 흐름도의 의미를 파악하고, 자료 흐름도의 네 가지 구성 요소와 표기 방법, 자료 사전에서 사용되는 기호와 의미를 정확히 숙지하세요.

자료의 흐름

자료는 각 절차에 따라 컴퓨터 기반의 시스템 내부를 흘러다니는데, 이를 자료의 흐름이라 합니다.

1 자료 흐름도(DFD)의 개요

자료 흐름도(DFD; Data Flow Diagram)는 요구사항 분석에서 자료의 흐름* 및 변환 과정과 기능을 도형 중심으로 기술하는 방법으로 자료 흐름 그래프, 버블 차트라고도 한다.

- 시스템 안의 프로세스와 자료 저장소 사이에 자료의 흐름을 나타내는 그래프로 자료 흐름과 처리를 중심으로 하는 구조적 분석 기법에 이용된다.
- 자료 흐름도는 자료 흐름과 기능을 자세히 표현하기 위해 단계적으로 세분화된다.
- 자료는 처리(Process)를 거쳐 변환될 때마다 새로운 이름이 부여되며, 처리는 입력 자료가 발생하면 기능을 수행한 후 출력 자료를 산출한다.

2 자료 흐름도 구성 요소 표기법

자료 흐름도에서는 자료의 흐름과 기능을 프로세스(Process), 자료 흐름(Flow), 자료 저장소(Data Store), 단말(Terminator)의 네 가지 기본 기호로 표시한다.

기호	의미	표기법
프로세스 (Process)	• 자료를 변환시키는 시스템의 한 부분(처리 과정)을 나타내며 처리, 기능, 변환, 버블이라고도 한다. • 원이나 둥근 사각형으로 표시하고 그 안에 프로세스 이름을 기입한다.	○
자료 흐름 (Data Flow)	• 자료의 이동(흐름)이나 연관관계를 나타낸다. • 화살표 위에 자료의 이름을 기입한다.	→
자료 저장소 (Data Store)	• 시스템에서의 자료 저장소(파일, 데이터베이스)를 나타낸다. • 도형 안에 자료 저장소 이름을 기입한다.	═
단말 (Terminator)	• 시스템과 교신하는 외부 개체로, 입력 데이터가 만들어지고 출력 데이터를 받는다(정보의 생산자와 소비자). • 도형 안에 이름을 기입한다.	▭

3 자료 사전(DD)

자료 사전(DD; Data Dictionary)은 자료 흐름도에 있는 자료를 더 자세히 정의하고 기록한 것이며, 이처럼 데이터를 설명하는 데이터를 데이터의 데이터 또는 메타 데이터(Meta Data)라고 한다.

- 자료 흐름도에 시각적으로 표시된 자료에 대한 정보를 체계적이고 조직적으로 모아 개발자나 사용자가 편리하게 사용할 수 있다.

자료 사전 표기법 및 작성 시 유의 사항

- 자료 사전의 한 항목은 자료에 대한 정의 부분과 설명 부분으로 구성되며, 정의 부분에는 자료의 이름을, 설명 부분에는 자료에 대한 자세한 내용을 표현한다.
- 이름으로 정의를 쉽게 찾을 수 있어야 하며, 이름이 중복되어서는 안 된다.
- 갱신하기 쉬워야 하며, 정의하는 방식이 명확해야 한다.
- 자료 사전에서 사용되는 표기 기호는 다음과 같다.

기호	의미
=	**자료의 정의** : ~로 구성되어 있다(is composed of)
+	**자료의 연결** : 그리고(and)
( )	**자료의 생략** : 생략 가능한 자료(Optional)
[|]	**자료의 선택** : 또는(or)
{ }	**자료의 반복** : Iteration of ① $\{\ \}_n$: n번 이상 반복 ② $\{\ \}^n$: 최대로 n번 반복 ③ $\{\ \}^n_m$: m 이상 n 이하로 반복
* *	**자료의 설명** : 주석(Comment)

예제

고객파일 = *구성은 주민등록번호에 따라 순차적임*
= {주민등록번호 + 고객명세 + 고객신용 + {입금상황}}
고객명세 = 고객성명 + 고객주소 + 거래 개시일
고객신용 = 신용상태 + 현재잔고
입금상황 = 입금일 + 입금액 + 입금방법
입금방법 = [현금 | 수표 | 신용카드]

2020년 1 · 2회 통합 정보처리기사 필기 기출문제

1. 자료 사전에서 자료의 생략을 의미하는 기호는?

① { }　　② **
③ =　　④ ( )

{ }는 자료의 반복, ( )는 자료의 생략, **는 자료의 설명, =는 자료의 정의입니다.

2. 데이터 흐름도(DFD)의 구성 요소에 포함되지 않는 것은?

① Process　　② Data Flow
③ Data Store　　④ Data Dictionary

데이터 흐름도(DFD)의 구성 요소 네 가지! 프로세스(Process), 자료 흐름(Data Flow), 자료 저장소(Data Store), 단말(Terminator)! 꼭 기억하세요.

3. DFD(Data Flow Diagram)에 대한 설명으로 거리가 먼 것은?

① 자료 흐름 그래프 또는 버블(Bubble) 차트라고도 한다.
② 구조적 분석 기법에 이용된다.
③ 시간 흐름의 개념을 명확하게 표현할 수 있다.
④ DFD의 요소는 화살표, 원, 사각형, 직선(단선/이중선)으로 표시한다.

DFD에서 'D'는 'Data'의 약자로 자료(Data)의 흐름을 표현합니다.

4. 자료 흐름도의 구성 요소와 표시 기호의 연결이 옳지 않은 것은?

① 종착지(Terminator) : 오각형
② 자료 흐름(Data Flow) : 화살표
③ 처리 공정(Process) : 원
④ 자료 저장소(Data Store) : 직선(평행선)

더 이상 설명이 필요 없는 문제죠? 자료 흐름도의 구성 요소 네 가지와 표기법! 정확히 알아두세요.

5. 자료 사전(Data Dictionary)에 사용되는 기호의 의미를 올바르게 나타낸 것으로 짝지어진 것은?

① { } : 자료의 생략 가능, ( ) : 자료의 선택
② ( ) : 자료의 설명, * * : 자료의 선택
③ = : 자료의 설명, * * : 자료의 정의
④ + : 자료의 연결, ( ) : 자료의 생략 가능

자료 사전에서 사용되는 기호와 의미! 아직 잊지 않아죠?

6. 자료 흐름도(DFD)를 작성하는 데 지침이 될 수 없는 항목은?

① 자료 흐름은 처리(Process)를 거쳐 변환될 때마다 새로운 이름을 부여한다.
② 어떤 처리(Process)가 출력 자료를 산출하기 위해서는 반드시 입력 자료가 발생해야 한다.
③ 자료 저장소에 입력 화살표가 있으면 반드시 출력 화살표도 표시되어야 한다.
④ 처리(Process)와 하위 자료 흐름도의 자료 흐름은 서로 일치돼야 한다.

자료 저장소로 입력되는 흐름은 자료의 입력, 수정, 삭제를 나타내는 것으로 입력 화살표가 있다고 하여 반드시 출력 화살표가 표시될 필요는 없습니다.

▶ 정답 : 1. ④　2. ④　3. ③　4. ①　5. ④　6. ③

병행학습
003
LIVE

CASE

1 CASE의 개요

CASE(Computer Aided Software Engineering)는 소프트웨어 개발 과정에서 사용되는 요구 분석, 설계, 구현, 검사 및 디버깅 과정 전체 또는 일부를 컴퓨터와 전용 소프트웨어 도구를 사용하여 자동화하는 것이다.

- 소프트웨어, 하드웨어, 데이터베이스, 테스트 등을 통합하여 소프트웨어를 개발하는 환경을 조성한다.
- 소프트웨어 생명 주기의 전체 단계를 연결해 주고 자동화해 주는 통합된 도구를 제공해 주는 기술이다.
- 소프트웨어 개발 도구와 방법론이 결합된 것으로, 정형화된 구조 및 방법(메커니즘)을 소프트웨어 개발에 적용하여 생산성 향상을 구현하는 공학 기법이다.
- 소프트웨어 개발의 모든 단계에 걸쳐 일관된 방법론을 제공하는 자동화 도구(CASE Tool)*들을 지원하고, 개발자들은 이 도구를 사용하여 소프트웨어 개발의 표준화를 지향하며, 자동화의 이점을 얻을 수 있게 해준다.
- CASE의 주요 기능 : 소프트웨어 생명주기 전 단계의 연결, 다양한 소프트웨어 개발 모형 지원, 그래픽 지원 등

2 CASE 사용의 이점

- 소프트웨어 개발 기간을 단축하고 개발 비용을 절감할 수 있다.
- 자동화된 기법을 통해 소프트웨어 품질이 향상된다.
- 소프트웨어의 유지보수를 간편하게 수행할 수 있다.
- 소프트웨어의 생산성이 향상되고 생산, 운용 활동을 효과적으로 관리 · 통제할 수 있다.
- 품질과 일관성을 효과적으로 제어할 수 있다.
- 소프트웨어 개발의 모든 단계에 걸친 표준을 확립할 수 있다.
- 소프트웨어 모듈의 재사용성이 향상된다.
- 소프트웨어의 개발 기법을 실용화할 수 있고, 문서화를 쉽게 작성할 수 있다.

전문가의 조언

먼저 CASE의 개요와 사용 이점을 숙지하세요. 그리고 CASE는 상위, 하위, 통합 CASE로 분류된다는 것과 각 CASE가 사용되는 소프트웨어 생명 주기가 어느 단계인지 알아두세요.

자동화 도구(CASE Tool)
소프트웨어 공학과 관련된 작업 중에서 하나의 작업을 자동화하는 패키지를 의미합니다.

3 CASE 분류

CASE는 소프트웨어 생명 주기의 어느 부분을 지원하느냐에 따라 다음과 같이 분류할 수 있다.

상위(Upper) CASE

- 소프트웨어 생명 주기의 전반부에서 사용되는 것으로, 문제를 기술(Description)하고 계획하며 요구 분석과 설계 단계를 지원하는 CASE이다.
- 여러 가지 명세와 문서를 작성하는 데 사용된다.
- 상위 CASE 도구(Tool)에는 SREM, PSL/PSA, SERA, FOUNDATION 등이 있다.
- 상위 CASE 도구(Tool)가 지원하는 기능 : 모델들 사이의 모순 검사, 모델의 오류 검증, 자료 흐름도 작성 등

하위(Lower) CASE

- 소프트웨어 생명 주기의 하반부에서 사용되는 것으로 코드의 작성과 테스트, 문서화하는 과정을 지원하는 CASE이다.
- 하위 CASE 도구에는 구문 중심 편집기, 코드 생성기 등이 있다.

통합(Integrate) CASE

- 소프트웨어 생명 주기에 포함되는 전체 과정을 지원하기 위한 CASE로, 공통의 정보 저장 장소와 통일된 사용자 인터페이스를 사용하여 도구들을 통합한다.
- 통합 CASE 도구에는 IEF, POWERTOOLS, TAGS/IORL, TEAMWORK 등이 있다.

예상문제

예상문제은행 병행학습 003

2020년 1 · 2회 통합 정보처리기사 필기 기출문제

1. CASE가 갖고 있는 주요 기능이 아닌 것은?

① 그래픽 지원
② 소프트웨어 생명주기 전 단계의 연결
③ 언어 번역
④ 다양한 소프트웨어 개발 모형 지원

언어 번역은 언어 번역 프로그램의 기능입니다.

2. 소프트웨어 개발 과정에서 사용되는 요구 분석, 설계, 구현, 검사 및 디버깅 과정을 컴퓨터와 전용의 소프트웨어 도구를 사용하여 자동화하는 작업을 무엇이라고 하는가?

① CAT(Computer Aided Testing)
② CAD/CAM(Computer Aided Design and Manufacturing)
③ CASE(Computer Aided Software Engineering)
④ CAI(Computer Aided Instruction)

CASE의 개념을 정확히 숙지하라고 했죠? CASE하면 '자동화'라는 것을 기억하고, 문제를 푸세요.

3. CASE에 대한 설명으로 거리가 먼 것은?

① 소프트웨어 모듈의 재사용성이 향상된다.

② 자동화된 기법을 통해 소프트웨어 품질이 향상된다.

③ 소프트웨어 사용자들이 소프트웨어 사용 방법을 신속히 숙지할 수 있도록 개발된 자동화 패키지이다.

④ 소프트웨어 유지보수를 간편하게 수행할 수 있다.

CASE가 소프트웨어 사용 방법을 자동화하는 것인가요? 생각해 보세요.

4. 소프트웨어, 하드웨어, 데이터베이스, 테스트 등을 통합하여 소프트웨어를 개발하는 환경을 조성한다는 의미를 가진 용어는?

① CAD ② CAI

③ CAM ④ CASE

CASE의 개념을 묻는 문제가 여러 유형으로 출제되고 있죠? 모두 알아두세요.

5. CASE(Computer Aided Software Engineering)에 대한 설명 중 틀린 것은?

① Case는 상위(Upper) CASE, 중위(Medium) CASE, 하위(Lower) CASE, 통합(Integrate) CASE의 4가지 형태로 나눌 수 있다.

② 통합 CASE는 소프트웨어 개발 주기 전체 과정을 지원한다.

③ 상위 CASE는 요구 분석과 설계 단계를 지원한다.

④ 하위 CASE는 코드를 작성하고 테스트하며 문서화하는 과정을 지원한다.

CASE의 세 가지 분류! 상위 CASE, 하위 CASE, 통합 CASE라는 것 기억하고 있어야겠죠?

▶ 정답 : 1. ③ 2. ③ 3. ③ 4. ④ 5. ①

병행학습 004 LIVE

HIPO

전문가의 조언

먼저 HIPO의 개념을 이해하세요. 그리고 HIPO의 세 가지 종류는 서로를 구분할 수 있도록 각각의 특징을 파악해 두세요.

1 HIPO의 개요

HIPO(Hierarchy Input Process Output)는 시스템의 분석 및 설계나 문서화할 때 사용되는 기법으로, 시스템 실행 과정인 입력, 처리, 출력의 기능을 나타낸다.

- 기본 시스템 모델은 입력, 처리, 출력으로 구성되며, 하향식 소프트웨어 개발을 위한 문서화 도구이다.
- 체계적인 문서 관리가 가능하다.
- 기호, 도표 등을 사용하므로 보기 쉽고 이해하기도 쉽다.
- 기능과 자료의 의존 관계를 동시에 표현할 수 있다.
- 변경, 유지보수가 용이하다.

2 HIPO Chart

HIPO Chart는 시스템의 기능을 여러 개의 고유 모듈들로 분할하여 이들 간의 인터페이스를 계층 구조로 표현한 것으로 가시적 도표(Visual Table of Contents), 총체적 도표(Overview Diagram), 세부적 도표(Detail Diagram)가 있다.

- **가시적 도표(도식 목차)** : 시스템의 전체적인 기능과 흐름을 보여주는 계층(Tree) 구조도이다.
- **총체적 도표(총괄도표, 개요 도표)** : 프로그램을 구성하는 기능을 기술한 것으로 입력, 처리, 출력에 대한 전반적인 정보를 제공하는 도표이다.
- **세부적 도표(상세 도표)** : 총체적 도표에 표시된 기능을 구성하는 기본 요소들을 상세히 기술하는 도표이다.

2020년 1 · 2회 통합 정보처리기사 필기 기출문제

1. HIPO(Hierarchy Input Process Output)에 대한 설명으로 거리가 먼 것은?

① 상향식 소프트웨어 개발을 위한 문서화 도구이다.

② HIPO 차트 종류에는 가시적 도표, 총체적 도표, 세부적 도표가 있다.

③ 기능과 자료의 의존 관계를 동시에 표현할 수 있다.

④ 보기 쉽고 이해하기 쉽다.

HIPO는 하향식 소프트웨어 개발을 위한 문서화 도구입니다.

2. HIPO(Hierarchy Input Process Output)에 대한 설명으로 옳지 않은 것은?

① HIPO 다이어그램에는 가시적 도표(Visual Table of Contents), 총체적 다이어그램(Overview Diagram), 세부적 다이어그램(Detail Diagram)의 세 종류가 있다.

② 가시적 도표(Visual Table of Contents)는 시스템에 있는 어떤 특별한 기능을 담당하는 부분의 입력, 처리, 출력에 대한 전반적인 정보를 제공한다.

③ HIPO 다이어그램은 분석 및 설계 도구로서 사용된다.

④ HIPO는 시스템의 설계나 시스템 문서화용으로 사용되고 있는 기법이며, 기본 시스템 모델은 입력, 처리, 출력으로 구성된다.

HIPO에 대한 전반적인 내용에 대해 다루고 있는 문제네요. 하지만 HIPO의 종류만 정확히 구분할 수 있다면 충분히 풀 수 있는 문제입니다. 가시적 도표는 시스템의 전체적인 기능과 흐름을 보여주는 계층(Tree) 구조도이며, ②번의 내용은 총체적 도표를 설명한 것입니다.

3. 프로그램을 구성하는 기능을 기술한 것으로 입력, 처리, 출력을 기술하는 HIPO 패키지에 해당하는 것은?

① Overview Diagram

② Detail Diagram

③ Visual Table of Contents

④ Index Diagram

가시적 도표는 전체적인 기능을 보여주는 것, 총체적 도표는 기능에 대한 입력, 처리, 출력의 전반적인 정보를 제공하는 것, 세부 도표는 총체적 도표의 기본 요소를 상세히 기술하는 것입니다. 세 가지를 구분할 수 있어야 합니다.

4. 시스템의 기능을 여러 개의 고유 모듈들로 분할하여 이들 간의 인터페이스를 계층 구조로 표현한 도형 또는 도면을 무엇이라 하는가?

① Flow Chart

② HIPO Chart

③ Control Specification

④ Box Diagram

HIPO! 더 이상 말 안해도 알겠죠.

▶ 정답 : 1. ① 2. ② 3. ① 4. ②

병행학습
005
● LIVE

제품 소프트웨어 패키징 도구

전문가의 조언

제품 소프트웨어 패키징 도구의 개념과 패키징 도구 활용 시 고려 사항에 대해 확실히 정리하세요.

1 제품 소프트웨어 패키징 도구

제품 소프트웨어 패키징 도구는 제품 소프트웨어의 배포를 위한 패키징 시 디지털 콘텐츠의 지적 재산권의 보호 및 관리 기능과 안전한 유통과 배포를 보장하는 도구이다.

- 제품 소프트웨어 패키징 도구는 불법적인 복제로부터 디지털 콘텐츠의 지적 재산권을 보호해 주는 사용 권한 제어 기술, 패키징 기술, 라이선스 관리, 권한 통제 기술 등을 포함한다.
- 제품 소프트웨어 패키징 도구 활용 시 고려할 사항은 다음과 같다.
  - 패키징 시 사용자에게 배포되는 소프트웨어이므로 내부 콘텐츠에 대한 암호화 및 보안을 고려한다.
  - 다른 여러 콘텐츠 및 단말기 간 DRM(디지털 저작권 관리) 연동을 고려한다.
  - 사용자의 편의성을 위한 복잡성 및 비효율성 문제를 고려한다.
  - 제품 소프트웨어의 종류에 적합한 암호화 알고리즘을 적용한다.

예상문제은행 병행학습 005

2020년 1 · 2회 통합 정보처리기사 필기 기출문제

1. SW 패키징 도구 활용 시 고려 사항과 거리가 먼 것은?

① 패키징 시 사용자에게 배포되는 SW이므로 보안을 고려한다.
② 사용자 편의성을 위한 복잡성 및 비효율성 문제를 고려한다.
③ 보안상 단일 기종에서만 사용할 수 있도록 해야 한다.
④ 제품 SW 종류에 적합한 암호화 알고리즘을 적용한다.

SW 패키징 도구 활용 시 다른 여러 콘텐츠 및 단말기 간 DRM(디지털 저작권 관리) 연동을 고려해야 합니다.

2. 제품 소프트웨어의 배포를 위한 패키징 시 불법적인 복제로부터 디지털 콘텐츠의 지적 재산권을 보호해 주는 기술에 포함되지 않는 것은?

① 사용 권한 제어 기술
② 패키징 기술
③ 라이선스 관리
④ 설치 파일 복원 기술

불법적인 복제로부터 디지털 콘텐츠의 지적 재산권을 보호해 주는 기술에는 사용 권한 제어 기술, 패키징 기술, 라이선스 관리, 권한 통제 기술 등이 포함됩니다.

▶ 정답 : 1. ③ 2. ④

병행학습 006 LIVE

유지보수

1 유지보수의 개요

유지보수(Maintenance)는 개발된 소프트웨어의 품질을 항상 최상의 상태로 유지하기 위한 것으로, 소프트웨어 개발 단계 중 가장 많은 노력과 비용이 투입되는 단계이다.

- 유지보수는 소프트웨어가 사용자에게 인수되어 설치된 후 발생하는 모든 공학적 작업이다.
- 소프트웨어 유지보수를 용이하게 하려면 시험 용이성, 이해성, 수정 용이성, 이식성 등이 고려되어야 한다.
- 유지보수는 수리(Corrective) 보수, 적응(Adaptive) 보수, 완전화(Perfective) 보수, 예방(Preventive) 보수 활동으로 구분할 수 있고, 이 활동을 통해 소프트웨어의 수명을 연장시키는 작업이다.

수정(Corrective) 보수 = 수리 · 교정 · 정정 · 하자 보수	시스템을 운영하면서 검사 단계에서 발견하지 못한 잠재적인 오류를 찾아 수정하는 활동으로, 오류의 수정과 진단을 포함한다.
적응(Adaptive) 보수 = 환경 적응, 조정 보수	• 소프트웨어의 수명 기간 중에 발생하는 환경의 변화(하드웨어, 운영체제 등)를 기존의 소프트웨어에 반영하기 위하여 수행하는 활동이다. • 운영체제나 컴파일러와 같은 프로그래밍 환경의 변화와 주변장치 또는 다른 시스템 요소가 향상되거나 변경될 때 대처할 수 있는 유지보수 활동이다.
완전화(Perfective) 보수 = 기능 개선, 기능 보수	• 소프트웨어의 본래 기능에 새로운 기능을 추가하거나 성능을 개선하기 위해 소프트웨어를 확장시키는 활동이다. • 유지보수 활동 중 가장 큰 업무 및 비용을 차지하는 활동이다.
예방(Preventive) 보수	• 장래의 유지보수성 또는 신뢰성을 개선하거나 소프트웨어의 오류 발생에 대비하여 미리 예방 수단을 강구해 두는 활동이다. • 예방 유지보수를 소프트웨어 재공학*이라고도 한다.

2 유지보수의 부작용

유지보수 활동을 통해 예기치 못한 부작용이 발생될 수 있으며 그 종류는 다음과 같습니다.

- 코딩 부작용 : 코딩 내용의 변경으로 인해 발생하는 부작용이다.
- 자료 부작용 : 자료나 자료 구조의 변경으로 인해 발생하는 부작용이다.
- 문서화 부작용 : 자료 코드에 대한 변경이 설계문서나 사용자가 사용하는 매뉴얼에 적용되지 않을 때에 발생하는 부작용이다.

전문가의 조언

유지보수의 활동(목적)을 중심으로 유지보수에 대한 전반적인 내용을 정리하고, 외계인 코드의 개념을 정확히 숙지하세요.

소프트웨어 재공학(Software Reengineering)

새로운 요구에 맞도록 기존 시스템을 이용하여 보다 나은 시스템을 구축하고, 새로운 기능을 추가하여 소프트웨어 성능을 향상시키는 것입니다.

3 외계인 코드(Alien Code)

외계인 코드는 아주 오래 전에 개발되거나 참고문서 또는 개발자가 없어 유지보수 작업이 매우 어려운 프로그램을 의미한다.

- 일반적으로 15년 전 또는 그 전에 개발된 프로그램을 의미하며, 문서화(Documentation)를 철저하게 해두면 외계인 코드를 방지할 수 있다.

예상문제

예상문제은행 병행학습 006

2020년 1 · 2회 통합 정보처리기사 필기 기출문제

1. 외계인 코드(Alien Code)에 대한 설명으로 옳은 것은?

① 프로그램의 로직이 복잡하여 이해하기 어려운 프로그램을 의미한다.
② 아주 오래되거나 참고문서 또는 개발자가 없어 유지보수 작업이 어려운 프로그램을 의미한다.
③ 오류가 없어 디버깅 과정이 필요 없는 프로그램을 의미한다.
④ 사용자가 직접 작성한 프로그램을 의미한다.

프로그램을 개발한 후 많은 시간이 지나 마치 외계인(Alien)이 작성해 놓은 코드처럼 코드를 읽기 어렵다면 유지보수 작업이 쉬울까요?

2. 소프트웨어 유지보수의 유형에 해당하지 않는 것은?

① 수정 보수(Corrective Maintenance)
② 기능 보수(Functional Maintenance)
③ 완전화 보수(Perfective Maintenance)
④ 예방 보수(Preventive Maintenance)

완전화 보수를 기능 보수라고도 하지만 영문 표현이 잘못되었기 때문에 ②번이 틀린 내용입니다.

3. 유지보수의 종류 중 소프트웨어 수명 기간 중에 발생하는 하드웨어, 운영체제 등 환경의 변화를 기존의 소프트웨어에 반영하기 위하여 수행하는 것은?

① Preventive Maintenance
② Perfective Maintenance
③ Corrective Maintenance
④ Adaptive Maintenance

문제 중에서 핵심이 되는 단어를 찾아야 합니다. 여기서 핵심 단어는 '환경의 변화'가 되겠죠? 환경이 변한다면 환경 변화에 대해 적응을 해야겠죠?

4. 소프트웨어 유지보수 유형 중 현재 수행 중인 기능의 수정, 새로운 기능의 추가, 전반적인 기능 개선 등의 요구를 사용자로부터 받았을 때 수행되는 유형으로서, 유지보수 유형 중 제일 많은 비용이 소요되는 것은?

① Preventive Maintenance
② Adaptive Maintenance
③ Corrective Maintenance
④ Perfective Maintenance

가장 많은 비용과 업무가 소요된다는 것은 그만큼 손이 많이 간다는 의미입니다. 그렇다면 생각해 보세요. 수정하는 것, 적응하는 것, 새로운 기능을 추가하는 것, 예방하는 것! 어떤 것이 손이 많이 갈까요? 용어에서도 답의 향기가 납니다. 당연히 새로운 기능을 추가하는 완전 정비겠죠!

5. 외계인 코드(Alien Code)를 방지하기 위한 방법으로 가장 적합한 것은?

① 프로그램 내에 문서화(Documentation)를 철저하게 해두어야 한다.
② 자료 흐름도(DFD)를 상세히 그려야 한다.
③ 프로그램 완성 시 Testing을 확실하게 해야 한다.
④ 프로그램 시 반드시 Visual tool을 사용해야 한다.

외계인 코드를 방지하기 위한 방법을 알아 두었나요? 다른 거 필요 없습니다. 해당 소프트웨어를 유지보수하기 쉽도록 철저하게 문서화하면 됩니다.

6. 유지보수의 종류 중 소프트웨어 테스팅 동안 밝혀지지 않은 모든 잠재적인 오류를 수정하기 위한 보수 형태로서 오류의 수정과 진단을 포함하는 것은?

① Adaptive Maintenance
② Perfective Maintenance
③ Preventive Maintenance
④ Corrective Maintenance

이 유지보수의 핵심은 '오류 수정'이 되겠죠? 오류를 수정할 수 있는 유지보수를 찾아보세요.

▶ 정답 : 1. ② 2. ② 3. ④ 4. ④ 5. ① 6. ④

병행학습

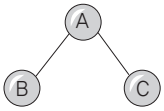
007 이진 트리의 운행법

● LIVE

- 트리를 구성하는 각 노드들을 찾아가는 방법을 운행법(Traversal)이라 한다.
- 이진 트리를 운행하는 방법은 산술식의 표기법과 연관성을 갖는다.

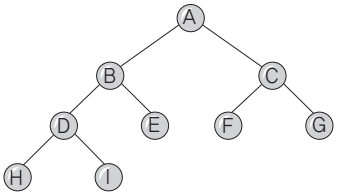
1 트리의 운행법

이진 트리의 운행법*은 다음 세 가지가 있다.

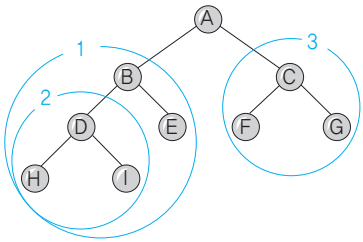


- Preorder 운행 : Root → Left → Right 순으로 운행한다. A, B, C
- Inorder 운행 : Left → Root → Right 순으로 운행한다. B, A, C
- Postorder 운행 : Left → Right → Root 순으로 운행한다. B, C, A

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



Preorder 운행법의 방문 순서



※ 서브트리를 하나의 노드로 생각할 수 있도록 그림과 같이 서브트리 단위로 묶는다. Preorder, Inorder, Postorder 모두 공통으로 사용한다.

❶ Preorder는 Root → Left → Right이므로 A13이 된다.

❷ 1은 B2E이므로 AB2E3이 된다.

❸ 2는 DHI이므로 ABDHIE3이 된다.

❹ 3은 CFG이므로 ABDHIECFG가 된다.

- 방문 순서 : ABDHIECFG

전문가의 조언

예제를 통하여 트리의 운행법을 확실하게 숙지하세요.

이진 트리 운행법

이진 트리 운행법의 이름은 Root의 위치가 어디 있느냐에 따라 정해진 것입니다. 즉 Root가 앞(Pre)에 있으면 Preorder, 안(In)에 있으면 Inorder, 뒤(Post)에 있으면 Postorder입니다.

Inorder 운행법의 방문 순서

❶ Inorder는 Left → Root → Right이므로 1A3이 된다.

❷ 1은 2BE이므로 **2BE**A3이 된다.

❸ 2는 HDI이므로 **HDI**BEA3이 된다.

❹ 3은 FCG이므로 HDIBEA**FCG**가 된다.

- 방문 순서 : HDIBEAFCG

Postorder

❶ Postorder는 Left → Right → Root이므로 13A가 된다.

❷ 1은 2EB이므로 **2EB**3A가 된다.

❸ 2는 HID이므로 **HID**EB3A가 된다.

❹ 3은 FGC이므로 HIDEB**FGC**A가 된다.

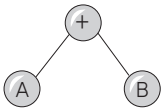
- 방문 순서 : HIDEBFGCA

전문가의 조언

중위식에서 후위식, 전위식 또는 반대의 관계로 표기할 수 있도록 연습하세요.

2 수식의 표기법

산술식을 계산하기 위해 기억공간에 기억시키는 방법으로 이진 트리를 많이 사용한다. 이진 트리로 만들어진 수식을 인오더, 프리오더, 포스트오더로 운행하면 각각 중위(Infix), 전위(Prefix), 후위(Postfix) 표기법이 된다.



- 전위 표기법(PreFix) : 연산자 → Left → Right, +AB
- 중위 표기법(InFix) : Left → 연산자 → Right, A+B
- 후위 표기법(PostFix) : Left → Right → 연산자, AB+

Infix 표기를 Postfix나 Prefix로 바꾸기

- Postfix나 Prefix는 스택을 이용하여 처리하므로 Infix는 Postfix나 Prefix로 바꾸어 처리한다.

예제 1 다음과 같이 Infix로 표기된 수식을 Prefix와 Postfix로 변환하시오.

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

- Prefix로 변환하기

❶ 연산 우선순위에 따라 괄호로 묶는다.

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

❷ 연산자를 해당 괄호의 앞(왼쪽)으로 옮긴다.

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

❸ 필요없는 괄호를 제거한다.

Prefix 표기 : = X + * / A B + C D E

- Postfix로 변환하기

❶ 연산 우선순위에 따라 괄호로 묶는다.

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

❷ 연산자를 해당 괄호의 뒤(오른쪽)로 옮긴다.

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

❸ 필요없는 괄호를 제거한다.

Postfix 표기 : X A B / C D + * E + =

Postfix나 Prefix로 표기된 수식을 Infix로 바꾸기

예제 2 다음과 같이 Postfix로 표기된 수식을 Infix로 변환하시오.

A B C − / D E F + * +

- Postfix는 Infix 표기법에서 연산자를 해당 피연산자 두 개의 뒤로 이동한 것이므로 연산자를 다시 해당 피연산자 두 개의 가운데로 옮기면 된다.

❶ 먼저 인접한 피연산자 두 개와 오른쪽의 연산자를 괄호로 묶는다.

((A (B C −) /) (D (E F +) *) +)

❷ 연산자를 해당 피연산자의 가운데로 이동시킨다.

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

❸ 필요 없는 괄호를 제거한다.

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

예제 3 다음과 같이 Prefix로 표기된 수식을 Infix로 변환하시오.

+ / A − B C * D + E F

- Prefix는 Infix 표기법에서 연산자를 해당 피연산자 두 개의 앞으로 이동한 것이므로 연산자를 다시 해당 피연산자 두 개의 가운데로 옮기면 된다.

❶ 먼저 인접한 피연산자 두 개와 왼쪽의 연산자를 괄호로 묶는다.

(+ (/ A (− B C)) (* D (+ E F)))

❷ 연산자를 해당 피연산자 사이로 이동시킨다.

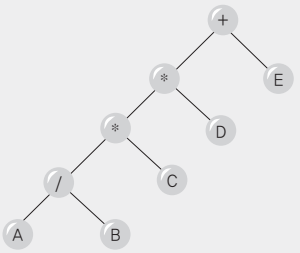
(+ (/ A (− B C)) (* D (+ E F))) → ((A/(B−C)) + (D * (E+F)))

❸ 필요 없는 괄호를 제거한다.

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

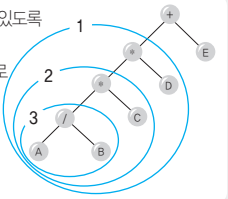
2020년 1 · 2회 통합 정보처리기사 필기 기출문제

1. 다음 트리의 전위 순회(Preorder Traversal)한 결과는?



① + * A B / * C D E
② A B / C * D * E +
③ A / B * C * D + E
④ + * * / A B C D E

먼저 서브트리를 하나의 노드로 생각할 수 있도록 서브트리 단위로 묶습니다.
❶ Preorder는 Root → Left → Right이므로 +1E입니다.
❷ 1은 *2D이므로 +*2DE입니다.
❸ 2는 *3C이므로 +**3CDE입니다.
❹ 3은 /AB이므로 +**/ABCDE입니다.



2. 다음과 같은 중위식(Infix)을 후위식(Postfix)으로 올바르게 표현한 것은?

X = A + (B + C / D) × E − F

① X = A + B + C / D × E − F
② X A B C D / + E × + F − =
③ = X − + A × + B / C D E F
④ X A B C D E F = + + / × −

연산자의 우선순위에 따라 괄호로 묶고 해당 괄호의 뒤로 연산자를 옮깁니다.

3. 다음과 같이 주어진 후위 표기 방식의 수식을 중위 표기 방식으로 나타낸 것은?

ABC − / DEF + * +

① A / (B − C) + F * E + D
② A / (B − C) + D * (E + F)
③ A / (B − C) + D + E * F
④ A / (B − C) * D + E + F

Postfix는 Infix로 표기된 것에서 연산자를 해당 피연산자 두 개의 뒤(오른쪽)로 이동한 것이므로 연산자를 다시 해당 피연산자 두 개의 가운데로 옮기면 됩니다.

4. 다음의 전위(Prefix) 표기식을 중위(Infix) 표기식으로 옳게 변환한 것은?

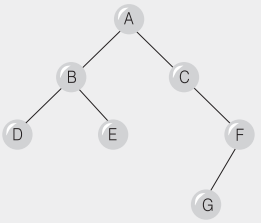
− + * A B C / D E

① B * D + A − E / C
② C * D + B − A / E
③ E * D + C − B / A
④ A * B + C − D / E

전위식으로 표현된 수식을 중위식으로 변환하려면 인접한 피연산자 두 개와 앞(왼쪽)쪽의 연산자를 괄호로 묶은 다음 연산자를 해당 피연산자 사이에 옮겨놓으면 됩니다.

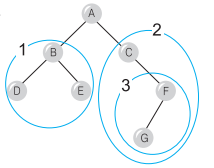
(− (+ (* A B) C) (/ D E))

5. 다음 이진 트리에 대한 Inorder 운행 결과는?



① A, B, D, E, C, F, G
② D, B, E, A, C, G, F
③ D, E, B, G, F, C, A
④ A, B, C, D, E, F, G

먼저 서브트리를 하나의 노드로 생각할 수 있도록 서브트리 단위로 묶습니다.
❶ Inorder는 Left → Root → Right이므로 1A2가 됩니다.
❷ 1은 DBE이므로 DBEA2가 됩니다.
❸ 2는 C3이므로 DBEAC3이 됩니다.
❹ 3은 GF이므로 DBEACGF가 됩니다.



▶ 정답 : 1. ④ 2. ② 3. ② 4. ④ 5. ②

병행학습
008
LIVE

알고리즘의 시간 복잡도/ 빅오 표기법

1 알고리즘

알고리즘은 어떤 문제를 해결하는 논리적인 절차를 의미한다.

- 알고리즘을 해결하는 방법은 다양하며 상황에 따라 적용되는 알고리즘은 모두 다르다.
- 알고리즘의 실행시간은 컴퓨터의 처리 속도, 사용된 프로그래밍 언어의 종류, 컴파일러의 속도에 의해 결정된다.

2 시간 복잡도

시간 복잡도는 알고리즘의 실행시간, 즉 알고리즘을 수행하기 위해 프로세스가 수행하는 연산 횟수를 수치화한 것을 의미한다.

- 시간 복잡도가 낮을수록 알고리즘의 실행시간이 짧고, 높을수록 실행시간이 길어진다.
- 시간 복잡도는 알고리즘의 실행시간이 하드웨어적 성능이나 프로그래밍 언어의 종류에 따라 달라지기 때문에 시간이 아닌 명령어의 실행 횟수를 표기하는데, 이러한 표기법을 점근 표기법이라고 한다.
- 점근 표기법의 종류

빅오 표기법 (Big-O Notation)	• 알고리즘의 실행시간이 최악일 때를 표기하는 방법이다. • 입력값에 대해 알고리즘을 수행했을 때 명령어의 실행 횟수는 어떠한 경우에도 표기 수치보다 많을 수 없다.
세타 표기법 (Big-θ Notation)	• 알고리즘의 실행시간이 평균일 때를 표기하는 방법이다. • 입력값에 대해 알고리즘을 수행했을 때 명령어 실행 횟수의 평균적인 수치를 표기한다.
오메가 표기법 (Big-Ω Notation)	• 알고리즘의 실행시간이 최상일 때를 표기하는 방법이다. • 입력값에 대해 알고리즘을 수행했을 때 명령어의 실행 횟수는 어떠한 경우에도 표기 수치보다 적을 수 없다.

3 빅오 표기법(Big-O Notation)

빅오 표기법은 알고리즘의 실행시간이 최악일 때를 표기하는 방법으로, 신뢰성이 떨어지는 오메가 표기법이나 평가하기 까다로운 세타 표기법에 비해 성능을 예측하기 용이하여 주로 사용된다.

- 일반적인 알고리즘에 대한 최악의 시간 복잡도를 빅오 표기법으로 표현하면 다음과 같다.

전문가의 조언

시간 복잡도를 표현하기 위한 점근 표기법의 종류를 기억하고, 빅오 표기법으로 표현한 알고리즘별 최악의 시간 복잡도는 확실히 기억해 두세요.

O(1)	입력값(n)에 관계 없이 일정하게 문제 해결에 하나의 단계만을 거친다. 예 스택의 삽입(Push), 삭제(Pop)
$O(\log_2 n)$	문제 해결에 필요한 단계가 입력값(n) 또는 조건에 의해 감소한다. 예 이진 트리(Binary Tree), 이진 검색(Binary Search)
O(n)	문제 해결에 필요한 단계가 입력값(n)과 1:1의 관계를 가진다. 예 for문
$O(n\log_2 n)$	문제 해결에 필요한 단계가 $n(\log_2 n)$번만큼 수행된다. 예 힙 정렬(Heap Sort), 2-Way 합병 정렬(Merge Sort)
$O(n^2)$	문제 해결에 필요한 단계가 입력값(n)의 제곱만큼 수행된다. 예 삽입 정렬(Insertion Sort), 쉘 정렬(Shell Sort), 선택 정렬(Selection Sort), 버블 정렬(Bubble Sort), 퀵 정렬(Quick Sort)
$O(2^n)$	문제 해결에 필요한 단계가 2의 입력값(n) 제곱만큼 수행된다. 예 피보나치 수열(Fibonacci Sequence)

예상문제은행

병행학습 008

2020년 1 · 2회 통합 정보처리기사 필기 기출문제

1. 알고리즘 시간 복잡도 O(1)이 의미하는 것은?

① 컴퓨터 처리가 불가

② 알고리즘 입력 데이터 수가 한 개

③ 알고리즘 수행시간이 입력 데이터 수와 관계없이 일정

④ 알고리즘 길이가 입력 데이터보다 작음

O(1)은 입력값(n)에 관계 없이 일정하게 문제 해결에 하나의 단계만을 거친다는 것! 기억해 두세요.

2. 정렬된 N개의 데이터를 처리하는데 $O(N\log_2 N)$의 시간이 소요되는 정렬 알고리즘은?

① 선택 정렬 ② 삽입 정렬

③ 버블 정렬 ④ 합병 정렬

힙 정렬과 2-Way 합병 정렬의 시간 복잡도는 $O(n\log_2 n)$, 나머지 정렬의 시간 복잡도는 $O(n^2)$입니다.

3. 다음 중 알고리즘의 실행시간이 최악일 때 표기하는 방법으로 틀린 것은?

① 일반적인 알고리즘에 대한 최악의 시간 복잡도를 빅오 표기법으로 표현한다.

② 삽입 정렬, 선택 정렬, 버블 정렬 알고리즘의 최악의 시간 복잡도는 $O(n^2)$이다.

③ 힙 정렬, 합병 정렬 알고리즘의 최악의 시간 복잡도는 $O(n\log_2 n)$이다.

④ 피보나치 수열 알고리즘의 최악의 시간 복잡도는 O(n)이다.

알고리즘별 최악의 시간 복잡도를 확실히 암기하라고 했죠? 피보나치 수열 알고리즘의 최악의 시간 복잡도는 $O(2^n)$입니다.

▶ 정답 : 1. ③ 2. ④ 3. ④

병행학습
009
LIVE

병행제어

1 병행제어의 정의

병행제어(Concurrency Control)란 다중 프로그램의 이점*을 활용하여 동시에 여러 개의 트랜잭션을 병행수행할 때, 동시에 실행되는 트랜잭션들이 데이터베이스의 일관성을 파괴하지 않도록 트랜잭션 간의 상호작용을 제어하는 것이다.

2 병행제어의 목적

- 데이터베이스의 공유를 최대화한다.
- 시스템의 활용도를 최대화한다.
- 데이터베이스의 일관성을 유지한다.
- 사용자에 대한 응답시간을 최소화한다.

3 병행수행의 문제점

병행제어(Concurrency Control) 기법에 의한 제어 없이 트랜잭션들이 데이터베이스에 동시에 접근하도록 허용할 경우 다음과 같은 문제점이 발생한다.

문제점	의미
갱신 분실 (Lost Update)	두 개 이상의 트랜잭션이 같은 자료를 공유하여 갱신할 때 갱신 결과의 일부가 없어지는 현상이다.
비완료 의존성 (Uncommitted Dependency)	• 하나의 트랜잭션 수행이 실패한 후 회복되기 전에 다른 트랜잭션이 실패한 갱신 결과를 참조하는 현상이다. • 임시 갱신이라고도 한다.
모순성 (Inconsistency)	• 두 개의 트랜잭션이 병행수행될 때 원치 않는 자료를 이용함으로써 발생하는 문제이다. • 불일치 분석(Inconsistent Analysis)이라고도 한다.
연쇄 복귀 (Cascading Rollback)	병행수행되던 트랜잭션들 중 어느 하나에 문제가 생겨 Rollback하는 경우 다른 트랜잭션도 함께 Rollback되는 현상이다.

4 로킹(Locking)

로킹은 병행제어 기법 중 하나로 주요 데이터의 액세스를 상호 배타적으로 하는 것이다.

- 트랜잭션들이 어떤 로킹 단위를 액세스하기 전에 Lock(잠금)을 요청해서 Lock이 허락되어야만 그 로킹 단위를 액세스할 수 있도록 하는 기법이다.

전문가의 조언

병행제어의 정의, 목적, 문제점을 정리하고, 로킹 단위의 특징을 정확히 기억해 두세요.

다중 프로그래밍의 이점

- 프로세서의 이용률이 증가한다.
- 전체 트랜잭션의 작업 처리율이 향상된다.

병행성 수준
병행성 수준이 낮다는 것은 데이터베이스 공유도가 감소한다는 의미이고, 병행성 수준이 높다는 것은 데이터베이스 공유도가 증가한다는 의미입니다.

로킹 단위(Locking Granularity)

- 로킹 단위는 병행제어에서 한꺼번에 로킹할 수 있는 객체의 크기를 의미한다.
- 데이터베이스, 파일, 레코드, 필드 등은 로킹 단위가 될 수 있다.
- 로킹 단위가 크면 로크 수가 작아 관리하기 쉽지만 병행성 수준이 낮아지고 데이터베이스 공유도가 저하된다.
- 로킹 단위가 작으면 로크 수가 많아 관리하기 복잡해 오버헤드가 증가하지만 병행성 수준*이 높아지고, 데이터베이스 공유도가 증가한다.

예상문제은행 병행학습 009

2020년 1 · 2회 통합 정보처리기사 필기 기출문제

1. 병행제어의 로킹(Locking) 단위에 대한 설명으로 옳지 않은 것은?

① 데이터베이스, 파일, 레코드 등은 로킹 단위가 될 수 있다.
② 로킹 단위가 작아지면 로킹 오버헤드가 감소한다.
③ 로킹 단위가 작아지면 데이터베이스 공유도가 증가한다.
④ 한꺼번에 로킹 할 수 있는 개체의 크기를 로킹 단위라고 한다.

이런 문제는 상식적으로 생각해도 쉽게 맞힐 수 있는 문제입니다. 로킹 단위가 크면 로크 수가 적어 관리하기 쉽지만 동시에 수행하는 작업 수도 적어지므로 병행성 수준이 낮아지고, 로킹 단위가 작으면 로크 수가 많아 관리하기 복잡하지만 동시에 수행하는 작업 수도 늘어나므로 병행성 수준이 높아지겠죠.

2. 병행제어(Concurrency Control) 기법에 의한 제어 없이 트랜잭션들이 데이터베이스를 동시에 접근하도록 허용할 경우 발생하는 문제점이 아닌 것은?

① 갱신 분실(Lost Update)
② 교착상태(Deadlock)
③ 모순성(Inconsistency)
④ 연쇄 복귀(Cascading Rollback)

교착상태는 병행제어를 위해 Lock한 자원을 서로 점유하기 위해 무한정 기다리고 있는 상태를 말합니다.

3. 트랜잭션의 병행제어 목적이 아닌 것은?

① 데이터베이스의 공유 최대화
② 시스템의 활용도 최대화
③ 데이터베이스의 일관성 최소화
④ 사용자에 대한 응답시간 최소화

데이터베이스의 일관성 최소화가 아니고 일관성 유지겠죠.

4. 병행처리를 수행할 때 로킹 방법에 의해 해결될 수 있는 것이 아닌 것은?

① 갱신 분실 문제 ② 비완료 의존성 문제
③ 불일치 문제 ④ 교착상태 문제

로킹 방법으로 트랜잭션들이 데이터베이스에 동시에 접근하도록 허용할 경우 발생하는 문제는 해결할 수 있습니다.

5. 병행 제어 기법을 적용하지 않을 경우 문제점 중 하나의 트랜잭션 수행이 실패한 후 회복되기 전에 다른 트랜잭션이 실패한 갱신 결과를 참조하는 현상은?

① Lost Update
② Inconsistency
③ Cascading Rollback
④ Uncommitted Dependency

비완료 의존성이란 말 그대로 무엇이 완료되기 전에 의존한다는 뜻이란 것을 염두에 두고 생각해 보세요.

▶ 정답 : 1. ② 2. ② 3. ③ 4. ④ 5. ④

병행학습

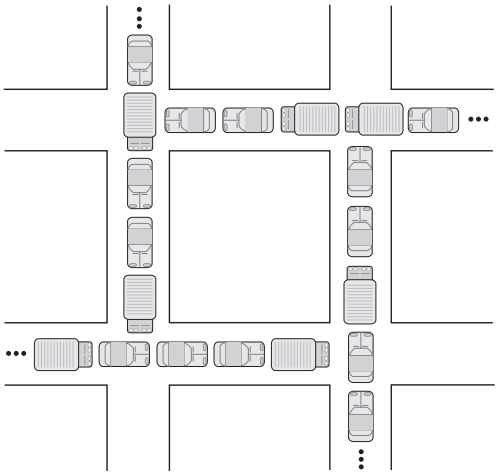
010 교착상태

LIVE

1 교착상태의 개요

교착상태(Dead Lock)*는 상호 배제에 의해 나타나는 문제점으로, 둘 이상의 프로세스들이 자원을 점유한 상태에서 서로 다른 프로세스가 점유하고 있는 자원을 요구하며 무한정 기다리는 현상을 의미한다.

- 아래 그림과 같이 자동차(프로세스)들이 현재 위치한 길(자원)을 점유함과 동시에 다른 차가 사용하는 길을 사용하려고 대기하고 있지만 다른 길을 사용할 수 없으며 현재 길에서도 벗어나지 못하는 상태이다.



교통의 교착상태

2 교착상태 발생의 필요 충분 조건

교착상태가 발생하기 위해서는 다음의 네 가지 조건이 충족되어야 하는데, 이 네 가지 조건 중 하나라도 충족되지 않으면 교착상태가 발생하지 않는다.

상호 배제(Mutual Exclusion)	한 번에 한 개의 프로세스만이 공유 자원을 사용할 수 있어야 한다.
점유와 대기(Hold and Wait)	최소한 하나의 자원을 점유하고 있으면서 다른 프로세스에 할당되어 사용되고 있는 자원을 추가로 점유하기 위해 대기하는 프로세스가 있어야 한다.
비선점(Non-preemption)	다른 프로세스에 할당된 자원은 사용이 끝날 때까지 강제로 빼앗을 수 없어야 한다.
환형 대기(Circular Wait)	공유 자원과 공유 자원을 사용하기 위해 대기하는 프로세스들이 원형으로 구성되어 있어 자신에게 할당된 자원을 점유하면서 앞이나 뒤에 있는 프로세스의 자원을 요구해야 한다.

전문가의 조언

먼저 교착상태의 개념을 숙지하세요. 그리고 교착상태가 발생하기 위한 필요 충분 조건과 교착상태 해결 기법은 각각의 특징을 구분할 수 있도록 정리하세요.

교착상태와 무한 연기

교착상태와 무한 연기는 무한정 기다리는 현상입니다. 하지만 전혀 가능성이 없는 상태에서 기다리는 교착상태와 달리, 무한 연기는 그래도 가능성이 있는 상태에서 기다리는 것을 의미합니다.

3 예방 기법(Prevention)

교착상태 예방 기법은 교착상태가 발생하지 않도록 사전에 시스템을 제어하는 방법으로, 교착상태 발생의 네 가지 조건 중에서 어느 하나를 제거(부정)함으로써 수행된다. 자원의 낭비가 가장 심한 기법이다.

- **상호 배제(Mutual Exclusion) 부정** : 한 번에 여러 개의 프로세스가 공유 자원을 사용할 수 있도록 한다.*
- **점유 및 대기(Hold and Wait) 부정** : 프로세스가 실행되기 전 필요한 모든 자원을 할당하여 프로세스 대기를 없애거나 자원이 점유되지 않은 상태에서만 자원을 요구하도록 한다.
- **비선점(Non-preemption) 부정** : 자원을 점유하고 있는 프로세스가 다른 자원을 요구할 때 점유하고 있는 자원을 반납하고, 요구한 자원을 사용하기 위해 기다리게 한다.
- **환형 대기(Circular Wait) 부정** : 자원을 선형 순서로 분류하여 고유 번호를 할당하고, 각 프로세스는 현재 점유한 자원의 고유 번호보다 앞이나 뒤 어느 한쪽 방향으로만 자원을 요구하도록 하는 것이다.

상호 배제(Mutual Exclusion)의 제거
공유 자원은 어느 한 시점에서 하나의 프로세스만 사용할 수 있어야 하므로, 상호 배제 부정을 수행하는 것은 올바르지 않고, 실제로도 구현하지 않습니다.

4 회피 기법(Avoidance)

교착상태 회피 기법은 교착상태가 발생할 가능성을 배제하지 않고 교착상태가 발생하면 적절히 피해나가는 방법으로, 주로 은행원 알고리즘(Banker's Algorithm)이 사용 된다.

은행원 알고리즘(Banker's Algorithm)

- 은행원 알고리즘은 E. J. Dijkstra가 제안한 것으로, 은행에서 모든 고객의 요구가 충족되도록 현금을 할당하는 데서 유래한 기법이다.
- 각 프로세스에게 자원을 할당하여 교착상태가 발생하지 않으며 모든 프로세스가 완료될 수 있는 상태를 안전 상태, 교착상태가 발생할 수 있는 상태를 불안전 상태*라고 한다.

불안전 상태
교착상태는 불안전 상태이지만 불안전 상태라고 해서 모두 교착상태인 것은 아닙니다.

5 발견 기법(Detection)

교착상태 발견 기법은 시스템에 교착상태가 발생했는지 점검하여 교착상태에 있는 프로세스와 자원을 발견하는 것을 의미한다.

- 교착상태 발견 알고리즘과 자원 할당 그래프 등을 사용할 수 있다.

6 회복 기법(Recovery)

교착상태 회복 기법은 교착상태를 일으킨 프로세스를 종료하거나 교착상태의 프로세스에 할당된 자원을 선점*하여 프로세스나 자원을 회복하는 것을 의미한다.

자원 선점 시 고려 사항
- **자원을 선점할 프로세스 선택 문제** : 최소의 피해를 줄 수 있는 프로세스를 선택합니다.
- **자원을 선점한 프로세스의 복귀 문제** : 자원이 부족한 상태이므로 대부분 일시 중지시키고 다시 시작하는 방법을 사용합니다.
- **기아 현상 문제** : 한 프로세스가 계속하여 자원 선점 대상이 되지 못하도록 고려해야 합니다.

프로세스 종료	교착상태에 있는 프로세스를 종료하는 것으로, 교착상태에 있는 모든 프로세스를 종료하는 방법과 교착상태에 있는 프로세스들을 하나씩 종료해 가며 교착상태를 해결하는 방법이 있다.
자원 선점	• 교착상태의 프로세스가 점유하고 있는 자원을 선점하여 다른 프로세스에게 할당하며, 해당 프로세스를 일시 정지시키는 방법이다. • 우선순위가 낮은 프로세스, 수행된 정도가 적은 프로세스, 사용되는 자원이 적은 프로세스 등을 위주로 해당 프로세스의 자원을 선점한다.

예상문제

예상문제은행

병행학습 010

2020년 1 · 2회 통합 정보처리기사 필기 기출문제

1. 은행가 알고리즘(Banker's Algorithm)은 교착상태의 해결 방법 중 어떤 기법에 해당하는가?

① Avoidance ② Detection
③ Prevention ④ Recovery

'은행원 알고리즘'! 하면, 교착상태 회피(Avoidance) 기법이란것, 잊지마세요!

2. 교착 상태 발생의 필요 충분 조건이 아닌 것은?

① 상호 배제(mutual exclusion)
② 점유와 대기(hold and wait)
③ 환형 대기(circular wait)
④ 선점(preemption)

교착상태 발생 조건 4가지 '상호 배제, 점유와 대기, 비선점, 환형 대기'! 잊으면 안됩니다!

3. 교착상태의 해결 방법 중 점유 및 대기 조건 방지, 비선점 조건 방지, 환형 대기 조건 방지와 가장 밀접한 관계가 있는 것은?

① Prevention ② Avoidance
③ Detection ④ Recovery

'방지'란 어떤 현상이 발생하지 않도록 **예방**하는 것임을 염두에 두고 답을 찾아보세요.

4. 교착상태(Deadlock)에 관한 설명으로 틀린 것은?

① 교착상태 발생의 필요 충분 조건은 상호 배제, 점유 및 대기, 환형 대기, 비선점 조건이다.
② 교착상태란 두 개 이상의 프로세스들이 자원을 점유한 상태에서 서로 다른 프로세스가 점유하고 있는 자원을 동시에 사용할 수 있는 현상을 의미한다.
③ 교착상태의 회피(Avoidance)는 교착상태에 빠질 가능성을 인정하고 적절히 이를 피해 가는 방법이다.
④ 교착상태의 회복(Recovery)은 교착상태에 빠져 있는 프로세스를 중지시켜 시스템이 정상적으로 동작할 수 있도록 하는 방법이다.

보기를 잘 읽으면 답을 찾을 수 있는 문제입니다. 교착상태는 두 개 이상의 프로세스들이 자원을 점유한 상태에서 서로 다른 프로세스가 점유하고 있는 자원을 동시에 사용할 수 있는 것이 아니라 동시에 사용할 수 없는 현상입니다.

5. 교착상태를 회피하는 데 사용하는 은행원 알고리즘에 대한 설명으로 옳지 않은 것은?

① 불안전 상태와 안전 상태로 구분한다.
② 안전 상태에서는 교착상태가 발생하지 않는다.
③ 불안전 상태와 교착상태는 동일한 의미이다.
④ Dijkstra가 제안한 기법이다.

은행원 알고리즘을 다시 한 번 살펴 볼까요? 은행이 보유한 현금이 적을 경우로, 잘 대출해 주면 안전 상태를 유지할 수 있지만 잘못 대출해 줄 경우 불안전 상태가 되며 불안전 상태에서 교착상태가 발생할 수 있습니다. 즉 불안전 상태는 교착상태가 될 가능성은 있지만 불안전 상태가 무조건 교착상태인 것은 아닙니다.

▶ 정답 : 1. ① 2. ④ 3. ① 4. ② 5. ③

병행학습 011 ● LIVE

LAN(근거리 통신망)

전문가의 조언

LAN의 개념과 특징을 파악하고, LAN의 표준안이 OSI 참조 모델의 어느 계층에 대응되는지와 각 계층의 기능을 정리하세요.

1 LAN의 개요

LAN(근거리 통신망, Local Area Network)은 광대역 통신망과는 달리 학교, 회사, 연구소 등 한 건물이나 일정 지역 내에서 컴퓨터나 단말기들을 고속 전송 회선으로 연결하여 프로그램 파일 또는 주변장치를 공유할 수 있도록 한 네트워크 형태이다.

- 단일 기관의 소유, 제한된 지역 내의 통신이다.
- 광대역 전송 매체의 사용으로 고속 통신이 가능하다.
- 공유 매체를 사용하므로 경로 선택 없이 매체에 연결된 모든 장치로 데이터를 전송한다.
- 오류 발생률이 낮다.
- 네트워크에 포함된 자원을 공유한다.
- 네트워크의 확장이나 재배치가 쉽다.
- 전송 매체로 꼬임선, 동축 케이블, 광섬유 케이블 등을 사용한다.
- 망의 구성 형태에 따라서 성형, 버스형, 링형, 계층형으로 분류할 수 있다.

2 LAN의 표준안

LAN의 표준안은 IEEE* 802 위원회에 의해 추진되었으며, OSI 참조 모델의 하위 두 개 계층인 물리 계층과 데이터 링크 계층을 대상으로 한다.

IEEE
IEEE(전기전자기술자협회)는 전기전자공학의 연구를 촉진하기 위해 설립되었으며, ANSI에 의해 미국 국가표준을 개발하도록 인증받은 전문 기구입니다.

LAN의 계층 구조

IEEE 802 위원회에서 지정한 LAN의 계층 구조는 다음과 같다.

계층	기능
물리 계층	OSI 7 계층의 물리 계층과 동일한 기능을 제공한다.
데이터 링크 계층	• OSI 7계층의 데이터 링크 계층에 해당하는 일을 담당하며, 매체 접근 제어(MAC, Media Access Control) 계층과 논리 링크 제어(LLC, Logical Link Control) 계층으로 분류된다. • 매체 접근 제어(MAC) 계층 : 물리 네트워크(공유 매체)에 대한 접근 제어 담당 • 논리 링크 제어(LLC) 계층 : 매체 접근 제어(MAC)에 의해 확보된 채널을 사용하여 메시지를 전송할 때 필요한 연결 관리 절차와 ACK, NAK, 순서화, 오류 제어, 흐름 제어 담당

IEEE 802의 주요 표준 규격

IEEE 802 위원회에서 지정한 LAN의 표준 규격은 다음과 같다.

표준 규격	내용
802.1	전체의 구성, OSI 참조 모델과의 관계, 통신망 관리 등에 관한 규약이다.
802.2	논리 링크 제어(LLC) 계층에 관한 규약이다.
802.3	CSMA/CD 방식*의 매체 접근 제어 계층에 관한 규약이다.
802.4	토큰 버스 방식*의 매체 접근 제어 계층에 관한 규약이다.
802.5	토큰 링 방식*의 매체 접근 제어 계층에 관한 규약이다.
802.6	도시형 통신망(MAN)에 관한 규약이다.
802.9	종합 음성/데이터 네트워크에 관한 규약이다.
802.11	무선 LAN에 관한 규약이다.

CSMA/CD 방식, 토큰 버스 방식, 토큰 링 방식은 LAN의 매체 접근 제어(MAC) 방식입니다.

잠깐만요

802.11의 버전

802.11 (초기 버전)	2.4GHz 대역 전파와 CSMA/CA 기술을 사용해 최고 2Mbps까지의 전송 속도를 지원합니다.
802.11a	5GHz 대역의 전파를 사용하며, OFDM 기술을 사용해 최고 54Mbps까지의 전송 속도를 지원합니다.
802.11b	802.11 초기 버전의 개선안으로 등장하였으며, 초기 버전의 대역 전파와 기술을 사용해 최고 11Mbps의 전송 속도로 기존에 비해 5배 이상 빠르게 개선되었습니다.
802.11e	802.11의 부가 기능 표준으로, QoS 기능을 지원하기 위해 매체 접근 제어(MAC) 계층*에 해당하는 부분을 수정하였습니다.
802.11g	2.4GHz 대역의 전파를 사용하지만 5GHz 대역의 전파를 사용하는 802.11a와 동일한 최고 54Mbps까지의 전송 속도를 지원합니다.
802.11n	2.4GHz 대역과 5GHz 대역을 사용하는 규격으로, 최고 600Mbps까지의 전송 속도를 지원합니다.

매체 접근 제어(MAC)
매체 접근 제어(MAC)는 OSI 참조 모델의 데이터 링크 계층의 일부로, 동일한 매체를 여러 단말들이 공유하는 LAN에서 단말 간 충돌 및 경합을 제어하는 역할을 수행합니다.

2020년 1 · 2회 통합 정보처리기사 필기 기출문제

1. IEEE 802.11 워킹 그룹의 무선 LAN 표준화 현황 중 QoS 강화를 위해 MAC 지원 기능을 채택한 것은?

① 802.11a ② 802.11b
③ 802.11g ④ 802.11e

'QoS 강화를 위한 MAC 지원'하면 802.11e라는 것을 기억해 두세요.

2. 다음의 설명 내용에 해당되는 것은?

- 제한된 지역 내의 통신
- 파일의 공용
- 공중망을 이용하는 광역 통신망에 대조되는 통신망
- 소단위의 고속 정보 통신망

① 종합정보 통신망(ISDN)
② 부가가치 통신망(VAN)
③ 근거리 통신망(LAN)
④ 가입 전산망(Teletex)

시험과 관련된 통신망 중에서 가장 중요한 통신망입니다. 꼭 기억해둬야겠죠?

3. LAN의 통신 구조가 가능한 한 호환성을 유지해야 한다는 필요성에 의해 OSI 참조 모델의 데이터링크 계층을 2개의 서브 계층으로 나누었는데 이에 해당하는 것은?

① 상위 서브 계층 : LLC, 하위 서브 계층 : LAPB
② 상위 서브 계층 : HDLC, 하위 서브 계층 : MAC
③ 상위 서브 계층 : HDLC, 하위 서브 계층 : LAPB
④ 상위 서브 계층 : LLC, 하위 서브 계층 : MAC

MAC은 매체 접근 제어, LLC는 논리 링크 제어인데, 정보 통신에서는 통신 매체와 같은 물리적인 대상을 관리하는 계층이 주로 하위 계층입니다.

4. 다음의 네트워크 표준들 중에 정보 전송에 있어 토큰 링 방식을 사용하는 것은?

① X.25 ② Frame relay
③ IEEE 802.5 ④ Ethernet

토큰 링 방식의 LAN은 IEEE 802.5에 정의되어 있습니다. 토큰 링 방식 외에도 이더넷, 무선 LAN의 표준안도 중요하니 LAN의 표준 규격(IEEE 802)을 꼭 기억하세요.

5. 무선 LAN, Wi-Fi(Wireless Fidelity)의 표준(규격) 제정을 담당하는 IEEE 워킹 그룹은?

① IEEE 802.3 ② IEEE 802.9
③ IEEE 802.10 ④ IEEE 802.11

802.3은 CSMA/CD, 802.5는 토큰 링, 802.9는 종합 음성/데이터 네트워크 방식에 관한 규약입니다.

6. IEEE 802.3 LAN에서 사용되는 전송매체 접속제어(MAC) 방식은?

① CSMA/CD ② Token Bus
③ Token Ring ④ Slotted Ring

IEEE 802의 주요 표준 규격은 모두 기억하고 있어야 합니다.

▶ 정답 : 1. ④ 2. ③ 3. ④ 4. ③ 5. ④ 6. ①

병행학습
012
LIVE

비선점 스케줄링

1 비선점 스케줄링의 개요

비선점 스케줄링에는 FCFS, SJF, HRN, 우선순위, 기한부 알고리즘이 있다.

잠깐만요

비선점 스케줄링과 선점 스케줄링

다음과 같이 대응되는 표를 이용하여 각 알고리즘이 선점 기법인지 비선점 기법인지 다시 정리한 후 학습 하세요.

비선점 스케줄링	FCFS	SJF	우선순위	HRN	기한부
선점 스케줄링	RR	SRT	선점 우선순위	다단계 큐	다단계 피드백 큐

2 FCFS(First Come First Service, 선입 선출) = FIFO(First In First Out)

- FCFS는 준비상태 큐(대기 큐, 준비 완료 리스트, 작업준비 큐, 스케줄링 큐)에 도착한 순서에 따라 차례로 CPU를 할당하는 기법으로, 가장 간단한 알고리즘이다.
- 먼저 도착한 것이 먼저 처리되어 공평성은 유지되지만 짧은 작업이 긴 작업을, 중요한 작업이 중요하지 않은 작업을 기다리게 된다.

예제 다음과 같은 프로세스들이 차례로 준비상태 큐에 들어왔다고 가정할 때, FCFS 기법을 이용하여 평균 실행 시간, 평균 대기 시간, 평균 반환 시간을 구하시오(제출시간은 없으며 시간의 단위는 초임).

프로세스 번호	P1	P2	P3
실행 시간	20	4	6

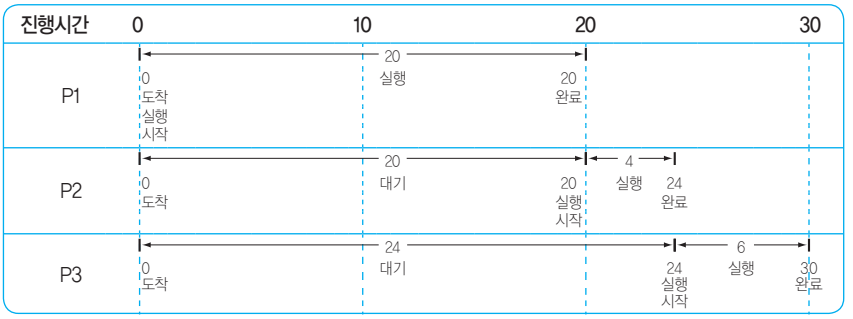
① 실행 시간을 이용하여 다음과 같이 각 프로세스의 대기 시간과 반환 시간을 구한다.

- 대기 시간 : 프로세스가 대기한 시간으로, 바로 앞 프로세스까지의 진행 시간으로 계산
- 반환 시간 : 프로세스의 대기 시간과 실행 시간의 합

② 실행 시간, 대기 시간, 반환 시간의 평균은 '각 프로세스 시간의 합/프로세스의 개수'를 이용한다.

전문가의 조언

비선점 기법에 해당하는 각 알고리즘의 종류와 각각의 의미를 정확히 이해해야 합니다. 각 알고리즘의 영문 약어는 개념을 이해하는 데 도움이 되므로 꼭 기억하세요.



- 평균 실행 시간 : (20+4+6)/3 = 10
- 평균 대기 기간 : (0+20+24)/3 = 14.6
- 평균 반환 시간 : (20+24+30)/3 = 24.6

3 SJF(Shortest Job First, 단기 작업 우선)

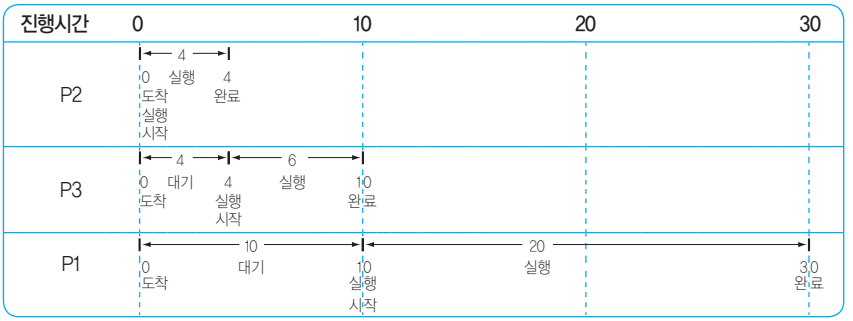
- SJF는 준비상태 큐에서 기다리고 있는 프로세스들 중에서 실행 시간이 가장 짧은 프로세스에게 먼저 CPU를 할당하는 기법이다.
- 가장 적은 평균 대기 시간을 제공하는 최적 알고리즘이다.
- 실행 시간이 긴 프로세스는 실행 시간이 짧은 프로세스에게 할당 순위가 밀려 무한 연기 상태가 발생될 수 있다.

예제 1 다음과 같은 프로세스들이 차례로 준비상태 큐에 들어왔다고 가정할 때, SJF 기법을 이용하여 평균 실행 시간, 평균 대기 시간, 평균 반환 시간을 구하시오(제출 시간이 없을 경우).

프로세스 번호	P1	P2	P3
실행 시간	20	4	6

① 아래와 같이 실행 시간이 짧은 프로세스를 먼저 처리하도록 이동시킨 후 각 프로세스의 대기 시간과 반환 시간을 구한다.

② 실행시간, 대기 시간, 반환 시간, 각 시간의 평균은 FCFS의 예제와 동일한 방법으로 구한다.



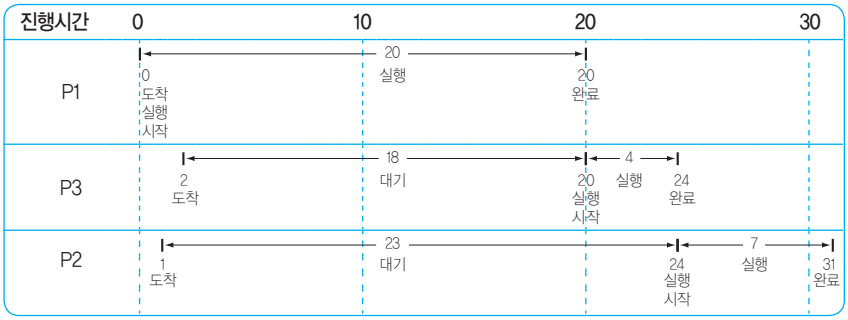
- 평균 실행 시간 : (4+6+20)/3 = 10
- 평균 대기 기간 : (0+4+10)/3 = 4.6
- 평균 반환 시간 : (4+10+30)/3 = 14.6

예제 2 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
제출 시간	0	1	2

① 가장 먼저 도착한 P1을 실행한 후 요구된 실행 시간이 적은 P3, P2 순으로 수행한다.

② 대기 시간은 현재 프로세스가 수행되기 전까지의 진행 시간에서 제출 시간을 차감하고, 반환 시간은 실행 시간과 대기 시간의 합으로 구한다.



- 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

4 HRN(Highest Response-ratio Next)

- 실행 시간이 긴 프로세스에 불리한 SJF 기법을 보완하기 위한 것으로, 대기 시간과 서비스(실행) 시간을 이용하는 기법이다.
- 우선순위 계산 공식을 이용하여 서비스(실행) 시간이 짧은 프로세스나 대기 시간이 긴 프로세스에게 우선순위를 주어 CPU를 할당한다.
- 서비스 실행 시간이 짧거나 대기 시간이 긴 프로세스일 경우 우선순위가 높아진다.
- 우선순위를 계산하여 그 숫자가 가장 높은 것부터 낮은 순으로 우선순위가 부여된다.
- 우선순위 계산식

$$\text{우선순위 계산식} = \frac{\text{대기 시간+서비스 시간}}{\text{서비스 시간}}$$

예제 다음과 같은 프로세스가 HRN 기법으로 스케줄링될 때 우선순위를 계산하시오.

프로세스 번호	P1	P2	P3
실행 시간	20	4	6
대기 시간	10	20	10
우선순위 계산	(20+10)/20=1.5	(4+20)/4=6	(6+10)/6=2.6
우선순위	P2 → P3 → P1		

5 기한부(Deadline)

- 프로세스에게 일정한 시간을 주어 그 시간 안에 프로세스를 완료하도록 하는 기법이다.
- 프로세스가 제한된 시간 안에 완료되지 않을 경우 제거되거나 처음부터 다시 실행해야 한다.
- 시스템은 프로세스에게 할당할 정확한 시간을 추정해야 하며, 이를 위해서 사용자는 시스템이 요구한 프로세스에 대해 정확한 정보를 제공해야 한다.
- 여러 프로세스들이 동시에 실행되면 스케줄링이 복잡해지며, 프로세스 실행 시 집중적으로 요구되는 자원 관리에 오버헤드가 발생한다.

6 우선순위(Priority)

- 준비상태 큐에서 기다리는 각 프로세스마다 우선순위를 부여하여 그 중 가장 높은 프로세스에게 먼저 CPU를 할당하는 기법이다.
- 우선순위가 동일할 경우 FCFS 기법으로 CPU를 할당한다.
- 우선순위는 프로세스의 종류나 특성에 따라 다르게 부여될 수 있다.
- 가장 낮은 순위를 부여받은 프로세스는 무한 연기* 또는 기아 상태(Starvation)*가 발생할 수 있다.

무한 연기 / 기아 상태
우선순위가 낮아 CPU 할당이 무한히 연기되는 상태를 무한 연기라 하고, 무한 연기 상태에서 결국 프로세스를 완료하지 못하는 상태를 기아 상태라 합니다.

잠깐만요

에이징(Aging) 기법

- 시스템에서 특정 프로세스의 우선순위가 낮아 무한정 기다리게 되는 경우, 한 번 양보하거나 기다린 시간에 비례하여 일정 시간이 지나면 우선순위를 한 단계씩 높여 가까운 시간 안에 자원을 할당받도록 하는 기법입니다.
- SJF나 우선순위 기법에서 발생할 수 있는 무한 연기 상태, 기아 상태를 예방할 수 있습니다.

2020년 1 · 2회 통합 정보처리기사 필기 기출문제

1. HRN(Highest Response-ratio Next) 스케줄링 방식에 대한 설명으로 옳지 않은 것은?

① 대기 시간이 긴 프로세스일 경우 우선순위가 높아진다.
② SJF 기법을 보완하기 위한 방식이다.
③ 긴 작업과 짧은 작업 간의 지나친 불평등을 해소할 수 있다.
④ 우선순위를 계산하여 그 수치가 가장 낮은 것부터 높은 순으로 우선순위가 부여된다.

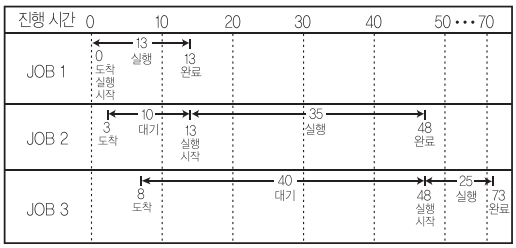
HRN 스케줄링 방식은 우선순위를 계산하여 그 수치가 높을수록 높은 우선순위가 부여됩니다.

2. FIFO 스케줄링에서 3개의 작업 도착 시간과 CPU 사용 시간이 다음 표와 같다. 이 때 모든 작업들의 평균 반환 시간은?

작업	도착 시간	CPU 사용 시간
JOB 1	0	13
JOB 2	3	35
JOB 3	8	25

① 16
② 20
③ 33
④ 41

FIFO 스케줄링은 준비상태 큐에 도착한 순서에 따라 CPU를 할당하는 기법입니다.



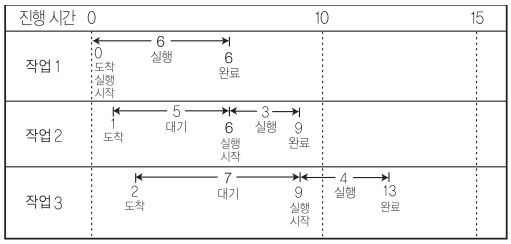
반환 시간은 '대기 시간+실행 시간'이므로, 'JOB 1'의 반환 시간은 13(0+13), 'JOB 2'의 반환 시간은 45(10+35), 'JOB 3'의 반환 시간은 65(40+25)입니다. 즉 평균 반환 시간은 (13+45+65)/3 = 41입니다.

3. 스케줄링 하고자 하는 세 작업의 도착 시간과 실행 시간은 다음 표와 같다. 이 작업을 SJF로 스케줄링 하였을 때, 작업 2의 종료 시간은? (단, 여기서 오버헤드는 무시한다.)

작업	도착 시간	실행 시간
1	0	6
2	1	3
3	2	4

① 3
② 6
③ 9
④ 13

SJF 스케줄링은 실행 시간이 가장 짧은 프로세스에게 먼저 CPU를 할당하는 기법입니다. 처음에 도착한 작업 1이 먼저 실행된 후 그 다음에 도착한 작업중 실행 시간이 짧은 작업을 먼저 실행하게 됩니다. 작업 1의 실행(6)을 마친 후 작업 2가 실행(3)되므로 작업 2의 종료 시간은 9가 됩니다.



4. HRN 스케줄링에서 우선순위 계산식으로 올바른 것은?

① (대기 시간 + 서비스 시간)/서비스 시간
② (대기 시간 + 서비스 시간)/대기 시간
③ (대기 시간 + 응답 시간)/응답 시간
④ (대기 시간 + 응답 시간)/대기 시간

HRN 기법은 서비스 시간이 짧거나 대기 시간이 긴 프로세스일 경우 우선순위가 높아진다는 것을 염두에 두고 계산식을 떠올려보세요.

5. HRN(Highest Response-ratio Next) 방식으로 스케줄링할 경우, 입력된 작업이 다음과 같을 때 우선순위가 가장 높은 작업은?

작업	대기 시간	서비스 시간
A	5	5
B	10	6
C	15	7
D	20	8

① A
② B
③ C
④ D

4번 문제에서 우선순위 계산식을 암기했다면 계산식에 대입해서 풀면 되겠죠? A 작업은 (5+5)/5=2, B 작업은 (10+6)/6=2.6, C 작업은 (15+7)/7=3.1, D 작업은 (20+8)/8=3.5입니다.

6. 프로세스가 자원을 기다리고 있는 시간에 비례하여 우선순위를 부여함으로써 무기한 문제를 방지하는 기법은?

① 노화(Aging) 기법
② 재사용(Reusable) 기법
③ 환형대기(Circular Wait)
④ 치명적인 포옹(Deadly Embrace)

Aging은 '나이를 먹음'이란 의미로, 준비상태 큐에서 오래 기다린 프로세스의 나이(우선순위)를 증가시켜 자원을 할당받도록 하는 것입니다. 의미를 생각하면 잊어 버리지 않겠죠?

▶ 정답 : 1. ④ 2. ④ 3. ③ 4. ① 5. ④ 6. ①

병행학습
013
LIVE

선점 스케줄링

전문가의 조언

선점 기법에 해당하는 각 알고리즘의 종류와 각각의 의미를 정확히 파악해 두세요.

선점 스케줄링에 해당하는 선점 우선순위, SRT, RR, 다단계 큐, 다단계 피드백 큐 알고리즘에 대해 알아보자.

1 선점 우선순위

- 준비상태 큐의 프로세스들 중에서 우선순위가 가장 높은 프로세스에게 먼저 CPU를 할당하는 기법이다.
- 비선점 우선순위 기법을 선점 형태로 변경한 것으로, 준비상태 큐에 새로 들어온 프로세스의 순위가 높을 경우 현재의 프로세스를 보류하고 새로운 프로세스를 실행한다.

2 SRT(Shortest Remaining Time)

- 비선점 스케줄링인 SJF 기법을 선점 형태로 변경한 기법으로, 선점 SJF 기법이라고도 한다.
- 현재 실행중인 프로세스의 남은 시간과 준비상태 큐에 새로 도착한 프로세스의 실행 시간을 비교하여 가장 짧은 실행 시간을 요구하는 프로세스에게 CPU를 할당하는 기법으로, 시분할 시스템에 유용하다.
- 준비상태 큐에 있는 각 프로세스의 실행 시간을 추적하여 보유하고 있어야 하므로 오버헤드가 증가한다.

3 RR(Round Robin)

- 시분할 시스템(Time Sharing System)을 위해 고안된 방식으로, FCFS(FIFO) 알고리즘을 선점 형태로 변형한 기법이다.
- FCFS 기법과 같이 준비상태 큐에 먼저 들어온 프로세스가 먼저 CPU를 할당받지만 각 프로세스는 시간 할당량(Time Slice, Quantum) 동안만 실행한 후 실행이 완료되지 않으면 다음 프로세스에게 CPU를 넘겨주고 준비상태 큐의 가장 뒤로 배치된다.
- 할당되는 시간이 클 경우 FCFS 기법과 같아지고, 할당되는 시간이 작을 경우 문맥 교환 및 오버헤드가 자주 발생되어 요청된 작업을 신속히 처리할 수 없다.
- 할당되는 시간의 크기가 작으면 작은 프로세스들에게 유리하다.

예제 다음과 같은 프로세스들이 차례로 준비상태 큐에 들어왔다고 가정할 때, 평균 대기시간, 평균 반환 시간을 구하시오(단, Time Slice는 4초이다).

프로세스 번호	P1	P2	P3
실행 시간	20	4	6

① 주어진 시간 할당량(Time Slice) 동안 실행되지 못할 경우 준비상태 큐의 가장 마지막으로 재배치하여 차례를 기다리므로 다음과 같이 표시할 수 있다.

진행 시간 → 0	4	8	12	16	18	22	26	30
프로세스 번호	P1	P2	P3	P1	P3	P1	P1	P1
실행 시간	4	4	4	4	2	4	4	4

※ 색 동그라미는 프로세스가 완료됨을 표시한 것이다(P2 → 8초, P3 → 18초, P1 → 30초).

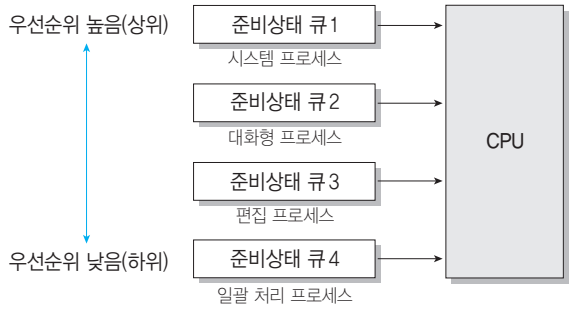
② 반환 시간 : 각 프로세스가 완료되는 시간을 이용하여 구한다.

③ 대기 시간 : 대기 시간을 구하고자 하는 프로세스의 가장 마지막 실행이 시작되기 전까지의 진행 시간을 이용하여 구하되, 해당 프로세스가 앞에서 실행되었을 경우 실행된 시간은 제외한다.

프로세스 번호	P1	P2	P3	평균
반환 시간	30	8	18	$\frac{56}{3} = 18.6$
대기 시간	26−16=10	4	16−4=12	$\frac{26}{3} = 8.6$

4 다단계 큐(MQ; Multi-level Queue)

- 프로세스를 특정 그룹으로 분류할 수 있을 경우 그룹에 따라 각기 다른 준비상태 큐를 사용하는 기법이다.
- FCFS(FIFO)와 RR 스케줄링 기법을 혼합한 것으로, 상위 단계에서 완료되지 못한 작업은 하위 단계로 전달되며 마지막 단계에서는 RR 스케줄링 기법을 사용한다.
- 일반적으로 프로세스 우선순위에 따라 시스템 프로세스, 대화형 프로세스, 편집 프로세스, 일괄 처리 프로세스 등으로 나누어 준비상태 큐를 상위, 중위, 하위 단계로 배치한다.



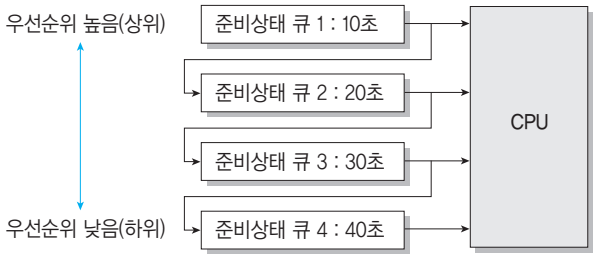
-

- 각 준비상태 큐는 독자적인 스케줄링을 가지고 있으므로 각 그룹의 특성에 따라 서로 다른 스케줄링 기법을 사용할 수 있다.
- 프로세스가 특정 그룹의 준비상태 큐에 들어갈 경우 다른 준비상태 큐로 이동할 수 없다.
- 하위 단계 준비상태 큐에 있는 프로세스를 실행하는 도중이라도 상위 단계 준비상태 큐에 프로세스가 들어오면 상위 단계 프로세스에게 CPU를 할당해야 한다.

5 다단계 피드백 큐(MFQ; Multi-level Feedback Queue)

- 특정 그룹의 준비상태 큐에 들어간 프로세스가 다른 준비상태 큐로 이동할 수 없는 다단계 큐 기법을 준비상태 큐 사이를 이동할 수 있도록 개선한 기법이다.
- 적응 기법(Adaptive Mechanism)*의 개념을 적용했다.
- 각 준비상태 큐마다 시간 할당량을 부여하여 그 시간 동안 완료하지 못한 프로세스는 다음 단계의 준비상태 큐로 이동된다.
- 상위 단계 준비상태 큐일수록 우선순위가 높고, 시간 할당량이 적다.
- 요구하는 시간이 적은 프로세스, 입 · 출력 중심의 프로세스, 낮은 우선순위에서 너무 오래 기다린 프로세스를 기준으로 높은 우선순위를 할당한다.
- 하위 단계 준비상태 큐에 있는 프로세스를 실행하는 도중이라도 상위 단계 준비상태 큐에 프로세스가 들어오면 상위 단계 프로세스에게 CPU를 할당하며, 마지막 단계 큐에서는 작업이 완료될 때까지 RR 스케줄링 기법을 사용한다.

적응 기법(Adaptive Mechanism)
시스템이 유동적인 상태 변화에 적절히 반응하도록 하는 기법을 의미합니다.



준비상태 큐

1. 스케줄링 기법에 대한 설명으로 옳지 않은 것은?

① RR 스케줄링은 주어진 시간 할당량(Time Slice) 안에 작업을 마치지 않으면 준비 완료 리스트(Ready List)의 가장 뒤로 배치되는 기법이다.

② SJF 스케줄링은 남아 있는 실행 시간의 추정치가 가장 작은 작업을 먼저 실행시키며, 언제라도 실행중인 작업이 강제로 실행을 멈출 수 있는 선점 기법이다.

③ HRN 스케줄링은 그 작업이 서비스받을 시간과 서비스를 기다린 시간으로 결정되는 우선순위에 따라 CPU를 할당한다.

④ 기한부(Deadline) 스케줄링은 제한된 시간 내에 반드시 작업이 완료되도록 스케줄링하는 기법이다.

선점, 비선점 스케줄링을 구분할 수 있으면 쉽게 풀리는 문제가 많습니다. 선점 스케줄링이든 비선점 스케줄링이든 하나의 종류만 확실히 알아두세요.

2. 스케줄링 기법 중 SJF 기법과 SRT 기법에 관한 설명으로 옳지 않은 것은?

① SJF는 비선점(Non-preemptive) 기법이다.

② SJF는 작업이 끝나기까지의 실행 시간 추정치가 가장 작은 작업을 먼저 실행시킨다.

③ SRT는 시분할 시스템에 유용하다.

④ SRT에서는 한 작업이 실행을 시작하면 강제로 실행을 멈출 수 없다.

이 문제도 선점인지 비선점인지만 구분하면 쉽게 풀 수 있는 문제입니다. SRT는 선점! SJF는 비선점! 확실히 기억하세요.

3. RR(Round-Robin) 스케줄링에 대한 설명으로 옳지 않은 것은?

① 우선순위 계산식은 '(대기 시간 + 서비스 시간)/서비스 시간'이다.

② Time Sharing System을 위해 고안된 방식이다.

③ 시간 할당량이 커지면 FCFS 스케줄링과 같은 효과를 얻을 수 있다.

④ 시간 할당이 작아지면 프로세스 문맥 교환이 자주 일어난다.

①번과 같은 계산식으로 우선순위를 지정하는 것은 HRN 기법입니다.

4. 준비상태 큐에 프로세스 A, B, C가 차례로 도착하였다. 라운드로빈(Round Robin)으로 스케줄링 할 때 타임 슬라이스를 4초로 한다면 평균 반환 시간은?

프로세스	A	B	C
실행시간(초)	17	4	5

① 12초 ② 14초

③ 17초 ④ 18초

라운드 로빈(Round-Robin) 방식은 FCFS 기법과 같이 준비상태 큐에 먼저 들어온 프로세스가 먼저 CPU를 할당받지만 각 프로세스는 시간 할당량(Time Slice) 동안만 실행한 후 실행이 완료되지 않으면 다음 프로세스에게 CPU를 넘겨주고 준비상태 큐의 가장 뒤로 배치되는 기법이죠? 이를 바탕으로 다음과 같이 표를 작성하여 계산해 보세요.

진행 시간	0	4	● 8	12	16	● 17	21	25	● 26
작업순서	A	B	C	A	C	A	A	A	
실행 시간	4	4	4	4	1	4	4	1	

※ ●는 각 작업이 종료되는 시점을 의미합니다.
프로세스 A의 종료 시간, 즉 반환 시간은 26초, B는 8초, C는 17초가 되므로 평균 반환 시간은 (26+8+17) / 3 = 17초가 됩니다.

5. 적응 기법(Adaptive Mechanism)이란 시스템이 유동적인 상태 변화에 적절히 반응하도록 하는 기법을 의미한다. 다음 스케줄링 기법 중 적응 기법의 개념을 적용하고 있는 것은?

① FIFO ② HRN

③ MFQ ④ RR

다단계 큐(Multi-level Queue)와 다단계 피드백 큐(Multi-level Feedback Queue) 차이점 위주로 알아두라고 했죠. 모르겠으면 다시 한 번 보고 오세요.

6. FIFO와 RR 스케줄링 방식을 혼합한 것으로 상위 단계에서 완료되지 못한 작업은 하위 단계로 전달되어 마지막 단계에서는 RR 방식을 사용하는 것은?

① SJF ② SRT

③ HRN ④ Multilevel Queue

앞에서 다단계 큐와 다단계 피드백 큐를 확실히 공부했다면 어렵지 않게 답을 찾을 수 있겠죠.

▶ 정답 : 1. ② 2. ④ 3. ① 4. ③ 5. ③ 6. ④

병행학습 014 ● LIVE

데이터 무결성 검사 도구

전문가의 조언

데이터 무결성 검사 도구의 개념과 특징을 파악하고, 무결성 검사 도구의 종류를 숙지하세요.

1 데이터 무결성 검사 도구

데이터 무결성 검사 도구는 시스템의 파일 변경 유무를 확인하고 파일이 변경되었을 경우 이를 관리자에게 알려주는 역할을 수행하는 도구이다.

- 크래커나 허가받지 않은 내부 사용자들이 시스템에 침입하면 백도어를 만들어 놓거나 시스템 파일을 변경하여 자신의 흔적을 감추는데, 무결성 검사 도구를 이용하여 이를 감지할 수 있다.
- 데이터 무결성 도구는 현재 파일 및 디렉터리의 상태를 해시(Hash) 함수를 이용하여 DB 저장한 후 모니터링을 수행하다가 파일이나 디렉터리가 DB에 저장해둔 상태와 달라지는 경우 관리자에게 변경 사실을 알려준다.
- 대표적인 데이터 무결성 검사 도구에는 Tripwire, AIDE, Samhain, Claymore, Slipwire, Fcheck 등이 있다.

예상문제은행 병행학습 014

2020년 1 · 2회 통합 정보처리기사 필기 기출문제

1. 크래커가 침입하여 백도어를 만들어 놓거나, 설정 파일을 변경했을 때 분석하는 도구는?

① trace
② tripwire
③ udpdump
④ cron

문제에서 말하는 도구란 데이터 무결성 검사 도구로서, 보기 중에서는 tripwire가 해당됩니다.

2. 다음 지문의 괄호에 공통적으로 들어갈 용어는 무엇인가?

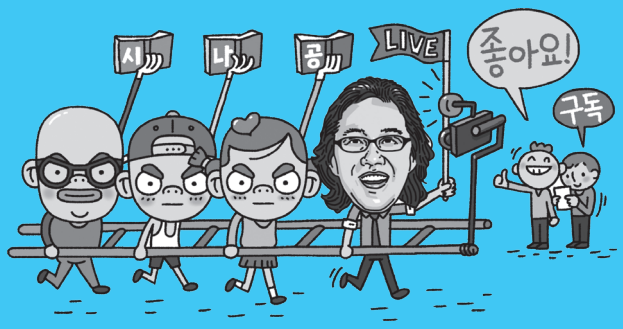
> 크래커나 허가받지 않은 내부 사용자들이 시스템에 침입하면 백도어를 만들어 놓거나 시스템 파일을 변경하여 자신의 흔적을 감추는데, 이를 (　　)를 이용하여 감지할 수 있다. (　　)는 현재 파일 및 디렉터리의 상태를 해시(Hash) 함수를 이용하여 DB 저장한 후 모니터링을 수행하다가 파일이나 디렉터리가 DB에 저장해둔 상태와 달라지는 경우 관리자에게 변경 사실을 알려준다

① 구조적 분석 도구
② 데이터 무결성 검사 도구
③ 시스템 자원 관리 도구
④ 자동화 추정 도구

데이터의 정확성(일관성)을 보장하기 위해 부정확한 자료가 저장되거나 기존의 데이터가 변경되었는지를 검사하는 것을 데이터 무결성 검사라고 합니다.

▶ 정답 : 1. ② 2. ②

최신기출문제



2020년 1·2회 통합 정보처리기사 필기 A형

440201

1과목 소프트웨어 설계

1. GoF(Gang of Four)의 디자인 패턴에서 행위 패턴에 속하는 것은?

① Builder ② Visitor
③ Prototype ④ Bridge

전문가 조언 | Visitor는 행위 패턴, Builder와 Prototype은 생성 패턴, Bridge는 구조 패턴에 속합니다.

병행학습 | 디자인 패턴의 종류

- **생성 패턴(Creational Pattern)** : 추상 팩토리(Abstract Factory), 빌더(Builder), 팩토리 메소드(Factory Method), 프로토타입(Prototype), 싱글톤(Singleton)
- **구조 패턴(Structural Pattern)** : 어댑터(Adapter), 브리지(Bridge), 컴포지트(Composite), 데코레이터(Decorator), 퍼싸드(Facade), 플라이웨이트(Flyweight), 프록시(Proxy)
- **행위 패턴(Behavioral Pattern)** : 책임 연쇄(Chain of Responsibility), 커맨드(Command), 인터프리터(Interpreter), 반복자(Iterator), 중재자(Mediator), 메멘토(Memento), 옵서버(Observer), 상태(State), 전략(Strategy), 템플릿 메소드(Template Method), 방문자(Visitor)

2. 객체지향 프로그램에서 데이터를 추상화하는 단위는?

① 메소드 ② 클래스
③ 상속성 ④ 메시지

전문가 조언 | 객체지향 프로그램에서 데이터를 추상화하는 단위를 클래스(Class)라고 합니다.

병행학습 | 객체지향 프로그래밍 언어의 구성 요소

- 객체(Object)
  - 데이터(속성)와 이를 처리하기 위한 연산(메소드)을 결합시킨 실체이다.
  - 데이터 구조와 그 위에서 수행되는 연산들을 가지고 있는 소프트웨어 모듈이다.
  - 속성(Attribute) : 한 클래스 내에 속한 객체들이 가지고 있는 데이터 값들을 단위별로 정의하는 것으로서 성질, 분류, 식별, 수량 또는 현재 상태 등을 표현함
  - 메소드(Method) : 객체가 메시지를 받아 실행해야 할 때 구체적인 연산을 정의하는 것으로, 객체의 상태를 참조하거나 변경하는 수단이 됨
- 클래스(Class)
  - 두 개 이상의 유사한 객체들을 묶어서 하나의 공통된 특성을 표현하는 요소이다. 즉 공통된 특성과 행위를 갖는 객체의 집합이라고 할 수 있다.
  - 객체의 유형 또는 타입(Object Type)을 의미한다.
- 메시지(Message)
  - 객체들 간에 상호작용을 하는데 사용되는 수단으로 객체의 메소드(동작, 연산)를 일으키는 외부의 요구 사항이다.
  - 메시지를 받은 객체는 대응하는 연산을 수행하여 예상된 결과를 반환하게 된다.

3. 객체지향 기법에서 클래스들 사이의 '부분-전체(Part-Whole)' 관계 또는 '부분(is-a-part-of)'의 관계로 설명되는 연관성을 나타내는 용어는?

① 일반화 ② 추상화
③ 캡슐화 ④ 집단화

전문가 조언 | 클래스들 사이의 '부분-전체(Part-Whole)' 관계 또는 '부분(is-a-part-of)'의 관계와 같이 하나의 사물이 다른 사물에 포함되어 있는 관계를 집합 또는 집단 관계라고 합니다.

병행학습 | 관계(Relationships)

- **연관(Association) 관계** : 2개 이상의 사물이 서로 관련되어 있음
- **집합(Aggregation) 관계** : 하나의 사물이 다른 사물에 포함되어 있는 관계
- **포함(Composition) 관계** : 집합 관계의 특수한 형태로, 포함하는 사물의 변화가 포함되는 사물에게 영향을 미치는 관계
- **일반화(Generalization) 관계** : 하나의 사물이 다른 사물에 비해 더 일반적인지 구체적인지를 표현하는 관계
- **의존(Dependency) 관계** : 연관 관계와 같이 사물 사이에 서로 연관은 있으나 필요에 의해 서로에게 영향을 주는 짧은 시간 동안만 연관을 유지하는 관계
- **실체화(Realization) 관계** : 사물이 할 수 있거나 해야 하는 기능(행위, 인터페이스)으로 서로를 그룹화 할 수 있는 관계

4. 객체지향 분석 방법론 중 E-R 다이어그램을 사용하여 객체의 행위를 모델링하며, 객체 식별, 구조 식별, 주체 정의, 속성 및 관계 정의, 서비스 정의 등의 과정으로 구성되는 것은?

① Coad와 Yourdon 방법
② Booch 방법
③ Jacobson 방법
④ Wirfs-Brocks 방법

전문가 조언 | 문제에 제시된 내용은 Coad와 Yourdon 방법에 대한 설명입니다.

병행학습 | 객체지향 분석의 방법론

- **Rumbaugh(럼바우) 방법** : 가장 일반적으로 사용되는 방법으로 분석 활동을 객체 모델, 동적 모델, 기능 모델로 나누어 수행하는 방법
- **Booch(부치) 방법** : 미시적(Micro) 개발 프로세스와 거시적(Macro) 개발 프로세스를 모두 사용하는 분석 방법으로, 클래스와 객체들을 분석 및 식별하고 클래스의 속성과 연산을 정의함
- **Jacobson 방법** : Use Case를 강조하여 사용하는 분석 방법
- **Coad와 Yourdon 방법** : E-R 다이어그램을 사용하여 객체의 행위를 모델링하며, 객체 식별, 구조 식별, 주체 정의, 속성과 인스턴스 연결 정의, 연산과 메시지 연결 정의 등의 과정으로 구성하는 기법
- **Wirfs-Brock 방법** : 분석과 설계 간의 구분이 없고, 고객 명세서를 평가해서 설계 작업까지 연속적으로 수행하는 기법

5. 코드 설계에서 일정한 일련번호를 부여하는 방식의 코드는?

① 연상 코드 ② 블록 코드
③ 순차 코드 ④ 표의 숫자 코드

전문가 조언 | 코드 설계에서 일정한 일련번호를 부여하는 방식의 코드는 순차 코드입니다.

병행학습 | **코드(Code)**

- 컴퓨터를 이용하여 자료를 처리하는 과정에서 분류 · 조합 및 집계를 용이하게 하고, 특정 자료의 추출을 쉽게 하기 위해서 사용하는 기호이다.
- **순차 코드**(Sequence Code) : 자료의 발생 순서, 크기 순서 등 일정 기준에 따라서 최초의 자료부터 차례로 일련번호를 부여하는 방법
- **블록 코드**(Block Code) : 코드화 대상 항목 중에서 공통성이 있는 것끼리 블록으로 구분하고, 각 블록 내에서 일련번호를 부여하는 방법
- **10진 코드**(Decimal Code) : 코드화 대상 항목을 0~9까지 10진 분할하고, 다시 그 각각에 대하여 10진 분할하는 방법을 필요한 만큼 반복하는 방법
- **그룹 분류 코드**(Group Classification Code) : 코드화 대상 항목을 일정 기준에 따라 대분류, 중분류, 소분류 등으로 구분하고, 각 그룹 안에서 일련번호를 부여하는 방법
- **연상 코드**(Mnemonic Code) : 코드화 대상 항목의 명칭이나 약호와 관계있는 숫자나 문자, 기호를 이용하여 코드를 부여하는 방법
- **표의 숫자 코드**(Significant Digit Code) : 코드화 대상 항목의 성질, 즉 길이, 넓이, 부피, 지름, 높이 등의 물리적 수치를 그대로 코드에 적용시키는 방법
- **합성 코드**(Combined Code) : 필요한 기능을 하나의 코드로 수행하기 어려운 경우 2개 이상의 코드를 조합하여 만드는 방법

6. 소프트웨어 설계 시 구축된 플랫폼의 성능 특성 분석에 사용되는 측정 항목이 아닌 것은?

① 응답시간(Response Time)
② 가용성(Availability)
③ 사용률(Utilization)
④ 서버 튜닝(Server Tuning)

전문가 조언 | 서버 튜닝은 서버의 성능을 개선하는 것으로, 성능 특성 분석에 사용되는 측정 항목이 될 수 없습니다.

병행학습 |

- **응답시간**(Response Time) : 요청을 전달한 시간부터 응답이 도착할 때까지 걸린 시간
- **가용성**(Availability) : 시스템을 사용할 필요가 있을 때 즉시 사용 가능한 정도
- **사용률**(Utilization) : 의뢰한 작업을 처리하는 동안의 CPU 사용량, 메모리 사용량, 네트워크 사용량 등 자원 사용률

7. 자료 사전에서 자료의 생략을 의미하는 기호는?

① { } ② **
③ = ④ ( )

전문가 조언 | 자료 사전에서 자료의 생략을 의미하는 기호는 ( )입니다.

병행학습 | **자료 사전(DD; Data Dictionary)**

- 자료 흐름도에 있는 자료를 더 자세히 정의하고 기록한 것이며, 이처럼 데이터를 설명하는 데이터를 데이터의 데이터 또는 메타 데이터(Meta Data)라고 한다.
- 자료 흐름도에 시각적으로 표시된 자료에 대한 정보를 체계적이고 조직적으로 모아 개발자나 사용자가 편리하게 사용할 수 있다.
- 자료 사전에서 사용되는 표기 기호

기호	의미
=	자료의 정의
+	자료의 연결
( )	자료의 생략
[|]	자료의 선택
{ }	자료의 반복
**	자료의 설명

8. 검토회의 전에 요구사항 명세서를 미리 배포하여 사전 검토한 후 짧은 검토 회의를 통해 오류를 조기에 검출하는데 목적을 두는 요구사항 검토 방법은?

① 빌드 검증 ② 동료 검토
③ 워크스루 ④ 개발자 검토

전문가 조언 | 문제에서 설명하는 요구사항 검토 방법은 워크스루(Walk Through)입니다.

병행학습 | **요구사항 검토(Requirements Review)**

- 요구사항 명세서의 오류 확인 및 표준 준수 여부 등의 결함 여부를 검토 담당자들이 수작업으로 분석하는 방법이다.
- 종류
  - 동료 검토(Peer Review) : 요구사항 명세서 작성자가 명세서 내용을 직접 설명하고 동료들이 이를 들으면서 결함을 발견하는 형태의 검토 방법
  - 워크스루(Walk Through) : 검토 회의 전에 요구사항 명세서를 미리 배포하여 사전 검토한 후에 짧은 검토 회의를 통해 결함을 발견하는 형태의 검토 방법
  - 인스펙션(Inspection) : 요구사항 명세서 작성자를 제외한 다른 검토 전문가들이 요구사항 명세서를 확인하면서 결함을 발견하는 형태의 검토 방법

9. CASE가 갖고 있는 주요 기능이 아닌 것은?

① 그래픽 지원
② 소프트웨어 생명주기 전 단계의 연결
③ 언어 번역
④ 다양한 소프트웨어 개발 모형 지원

전문가 조언 | CASE의 주요 기능에는 소프트웨어 생명주기 전 단계의 연결, 다양한 소프트웨어 개발 모형 지원, 그래픽 지원 등이 있습니다.

정답 : 1.② 2.② 3.④ 4.① 5.③ 6.④ 7.④ 8.③ 9.③

병행학습 | CASE(Computer Aided Software Engineering)

- 소프트웨어 개발 과정에서 사용되는 요구 분석, 설계, 구현, 검사 및 디버깅 과정 전체 또는 일부를 컴퓨터와 전용 소프트웨어 도구를 사용하여 자동화하는 것이다.
- 소프트웨어, 하드웨어, 데이터베이스, 테스트 등을 통합하여 소프트웨어를 개발하는 환경을 조성한다.
- 소프트웨어 생명 주기의 전체 단계를 연결해 주고 자동화해 주는 통합된 도구를 제공해 주는 기술이다.
- 소프트웨어 개발 도구와 방법론이 결합된 것으로, 정형화된 구조 및 방법(메커니즘)을 소프트웨어 개발에 적용하여 생산성 향상을 구현하는 공학 기법이다.
- 소프트웨어 개발의 모든 단계에 걸쳐 일관된 방법론을 제공하는 자동화 도구(CASE Tool)들을 지원하고, 개발자들은 이 도구를 사용하여 소프트웨어 개발의 표준화를 지향하며, 자동화의 이점을 얻을 수 있게 해준다.
- **CASE의 주요 기능** : 소프트웨어 생명주기 전 단계의 연결, 다양한 소프트웨어 개발 모형 지원, 그래픽 지원 등

10. XP(eXtreme Programing)의 5가지 가치로 거리가 먼 것은?

① 용기 ② 의사소통
③ 정형 분석 ④ 피드백

전문가 조언 | XP의 5가지 핵심 가치는 의사소통(Communication), 단순성(Simplicity), 용기(Courage), 존중(Respect), 피드백(Feedback)입니다.

병행학습 | XP(eXtreme Programming)

- 수시로 발생하는 고객의 요구사항에 유연하게 대응하기 위해 고객의 참여와 개발 과정의 반복을 극대화하여 개발 생산성을 향상시키는 방법이다.
- XP는 짧고 반복적인 개발 주기, 단순한 설계, 고객의 적극적인 참여를 통해 소프트웨어를 빠르게 개발하는 것을 목적으로 한다.
- 릴리즈의 기간을 짧게 반복하면서 고객의 요구사항 반영에 대한 가시성을 높인다.
- 릴리즈 테스트마다 고객을 직접 참여시킴으로써 요구한 기능이 제대로 작동하는지 고객이 직접 확인할 수 있다.
- 비교적 소규모 인원의 개발 프로젝트에 효과적이다.
- **XP의 5가지 핵심 가치** : 의사소통(Communication), 단순성(Simplicity), 용기(Courage), 존중(Respect), 피드백(Feedback)

11. DBMS 분석 시 고려사항으로 거리가 먼 것은?

① 가용성 ② 성능
③ 네트워크 구성도 ④ 상호 호환성

전문가 조언 | DBMS 분석 시 고려사항에는 가용성, 성능, 기술 지원, 상호 호환성, 구축 비용이 있습니다.

병행학습 | DBMS 관련 요구사항 식별 시 고려사항

- 가용성
  - 시스템의 장시간 운영으로 인해 발생할 수 있는 운영체제 고유의 장애 발생 가능성
  - DBMS의 결함 등으로 인한 패치 설치를 위한 재가동
  - 백업이나 복구의 편의성
  - DBMS 이중화 및 복제 지원
- 성능
  - 대규모 데이터 처리 성능(분할 테이블 지원 여부)
  - 대용량 트랜잭션 처리 성능
  - 튜닝 옵션의 다양한 지원
  - 최소화된 설정과 비용 기반 질의 최적화 지원
- 기술 지원
  - 제작업체의 안정적인 기술 지원
  - 여러 사용자들 간의 정보 공유
  - 오픈 소스 여부
- 상호 호환성
  - 설치 가능한 운영체제의 종류
  - JDBC, ODBC와의 호환 여부
- 구축 비용
  - 라이선스 정책 및 비용
  - 유지관리 비용
  - 총 소유 비용(TCO)

12. HIPO(Hierarchy Input Process Output)에 대한 설명으로 거리가 먼 것은?

① 상향식 소프트웨어 개발을 위한 문서화 도구이다.
② HIPO 차트 종류에는 가시적 도표, 총체적 도표, 세부적 도표가 있다.
③ 기능과 자료의 의존 관계를 동시에 표현할 수 있다.
④ 보기 쉽고 이해하기 쉽다.

전문가 조언 | HIPO는 시스템의 분석 및 설계나 문서화할 때 사용되는 기법으로, 하향식 소프트웨어 개발을 위한 문서화 도구입니다.

병행학습 | HIPO(Hierarchy Input Process Output)

- 시스템의 분석 및 설계나 문서화할 때 사용되는 기법으로, 시스템 실행 과정인 입력, 처리, 출력의 기능을 나타낸다.
- 기본 시스템 모델은 입력, 처리, 출력으로 구성되며, 하향식 소프트웨어 개발을 위한 문서화 도구이다.
- 체계적인 문서 관리가 가능하다.
- 기호, 도표 등을 사용하므로 보기 쉽고 이해하기도 쉽다.
- 기능과 자료의 의존 관계를 동시에 표현할 수 있다.
- 변경, 유지보수가 용이하다.
- 시스템의 기능을 여러 개의 고유 모듈들로 분할하여 이들 간의 인터페이스를 계층 구조로 표현한 것을 HIPO Chart라고 한다.
- **HIPO Chart의 종류**
  - 가시적 도표(도식 목차) : 시스템의 전체적인 기능과 흐름을 보여주는 계층(Tree) 구조도
  - 총체적 도표(총괄도표, 개요 도표) : 프로그램을 구성하는 기능을 기술한 것으로 입력, 처리, 출력에 대한 전반적인 정보를 제공하는 도표
  - 세부적 도표(상세 도표) : 총체적 도표에 표시된 기능을 구성하는 기본 요소들을 상세히 기술하는 도표

13. UI 설계 원칙에서 누구나 쉽게 이해하고 사용할 수 있어야 한다는 것은?

① 유효성 ② 직관성
③ 무결성 ④ 유연성

전문가 조언 | 누구나 쉽게 이해하고 사용할 수 있어야 한다는 UI의 설계 원칙은 직관성입니다.

병행학습 | 사용자 인터페이스(UI)의 기본 원칙

- **직관성** : 누구나 쉽게 이해하고 사용할 수 있어야 함
- **유효성** : 사용자의 목적을 정확하고 완벽하게 달성해야 함
- **학습성** : 누구나 쉽게 배우고 익힐 수 있어야 함
- **유연성** : 사용자의 요구사항을 최대한 수용하고 실수를 최소화해야 함

14. 럼바우(Rumbaugh)의 객체지향 분석 절차를 가장 바르게 나열한 것은?

① 객체 모형 → 동적 모형 → 기능 모형
② 객체 모형 → 기능 모형 → 동적 모형
③ 기능 모형 → 동적 모형 → 객체 모형
④ 기능 모형 → 객체 모형 → 동적 모형

전문가 조언 | 럼바우 분석 기법의 활동은 객체 모델링, 동적 모델링, 기능 모델링 순으로 이루어집니다.

병행학습 | 럼바우(Rumbaugh)의 분석 기법

- 모든 소프트웨어 구성 요소를 그래픽 표기법을 이용하여 모델링하는 기법으로, 객체 모델링 기법(OMT, Object-Modeling Technique)이라고도 한다.
- 분석 활동은 객체 모델링, 동적 모델링, 기능 모델링을 통해 이루어진다.
- **객체 모델링(Object Modeling)**
  - 정보 모델링이라고도 하며, 시스템에서 요구되는 객체를 찾아내어 속성과 연산 식별 및 객체들 간의 관계를 규정하여 객체 다이어그램으로 표시하는 것이다.
  - 분석 활동의 세 가지 모델 중 가장 중요하며 선행되어야 할 모델링이다.
- **동적 모델링(Dynamic Modeling)**
  - 상태 다이어그램(상태도)을 이용하여 시간의 흐름에 따른 객체들 간의 제어 흐름, 상호 작용, 동작 순서 등의 동적인 행위를 표현하는 모델링이다.
  - 동적 모델링에서는 객체나 클래스의 상태, 사건을 중심으로 다룬다.
- **기능 모델링(Functional Modeling)**
  - 자료 흐름도(DFD)를 이용하여 다수의 프로세스들 간의 자료 흐름을 중심으로 처리 과정을 표현한 모델링이다.
  - 어떤 데이터를 입력하여 어떤 결과를 구할 것인지를 표현하는 것이다.

15. 데이터 흐름도(DFD)의 구성 요소에 포함되지 않는 것은?

① Process ② Data Flow
③ Data Store ④ Data Dictionary

전문가 조언 | 자료 흐름도(DFD)의 구성 요소에는 프로세스(Process), 자료 흐름(Data Flow), 자료 저장소(Data Store), 단말(Terminator)이 있습니다.

병행학습 | 자료 흐름도(DFD; Data Flow Diagram)

- 요구사항 분석에서 자료의 흐름 및 변환 과정과 기능을 도형 중심으로 기술하는 방법으로 자료 흐름 그래프, 버블 차트라고도 한다.
- 시스템 안의 프로세스와 자료 저장소 사이에 자료의 흐름을 나타내는 그래프로 자료 흐름과 처리를 중심으로 하는 구조적 분석 기법에 이용된다.
- 자료 흐름도는 자료 흐름과 기능을 자세히 표현하기 위해 단계적으로 세분화된다.
- 자료는 처리(Process)를 거쳐 변환될 때마다 새로운 이름이 부여되며, 처리는 입력 자료가 발생하면 기능을 수행한 후 출력 자료를 산출한다.
- 자료 흐름도 구성 요소 표기법
  - 프로세스(Process)
    - ▶ 자료를 변환시키는 시스템의 한 부분(처리 과정)을 나타내며 처리, 기능, 변환, 버블이라고도 한다.
    - ▶ 원이나 둥근 사각형으로 표시하고 그 안에 프로세스 이름을 기입한다.
  - 자료 흐름(Data Flow)
    - ▶ 자료의 이동(흐름)이나 연관관계를 나타낸다.
    - ▶ 화살표 위에 자료의 이름을 기입한다.
  - 자료 저장소(Data Store)
    - ▶ 시스템에서의 자료 저장소(파일, 데이터베이스)를 나타낸다.
    - ▶ 도형 안에 자료 저장소 이름을 기입한다.
  - 단말(Terminator)
    - ▶ 시스템과 교신하는 외부 개체로, 입력 데이터가 만들어지고 출력 데이터를 받는다(정보의 생산자와 소비자).
    - ▶ 도형 안에 이름을 기입한다.

16. UML 확장 모델에서 스테레오 타입 객체를 표현할 때 사용하는 기호로 맞는 것은?

① 《 》 ② (( ))
③ {{ }} ④ [[ ]]

전문가 조언 | UML 확장 모델에서 스테레오 타입 객체를 표현할 때 사용하는 기호는 《 》입니다.

17. 트랜잭션이 올바르게 처리되고 있는지 데이터를 감시하고 제어하는 미들웨어는?

① RPC ② ORB
③ TP monitor ④ HUB

전문가 조언 | 트랜잭션이 올바르게 처리되고 있는지 데이터를 감시하고 제어하는 미들웨어는 TP-Monitor(Transaction Processing Monitor)입니다.

병행학습 | 미들웨어(Middleware)의 종류

- DB(DataBase) : 데이터베이스 벤더(Vendor)에서 제공하는 클라이언트에서 원격의 데이터베이스와 연결하기 위한 미들웨어
- RPC(Remote Procedure Call) : 응용 프로그램의 프로시저를 사용하여 원격 프로시저를 마치 로컬 프로시저처럼 호출하는 방식의 미들웨어
- MOM(Message Oriented Middleware) : 메시지 기반의 비동기형 메시지를 전달하는 방식의 미들웨어
- TP-Monitor(Transaction Processing Monitor) : 항공기나 철도 예약 업무 등과 같은 온라인 트랜잭션 업무에서 트랜잭션을 처리 및 감시하는 미들웨어
- ORB(Object Request Broker) : 객체 지향 미들웨어로 코바(CORBA) 표준 스펙을 구현한 미들웨어
- WAS(Web Application Server) : 사용자의 요구에 따라 변하는 동적인 콘텐츠를 처리하기 위해 사용되는 미들웨어

18. 소프트웨어 개발 방법 중 요구사항 분석(Requirements Analysis)과 거리가 먼 것은?

① 비용과 일정에 대한 제약설정
② 타당성 조사
③ 요구사항 정의 문서화
④ 설계 명세서 작성

전문가 조언 | 설계 명세서를 작성하는 과정은 요구사항 명세(Requirement Specification)에 해당합니다.

병행학습 | 요구사항 분석(Requirement Analysis)

- 개발 대상에 대한 사용자의 요구사항 중 명확하지 않거나 모호하여 이해되지 않는 부분을 발견하고 이를 걸러내기 위한 과정이다.
- 사용자 요구사항의 타당성을 조사하고 비용과 일정에 대한 제약을 설정한다.
- 내용이 중복되거나 하나로 통합되어야 하는 등 서로 상충되는 요구사항이 있으면 이를 해결한다.
- 도출된 요구사항들을 토대로 소프트웨어의 범위를 파악한다.
- 도출된 요구사항들을 토대로 소프트웨어와 주변 환경이 상호 작용하는 방법을 이해한다.

19. 공통 모듈에 대한 명세 기법 중 해당 기능에 대해 일관되게 이해되고 한 가지로 해석될 수 있도록 작성하는 원칙은?

① 상호작용성 ② 명확성
③ 독립성 ④ 내용성

전문가 조언 | 문제에서 설명하는 공통 모듈의 명세 기법은 명확성(Clarity)입니다.

병행학습 | 공통 모듈의 명세 기법

- **정확성(Correctness)** : 시스템 구현 시 해당 기능이 필요하다는 것을 알 수 있도록 정확히 작성
- **명확성(Clarity)** : 해당 기능을 이해할 때 중의적으로 해석되지 않도록 명확하게 작성
- **완전성(Completeness)** : 시스템 구현을 위해 필요한 모든 것을 기술
- **일관성(Consistency)** : 공통 기능들 간 상호 충돌이 발생하지 않도록 작성
- **추적성(Traceability)** : 기능에 대한 요구사항의 출처, 관련 시스템 등의 관계를 파악할 수 있도록 작성

20. UML 모델에서 사용하는 Structural Diagram에 속하지 않은 것은?

① Class Diagram
② Object Diagram
③ Component Diagram
④ Activity Diagram

전문가 조언 | 활동 다이어그램(Activity Diagram)은 행위 다이어그램(Behavioral Diagram)에 속합니다.

병행학습 | UML 다이어그램의 종류

- **구조적(Structural) 다이어그램의 종류**
  - 클래스 다이어그램(Class Diagram) : 클래스와 클래스가 가지는 속성, 클래스 사이의 관계를 표현
  - 객체 다이어그램(Object Diagram) : 클래스에 속한 사물(객체)들, 즉 인스턴스(Instance)를 특정 시점의 객체와 객체 사이의 관계로 표현
  - 컴포넌트 다이어그램(Component Diagram) : 실제 구현 모듈인 컴포넌트 간의 관계나 컴포넌트 간의 인터페이스를 표현
  - 배치 다이어그램(Deployment Diagram) : 결과물, 프로세스, 컴포넌트 등 물리적 요소들의 위치를 표현
  - 복합체 구조 다이어그램(Composite Structure Diagram) : 클래스나 컴포넌트가 복합 구조를 갖는 경우 그 내부 구조를 표현
  - 패키지 다이어그램(Package Diagram) : 유스케이스나 클래스 등의 모델 요소들을 그룹화한 패키지들의 관계를 표현
- **행위(Behavioral) 다이어그램의 종류**
  - 유스케이스 다이어그램(Use Case Diagram) : 사용자의 요구를 분석하는 것으로 기능 모델링 작업에 사용함
  - 시퀀스 다이어그램(Sequence Diagram) : 상호 작용하는 시스템이나 객체들이 주고받는 메시지를 표현
  - 커뮤니케이션 다이어그램(Communication Diagram) : 시퀀스 다이어그램과 같이 동작에 참여하는 객체들이 주고받는 메시지를 표현하는데, 메시지뿐만 아니라 객체들 간의 연관까지 표현
  - 상태 다이어그램(State Diagram) : 하나의 객체가 자신이 속한 클래스의 상태 변화 혹은 다른 객체와의 상호 작용에 따라 상태가 어떻게 변화하는지를 표현
  - 활동 다이어그램(Activity Diagram) : 시스템이 어떤 기능을 수행하는지 객체의 처리 로직이나 조건에 따른 처리의 흐름을 순서에 따라 표현
  - 상호작용 개요 다이어그램(Interaction Overview Diagram) : 상호작용 다이어그램 간의 제어 흐름을 표현
  - 타이밍 다이어그램(Timing Diagram) : 객체 상태 변화와 시간 제약을 명시적으로 표현

440202

2과목 소프트웨어 개발

21. 평가 점수에 따른 성적부여는 다음 표와 같다. 이를 구현한 소프트웨어를 경계 값 분석 기법으로 테스트 하고자 할 때 다음 중 테스트 케이스의 입력 값으로 옳지 않은 것은?

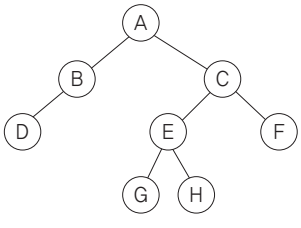
평가 점수	성적
80~100	A
60~79	B
0~59	C

① 59 ② 80
③ 90 ④ 101

전문가 조언 | 경계값 분석 기법은 입력 조건의 경계값을 테스트 케이스로 선정하여 검사하는 기법으로, 성적이 분리되는 평가 점수의 경계값인 101, 100, 80, 79, 60, 59, 0, –1이 적절한 입력값에 해당됩니다.

병행학습 | **경계값 분석(Boundary Value Analysis)**
- 블랙박스 테스트의 종류 중 하나로, 입력 자료에만 치중한 동치 분할 기법을 보완하기 위한 기법이다.
- 입력 조건의 중간값보다 경계값에서 오류가 발생될 확률이 높다는 점을 이용하여 입력 조건의 경계값을 테스트 케이스로 선정하여 검사한다.

22. 다음 트리의 차수(Degree)와 단말 노드(Terminal Node)의 수는?



① 차수 : 4, 단말 노드 : 4
② 차수 : 2, 단말 노드 : 4
③ 차수 : 4, 단말 노드 : 8
④ 차수 : 2, 단말 노드 : 8

전문가 조언 |
- 트리(Tree)의 차수(Degree)는 가장 차수가 많은 노드의 차수를 의미합니다. 문제에 주어진 트리에서 각 노드의 차수는 A=2, B=1, C=2, E=2이므로 A, C, E 노드의 차수인 2가 트리의 차수입니다.
- 단말 노드는 자식이 하나도 없는 노드를 의미합니다. 제시된 그림에서 자식이 없는 단말 노드는 D, F, G, H로 총 4개입니다.

23. 검증 검사 기법 중 개발자의 장소에서 사용자가 개발자가 앞에서 행하는 기법이며, 일반적으로 통제된 환경에서 사용자와 개발자가 함께 확인하면서 수행되는 검사는?

① 동치 분할 검사 ② 형상 검사
③ 알파 검사 ④ 베타 검사

전문가 조언 | 문제에 제시된 내용은 알파 검사에 대한 설명입니다.

병행학습 | **인수 테스트의 종류**
- **사용자 인수 테스트** : 사용자가 시스템 사용의 적절성 여부를 확인
- **운영상의 인수 테스트** : 시스템 관리자가 시스템 인수 시 수행하는 테스트 기법으로, 백업/복원 시스템, 재난 복구, 사용자 관리, 정기 점검 등을 확인
- **계약 인수 테스트** : 계약상의 인수/검수 조건을 준수하는지 여부를 확인
- **규정 인수 테스트** : 소프트웨어가 정부 지침, 법규, 규정 등 규정에 맞게 개발되었는지 확인
- **알파 테스트** : 개발자의 장소에서 사용자가 개발자 앞에서 행하는 테스트 기법으로, 통제된 환경에서 행해지며, 오류와 사용상의 문제점을 사용자와 개발자가 함께 확인하면서 기록함
- **베타 테스트** : 선정된 최종 사용자가 여러 명의 사용자 앞에서 행하는 테스트 기법으로, 실업무를 가지고 사용자가 직접 테스트하고, 개발자에 의해 제어되지 않은 상태에서 테스트가 행해지며, 발견된 오류와 사용상의 문제점을 기록하고 개발자에게 주기적으로 보고함

24. 하향식 통합에 있어서 모듈 간의 통합 시험을 위해 일시적으로 필요한 조건만을 가지고 임시로 제공되는 시험용 모듈을 무엇이라고 하는가?

① Stub ② Driver
③ Procedure ④ Function

전문가 조언 | 하향식 통합 테스트에서 모듈 간의 통합 시험을 위해 일시적으로 필요한 조건만을 가지고 임시로 제공되는 시험용 모듈을 스텁(Stub)이라고 합니다.

병행학습 | **하향식 통합 테스트(Top Down Integration Test)**
- 프로그램의 상위 모듈에서 하위 모듈 방향으로 통합하면서 테스트하는 기법이다.
- 주요 제어 모듈을 기준으로 하여 아래 단계로 이동하면서 통합하는데, 이때 깊이 우선 통합법이나 넓이 우선 통합법을 사용한다.
- 테스트 초기부터 사용자에게 시스템 구조를 보여줄 수 있다.
- 상위 모듈에서는 테스트 케이스를 사용하기 어렵다.
- 하향식 통합 방법의 절차
  ❶ 주요 제어 모듈은 작성된 프로그램을 사용하고, 주요 제어 모듈의 종속 모듈들은 스텁(Stub)으로 대체한다.
  ❷ 깊이 우선 또는 넓이 우선 등의 통합 방식에 따라 하위 모듈인 스텁들이 한 번에 하나씩 실제 모듈로 교체된다.
  ❸ 모듈이 통합될 때마다 테스트를 실시한다.
  ❹ 새로운 오류가 발생하지 않음을 보증하기 위해 회귀 테스트를 실시한다.

※ **테스트 스텁(Test Stub)** : 제어 모듈이 호출하는 타 모듈의 기능을 단순히 수행하는 도구로, 일시적으로 필요한 조건만을 가지고 있는 시험용 모듈

25. 소프트웨어 품질 측정을 위해 개발자 관점에서 고려해야 할 항목으로 거리가 먼 것은?

① 정확성 ② 무결성
③ 사용성 ④ 간결성

전문가 조언 | 소프트웨어 품질 측정을 위해 개발자 관점에서 고려해야 할 항목에는 정확성, 사용성, 무결성 등이 있습니다.

병행학습 | ISO/IEC 9126의 소프트웨어 품질 특성별 상세 품질 요구사항

- **기능성(Functionality)** : 적절성/정합성(Suitability), 정밀성/정확성(Accuracy), 상호운용성(Interoperability), 보안성(Security), 호환성(Compliance)
- **신뢰성(Reliability)** : 성숙성(Maturity), 고장 허용성(Fault Tolerance), 회복성(Recoverability)
- **사용성(Usability)** : 이해성(Understandability), 학습성(Learnability), 운용성(Operability), 친밀성(Attractiveness)
- **효율성(Efficiency)** : 시간 효율성(Time Behaviour), 자원 효율성(Resource Behaviour)
- **유지 보수성(Maintainability)** : 분석성(Analyzability), 변경성(Changeability), 안정성(Stability), 시험성(Testability)
- **이식성(Portability)** : 적응성(Adaptability), 설치성(Installability), 대체성(Replaceability), 공존성(Co-existence)

26. 소프트웨어 테스트에서 오류의 80%는 전체 모듈의 20% 내에서 발견된다는 법칙은?

① Brooks의 법칙 ② Boehm의 법칙
③ Pareto의 법칙 ④ Jackson의 법칙

전문가 조언 | 소프트웨어 테스트에서 오류의 80%는 전체 모듈의 20% 내에서 발견된다는 법칙은 파레토 법칙(Pareto Principle)입니다.

병행학습 | 애플리케이션 테스트의 기본 원리

- 애플리케이션 테스트는 소프트웨어의 잠재적인 결함을 줄일 수 있지만 소프트웨어에 결함이 없다고 증명할 수는 없다. 즉 완벽한 소프트웨어 테스팅은 불가능하다.
- 애플리케이션의 결함은 대부분 개발자의 특성이나 애플리케이션의 기능적 특징 때문에 특정 모듈에 집중되어 있다. 애플리케이션의 20%에 해당하는 코드에서 전체 80%의 결함이 발견된다고 하여 파레토 법칙을 적용하기도 한다.
- 애플리케이션 테스트에서는 동일한 테스트 케이스로 동일한 테스트를 반복하면 더 이상 결함이 발견되지 않는 '살충제 패러독스(Pesticide Paradox)' 현상이 발생한다. 살충제 패러독스를 방지하기 위해서 테스트 케이스를 지속적으로 보완 및 개선해야 한다.
- 애플리케이션 테스트는 소프트웨어 특징, 테스트 환경, 테스터 역량 등 정황(Context)에 따라 테스트 결과가 달라질 수 있으므로, 정황에 따라 테스트를 다르게 수행해야 한다.
- 소프트웨어의 결함을 모두 제거해도 사용자의 요구사항을 만족시키지 못하면 해당 소프트웨어는 품질이 높다고 말할 수 없다. 이것을 오류-부재의 궤변(Absence of Errors Fallacy)이라고 한다.
- 테스트와 위험은 반비례한다. 테스트를 많이 하면 할수록 미래에 발생할 위험을 줄일 수 있다.
- 테스트는 작은 부분에서 시작하여 점점 확대하며 진행해야 한다.
- 테스트는 개발자와 관계없는 별도의 팀에서 수행해야 한다.

27. 디지털 저작권 관리(DRM)의 기술 요소가 아닌 것은?

① 크랙 방지 기술
② 정책 관리 기술
③ 암호화 기술
④ 방화벽 기술

전문가 조언 | 방화벽 기술은 디지털 저작권 관리 기술이 아니고 기업이나 조직 내부의 네트워크와 인터넷 간에 전송되는 정보를 선별하여 수용 · 거부 · 수정하는 기능을 가진 침입 차단 시스템입니다.

병행학습 | 디지털 저작권 관리(DRM)의 기술 요소

- **암호화(Encryption)** : 콘텐츠 및 라이선스를 암호화하고 전자 서명을 할 수 있는 기술
- **키 관리(Key Management)** : 콘텐츠를 암호화한 키에 대한 저장 및 분배 기술
- **암호화 파일 생성(Packager)** : 콘텐츠를 암호화된 콘텐츠로 생성하기 위한 기술
- **식별 기술(Identification)** : 콘텐츠에 대한 식별 체계 표현 기술
- **저작권 표현(Right Expression)** : 라이선스의 내용 표현 기술
- **정책 관리(Policy Management)** : 라이선스 발급 및 사용에 대한 정책 표현 및 관리 기술
- **크랙 방지(Tamper Resistance)** : 크랙에 의한 콘텐츠 사용 방지 기술
- **인증(Authentication)** : 라이선스 발급 및 사용의 기준이 되는 사용자 인증 기술

28. 인터페이스 보안을 위해 네트워크 영역에 적용될 수 있는 솔루션과 거리가 먼 것은?

① IPSec ② SMTP
③ SSL ④ S-HTTPS

전문가 조언 | SMTP(Simple Mail Transfer Protocol)는 전자 우편을 교환하는 서비스로, 인터페이스 보안과는 무관합니다.

병행학습 | 영역별 인터페이스 보안

- **네트워크 영역**
  - 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 등의 다양한 방식으로 적용한다.
- **애플리케이션 영역** : 소프트웨어 개발 보안 가이드를 참조하여 애플리케이션 코드 상의 보안 취약점을 보완하는 방향으로 애플리케이션 보안 기능을 적용함
- **데이터베이스 영역**
  - 데이터베이스, 스키마, 엔티티의 접근 권한과 프로시저(Procedure), 트리거(Trigger) 등 데이터베이스 동작 객체의 보안 취약점에 보안 기능을 적용한다.
  - 개인 정보나 업무상 민감한 데이터의 경우 암호화나 익명화 등 데이터 자체의 보안 방안도 고려한다.

29. 인터페이스 구현 검증 도구 중 아래에서 설명하는 것은?

> • 서비스 호출, 컴포넌트 재사용 등 다양한 환경을 지원하는 테스트 프레임워크
> • 각 테스트 대상 분산 환경에 데몬을 사용하여 테스트 대상 프로그램을 통해 테스트를 수행하고, 통합하여 자동화하는 검증 도구

① xUnit ② STAF
③ FitNesse ④ RubyNode

전문가 조언 | 인터페이스 구현 검증 도구 중 서비스 호출, 컴포넌트 재사용 등 다양한 환경을 지원하는 테스트 프레임워크는 STAF입니다.

병행학습 | 인터페이스 구현 검증 도구

- xUnit : Java(Junit), C++(Cppunit), .Net(Nunit) 등 다양한 언어를 지원하는 단위 테스트 프레임워크
- STAF : 서비스 호출 및 컴포넌트 재사용 등 다양한 환경을 지원하는 테스트 프레임워크
- FitNesse : 웹 기반 테스트케이스 설계, 실행, 결과 확인 등을 지원하는 테스트 프레임워크
- NTAF : FitNesse의 장점인 협업 기능과 STAF의 장점인 재사용 및 확장성을 통합한 NHN(Naver)의 테스트 자동화 프레임워크
- Selenium : 다양한 브라우저 및 개발 언어를 지원하는 웹 애플리케이션 테스트 프레임워크
- watir : Ruby를 사용하는 애플리케이션 테스트 프레임워크

30. SW 패키징 도구 활용 시 고려 사항과 거리가 먼 것은?

① 패키징 시 사용자에게 배포되는 SW이므로 보안을 고려한다.
② 사용자 편의성을 위한 복잡성 및 비효율성 문제를 고려한다.
③ 보안상 단일 기종에서만 사용할 수 있도록 해야 한다.
④ 제품 SW종류에 적합한 암호화 알고리즘을 적용한다.

전문가 조언 | SW 패키징 도구 활용 시에는 사용자의 편의성을 고려하여 다양한 기종과 호환되도록 해야 합니다.

병행학습 | 제품 소프트웨어 패키징 도구

- 제품 소프트웨어의 배포를 위한 패키징 시 디지털 콘텐츠의 지적 재산권의 보호 및 관리 기능과 안전한 유통과 배포를 보장하는 도구이다.
- 불법적인 복제로부터 디지털 콘텐츠의 지적 재산권을 보호해 주는 사용 권한 제어 기술, 패키징 기술, 라이선스 관리, 권한 통제 기술 등을 포함한다.
- 제품 소프트웨어 패키징 도구 활용 시 고려할 사항은 다음과 같다.
  - 패키징 시 사용자에게 배포되는 소프트웨어이므로 내부 콘텐츠에 대한 암호화 및 보안을 고려한다.
  - 다른 여러 콘텐츠 및 단말기 간 DRM(디지털 저작권 관리) 연동을 고려한다.
  - 사용자의 편의성을 위한 복잡성 및 비효율성 문제를 고려한다.
  - 제품 소프트웨어의 종류에 적합한 암호화 알고리즘을 적용한다.

31. 소프트웨어 형상 관리의 의미로 적절한 것은?

① 비용에 관한 사항을 효율적으로 관리하는 것
② 개발 과정의 변경 사항을 관리하는 것
③ 테스트 과정에서 소프트웨어를 통합하는 것
④ 개발 인력을 관리하는 것

전문가 조언 | 형상 관리는 소프트웨어의 개발 과정에서 소프트웨어의 변경 사항을 관리하기 위해 개발된 일련의 활동을 의미합니다.

병행학습 | 형상 관리(SCM; Software Configuration Management)

- 소프트웨어의 개발 과정에서 소프트웨어의 변경 사항을 관리하기 위해 개발된 일련의 활동이다.
- 소프트웨어 변경의 원인을 알아내고 제어하며, 적절히 변경되고 있는지 확인하여 해당 담당자에게 통보한다.
- 형상 관리는 소프트웨어 개발의 전 단계에 적용되는 활동이며, 유지보수 단계에서도 수행된다.
- 형상 관리는 소프트웨어 개발의 전체 비용을 줄이고, 개발 과정의 여러 방해 요인이 최소화되도록 보증하는 것을 목적으로 한다.
- **형상 관리 기능의 종류**
  - 형상 식별 : 형상 관리 대상에 이름과 관리 번호를 부여하고, 계층(Tree) 구조로 구분하여 수정 및 추적이 용이하도록 하는 작업
  - 버전 제어 : 소프트웨어 업그레이드나 유지 보수 과정에서 생성된 다른 버전의 형상 항목을 관리하고, 이를 위해 특정 절차와 도구(Tool)를 결합시키는 작업
  - 형상 통제(변경 관리) : 식별된 형상 항목에 대한 변경 요구를 검토하여 현재의 기준선(Base Line)이 잘 반영될 수 있도록 조정하는 작업
  - 형상 감사 : 기준선의 무결성을 평가하기 위해 확인, 검증, 검열 과정을 통해 공식적으로 승인하는 작업
  - 형상 기록(상태 보고) : 형상의 식별, 통제, 감사 작업의 결과를 기록 · 관리하고 보고서를 작성하는 작업

32. White Box Testing에 대한 설명으로 옳지 않은 것은?

① Base Path Testing, Boundary Value Analysis가 대표적인 기법이다.
② Source Code의 모든 문장을 한 번 이상 수행함으로써 진행 된다.
③ 모듈 안의 작동을 직접 관찰할 수 있다.
④ 산출물의 각 기능별로 적절한 프로그램의 제어구조에 따라 선택, 반복 등의 부분들을 수행함으로써 논리적 경로를 점검 한다.

전문가 조언 | 경계값 분석(Boundary Value Analysis)은 블랙박스 테스트의 종류에 해당합니다.

병행학습 |

블랙박스 테스트(Black Box Test)

- 소프트웨어가 수행할 특정 기능을 알기 위해서 각 기능이 완전히 작동되는 것을 입증하는 검사로서, 기능 검사라고도 한다.
- 부정확하거나 누락된 기능, 인터페이스 오류, 자료 구조나 외부 데이터베이스 접근에 따른 오류, 행위나 성능 오류, 초기화와 종료 오류 등을 발견하기 위해 사용되며 테스트 과정의 후반부에 적용된다.
- 소프트웨어 산물의 각 기능별로 적절한 정보 영역(입 · 출력)을 정하여 적합한 입력에 대한 출력의 정확성을 점검한다.
- **종류** : 동치 분할 검사(Equivalence Partitioning Testing), 경계 값 분석(Boundary Value Analysis), 원인-효과 그래프 검사(Cause-Effect Graphing Testing), 오류 예측 검사(Error Guessing), 비교 검사(Comparison Testing) 등

화이트박스 테스트(White Box Test)

- 모듈의 원시 코드를 오픈시킨 상태에서 원시 코드의 논리적인 모든 경로를 검사하여 검사 사례를 설계하는 방법이다.
- 설계된 절차에 초점을 둔 구조적 테스트로, 프로시저(절차) 설계의 제어 구조를 사용하여 검사 사례를 설계하며, 테스트 과정의 초기에 적용된다.
- 모듈 안의 작동을 직접 관찰한다.
- 원시 코드(모듈)의 모든 문장을 한 번 이상 수행함으로써 수행된다.
- 프로그램의 제어 구조에 따라 선택, 반복 등의 분기점 부분들을 수행함으로써 논리적 경로를 제어한다.
- 각 조건에서의 참과 거짓의 모든 논리적 결정이 적어도 한 번 이상 실행된다.
- **종류** : 기초 경로 검사, 제어 구조 검사(조건 검사(Condition Testing), 루프 검사(Loop Testing), 데이터 흐름 검사(Data Flow Testing)) 등

33. 외계인 코드(Alien Code)에 대한 설명으로 옳은 것은?

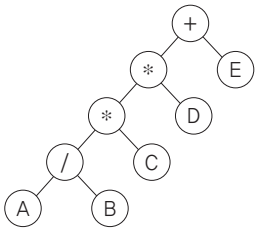
① 프로그램의 로직이 복잡하여 이해하기 어려운 프로그램을 의미한다.
② 아주 오래되거나 참고문서 또는 개발자가 없어 유지보수 작업이 어려운 프로그램을 의미한다.
③ 오류가 없어 디버깅 과정이 필요 없는 프로그램을 의미한다.
④ 사용자가 직접 작성한 프로그램을 의미한다.

전문가 조언 | 아주 오래되거나 참고문서 또는 개발자가 없어 유지보수 작업이 어려운 프로그램을 외계인 코드(Alien Code)라고 합니다.

병행학습 | 외계인 코드(Alien Code)

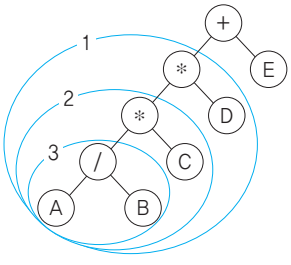
- 아주 오래 전에 개발되어 유지보수 작업이 매우 어려운 프로그램을 의미한다.
- 일반적으로 15년 전 또는 그 전에 개발된 프로그램을 의미하며, 문서화(Documentation)를 철저하게 해두면 방지할 수 있다.

34. 다음 트리를 전위 순회(Preorder Traversal)한 결과는?



① + * A B / * C D E
② A B / C * D * E +
③ A / B * C * D + E
④ + * * / A B C D E

전문가 조언 | 먼저 서브트리를 하나의 노드로 생각할 수 있도록 서브트리 단위로 묶습니다.



❶ Preorder는 Root → Left → Right이므로 +1E입니다.
❷ 1은 *2D이므로 +*2DE입니다.
❸ 2는 *3C이므로 +**3CDE입니다.
❹ 3은 /AB이므로 +**/ABCDE입니다.

35. 알고리즘 시간 복잡도 O(1)이 의미하는 것은?

① 컴퓨터 처리가 불가
② 알고리즘 입력 데이터 수가 한 개
③ 알고리즘 수행시간이 입력 데이터 수와 관계없이 일정
④ 알고리즘 길이가 입력 데이터보다 작음

전문가 조언 | O(1)은 빅오 표기법의 시간 복잡도를 표기하는 방법의 하나로, 입력 데이터 수에 관계없이 일정하게 문제 해결에 하나의 단계만을 거친다는 것을 의미합니다.

병행학습 | 빅오 표기법(Big-O Notation)

- 알고리즘의 실행시간이 최악일 때를 표기하는 방법으로, 신뢰성이 떨어지는 오메가 표기법이나 평가하기 까다로운 세타 표기법에 비해 성능을 예측하기 용이하여 주로 사용되는 표기법이다.
- O(1) : 입력값(n)에 관계 없이 일정하게 문제 해결에 하나의 단계만을 거침
  예 스택의 삽입(Push), 삭제(Pop)
- $O(\log_2 n)$: 문제 해결에 필요한 단계가 입력값(n) 또는 조건에 의해 감소함
  예 이진 트리(Binary Tree), 이진 검색(Binary Search)

- O(n) : 문제 해결에 필요한 단계가 입력값(n)과 1:1의 관계를 가짐
  예 for문
- O(nlog$_2$n) : 문제 해결에 필요한 단계가 n(log$_2$n)번만큼 수행됨
  예 힙 정렬(Heap Sort), 2-Way 합병 정렬(Merge Sort)
- O(n^2) : 문제 해결에 필요한 단계가 입력값(n)의 제곱만큼 수행됨
  예 삽입 정렬(Insertion Sort), 쉘 정렬(Shell Sort), 선택 정렬(Selection Sort), 버블 정렬(Bubble Sort), 퀵 정렬(Quick Sort)
- O(2^n) : 문제 해결에 필요한 단계가 2의 입력값(n) 제곱만큼 수행됨
  예 피보나치 수열(Fibonacci Sequence)

36. 정렬된 N개의 데이터를 처리하는데 O(Nlog$_2$N)의 시간이 소요되는 정렬 알고리즘은?

① 선택 정렬
② 삽입 정렬
③ 버블 정렬
④ 합병 정렬

전문가 조언 | 정렬된 N개의 데이터를 처리하는데 O(Nlog$_2$N)의 시간이 소요되는 정렬 알고리즘에는 힙 정렬과 2-Way 합병 정렬이 있습니다. ①, ②, ③번 정렬의 시간 복잡도는 O(n^2)입니다.

37. ISO/IEC 9126의 소프트웨어 품질 특성 중 기능성(Functionality)의 하위 특성으로 옳지 않은 것은?

① 학습성
② 적합성
③ 정확성
④ 보안성

전문가 조언 | 학습성은 ISO/IEC 9126의 소프트웨어 품질 특성 중 사용성의 하위 특성입니다.

38. EAI(Enterprise Application Integration)의 구축 유형으로 옳지 않은 것은?

① Point-to-Point
② Hub & Spoke
③ Message Bus
④ Tree

전문가 조언 | EAI의 구축 유형에는 Point-to-Point, Hub & Spoke, Message Bus(ESB), Hybrid가 있습니다.

병행학습 | EAI(Enterprise Application Integration)

- 기업 내 각종 애플리케이션 및 플랫폼 간의 정보 전달, 연계, 통합 등 상호 연동이 가능하게 해주는 솔루션이다.
- 비즈니스 간 통합 및 연계성을 증대시켜 효율성 및 각 시스템 간의 확정성(Determinacy)을 높여 준다.
- EAI의 구축 유형
  - Point-to-Point : 가장 기본적인 애플리케이션 통합 방식으로, 애플리케이션을 1 : 1로 연결하며 변경 및 재사용이 어려움
  - Hub & Spoke : 단일 접점인 허브 시스템을 통해 데이터를 전송하는 중앙 집중형 방식으로, 확장 및 유지 보수가 용이하지만 허브 장애 발생 시 시스템 전체에 영향을 미침
  - Message Bus(ESB 방식) : 애플리케이션 사이에 미들웨어를 두어 처리하는 방식으로, 확장성이 뛰어나며 대용량 처리가 가능함
  - Hybrid : Hub & Spoke와 Message Bus의 혼합 방식으로, 그룹 내에서는 Hub & Spoke 방식을, 그룹 간에는 Message Bus 방식을 사용함

39. 소스 코드 품질 분석 도구 중 정적 분석 도구가 아닌 것은?

① pmd
② cppcheck
③ valMeter
④ checkstyle

전문가 조언 | 소스 코드 품질 분석 도구는 정적 분석 도구와 동적 분석 도구로 나뉘며, 정적 분석 도구에는 pmd, cppcheck, SonarQube, checkstyle, ccm, cobertura 등이 있고, 동적 분석 도구에는 Avalanche, Valgrind 등이 있습니다.

병행학습 | 소스 코드 품질 분석 도구

- 소스 코드의 코딩 스타일, 코드에 설정된 코딩 표준, 코드의 복잡도, 코드에 존재하는 메모리 누수 현상, 스레드 결함 등을 발견하기 위해 사용하는 분석 도구로, 크게 정적 분석 도구와 동적 분석 도구로 나뉜다.
- 정적 분석 도구
  - 작성한 소스 코드를 실행하지 않고 코딩 표준이나 코딩 스타일, 결함 등을 확인하는 코드 분석 도구이다.
  - 비교적 애플리케이션 개발 초기의 결함을 찾는데 사용되고, 개발 완료 시점에서는 개발된 소스 코드의 품질을 검증하는 차원에서 사용된다.
  - 동적 분석 도구로는 발견하기 어려운 결함을 찾아내고, 소스 코드에서 코딩의 복잡도, 모델 의존성, 불일치성 등을 분석할 수 있다.
  - 종류 : pmd, cppcheck, SonarQube, checkstyle, ccm, cobertura 등
- 동적 분석 도구
  - 작성한 소스 코드를 실행하여 코드에 존재하는 메모리 누수, 스레드 결함 등을 분석하는 도구이다.
  - 종류 : Avalanche, Valgrind 등

정답 : 33.② 34.④ 35.③ 36.④ 37.① 38.④ 39.③

40. 반정규화(Denormalization) 유형 중 중복 테이블을 추가하는 방법에 해당하지 않는 것은?

① 빌드 테이블의 추가
② 집계 테이블의 추가
③ 진행 테이블의 추가
④ 특정 부분만을 포함하는 테이블 추가

전문가 조언 | 반정규화에서 중복 테이블을 추가하는 방법에는 집계 테이블의 추가, 진행 테이블의 추가, 특정 부분만을 포함하는 테이블의 추가가 있습니다.

병행학습 | **반정규화(Denormalization)**

개념 및 특징

- 시스템의 성능 향상, 개발 및 운영의 편의성 등을 위해 정규화된 데이터 모델을 통합, 중복, 분리하는 과정으로, 의도적으로 정규화 원칙을 위배하는 행위이다.
- 반정규화를 수행하면 시스템의 성능이 향상되고 관리 효율성은 증가하지만 데이터의 일관성 및 정합성이 저하될 수 있다.
- 과도한 반정규화는 오히려 성능을 저하시킬 수 있다.
- 반정규화를 위해서는 사전에 데이터의 일관성과 무결성을 우선으로 할지, 데이터베이스의 성능과 단순화를 우선으로 할지를 결정해야 한다.

반정규화 방법

- **테이블 통합** : 두 개의 테이블이 조인(Join)되는 경우가 많아 하나의 테이블로 합쳐 사용하는 것이 성능 향상에 도움이 될 경우 수행함
- **테이블 분할**
  - 수평 분할(Horizontal Partitioning) : 레코드(Record)를 기준으로 테이블을 분할하는 것으로, 레코드별로 사용 빈도의 차이가 큰 경우 사용 빈도에 따라 테이블을 분할함
  - 수직 분할(Vertical Partitioning) : 하나의 테이블에 속성이 너무 많을 경우 속성을 기준으로 테이블을 분할함
- **중복 테이블 추가**
  - 여러 테이블에서 데이터를 추출해서 사용해야 하거나 다른 서버에 저장된 테이블을 이용해야 하는 경우 중복 테이블을 추가하여 작업의 효율성을 향상시킬 수 있음
  - 집계 테이블의 추가 : 집계 데이터를 위한 테이블을 생성하고, 각 원본 테이블에 트리거(Trigger)를 설정하여 사용하는 것으로, 트리거의 오버헤드(Overhead)에 유의해야 함
  - 진행 테이블의 추가 : 이력 관리 등의 목적으로 추가하는 테이블로, 적절한 데이터 양의 유지와 활용도를 높이기 위해 기본키를 적절히 설정함
  - 특정 부분만을 포함하는 테이블의 추가 : 데이터가 많은 테이블의 특정 부분만을 사용하는 경우 해당 부분만으로 새로운 테이블을 생성함
- **중복 속성 추가** : 조인해서 데이터를 처리할 때 데이터를 조회하는 경로를 단축하기 위해 자주 사용하는 속성을 하나 더 추가하는 것

440203

3과목 데이터베이스 구축

41. SQL의 분류 중 DDL에 해당하지 않는 것은?

① UPDATE ② ALTER
③ DROP ④ CREATE

전문가 조언 | DDL(데이터 정의어)의 3가지 명령어는 CREATE, ALTER, DROP입니다. UPDATE는 DML(데이터 조작어)의 명령어입니다.

병행학습 |

- **DML(데이터 조작어)** : SELECT, INSERT, UPDATE, DELETE
- **DCL(데이터 제어어)** : COMMIT, ROLLBACK, GRANT, REVOKE

42. STUDENT 테이블에 독일어과 학생 50명, 중국어과 학생 30명, 영어영문학과 학생 50명의 정보가 저장되어 있을 때, 다음 두 SQL문의 실행 결과 튜플 수는? (단, DEPT 컬럼은 학과명)

ⓐ SELECT DEPT FROM STUDENT;
ⓑ SELECT DISTINCT DEPT FROM STUDENT;

① ⓐ 3, ⓑ 3 ② ⓐ 50, ⓑ 3
③ ⓐ 130, ⓑ 3 ④ ⓐ 130, ⓑ 130

전문가 조언 | ⓐ STUDENT 테이블에서 DEPT를 검색합니다. 총 130개의 튜플이 들어 있고 검색 조건이 없으므로 튜플의 수는 130개입니다.
ⓑ STUDENT 테이블에서 DEPT를 검색하는 데 중복된 결과는 처음의 한 개만 검색에 포함시킵니다. 독일어과 50개 튜플의 DEPT 속성의 값이 같으므로 1개, 중국어과 30개 튜플의 DEPT 속성의 값이 같으므로 1개, 영어영문학과 50개 튜플의 DEPT 속성의 값이 같으므로 1개를 검색에 포함시키므로 총 3개의 튜플이 검색됩니다.

43. 다음 두 릴레이션에서 외래키로 사용된 것은? (단, 밑줄 친 속성은 기본키이다.)

과목(<u>과목번호</u>, 과목명)
수강(<u>수강번호</u>, 학번, 과목번호, 학기)

① 수강번호 ② 과목번호
③ 학번 ④ 과목명

전문가 조언 | 두 릴레이션에 공통으로 존재하는 속성명은 '과목번호'입니다. 〈과목〉 릴레이션의 '과목번호'는 기본키 속성으로 동일한 속성값이 존재할 수 없고, 〈수강〉 릴레이션의 '과목번호'는 일반 속성으로 여러 속성값이 존재할 수 있으므로 〈수강〉 릴레이션의 '과목번호' 속성이 〈과목〉 릴레이션의 기본키 속성 '과목번호'를 참조하는 외래키가 됩니다.

44. 정규화 과정 중 1NF에서 2NF가 되기 위한 조건은?

① 1NF를 만족하고 모든 도메인이 원자값이어야 한다.
② 1NF를 만족하고, 키가 아닌 모든 애트리뷰트들이 기본키에 이행적으로 함수 종속되지 않아야 한다.
③ 1NF를 만족하고 다치 종속이 제거되어야 한다.
④ 1NF를 만족하고 키가 아닌 모든 속성이 기본키에 대하여 완전 함수적 종속 관계를 만족해야 한다.

전문가 조언 | 제2정규형(2NF)이 되기 위해서는 제1정규형(1NF)를 만족하고, 기본키가 아닌 모든 속성이 기본키에 대하여 완전 함수적 종속을 만족해야 합니다.

병행학습 | 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

정규화 단계 암기 요령

두부를 좋아하는 정규화가 두부가게에 가서 가게에 있는 두부를 다 달라고 말하니 주인이 깜짝 놀라며 말했다.

두부이걸다줘? ≒ 도부이결다조

도메인이 원자값
부분적 함수 종속 제거
이행적 함수 종속 제거
결정자이면서 후보키가 아닌 것 제거
다치 종속 제거
조인 종속성 이용

45. 데이터 무결성 제약조건 중 "개체 무결성 제약" 조건에 대한 설명으로 맞는 것은?

① 릴레이션 내의 튜플들이 각 속성의 도메인에 지정된 값만을 가져야 한다.
② 기본키에 속해 있는 애트리뷰트는 널 값이나 중복 값을 가질 수 없다.
③ 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다.
④ 외래키 값은 참조 릴레이션의 기본키 값과 동일해야 한다.

전문가 조언 | 개체 무결성 제약 조건은 기본 테이블의 기본키를 구성하는 어떤 속성도 Null 값이나 중복값을 가질 수 없다는 규정입니다. ①번은 도메인 무결성, ③, ④번은 참조 무결성에 대한 설명입니다.

병행학습 | 무결성 제약 조건

- 데이터베이스에 저장된 데이터 값과 그것이 표현하는 현실 세계의 실제값이 일치하는 정확성을 의미한다.
- 무결성 제약 조건은 데이터베이스에 들어 있는 데이터의 정확성을 보장하기 위해 부정확한 자료가 데이터베이스 내에 저장되는 것을 방지하기 위한 제약 조건을 말한다.
- 무결성의 종류에는 개체 무결성, 도메인 무결성, 참조 무결성, 사용자 정의 무결성 등이 있다.
- **개체 무결성(Entity Integrity, 실체 무결성)** : 기본 테이블의 기본키를 구성하는 어떤 속성도 Null 값이나 중복값을 가질 수 없다는 규정
- **도메인 무결성(Domain Integrity, 영역 무결성)** : 주어진 속성 값이 정의된 도메인에 속한 값이어야 한다는 규정
- **참조 무결성(Referential Integrity)** : 외래키 값은 Null이거나 참조 릴레이션의 기본키 값과 동일해야 하고, 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다는 규정
- **사용자 정의 무결성(User-Defined Integrity)** : 속성 값들이 사용자가 정의한 제약 조건에 만족해야 한다는 규정

46. 이행적 함수 종속 관계를 의미하는 것은?

① A → B이고 B → C일 때, A → C를 만족하는 관계
② A → B이고 B → C일 때, C → A를 만족하는 관계
③ A → B이고 B → C일 때, B → A를 만족하는 관계
④ A → B이고 B → C일 때, C → B를 만족하는 관계

전문가 조언 | 이행적 함수 종속은 A → B이고 B → C일 때, A → C를 만족하는 관계를 의미합니다.

47. DML에 해당하는 SQL 명령으로만 나열된 것은?

① DELETE, UPDATE, CREATE, ALTER
② INSERT, DELETE, UPDATE, DROP
③ SELECT, INSERT, DELETE, UPDATE
④ SELECT, INSERT, DELETE, ALTER

전문가 조언 | DML(데이터 조작어)의 4가지 명령어는 SELECT, INSERT, DELETE, UPDATE입니다.

병행학습 |

- **DDL(데이터 정의어)** : CREATE, ALTER, DROP
- **DCL(데이터 제어어)** : COMMIT, ROLLBACK, GRANT, REVOKE

48. 데이터베이스 시스템에서 삽입, 갱신, 삭제 등의 이벤트가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL은?

① 트리거(Trigger) ② 무결성(Integrity)
③ 잠금(Lock) ④ 복귀(Rollback)

전문가 조언 | 문제에 제시된 내용은 트리거(Trigger)에 대한 설명입니다.

병행학습 | **트리거(Trigger)**

- 데이터베이스 시스템에서 데이터의 삽입(Insert), 갱신(Update), 삭제(Delete) 등의 이벤트(Event)가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL이다.
- 트리거는 데이터베이스에 저장되며, 데이터 변경 및 무결성 유지, 로그 메시지 출력 등의 목적으로 사용된다.
- 트리거의 구문에는 DCL(데이터 제어어)을 사용할 수 없으며, DCL이 포함된 프로시저나 함수를 호출하는 경우에도 오류가 발생한다.
- 트리거에 오류가 있는 경우 트리거가 처리하는 데이터에도 영향을 미치므로 트리거를 생성할 때 세심한 주의가 필요하다.

49. 데이터베이스의 논리적 설계(Logical Design) 단계에서 수행 하는 작업이 아닌 것은?

① 레코드 집중의 분석 및 설계
② 논리적 데이터베이스 구조로 매핑(mapping)
③ 트랜잭션 인터페이스 설계
④ 스키마의 평가 및 정제

전문가 조언 | 레코드 집중의 분석은 물리적 설계 단계에서 수행하는 작업입니다.

병행학습 |

논리적 설계(데이터 모델링)

- 논리적 설계 단계란 현실 세계에서 발생하는 자료를 컴퓨터가 이해하고 처리할 수 있는 물리적 저장장치에 저장할 수 있도록 변환하기 위해 특정 DBMS가 지원하는 논리적 자료 구조로 변환(매핑, mapping)시키는 과정이다.
- 개념 세계의 데이터를 필드로 기술된 데이터 타입과 이 데이터 타입들 간의 관계로 표현되는 논리적 구조의 데이터로 모델화한다.
- 개념적 설계가 개념 스키마를 설계하는 단계라면 논리적 설계에서는 개념 스키마를 평가 및 정제하고 DBMS에 따라 서로 다른 논리적 스키마를 설계하는 단계이다.
- 트랜잭션의 인터페이스를 설계한다.
- 관계형 데이터베이스라면 테이블을 설계하는 단계이다.

물리적 설계(데이터 구조화)

- 물리적 설계란 논리적 설계 단계에서 논리적 구조로 표현된 데이터를 디스크 등의 물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 과정이다.
- 물리적 설계 단계에서는 다양한 데이터베이스 응용에 대해 처리 성능을 얻기 위해 데이터베이스 파일의 저장 구조 및 액세스 경로를 결정한다.
- 저장 레코드의 형식, 순서, 접근 경로와 같은 정보를 사용하여 데이터가 컴퓨터에 저장되는 방법을 묘사한다.
- 물리적 설계 단계에 꼭 포함되어야 할 것은 저장 레코드의 양식 설계, 레코드 집중(Record Clustering)의 분석 및 설계, 접근 경로 설계 등이다.

50. E-R 모델의 표현 방법으로 옳지 않은 것은?

① 개체 타입 : 사각형 ② 관계 타입 : 마름모
③ 속성 : 오각형 ④ 연결 : 선

전문가 조언 | E-R 모델에서 속성은 타원으로 표현합니다.

병행학습 | **E-R 도형**

사각형	개체(Entity) 타입
다이아몬드(마름모)	관계(Relationship) 타입
타원	속성(Attribute)
이중 타원	다중값 속성(복합 속성)
밑줄 타원	기본 키 속성
복수 타원	복합 속성
선, 링크	개체 타입과 속성을 연결

51. 병행제어의 로킹(Locking) 단위에 대한 설명으로 옳지 않은 것은?

① 데이터베이스, 파일, 레코드 등은 로킹 단위가 될 수 있다.
② 로킹 단위가 작아지면 로킹 오버헤드가 감소한다.
③ 로킹 단위가 작아지면 데이터베이스 공유도가 증가한다.
④ 한꺼번에 로킹할 수 있는 객체의 크기를 로킹 단위라고 한다.

전문가 조언 | 로킹 단위가 작아지면 로크 수가 많아 관리하기 복잡해 오버헤드가 증가합니다.

병행학습 | **로킹 단위(Locking Granularity)**

- 로킹 단위는 병행제어에서 한꺼번에 로킹할 수 있는 객체의 크기를 의미한다.
- 데이터베이스, 파일, 레코드, 필드 등은 로킹 단위가 될 수 있다.
- 로킹 단위가 크면 로크 수가 작아 관리하기 쉽지만 병행성 수준이 낮아지고 데이터베이스 공유도가 저하된다.
- 로킹 단위가 작으면 로크 수가 많아 관리하기 복잡해 오버헤드가 증가하지만 병행성 수준이 높아지고, 데이터베이스 공유도가 증가한다.

52. 뷰(View)에 대한 설명으로 옳지 않은 것은?

① 뷰는 CREATE 문을 사용하여 정의한다.
② 뷰는 데이터의 논리적 독립성을 제공한다.
③ 뷰를 제거할 때에는 DROP 문을 사용한다.
④ 뷰는 저장장치 내에 물리적으로 존재한다.

전문가 조언 | 뷰(View)는 저장장치 내에 물리적으로 존재하지 않는 가상 테이블입니다.

병행학습 | 뷰(View)

- 뷰는 사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블이다.
- 뷰는 저장장치 내에 물리적으로 존재하지 않지만, 사용자에게는 있는 것처럼 간주된다.
- 뷰는 기본 테이블로부터 유도된 테이블이기 때문에 기본 테이블과 같은 형태의 구조를 사용하며, 조작도 기본 테이블과 거의 같다.
- 뷰는 가상 테이블이기 때문에 물리적으로 구현되어 있지 않다.
- 데이터의 논리적 독립성이 어느 정도 보장된다.
- 필요한 데이터만 뷰로 정의해서 처리할 수 있기 때문에 관리가 용이하고 명령문이 간단해진다.
- 뷰를 통해서만 데이터에 접근하게 하면 뷰에 나타나지 않는 데이터를 안전하게 보호하는 효율적인 기법으로 사용할 수 있다.
- 기본 테이블의 기본키를 포함한 속성(열) 집합으로 뷰를 구성해야만 삽입, 삭제, 갱신 연산이 가능하다.
- 정의된 뷰는 다른 뷰의 정의에 기초가 될 수 있다.
- 하나의 뷰를 삭제하면 그 뷰를 기초로 정의된 다른 뷰도 자동으로 삭제된다.

53. 하나의 애트리뷰트가 가질 수 있는 원자값들의 집합을 의미하는 것은?

① 도메인 ② 튜플
③ 엔티티 ④ 다형성

전문가 조언 | 하나의 애트리뷰트가 가질 수 있는 원자값들의 집합을 의미하는 것은 도메인(Domain)입니다.

병행학습 | 관계형 데이터베이스 관련 용어

- 튜플(Tupel)
  - 튜플은 릴레이션을 구성하는 각각의 행을 말한다.
  - 튜플은 속성의 모임으로 구성된다.
  - 파일 구조에서 레코드와 같은 의미이다.
  - 튜플의 수를 카디널리티(Cardinality) 또는 기수, 대응수라고 한다.
- 속성(Attribute)
  - 속성은 데이터베이스를 구성하는 가장 작은 논리적 단위이다.
  - 파일 구조상의 데이터 항목 또는 데이터 필드에 해당된다.
  - 속성은 개체의 특성을 기술한다.
  - 속성의 수를 디그리(Degree) 또는 차수라고 한다.
- 도메인(Domain)
  - 도메인은 하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자(Atomic)값들의 집합이다.
  - 도메인은 실제 애트리뷰트 값이 나타날 때 그 값의 합법 여부를 시스템이 검사하는데에도 이용된다.

  예 성별 애트리뷰트의 도메인은 '남'과 '여'로, 그 외의 값은 입력될 수 없다.

54. 관계대수 연산에서 두 릴레이션이 공통으로 가지고 있는 속성을 이용하여 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 연산은?

① ⋈ ② ⊃ ③ π ④ σ

전문가 조언 | 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 Join 연산자의 기호는 ⋈입니다.

병행학습 | 순수 관계 연산자

- Select
  - 릴레이션에 존재하는 튜플 중에서 선택 조건을 만족하는 튜플의 부분집합을 구하여 새로운 릴레이션을 만드는 연산이다.
  - 릴레이션의 행(가로)에 해당하는 튜플을 구하는 것이므로 수평 연산이라고도 한다.
  - 연산자의 기호는 시그마(σ)를 사용한다.
- Project
  - 주어진 릴레이션에서 속성 리스트(Attribute List)에 제시된 속성 값만을 추출하여 새로운 릴레이션을 만드는 연산이다. 단 연산 결과에 중복이 발생하면 중복이 제거된다.
  - 릴레이션의 열(세로)에 해당하는 Attribute를 추출하는 것이므로 수직 연산자라고도 한다.
  - 연산자의 기호는 파이(π)를 사용한다.
- Join
  - 공통 속성을 중심으로 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 연산이다.
  - 연산자의 기호는 ⋈를 사용한다.
- 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.
  - 연산자의 기호는 ÷를 사용한다.

55. 분산 데이터베이스 목표 중 "데이터베이스의 분산된 물리적 환경에서 특정 지역의 컴퓨터 시스템이나 네트워크에 장애가 발생해도 데이터 무결성이 보장된다."는 것과 관계있는 것은?

① 장애 투명성 ② 병행 투명성
③ 위치 투명성 ④ 중복 투명성

전문가 조언 | 문제에 제시된 내용과 관계있는 분산 데이터베이스 목표는 장애 투명성(Failure Transparency)입니다.

병행학습 | 분산 데이터베이스(Distributed Database)

- 분산 데이터베이스는 논리적으로는 같은 시스템에 속하지만 물리적으로는 컴퓨터 네트워크를 통해 분산되어 있는 데이터베이스로 목표는 다음과 같다.
- **위치 투명성**(Location Transparency) : 접근하려는 데이터베이스의 실제 위치를 알 필요 없이 단지 데이터베이스의 논리적인 명칭만으로 접근할 수 있음
- **중복 투명성**(Replication Transparency) : 동일한 데이터가 여러 곳에 중복되어 있더라도 사용자는 마치 하나의 데이터만 존재하는 것처럼 사용할 수 있고, 시스템은 자동으로 여러 데이터에 대한 작업을 수행함

- 병행 투명성(Concurrency Transparency) : 분산 데이터베이스와 관련된 다수의 트랜잭션들이 동시에 실행되더라도 그 트랜잭션들의 수행 결과는 서로 영향을 받지 않음
- 장애 투명성(Failure Transparency) : 트랜잭션, DBMS, 네트워크, 컴퓨터 장애에도 불구하고 트랜잭션은 정확하게 수행됨

56. 다음 설명의 ( ) 안에 들어갈 내용으로 적합한 것은?

> 후보키는 릴레이션에 있는 모든 튜플에 대해 유일성과 ( )을 모두 만족시켜야 한다.

① 중복성 ② 최소성
③ 참조성 ④ 동일성

전문가 조언 | 후보키는 릴레이션에 있는 모든 튜플에 대해 유일성과 최소성을 모두 모두 만족시켜야 합니다.

병행학습 | **키(Key)의 개념 및 종류**

- 키(Key)는 데이터베이스에서 조건에 만족하는 튜플을 찾거나 순서대로 정렬할 때 기준이 되는 속성이다.
- **슈퍼키**(Super Key) : 한 릴레이션 내에 있는 속성들의 집합으로 구성된 키로, 릴레이션을 구성하는 모든 튜플에 대해 유일성(Unique)은 만족하지만, 최소성(Minimality)은 만족하지 못함
- **후보키**(Candidate Key) : 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용되는 속성들의 부분집합으로, 유일성과 최소성을 모두 만족함
- **기본키**(Primary Key) : 후보키 중에서 특별히 선정된 키로 중복된 값과 NULL 값을 가질 수 없음
- **대체키**(Alternate Key) : 후보키 중에서 선정된 기본키를 제외한 나머지 후보키를 의미함
- **외래키**(Foreign Key) : 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미하며, 릴레이션 간의 관계를 표현할 때 사용함

57. 다음 SQL문의 실행 결과는?

```
SELECT 가격 FROM 도서가격
WHERE 책번호=(SELECT 책번호 FROM 도서 WHERE 책명
='자료구조');
```

[도서]

책번호	책명
111	운영체제
222	자료구조
333	컴퓨터구조

[도서가격]

책번호	가격
111	20,000
222	25,000
333	10,000
444	15,000

① 10,000 ② 15,000
③ 20,000 ④ 25,000

전문가 조언 | 문제의 질의문은 하위 질의가 있는 질의문입니다. 먼저 WHERE 조건에 지정된 하위 질의의 SELECT문을 검색합니다. 그리고 검색 결과를 본 질의의 조건에 있는 '책번호' 속성과 비교합니다.

❶ SELECT 책번호 FROM 도서 WHERE 책명 = '자료구조'; : '도서' 테이블에서 '책명' 속성의 값이 '자료구조'와 같은 레코드의 '책번호' 속성의 값을 검색합니다. 결과는 '222'입니다.

❷ SELECT 가격 FROM 도서가격 WHERE 책번호 = '222'; : '도서가격' 테이블에서 '책번호' 속성의 값이 '222'와 같은 레코드의 '가격' 속성의 값을 검색합니다. 결과는 '25,000'입니다.

58. 데이터 제어 언어(DCL)의 기능으로 옳지 않은 것은?

① 데이터 보안
② 논리적, 물리적 데이터 구조 정의
③ 무결성 유지
④ 병행수행 제어

전문가 조언 | 논리적, 물리적 데이터 구조를 정의하는 것은 데이터 정의 언어(DDL; Data Definition Language)의 기능입니다.

병행학습 |

DDL(Data Define Language, 데이터 정의어)

- SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의하거나 변경 또는 삭제할 때 사용하는 언어이다.
- 논리적 데이터 구조와 물리적 데이터 구조의 사상을 정의한다.
- 데이터베이스 관리자나 데이터베이스 설계자가 사용한다.
- DDL(데이터 정의어)의 세 가지 유형
  - CREATE : SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의함
  - ALTER : TABLE에 대한 정의를 변경하는 데 사용함
  - DROP : SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 삭제함

DML(Data Manipulation Language, 데이터 조작어)

- 데이터베이스 사용자가 응용 프로그램이나 질의어를 통하여 저장된 데이터를 실질적으로 처리하는 데 사용되는 언어이다.
- 데이터베이스 사용자와 데이터베이스 관리 시스템 간의 인터페이스를 제공한다.
- DML(데이터 조작어)의 네 가지 유형
  - SELECT : 테이블에서 조건에 맞는 튜플을 검색함
  - INSERT : 테이블에 새로운 튜플을 삽입함
  - DELETE : 테이블에서 조건에 맞는 튜플을 삭제함
  - UPDATE : 테이블에서 조건에 맞는 튜플의 내용을 변경함

DCL(Data Control Language, 데이터 제어어)

- 데이터의 보안, 무결성, 회복, 병행 수행 제어 등을 정의하는 데 사용되는 언어이다.
- 데이터베이스 관리자가 데이터 관리를 목적으로 사용한다.
- DCL(데이터 제어어)의 종류
  - 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

59. 참조 무결성을 유지하기 위하여 DROP문에서 부모 테이블의 항목 값을 삭제할 경우 자동적으로 자식 테이블의 해당 레코드를 삭제하기 위한 옵션은?

① CLUSTER ② CASCADE
③ SET-NULL ④ RESTRICTED

전문가 조언 | 문제에 제시된 내용은 CASCADE에 대한 설명입니다.

병행학습 |
- CASCADE : 제거할 요소를 참조하는 다른 모든 개체를 함께 제거함. 즉 주 테이블의 데이터 제거 시 각 외래키와 관계를 맺고 있는 모든 데이터를 제거하는 참조 무결성 제약 조건을 설정하기 위해 사용됨
- RESTRICTED : 다른 개체가 제거할 요소를 참조중일 때는 제거를 취소함

60. 트랜잭션의 특성 중 다음 설명에 해당하는 것은?

> 트랜잭션의 연산은 데이터베이스에 모두 반영되든지 아니면 전혀 반영되지 않아야 한다.

① Durability ② Share
③ Consistency ④ Atomicity

전문가 조언 | 문제의 지문에 제시된 트랜잭션의 특성은 원자성(Atomicity)입니다.

병행학습 | **트랜잭션의 특성**
- Atomicity(원자성)
  - 트랜잭션의 연산은 데이터베이스에 모두 반영되도록 완료(Commit)되든지 아니면 전혀 반영되지 않도록 복구(Rollback)되어야 한다.
  - 트랜잭션 내의 모든 명령은 반드시 완벽히 수행되어야 하며, 모두가 완벽히 수행되지 않고 어느 하나라도 오류가 발생하면 트랜잭션 전부가 취소되어야 한다.
- Consistency(일관성)
  - 트랜잭션이 그 실행을 성공적으로 완료하면 언제나 일관성 있는 데이터베이스 상태로 변환한다.
  - 시스템이 가지고 있는 고정 요소는 트랜잭션 수행 전과 트랜잭션 수행 완료 후의 상태가 같아야 한다.
- Isolation(독립성, 격리성, 순차성)
  - 둘 이상의 트랜잭션이 동시에 병행 실행되는 경우 어느 하나의 트랜잭션 실행 중에 다른 트랜잭션의 연산이 끼어들 수 없다.
  - 수행중인 트랜잭션은 완전히 완료될 때까지 다른 트랜잭션에서 수행 결과를 참조할 수 없다.
- Durability(영속성, 지속성) : 성공적으로 완료된 트랜잭션의 결과는 시스템이 고장나더라도 영구적으로 반영되어야 함

440204

4과목 프로그래밍 언어 활용

61. UNIX의 쉘(Shell)에 관한 설명으로 옳지 않은 것은?

① 명령어 해석기이다.
② 시스템과 사용자 간의 인터페이스를 담당한다.
③ 여러 종류의 쉘이 있다.
④ 프로세스, 기억장치, 입출력 관리를 수행한다.

전문가 조언 | 프로세스 관리, 기억장치 관리, 파일 관리 등의 작업을 수행하는 것은 커널(Kernel)입니다.

병행학습 | **UNIX 시스템의 구성**

커널 (Kernel)	• UNIX의 가장 핵심적인 부분 • 하드웨어를 보호(캡슐화)하고, 프로그램들과 하드웨어 간의 인터페이스 역할을 담당함 • 프로세스, 기억장치, 파일, 입 · 출력 관리, 프로세스 간 통신, 데이터 전송 및 변환 등 여러 가지 기능을 수행함 • 컴퓨터 부팅 시 주기억장치에 적재되어 상주하면서 실행됨
쉘 (Shell)	• 사용자의 명령어를 인식하여 프로그램을 호출하고, 명령을 수행하는 명령어 해석기 • 시스템과 사용자 간의 인터페이스를 담당함 • DOS의 COMMAND.COM과 같은 기능을 수행함 • 주기억장치에 상주하지 않고, 명령어가 포함된 파일 형태로 존재하며 보조기억장치에서 교체 처리가 가능함 • 종류 : Bourne Shell, C Shell, Korn Shell 등

62. TCP/IP 프로토콜 중 전송 계층 프로토콜은?

① HTTP ② SMTP
③ FTP ④ TCP

전문가 조언 | TCP는 전송 계층, 나머지는 응용 계층의 프로토콜입니다.

병행학습 | **TCP/IP의 구조**

OSI	TCP/IP	기능
응용 계층 표현 계층 세션 계층	응용 계층	• 응용 프로그램 간의 데이터 송 · 수신을 제공 • TELNET, FTP, SMTP, SNMP, DNS, HTTP 등
전송 계층	전송 계층	• 호스트들 간의 신뢰성 있는 통신을 제공 • TCP, UDP
네트워크 계층	인터넷 계층	• 데이터 전송을 위한 주소 지정, 경로 설정을 제공 • IP, ICMP, IGMP, ARP, RARP
데이터 링크 계층 물리 계층	네트워크 액세스 계층	• 실제 데이터(프레임)를 송 · 수신하는 역할 • Ethernet, IEEE 802, HDLC, X.25, RS-232C, ARQ 등

정답 : 56.② 57.④ 58.② 59.② 60.④ 61.④ 62.④

63. C 언어에서 비트 논리 연산자에 해당하지 않는 것은?

① ^ ② ? ③ & ④ ~

전문가 조언 | C 언어의 비트 연산자에는 &, ^, |, ~, ≪, ≫ 등이 있습니다.

병행학습 | 비트 연산자

비트별(0, 1)로 연산하여 결과를 얻는 연산자이다.

연산자	의미	비고
&	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

64. 시스템에서 모듈 사이의 결합도(Coupling)에 대한 설명으로 옳은 것은?

① 한 모듈 내에 있는 처리 요소들 사이의 기능적인 연관 정도를 나타낸다.
② 결합도가 높으면 시스템 구현 및 유지보수 작업이 쉽다.
③ 모듈 간의 결합도를 약하게 하면 모듈 독립성이 향상된다.
④ 자료결합도는 내용결합도보다 결합도가 높다.

전문가 조언 | 시스템에서 모듈 간의 결합도를 약하게 하면 모듈 독립성이 향상됩니다. 나머지 보기가 틀린 이유는 다음과 같습니다.

① 결합도는 모듈 상호 간에 의존하는 정도를 의미합니다. 한 모듈 내에 있는 처리 요소들 사이의 기능적인 연관 정도를 나타내는 것은 응집도입니다.
② 결합도가 높으면 시스템 구현 및 유지보수 작업이 어렵습니다.
④ 자료 결합도는 내용 결합도보다 결합도가 낮습니다.

병행학습 | 결합도(Coupling)

- 모듈 간에 상호 의존하는 정도 또는 두 모듈 사이의 연관 관계를 의미한다.
- 다양한 결합으로 모듈을 구성할 수 있으나 결합도가 약할수록 품질이 높고, 강할수록 품질이 낮다.
- 결합도가 강하면 시스템 구현 및 유지보수 작업이 어렵다.
- 결합도의 종류(낮음에서 높음순)
  - 자료 결합도(Data Coupling) : 모듈 간의 인터페이스가 자료 요소로만 구성될 때의 결합도
  - 스탬프(검인) 결합도(Stamp Coupling) : 모듈 간의 인터페이스로 배열이나 레코드 등의 자료 구조가 전달될 때의 결합도
  - 제어 결합도(Control Coupling) : 어떤 모듈이 다른 모듈 내부의 논리적인 흐름을 제어하기 위해 제어 신호를 이용하여 통신하거나 제어 요소를 전달하는 결합도
  - 외부 결합도(External Coupling) : 어떤 모듈에서 외부로 선언한 데이터(변수)를 다른 모듈에서 참조할 때의 결합도
  - 공통(공유) 결합도(Common Coupling) : 공유되는 공통 데이터 영역을 여러 모듈이 사용할 때의 결합도
  - 내용 결합도(Content Coupling) : 한 모듈이 다른 모듈의 내부 기능 및 그 내부 자료를 직접 참조하거나 수정할 때의 결합도

65. 스레드(Thread)에 대한 설명으로 옳지 않은 것은?

① 한 개의 프로세스는 여러 개의 스레드를 가질 수 없다.
② 커널 스레드의 경우 운영체제에 의해 스레드를 운용한다.
③ 사용자 스레드의 경우 사용자가 만든 라이브러리를 사용하여 스레드를 운용한다.
④ 스레드를 사용함으로써 하드웨어, 운영체제의 성능과 응용 프로그램의 처리율을 향상시킬 수 있다.

전문가 조언 | 한 개의 프로세스는 여러 개의 스레드를 가질 수 있습니다. 이를 다중 스레드라고 합니다.

병행학습 | 스레드(Thread)

- 프로세스 내에서의 작업 단위로서 시스템의 여러 자원을 할당받아 실행하는 프로그램의 단위이다.
- 하나의 프로세스에 하나의 스레드가 존재하는 경우에는 단일 스레드, 하나 이상의 스레드가 존재하는 경우에는 다중 스레드라고 한다.
- 프로세스의 일부 특성을 갖고 있기 때문에 경량(Light Weight) 프로세스라고도 한다.
- 스레드 기반 시스템에서 스레드는 독립적인 스케줄링의 최소 단위로서 프로세스의 역할을 담당한다.
- 동일 프로세스 환경에서 서로 독립적인 다중 수행이 가능하다.

스레드의 분류

- 사용자 수준의 스레드
  - 사용자가 만든 라이브러리를 사용하여 스레드를 운용한다.
  - 속도는 빠르지만 구현이 어렵다.
- 커널 수준의 스레드
  - 운영체제의 커널에 의해 스레드를 운용한다.
  - 구현이 쉽지만 속도가 느리다.

66. C 언어에서 배열 b[5]의 값은?

```
static int b[9]={1, 2, 3};
```

① 0 ② 1 ③ 2 ④ 3

전문가 조언 | C 언어의 변수 선언 시 앞에 static을 붙이면 정적 변수로 선언됩니다. 정적 변수는 초기화를 생략하면 0으로 자동 초기화 됩니다.

병행학습 | 기억 클래스의 종류

- 자동 변수(Automatic Variable)
  - 함수나 코드의 범위를 한정하는 블록 내에서 선언되는 변수이다.
  - 함수나 블록이 실행되는 동안에만 존재하며 이를 벗어나면 자동으로 소멸된다.
  - 초기화하지 않으면 쓰레기값(Garbage Value)이 저장된다.

- 외부 변수(External Variable)
  - 현재 파일이나 다른 파일에서 선언된 변수나 함수를 참조(reference)하기 위한 변수이다.
  - 외부 변수는 함수 밖에서 선언한다.
  - 함수가 종료된 뒤에도 값이 소멸되지 않는다.
  - 초기화하지 않으면 자동으로 0으로 초기화 된다.
  - 다른 파일에서 선언된 변수를 참조할 경우 초기화 할 수 없다.
- 정적 변수(Static Variable)
  - 함수나 블록 내에서 선언하는 내부 정적 변수와 함수 외부에서 선언하는 외부 정적 변수가 있다.
  - 내부 정적 변수는 선언한 함수나 블록 내에서만 사용할 수 있고, 외부 정적 변수는 모든 함수에서 사용할 수 있다.
  - 두 변수 모두 함수나 블록이 종료된 뒤에도 값이 소멸되지 않는다.
  - 초기화는 변수 선언 시 한 번만 할 수 있으며, 초기화를 생략하면 자동으로 0으로 초기화 된다.
- 레지스터 변수(Register Variable)
  - 메모리가 아닌 CPU 내부의 레지스터에 기억영역을 할당받는 변수이다.
  - 자주 사용되는 변수를 레지스터에 저장하여 처리 속도를 높이기 위해 사용한다.
  - 함수나 블록이 실행되는 동안에만 존재하며 이를 벗어나면 자동으로 소멸된다.
  - 레지스터의 사용 개수는 한정되어 있어 데이터를 저장할 레지스터가 없는 경우 자동 변수로 취급되어 메모리에 할당된다.
  - CPU에 저장되어 메모리 주소를 가질 수 없기 때문에 변수의 주소를 구하는 주소 연산자(&)를 사용할 수 없다.

67. 은행가 알고리즘(Banker's Algorithm)은 교착상태의 해결 방법 중 어떤 기법에 해당하는가?

① Avoidance ② Detection
③ Prevention ④ Recovery

전문가 조언 | 은행가 알고리즘은 교착상태의 해결 방법 중 회피 기법(Avoidance)에 해당합니다.

병행학습 | 교착상태 해결 방법

- 예방 기법(Prevention)
  - 교착상태가 발생하지 않도록 사전에 시스템을 제어하는 방법으로, 교착상태 발생의 네 가지 조건 중에서 어느 하나를 제거(부정)함으로써 수행된다.
  - 상호 배제(Mutual Exclusion) 부정 : 한 번에 여러 개의 프로세스가 공유 자원을 사용할 수 있도록 함
  - 점유 및 대기(Hold and Wait) 부정 : 프로세스가 실행되기 전 필요한 모든 자원을 할당하여 프로세스 대기를 없애거나 자원이 점유되지 않은 상태에서만 자원을 요구하도록 함
  - 비선점(Non-preemption) 부정 : 자원을 점유하고 있는 프로세스가 다른 자원을 요구할 때 점유하고 있는 자원을 반납하고, 요구한 자원을 사용하기 위해 기다리게 함
  - 환형 대기(Circular Wait) 부정 : 자원을 선형 순서로 분류하여 고유 번호를 할당하고, 각 프로세스는 현재 점유한 자원의 고유 번호보다 앞이나 뒤 어느 한쪽 방향으로만 자원을 요구하도록 하는 것
- 회피 기법(Avoidance) : 교착상태가 발생할 가능성을 배제하지 않고 교착상태가 발생하면 적절히 피해나가는 방법으로, 주로 은행원 알고리즘(Banker's Algorithm)이 사용됨
- 발견(Detection) 기법 : 시스템에 교착 상태가 발생했는지 점검하여 교착 상태에 있는 프로세스와 자원을 발견하는 것으로, 자원 할당 그래프 등을 사용함
- 회복(Recovery) 기법 : 교착 상태를 일으킨 프로세스를 종료하거나 교착 상태의 프로세스에 할당된 자원을 선점하여 프로세스나 자원을 회복하는 것

68. IEEE 802.11 워킹 그룹의 무선 LAN 표준화 현황 중 QoS 강화를 위해 MAC 지원 기능을 채택한 것은?

① 802.11a ② 802.11b
③ 802.11g ④ 802.11e

전문가 조언 | QoS 강화를 위해 MAC 지원 기능을 채택한 802.11 버전은 802.11e입니다.

병행학습 | 802.11의 버전

- 802.11(초기 버전) : 2.4GHz 대역 전파와 CSMA/CA 기술을 사용해 최고 2Mbps까지의 전송 속도를 지원함
- 802.11a : 5GHz 대역의 전파를 사용하며, OFDM 기술을 사용해 최고 54Mbps까지의 전송 속도를 지원함
- 802.11b : 802.11 초기 버전의 개선안으로 등장하였으며, 초기 버전의 대역 전파와 기술을 사용해 최고 11Mbps의 전송 속도로 기존에 비해 5배 이상 빠르게 개선되었음
- 802.11e : 802.11의 부가 기능 표준으로, QoS 기능이 지원되도록 하기 위해 매체 접근 제어(MAC) 계층에 해당하는 부분을 수정하였음
- 802.11g : 2.4GHz 대역의 전파를 사용하지만 5GHz 대역의 전파를 사용하는 802.11a와 동일한 최고 54Mbps까지의 전송 속도를 지원함
- 802.11n : 2.4GHz 대역과 5GHz 대역을 사용하는 규격으로, 최고 600Mbps까지의 전송 속도를 지원함

69. TCP/IP 네트워크에서 IP 주소를 MAC 주소로 변환하는 프로토콜은?

① UDP ② ARP
③ TCP ④ ICMP

전문가 조언 | TCP/IP 네트워크에서 IP 주소를 MAC 주소로 변환하는 프로토콜은 ARP(Address Resolution Protocol)입니다.

병행학습 |

- UDP(User Datagram Protocol) : 데이터 전송 전에 연결을 설정하지 않는 비연결형 서비스를 제공하고, TCP에 비해 상대적으로 단순한 헤더 구조를 가지므로, 오버헤드가 적음
- TCP(Transmission Control Protocol) : 신뢰성 있는 연결형 서비스를 제공하고, 패킷의 다중화, 순서 제어, 오류 제어, 흐름 제어 기능을 제공함
- ICMP(Internet Control Message Protocol, 인터넷 제어 메시지 프로토콜) : IP와 조합하여 통신중에 발생하는 오류의 처리와 전송 경로 변경 등을 위한 제어 메시지를 관리하는 역할을 하며, 헤더는 8Byte로 구성됨

70. HRN(Highest Response-ratio Next) 스케줄링 방식에 대한 설명으로 옳지 않은 것은?

① 대기 시간이 긴 프로세스일 경우 우선순위가 높아진다.
② SJF 기법을 보완하기 위한 방식이다.
③ 긴 작업과 짧은 작업 간의 지나친 불평등을 해소할 수 있다.
④ 우선순위를 계산하여 그 수치가 가장 낮은 것부터 높은 순으로 우선순위가 부여된다.

전문가 조언 | HRN 스케줄링 방식은 우선순위를 계산하여 그 숫자가 가장 높은 것부터 낮은 순으로 우선순위가 부여됩니다.

병행학습 | HRN(Hightest Response-ratio Next)

- 실행 시간이 긴 프로세스에 불리한 SJF 기법을 보완하기 위한 것으로, 대기 시간과 서비스(실행) 시간을 이용하는 기법이다.
- 우선순위 계산 공식을 이용하여 서비스(실행) 시간이 짧은 프로세스나 대기 시간이 긴 프로세스에게 우선순위를 주어 CPU를 할당한다.
- 서비스 실행 시간이 짧거나 대기 시간이 긴 프로세스일 경우 우선순위가 높아진다.
- 우선순위를 계산하여 그 숫자가 가장 높은 것부터 낮은 순으로 우선순위가 부여된다.
- **우선순위 계산식** : (대기 시간+서비스 시간) / 서비스 시간

71. 교착 상태 발생의 필요 충분 조건이 아닌 것은?

① 상호 배제(Mutual Exclusion)
② 점유와 대기(Hold and Wait)
③ 환형 대기(Circular Wait)
④ 선점(Preemption)

전문가 조언 | 교착 상태 발생의 필요 충분 조건 4가지는 상호 배제, 점유와 대기, 환형 대기, 비전섬입니다.

병행학습 | **교착상태 발생의 필요 충분 조건**

- **상호 배제(Mutual Exclusion)** : 한 번에 한 개의 프로세스만이 공유 자원을 사용할 수 있어야 함
- **점유와 대기(Hold and Wait)** : 최소한 하나의 자원을 점유하고 있으면서 다른 프로세스에 할당되어 사용되고 있는 자원을 추가로 점유하기 위해 대기하는 프로세스가 있어야 함
- **비선점(Non-Preemption)** : 다른 프로세스에 할당된 자원은 사용이 끝날 때까지 강제로 빼앗을 수 없어야 함
- **환형 대기(Circular Wait)** : 공유 자원과 공유 자원을 사용하기 위해 대기하는 프로세스들이 원형으로 구성되어 있어 자신에게 할당된 자원을 점유하면서 앞이나 뒤에 있는 프로세스의 자원을 요구해야 함

72. 다음의 페이지 참조 열(Page Reference)에 대해 페이지 교체 기법으로 선입선출 알고리즘을 사용할 경우 페이지 부재(Page Fault) 횟수는? (단, 할당된 페이지 프레임 수는 3이고, 처음에는 모든 프레임이 비어 있다.)

〈페이지 참조 열〉

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0

① 13 ② 14
③ 15 ④ 20

전문가 조언 | 3개의 페이지를 수용할 수 있는 주기억장치이므로 다음 그림과 같이 3개의 페이지 프레임으로 표현할 수 있습니다.

참조 페이지	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0
페이지 프레임	7	7	7	2	2	2	2	4	4	4	0	0	0	0	0	0	0	7	7
		0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1	0
			1	1	1	1	0	0	0	3	3	3	3	3	2	2	2	2	2
부재 발생	●	●	●	●		●	●	●	●	●	●			●	●			●	●

참조 페이지가 페이지 프레임에 없을 경우 페이지 결함(부재)이 발생됩니다. 초기에는 모든 페이지 프레임이 비어 있으므로 처음 7, 0, 1 페이지 적재 시 페이지 결함이 발생됩니다. 선입선출(FIFO) 기법은 각 페이지가 주기억장치에 적재될 때마다 그때의 시간을 기억시켜 가장 먼저 들어와서 가장 오래 있었던 페이지를 교체하는 기법이므로, 참조 페이지 2를 참조할 때에는 7을 제거한 후 2를 가져오게 됩니다. 이와 같은 방식으로 모든 페이지 요청을 처리하고 나면 총 페이지 결함 발생 수는 14번입니다.

73. C언어에서 사용할 수 없는 변수명은?

① student2019 ② text-color
③ _korea ④ amount

전문가 조언 | 변수명에는 공백이나 *, +, -, / 등의 특수문자를 사용할 수 없습니다.

병행학습 | **변수명 작성 규칙**

- 영문자, 숫자, _(under bar)를 사용할 수 있다.
- 첫 글자는 영문자나 _(under bar)로 시작해야 하며, 숫자는 올 수 없다.
- 글자 수에 제한이 없다.
- 공백이나 *, +, -, / 등의 특수문자를 사용할 수 없다.
- 대 · 소문자를 구분한다.
- 예약어를 변수명으로 사용할 수 없다.
- 변수 선언 시 문장 끝에 반드시 세미콜론(;)을 붙여야 한다.

74. IPv6에 대한 설명으로 틀린 것은?

① 128비트의 주소 공간을 제공한다.
② 인증 및 보안 기능을 포함하고 있다.
③ 패킷 크기가 64Kbyte로 고정되어 있다.
④ IPv6 확장 헤더를 통해 네트워크 기능 확장이 용이하다.

전문가 조언 | IPv6의 패킷에는 크기의 제한이 없습니다. 패킷 크기가 64Kbyte로 고정되어 있는 것은 IPv4입니다.

병행학습 | IPv6(Internet Protocol version 6)

- 현재 사용하고 있는 IP 주소 체계인 IPv4의 주소 부족 문제를 해결하기 위해 개발되었다.
- 16비트씩 8부분, 총 128비트로 구성되어 있다.
- 각 부분을 16진수로 표현하고, 콜론(:)으로 구분한다.
- IPv4에 비해 자료 전송 속도가 빠르고, IPv4와 호환성이 뛰어나다.
- 인증성, 기밀성, 데이터 무결성의 지원으로 보안 문제를 해결할 수 있다.
- IPv6의 주소 체계
  - 유니캐스트(Unicast) : 단일 송신자와 단일 수신자 간의 통신(1:1 통신에 사용)
  - 멀티캐스트(Multicast) : 단일 송신자와 다중 수신자 간의 통신(1:N 통신에 사용)
  - 애니캐스트(Anycast) : 단일 송신자와 가장 가까이 있는 단일 수신자 간의 통신(1:1 통신에 사용)

75. 프로세스 상태의 종류가 아닌 것은?

① Ready ② Running
③ Request ④ Exit

전문가 조언 | 프로세스 상태의 종류에는 제출(Submit), 접수(Hold), 준비(Ready), 실행(Run), 대기(Wait), 종료(Terminated, Exit) 등이 있습니다.

병행학습 | 프로세스 상태 전이

- **제출(Submit)** : 작업을 처리하기 위해 사용자가 작업을 시스템에 제출한 상태
- **접수(Hold)** : 제출된 작업이 스풀 공간인 디스크의 할당 위치에 저장된 상태
- **준비(Ready)** : 프로세스가 프로세서를 할당받기 위해 기다리고 있는 상태
- **실행(Run)** : 준비상태 큐에 있는 프로세스가 프로세서를 할당받아 실행되는 상태
- **대기(Wait), 블록(Block)** : 프로세스에 입 · 출력 처리가 필요하면 현재 실행 중인 프로세스가 중단되고, 입 · 출력 처리가 완료될 때까지 대기하고 있는 상태
- **종료(Terminated, Exit)** : 프로세스의 실행이 끝나고 프로세스 할당이 해제된 상태

프로세스 상태 전이 관련 용어

- Dispatch : 준비 상태에서 대기하고 있는 프로세스 중 하나가 프로세서를 할당받아 실행 상태로 전이되는 과정
- Wake Up : 입 · 출력 작업이 완료되어 프로세스가 대기 상태에서 준비 상태로 전이 되는 과정
- Spooling : 입 · 출력장치의 공유 및 상대적으로 느린 입 · 출력장치의 처리 속도를 보완하고 다중 프로그래밍 시스템의 성능을 향상시키기 위해 입 · 출력할 데이터를 직접 입 · 출력장치에 보내지 않고 나중에 한꺼번에 입 · 출력하기 위해 디스크에 저장하는 과정

76. IPv6의 주소 체계로 거리가 먼 것은?

① Unicast ② Anycast
③ Broadcast ④ Multicast

전문가 조언 | IPv6의 주소 체계에 유니캐스트(Unicast), 멀티캐스트(Multicast), 애니캐스트(Anycast)가 있습니다.

77. 응집도가 가장 낮은 것은?

① 기능적 응집도 ② 시간적 응집도
③ 절차적 응집도 ④ 우연적 응집도

전문가 조언 | 응집도를 강한 것에서 약한 것 순으로 나열하면 '기능적 응집도 → 순차적 응집도 → 교환(통신)적 응집도 → 절차적 응집도 → 시간적 응집도 → 논리적 응집도 → 우연적 응집도'입니다.

병행학습 | 응집도(Cohesion)

- 정보 은닉 개념을 확장한 것으로, 명령어나 호출문 등 모듈의 내부 요소들의 서로 관련되어 있는 정도, 즉 모듈이 독립적인 기능으로 정의되어 있는 정도를 의미한다.
- 다양한 기준으로 모듈을 구성할 수 있으나 응집도가 강할수록 품질이 높고, 약할수록 품질이 낮다.
- 응집도의 종류(강함에서 낮음순)
  - 기능적 응집도(Functional Cohesion) : 모듈 내부의 모든 기능 요소들이 단일 문제와 연관되어 수행될 경우의 응집도
  - 순차적 응집도(Sequential Cohesion) : 모듈 내 하나의 활동으로부터 나온 출력 데이터를 그 다음 활동의 입력 데이터로 사용할 경우의 응집도
  - 교환(통신)적 응집도(Communication Cohesion) : 동일한 입력과 출력을 사용하여 서로 다른 기능을 수행하는 구성 요소들이 모였을 경우의 응집도
  - 절차적 응집도(Procedural Cohesion) : 모듈이 다수의 관련 기능을 가질 때 모듈 안의 구성 요소들이 그 기능을 순차적으로 수행할 경우의 응집도
  - 시간적 응집도(Temporal Cohesion) : 특정 시간에 처리되는 몇 개의 기능을 모아 하나의 모듈로 작성할 경우의 응집도
  - 논리적 응집도(Logical Cohesion) : 유사한 성격을 갖거나 특정 형태로 분류되는 처리 요소들로 하나의 모듈이 형성되는 경우의 응집도
  - 우연적 응집도(Coincidental Cohesion) : 모듈 내부의 각 구성 요소들이 서로 관련 없는 요소로만 구성된 경우의 응집도

78. JAVA 언어에서 접근 제한자가 아닌 것은?

① public ② protected
③ package ④ private

전문가 조언 | JAVA의 접근 제한자에는 Public, Default, Private, Protected가 있습니다.

79. 스크립트 언어가 아닌 것은?

① PHP ② Cobol
③ Basic ④ Python

전문가 조언 | Cobol은 컴파일러 언어이기 때문에 스크립트 언어로 사용할 수 없습니다.

병행학습 | 스크립트 언어(Script Language)

- HTML 문서 안에 직접 프로그래밍 언어를 삽입하여 사용하는 것으로, 기계어로 컴파일 되지 않고 별도의 번역기가 소스를 분석하여 동작하게 하는 언어이다.
- 게시판 입력, 상품 검색, 회원 가입 등과 같은 데이터베이스 처리 작업을 수행하기 위해 주로 사용한다.
- 스크립트 언어는 클라이언트의 웹 브라우저에서 해석되어 실행되는 클라이언트용 스크립트 언어와 서버에서 해석되어 실행된 후 결과만 클라이언트로 보내는 서버용 스크립트 언어가 있다.
  - 서버용 스크립트 언어 : ASP, JSP, PHP, 파이썬(Python), 펄(Perl), 루비(Ruby) 등
  - 클라이언트용 스크립트 언어 : 자바 스크립트(Java Script), VB 스크립트(Visual Basic Script) 등

80. OSI-7계층에서 종단 간 신뢰성 있고 효율적인 데이터를 전송하기 위해 오류 검출과 복구, 흐름 제어를 수행하는 계층은?

① 전송 계층 ② 세션 계층
③ 표현 계층 ④ 응용 계층

전문가 조언 | 문제에 제시된 내용은 OSI 계층 중 전송 계층에 대한 설명입니다.

병행학습 | OSI(Open System Interconnection) 참조 모델

- 다른 시스템 간의 원활한 통신을 위해 ISO(국제표준화기구)에서 제안한 통신 규약(Protocol)이다.
- OSI 7계층 : 하위 계층(물리 계층 → 데이터 링크 계층 → 네트워크 계층), 상위 계층(전송 계층 → 세션 계층 → 표현 계층 → 응용 계층)
- 물리 계층(Physical Layer) : 전송에 필요한 두 장치 간의 실제 접속과 절단 등 기계적, 전기적, 기능적, 절차적 특성에 대한 규칙을 정의함
- 데이터 링크 계층(Data Link Layer)
  - 두 개의 인접한 개방 시스템들 간에 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 함
  - 흐름 제어, 프레임 동기화, 오류 제어, 순서 제어
- 네트워크 계층(Network Layer, 망 계층)
  - 개방 시스템들 간의 네트워크 연결을 관리하는 기능과 데이터의 교환 및 중계 기능을 함
  - 경로 설정(Routing), 트래픽 제어, 패킷 정보 전송
- 전송 계층(Transport Layer)
  - 종단 시스템(End-to-End) 간의 전송 연결 설정, 데이터 전송, 연결 해제 기능을 함
  - 주소 설정, 다중화(데이터의 분할과 재조립), 오류 제어, 흐름 제어
- 세션 계층(Session Layer)
  - 송 · 수신 측 간의 관련성을 유지하고 대화 제어를 담당함
  - 대화(회화) 구성 및 동기 제어, 데이터 교환 관리 기능
- 표현 계층(Presentation Layer)
  - 응용 계층으로부터 받은 데이터를 세션 계층에 맞게, 세션 계층에서 받은 데이터는 응용 계층에 맞게 변환하는 기능
  - 코드 변환, 데이터 암호화, 데이터 압축, 구문 검색, 정보 형식(포맷) 변환, 문맥 관리 기능
- 응용 계층(Application Layer) : 사용자(응용 프로그램)가 OSI 환경에 접근할 수 있도록 응용 프로세스 간의 정보 교환, 전자 사서함, 파일 전송, 가상 터미널 등의 서비스를 제공함

440205

5과목 정보시스템 구축 관리

81. 웹과 컴퓨터 프로그램에서 용량이 적은 데이터를 교환하기 위해 데이터 객체를 속성 · 값의 쌍 형태로 표현하는 형식으로, 자바스크립트(JavaScript)를 토대로 개발되어진 형식은?

① Python ② XML
③ JSON ④ WEB SEVER

전문가 조언 | 문제에 제시된 내용은 JSON(JavaScript Object Notation)에 대한 설명입니다.

병행학습 |

- 파이썬(Python) : 객체지향 기능을 지원하는 대화형 인터프리터 언어로, 플랫폼에 독립적이고 문법이 간단하여 배우기 쉬움
- XML(eXtensible Markup Language) : 특수한 목적을 갖는 마크업 언어를 만드는데 사용되는 다목적 마크업 언어
- 웹 서버(Web Server) : 클라이언트로부터 직접 요청을 받아 처리하는 서버로, 저용량의 정적 파일들을 제공함

82. 최대 홉수를 15로 제한한 라우팅 프로토콜은?

① RIP ② OSPF
③ Static ④ EIGRP

전문가 조언 | 최대 홉수를 15로 제한한 라우팅 프로토콜은 RIP(Routing Information Protocol)입니다.

병행학습 |

RIP(Routing Information Protocol)

- 현재 가장 널리 사용되는 라우팅 프로토콜이다.
- 소규모 동종의 네트워크(자율 시스템, AS) 내에서 효율적인 방법이다.
- 최대 홉(Hop)수를 15로 제한하므로 15 이상의 경우는 도달할 수 없는 네트워크를 의미하는데 이것은 대규모 네트워크에서는 RIP를 사용할 수 없음을 의미한다.
- 라우팅 정보를 30초마다 네트워크 내의 모든 라우터에 알리며, 180초 이내에 새로운 라우팅 정보가 수신되지 않으면 해당 경로를 이상 상태로 간주한다.

OSPF(Open Shortest Path First protocol)

- 대규모 네트워크에서 많이 사용되는 라우팅 프로토콜이다.
- 라우팅 정보에 변화가 생길 경우, 변화된 정보만 네트워크 내의 모든 라우터에 알린다.

83. IP 또는 ICMP의 특성을 악용하여 특정 사이트에 집중적으로 데이터를 보내 네트워크 또는 시스템의 상태를 불능으로 만드는 공격 방법은?

① TearDrop ② Smishing
③ Qshing ④ Smurfing

전문가 조언 | 문제에서 설명하는 공격 방법은 스머핑(Smurfing)입니다.

병행학습 |

- **티어드롭(TearDrop)** : 데이터의 송 · 수신 과정에서 패킷의 크기가 커 여러 개로 분할되어 전송될 때 분할 순서를 알 수 있도록 Fragment Offset 값을 함께 전송하는데, 티어드롭은 이 Offset 값을 변경시켜 수신 측에서 패킷을 재조립할 때 오류로 인한 과부하를 발생시킴으로써 시스템이 다운되도록 공격함
- **스미싱(Smishing)** : 문자 메시지(SMS)를 이용해 사용자의 개인 신용 정보를 빼내는 수법으로, 현재 각종 행사 안내, 경품 안내 등의 문자 메시지에 링크를 걸어 안드로이드 앱 설치 파일인 apk 파일을 설치하도록 유도하여 사용자 정보를 빼가는 수법으로 발전하고 있음
- **큐싱(Qshing)** : QR코드를 통해 악성 앱의 다운로드를 유도하거나 악성 프로그램을 설치하도록 하는 금융사기 기법의 하나로, QR코드와 개인정보 및 금융정보를 낚는다(Fishing)는 의미의 합성 신조어임

84. CMM(Capability Maturity Model) 모델의 레벨로 옳지 않은 것은?

① 최적단계 ② 관리단계
③ 정의단계 ④ 계획단계

전문가 조언 | CMM 모델은 초기, 관리, 정의, 정량적 관리, 최적화의 5단계로 구분합니다.

병행학습 | CMM(Capability Maturity Model)

- 소프트웨어 개발 조직의 업무 능력 및 조직의 성숙도를 평가하는 모델로, 미국 카네기멜론 대학교의 소프트웨어 공학연구소(SEI)에서 개발하였다.
- CMM의 소프트웨어 프로세스 성숙도는 초기, 관리, 정의, 정량적 관리, 최적화의 5단계로 구분한다.

단계	프로세스	특징
초기(Initial)	정의된 프로세스 없음	작업자 능력에 따라 성공 여부 결정
관리(Managed)	규칙화된 프로세스	특정한 프로젝트 내의 프로세스 정의 및 수행
정의(Defined)	표준화된 프로세스	조직의 표준 프로세스를 활용하여 업무 수행
정량적 관리 (Quantitatively Managed)	예측 가능한 프로세스	프로젝트를 정량적으로 관리 및 통제
최적화(Optimizing)	지속적 개선 프로세스	프로세스 역량 향상을 위해 지속적인 프로세스 개선

85. 여러 개의 독립된 통신장치가 UWB(Ultra Wide Band)기술 또는 블루투스 기술을 사용하여 통신망을 형성하는 무선 네트워크 기술은?

① PICONET ② SCRUM
③ NFC ④ WI-SUN

전문가 조언 | 문제에서 설명하는 무선 네트워크 기술은 PICONET입니다.

병행학습 |

- **SCRUM** : 럭비에서 반칙으로 경기가 중단된 경우 양 팀의 선수들이 럭비공을 가운데 두고 상대팀을 밀치기 위해 서로 대치해 있는 대형을 의미하는 것으로, 이처럼 팀이 중심이 되어 개발의 효율성을 높인다는 의미가 내포된 프로세스 모델임
- **NFC(Near Field Communication, 근거리 무선 통신)** : 고주파(HF)를 이용한 근거리 무선 통신 기술로, 13.56MHz 주파수를 이용해 10cm 내에서 최고 424Kbps의 속도로 데이터 전송을 지원함
- **와이선(Wi-SUN)** : 스마트 그리드와 같은 장거리 무선 통신을 필요로 하는 사물인터넷(IoT) 서비스를 위한 저전력 장거리(LPWA) 통신 기술로, 짧은 시간 동안 데이터 전송이 빈번한 검침 분야에 유용하며, 낮은 지연속도, 메시 네트워크 기반 확장성, 펌웨어 업그레이드 용이성 면에서 다른 저전력 장거리 통신 기술에 비해 우월함

86. COCOMO model 중 기관 내부에서 개발된 중소 규모의 소프트웨어로, 일괄 자료 처리나 과학기술 계산용, 비즈니스 자료 처리용으로 5만 라인 이하의 소프트웨어를 개발하는 유형은?

① embeded ② organic
③ semi-detached ④ semi-embeded

전문가 조언 | 문제에서 설명하는 COCOMO 모델의 소프트웨어 개발 유형은 조직형(Organic Mode)입니다.

병행학습 | COCOMO의 소프트웨어 개발 유형

- **조직형(Organic Mode)** : 기관 내부에서 개발된 중 · 소 규모의 소프트웨어로, 일괄 자료 처리나 과학 기술 계산용, 비즈니스 자료 처리용으로 5만(50KDSI) 라인 이하의 소프트웨어를 개발하는 유형이며, 사무 처리용, 업무용, 과학용 응용 소프트웨어 개발에 적합함
- **반분리형(Semi-Detached Mode)** : 조직형과 내장형의 중간형으로, 트랜잭션 처리 시스템이나 운영체제, 데이터베이스 관리 시스템 등의 30만(300KDSI) 라인 이하의 소프트웨어를 개발하는 유형이며, 컴파일러, 인터프리터와 같은 유틸리티 개발에 적합함
- **내장형(Embedded Mode)** : 최대형 규모의 트랜잭션 처리 시스템이나 운영체제 등의 30만(300KDSI) 라인 이상의 소프트웨어를 개발하는 유형으로, 신호기 제어 시스템, 미사일 유도 시스템, 실시간 처리 시스템 등의 시스템 프로그램 개발에 적합함

정답 : 79.② 80.① 81.③ 82.① 83.④ 84.④ 85.① 86.②

87. 컴퓨터 사용자의 키보드 움직임을 탐지해 ID, 패스워드 등 개인의 중요한 정보를 몰래 빼가는 해킹 공격은?

① Key Logger Attack ② Worm
③ Rollback ④ Zombie Worm

전문가 조언 | 컴퓨터 사용자의 키보드 움직임을 탐지해 ID, 패스워드 등 개인의 중요한 정보를 몰래 빼가는 해킹 공격은 키로거 공격(Key Logger Attack)입니다.

병행학습 | 정보 보안 침해 공격 관련 용어

- **좀비(Zombie) PC** : 악성코드에 감염되어 다른 프로그램이나 컴퓨터를 조종하도록 만들어진 컴퓨터로, C&C(Command & Control) 서버의 제어를 받아 주로 DDoS 공격 등에 이용됨
- **C&C 서버** : 해커가 원격지에서 감염된 좀비 PC에 명령을 내리고 악성코드를 제어하기 위한 용도로 사용하는 서버를 의미함
- **봇넷(Botnet)** : 악성 프로그램에 감염되어 악의적인 의도로 사용될 수 있는 다수의 컴퓨터들이 네트워크로 연결된 형태를 의미함
- **웜(Worm)** : 네트워크를 통해 연속적으로 자신을 복제하여 시스템의 부하를 높임으로써 결국 시스템을 다운시키는 바이러스의 일종으로, 분산 서비스 거부 공격, 버퍼 오버플로 공격, 슬래머 등이 웜 공격의 한 형태임
- **제로 데이 공격(Zero Day Attack)** : 보안 취약점이 발견되었을 때 발견된 취약점의 존재 자체가 널리 공표되기도 전에 해당 취약점을 통하여 이루어지는 보안 공격으로, 공격의 신속성을 의미함
- **키로거 공격(Key Logger Attack)** : 컴퓨터 사용자의 키보드 움직임을 탐지해 ID, 패스워드, 계좌번호, 카드번호 등과 같은 개인의 중요한 정보를 몰래 빼가는 해킹 공격
- **랜섬웨어(Ransomware)** : 인터넷 사용자의 컴퓨터에 잠입해 내부 문서나 파일 등을 암호화해 사용자가 열지 못하게 하는 프로그램으로, 암호 해독용 프로그램의 전달을 조건으로 사용자에게 돈을 요구하기도 함
- **백도어(Back Door, Trap Door)** : 시스템 설계자가 서비스 기술자나 유지 보수 프로그램 작성자(Programmer)의 액세스 편의를 위해 시스템 보안을 제거하여 만들어놓은 비밀 통로로, 컴퓨터 범죄에 악용되기도 함
- **트로이 목마(Trojan Horse)** : 정상적인 기능을 하는 프로그램으로 위장하여 프로그램 내에 숨어 있다가 해당 프로그램이 동작할 때 활성화되어 부작용을 일으키는 것으로, 자기 복제 능력은 없음

88. LOC 기법에 의하여 예측된 총 라인수가 50000 라인, 프로그래머의 월 평균 생산성이 200 라인, 개발에 참여할 프로그래머가 10인 일 때, 개발 소요 기간은?

① 25개월 ② 50개월
③ 200개월 ④ 2000개월

전문가 조언 | 프로그래머의 수가 10명이고, 월 평균 생산이 200 라인이라면 10명이 생산하는 월 생산은 2,000 라인입니다. 총 라인 수가 50,000일 경우 개발에 소요되는 기간은 25(50,000 / 2,000)개월입니다.

병행학습 | LOC 기법 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)

89. Rayleigh-Norden 곡선의 노력 분포도를 이용한 프로젝트 비용 산정기법은?

① Putnam 모형 ② 델파이 모형
③ COCOMO 모형 ④ 기능점수 모형

전문가 조언 | Rayleigh-Norden 곡선의 노력 분포도를 이용한 프로젝트 비용 산정기법은 Putnam 모형입니다.

병행학습 |

- **델파이 모형** : 전문가 감정 기법의 주관적인 편견을 보완하기 위해 많은 전문가의 의견을 종합하여 산정하는 기법으로, 전문가들의 편견이나 분위기에 지배되지 않도록 한 명의 조정자와 여러 전문가로 구성됨
- **COCOMO 모형** : 원시 프로그램의 규모인 LOC(원시 코드 라인 수)에 의한 비용 산정 기법으로, 보헴(Boehm)이 제안하였으며, 개발할 소프트웨어의 규모(LOC)를 예측한 후 이를 소프트웨어 종류에 따라 다르게 책정되는 비용 산정 방정식에 대입하여 비용을 산정함
- **기능점수 모형** : 알브레히트(Albrecht)가 제안한 것으로, 소프트웨어의 기능을 증대시키는 요인별로 가중치를 부여하고, 요인별 가중치를 합산하여 총 기능 점수를 산출하며 총 기능 점수와 영향도를 이용하여 기능 점수(FP)를 구한 후 이를 이용해서 비용을 산정하는 기법

90. 소인수 분해 문제를 이용한 공개키 암호화 기법에 널리 사용되는 암호 알고리즘 기법은?

① RSA ② ECC
③ PKI ④ PRM

전문가 조언 | 소인수 분해 문제를 이용한 공개키 암호화 기법에 널리 사용되는 암호 알고리즘 기법은 RSA(Rivest Shamir Adleman)입니다.

병행학습 | 주요 암호화 알고리즘

- SEED
  - 1999년 한국인터넷진흥원(KISA)에서 개발한 블록 암호화 알고리즘이다.
  - 블록 크기는 128비트이며, 키 길이에 따라 128, 256으로 분류한다.
- ARIA(Academy, Research Institute, Agency)
  - 2004년 국가정보원과 산학연협회가 개발한 블록 암호화 알고리즘이다.
  - 블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류한다.
- DES(Data Encryption Standard)
  - 1975년 미국 NBS에서 발표한 개인키 암호화 알고리즘이다.
  - 블록 크기는 64비트이며, 키 길이는 56비트이다.
- AES(Advanced Encryption Standard)
  - 2001년 미국 표준 기술 연구소(NIST)에서 발표한 개인키 암호화 알고리즘이다.
  - 블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류한다.
- RSA(Rivest Shamir Adleman)
  - 1978년 MIT의 라이베스트(Rivest), 샤미르(Shamir), 애들먼(Adelman)에 의해 제안된 공개키 암호화 알고리즘이다.
  - 소인수 분해 문제를 이용한 공개키 암호화 기법에 널리 사용된다.

91. 프로토타입을 지속적으로 발전시켜 최종 소프트웨어 개발까지 이르는 개발방법으로, 위험관리가 중심인 소프트웨어 생명주기 모형은?

① 나선형 모형 ② 델파이 모형
③ 폭포수 모형 ④ 기능점수 모형

전문가 조언 | 문제에서 설명하는 소프트웨어 생명 주기 모형은 나선형 모형(Spiral Model)입니다.

병행학습 |

- **폭포수 모형(Waterfall Model)** : 폭포에서 한번 떨어진 물은 거슬러 올라갈 수 없듯이 소프트웨어 개발도 이전 단계로 돌아갈 수 없다는 전제하에 각 단계를 확실히 매듭짓고 그 결과를 철저하게 검토하여 승인 과정을 거친 후에 다음 단계를 진행하는 개발 방법론
- **프로토타입 모형(Prototype Model, 원형 모형)** : 사용자의 요구사항을 정확히 파악하기 위해 실제 개발될 소프트웨어에 대한 견본품(Prototype)을 만들어 최종 결과물을 예측하는 모형
- **나선형 모형(Spiral Model, 점진적 모형)** : 폭포수 모형과 프로토타입 모형의 장점에 위험 분석 기능을 추가한 모형으로, 나선을 따라 돌듯이 여러 번의 소프트웨어 개발 과정을 거쳐 점진적으로 완벽한 최종 소프트웨어를 개발함
- **애자일 모형(Agile Model)** : 고객의 요구사항 변화에 유연하게 대응할 수 있도록 일정한 주기를 반복하면서 개발과정을 진행하는 모형으로, 어느 특정 개발 방법론이 아니라 좋은 것을 빠르고 낭비 없게 만들기 위해 고객과의 소통에 초점을 맞춘 방법론을 통칭함

92. 다음 설명의 정보보안 침해 공격 관련 용어는?

> 인터넷 사용자의 컴퓨터에 침입해 내부 문서 파일 등을 암호화해 사용자가 열지 못하게 하는 공격으로, 암호 해독용 프로그램의 전달을 조건으로 사용자에게 돈을 요구하기도 한다.

① Smishing ② C-brain
③ Trojan Horse ④ Ransomware

전문가 조언 | 문제의 지문에 제시된 내용은 랜섬웨어(Ransomware)에 대한 설명입니다.

93. 백도어 탐지 방법으로 틀린 것은?

① 무결성 검사 ② 닫힌 포트 확인
③ 로그 분석 ④ SetUID 파일 검사

전문가 조언 | 백도어 탐지 방법에는 무결성 검사, 동작중인 프로세스 및 열린 포트 확인, 로그 분석, SetUID 파일 검사 등이 있습니다.

94. 메모리상에서 프로그램의 복귀 주소와 변수 사이에 특정 값을 저장해 두었다가 그 값이 변경되었을 경우 오버플로우 상태로 가정하여 프로그램 실행을 중단하는 기술은?

① 모드체크 ② 리커버리 통제
③ 시스로그 ④ 스택가드

전문가 조언 | 문제에 제시된 내용은 스택가드(StackGuard)에 대한 설명입니다.

병행학습 |

- **모드체크(Mode Check)** : 영문, 숫자, 특수문자 등 입력될 수 있는 문자의 종류가 제한된 경우 입력 문자를 확인하여 이상 유무를 검색하는 것
- **시스로그(Syslog)** : 시스템 관리자가 송 · 수신한 메시지나 명령, 시스템 이벤트 발생 내용, 시스템 작업과 관련된 정보 등을 저장한 자료의 집합

95. 소프트웨어 개발 프레임워크를 적용할 경우 기대 효과로 거리가 먼 것은?

① 품질 보증
② 시스템 복잡도 증가
③ 개발 용이성
④ 변경 용이성

전문가 조언 | 소프트웨어 개발 프레임워크를 적용할 경우 시스템 복잡도의 증가가 아닌 감소 효과를 기대할 수 있습니다.

병행학습 | **소프트웨어 개발 프레임워크**

- 프레임워크(Framework)는 소프트웨어 개발에 공통적으로 사용되는 구성 요소와 아키텍처를 일반화하여 손쉽게 구현할 수 있도록 여러 가지 기능들을 제공해주는 반제품 형태의 소프트웨어 시스템이다.
- **주요 기능** : 예외 처리, 트랜잭션 처리, 메모리 공유, 데이터 소스 관리, 서비스 관리, 쿼리 서비스, 로깅 서비스, 사용자 인증 서비스 등
- **스프링 프레임워크(Spring Framework)** : 자바 플랫폼을 위한 오픈 소스 경량형 애플리케이션 프레임워크로, 동적인 웹 사이트의 개발을 위해 다양한 서비스를 제공함
- **전자정부 프레임워크** : 우리나라의 공공부문 정보화 사업 시 효율적인 정보 시스템의 구축을 지원하기 위해 필요한 기능 및 아키텍처를 제공하는 프레임워크로, 개발 프레임워크의 표준 정립으로 응용 소프트웨어의 표준화, 품질 및 재사용성의 향상을 목적으로 함
- **닷넷 프레임워크(.NET Framework)** : Windows 프로그램의 개발 및 실행 환경을 제공하는 프레임워크로, Microsoft 사에서 통합 인터넷 전략을 위해 개발하였으며, 실행을 관리하는 CLR(Common Language Runtime, 공용 언어 런타임)이라는 이름의 가상머신 상에서 작동함

96. 시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있는 보안 요소는?

① 기밀성 ② 부인방지
③ 가용성 ④ 무결성

전문가 조언 | 시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있는 보안 요소는 무결성(Integrity)입니다.

병행학습 | **보안 요소**

- 소프트웨어 개발에 있어 충족시켜야할 요소 및 요건을 의미한다.
- 주요 보안 요소에는 기밀성, 무결성, 가용성이 있으며, 그 외에도 인증, 부인 방지 등이 있다.
  - 기밀성(Confidentiality) : 시스템 내의 정보와 자원은 인가된 사용자에게만 접근이 허용되며, 정보가 전송 중에 노출되더라도 데이터를 읽을 수 없음
  - 무결성(Integrity) : 시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있음
  - 가용성(Availability) : 인가받은 사용자는 언제라도 사용할 수 있음
  - 인증(Authentication) : 시스템 내의 정보와 자원을 사용하려는 사용자가 합법적인 사용자인지를 확인하는 모든 행위로, 대표적 방법으로는 패스워드, 인증용 카드, 지문 검사 등이 있음
  - 부인 방지(NonRepudiation) : 데이터를 송 · 수신한 자가 송 · 수신 사실을 부인할 수 없도록 송 · 수신 증거를 제공함

97. 다음이 설명하는 용어로 옳은 것은?

- 오픈 소스를 기반으로 한 분산 컴퓨팅 플랫폼이다.
- 일반 PC급 컴퓨터들로 가상화된 대형 스토리지를 형성한다.
- 다양한 소스를 통해 생성된 빅데이터를 효율적으로 저장하고 처리한다.

① 하둡(Hadoop)
② 비컨(Beacon)
③ 포스퀘어(Foursquare)
④ 맴리스터(Memristor)

전문가 조언 | 문제의 지문에 제시된 내용은 하둡(Hadoop)에 대한 설명입니다.

병행학습 |

- **비컨(Beacon)** : 전파를 이용하는 무선 통신 기술에서 주기적으로 프레임 신호 동기를 맞추고, 송 · 수신 관련 시스템 정보를 전송하며, 수신 데이터 정보를 전달하는 신호 기술
- **포스퀘어(Foursquare)** : 특정 지역이나 장소를 방문할 때 자신의 위치를 지도에 표시하고, 방문한 곳의 정보를 남길 수 있는 기능을 제공하는 위치 기반 소셜 네트워크 서비스(SNS)
- **멤리스터(Memristor)** : 메모리(Memory)와 레지스터(Resister)의 합성어로, 전류의 방향과 양 등 기존의 경험을 모두 기억하는 특별한 소자이며, 전원 공급이 끊어졌을 때도 직전에 통과한 전류의 방향과 양을 기억하기 때문에 다시 전원이 공급되면 기존의 상태가 그대로 복원됨

98. 크래커가 침입하여 백도어를 만들어 놓거나, 설정 파일을 변경했을 때 분석하는 도구는?

① trace ② tripwire
③ udpdump ④ cron

전문가 조언 | 크래커가 침입하여 백도어를 만들어 놓거나, 설정 파일을 변경했을 때 분석하는 도구를 데이터 무결성 검사 도구라고 하며, 보기 중에서는 tripwire가 해당됩니다.

병행학습 | **데이터 무결성 검사 도구**

- 시스템의 파일 변경 유무를 확인하고 파일이 변경되었을 경우 이를 관리자에게 알려주는 역할을 수행한다.
- 크래커나 허가받지 않은 내부 사용자들이 시스템에 침입하면 백도어를 만들어 놓거나 시스템 파일을 변경하여 자신의 흔적을 감추는데, 이를 무결성 검사 도구를 이용하여 감지할 수 있다.
- 데이터 무결성 도구는 현재 파일 및 디렉토리의 상태를 해시(Hash) 함수를 이용하여 DB 저장한 후 모니터링을 수행하다가 파일이나 디렉토리가 DB에 저장해둔 상태와 달라지는 경우 관리자에게 변경 사실을 알려준다.
- 대표적인 데이터 무결성 검사 도구에는 Tripwire, AIDE, Samhain, Claymore, Slipwire, Fcheck 등이 있다.

99. 폭포수 모형의 특징으로 거리가 먼 것은?

① 개발 중 발생한 요구사항을 쉽게 반영할 수 있다.
② 순차적인 접근방법을 이용한다.
③ 단계적 정의와 산출물이 명확하다.
④ 모형의 적용 경험과 성공사례가 많다.

전문가 조언 | 폭포수 모형은 이전 단계로 돌아갈 수 없으므로 개발 중 발생한 요구사항을 쉽게 반영할 수 없습니다.

병행학습 | **폭포수 모형(Waterfall Model)**

- 폭포에서 한번 떨어진 물은 거슬러 올라갈 수 없듯이 소프트웨어 개발도 이전 단계로 돌아갈 수 없다는 전제하에 각 단계를 확실히 매듭짓고 그 결과를 철저하게 검토하여 승인 과정을 거친 후에 다음 단계를 진행하는 개발 방법론이다.
- 폭포수 모형은 소프트웨어 공학에서 가장 오래되고 가장 폭넓게 사용된 전통적인 소프트웨어 생명 주기 모형으로, 고전적 생명 주기 모형이라고도 한다.
- 소프트웨어 개발 과정의 한 단계가 끝나야만 다음 단계로 넘어갈 수 있는 선형 순차적 모형이다.
- 제품의 일부가 될 매뉴얼을 작성해야 한다.
- 각 단계가 끝난 후에는 다음 단계를 수행하기 위한 결과물이 명확하게 산출되어야 한다.
- 두 개 이상의 과정이 병행하여 수행되지 않는다.

100. 테일러링(Tailoring) 개발 방법론의 내부 기준에 해당하지 않는 것은?

① 납기/비용 ② 기술 환경
③ 구성원 능력 ④ 국제 표준 품질 기준

전문가 조언 | 국제 표준 품질 기준은 테일러링 개발 방법론의 외부 기준에 해당합니다.

병행학습 | 소프트웨어 개발 방법론 테일러링

- 프로젝트 상황 및 특성에 맞도록 정의된 소프트웨어 개발 방법론의 절차, 사용기법 등을 수정 및 보완하는 작업이다.
- 소프트웨어 개발 방법론 테일러링 작업 시 고려해야 할 사항에는 내부적 요건과 외부적 요건이 있다.
- 내부적 요건

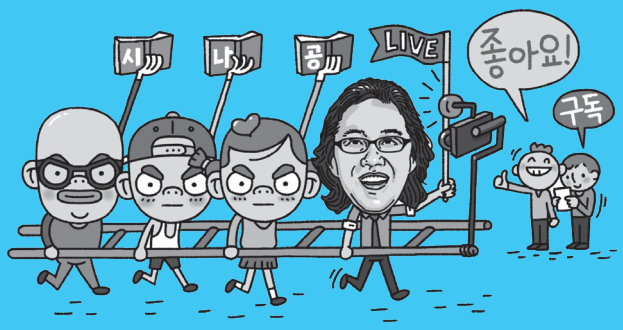
목표 환경	시스템의 개발 환경과 유형이 서로 다른 경우 테일러링이 필요함
요구사항	프로젝트의 생명 주기 활동에서 개발, 운영, 유지보수 등 프로젝트에서 우선적으로 고려할 요구사항이 서로 다른 경우 테일러링이 필요함
프로젝트 규모	비용, 구성원(인력), 기간 등 프로젝트의 규모가 서로 다른 경우 테일러링이 필요함
보유 기술	프로세스, 개발 방법론, 산출물 등이 서로 다른 경우 테일러링이 필요함

- 외부적 요건

법적 제약사항	프로젝트별로 적용될 IT Compliance가 서로 다른 경우 테일러링이 필요함
표준 품질 기준	금융, 제도 등 분야별 표준 품질 기준이 서로 다른 경우 테일러링이 필요함

m·e·m·o

최종점검 모의고사



정보처리기사 필기 최종점검 모의고사 1회

1과목 소프트웨어 설계

1. 소프트웨어 개발 방법론 중 애자일 모형의 핵심 가치에 대한 설명으로 옳지 않는 것은?
① 계약 협상보다는 고객과 협업에 더 가치를 둔다.
② 변화에 반응하기 보다는 계획과 일정을 준수하는 것에 더 가치를 둔다.
③ 프로세스와 도구보다는 개인과 상호작용에 더 가치를 둔다.
④ 방대한 문서보다는 실행되는 SW에 더 가치를 둔다.

2. 다음 보기에 제시된 스크럼(Scrum) 개발 과정을 진행 순서에 맞게 나열한 것은?

〈보기〉

㉠ 스프린트 검토 회의(Sprint Review)
㉡ 스프린트 계획 회의(Sprint Planning Meeting)
㉢ 일일 스크럼 회의(Daily Scrum Meeting)
㉣ 스프린트(Sprint)
㉤ 스프린트 회고(Sprint Retrospective)

① ㉡ → ㉢ → ㉠ → ㉤ → ㉣
② ㉡ → ㉣ → ㉢ → ㉠ → ㉤
③ ㉡ → ㉣ → ㉠ → ㉢ → ㉤
④ ㉡ → ㉠ → ㉤ → ㉢ → ㉣

3. 새로운 시스템을 개발하기 위해 현행 시스템의 하드웨어 구성을 파악하고자 한다. 이 때 고려해야 할 사항이 아닌 것은?
① 운용 서버의 주요 사양
② 운용 서버의 용도
③ 운용 서버의 이중화 여부
④ 운용 서버의 사용 기간

4. 요구사항 분석 기법 중 정형 분석(Formal Analysis)에 대한 설명으로 옳지 않은 것은?
① 구문(Syntax)과 의미(Semantics)를 갖는 정형화된 언어를 사용한다.
② 요구사항을 수학적 기호로 표현하여 분석한다.
③ 요구사항 분석의 마지막 단계에서 이루어진다.
④ 요구사항을 만족시키기 위한 구성 요소를 식별한다.

5. 요구사항의 유형 중 기능적 요구사항(Functional Requirement)에 대한 설명으로 옳지 않은 것은?
① 시스템이 무엇을 하는지, 어떤 기능을 하는지에 대한 사항
② 프로젝트의 원활한 수행을 위한 관리 방법에 대한 사항
③ 시스템이 반드시 수행해야 하는 기능
④ 사용자가 시스템을 통해 제공받기를 원하는 기능

6. 웹 애플리케이션 서버(WAS)와 관련된 요구사항 식별 시 고려해야 할 사항이 아닌 것은?
① 가용성 ② 성능
③ 이식성 ④ 기술 지원

7. 다음 설명에 해당하는 요구사항 확인 기법은?

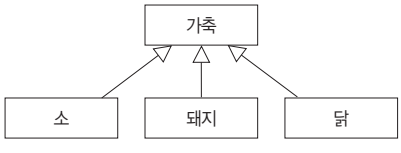
- 사용자가 실제로 사용될 환경에서 요구사항들이 모두 충족되는지 사용자 입장에서 확인하는 과정이다.
- 각각의 요구사항을 어떻게 확인할 것인지에 대한 계획이 필요하다.

① 인수 테스트 ② 모델 검증
③ 프로토타이핑 ④ 요구사항 검토

8. 다음 두 개체 간의 관계는 UML 다이어그램에서 어떤 관계에 해당하는가?

① 연관(Association) 관계
② 집합(Aggregation) 관계
③ 포함(Composition) 관계
④ 일반화(Generalization) 관계

9. UML의 다이어그램에서 관계를 완성하고자 한다. 다음 관계의 표현으로 가장 적합한 것은?



① 의존 관계(Dependency)
② 실체화 관계(Realization)
③ 일반화 관계(Generalization)
④ 연관 관계(Association)

10. 다음 〈실무 사례〉에 해당하는 사용자 인터페이스는 무엇인가?

> 〈실무 사례〉
> 시중에 출시된 VR 기기들은 사용자의 위치와 방향, 움직임을 추적하는 기능을 탑재하고 있어, 사용자는 앞 · 뒤 · 좌 · 우로 이동하거나 팔을 휘두르거나, 허공에서 버튼을 누르는 등 다양한 동작을 통해 기기를 조작하는 것이 가능하다.

① CLI(Command Line Interface)
② GUI(Graphical User Interface)
③ VUI(Voice User Interface)
④ NUI(Natural User Interface)

11. 웹 사이트 개발 시 고려해야 할 웹의 3요소에 해당하지 않는 것은?
① 웹 가용성(Web Availability)
② 웹 표준(Web Standards)
③ 웹 접근성(Web Accessibility)
④ 웹 호환성(Cross Browsing)

12. 다음 중 사용자 인터페이스(User Interface)의 설계 도구에 대한 설명으로 옳지 않은 것은?
① 화면 설계 도구에는 파워포인트, 스토리보드, 와이어프레임, 목업 등이 있다.
② 목업은 실제 화면과 유사한 동적인 형태의 모형이다.
③ 프로토타입에는 페이퍼 프로토타입과 디지털 프로토타입이 있다.
④ 와이어프레임은 기획 단계에서 페이지 레이아웃이나 구성 요소 등 뼈대를 설계하는 단계이다.

13. 다음의 정황 시나리오에서 추출된 요구사항으로 옳지 않은 것은?

> • PC를 통해 다른 프로그램으로 작성했던 표와 그래프를 불러와 발표용 자료를 만들었다.
> • 스마트폰으로 해당 자료를 확인하는 한편 잘못된 곳을 수정하였다.
> • 발표 주제와 관련된 사람들에게 해당 자료를 공유하여 문제점은 없는지 자문을 요청하였다.

① 외부 프로그램의 개체를 가져와 사용할 수 있어야 한다.
② PC와 스마트폰에서 모두 구현이 가능해야 한다.
③ 공유가 가능해야 한다.
④ 공유된 자료를 동시에 열람할 수 있어야 한다.

14. 자료 사전에서 자료의 반복을 의미하는 것은?
① =　　② { }
③ ( )　　④ [ ]

15. 소프트웨어 아키텍처 설계의 기본 원리에 대한 설명으로 틀린 것은?
① 소프트웨어 아키텍처 설계의 기본 원리에는 모듈화, 추상화, 단계적 분해, 정보은닉이 있다.
② 모듈화는 소프트웨어의 수정 및 재사용, 유지 관리 등을 용이하도록 기능들을 모듈 단위로 나누는 것이다.
③ 추상화는 문제의 세부적인 개념을 설계한 후 차례로 통합하여 포괄적인 개념을 구성하는 것이다.
④ 정보은닉은 다른 모듈이 접근하거나 변경하지 못하도록 한 모듈 내부에 포함된 절차와 자료들의 정보를 감추는 것이다.

16. 아키텍처 패턴 중 장애 허용 시스템과 병렬 컴퓨팅 시스템에서 주로 활용되는 패턴으로, 메인 컴포넌트에서 서브 컴포넌트로 작업을 분할하여 배정한 후, 처리된 결과물을 다시 돌려받는 방식의 패턴을 무엇이라 하는가?
① 클라이언트-서버
② 파이프-필터
③ 마스터-슬레이브
④ 모델-뷰-컨트롤러

17. 모듈 결합도의 종류 중 스탬프 결합도(Stamp Coupling)에 대한 설명으로 틀린 것은?
① 모듈 간의 인터페이스로 배열이나 레코드 등의 자료 구조가 전달될 때의 결합도이다.
② 결합의 정도가 자료 결합도보다 강하고, 제어 결합도보다 약하다.
③ 두 모듈이 동일한 자료 구조를 조회하는 경우의 결합도이다.
④ 자료 구조의 변화가 다른 모듈에 영향을 끼치지 않는다.

18. 다음 설명에 해당하는 코드의 종류는 무엇인가?

> 코드화 대상 항목의 성질, 즉 길이, 넓이, 부피, 지름, 높이 등의 물리적 수치를 그대로 코드에 적용시키는 방법이다. 대표적인 예로 가로가 55cm, 세로가 120cm인 합판을 코드로 표현하면 '55-120'이 된다.

① 표의 숫자 코드(Significant Digit Code)
② 순차 코드(Sequence Code)
③ 연상 코드(Mnemonic Code)
④ 블록 코드(Block Code)

정답 : 1.② 2.② 3.④ 4.④ 5.② 6.③ 7.① 8.③ 9.③ 10.④ 11.① 12.② 13.④ 14.② 15.③ 16.③ 17.④ 18.①

19. 다음이 설명하는 미들웨어는?

> • 은행 계정, 항공기와 버스 예약 업무 등 온라인 트랜잭션 업무에서 트랜잭션을 처리하고 감시하는 미들웨어이다.
> • 사용자 수가 증가하여도 빠른 응답 속도를 유지해야 하는 업무에 적합한 미들웨어이다.

① TP-Monitor ② RPC
③ WAS ④ DB

20. DFD(Data Flow Diagram)에 대한 설명으로 거리가 먼 것은?

① DFD의 구성 요소는 Process, Data Flow, Data Store, Terminator이다.
② DFD는 자료 흐름과 기능을 자세히 표현하기 위하여 단계적으로 세분화된다.
③ DFD의 구성 요소 중 Process는 원으로 표시한다.
④ DFD로 데이터가 논리적 시스템들 간에 어떻게 흘러 다니는지는 알 수 있지만, 제어나 순서에 관한 정보는 알 수가 없다.

2과목 소프트웨어 개발

21. 다음 설명에 해당하는 자료 구조는 무엇인가?

> • 동일한 자료형의 데이터들이 같은 크기로 나열되어 순서를 갖고 있는 집합이다.
> • 기억장소의 추가가 어렵고, 데이터 삭제 시 데이터가 저장되어 있던 기억장소는 빈 공간으로 남아있다.
> • 데이터마다 동일한 이름의 변수를 사용한다.

① 리스트(List) ② 배열(Array)
③ 스택(Stack) ④ 트리(Tree)

22. 데이터의 독립성에 대한 설명으로 옳지 않은 것은?

① 응용 프로그램과 데이터베이스를 독립시킨 것은 논리적 독립성이다.
② 논리적 독립성은 데이터의 구조가 변경되어도 응용 프로그램에 영향을 주지 않는 것을 의미한다.
③ 물리적 독립성은 응용 프로그램과 하드웨어를 독립시킨 것이다.
④ 물리적 독립성은 DBMS가 변경되어도 응용 프로그램에 영향을 주지 않는 것을 의미한다.

23. 다음 괄호에 들어갈 적합한 용어는 무엇인가?

> 데이터베이스의 상태를 변환시키는 하나의 논리적 기능을 수행하기 위한 작업의 단위를 트랜잭션이라고 하며, 이를 제어하기 위해 사용하는 COMMIT, ROLLBACK과 같은 명령어를 ( )이라고 한다.

① SQL ② DDL
③ TCL ④ DML

24. 절차형 SQL에 대한 설명으로 옳지 않은 것은?

① BEGIN ~ END 형식으로 작성되는 블록(Block) 구조로 구성된다.
② DBMS 엔진에서 직접 실행되기 때문에 일반적인 프로그래밍 언어에 비해 효율이 뛰어나다.
③ 프로그래밍 언어와 같이 연속적인 실행이나 분기, 반복 등의 제어가 가능하다.
④ 기능별 모듈화가 가능하다.

25. 모듈 간 통신 방식을 구현하기 위해 사용되는 IPC의 메소드에 속하지 않는 것은?

① Adapter ② Shared Memory
③ Semaphores ④ Message Queueing

26. 다음 중 ISO/IEC/IEEE 29119-3 표준에 따른 테스트 케이스의 구성 요소가 아닌 것은?

① 테스트 항목(Test Item)
② 테스트 스크립트(Test Script)
③ 입력 명세(Input Specification)
④ 환경 설정(Environmental Needs)

27. 협업 도구에 대한 설명으로 옳지 않은 것은?

① 그룹웨어, 협업 소프트웨어 등으로도 불린다.
② 개발에 참여하는 사람들이 서로 원활히 프로젝트를 수행할 수 있도록 도와주는 도구를 의미한다.
③ 협업 도구의 사용이 익숙하지 않더라도 일단 사용한다면 생산성의 향상을 기대할 수 있다.
④ 진행 현황 및 일정 등을 관리해주거나, 팀원 간 정보 공유를 위한 도구 등이 협업 도구에 속한다.

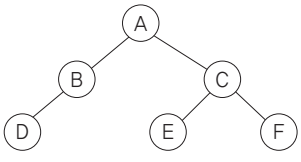
28. 모듈별로 생성한 실행 파일들을 묶어 배포용 설치 파일을 만드는 것을 무엇이라 하는가?

① 패키징(Packaging)
② 빌딩(Building)
③ 모듈화(Modularity)
④ 컴파일링(Compiling)

29. 릴리즈 노트(Release Note)를 통해 확인할 수 있는 정보가 아닌 것은?
① 테스트 진행 방법에 대한 결과
② 소프트웨어에서 사용한 외부 라이브러리
③ 소프트웨어 사양에 대한 개발팀의 정확한 준수 여부
④ 소프트웨어에 포함된 전체 기능

30. 디지털 저작권 관리(DRM)에 대한 설명으로 잘못된 것은?
① 아날로그 콘텐츠인 경우에는 디지털로 변환하여 DRM 패키징을 수행한다.
② 패키징을 수행하면 콘텐츠에 암호화된 저작권자의 전자서명이 포함된다.
③ 패키징된 콘텐츠의 라이선스 정보는 DRM 컨트롤러에 등록된다.
④ 크기가 작은 콘텐츠는 실시간으로 패키징되어 배포되며, 크기가 크면 미리 패키징을 수행한 후 배포된다.

31. 다음 트리를 후위 순회(Post Traversal)한 결과는?



① A B D C E F　　② D B E F C A
③ A B C D E F　　④ D B A E C F

32. 다음 형상 관리의 기능 중 형상 통제에 대한 설명으로 옳은 것은?
① 형상 항목에 대한 변경 요구를 검토하여 현재의 베이스 라인이 잘 반영될 수 있도록 조정하는 작업
② 베이스 라인의 무결성을 평가하기 위해 확인, 검증, 검열 과정을 통해 공식적으로 승인하는 작업
③ 형상 관리 대상을 계층(Tree) 구조로 구분하여 수정 및 추적이 용이하도록 하는 작업
④ 소프트웨어 업그레이드나 유지 보수 과정에서 생성된 다른 버전의 형상 항목을 관리하는 것

33. 다음 설명에 해당하는 소프트웨어 버전 관리 도구는 무엇인가?

> • 클라이언트/서버 방식의 버전 관리 도구로, 서버에서 최신 버전의 파일들과 변경 내역이 관리되는 오픈 소스 도구이다.
> • 모든 개발 작업은 trunk에서 수행되며, 추가 작업은 branches 안에 별도의 디렉터리를 만들어 작업을 완료한 후 병합한다.
> • 커밋할 때마다 리비전이 1씩 증가한다.

① SCCS　　② DCVS
③ SVN　　④ GIT

34. 다음 중 애플리케이션 테스트가 필요한 이유로 가장 옳지 않은 것은?
① 테스터의 역량 강화를 위해
② 새로운 오류의 유입을 예방하기 위해
③ 사용자의 요구사항이나 기대 수준 등을 만족시키기 위해
④ 프로그램 실행 전에 오류를 발견하여 예방하기 위해

35. 구문 기반 테스트, 결정 기반 테스트 등 소프트웨어 내부의 논리 흐름에 따라 테스트 케이스를 작성하고 확인하는 테스트를 무엇이라 하는가?
① 구조 기반 테스트　　② 명세 기반 테스트
③ 경험 기반 테스트　　④ 정적 테스트

36. 다음 중 블랙박스 테스트에 속하지 않는 테스트 방식은?
① 동등 분할 검사
② 제어 구조 검사
③ 경계값 분석
④ 오류 예측 검사

37. 다음 괄호에 들어갈 적합한 테스트는 무엇인가?

> (　　)는 이미 테스트된 프로그램의 테스팅을 반복하는 것으로, 통합 테스트로 인해 변경된 모듈이나 컴포넌트에 새로운 오류가 있는지 확인하는 테스트이다. (　　)는 수정한 모듈이나 컴포넌트가 다른 부분에 영향을 미치는지, 오류가 생기지 않았는지 테스트하여 새로운 오류가 발생하지 않음을 보증하기 위해 반복 테스트한다.

① 회귀 테스트
② 혼합식 통합 테스트
③ 상향식 통합 테스트
④ 하향식 통합 테스트

38. 테스트 데이터에 대한 설명 중 틀린 것은?
① 시스템의 기능이나 적합성 등을 테스트하기 위해 만든 데이터 집합이다.
② 소프트웨어의 기능을 차례대로 테스트할 수 있도록 만든 데이터이다.
③ 잘못된 테스트 데이터는 잘못된 결과를 도출하기 때문에 신중하게 작업해야 한다.
④ 테스트 데이터는 실제 운영되는 데이터가 아닌 스크립트를 통해서 인위적으로 만든 데이터만을 이용해야 한다.

정답 : 19.① 20.④ 21.② 22.④ 23.③ 24.② 25.① 26.② 27.③ 28.① 29.② 30.③ 31.② 32.① 33.③ 34.① 35.① 36.② 37.① 38.④

39. 테스트 하네스(Test Harness)의 구성 요소에 해당하지 않는 것은?
① 테스트 드라이버(Test Driver)
② 테스트 블록(Test Block)
③ 테스트 케이스(Test Case)
④ 목 오브젝트(Mock Object)

40. ESB(Enterprise Service Bus)에 대한 설명으로 틀린 것은?
① 애플리케이션 간 연계, 데이터 변환 등의 인터페이스를 제공하는 솔루션이다.
② 서비스 보다는 애플리케이션 중심의 통합을 지향한다.
③ 범용적 사용을 위해 애플리케이션과의 결합도가 약하다.
④ 관리 및 보안 유지가 쉽고, 높은 수준의 품질 지원이 가능하다.

3과목 데이터베이스 구축

41. 데이터 모델에 표시하는 요소 중의 하나로, 데이터의 무결성 보장을 위해 저장될 수 있는 실제 데이터에 조건을 설정하여 해당 조건에 부합하지 않으면 저장되지 않도록 하는 것은 무엇인가?
① Operation ② Relationship
③ Constraint ④ Structure

42. 다음은 개체 구성 방식에 따라 속성(Attribute)을 분류한 것이다. 설명이 틀린 것은?
① 기본키 속성은 개체를 식별할 수 있는 속성을 의미한다.
② 외래키 속성은 다른 개체와의 관계에서 만들어진 속성이다.
③ 외래키 속성과 관계된 개체가 변경되어도 외래키 속성에는 영향을 주지 않는다.
④ 일반 속성은 기본키와 외래키에 속하지 않은 속성들을 의미한다.

43. 다음의 설명에 해당하는 관계는 무엇인가?

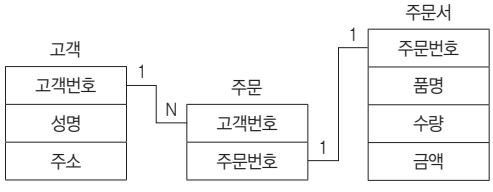
- 〈도시〉 개체와 〈시민〉 개체는 시민이 도시에 거주하는 관계이다.
- 한 명의 시민은 하나의 도시만 거주할 수 있다.
- 하나의 도시는 여러 명의 시민을 거주시킬 수 있다.

① 1:0 ② 1:1
③ 1:N ④ N:M

44. 주 식별자(Primary Identifier)의 속성이 두 개 이상인 경우 속성들을 하나의 속성으로 묶어 사용하는 식별자를 무엇이라 하는가?
① 대리 식별자(Surrogate Identifier)
② 복합 식별자(Composit Identifier)
③ 외부 식별자(Foreign Identifier)
④ 보조 식별자(Alternate Identifier)

45. 개체-관계 모델(E-R Model)에 대한 설명으로 옳지 않은 것은?
① 개체 타입과 이들 간의 관계 타입을 기본 요소로 이용하여 현실 세계를 개념적으로 표현한다.
② 오너-멤버(Owner-Member) 관계라고도 한다.
③ E-R 다이어그램에서 개체 타입은 사각형으로 나타낸다.
④ E-R 다이어그램에서 속성은 타원으로 나타낸다.

46. 다음 E-R 모델에서 '고객'과 '주문서'는 개체 테이블이고, '주문'은 관계(Relationship)로 만들어진 테이블이다. 다음 E-R 모델에 대한 설명으로 가장 옳지 않은 것은?



① 〈고객〉 테이블과 〈주문서〉 테이블은 〈주문〉 테이블을 관계로 하여 연결되어 있다.
② 〈고객〉 테이블에 있는 '고객번호'와 〈주문서〉 테이블에 있는 '주문번호'는 기본키이고, 〈주문〉 테이블에 있는 '고객번호'와 '주문번호'는 외래키이다.
③ 〈고객〉과 〈주문〉의 관계는 1:N이고, 〈주문서〉와 〈주문〉의 관계는 1:1이다.
④ 〈고객〉과 〈주문서〉의 관계는 N:M이 된다.

47. SQL문에서 사원(사번, 이름, 부서) 테이블에 사번이 1201, 이름이 "홍길동", 부서가 "인사"인 튜플을 삽입하는 명령으로 옳은 것은?
① INSERT FROM 사원(사번, 이름, 부서) VALUES(1201, '홍길동', '인사');
② INSERT 사원 INTO VALUES(1201, '홍길동', '인사');
③ INSERT TO 사원(사번, 이름, 부서) VALUES(1201, '홍길동', '인사');
④ INSERT INTO 사원 VALUES(1201, '홍길동', '인사');

48. 테이블 PRODUCT(CODE, NAME, TYPE, SALES)에 대해 다음과 같은 SQL문이 실행되었을 때 그 결과에 대한 설명으로 가장 옳지 않은 것은?

```
SELECT TYPE, COUNT(*), SUM(SALES)
FROM PRODUCT
GROUP BY TYPE
HAVING SUM(SALES) >= 300
ORDER BY TYPE DESC;
```

① TYPE 속성을 기준으로 그룹화 되어 출력된다.
② 각 튜플들은 TYPE 속성을 기준으로 내림차순 출력된다.
③ TYPE 속성으로 그룹화된 튜플들의 SALES 속성 합계가 300 미만인 경우 출력되지 않는다.
④ HAVING의 조건 "SUM(SALES) >= 300"을 WHERE 조건으로 바꿔도 동일한 결과가 출력된다.

49. 다음 Oracle로 작성된 SQL문에 대한 설명으로 가장 옳지 않은 것은? (단, MOD(A, B)는 A를 B로 나누어 나머지를 구하는 함수이다.)

```
CREATE PROCEDURE GET_DEPT(i IN INT) IS
BEGIN
  IF (MOD(i, 10) = 2) THEN
     INSERT INTO 회원(코드, 분류) VALUES(i, 'VIP');
  ELSE
     INSERT INTO 회원(코드, 분류) VALUES(i, '일반');
  END IF;
END;
```

① RUN GET_DEPT(30)을 입력하면 해당 프로시저가 실행된다.
② 파라미터로 i를 전달받아 수행되는 프로시저이다.
③ i를 10으로 나눈 나머지의 값에 따라 수행되는 코드가 다르다.
④ i의 값이 302이면 〈회원〉 테이블의 '코드' 속성에는 i의 값 302가, '분류' 속성에는 'VIP'가 저장된다.

50. 관계형 데이터베이스에서 사용하는 용어 중 카디널리티(Cardinality)에 대한 설명으로 옳은 것은?
① 디그리 또는 차수라고 한다.
② 테이블에 속한 튜플의 수를 의미한다.
③ 속성이 취할 수 있는 원자값들의 집합이다.
④ 파일 구조상의 데이터 항목 또는 필드에 해당한다.

51. 데이터 무결성을 강화하는 방법에 대한 설명으로 틀린 것은?
① 데이터를 조작하는 프로그램에 무결성 검증 코드를 삽입한다.
② 데이터베이스의 트리거 개체를 생성하여 무결성 검증 코드를 삽입한다.
③ 제약 조건을 설정하여 무결성을 유지한다.
④ 네트워크 채널을 통제하여 무결성 조건에 부합하지 않는 데이터를 반송한다.

52. 릴레이션 R의 차수가 3이고 카디널리티가 4이며, 릴레이션 S의 차수가 5이고 카디널리티가 6일 때, 두 개의 릴레이션에 대해 카티션 프로덕트를 수행하면 새롭게 생성되는 릴레이션의 차수와 카디널리티는 얼마인가?
① 8, 24 ② 15, 10
③ 7, 11 ④ 12, 30

53. 정규화(Normalization)의 각 정규형에 대한 설명으로 틀린 것은?
① 1NF는 모든 도메인이 원자값만으로 구성되어 있다.
② 2NF는 모든 속성이 기본키에 완전 함수적 종속이다.
③ 3NF는 모든 속성이 다치 종속의 결정자에 함수적 종속 관계를 만족한다.
④ BCNF는 모든 결정자가 후보키이다.

54. 인덱스 대상 속성 설정 기준에 대한 설명으로 가장 옳지 않은 것은?
① 인덱스 속성의 분포도가 20% 이상인 속성
② 수정이 빈번하지 않은 속성
③ Primary key 또는 Unique 제약 조건이 설정된 속성
④ 입 · 출력 장표 등에서 조회 및 출력 조건으로 사용되는 속성

55. 다음 설명에 해당하는 용어는 무엇인가?

> 분산 데이터베이스의 4대 목표 중 하나로, 동일 데이터가 여러 곳에 중복되어 있더라도 사용자는 마치 하나의 데이터만 존재하는 것처럼 사용하고, 시스템은 자동으로 여러 자료에 대한 작업을 수행한다.

① 위치 투명성 ② 중복 투명성
③ 병행 투명성 ④ 장애 투명성

56. 데이터베이스에서 데이터 저장 시 데이터의 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법을 무엇이라 하는가?
① Cluster ② Index
③ Constraint ④ Partition

정답 : 39.② 40.② 41.③ 42.③ 43.③ 44.① 45.② 46.④ 47.④ 48.④ 49.① 50.② 51.④ 52.① 53.③ 54.① 55.② 56.①

57. 후보키(Candidate Key)에 대한 설명으로 틀린 것은?
① 기본키로 사용할 수 있는 속성들을 말한다.
② 구성에 따라 후보키가 없는 릴레이션이 존재할 수 있다.
③ 유일성과 최소성을 만족한다.
④ 튜플을 유일하게 식별하기 위해 사용하는 속성들의 부분집합이다.

58. 다음 SQL문의 실행결과를 가장 올바르게 설명한 것은?

```
DROP TABLE 인사 CASCADE;
```

① 인사 테이블을 제거한다.
② 인사 테이블이 참조중이면 제거하지 않는다.
③ 인사 테이블을 참조하는 테이블과 인사 테이블을 제거한다.
④ 인사 테이블을 제거할 지의 여부를 사용자에게 다시 질의한다.

59. 서버와 저장장치를 연결하는 스토리지의 종류 중 하나로, 서버와 저장장치를 연결하는 전용 네트워크를 별도로 구성하는 방식을 무엇이라 하는가?
① DAS ② NAS
③ DES ④ SAN

60. 논리 데이터 모델의 슈퍼타입 속성들을 각각의 서브타입에 추가하여 서브타입들을 물리 데이터 모델의 개별 테이블로 만드는 변환 방법에 대한 설명으로 가장 옳지 않은 것은?
① 각 서브타입 속성들의 선택 사양이 불명확한 경우에 유리하다.
② 처리할 때마다 서브타입 유형을 구분할 필요가 없다.
③ 여러 개의 테이블로 통합하므로 테이블당 크기가 감소하여 전체 테이블 스캔시 유리하다.
④ UID(Unique Identifier)의 유지 관리가 어렵다.

4과목 프로그래밍 언어 활용

61. 다음 중 C언어의 변수명 작성에 대한 설명으로 가장 옳지 않은 것은?
① 대 · 소문자를 구분하지 않는다.
② 영문, 숫자, 언더바 등을 사용할 수 있다.
③ 글자 수에 제한이 없다.
④ 예약어는 변수명으로 사용할 수 없다.

62. 다음 중 C언어에서 사용하는 기억 클래스의 종류가 아닌 것은?
① Automatic Variable ② Dynamic Variable
③ Register Variable ④ Static Variable

63. FIFO 스케줄링에서 3개의 작업 도착시간과 CPU 사용시간(Burst Time)이 다음 표와 같다. 이 때 모든 작업들의 평균 반환시간(Turn Around Time)은? (단, 소수점 이하는 반올림 처리한다.)

작업	도착시간	CPU 사용시간(Burst Time)
JOB 1	0	13
JOB 2	3	35
JOB 3	8	10

① 12 ② 36
③ 58 ④ 69

64. C언어의 포인터 형(Pointer Type)에 대한 설명으로 틀린 것은?
① 변수의 주소를 지정할 때 사용하는 변수를 포인터 변수라고 한다.
② 포인터 변수는 기억장소의 번지를 기억하는 정적 변수이다.
③ 보통 변수의 번지를 참조하려면 번지 연산자 &을 변수 앞에 쓴다.
④ 실행문에서는 간접 연산자 *를 사용하여 포인터 변수가 지시하고 있는 내용을 참조한다.

65. 다음 C 코드의 결과 값은?

```
#include <stdio.h>
main( ) {
    int a = 1, sum = 0;
    while (a++ < 10)
        sum += a * 3 >= sum ? a : 1;
    printf("%d", sum);
}
```

① 11 ② 30
③ 39 ④ 54

66. 다음 JAVA 코드의 결과 값은?

```
public class Test {
    public static void main(String[ ] args) {
        int array[ ] = { 1, 3, 5, 7, 9 };
        for (int a : array)
            if(a % 3 == 0) System.out.printf("*");
    }
}
```

① * ② **
③ *** ④ *****

67. 다음 중 printf( ) 함수의 사용이 잘못된 것은?
① printf("%d", 152);
② printf("%f", "programming");
③ printf("%c", 'e');
④ printf("%e", 1.5e+3);

68. 연산자에 대한 설명으로 가장 옳지 않은 것은?
① 이항 연산자는 대입 연산자보다 먼저 계산한다.
② 시프트 연산자는 결합규칙에 따라 왼쪽부터 계산한다.
③ 대입 연산자는 조건 연산자보다 나중에 계산된다.
④ 단항 연산자에는 ^, !, ++, -- 등이 있다.

69. RR(Round-Robin) 스케줄링에 대한 설명으로 옳지 않은 것은?
① Time slice를 크게 하면 입출력 위주의 작업이나 긴급을 요하는 작업에 신속히 반응하지 못한다.
② Time slice가 작을 경우 FCFS 스케줄링과 같아진다.
③ Time sharing System을 위해 고안된 방식이다.
④ Time slice가 작을수록 문맥교환에 따른 오버헤드가 자주 발생한다.

70. UNIX에서 커널(Kernel)에 대한 설명으로 옳지 않은 것은?
① 프로세스 관리, 기억장치 관리 등을 담당한다.
② UNIX 시스템의 중심부에 해당한다.
③ 컴퓨터 부팅시 주기억장치에 적재되어 상주하면서 실행된다.
④ 사용자와 시스템 간의 인터페이스를 제공한다.

71. 다음 중 주기억장치 관리 전략과 거리가 먼 것은?
① Paging 전략
② Placement 전략
③ Fetch 전략
④ Replacement 전략

72. 주기억장치에 완전히 비어 있는 3개의 페이지가 있다. 페이지 교체 방법으로 FIFO(선입선출)를 사용할 때 다음의 순서로 페이지 번호를 요청할 경우 페이지 부재(Fault)는 몇 번 발생하는가?

0, 1, 2, 3, 2, 1, 4, 0

① 5 ② 6
③ 7 ④ 8

73. 다음 중 워킹 셋(Working Set)에 대한 설명으로 옳지 않은 것은?
① 프로그램의 구역성(Locality) 특징을 이용한다.
② 주기억장치에 적재되지 않으면 스래싱이 발생할 수 있다.
③ 프로세스 실행중에는 크기가 변하지 않는다.
④ 워킹 셋에 속한 페이지를 참조하면 프로세스의 기억장치 사용은 안정상태가 된다.

74. 스레드(Thread)에 대한 설명으로 옳지 않은 것은?
① 스레드는 동일 프로세스 환경에서 서로 독립적인 다중 수행이 불가능하다.
② 스레드는 프로세스의 일부 특성을 갖고 있기 때문에 경량(Light Weight) 프로세스라고도 한다.
③ 사용자 스레드의 경우 사용자가 만든 라이브러리를 사용하여 스레드를 운용한다.
④ 스레드를 사용함으로써 하드웨어, 운영체제의 성능과 응용 프로그램의 처리율을 향상시킬 수 있다.

75. 다음 설명에 해당하는 용어는 무엇인가?

준비 상태에서 대기하고 있는 프로세스 중 하나가 스케줄링되어 중앙처리장치를 할당받아 실행 상태로 전이되는 과정이다.

① 실행(Run)
② 대기(Wait)
③ 준비(Ready)
④ 디스패치(Dispatch)

정답 : 57.② 58.③ 59.④ 60.① 61.① 62.② 63.② 64.② 65.③ 66.② 67.② 68.④ 69.② 70.④ 71.① 72.② 73.③ 74.① 75.④

76. IPv6에 대한 설명으로 틀린 것은?
① 주소의 확장성, 융통성, 연동성이 뛰어나다.
② 더 많은 IP 주소를 지원할 수 있도록 주소의 크기는 64비트이다.
③ 인증성, 기밀성 등의 지원으로 보안 성능이 향상되었다.
④ 유니캐스트, 멀티캐스트, 애니캐스트를 지원한다.

77. OSI 7계층의 데이터 링크 계층 기능에 대한 설명으로 적합하지 않은 것은?
① 종단 시스템 간에 End-to-End 통신을 지원한다.
② 에러 제어, 흐름 제어 등의 기능을 수행한다.
③ 대표적 프로토콜로는 HDLC, LAPB 등이 있다.
④ 프레임의 시작과 끝을 구분하기 위한 프레임의 동기화 기능을 한다.

78. 다음과 같은 기능을 가지고 있는 프로토콜은?

- 양방향 연결과 가상 회선 연결 형태의 서비스를 제공한다.
- 스트림 위주로 데이터를 전달한다.
- 신뢰성 있는 경로를 확립하고 메시지 전송을 감독한다.
- 순서 제어, 오류 제어, 흐름 제어 기능을 제공한다.

① RTCP(Real-Time Control Protocol)
② FTP(File Transfer Protocol)
③ UDP(User Datagram Protocol)
④ TCP(Transmission Control Protocol)

79. 프레임워크(Framework)의 특성 중 제어의 역흐름에 대한 설명으로 옳은 것은?
① 모듈화를 강화하고 설계 및 구현의 변경에 따른 영향을 최소화한다.
② 재사용 가능한 모듈들을 제공하여 생산성을 향상시킨다.
③ 다형성(Polymorphism)을 통해 인터페이스의 확장이 가능하다.
④ 객체들의 제어를 개발자가 아닌 프레임워크가 담당한다.

80. 패키지 소프트웨어와 전용 개발 소프트웨어에 대한 설명으로 틀린 것은?
① 전용 개발 소프트웨어는 패키지 소프트웨어에 비해 인력과 시간이 절약된다.
② 전용 개발 소프트웨어는 개발자의 역량에 따라 품질이 크게 달라진다.
③ 패키지 소프트웨어는 소프트웨어에 대한 라이선스가 판매자에게 있다.
④ 패키지 소프트웨어는 기능 요구사항의 70% 이상을 충족하는 소프트웨어가 존재하는 경우에만 이용하는 것이 좋다.

5과목 정보시스템 구축 관리

81. 다음 중 SPICE(ISO/IEC 15504)의 프로세스 범주에 속하지 않는 것은?
① Customer-Supplier Process
② Engineering Process
③ Maintenance Process
④ Management Process

82. 소프트웨어 개발 방법론의 테일러링(Tailoring)에 대한 설명으로 틀린 것은?
① 프로젝트 상황 및 특성에 맞도록 절차, 사용기법 등을 수정 및 보완한다.
② 프로젝트의 규모, 복잡도, 구성원 등에 따라 적용되는 다양한 테일러링 기법이 존재한다.
③ 목표 환경, 요구사항 등의 내부적 요건을 고려해야 한다.
④ 보유 기술, 프로젝트 규모 등의 외부적 요건을 고려해야 한다.

83. 다음 중 소프트웨어 비용을 결정할 때 사용되는 프로젝트 요소에 대한 설명으로 옳지 않은 것은?
① 제품의 복잡도 : 응용, 유틸리티, 시스템 소프트웨어 등 소프트웨어의 종류에 따라 달라지는 문제의 난이도를 의미한다.
② 시스템의 크기 : 대형, 소형 등 소프트웨어의 규모에 따라 개발할 시스템의 규모를 의미한다.
③ 요구되는 신뢰도 : 정확성, 견고성, 완전성, 일관성 등 프로그램이 일정한 기간 내에 주어진 조건하에서 필요한 기능을 수행하는 정도를 의미한다.
④ 개발 기간 : 소프트웨어를 개발하는 기간을 의미한다.

84. 전문가 감정 기법의 주관적인 편견을 보완하기 위해 많은 전문가의 의견을 종합하여 산정하는 비용 산정 기법은?
① 기능 점수 모형 ② 델파이 기법
③ LOC 기법 ④ COCOMO 모형

85. 원시 코드 라인 수(LOC) 기법에 의하여 예측된 총 라인 수가 25,000라인, 개발에 참여할 프로그래머가 5명, 프로그래머들의 평균 생산성이 월간 500라인일 때 개발에 소요되는 기간은?
① 8개월 ② 10개월
③ 11개월 ④ 20개월

86. 2명의 개발자가 5개월에 걸쳐 15,000 라인의 코드를 개발하였을 때, 월별(Person Month) 생산성 측정을 위한 계산 방식으로 가장 적합한 것은?

① 15,000 / 2 ② 15,000 / (5 × 2)
③ 15,000 / 5 ④ (2 × 15,000) / 5

87. 다음 중 정보통신기술에 대한 설명으로 적당하지 않은 것은?

① 메시 네트워크(Mesh Network)는 차세대 이동통신, 홈네트워킹, 공공 안전 등 특수 목적을 위한 새로운 방식의 네트워크 기술이다.
② 와이선(Wi-SUN)은 스마트 그리드와 같은 장거리 무선 통신을 필요로 하는 사물 인터넷(IoT) 서비스를 위한 저전력 장거리 통신기술이다.
③ 피코넷(PICONET)은 여러 개의 독립된 통신장치가 블루투스 기술이나 UWB 통신기술을 사용하여 통신망을 형성하는 무선 네트워크 기술이다.
④ NFC는 각종 컴퓨팅 자원을 중앙 컴퓨터에 두고 인터넷 기능을 갖는 단말기로 언제 어디서나 인터넷을 통해 컴퓨터 작업을 수행할 수 있는 환경을 의미한다.

88. 다음 중 통신망의 구성 형태에 대한 설명으로 틀린 것은?

① 버스형(Bus)은 기밀 보장이 어렵고, 통신 회선의 길이에 제한이 있다.
② 성형(Star)에 연결된 각 단말장치들은 중앙 컴퓨터를 통하여 데이터를 교환한다.
③ 망형(Mesh)은 많은 단말장치로부터 많은 양의 통신을 필요로 하는 경우에 유리하며, LAN에서 가장 많이 사용되는 방식이다.
④ 계층형(Tree)은 일정 지역의 단말장치까지는 하나의 통신 회선으로 연결하고, 이웃하는 단말장치는 일정 지역 내에 설치된 중간 단말장치로부터 다시 연결하는 형태이다.

89. 다음 설명에 해당하는 용어는 무엇인가?

> 소프트웨어를 제공하는 입장에서는 악의적이지 않은 유용한 소프트웨어라고 주장할 수 있지만 사용자 입장에서는 유용할 수도 있고 악의적일 수도 있는 소프트웨어이다.

① 매시업(Mashup)
② RIA
③ 그레이웨어(Grayware)
④ 뉴럴링크(Neuralink)

90. 현실의 사물을 소프트웨어로 가상화한 모델로, 현실의 사물을 대신해 다양한 상황을 모의 실험하기 위한 용도로 사용되는 것은 무엇인가?

① 디지털 트윈(Digital Twin)
② 소프트웨어 에스크로(Software Escrow)
③ 서비스형 소프트웨어(SaaS)
④ 오픈 그리드 서비스 아키텍처(OGSA)

91. 행정안전부가 보안 활동의 주체로서 수행해야 하는 역할이 아닌 것은?

① 보안 정책을 총괄한다.
② 보안 법규, 제도 등을 정비한다.
③ 보안 가이드를 개발한다.
④ 보안 전문가의 양성을 위한 업무를 수행한다.

92. 직무별로 수행해야하는 보안 활동으로 틀린 것은?

① 프로젝트 관리자는 구성원들에게 개발 소프트웨어의 보안 전략을 이해시켜야 한다.
② 설계자는 소프트웨어의 안전 여부를 확인할 수 있도록 문서화해야 한다.
③ 아키텍트는 아키텍처 설계 시 보안 요구사항을 적용해야 한다.
④ 개발자는 시큐어 코딩 표준을 준수하여 개발해야 한다.

93. 다음 괄호에 공통으로 들어갈 용어는 무엇인가?

> • ( )는 전류의 방향과 양 등 기존의 경험을 모두 기억하는 특별한 소자로, 레지스터(Resister), 커패시터(Capacitor), 인덕터(Inductor)에 이어 네 번째 전자회로 구성 요소라고 불린다.
> • ( )는 전원 공급이 끊어졌을 때도 직전에 통과한 전류의 방향과 양을 기억하기 때문에 다시 전원이 공급되면 기존의 상태가 그대로 복원된다.

① M-DISC ② Memristor
③ MEMS ④ RAID

94. Secure OS의 자원 보호 방법에 대한 설명으로 가장 옳지 않은 것은?

① 프로세스의 논리적 구역을 지정하여 구역을 벗어나는 행위를 제한한다.
② 동일 시간에 하나의 프로세스만 수행되도록 하여 동시 실행으로 발생하는 보안 취약점을 제거한다.
③ 사용자별로 특정 장비만 사용하도록 제한한다.
④ 내부 정보를 암호화 하는 방법이 자원 보호 방법 중 가장 단순한 방법에 속한다.

정답 : 76.② 77.① 78.④ 79.④ 80.① 81.③ 82.④ 83.④ 84.② 85.② 86.② 87.④ 88.③ 89.③ 90.① 91.③ 92.② 93.② 94.④

95. 병행제어 기법에서 병행수행되던 트랜잭션들 중 어느 하나에 문제가 생겨 Rollback하는 경우 다른 트랜잭션도 함께 Rollback되는 현상을 무엇이라 하는가?
① 연쇄 복귀(Cascading Rollback)
② 모순성(Inconsistency)
③ 갱신 분실(Lost Update)
④ 비완료 의존성(Uncommitted Dependency)

96. 소프트웨어 개발에 있어 충족시켜야 할 3대 주요 보안 요소에 속하지 않는 것은?
① 호환성(compatibility)
② 기밀성(Confidentiality)
③ 무결성(Integrity)
④ 가용성(Availability)

97. 데이터 표준화에 대한 설명으로 틀린 것은?
① 데이터 표준화는 중요한 데이터가 저장되는 특정 오브젝트를 대상으로 한다.
② 명칭은 업무적 보편성을 갖는 이름으로 정의한다.
③ 사용 목적과 유사한 데이터는 형식을 일관되게 정의한다.
④ 정합성을 위해 기본값, 허용 범위 등을 사전에 지정한다.

98. 임의의 길이의 입력 데이터나 메시지를 고정된 길이의 값이나 키(key)로 변환하는 알고리즘으로, 데이터 암호화나 무결성 검증 등에 사용되는 것은 무엇인가?
① Rivest Shamir Adleman
② Data Encryption Standard
③ Hash Algorithm
④ Public Key Infrastructure

99. 다음에서 설명하는 네트워크 공격 방법은 무엇인가?

다양한 IT 기술과 방식들을 이용해 조직적으로 특정 기업이나 조직 네트워크에 침투해 활동 거점을 마련한 뒤 때를 기다리면서 보안을 무력화시키고 정보를 수집한 다음 외부로 빼돌리는 형태의 공격이다.

① APT ② Smishing
③ Qshing ④ DDoS Attack

100. 침입 탐지 시스템(IDS)에 대한 설명으로 틀린 것은?
① 컴퓨터 시스템의 비정상적인 사용, 오용, 남용 등을 실시간으로 탐지하는 시스템이다.
② 비정상적인 트래픽을 능동적으로 차단하고 격리하는 등의 방어 조치를 취한다.
③ 패킷이 라우터(Router)에 들어오기 전 네트워크에 시도되는 모든 공격의 탐지가 가능하다.
④ 평균적인 시스템의 상태를 기준으로 비정상적인 행위나 자원의 사용이 감지되면 이를 알려준다.

정답 : 95.① 96.① 97.① 98.③ 99.① 100.②

정보처리기사 필기 최종점검 모의고사 2회

1과목 소프트웨어 설계

1. 애자일 모형의 실무 적용 시 준수해야 할 사항이 아닌 것은?
① 유용한 소프트웨어를 빠르고, 지속적으로 제공하여 고객을 만족시킨다.
② 개발 막바지라도 요구사항 변경을 적극 수용한다.
③ 최소한 몇 개월 단위로 실행되는 소프트웨어를 제공한다.
④ 기술적 우수성과 좋은 설계에 지속적인 관심을 기울이면 민첩성이 향상된다.

2. 다음이 설명하고 있는 생명주기 모델은?

> • 여러 번의 개발 과정을 거쳐 완벽한 최종 소프트웨어를 개발한다.
> • 점진적으로 개발 과정이 반복되므로 누락되거나 추가된 요구사항을 첨가할 수 있다.

① 애자일 모형(Agile Model)
② 프로토타입 모형(Prototype Model)
③ 나선형 모형(Spiral Model)
④ 폭포수 모형(Waterfall Model)

3. 다음 중 스크럼(Scrum)에 대한 설명으로 잘못된 것은?
① 팀원들이 스스로 팀을 구성하며, 개발 작업의 모든 것을 스스로 해결할 수 있어야 한다.
② 개발에 필요한 요구사항에 우선순위를 부여한 제품기능 목록(Product Backlog)을 작성한다.
③ 소멸(Burn-down) 차트를 통해 작업의 진행 상황을 확인할 수 있다.
④ 스프린트 검토 회의(Sprint Review)를 통해 스프린트 동안 발생한 문제점을 파악하고 이에 대한 해결 방안을 모색한다.

4. 다음 중 XP의 주요 실천 방법에 대한 설명 중 옳지 않은 것은?
① 개발자가 실제 코드를 작성하기 전에 테스트 케이스를 먼저 작성한다.
② 릴리즈 기간을 길게 반복하여 고객의 요구 변화에 신속히 대응한다.
③ 모듈 단위로 나눠서 개발된 코드들은 하나의 작업이 마무리될 때마다 지속적으로 통합된다.
④ 다른 사람과 함께 프로그래밍을 수행함으로써 개발에 대한 책임을 공동으로 나눠 갖는 환경을 조성한다.

5. 다음 UML의 구성 요소 중 관계에 대한 설명으로 옳지 않은 것은?
① 포함(Composition) 관계 : 하나의 사물이 다른 사물에 포함되어 있는 관계로 두 사물이 서로 독립적이다.
② 실체화(Realization) 관계 : 사물이 할 수 있거나 해야 하는 기능으로 서로를 그룹화 할 수 있는 관계를 표현한다.
③ 일반화(Generalization) 관계 : 하나의 사물이 다른 사물에 비해 더 일반적인지 구체적인지를 표현한다.
④ 의존(Dependency) 관계 : 사물 사이에 서로 연관은 있으나 필요에 의해 서로에게 영향을 주는 짧은 시간 동안만 연관을 유지하는 관계를 표현한다.

6. 다음이 설명하고 있는 UML 다이어그램은?

> 결과물, 프로세스, 컴포넌트 등 물리적인 자원의 위치를 표시하는 것으로, 구현 단계에서 사용된다.

① 배치 다이어그램(Deployment Diagram)
② 순차 다이어그램(Sequence Diagram)
③ 객체 다이어그램(Object Diagram)
④ 상태 다이어그램(State Diagram)

7. 다음 중 사용자 인터페이스(UI)를 설계할 때 고려할 사항으로 거리가 먼 것은?
① 사용자가 쉽게 이해하고 편리하게 사용할 수 있는 환경을 제공해야 한다.
② 메인 화면에 주요 기능을 노출시켜 최대한 조작이 쉽도록 설계해야 한다.
③ 버튼이나 조작 방법 등을 새로운 방식으로 제공해야한다.
④ 사용자가 개념적으로 쉽게 인지할 수 있도록 설계해야 한다.

8. 다음 중 사용자 인터페이스를 설계할 때 사용하는 도구로 옳지 않은 것은?
① 와이어프레임(Wireframe)
② 이클립스(Eclipse)
③ 유스케이스(Use Case)
④ 스토리보드(Story Board)

정답 : 1.③ 2.③ 3.④ 4.② 5.① 6.① 7.③ 8.②

9. 다음 중 유스케이스(Use Case)에 관한 설명으로 잘못된 것은?
① 완성된 유스케이스에 대해 유스케이스 명세서를 작성한다.
② 사용자의 요구사항을 정리하고 기록하기 위한 도구이다.
③ 유스케이스는 일반적으로 다이어그램 형식으로 작성된다.
④ 현재 진행 상태 등을 공유하기 위해 사용한다.

10. 소프트웨어 아키텍처 설계의 기본 원리 중 추상화(Abstraction)에 대한 설명으로 가장 옳지 않은 것은?
① 문제의 전체적이고 포괄적인 개념을 설계한 후 차례로 세분화하여 구체화시켜 나가는 것이다.
② 추상화는 최소의 비용으로 실제 상황에 대처할 수 있고, 시스템의 구조 및 구성을 대략적으로 파악할 수 있게 해준다.
③ 데이터 추상화는 자세한 수행 과정을 정의하지 않고, 전반적인 흐름만 파악할 수 있게 설계하는 방법이다.
④ 제어 추상화는 이벤트 발생의 정확한 절차나 방법을 정의하지 않고, 대표할 수 있는 표현으로 대체하는 방법이다.

11. 다음 중 UI 프로토타입의 장 · 단점에 대한 설명으로 잘못된 것은?
① 사용자를 설득하고 이해시키기 쉽다.
② 사전에 오류를 발견할 수 있다.
③ 반복적인 개선 및 보완 작업이 없어 작업 시간을 최소화할 수 있다.
④ 부분적으로 프로토타이핑을 진행하다보면 중요한 작업이 생략될 수 있다.

12. 소프트웨어 아키텍처의 설계 과정을 순서대로 가장 옳게 나열한 것은?

㉠ 시스템과 서브시스템의 타입을 결정한다. ㉡ 서브시스템의 기능과 서브시스템 간의 인터페이스를 정의한다. ㉢ 표준 아키텍처를 설계한다. ㉣ 요구사항을 분석하여 전체 시스템의 설계 목표를 설정한다.

① ㉣ → ㉠ → ㉡ → ㉢　② ㉣ → ㉡ → ㉠ → ㉢
③ ㉣ → ㉢ → ㉠ → ㉡　④ ㉣ → ㉠ → ㉢ → ㉡

13. 다음 중 아키텍처 패턴(Architecture Pattern)에 관한 설명 중 가장 옳지 않은 것은?
① 검증된 구조로 개발하기 때문에 오류가 적어 개발시간을 단축할 수 있다.
② 시스템의 구조를 이해하는 것이 어려워 개발에 참여한 개발자에 의해서만 유지보수가 수행될 수 있다.
③ 서브시스템들에 대한 역할을 정의하고 있으며, 그들 간의 인터페이스에 대한 지침 등이 포함되어 있다.
④ 이해관계자들이 공통된 아키텍처를 공유할 수 있어 의사소통이 간편해진다.

14. 다음 중 객체(Object)에 대한 특성을 설명한 것으로 가장 옳지 않은 것은?
① 식별성을 가진다.
② 상호 연관성에 의한 관계가 형성된다.
③ 행위에 대한 특징은 나타낼 수 없다.
④ 일정한 기억장소를 가지고 있다.

15. 다음이 설명하고 있는 결합도(Coupling)는?

모듈이 파라미터나 인수로 다른 모듈에게 데이터를 넘겨주고 호출 받은 모듈은 받은 데이터에 대한 처리 결과를 다시 돌려주는 유형의 모듈 결합도이다.

① 내용 결합도(Content Coupling)
② 외부 결합도(External Coupling)
③ 제어 결합도(Control Coupling)
④ 데이터 결합도(Data Coupling)

16. 다음 중 모듈의 응집도(Cohesion)에 대한 설명 중 틀린 것은?
① 시간적 응집도는 특정 시간에 처리되는 몇 개의 기능을 모아 하나의 모듈로 작성할 경우의 응집도이다.
② 우연적 응집도는 한 모듈 내부의 한 기능 요소에 의한 출력 자료가 다음 기능 요소의 입력 자료로서 제공되는 형태의 응집도이다.
③ 기능적 응집도는 모듈 내부의 모든 기능 요소들이 단일 문제와 연관되어 수행될 경우의 응집도이다.
④ 논리적 응집도는 유사한 성격을 갖거나 특정 형태로 분류되는 처리 요소들로 하나의 모듈이 형성되는 경우의 응집도이다.

17. 운영체제와 해당 운영체제에서 실행되는 응용 프로그램 사이에서 운영체제가 제공하는 서비스 이외에 추가적인 서비스를 제공하는 소프트웨어를 무엇이라 하는가?
① Middleware　② ORM
③ OOP　④ Interface

18. 송 · 수신 데이터의 처리 방식 중 대량의 데이터를 처리할 때 사용하는 방식은?
① 실시간 방식　② 분산 처리 방식
③ 배치 방식　④ 지연 처리 방식

19. 다음 중 미들웨어(Middleware)들의 개별적인 특징으로 옳지 않은 것은?
① RPC : 항공기나 철도 예약 업무 등과 같은 온라인 트랜잭션 업무에서 트랜잭션을 처리 및 감시하는 미들웨어이다.
② MOM : 메시지 기반의 비동기형 메시지를 전달하는 방식의 미들웨어이다.
③ ORB : 객체 지향 미들웨어로, 코바(CORBA) 표준 스펙을 구현한 미들웨어이다.
④ DB : 데이터베이스 벤더에서 제공하는 클라이언트에서 원격의 데이터베이스와 연결하기 위한 미들웨어이다.

20. 다음 중 GoF의 디자인 패턴에 대한 설명으로 틀린 것은?
① 수많은 디자인 패턴들 중 가장 일반적인 사례에 적용될 수 있는 패턴들을 분류하여 정리한 것이다.
② 디자인 패턴은 서브시스템에 속하는 컴포넌트들과 그 관계를 설계하기 위한 참조 모델을 말한다.
③ 디자인 패턴은 아키텍처 패턴보다 상위 수준의 설계에 사용된다.
④ 디자인 패턴은 유형에 따라 생성, 구조, 행위로 구분한다.

2과목 소프트웨어 개발

21. 자료들을 반드시 연속적으로 배열시키지는 않고 임의의 기억공간에 기억시키되, 자료 항목의 순서에 따라 노드의 포인터 부분을 이용하여 서로 연결시킨 자료 구조를 무엇이라고 하는가?
① 배열(Array)
② 연속 리스트(Contiguous List)
③ 연결 리스트(Linked List)
④ 스택(Stack)

22. 데이터베이스의 정의 중 통합된 데이터(Integrated Data)에 대한 설명으로 옳은 것은?
① 컴퓨터가 접근할 수 있는 저장 매체에 저장된 자료이다.
② 자료의 중복을 배제한 데이터의 모임이다.
③ 조직의 고유한 업무를 수행하는 데 존재 가치가 확실하고 없어서는 안 될 반드시 필요한 자료이다.
④ 여러 응용 시스템들이 공동으로 소유하고 유지하는 자료이다.

23. 다음 전위식(Prefix)을 후위식(postfix)으로 옳게 표현한 것은?

− / * A + B C D E

① A B C + * D / E −
② A B * C D / + E −
③ A B * C + D / E −
④ A B C + D / * E −

24. 다음 중 인수 테스트(Acceptance Test)에 대한 설명으로 가장 옳지 않은 것은?
① 사용자의 요구사항을 충족하는지에 중점을 두고 테스트한다.
② 문제가 없으면 사용자는 소프트웨어를 인수하게 되고, 프로젝트는 종료된다.
③ 개발한 소프트웨어가 인수되기 전 개발자가 수행하는 테스트이다.
④ 종류에는 사용자 인수 테스트, 알파 테스트, 베타 테스트 등이 있다.

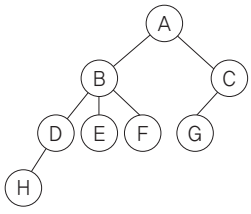
25. 다음 중 테스트 시나리오의 작성 방법에 대한 설명 중 틀린 것은?
① 소프트웨어를 하나의 시나리오로 통합하여 작성해야 한다.
② 사용자의 요구사항과 설계 문서 등을 토대로 작성해야 한다.
③ 유스케이스(Use Case) 간 업무 흐름이 정상적인지 확인할 수 있도록 작성해야 한다.
④ 모듈 간의 연계가 정상적인지 확인할 수 있도록 작성해야 한다.

26. 다음 중 결함 관리 프로세스에 대한 설명으로 가장 옳지 않은 것은?
① 전체 프로세스에 대한 결함 관리 일정, 인력, 업무 프로세스 등을 확보하여 계획을 수립해야 한다.
② 결함 관리 DB를 구축하여 발견된 결함을 관리한다.
③ 결함은 테스터, 개발팀 리더, QA 담당자 등이 검토한 후 개발자에게 전달된다.
④ DB에 기록된 결함들은 '수정 → 검토 → 재확인' 순서로 처리된다.

27. 애플리케이션의 성능을 측정하기 위한 테스트 중 하나로, 애플리케이션이 과부하 상태에서 어떻게 작동하는지 확인하는 테스트는 무엇인가?
① 부하 테스트 ② 스트레스 테스트
③ 연계 테스트 ④ 침투 테스트

정답 : 9.④ 10.③ 11.③ 12.④ 13.② 14.③ 15.④ 16.② 17.① 18.③ 19.① 20.③ 21.③ 22.② 23.① 24.③ 25.① 26.④ 27.②

28. 다음 그림에서 트리의 차수는?



① 3 ② 4
③ 6 ④ 8

29. 다음 설명에 해당하는 소프트웨어는 무엇인가?

> 접속자, 자원 현황, 트랜잭션 수행 내역, 장애 진단 등 다양한 모니터링 기능을 제공하는 소프트웨어로, 리소스 방식과 엔드투엔드(end-to-end) 방식 두 가지 유형으로 구분된다.

① APM ② ESB
③ EAI ④ Optimizer

30. 다음 중 데이터의 보안, 무결성, 회복, 병행 수행 제어 등을 정의하는 데 사용되는 SQL 언어를 무엇이라 하는가?

① RDB(Relational Database)
② DDL(Data Define Language)
③ DML(Data Manipulation Language)
④ DCL(Data Control Language)

31. 디지털 저작권 관리(DRM)에 대한 다음 설명에서 괄호에 들어갈 가장 적합한 용어는 무엇인가?

> (　　)는 저작권에 대한 사용 권한, 라이선스 발급, 사용량에 따른 결제 관리 등을 수행하는 곳으로, 배포하려는 소프트웨어의 패키징을 수행하면 콘텐츠에는 암호화된 저작권자의 전자서명이 포함되고 저작권자가 설정한 라이선스 정보가 (　　)에 등록된다.

① 패키저(Packager)
② 클리어링 하우스(Clearing House)
③ 콘텐츠 분배자(Contents Distributor)
④ DRM 컨트롤러(DRM Controller)

32. 소프트웨어 버전 관리 도구 중 분산 저장소 방식에 대한 설명으로 틀린 것은?

① 버전 관리 자료가 하나의 원격 저장소와 분산된 개발자 PC의 로컬 저장소에 함께 저장되어 관리되는 방식이다.
② 로컬 저장소에서 버전 관리가 가능하므로 원격 저장소에 문제가 생겨도 로컬 저장소의 자료를 이용하여 작업 할 수 있다.
③ 파일을 잘못 복사하거나 다른 위치로 복사하는 것에 대비하기 위해 파일의 변경 사항을 데이터베이스에 기록하여 관리한다.
④ 종류에는 Git, GNU arch, DCVS, Bazaar 등이 있다.

33. 다음 중 블랙박스(Black Box) 테스트에 대한 설명으로 가장 옳지 않은 것은?

① 각 기능이 완전히 작동되는 것을 입증하는 테스트로, 기능 테스트라고도 한다.
② 소프트웨어 인터페이스에서 실시되는 테스트이다.
③ 프로그램의 제어 구조에 따라 선택, 반복 등의 분기점 부분들을 수행함으로써 논리적 경로를 제어한다.
④ 종류에는 동치 분할 검사, 경계값 분석, 원인-효과 그래프 검사, 오류 예측 검사, 비교 검사 등이 있다.

34. 다음 설명에 해당하는 테스트의 종류는 무엇인가?

> 대표적인 화이트박스 테스트 기법의 하나로, 테스트 케이스 설계자가 절차적 설계의 논리적 복잡성을 측정할 수 있게 해주는 테스트이다. 이 테스트의 결과는 실행 경로의 기초를 정의하는 지침으로 사용된다.

① 기초 경로 검사 ② 제어 구조 검사
③ 동치 분할 검사 ④ 오류 예측 검사

35. 다음 중 하향식 통합 테스트(Top Down Integration Test)에 대한 설명으로 가장 옳지 않은 것은?

① 상위 모듈에서 하위 모듈 방향으로 통합하면서 테스트하는 기법이다.
② 테스트 초기부터 사용자에게 시스템 구조를 보여줄 수 있다.
③ 상위 모듈에서는 테스트 케이스를 사용하기 어렵다.
④ 주요 제어 모듈의 종속 모듈들은 클러스터(Cluster)로 대체한다.

36. 다음 설명에 해당하는 모듈 연계 방법은 무엇인가?

> • 기업 내 각종 애플리케이션 및 플랫폼 간의 정보 전달, 연계, 통합 등 상호 연동이 가능하게 해주는 솔루션이다.
> • 비즈니스 간 통합 및 연계성을 증대시켜 효율성 및 각 시스템 간의 확정성(Determinacy)을 높인다.
> • 서비스 보다는 애플리케이션 중심의 통합이 이루어진다.

① ESB(Enterprise Service Bus)
② EAI(Enterprise Application Integration)
③ IDE(Integrated Development Environment)
④ Middleware

37. 다음 중 소프트웨어 패키징에 대한 설명으로 가장 옳지 않은 것은?
① 모듈별로 생성한 실행 파일들을 묶어 배포용 설치 파일을 만드는 것이다.
② 다양한 환경에서 소프트웨어를 손쉽게 사용할 수 있도록 일반적인 배포 형태로 패키징한다.
③ 다양한 사용자의 요구사항을 반영할 수 있도록 패키징의 변경 및 개선에 대한 관리를 항상 고려한다.
④ 개발자를 중심으로 작업이 진행되므로 개발자의 편의성 및 실행 환경을 우선적으로 고려해야 한다.

38. 애플리케이션 테스트에서 동일한 테스트를 반복하면 더 이상 결함이 발견되지 않는 현상을 가리키는 용어는?
① 결함 집중(Defect Clustering)
② 파레토 법칙(Pareto Principle)
③ 살충제 패러독스(Pesticide Paradox)
④ 오류-부재의 궤변(Absence of Errors Fallacy)

39. 다음 중 인터페이스 보안 기능에 대한 설명으로 가장 옳지 않은 것은?
① 인터페이스 기능과 보안 취약점을 기반으로 네트워크, 애플리케이션, 데이터베이스 영역에 인터페이스 보안 기능을 적용한다.
② 데이터 탈취 및 변조 위협을 방지하기 위해 네트워크 트래픽에 대한 암호화를 설정한다.
③ 애플리케인션 코드 상의 보안 취약점을 보완하여 애플리케이션 영역에 보안 기능을 적용한다.
④ 데이터베이스에서 민감한 데이터의 암호화를 위해 IPSec, SSL, S-HTTP 등의 방법을 사용한다.

40. 코드의 자동 생성, 컴파일, 빌드, 디버그, 라이브러리 관리 등 개발을 위한 다양한 기능이나 툴을 하나의 인터페이스로 제공하는 소프트웨어는 무엇인가?
① 빌드 도구(Build Tool)
② 통합 개발 환경(IDE) 도구
③ 형상 관리(Configuration Management) 도구
④ 그룹웨어(Groupware)

3과목 데이터베이스 구축

41. 다음 중 데이터베이스 설계에 대한 설명으로 잘못된 것은?
① 사용자의 요구를 분석하여 특정 DBMS로 데이터베이스를 구축하는 과정이다.
② 무결성, 일관성 회복, 보안 등의 다양한 요소들을 고려해야 한다.
③ 데이터 모델링은 현실의 자료를 논리적 자료 구조로 변환시키는 과정이다.
④ 데이터 구조화는 트랜잭션의 인터페이스와 테이블을 설계하는 단계이다.

42. 다음 중 데이터 모델의 개체(Entity)에 대한 설명으로 틀린 것은?
① 속성은 개체가 가지고 있는 특성이다.
② 개체 타입은 속성으로만 기술된 개체의 정의를 의미한다.
③ 개체 인스턴스는 개체를 구성하고 있는 각 속성들이 값을 가져 하나의 개체를 나타내는 것을 의미한다.
④ 속성을 개체 어커런스(Occurrence)라고도 한다.

43. 다음 중 릴레이션(Relation)의 특징에 대한 설명으로 가장 옳지 않은 것은?
① 한 릴레이션에는 동일한 튜플이 존재할 수 있다.
② 속성의 명칭은 유일해야 하지만, 속성을 구성하는 값은 동일한 값이 있을 수 있다.
③ 속성의 값은 논리적으로 더 이상 쪼갤 수 없는 원자값만을 저장한다.
④ 릴레이션 스키마를 구성하는 속성들 간의 순서는 중요하지 않다.

44. 다음 설명에 해당하는 키의 종류는 무엇인가?

- 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미한다.
- 참조되는 릴레이션의 기본키와 대응되어 릴레이션 간에 참조 관계를 표현하는데 중요한 도구이다.
- 참조 릴레이션의 기본키에 없는 값은 입력할 수 없다.

① 후보키(Candidate Key)
② 외래키(Foreign Key)
③ 기본키(Primary Key)
④ 슈퍼키(Super Key)

정답 : 28.① 29.① 30.④ 31.② 32.③ 33.③ 34.① 35.④ 36.② 37.④ 38.③ 39.④ 40.② 41.④ 42.④ 43.① 44.②

45. 〈ORDERS〉 릴레이션과 〈CUSTOMERS〉 릴레이션에 공통으로 존재하는 'OC' 속성을 기준으로 두 릴레이션을 합치고자 할 때, 이를 관계대수식으로 올바르게 표현한 것은?

① ORDERS $\bowtie_{OC=OC}$ CUSTOMERS
② π_{OC}(ORDERS) $\cap$ π_{OC}(CUSTOMERS)
③ ORDERS[OC ÷ OC]CUSTOMERS
④ π_{OC}(ORDERS) $\cup$ π_{OC}(CUSTOMERS)

46. 다음 중 제 3정규형에서 BCNF 수행 시의 작업으로 옳은 것은?

① 이행적 함수 종속성 제거
② 다치 종속 제거
③ 모든 결정자가 후보키가 되도록 분해
④ 부분 함수 종속성 제거

47. 의도적으로 정규화 원칙을 위배하여 시스템의 성능 향상, 개발 및 운영의 편의성 등을 위해 정규화된 데이터 모델을 통합, 중복, 분리하는 과정을 무엇이라 하는가?

① 정규화(Normalization)
② 반정규화(Denormalization)
③ 조인(Join)
④ 분할(Partitioning)

48. 다음 중 인덱스(Index) 설계 시 고려해야 할 사항으로 가장 옳지 않은 것은?

① 새로 추가되는 인덱스는 기존 액세스 경로에 영향을 미칠 수 있다.
② 인덱스를 지나치게 많이 만들면 오버헤드가 발생한다.
③ 인덱스를 위한 추가적인 저장 공간은 필요하지 않다.
④ 인덱스와 테이블 데이터의 저장 공간이 분리되도록 설계해야 한다.

49. 다음 설명에 해당하는 데이터베이스 개체는 무엇인가?

> • 사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블이다.
> • 기본 테이블로부터 유도된 테이블이기 때문에 기본 테이블과 같은 형태의 구조를 사용하며, 조작도 기본 테이블과 거의 같다.

① 뷰(View) ② 인덱스(Index)
③ 클러스터(Cluster) ④ 파티션(Partition)

50. 다음 중 클러스터(Cluster)에 대한 설명으로 가장 옳지 않은 것은?

① 데이터 액세스 효율을 향상시키기 위해 데이터 접근이 많은 테이블을 여러 곳에 분산 저장하는 물리적 저장 방법이다.
② 클러스터링키로 지정된 컬럼 값의 순서대로 저장된다.
③ 여러 개의 테이블이 하나의 클러스터에 저장된다.
④ 클러스터링 된 테이블은 데이터 조회 속도가 향상되지만 데이터 입력, 수정, 삭제에 대한 성능은 저하된다.

51. 범위 분할로 분할한 다음 해시 함수를 적용하여 다시 분할하는 파티셔닝 방식으로, 범위 분할한 파티션이 너무 커서 관리가 어려울 때 유용한 파티셔닝 방식은 무엇인가?

① 범위 분할(Range Partitioning)
② 해시 분할(Hash Partitioning)
③ 조합 분할(Composite Partitioning)
④ 인덱스 분할(Index Partitioning)

52. 다음 중 분산 데이터베이스에 대한 설명으로 틀리 것은?

① 분산 데이터베이스를 구성하기 위해서는 자체적으로 처리 능력이 있는 분산 처리기가 필요하다.
② 사이트나 회선의 고장으로부터 회복할 수 있는 기능이 있어야 한다.
③ 네트워크를 통해 원격 접근 기능이 있어야 한다.
④ DBMS가 수행할 기능이 복잡해지지만 개발 및 처리 비용이 감소한다.

53. 다음 〈회원〉 테이블에서 "SELECT COUNT(*) FROM 회원 GROUP BY 가입방식 ORDER BY 가입방식 DESC;"를 수행했을 때 나타날 결과로 알맞은 것은?

〈회원〉

아이디	성별	가입일	가입방식
bakdows	남	2017-04-03	인터넷
nanacle	여	2018-11-05	매장
se7en34	여	2018-05-30	매장
ori_2132	남	2019-09-29	인터넷
dmsgud33	남	2019-12-15	인터넷

①

COUNT(*)
5

②

COUNT(*)
3
2

③

COUNT(*)
2
3

④

가입방식	COUNT(*)
매장	2
인터넷	3

54. 사용자 ID가 "guest03"인 사람에게 〈EMPLIST〉 테이블에 대한 조회 권한을 부여하는 SQL문으로 옳은 것은?

① GRANT SELECT ON EMPLIST TO guest03;
② GRANT SELECT ON guest03 TO EMPLIST;
③ GRANT ALL ON EMPLIST TO guest03;
④ GRANT ALL ON guest03 TO EMPLIST;

55. 다음 SQL문에 대한 설명으로 가장 옳지 않은 것은?

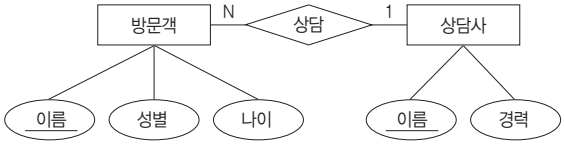
```
CREATE TABLE 상품(
    상품명 VARCHAR(10) PRIMARY KEY,
    단가 INT,
    재고량 INT,
    제조사 VARCHAR(20),
    FOREIGN KEY(제조사) REFERENCES 제조처(이름)
        ON DELETE SET NULL);
```

① '상품명' 속성을 기본키로 하는 테이블이다.
② '단가'와 '재고량' 속성은 정수를 저장한다.
③ '제조사' 속성은 외래키이다.
④ 〈제조처〉 테이블의 '이름' 속성이 변경되면 '제조사' 속성의 값이 NULL로 변경된다.

56. 다음 중 트리거(Trigger)에 대한 설명으로 틀린 것은?

① 데이터 변경 및 무결성 유지, 로그 메시지 출력 등의 목적으로 사용된다.
② 트리거의 생성문에는 반드시 값을 반환하는 RETURN 명령어가 사용되어야 한다.
③ 데이터의 삽입, 갱신, 삭제 등의 이벤트가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL이다.
④ CREATE TRIGGER 명령어를 통해 생성된다.

57. 다음 E-R 다이어그램에 대한 설명으로 가장 옳지 않은 것은?



① '방문객'과 '상담사'는 개체 타입이다.
② '이름', '성별', '나이', '경력'은 속성에 해당한다.
③ '상담'은 '방문객'과 '상담사' 개체의 관계를 의미한다.
④ '방문객'과 '상담사'의 '상담' 관계는 일 대 다의 관계, 즉 한 사람의 방문객이 다수의 상담사를 만날 수 있음을 의미한다.

58. 다음 설명에 해당하는 DBMS 접속 기술은 무엇인가?

> Java 언어로 다양한 종류의 데이터베이스에 접속하고 SQL문을 수행할 때 사용되는 표준 API로, 1997년 2월 썬 마이크로시스템에서 개발하였다. Java SE에 포함되어 있으며 접속하려는 DBMS에 대한 드라이버가 필요하다.

① JDBC ② ODBC
③ ORM ④ MyBatis

59. 다음 중 쿼리문의 성능을 최적화하기 위한 방법으로 가장 옳지 않은 것은?

① WHERE 절을 추가하여 일부 레코드만 조회하게 함으로써 조회에 들어가는 비용을 줄인다.
② 서브 쿼리에 특정 데이터가 존재하는지 확인할 때는 EXISTS보다 IN을 활용한다.
③ 옵티마이저의 실행 계획이 잘못되었다고 판단되는 경우 힌트를 활용하여 실행 계획의 액세스 경로 및 조인 순서를 변경한다.
④ 실행 계획을 참고하여 인덱스를 추가하거나 기존 인덱스의 열 순서를 변경한다.

60. 다음 중 데이터 전환(Data Migration)에 대한 설명으로 틀린 것은?

① 추출(Extraction), 변환(Transformation), 적재(Load) 과정의 앞 글자를 따 ETL이라고도 한다.
② 운영 중인 기존 정보 시스템에 축적되어 있는 데이터를 추출하여 새로 개발할 정보 시스템에서 운영 가능하도록 변환한 후, 적재하는 일련의 과정이다.
③ 데이터 전환 계획서를 작성하여 대상 및 범위, 환경, 담당 조직 및 역할, 일정 등을 상세하게 기록해야 한다.
④ 데이터 전환 계획서에 기재되는 내용 중 비상 계획은 데이터 전환 과정 중에 오류가 발생할 경우에 대비한 데이터의 백업 및 복구 방안을 의미한다.

정답 : 45.① 46.③ 47.② 48.③ 49.① 50.① 51.③ 52.④ 53.② 54.① 55.④ 56.② 57.④ 58.① 59.② 60.④

4과목 프로그래밍 언어 활용

61. 웹 서버(Web Server)의 기능 중 하나로, 네트워크 트래픽의 포화를 방지하기 위해 응답 속도를 제한하는 기능을 무엇이라 하는가?

① HTTP/HTTPS ② 정적 파일 관리
③ 대역폭 제한 ④ 가상 호스팅

62. 다음 중 프레임워크(Framework)에 대한 설명으로 가장 옳지 않은 것은?

① 추상화를 통해 설계 및 구현의 변경에 따른 영향을 최소화함으로서 소프트웨어의 품질을 향상시킨다.
② 재사용 가능한 모듈들을 제공함으로써 생산성을 향상시킨다.
③ 인터페이스 확장을 통해 다양한 형태와 기능을 가진 애플리케이션 개발이 가능하다.
④ 객체들의 제어를 프레임워크에 넘길 수 있다.

63. 다음 중 C언어에서 사용하는 기본적인 데이터 형식이 아닌 것은?

① char ② boolean
③ int ④ long double

64. 자주 사용되는 변수의 처리속도를 높이기 위해 사용하는 기억 클래스로, 메모리가 아닌 CPU 내부의 저장장소를 기억영역으로 할당받는 변수는 무엇인가?

① 자동 변수(Automatic Variable)
② 외부 변수(External Variable)
③ 정적 변수(Static Variable)
④ 레지스터 변수(Register Variable)

65. 다음 C 코드의 결과 값은?

```
#include <stdio.h>
main( ) {
    int a = 3, b = 5;
    int *c = &a;
    *c += ++b;
    printf("%d", a + b + *c);
}
```

① 11 ② 19
③ 23 ④ 24

66. 문자 배열 c[5]에 5개의 문자를 scanf( ) 함수를 이용해 입력받고자 할 때 함수의 사용이 올바른 것은?

① scanf("%5c", c);
② scanf("%5d", &c);
③ scanf("%5e", &c);
④ scanf("%.5f", c);

67. 다음은 정수 배열 a에 0부터 9까지의 무작위 수를 저장하는 Java 코드이다. 괄호에 들어갈 알맞은 코드는 무엇인가?

```
import (    ).*;
public class Test {
    public static void main(String[ ] args) {
        Random rand = new Random( );
        int a[ ] = new int[6];
        for(int i = 0; i < 6; i++)
            a[i] = rand.nextInt(10);
    }
}
```

① java.lang ② java.util
③ java.io ④ java.net

68. C, Java 언어 등에서 사용되는 예약어로, 반복문 안에서 이후의 문장을 실행하지 않고 제어를 반복문의 처음으로 옮기는 명령어는 무엇인가?

① break ② continue
③ next ④ release

69. 다음 중 운영체제(OS)의 기능에 대한 설명으로 가장 옳지 않은 것은?

① 각종 입 · 출력 장치를 관리 및 제어한다.
② 시스템의 오류를 검사하고 복구한다.
③ 이기종 분산 데이터 시스템의 데이터 동기를 지원한다.
④ 사용자와 시스템 간의 편리한 인터페이스를 제공한다.

70. 다음 설명에 해당하는 것은 무엇인가?

> 사용자의 명령어를 인식하여 프로그램을 호출하고 명령을 수행하는 UNIX의 명령어 해석기이다. 시스템과 사용자 간의 인터페이스를 담당하며, 파이프라인 기능을 지원하고 입 · 출력 재지정을 통해 출력과 입력의 방향을 변경할 수 있다.

① 파이프(Pipe) ② 유틸리티(Utility)
③ 커널(Kernel) ④ 쉘(Shell)

71. 주기억장치에 완전히 비어 있는 3개의 페이지가 있다. 페이지 교체 방법으로 LRU를 사용할 때 요청된 페이지 번호의 순서가 0, 1, 2, 0, 3, 1, 0, 2인 경우 페이지 부재(Fault)는 몇 번 발생하는가?

① 5 ② 6
③ 7 ④ 8

72. 다음과 같이 저장장치에 작업이 배치되어 있고 9K의 작업이 공간 할당을 요구할 때, 최적 적합(Best Fit) 전략을 사용한다면 어느 주소에 배치되는가?

영역A	8K
영역B	12K(사용중)
영역C	10K
영역D	15K

① 영역A ② 영역B
③ 영역C ④ 영역D

73. 다중 프로그래밍의 정도가 높아짐에 따라 주로 나타나는 현상으로, 프로세스 처리 시간보다 페이지 교체에 소요되는 시간이 더 많아지는 것을 무엇이라 하는가?

① 구역성(Locality)
② 워킹 셋(Working set)
③ 페이지 부재(Page Fault)
④ 스래싱(Thrashing)

74. 다음 중 프로세스에 대한 설명으로 가장 옳지 않은 것은?

① 프로세서가 할당되는 실체로서, 디스패치가 가능한 단위이다.
② 한 프로그램에 속한 작은 프로그램으로, 부프로그램이라고도 한다.
③ 프로세스에 대한 중요한 정보를 저장해 놓는 곳을 PCB라고 한다.
④ 프로세스 내의 작업 단위를 스레드라고 한다.

75. 다음 중 IP(Internet Protocol) 주소에 대한 설명으로 가장 옳지 않은 것은?

① A, B, C 클래스만이 네트워크 주소와 호스트 주소 체계의 구조를 가진다.
② D 클래스 주소는 유니캐스팅(Unicasting)을 사용하기 위해 예약되어 있다.
③ E 클래스는 실험적 주소로 공용되지 않는다.
④ 5개의 클래스(A, B, C, D, E)로 분류되어 있다.

76. OSI 참조 모델에서 시스템들 간에 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 흐름 제어, 프레임 동기화, 오류 제어, 순서 제어 등을 지원하는 계층은 무엇인가?

① 세션 계층 ② 전송 계층
③ 네트워크 계층 ④ 데이터 링크 계층

77. 다음 설명에 해당하는 프로토콜은 무엇인가?

- 전송 계층의 프로토콜이다.
- 오버헤드가 적으며, 비연결형 서비스를 제공한다.
- 실시간 전송에 유리하고, 신뢰성보다는 속도가 중요시 된다.

① UDP ② TCP
③ RTCP ④ FTP

78. 다음 중 프로세스 스케줄링 기법에 대한 설명으로 가장 옳지 않은 것은?

① 비선점 스케줄링은 이미 할당된 CPU를 다른 프로세스가 강제로 빼앗아 사용할 수 없는 스케줄링 기법이다.
② 비선점 스케줄링은 프로세스 응답 시간의 예측이 용이하며, 일괄 처리 방식에 적합하다.
③ 선점 스케줄링은 주로 빠른 응답 시간을 요구하는 대화식 시분할 시스템에 사용된다.
④ 선점 스케줄링은 오버헤드를 최소화할 수 있다.

79. 사용자와 상호 작용 없이 여러 작업들이 정해진 일련의 순서에 따라 자동으로 수행되도록 지원하는 도구를 무엇이라 하는가?

① 배치 스케줄러(Batch Scheduler)
② 응용 프로그램(Application Program)
③ 패키지 소프트웨어(Package Software)
④ 프로시저(Procedure)

80. OSI의 전 계층에서 프로토콜 구조가 다른 네트워크 간의 연결을 위해 프로토콜 변환 기능을 수행하는 네트워크 장비는 무엇인가?

① 리피터(Repeater)
② 브리지(Bridge)
③ 라우터(Router)
④ 게이트웨이(Gateway)

정답 : 61.③ 62.① 63.② 64.④ 65.④ 66.① 67.② 68.② 69.③ 70.④ 71.② 72.③ 73.④ 74.② 75.② 76.④ 77.① 78.④ 79.① 80.④

5과목 정보시스템 구축 관리

81. 다음 중 객체지향 방법론의 구성 요소에 해당하지 않는 것은?

① 객체(Object) ② 패키지(Package)
③ 클래스(Class) ④ 메시지(Message)

82. 다음 설명에 해당하는 소프트웨어 개발 방법론은 무엇인가?

- 기존의 시스템이나 소프트웨어를 구성하는 컴포넌트를 조합하여 하나의 새로운 애플리케이션을 만드는 방법론이다.
- 컴포넌트의 재사용(Reusability)이 가능하여 시간과 노력을 절감할 수 있다.

① 구조적 방법론 ② 정보공학 방법론
③ Agile 방법론 ④ CBD 방법론

83. 다음 중 소프트웨어 비용 산정에 대한 설명으로 옳지 않은 것은?

① 소프트웨어의 규모, 인력 등의 요소를 기반으로 개발에 필요한 비용을 예측하는 것이다.
② 소프트웨어 비용 산정 기법에는 상향식, 하향식, 혼합식 기법이 있다.
③ 소프트웨어 비용을 높게 산정할 경우 예산 낭비와 일의 효율성 저하를 초래할 수 있다.
④ 소프트웨어 비용 결정 요소에는 프로젝트 요소, 자원 요소, 생산성 요소가 있다.

84. 다음 중 하향식 비용 산정 기법으로만 올바르게 짝지어진 것은?

① 전문가 감정 기법, 델파이 기법
② 델파이 기법, 수학적 산정 기법
③ LOC 기법, 전문가 감정 기법
④ 델파이 기법, LOC 기법

85. LOC 기법에 의해 다음과 같은 조건이 제시되었을 때 개발에 필요한 프로그래머의 수는 몇 명인가?

- 총 라인 수 : 30,000
- 1인당 월평균 생산 코드 라인 수 : 600
- 개발 기간 : 10개월

① 3명 ② 4명
③ 5명 ④ 10명

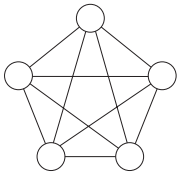
86. 다음 중 Putnam 모형에 대한 설명으로 가장 옳지 않은 것은?

① 소프트웨어 생명 주기의 전 과정 동안에 사용될 노력의 분포를 가정해 주는 모형이다.
② 대형 프로젝트의 노력 분포 산정에 이용되는 기법이다.
③ 자동화 추정 도구 ESTIMACS는 Putnam 모형을 기초로 하여 개발되었다.
④ 개발 기간이 늘어날수록 프로젝트 적용 인원의 노력이 감소한다.

87. 각종 컴퓨팅 자원을 중앙 컴퓨터에 두고 인터넷 기능을 갖는 단말기로 언제 어디서나 인터넷을 통해 컴퓨터 작업을 수행할 수 있는 환경으로, 키보드와 모니터, 마우스를 갖추고 통신 포트만 연결하면 업무 수행이 가능해지는 기술을 무엇이라 하는가?

① 클라우드 컴퓨팅(Cloud Computing)
② 그리드 컴퓨팅(Grid Computing)
③ 사물 인터넷(IoT)
④ 차세대 통신망(NGN)

88. 다음 그림과 같은 형태의 네트워크 토폴로지에 대한 설명으로 옳지 않은 것은?



① 모든 지점의 컴퓨터와 단말장치를 서로 연결한 형태로, 노드의 연결성이 높다.
② 많은 단말장치로부터 많은 양의 통신을 필요로 하는 경우에 유리하다.
③ 노드의 수가 5개이므로, 총 10개의 회선이 필요하다.
④ 보통 분산 처리 시스템에서 사용되며, 통신 회선의 총 경로가 가장 길다.

89. 다음 설명에 해당하는 기술은 무엇인가?

실시간으로 발생하는 많은 사건들 중 의미가 있는 것만을 추출할 수 있도록 사건 발생 조건을 정의하는 데이터 처리 방법으로, 대용량 데이터 스트림에 대한 요구에 실시간으로 대응하기 위해 개발되었다.

① 복잡 이벤트 처리(CEP)
② 소프트웨어 에스크로(Software Escrow)
③ 서비스형 소프트웨어(SaaS)
④ 딥 러닝(Deep Learning)

90. 여러 개의 하드디스크로 디스크 배열을 구성하여 디스크의 속도를 향상시키거나 데이터를 복구할 수 있도록 하는 기술은 무엇인가?

① HA ② MEMS
③ RAID ④ M-DISC

91. 다음 괄호에 공통으로 들어갈 용어는 무엇인가?

> ( )는 일련의 데이터를 정의하고 설명해 주는 데이터로, 컴퓨터에서는 데이터 사전의 내용, 스키마 등을 의미한다. ( )는 여러 용도로 사용되나 주로 빠르게 검색하거나 내용을 간략하고 체계적으로 하기 위해 많이 사용된다.

① View ② Broad Data
③ Big Data ④ Meta Data

92. 다음 중 Secure SDLC에 대한 설명으로 가장 옳지 않은 것은?

① 보안상 안전한 소프트웨어를 개발하기 위해 SDLC에 보안 강화를 위한 프로세스를 포함한 것이다.
② 소프트웨어의 유지 보수 단계에서 보안 이슈의 해결에 소모되는 많은 비용을 최소화하기 위해 등장하였다.
③ 요구사항 분석, 설계, 구현, 테스트, 유지 보수 등 SDLC의 모든 단계에 걸쳐 수행되어야 할 보안 활동을 제시한다.
④ 설계 단계에서는 조직의 정보보호 관련 보안 정책을 참고하여 적용할 수 있는 항목들을 문서화한다.

93. 공개키 암호화(Public Key Encryption) 기법에 대한 설명으로 가장 옳지 않은 것은?

① 데이터를 암호화할 때 사용하는 공개키(Public Key)로 데이터를 암호화하고 복호화 한다.
② 비대칭 암호 기법이라고도 불린다.
③ RSA 기법이 대표적인 공개키 암호화 기법이다.
④ 암호화/복호화 속도가 느리지만, 키의 분배가 용이하고 관리해야 할 키의 개수가 적다.

94. 다음에서 설명하는 네트워크 공격 방법은 무엇인가?

> 여러 곳에 분산된 공격 지점에서 한 곳의 서버를 공격하는 것으로, 네트워크에서 취약점이 있는 호스트들을 탐색한 후 이들 호스트들에 공격용 툴을 설치하여 에이전트로 만든 후 공격에 이용한다.

① 스미싱 ② DDoS 공격
③ 무작위 대입 공격 ④ 트로이 목마

95. 공중 네트워크와 암호화 기술을 이용하여 인터넷망을 전용선의 사설망을 구축한 것처럼 이용하는 기술로, 이동 근무자가 지역적인 제한 없이 업무를 수행할 수 있게 도와주는 보안 솔루션을 무엇이라 하는가?

① ESM(Enterprise Security Management)
② IPS(Intrusion Prevention System)
③ VPN(Virtual Private Network)
④ NAC(Network Access Control)

96. 흐름 제어에서 확인 신호(ACK)를 받은 후에 다음 패킷을 전송하는 방식을 무엇이라 하는가?

① 정지 및 대기
② 슬라이딩 윈도우
③ 다중 전송
④ 적응적 ARQ

97. 코드 오류로 발생할 수 있는 보안 약점 중 널 포인터(Null Pointer) 역참조가 발생하지 않도록 프로그래머가 수행해야 하는 일로 가장 옳은 것은?

① 반환된 메모리에 접근할 수 없도록 주소를 저장하고 있는 포인터를 초기화한다.
② 변수 선언 시 할당된 메모리를 초기화한다.
③ 포인터를 이용하기 전에 널 값을 갖고 있는지 검사한다.
④ 자원 반환 코드가 누락되었는지 확인한다.

98. Public 메소드로부터 Private 배열이 반환될 경우 발생하는 문제점으로 가장 옳은 것은?

① 메소드로의 접근이 불가능해진다.
② 배열의 주소가 외부로 공개되어 외부에서 접근할 수 있게 된다.
③ 시스템의 내부 정보가 노출된다.
④ 동기화 오류가 발생한다.

99. 다음 설명에 해당하는 방법을 이용한 인증(Authentication) 유형은 무엇인가?

> 아이핀(i-PIN)은 인터넷에서 주민등록번호 대신 쓸 수 있도록 만든 사이버 주민등록번호로, 사용자에 대한 신원확인을 완료한 후에 본인확인기관에서 온라인으로 발행한다.

① 지식 기반 인증(Something You Know)
② 소유 기반 인증(Something You Have)
③ 생체 기반 인증(Something You Are)
④ 위치 기반 인증(Somewhere You Are)

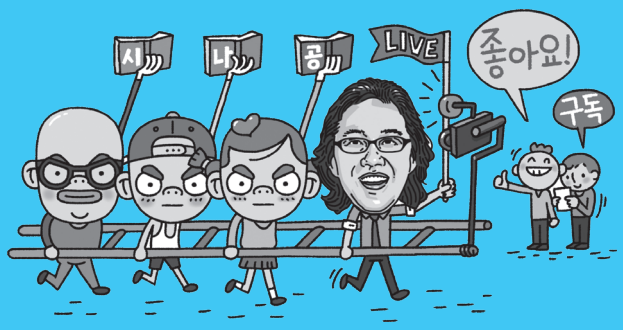
정답 : 81.② 82.④ 83.② 84.① 85.③ 86.③ 87.① 88.④ 89.① 90.③ 91.④ 92.④ 93.① 94.② 95.③ 96.① 97.③ 98.② 99.①

100. 다음 중 세션ID의 관리 방법에 대한 설명으로 틀린 것은?

① 안전한 서버에서 최소 128비트의 길이로 생성한다.
② 세션ID가 노출되지 않도록 URL Rewrite 기능을 사용한다.
③ 예측이 불가능하도록 안전한 난수 알고리즘을 적용한다.
④ 로그인 시 로그인 전의 세션ID를 삭제하고 재할당한다.

정답 : 100.②

최종점검 모의고사
해설



정보처리기사 필기 최종점검 모의고사 1회 해설

1 애자일 모형(Agile Model)은 변화에 기민하고 유연하게 대응하는 것이 특징인 모형으로, 계획을 따르기 보다는 변화에 반응하는 것에 더 가치를 둡니다.

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.

2 스크럼(Scrum) 개발 과정은 계획한 내용을 토대로 일정 기간 동안 스프린트를 수행하면서 진행 상황을 매일 점검하고 하나의 스프린트가 끝나면 검토한 후 진행을 되돌아봅니다.

3 새로운 시스템 개발에 있어서 하드웨어 구성 파악 시 고려해야 할 사항에는 운용 서버의 용도, 수량, 주요 사양, 이중화 적용 여부가 있습니다.

4 정형 분석은 구문(Syntax)과 의미(Semantics)를 갖는 정형화된 언어를 이용해 요구사항을 수학적 기호로 표현한 후 이를 분석하는 과정입니다. ④번의 내용은 요구사항 할당(Requirement Allocation)에 대한 설명입니다.

요구사항 분석 기법

개발 대상에 대한 사용자의 요구사항 중 명확하지 않거나 모호한 부분을 걸러내기 위한 방법이다.

- 요구사항 분류(Requirement Classification) : 요구사항을 명확히 확인할 수 있도록 분류함
- 개념 모델링(Conceptual Modeling) : 요구사항을 보다 쉽게 이해할 수 있도록 현실 세계의 상황을 단순화하여 개념적으로 표현한 것을 모델이라고 하며, 이러한 모델을 만드는 과정을 모델링이라고 함
- 요구사항 할당(Requirement Allocation) : 요구사항을 만족시키기 위한 구성 요소를 식별하는 것
- 요구사항 협상(Requirement Negotiation) : 요구사항이 서로 충돌될 경우 이를 적절히 해결하는 과정
- 정형 분석(Formal Analysis) : 구문(Syntax)과 의미(Semantics)를 갖는 정형화된 언어를 이용해 요구사항을 수학적 기호로 표현한 후 이를 분석하는 과정

5 ②번은 비기능 요구사항(Non-functional requirements) 중 하나인 프로젝트 관리 요구사항에 해당합니다.

기능 요구사항(Functional Requirements)

- 시스템이 무엇을 하는지, 어떤 기능을 하는지에 대한 사항
- 시스템의 입력이나 출력으로 무엇이 포함되어야 하는지, 시스템이 어떤 데이터를 저장하거나 연산을 수행해야 하는지에 대한 사항
- 시스템이 반드시 수행해야 하는 기능
- 사용자가 시스템을 통해 제공받기를 원하는 기능

비기능 요구사항(Non-Functional Requirements)

- 시스템 장비 구성 요구사항
- 성능 요구사항
- 인터페이스 요구사항
- 데이터 요구사항
- 테스트 요구사항
- 보안 요구사항
- 품질 요구사항
- 제약사항
- 프로젝트 관리 요구사항
- 프로젝트 지원 요구사항

6 웹 애플리케이션 서버(WAS)에 대한 요구사항 식별 시 고려해야 할 사항에는 가용성, 성능, 기술 지원, 구축 비용이 있습니다.

애플리케이션 서버(WAS) 관련 요구사항 식별 시 고려사항

- 가용성 : 장시간 운영 시 발생할 수 있는 고유의 장애 발생 가능성
- 성능 : 대규모의 트랜잭션 처리 능력 및 다양한 옵션 지원
- 기술 지원 : 제조업체의 안정적인 기술 지원 및 여러 사용자들 간의 정보 공유
- 구축 비용 : 라이선스, 유지관리 등에 대한 비용

7 인수 테스트는 사용자가 실제로 사용할 환경에서 요구사항들이 모두 충족되는지 사용자 입장에서 확인하는 과정입니다.

요구사항 확인 기법

요구사항 개발 과정을 거쳐 문서화된 요구사항 관련 내용을 확인하고 검증하는 방법이다.

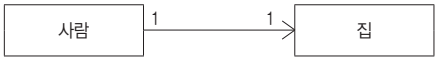
- 모델 검증(Model Verification) : 요구사항 분석 단계에서 개발된 모델이 요구사항을 충족시키는지 검증하는 것
- 프로토타이핑(Prototyping) : 초기 도출된 요구사항을 토대로 프로토타입(초기 모형)을 만든 후 대상 시스템의 개발이 진행되는 동안 도출되는 요구사항을 반영하면서 지속적으로 프로토타입을 재작성하는 과정
- 요구사항 검토(Requirement Reviews) : 문서화된 요구사항을 훑어보면서 확인하는 것으로 가장 일반적인 요구사항 검증 방법
- 인수 테스트(Acceptance Tests) : 사용자가 실제로 사용할 환경에서 요구사항들이 모두 충족되는지 사용자 입장에서 확인하는 과정

8 포함 관계는 집합 관계의 특수한 형태로, 포함하는 사물의 변화가 포함되는 사물에게 영향을 미치는 관계를 표현합니다. 문제의 관계 표현에서 '사원번호'에 해당하는 '사원명'은 하나이며, 퇴직하여 '사원번호'가 없어지게 된다면 '사원명' 또한 필요가 없어집니다.

관계(Relationships)

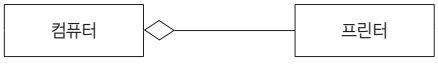
- 연관(Association) 관계
  - 2개 이상의 사물이 서로 관련되어 있음

예



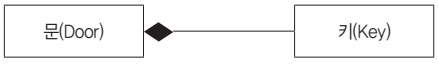
- 집합(Aggregation) 관계
  - 하나의 사물이 다른 사물에 포함되어 있는 관계

예



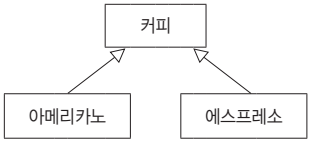
- 포함(Composition) 관계
  - 집합 관계의 특수한 형태로, 포함하는 사물의 변화가 포함되는 사물에게 영향을 미치는 관계

예



- 일반화(Generalization) 관계
  - 하나의 사물이 다른 사물에 비해 더 일반적인지 구체적인지를 표현하는 관계

예



- 의존(Dependency) 관계
  - 연관 관계와 같이 사물 사이에 서로 연관은 있으나 필요에 의해 서로에게 영향을 주는 짧은 시간 동안만 연관을 유지하는 관계

예

9 그림은 하나의 사물이 다른 사물에 비해 더 일반적인지 구체적인지를 표현하는 일반화(Generalization) 관계를 표현하고 있습니다. 가축을 구체적으로 표현하면 소, 돼지, 닭이 되고 반대로 소, 돼지, 닭을 일반적으로 표현하면 가축이 됩니다.

10 사용자의 동작으로 기기를 조작하는 인터페이스는 NUI입니다.

사용자 인터페이스의 종류

- 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
- VUI(Voice User Interface) : 사람의 음성으로 기기를 조작하는 인터페이스

11 웹의 3요소에는 웹 표준, 웹 접근성, 웹 호환성이 있습니다.

웹의 3요소

- 웹 표준(Web Standards) : 웹에서 사용되는 규칙 또는 기술을 의미하는 것으로, 웹 사이트 작성 시 이용하는 HTML, JavaScript 등에 대한 규정, 웹 페이지가 다른 기종이나 플랫폼에서도 구현되도록 제작하는 기법 등을 포함함
- 웹 접근성(Web Accessibility) : 누구나, 어떠한 환경에서도 웹 사이트에서 제공하는 모든 정보를 접근하여 이용할 수 있도록 보장하는 것
- 웹 호환성(Cross Browsing) : 하드웨어나 소프트웨어 등이 다른 환경에서도 모든 이용자에게 동등한 서비스를 제공하는 것

12 목업(Mockup)은 실제 화면과 유사하지만 실제로는 구현되지 않은 정적인 형태의 모형입니다.

UI 설계 도구

- 와이어프레임(Wireframe) : 페이지에 대한 개략적인 레이아웃이나 UI 요소 등에 대한 뼈대를 설계하는 단계
- 목업(Mockup) : 디자인, 사용 방법 설명, 평가 등을 위해 와이어프레임보다 좀 더 실제 화면과 유사하게 만든 정적인 형태의 모형
- 스토리보드(Story Board) : 와이어프레임에 콘텐츠에 대한 설명, 페이지 간 이동 흐름 등을 추가한 문서
- 프로토타입(Prototype) : 와이어프레임이나 스토리보드 등에 인터랙션을 적용함으로써 실제 구현된 것처럼 테스트가 가능한 동적인 형태의 모형
- 유스케이스(Use Case) : 사용자가 원하는 목표를 달성하기 위해 수행할 내용을 기술한 것

13 정황 시나리오는 사용자의 요구사항을 도출하기 위해 작성하는 것으로, 사용자가 목표를 달성하기 위해 수행하는 방법을 순차적으로 묘사한 것입니다. 지문에 공유에 대한 내용은 있지만 동시 열람에 관한 내용은 없습니다.

정황 시나리오

- 사용자의 요구사항을 도출하기 위해 작성하는 것으로, 사용자가 목표를 달성하기 위해 수행하는 방법을 순차적으로 묘사한 것이다.
- 요구사항 정의에 사용되는 초기 시나리오이다.
- 개발하는 서비스의 모습을 상상하는 첫 번째 단계로 사용자 관점에서 시나리오를 작성해야 한다.
- 사용자가 주로 사용하는 기능 위주로 작성해야 한다.
- 함께 발생되는 기능들은 하나의 시나리오에 통합한다.

- 육하원칙에 따라 간결하고 명확하게 작성한다.
- 작성된 시나리오는 외부 전문가 또는 경험이 풍부한 사람에게 검토를 의뢰한다.

14 자료 사전에서 자료의 반복을 의미하는 기호는 { }입니다.

자료 사전에서 사용되는 표기 기호

=	자료의 정의 : ~로 구성되어 있다(is composed of)
+	자료의 연결 : 그리고(and)
( )	자료의 생략 : 생략 가능한 자료(Optional)
[|]	자료의 선택 : 또는(or)
{ }	자료의 반복
**	자료의 설명 : 주석(Comment)

15 소프트웨어 아키텍처 설계의 기본 원리 중 추상화는 문제의 전체적이고 포괄적인 개념을 설계한 후 차례로 세분화하여 구체화시켜 나가는 것입니다.

아키텍처 설계의 기본 원리

- 모듈화(Modularity) : 소프트웨어의 성능을 향상시키거나 시스템의 수정 및 재사용, 유지 관리 등이 용이하도록 시스템의 기능들을 모듈 단위로 나누는 것
- 추상화(Abstraction) : 문제의 전체적이고 포괄적인 개념을 설계한 후 차례로 세분화하여 구체화시켜 나가는 것
- 단계적 분해(Stepwise Refinement) : 문제를 상위의 중요 개념으로부터 하위의 개념으로 구체화시키는 분할 기법
- 정보 은닉(Information Hiding) : 한 모듈 내부에 포함된 절차와 자료들의 정보가 감추어져 다른 모듈이 접근하거나 변경하지 못하도록 하는 기법

16 마스터 컴포넌트에서 슬레이브 컴포넌트로 작업을 분할한 후, 슬레이브 컴포넌트로부터 처리된 결과물을 다시 돌려받는 방식으로 작업을 수행하는 패턴을 마스터-슬레이브 패턴(Master-Slave Pattern)이라고 합니다.

주요 아키텍처 패턴(Patterns)의 종류

- 레이어 패턴(Layers pattern) : 시스템을 계층(Layer)으로 구분하여 구성하는 고전적인 방법 중의 하나로 각각의 서브시스템들이 계층 구조를 이루며, 상위 계층은 하위 계층에 대한 서비스 제공자가 되고, 하위 계층은 상위 계층의 클라이언트가 됨
- 클라이언트-서버 패턴 : 하나의 서버 컴포넌트와 다수의 클라이언트 컴포넌트로 구성되는 패턴으로, 클라이언트가 서버에 요청하고 응답을 받아 사용자에게 제공하는 방식
- 파이프-필터 패턴 : 데이터 스트림 절차의 각 단계를 필터(Filter) 컴포넌트로 캡슐화하여 파이프(Pipe)를 통해 데이터를 전송하는 패턴
- 모델-뷰-컨트롤러 패턴 : 서브시스템을 모델(Model), 뷰(View), 컨트롤러(Controller)의 세 부분으로 구조화하는 패턴

17 스탬프 결합도에서 자료 구조의 어떠한 변화, 즉 포맷이나 구조의 변화는 그것을 조회하는 모든 모듈 및 변화되는 필드를 실제로 조회하지 않는 모듈에도 영향을 미칩니다.

결합도(Coupling)

- 결합도는 모듈 간에 상호 의존하는 정도 또는 두 모듈 사이의 연관 관계를 의미한다.
- 다양한 결합으로 모듈을 구성할 수 있으나 결합도가 약할수록 품질이 높고, 강할수록 품질이 낮다.
- 결합도가 강하면 시스템 구현 및 유지보수 작업이 어렵다.
- 결합도의 종류
  - 자료 결합도(Data Coupling) : 모듈 간의 인터페이스가 자료 요소로만 구성될 때
  - 스탬프(검인) 결합도(Stamp Coupling) : 모듈 간의 인터페이스로 배열이나 레코드 등의 자료 구조가 전달될 때
  - 제어 결합도(Control Coupling) : 어떤 모듈이 다른 모듈 내부의 논리적인 흐름을 제어하기 위해 제어 신호를 이용하여 통신하거나 제어 요소를 전달할 때
  - 외부 결합도(External Coupling) : 어떤 모듈에서 선언한 데이터(변수)를 외부의 다른 모듈에서 참조할 때
  - 공통(공유) 결합도(Common Coupling) : 공유되는 공통 데이터 영역을 여러 모듈이 사용할 때
  - 내용 결합도(Content Coupling) : 한 모듈이 다른 모듈의 내부 기능 및 그 내부 자료를 직접 참조하거나 수정할 때

18 표의 숫자 코드는 코드화 대상 항목의 성질, 즉 길이, 넓이, 부피, 지름, 높이 등의 물리적 수치를 그대로 코드에 적용시키는 방법으로, 유효 숫자 코드라고도 합니다.

코드(Code)

- 컴퓨터를 이용하여 자료를 처리하는 과정에서 분류 · 조합 및 집계를 용이하게 하고, 특정 자료의 추출을 쉽게 하기 위해서 사용하는 기호이다.
- 순차 코드(Sequence Code) : 자료의 발생순서, 크기순서 등 일정 기준에 따라서 최초의 자료부터 차례로 일련번호를 부여하는 방법
- 블록 코드(Block Code) : 코드화 대상 항목 중에서 공통성이 있는 것끼리 블록으로 구분하고, 각 블록 내에서 일련번호를 부여하는 방법
- 10진 코드(Decimal Code) : 코드화 대상 항목을 0~9까지 10진 분할하고, 다시 그 각각에 대하여 10진 분할하는 방법을 필요한 만큼 반복하는 방법
- 그룹 분류 코드(Group Classification Code) : 코드화 대상 항목을 일정 기준에 따라 대분류, 중분류, 소분류 등으로 구분하고, 각 그룹 안에서 일련번호를 부여하는 방법
- 연상 코드(Mnemonic Code) : 코드화 대상 항목의 명칭이나 약호

와 관계있는 숫자나 문자, 기호를 이용하여 코드를 부여하는 방법

- 표의 숫자 코드(Significant Digit Code) : 코드화 대상 항목의 성질, 즉 길이, 넓이, 부피, 지름, 높이 등의 물리적 수치를 그대로 코드에 적용시키는 방법
- 합성 코드(Combined Code) : 필요한 기능을 하나의 코드로 수행하기 어려운 경우 2개 이상의 코드를 조합하여 만드는 방법

19 트랜잭션(Transaction)을 처리(Processing)하고 감시(Monitoring)하는 미들웨어는 TP-Monitor입니다. 다른 미들웨어에 대한 개념도 확인하고 넘어가세요.

미들웨어(Middleware)의 종류

- DB(DataBase) : 데이터베이스 벤더(Vendor)에서 제공하는 클라이언트에서 원격의 데이터베이스와 연결하기 위한 미들웨어
- RPC(Remote Procedure Call) : 응용 프로그램의 프로시저를 사용하여 원격 프로시저를 마치 로컬 프로시저처럼 호출하는 방식의 미들웨어
- MOM(Message Oriented Middleware) : 메시지 기반의 비동기형 메시지를 전달하는 방식의 미들웨어
- ORB(Object Request Broker) : 객체 지향 미들웨어로 코바(CORBA) 표준 스펙을 구현한 미들웨어
- WAS(Web Application Server) : 사용자의 요구에 따라 변하는 동적인 콘텐츠를 처리하기 위해 사용되는 미들웨어

20 자료 흐름도(DFD)는 요구사항을 분석하여 자료의 흐름 및 변환 과정과 기능을 도형 중심으로 기술하는 방법으로, 제어나 순서에 관한 정보도 알 수 있습니다.

자료 흐름도(DFD; Data Flow Diagram)

- 요구사항 분석에서 자료의 흐름 및 변환 과정과 기능을 도형 중심으로 기술하는 방법으로 자료 흐름 그래프, 버블 차트라고도 한다.
- 시스템 안의 프로세스와 자료 저장소 사이에 자료의 흐름을 나타내는 그래프로 자료 흐름과 처리를 중심으로 하는 구조적 분석 기법에 이용된다.
- 자료 흐름도는 자료 흐름과 기능을 자세히 표현하기 위해 단계적으로 세분화된다.
- 자료는 처리(Process)를 거쳐 변환될 때마다 새로운 이름이 부여되며, 처리는 입력 자료가 발생하면 기능을 수행한 후 출력 자료를 산출한다.
- 자료 흐름도에서는 자료의 흐름과 기능을 프로세스(Process), 자료 흐름(Data Flow), 자료 저장소(Data Store), 단말(Terminator)의 네 가지 기본 기호로 표시한다.

기호	의미
프로세스 (Process)	• 자료를 변환시키는 시스템의 한 부분(처리 과정)을 나타내며 처리, 기능, 변환, 버블이라고도 함 • 원(○)이나 둥근 사각형(▭)으로 표시하고 그 안에 프로세스 이름을 기입함
자료 흐름 (Data Flow)	• 자료의 이동(흐름)이나 연관관계를 나타냄 • 화살표(→) 위에 자료의 이름을 기입함
자료 저장소 (Data Store)	• 시스템에서의 자료 저장소(파일, 데이터베이스)를 나타냄 • 평행선(=) 안에 자료 저장소 이름을 기입함
단말 (Terminator)	• 시스템과 교신하는 외부 개체로, 입력 데이터가 만들어지고 출력 데이터를 받음(정보의 생산자와 소비자) • 사각형(▭) 안에 이름을 기입함

21 동일한 자료형의 데이터들이 같은 크기로 나열되어 순서를 갖고 있는 집합 형태의 자료 구조는 배열입니다.

자료 구조의 종류

배열(Array)	동일한 자료형의 데이터들이 같은 크기로 나열되어 순서를 갖고 있는 집합
리스트(List)	배열 또는 포인터를 이용하여 일정한 순서에 의해 나열된 자료 구조
스택(Stack)	리스트의 한쪽 끝으로만 자료의 삽입, 삭제 작업이 이루어지는 자료 구조
큐(Queue)	리스트의 한쪽에서는 삽입 작업이 이루어지고 다른 한쪽에서는 삭제 작업이 이루어지도록 구성한 자료 구조
트리(Tree)	정점(Node, 노드)과 선분(Branch, 가지)을 이용하여 사이클을 이루지 않도록 구성한 그래프(Graph)의 특수한 형태

22 물리적 독립성은 데이터베이스 시스템의 성능 향상을 위해 새로운 디스크를 추가하는 등의 단순한 하드웨어 변경으로부터 응용 프로그램이 영향을 받지 않는 성질을 가리킵니다. 응용 프로그램과 데이터베이스 시스템을 연결해주는 DBMS의 변경은 응용 프로그램에 영향을 주지 않을 수가 없습니다.

데이터의 독립성

- 종속성에 대비되는 말로 DBMS의 궁극적 목표이다.
- 데이터의 독립성에는 논리적 독립성과 물리적 독립성이 있다.
- 논리적 독립성 : 응용 프로그램과 데이터베이스를 독립시킴으로써, 데이터의 논리적 구조를 변경시키더라도 응용 프로그램은 변경되지 않음
- 물리적 독립성 : 응용 프로그램과 보조기억장치 같은 물리적 장치를 독립시킴으로써, 데이터베이스 시스템의 성능 향상을 위해 새로운 디스크를 도입하더라도 응용 프로그램에는 영향을 주지 않고 데이터의 물리적 구조만을 변경함

23 트랜잭션을 제어하기 위해서 사용하는 명령어를 TCL(Transaction Control Language)이라고 하며, TCL의 종류에는 COMMIT, ROLLBACK, SAVEPOINT가 있습니다. TCL은 기능을 제어하는 명령이라는 공통점으로 DCL의 일부로 분류하기도 합니다. 보기로 나온 다른 용어들의 개념도 확인하고 넘

어가세요.

- SQL : 관계형 데이터베이스를 지원하는 언어로, 데이터 정의어(DDL), 데이터 조작어(DML), 데이터 제어어(DCL)로 구분됨
- 데이터 정의어(DDL) : SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의하거나 변경 또는 삭제할 때 사용하는 언어
- 데이터 조작어(DML) : 데이터베이스 사용자가 응용 프로그램이나 질의어를 통하여 저장된 데이터를 실질적으로 처리하는 데 사용되는 언어
- 데이터 제어어(DCL) : 데이터의 보안, 무결성, 회복, 병행 수행 제어 등을 정의하는 데 사용되는 언어

24 절차형 SQL은 프로그래밍 언어와 같이 연속적인 실행이나 분기, 반복 등의 제어가 가능한 SQL이지만, 일반적인 프로그래밍 언어에 비해 효율은 떨어집니다.

절차형 SQL

- C, JAVA 등의 프로그래밍 언어와 같이 연속적인 실행이나 분기, 반복 등의 제어가 가능한 SQL을 의미한다.
- 일반적인 프로그래밍 언어에 비해 효율은 떨어지지만 단일 SQL 문장으로 처리하기 어려운 연속적인 작업들을 처리하는 데 적합하다.
- 절차형 SQL을 활용하여 다양한 기능을 수행하는 저장 모듈을 생성할 수 있다.
- DBMS 엔진에서 직접 실행되기 때문에 입 · 출력 패킷이 적은 편이다.
- BEGIN ~ END 형식으로 작성되는 블록(Block) 구조로 되어 있기 때문에 기능별 모듈화가 가능하다.
- 종류에는 프로시저, 트리거, 사용자 정의 함수가 있다.

25 IPC는 모듈 간 통신 방식을 구현하기 위해 사용되는 대표적인 프로그래밍 인터페이스 집합이며, IPC의 메소드 중 Adapter라는 메소드는 없습니다.

IPC의 대표 메소드

- Shared Memory : 다수의 프로세스가 공유 가능한 메모리를 구성하여 프로세스 간 통신을 수행함
- Socket : 네트워크 소켓을 이용하여 네트워크를 경유하는 프로세스들 간 통신을 수행함
- Semaphores : 공유 자원에 대한 접근 제어를 통해 프로세스 간 통신을 수행함
- Pipes&named Pipes : 'Pipe'라고 불리는 선입선출 형태로 구성된 메모리를 여러 프로세스가 공유하여 통신을 수행함
- Message Queueing : 메시지가 발생하면 이를 전달하는 형태로 프로세스 간 통신을 수행함

26 테스트 스크립트는 ISO/IEC/IEEE 29119-3 표준에 따른 테스트 케이스의 구성 요소에 포함되지 않습니다.

ISO/IEC/IEEE 29119-3 표준의 케이스 구성 요소

- 식별자(Identifier) : 항목 식별자, 일련번호
- 테스트 항목(Test Item) : 테스트 대상(모듈 또는 기능)
- 입력 명세(Input Specification) : 입력 데이터 또는 테스트 조건
- 출력 명세(Output Specification) : 테스트 케이스 수행 시 예상되는 출력 결과
- 환경 설정(Environmental Needs) : 필요한 하드웨어나 소프트웨어의 환경
- 특수 절차 요구(Special Procedure Requirement) : 테스트 케이스 수행 시 특별히 요구되는 절차
- 의존성 기술(Inter-Case Dependencies) : 테스트 케이스 간의 의존성

27 협업 도구에 익숙하지 않거나 이용할 의지가 없으면 협업 도구가 오히려 협업의 방해 요소가 될 수 있습니다.

28 빌딩(Building)은 소스 코드 파일들을 실행 파일로 변환하는 것, 모듈화(Modularity)는 시스템을 각 기능별로 나누는 것, 컴파일링(Compiling)은 소스 코드를 기계어로 변환하는 것을 의미합니다.

29 소프트웨어에서 사용한 외부 라이브러리는 릴리즈 노트를 통해 확인할 수 있는 정보가 아닙니다. 릴리즈 노트를 통해 확인할 수 있는 정보들은 다음과 같습니다.

- 테스트 진행 방법에 대한 결과
- 소프트웨어 사양에 대한 개발팀의 정확한 준수 여부
- 소프트웨어에 포함된 전체 기능
- 소프트웨어의 사용 환경
- 서비스의 내용과 개선 사항

30 패키징된 콘텐츠의 라이선스 정보는 DRM 컨트롤러가 아닌 클리어링 하우스에 등록됩니다.

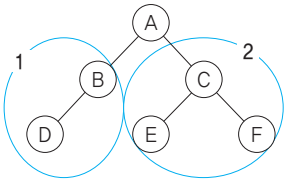
※ 클리어링 하우스(Clearing House) : 디지털 저작권 라이선스의 중개 및 발급을 수행하는 곳으로, 디지털 저작물의 이용 내역을 근거로 저작권료의 정산 및 분배가 수행됨

디지털 저작권 관리(DRM; Digital Right Management)

- 저작권자가 배포한 디지털 콘텐츠가 저작권자가 의도한 용도로만 사용되도록 디지털 콘텐츠의 생성, 유통, 이용까지의 전 과정에 걸쳐 사용되는 디지털 콘텐츠 관리 및 보호 기술이다.
- 원본 콘텐츠가 아날로그인 경우에는 디지털로 변환한 후 패키저(Packager)에 의해 DRM 패키징을 수행한다.
- 콘텐츠의 크기에 따라 음원이나 문서와 같이 크기가 작은 경우에는 사용자가 콘텐츠를 요청하는 시점에서 실시간으로 패키징을 수행하고, 크기가 큰 경우에는 미리 패키징을 수행한 후 배포한다.
- 패키징을 수행하면 콘텐츠에는 암호화된 저작권자의 전자서

명이 포함되고 저작권자가 설정한 라이선스 정보가 클리어링 하우스(Clearing House)에 등록된다.
- 사용자가 콘텐츠를 사용하기 위해서는 클리어링 하우스에 등록된 라이선스 정보를 통해 사용자 인증과 콘텐츠 사용 권한 소유 여부를 확인받아야 한다.
- 종량제 방식을 적용한 소프트웨어의 경우 클리어링 하우스를 통해 서비스의 실제 사용량을 측정하여 이용한 만큼의 요금을 부과한다.

31 먼저 서브 트리를 하나의 노드로 생각할 수 있도록 서브 트리 단위로 묶습니다.



❶ Postorder는 Left → Right → Root이므로 12A가 됩니다.
❷ 1은 DB이므로 DB2A가 됩니다.
❸ 2는 EFC이므로 DBEFCA가 됩니다.

32 형상 항목에 대한 변경 요구를 검토하여 현재의 베이스 라인이 잘 반영될 수 있도록 조정하는 작업을 형상 통제라고 합니다. ②번은 형상 감사, ③번은 형상 식별, ④번은 버전 제어에 대한 설명입니다.

형상 관리 기능의 종류
- **형상 식별** : 형상 관리 대상에 이름과 관리 번호를 부여하고, 계층(Tree) 구조로 구분하여 수정 및 추적이 용이하도록 하는 작업
- **버전 제어** : 소프트웨어 업그레이드나 유지 보수 과정에서 생성된 다른 버전의 형상 항목을 관리하고, 이를 위해 특정 절차와 도구(Tool)를 결합시키는 작업
- **형상 통제(변경 관리)** : 식별된 형상 항목에 대한 변경 요구를 검토하여 현재의 기준선(Base Line)이 잘 반영될 수 있도록 조정하는 작업
- **형상 감사** : 기준선의 무결성을 평가하기 위해 확인, 검증, 검열 과정을 통해 공식적으로 승인하는 작업
- **형상 기록(상태 보고)** : 형상의 식별, 통제, 감사 작업의 결과를 기록 · 관리하고 보고서를 작성하는 작업

33 보기 중 클라이언트/서버 방식의 버전 관리 도구는 SVN입니다. SCCS는 공유 폴더 방식, DCVS와 GIT은 분산 저장소 방식을 사용하는 소프트웨어 버전 관리 도구입니다.

소프트웨어 버전 관리 도구
- 공유 폴더 방식
  - 버전 관리 자료가 로컬 컴퓨터의 공유 폴더에 저장되어 관리되는 방식
  - 종류 : SCCS, RCS, PVCS, QVCS 등
- 클라이언트/서버 방식
  - 버전 관리 자료가 중앙 시스템(서버)에 저장되어 관리되는 방식
  - 종류 : CVS, SVN(Subversion), CVSNT, Clear Case, CMVC, Perforce 등
- 분산 저장소 방식
  - 버전 관리 자료가 하나의 원격 저장소와 분산된 개발자 PC의 로컬 저장소에 함께 저장되어 관리되는 방식
  - 종류 : Git, GNU arch, DCVS, Bazaar, Mercurial, TeamWare, Bitkeeper, Plastic SCM 등

34 테스터의 역량은 테스트 수행 전에 테스터가 갖춰줘야 할 조건입니다.

35 소프트웨어 내부의 논리 흐름에 따라 테스트 케이스를 작성하고 확인하는 테스트를 구조 기반 테스트라고 합니다. 나머지 보기로 제시된 테스트의 개념도 확인하고 넘어가세요.
- **명세 기반 테스트** : 사용자의 요구사항에 대한 명세를 빠짐없이 테스트 케이스로 만들어 구현하고 있는지 확인하는 테스트
- **경험 기반 테스트** : 유사 소프트웨어나 기술 등에 대한 테스터의 경험을 기반으로 수행하는 테스트
- **정적 테스트** : 프로그램을 실행하지 않고 명세서나 소스 코드를 대상으로 분석하는 테스트

36 블랙 박스 테스트에는 동치(동등) 분할 검사, 경계값 분석, 원인-효과 그래프 검사, 오류 예측 검사, 비교 검사가 있습니다. 제어 구조 검사는 화이트박스 테스트에 속합니다.

화이트 박스 테스트의 종류
- **기초 경로 검사** : 대표적인 화이트박스 테스트 기법
- **제어 구조 검사**
  - 조건 검사(Condition Testing) : 프로그램 모듈 내에 있는 논리적 조건을 테스트하는 테스트 케이스 설계 기법
  - 루프 검사(Loop Testing) : 프로그램의 반복(Loop) 구조에 초점을 맞춰 실시하는 테스트 케이스 설계 기법
  - 데이터 흐름 검사(Data Flow Testing) : 프로그램에서 변수의 정의와 변수 사용의 위치에 초점을 맞춰 실시하는 테스트 케이스 설계 기법

블랙 박스 테스트의 종류
- **동치 분할 검사(Equivalence Partitioning Testing)** : 입력 자료에 초점을 맞춰 테스트 케이스를 만들고 검사하는 방법으로 동등 분할 기법이라고도 함
- **경계값 분석(Boundary Value Analysis)** : 입력 자료에만 치중한 동

치 분할 기법을 보완하기 위한 기법

- 원인- 효과 그래프 검사(Cause-Effect Graphing Testing) : 입력 데이터 간의 관계와 출력에 영향을 미치는 상황을 체계적으로 분석한 다음 효용성이 높은 테스트 케이스를 선정하여 검사하는 기법
- 오류 예측 검사(Error Guessing) : 과거의 경험이나 확인자의 감각으로 테스트하는 기법
- 비교 검사(Comparison Testing) : 여러 버전의 프로그램에 동일한 테스트 자료를 제공하여 동일한 결과가 출력되는지 테스트하는 기법

37 통합 테스트로 인해 변경된 모듈이나 컴포넌트에 새로운 오류가 있는지 확인하는 테스트를 회귀 테스트라고 합니다. 나머지 보기로 제시된 테스트의 개념도 확인하고 넘어가세요.

- 혼합식 통합 테스트 : 하위 수준에서는 상향식 통합, 상위 수준에서는 하향식 통합을 사용하여 최적의 테스트를 지원하는 방식
- 상향식 통합 테스트 : 프로그램의 하위 모듈에서 상위 모듈 방향으로 통합하면서 테스트하는 기법
- 하향식 통합 테스트 : 프로그램의 상위 모듈에서 하위 모듈 방향으로 통합하면서 테스트하는 기법

38 테스트 데이터의 종류에는 선행된 연산에 의해 만든 데이터, 실제 운영되는 데이터를 복제한 실제 데이터, 스크립트를 통해서 인위적으로 만든 가상 데이터 등이 있습니다.

테스트 데이터

- 시스템의 기능이나 적합성 등을 테스트하기 위해 만든 데이터 집합이다.
- 소프트웨어의 기능을 차례대로 테스트할 수 있도록 만든 데이터이다.
- 잘못된 데이터는 잘못된 결과를 도출하기 때문에 효율적인 테스트를 위해서는 올바른 테스트 데이터를 준비해야 한다.
- 테스트 데이터 종류
  - 실제 데이터 : 선행된 연산에 의해 만들거나 실제 운영되는 데이터를 복제한 데이터
  - 가상 데이터 : 스크립트를 통해서 인위적으로 만든 데이터

39 테스트 블록(Test Block)은 테스트 하네스의 구성 요소에 속하지 않습니다.

테스트 하네스(Test Harness)의 구성 요소

- 테스트 드라이버(Test Driver) : 테스트 대상의 하위 모듈을 호출하고, 파라미터를 전달하고, 모듈 테스트 수행 후의 결과를 도출하는 도구
- 테스트 스텁(Test Stub) : 제어 모듈이 호출하는 타 모듈의 기능을 단순히 수행하는 도구로, 일시적으로 필요한 조건만을 가지고 있는 테스트용 모듈
- 테스트 슈트(Test Suites) : 테스트 대상 컴포넌트나 모듈, 시스템에 사용되는 테스트 케이스의 집합
- 테스트 케이스(Test Case) : 사용자의 요구사항을 정확하게 준수했는지 확인하기 위한 입력 값, 실행 조건, 기대 결과 등으로 만들어진 테스트 항목의 명세서
- 테스트 스크립트(Test Script) : 자동화된 테스트 실행 절차에 대한 명세서
- 목 오브젝트(Mock Object) : 사전에 사용자의 행위를 조건부로 입력해 두면, 그 상황에 맞는 예정된 행위를 수행하는 객체

40 ESB는 애플리케이션 보다는 서비스 중심의 통합을 지향합니다.

ESB(Enterprise Service Bus)

- 애플리케이션 간 연계, 데이터 변환, 웹 서비스 지원 등 표준 기반의 인터페이스를 제공하는 솔루션이다.
- 애플리케이션 통합 측면에서 EAI와 유사하지만 애플리케이션 보다는 서비스 중심의 통합을 지향한다.
- 특정 서비스에 국한되지 않고 범용적으로 사용하기 위하여 애플리케이션과의 결합도(Coupling)를 약하게(Loosely) 유지한다.
- 관리 및 보안 유지가 쉽고, 높은 수준의 품질 지원이 가능하다.

41 데이터 모델에 표시하는 구조(Structure), 연산(Operation), 제약 조건(Constraint) 중 실제 데이터에 조건을 설정하는 것은 제약 조건(Constraint)입니다.

데이터 모델에 표시할 요소

- 구조(Structure) : 논리적으로 표현된 개체 타입들 간의 관계로서 데이터 구조 및 정적 성질 표현
- 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

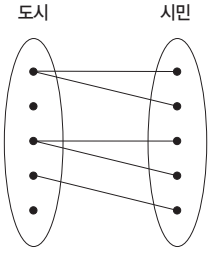
42 외래키는 다른 개체와의 '관계(Relationship)'를 통해 생성된 속성이므로, 관계 있는 개체가 변경되면 외래키 속성도 영향을 받게 됩니다.

속성의 종류

- 속성의 특성에 따른 분류
  - 기본 속성(Basic Attribute) : 업무 분석을 통해 정의한 속성
  - 설계 속성(Designed Attribute) : 원래 업무상 존재하지 않고 설계 과정에서 도출해내는 속성
  - 파생 속성(Derived Attribute) : 다른 속성으로부터 계산이나 변형 등의 영향을 받아 발생하는 속성
- 개체 구성 방식에 따른 분류
  - 기본키 속성(Primary Key Attribute) : 개체를 식별할 수 있는 속성

– 외래키 속성(Foreign Key Attribute) : 다른 개체와의 관계에서 포함된 속성
– 일반 속성 : 개체에 포함되어 있고 기본키, 외래키에 포함되지 않은 속성

43 〈도시〉의 각 원소는 〈시민〉의 원소 여러 개와 대응하고 있지만, 〈시민〉의 각 원소는 〈도시〉의 원소 한 개와 대응하는 관계이므로 일 대 다(1:N) 관계가 됩니다.



관계의 형태
- 일 대 일(1 : 1) : 개체 집합 A의 각 원소가 개체 집합 B의 원소 한 개와 대응하는 관계
- 일 대 다(1 : N) : 개체 집합 A의 각 원소는 개체 집합 B의 원소 여러 개와 대응하고 있지만, 개체 집합 B의 각 원소는 개체 집합 A의 원소 한 개와 대응하는 관계
- 다 대 다(N : M) : 개체 집합 A의 각 원소는 개체 집합 B의 원소 여러 개와 대응하고, 개체 집합 B의 각 원소도 개체 집합 A의 원소 여러 개와 대응하는 관계

44 주 식별자의 속성이 두 개 이상인 경우 속성들을 하나의 속성으로 묶어 사용하는 식별자를 대리 식별자라고 합니다.

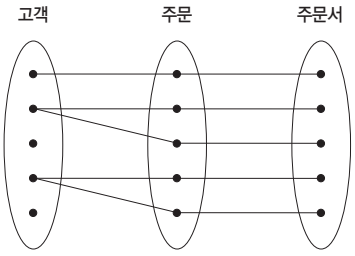
식별자의 분류
- 주 식별자/보조 식별자
  – 주 식별자(Primary Identifier) : 개체를 대표하는 유일한 식별자
  – 보조 식별자(Alternate Identifier) : 주 식별자를 대신하여 개체를 식별할 수 있는 속성
- 내부 식별자/외부 식별자
  – 내부 식별자(Internal Identifier) : 개체 내에서 스스로 만들어지는 식별자
  – 외부 식별자(Foreign Identifier) : 다른 개체와의 관계(Relationship)에 의해 외부 개체의 식별자를 가져와 사용하는 식별자
- 단일 식별자/복합 식별자
  – 단일 식별자(Single Identifier) : 주 식별자가 한 가지 속성으로만 구성된 식별자
  – 복합 식별자(Composit Identifier) : 주 식별자가 두 개 이상의 속성으로 구성된 식별자
- 원조 식별자/대리 식별자
  – 원조 식별자(Original Identifier) : 업무에 의해 만들어지는 가공되지 않은 원래의 식별자로, 본질 식별자라고도 함
  – 대리 식별자(Surrogate Identifier) : 주 식별자의 속성이 두 개 이상인 경우 속성들을 하나의 속성으로 묶어 사용하는 식별자

45 오너-멤버(Owner-Member) 관계라고도 불리는 데이터 모델은 논리적 데이터 모델 중 하나인 네트워크(망)형 데이터 모델입니다.

개체-관계(Entity-Relationship) 모델
- 개념적 데이터 모델의 가장 대표적인 것으로, 1976년 피터 첸(Peter Chen)에 의해 제안되고 기본적인 구성 요소가 정립되었다.
- E-R 모델은 개체와 개체 간의 관계를 기본 요소로 이용하여 현실 세계의 무질서한 데이터를 개념적인 논리 데이터로 표현하기 위한 방법으로 많이 사용되고 있다.
- E-R 모델은 개체 타입(Entity Type)과 이들 간의 관계 타입(Relationship Type)을 이용해 현실 세계를 개념적으로 표현한다.
- E-R 모델에서는 데이터를 개체(Entity), 관계(Relationship), 속성(Attribute)으로 묘사한다.
- E-R 모델은 특정 DBMS를 고려한 것은 아니다.
- E-R 다이어그램으로 표현하며, 1:1, 1:N, N:M 등의 관계 유형을 제한 없이 나타낼 수 있다.
- 최초에는 개체, 관계, 속성과 같은 개념들로 구성되었으나 나중에는 일반화 계층 같은 복잡한 개념들이 첨가되어 확장된 모델로 발전했다.
- E-R 도형

사각형	개체(Entity) 타입
다이아몬드(마름모)	관계(Relationship) 타입
타원	속성(Attribute)
이중 타원	다중값 속성(복합 속성)
밑줄 타원	기본 키 속성
복수 타원	복합 속성
선, 링크	개체 타입과 속성을 연결

46 한 명의 〈고객〉은 여러 〈주문〉이 가능하고, 각 〈주문〉은 하나의 〈주문서〉를 가지므로, 한 명의 〈고객〉은 여러 〈주문서〉를 갖는 일 대 다(1:N) 관계가 형성됩니다.



47 INSERT문의 기본 형식은 'INSERT INTO 테이블명(속성명) VALUES(값)'이지만, 테이블의 모든 속성에 값을 삽입할 때는 속성명을 생략하여 'INSERT INTO 테이블명 VALUES(값)'으로 사용할 수 있습니다.

- INSERT INTO **사원** : 〈사원〉 테이블에 삽입하라.
- VALUES(1201, **'홍길동'**, **'인사'**); : 첫 번째 필드부터 순서대로 1201, '홍길동', '인사'를 삽입하라.

48 WHERE는 각각의 튜플에 조건을 지정하는 것이고, HAVING은 그룹에 대해 조건을 지정하는 명령이므로 HAVING 대신 WHERE를 사용하면 동일한 결과가 출력되지 않습니다. 또한 WHERE 조건문에는 SUM, AVG 등과 같은 그룹 함수를 사용할 수도 없습니다. 문제에 제시된 코드를 살펴보면 다음과 같습니다.

```
❶ SELECT TYPE, COUNT(*), SUM(SALES)
❷ FROM PRODUCT
❸ GROUP BY TYPE
❹ HAVING SUM(SALES) >= 300
❺ ORDER BY TYPE DESC;
```

❶ 'TYPE', 개수, 'SALES'의 합계를 표시합니다.
❷ 〈PRODUCT〉 테이블을 대상으로 검색합니다.
❸ 'TYPE'을 기준으로 그룹을 지정합니다.
❹ 그룹 중 'SALES'의 합계가 300 이상인 자료만을 대상으로 합니다.
❺ 'TYPE'을 기준으로 내림차순 정렬합니다.

49 프로시저의 실행 명령어는 EXECUTE, EXEC, CALL 3가지입니다. 문제에 제시된 코드를 살펴보면 다음과 같습니다.

```
❶ CREATE PROCEDURE GET_DEPT(i IN INT) IS
  BEGIN
❷   IF (MOD(i, 10) = 2) THEN
❸     INSERT INTO 회원(코드, 분류) VALUES(i, 'VIP');
    ELSE
❹     INSERT INTO 회원(코드, 분류) VALUES(i, '일반');
    END IF;
  END;
```

❶ 호출 프로그램이 전달한 값을 'i'에 저장한 후 사용하는 프로시저 'GET_DEPT'를 생성합니다.
❷ i를 10으로 나눈 나머지가 2이면 ❸번을 수행하고 아니면 ❹번을 수행합니다.
❸ 〈회원〉 테이블의 '코드', '분류' 속성에 각각 i의 값과 'VIP'를 입력합니다.
❹ 〈회원〉 테이블의 '코드', '분류' 속성에 각각 i의 값과 '일반'을 입력합니다.

50 테이블에 속한 튜플의 수를 카디널리티(Cardinality)라고 합니다. ①번은 속성의 수, ③번은 도메인(Domain), ④번은 속성(Attribute)에 대한 설명입니다.

관계형 데이터베이스 관련 용어

- 튜플(Tupel)
  - 튜플은 릴레이션을 구성하는 각각의 행을 말한다.
  - 튜플은 속성의 모임으로 구성된다.
  - 파일 구조에서 레코드와 같은 의미이다.
  - 튜플의 수를 카디널리티(Cardinality) 또는 기수, 대응수라고 한다.
- 속성(Attribute)
  - 속성은 데이터베이스를 구성하는 가장 작은 논리적 단위이다.
  - 파일 구조상의 데이터 항목 또는 데이터 필드에 해당된다.
  - 속성은 개체의 특성을 기술한다.
  - 속성의 수를 디그리(Degree) 또는 차수라고 한다.
- 도메인(Domain)
  - 도메인은 하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자(Atomic)값들의 집합이다.
  - 도메인은 실제 애트리뷰트 값이 나타날 때 그 값의 합법 여부를 시스템이 검사하는데에도 이용된다.

  예 성별 애트리뷰트의 도메인은 '남'과 '여'로, 그 외의 값은 입력될 수 없다.

51 데이터 무결성은 애플리케이션(프로그램), 데이터베이스 트리거, 제약 조건을 이용하여 강화할 수 있습니다.

데이터 무결성 강화 방법

- 애플리케이션
  - 데이터 생성, 수정, 삭제 시 무결성 조건을 검증하는 코드를 데이터를 조작하는 프로그램 내에 추가한다.
  - 데이터베이스에서는 데이터 무결성 제약을 정의할 수 없으므로 복잡한 규칙 등은 애플리케이션 내에서 처리한다.
  - 장점 : 사용자 정의 같은 복잡한 무결성 조건의 구현이 가능함
  - 단점 : 소스 코드에 분산되어 있어 관리가 힘들고, 개별적인 시행으로 인해 적정성 검토가 어려움
- 데이터베이스 트리거
  - 트리거 이벤트에 무결성 조건을 실행하는 절차형 SQL을 추가한다.
  - 장점 : 통합 관리가 가능하고, 복잡한 요구 조건의 구현이 가능함

– 단점 : 운영 중 변경이 어렵고, 사용상 주의가 필요함

• 제약 조건

– 데이터베이스에 제약 조건을 설정하여 무결성을 유지한다.

– 장점 : 통합 관리 가능, 간단한 선언으로 구현 가능, 변경 용이, 오류 데이터 발생 방지 등이 있음

– 단점 : 복잡한 제약 조건의 구현과 예외적인 처리가 불가능함

52 카티션 프로덕트(Cartesian Product), 즉 교차곱은 두 릴레이션의 차수(Degree, 속성의 수)는 더하고, 카디널리티(튜플의 수)는 곱하면 됩니다. 즉 차수는 3+5 = 8이고, 카디널리티는 4 ×6 = 24입니다.

53 3NF는 기본키가 아닌 모든 속성이 기본키에 대해 이행적 종속을 만족하지 않는 정규형입니다. ③번의 내용은 4NF에 대한 설명입니다.

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

정규화 단계 암기 요령

정규화가 두부가게에 가서 모든 두부를 달라고 하니 주인이 깜짝 놀라 말했다.

두부이걸다줘? ≒ 도부이결다조

도메인이 원자값

부분적 함수 종속 제거

이행적 함수 종속 제거

결정자이면서 후보키가 아닌 것 제거

다치 종속 제거

조인 종속성 이용

54 인덱스 대상 속성(컬럼) 선정 기준에서 인덱스 속성의 적절한 분포도는 10~15% 이내입니다.

인덱스 대상 컬럼 선정 기준

• 인덱스 컬럼의 분포도가 10~15% 이내인 컬럼

– 분포도 = (컬럼값의 평균 Row 수 / 테이블의 총 Row 수) × 100

• 분포도가 10~15% 이상이어도 부분 처리를 목적으로 하는 컬럼

• 입 · 출력 장표 등에서 조회 및 출력 조건으로 사용되는 컬럼

• 인덱스가 자동 생성되는 기본키와 Unique키 제약 조건을 사용한 컬럼

• 가능한 한 수정이 빈번하지 않은 컬럼

• ORDER BY, GROUP BY, UNION이 빈번한 컬럼

• 분포도가 좁은 컬럼은 단독 인덱스로 생성

• 인덱스들이 자주 조합되어 사용되는 경우 하나의 결합 인덱스(Concatenate Index)로 생성

55 문제의 지문에 제시된 내용은 분산 데이터베이스의 4대 목표 중 중복 투명성(Replication Transparency)에 대한 설명입니다.

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

56 데이터베이스에서 데이터 저장 시 데이터의 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법을 클러스터(Cluster)라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

• 인덱스(Index) : 데이터 레코드를 빠르게 접근하기 위해 〈키 값, 포인터〉 쌍으로 구성되는 데이터 구조

• 제약 조건(Constraint) : 데이터베이스에 저장될 수 있는 실제 데이터의 논리적인 제약 조건

• 파티션(Partition) : 대용량의 테이블이나 인덱스를 작은 논리적 단위인 파티션으로 나누는 것

57 관계형 데이터베이스에서 하나의 릴레이션에는 중복된 튜플이 존재할 수 없으므로 모든 릴레이션에는 반드시 하나 이상의 후보키가 있어야 합니다.

후보키(Candidate Key)

• 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용하는 속성들의 부분집합, 즉 기본키로 사용할 수 있는 속성들을 말한다.

• 하나의 릴레이션내에서는 중복된 튜플들이 있을 수 없으므로 모든 릴레이션에는 반드시 하나 이상의 후보키가 존재한다.

- 후보키는 릴레이션에 있는 모든 튜플에 대해서 유일성과 최소성을 만족시켜야 한다.
  - 유일성(Unique) : 하나의 키 값으로 하나의 튜플만을 유일하게 식별할 수 있어야 함
  - 최소성(Minimality) : 모든 레코드들을 유일하게 식별하는데 꼭 필요한 속성으로만 구성되어야 함

  예 〈학생〉 릴레이션에서 '학번'이나 '주민번호'는 다른 레코드를 유일하게 구별할 수 있는 기본키로 사용할 수 있으므로 후보키이다.

58 SQL의 DROP 문은 스키마, 테이블, 뷰 등의 제거에 사용되는 명령이고, CASCADE 옵션은 삭제할 요소를 참조하는 다른 모든 개체를 함께 삭제하는 옵션입니다. 즉 인사 테이블을 참조하는 테이블과 인사 테이블을 모두 제거합니다.

DROP

- 스키마, 도메인, 기본 테이블, 뷰 테이블, 인덱스, 제약 조건 등을 제거하는 명령문이다.
- 표기 형식

```
DROP SCHEMA 스키마명 [CASCADE | RESTRICTED];
DROP DOMAIN 도메인명 [CASCADE | RESTRICTED];
DROP TABLE 테이블명 [CASCADE | RESTRICTED];
DROP VIEW 뷰명 [CASCADE | RESTRICTED];
DROP INDEX 인덱스명 [CASCADE | RESTRICTED];
DROP CONSTRAINT 제약조건명;
```

- DROP SCHEMA : 스키마를 제거함
- DROP DOMAIN : 도메인을 제거함
- DROP TABLE : 테이블을 제거함
- DROP VIEW : 뷰를 제거함
- DROP INDEX : 인덱스를 제거함
- DROP CONSTRAINT : 제약 조건을 제거함
- CASCADE : 제거할 요소를 참조하는 다른 모든 개체를 함께 제거함. 즉 주 테이블의 데이터 제거 시 각 외래키와 관계를 맺고 있는 모든 데이터를 제거하는 참조 무결성 제약 조건을 설정하기 위해 사용됨
- RESTRICTED : 다른 개체가 제거할 요소를 참조중일 때는 제거를 취소함

59 서버와 저장장치를 연결하는 스토리지의 종류 중 하나로, 서버와 저장장치를 연결하는 전용 네트워크를 별도로 구성하는 방식을 SAN(Storage Area Network)이라고 합니다. DES는 암호화 알고리즘이고 나머지 보기는 다른 종류의 스토리지입니다.

스토리지(Storage)의 종류

- DAS(Direct Attached Storage) : 서버와 저장장치를 전용 케이블로 직접 연결하는 방식
- NAS(Network Attached Storage) : 서버와 저장장치를 네트워크를 통해 연결하는 방식
- SAN(Storage Area Network) : DAS의 빠른 처리와 NAS의 파일 공유 장점을 혼합한 방식으로, 서버와 저장 장치를 연결하는 전용 네트워크를 별도로 구성하는 방식

60 서브타입 기준 테이블 변환은 각 서브타입 속성들의 선택 사양이 명확한 경우에 유리합니다.

서브타입 기준 테이블 변환

- 슈퍼타입 속성들을 각각의 서브타입에 추가하여 서브타입들을 개별적인 테이블로 만드는 것이다.
- 서브타입에 속성이나 관계가 많이 포함된 경우 적용한다.
- 장점
  - 각 서브타입 속성들의 선택 사양이 명확한 경우에 유리하다.
  - 처리할 때마다 서브타입 유형을 구분할 필요가 없다.
  - 여러 개의 테이블로 통합하므로 테이블당 크기가 감소하여 전체 테이블 스캔시 유리하다.
- 단점
  - 수행 속도가 감소할 수 있다.
  - 복잡한 처리를 하는 SQL의 통합이 어렵다.
  - 부분 범위에 대한 처리가 곤란해진다.
  - 여러 테이블을 통합한 뷰는 조회만 가능하다.
  - UID(Unique Identifier, 식별자)의 유지 관리가 어렵다.

61 C언어에서 변수명은 대 · 소문자를 구분합니다.

C언어의 변수명 작성 규칙

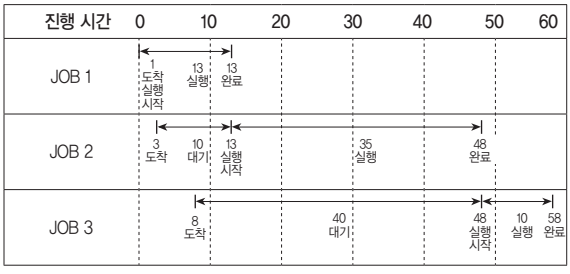
- 영문자, 숫자, _(under bar)를 사용할 수 있다.
- 첫 글자는 영문자나 _(under bar)로 시작해야 하며, 숫자는 올 수 없다.
- 글자 수에 제한이 없다.
- 공백이나 *, +, −, / 등의 특수문자를 사용할 수 없다.
- 대 · 소문자를 구분한다.
- 예약어를 변수명으로 사용할 수 없다.
- 변수 선언 시 문장 끝에 반드시 세미콜론(;)을 붙여야 한다.

62 C언어에서 제공하는 기억 클래스에는 자동 변수(Automatic Variable), 외부 변수(External Variable), 정적 변수(Static Variable), 레지스터 변수(Register Variable)가 있습니다.

기억 클래스의 종류

- 자동 변수(Automatic Variable) : 함수나 코드의 범위를 한정하는 블록 내에서 선언되는 변수
- 외부 변수(External Variable) : 현재 파일이나 다른 파일에서 선언된 변수나 함수를 참조(Reference)하기 위한 변수
- 정적 변수(Static Variable) : 함수나 블록 내에서 선언하는 내부 정적 변수와 함수 외부에서 선언하는 외부 정적 변수
- 레지스터 변수(Register Variable) : 레지스터 변수는 메모리가 아닌 CPU 내부의 레지스터에 기억영역을 할당받는 변수

63 FIFO 스케줄링은 준비상태 큐에 도착한 순서에 따라 CPU를 할당하는 기법입니다. 그러므로 작업 수행 순서와 평균 반환 시간은 다음과 같습니다.



- JOB 1 : 도착하자마자 실행하여 13에서 작업이 완료되므로 대기 시간은 0이고, 반환 시간은 13입니다.
- JOB 2 : 3에 도착하여 JOB 1이 완료될 때까지 대기한 후 JOB 1이 완료되는 13에서 실행을 시작하여 48에 작업이 완료됩니다. 그러므로 대기 시간은 10이고, 반환 시간은 45입니다.
- JOB 3 : 8에 도착하여 JOB 2가 완료될 때까지 대기한 후 JOB 2가 완료되는 48에서 실행을 시작하여 58에 작업이 완료됩니다. 그러므로 대기 시간은 40이고, 반환 시간은 50입니다.

∴ 평균 반환 시간은 (13+45+50)/3 = 36

64 포인터 변수는 필요에 의해 동적으로 할당되는 메모리 영역인 힙 영역에 접근하는 동적 변수입니다.

포인터(Pointer)

- 포인터는 변수의 주소를 말하며, C언어에서는 주소를 제어할 수 있는 기능을 제공한다.
- C언어에서 변수의 주소를 저장할 때 사용하는 변수를 포인터 변수라 한다.
- 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;
- 포인터 변수는 필요에 의해 동적으로 할당되는 메모리 영역인 힙 영역에 접근하는 동적 변수이다.

65 사용된 코드의 의미는 다음과 같습니다.

```
#include <stdio.h>
main( ) {
❶   int a = 1, sum = 0;
❷   while (a++ < 10)
❸      sum += a * 3 >= sum ? a : 1;
❹   printf("%d", sum);
}
```

❶ 정수형 변수 a와 sum을 선언한 후 각각 1과 0으로 초기화한다.

❷ a가 10보다 작은 동안 ❸번을 반복하여 수행한다. 조건을 비교한 후 a는 1 증가한다.

❸ a*3이 sum보다 크면 a를 sum에 누적하고, 아니면 1을 sum에 누적한다.

❹ sum의 값을 출력한다.

반복문 실행에 따른 변수의 변화는 다음과 같습니다.

반복횟수	❷의 조건	a	❸의 조건	sum
		1		0
1	1<10 (참)	2	6>=0 (참)	2
2	2<10 (참)	3	9>=2 (참)	5
3	3<10 (참)	4	12>=5 (참)	9
4	4<10 (참)	5	15>=9 (참)	14
5	5<10 (참)	6	18>=14 (참)	20
6	6<10 (참)	7	21>=20 (참)	27
7	7<10 (참)	8	24>=27 (거짓)	28
8	8<10 (참)	9	27>=28 (거짓)	29
9	9<10 (참)	10	30>=29 (참)	39
	10<10 (거짓)	11		

66 사용된 코드의 의미는 다음과 같습니다.

```
public class Test {
    public static void main(String[ ] args) {
❶       int array[ ] = { 1, 3, 5, 7, 9 };
❷       for (int a : array)
❸           if(a % 3 == 0) System.out.printf("*");
    }
}
```

❶ 정수형 배열 array를 선언하고 다음과 같이 초기화한다.

array	1	3	5	7	9

❷ array 배열의 각 요소를 a에 저장하면서 배열의 요소 수만큼 ❸번을 반복 수행한다.

❸ a를 3으로 나눈 나머지가 0이면 "*"을 출력하고, 아니면 ❷번으로 돌아간다.

반복문 실행에 따른 변수의 변화는 다음과 같습니다.

반복횟수	a	a%3	출력
1	1	1	
2	3	0	*
3	5	2	*
4	7	1	*
5	9	0	**

67 서식 지정자 '%f'는 소수점을 포함하는 실수를 입 · 출력하는데 사용합니다. "programming"과 같은 문자열을 출력하기 위해서는 서식 지정자 '%s'를 사용해야 합니다.

서식 문자열

서식 문자열	의미
%d	정수형 10진수를 입 · 출력하기 위해 지정함
%u	부호없는 정수형 10진수를 입 · 출력하기 위해 지정함
%o	정수형 8진수를 입 · 출력하기 위해 지정함
%x	정수형 16진수를 입 · 출력하기 위해 지정함
%c	문자를 입 · 출력하기 위해 지정함
%s	문자열을 입 · 출력하기 위해 지정함
%f	소수점을 포함하는 실수를 입 · 출력하기 위해 지정함
%e	지수형 실수를 입 · 출력하기 위해 지정함
%ld	long형 10진수를 입 · 출력하기 위해 지정함
%lo	long형 8진수를 입 · 출력하기 위해 지정함
%lx	long형 16진수를 입 · 출력하기 위해 지정함
%p	주소를 16진수로 입 · 출력하기 위해 지정함

68 단항 연산자에는 !, ~, ++, --, sizeof가 있습니다. ^는 비트끼리 XOR 연산을 수행하는 이항 연산자입니다.

연산자의 종류

- 단항 연산자 : !(논리 not), ~(비트 not), ++(증가), --(감소), sizeof(기타)
- 이항 연산자
  - 산술 연산자 : *, /, %(나머지), +, -
  - 시프트 연산자 : 〈〈, 〉〉
  - 관계 연산자 : 〈, 〈=, 〉=, 〉, ==(같다), !=(같지 않다)
  - 비트 연산자 : &(비트 and), ^(비트 xor), |(비트 or)
  - 논리 연산자 : &&(논리 and), ||(논리 or)
- 삼항 연산자 : (조건식) ? (참) : (거짓)
- 대입 연산자 : =, +=, -=, *=, /=, %=, 〈〈=, 〉〉= 등
- 순서 연산자 : ,

69 RR(Round-Robin) 스케줄링에서 FCFS 스케줄링과 같아질 경우는 Time Slice가 작을 경우가 아니라 Time Slice가 커질 경우입니다.

라운드 로빈(RR; Round Robin)

- 시분할 시스템(Time Sharing System)을 위해 고안된 방식으로, FCFS(FIFO) 알고리즘을 선점 형태로 변형한 기법이다.
- FCFS 기법과 같이 준비상태 큐에 먼저 들어온 프로세스가 먼저 CPU를 할당받지만 각 프로세스는 시간 할당량(Time Slice, Quantum) 동안만 실행한 후 실행이 완료되지 않으면 다음 프로세스에게 CPU를 넘겨주고 준비상태 큐의 가장 뒤로 배치된다.
- 할당되는 시간이 클 경우 FCFS 기법과 같아지고, 할당되는 시간이 작을 경우 문맥 교환 및 오버헤드가 자주 발생되어 요청된 작업을 신속히 처리할 수 없다.
- 할당되는 시간의 크기가 작으면 작은 프로세스들에게 유리하다.

70 사용자와 시스템 간의 인터페이스를 제공하는 것은 쉘(Shell)입니다.

UNIX 시스템의 구성

커널(Kernel)

- UNIX의 가장 핵심적인 부분이다.
- 컴퓨터가 부팅될 때 주기억장치에 적재된 후 상주하면서 실행된다.
- 하드웨어를 보호하고, 프로그램과 하드웨어 간의 인터페이스 역할을 담당한다.
- 프로세스(CPU 스케줄링) 관리, 기억장치 관리, 파일 관리, 입 · 출력 관리, 프로세스간 통신, 데이터 전송 및 변환 등 여러 가지 기능을 수행한다.

쉘(Shell)

- 사용자의 명령어를 인식하여 프로그램을 호출하고 명령을 수행하는 명령어 해석기이다.
- 시스템과 사용자 간의 인터페이스를 담당한다.
- DOS의 COMMAND.COM과 같은 기능을 수행한다.
- 주기억장치에 상주하지 않고, 명령어가 포함된 파일 형태로 존재하며 보조 기억장치에서 교체 처리가 가능하다.
- 파이프라인 기능을 지원하고 입 · 출력 재지정을 통해 출력과 입력의 방향을 변경할 수 있다.
- 공용 Shell(Bourne Shell, C Shell, Korn Shell)이나 사용자 자신이 만든 Shell을 사용할 수 있다.

유틸리티(Utility Program)

- 일반 사용자가 작성한 응용 프로그램을 처리하는 데 사용한다.
- DOS에서의 외부 명령어에 해당된다.
- 유틸리티 프로그램에는 에디터, 컴파일러, 인터프리터, 디버거 등이 있다.

71 기억장치 관리 전략에는 반입(Fetch), 배치(Placement), 교체(Replacement) 전략이 있습니다. Paging 전략은 가상기억장치에서 사용되는 기법입니다.

기억장치의 관리 전략

- 반입(Fetch) 전략 : 보조기억장치에 보관중인 프로그램이나 데이터를 언제 주기억장치로 적재할 것인지를 결정하는 전략
- 배치(Placement) 전략 : 새로 반입되는 프로그램이나 데이터를 주기억장치의 어디에 위치시킬 것인지를 결정하는 전략
- 교체(Replacement) 전략 : 주기억장치의 모든 영역이 이미 사용중인 상태에서 새로운 프로그램이나 데이터를 주기억장치에 배치하려고 할 때, 이미 사용되고 있는 영역 중에서 어느 영역을 교체하여 사용할 것인지를 결정하는 전략

72 3개의 페이지를 수용할 수 있는 주기억장치이므로 아래 그림과 같이 3개의 페이지 프레임으로 표현할 수 있습니다.

참조 페이지	0	1	2	3	2	1	4	0
페이지 프레임	0	0	0	3	3	3	3	3
		1	1	1	1	1	4	4
			2	2	2	2	2	0
부재 발생	●	●	●	●			●	●

※ ● : 페이지 부재 발생

참조 페이지가 페이지 테이블에 없을 경우 페이지 결함(부재)이 발생됩니다. 초기에는 모든 페이지가 비어 있으므로 처음 0, 1, 2 페이지 적재 시 페이지 결함이 발생됩니다. FIFO 기법은 가장 먼저 들어와 있었던 페이지를 교체하는 기법이므로 참조 페이지 3을 참조할 때에는 0을 제거한 후 3을 가져오게 됩니다. 이러한 과정으로 모든 페이지에 대한 요구를 처리하고 나면 총 페이지 결함 발생 횟수는 6회입니다.

73 워킹 셋(Working Set)은 자주 참조하는 페이지들의 집합을 의미하는 것으로, 스래싱이 발생되지 않도록 워킹 셋을 계속하여 변경해 주기 때문에 프로세스 실행 중에 워킹 셋의 크기가 변경될 수 있습니다.

워킹 셋(Working Set)

- 프로세스가 일정 시간 동안 자주 참조하는 페이지들의 집합이다.
- 데닝(Denning)이 제안한 프로그램의 움직임에 대한 모델로, 프로그램의 Locality 특징을 이용한다.
- 자주 참조되는 워킹 셋을 주기억장치에 상주시킴으로써 페이지 부재 및 페이지 교체 현상이 줄어들어 프로세스의 기억장치 사용이 안정된다.
- 시간이 지남에 따라 자주 참조하는 페이지들의 집합이 변화하기 때문에 워킹 셋은 시간에 따라 변경된다.

74 스레드(Thread)는 동일 프로세스 환경에서 서로 독립적인 다중 수행이 가능합니다.

스레드(Thread)

- 프로세스 내에서의 작업 단위로서 시스템의 여러 자원을 할당받아 실행하는 프로그램의 단위이다.
- 하나의 프로세스에 하나의 스레드가 존재하는 경우에는 단일 스레드, 하나 이상의 스레드가 존재하는 경우에는 다중 스레드라고 한다.
- 프로세스의 일부 특성을 갖고 있기 때문에 경량(Light Weight) 프로세스라고도 한다.
- 스레드 기반 시스템에서 스레드는 독립적인 스케줄링의 최소 단위로서 프로세스의 역할을 담당한다.
- 동일 프로세스 환경에서 서로 독립적인 다중 수행이 가능하다.
- 스레드의 분류
  - 사용자 수준의 스레드 : 사용자가 만든 라이브러리를 사용하여 스레드를 운용
  - 커널 수준의 스레드 : 운영체제의 커널에 의해 스레드를 운용
- 스레드 사용의 장점
  - 하나의 프로세스를 여러 개의 스레드로 생성하여 병행성을 증진시킬 수 있다.
  - 하드웨어, 운영체제의 성능과 응용 프로그램의 처리율을 향상시킬 수 있다.
  - 응용 프로그램의 응답 시간(Response Time)을 단축시킬 수 있다.
  - 실행 환경을 공유시켜 기억장소의 낭비가 줄어든다.
  - 프로세스들 간의 통신이 향상된다.
  - 스레드는 공통적으로 접근 가능한 기억장치를 통해 효율적으로 통신한다.

75 준비 상태에서 대기하고 있는 프로세스 중 하나가 스케줄링되어 중앙처리장치를 할당받아 실행 상태로 전이되는 과정을 디스패치(Dispatch)라고 합니다.

프로세스 상태 전이 관련 용어

- 제출(Submit) : 작업을 처리하기 위해 사용자가 작업을 시스템에 제출한 상태
- 접수(Hold) : 제출된 작업이 스풀 공간인 디스크의 할당 위치에 저장된 상태
- 준비(Ready) : 프로세스가 프로세서를 할당받기 위해 기다리고 있는 상태
- 실행(Run) : 준비상태 큐에 있는 프로세스가 프로세서를 할당받아 실행되는 상태
- 대기(Wait), 보류, 블록(Block): 프로세스에 입 · 출력 처리가 필요하면 현재 실행 중인 프로세스가 중단되고, 입 · 출력 처리가 완료될 때까지 대기하고 있는 상태
- 종료(Terminated, Exit) : 프로세스의 실행이 끝나고 프로세스 할당이 해제된 상태
- Dispatch : 준비 상태에서 대기하고 있는 프로세스 중 하나가 프로세서를 할당받아 실행 상태로 전이되는 과정
- Wake Up : 입 · 출력 작업이 완료되어 프로세스가 대기 상태에서 준비 상태로 전이 되는 과정
- Spooling : 입 · 출력장치의 공유 및 상대적으로 느린 입 · 출력장치의 처리 속도를 보완하고 다중 프로그래밍 시스템의 성능을 향상시키기 위해 입 · 출력할 데이터를 직접 입 · 출력장치에 보내지 않고 나중에 한꺼번에 입 · 출력하기 위해 디스크에 저장하는 과정
- 교통량 제어기(Traffic Controller) : 프로세스의 상태에 대한 조사와 통보를 담당

76 IPv6은 16비트씩 8부분, 총 128비트로 구성되어 있습니다.

IPv6(Internet Protocol version 6)

- 현재 사용하고 있는 IP 주소 체계인 IPv4의 주소 부족 문제를 해결하기 위해 개발되었다.
- 128비트의 긴 주소를 사용하여 주소 부족 문제를 해결할 수 있으며, IPv4에 비해 자료 전송 속도가 빠르다.
- 인증성, 기밀성, 데이터 무결성의 지원으로 보안 문제를 해결할 수 있다.
- IPv4와 호환성이 뛰어나다.
- 주소의 확장성, 융통성, 연동성이 뛰어나며, 실시간 흐름 제어로 향상된 멀티미디어 기능을 지원한다.
- Traffic Class, Flow Label을 이용하여 등급별, 서비스별로 패킷을 구분할 수 있어 품질 보장이 용이하다.

77 양단 간(End-to-End)의 통신(전송)을 지원하는 기능은 전송 계층(Transport Layer)의 대표적인 기능입니다.

OSI 7계층

물리 계층	• 전송에 필요한 두 장치 간의 실제 접속과 절단 등 기계적, 전기적, 기능적, 절차적 특성에 대한 규칙을 정의함 • 물리적 전송 매체와 전송 신호 방식을 정의하며, RS-232C, X.21 등의 표준이 있음
데이터 링크 계층	• 두 개의 인접한 개방 시스템들 간에 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 함 • 송신 측과 수신 측의 속도 차이를 해결하기 위한 흐름 제어 기능을 함 • 프레임의 시작과 끝을 구분하기 위한 프레임의 동기화 기능을 함 • 오류의 검출과 회복을 위한 오류 제어 기능을 함 • 프레임의 순서적 전송을 위한 순서 제어 기능을 함 • HDLC, LAPB, LLC, LAPD, PPP 등의 표준이 있음
네트워크 계층	• 개방 시스템들 간의 네트워크 연결을 관리하는 기능과 데이터의 교환 및 중계 기능을 함 • 네트워크 연결을 설정, 유지, 해제하는 기능을 함 • 경로 설정(Routing), 데이터 교환 및 중계, 트래픽 제어, 패킷 정보 전송을 수행함 • 관련 표준으로는 X.25, IP 등이 있음
전송 (트랜스포트) 계층	• 논리적 안정과 균일한 데이터 전송 서비스를 제공함으로써 종단 시스템(End-to-End) 간에 투명한 데이터 전송을 가능하게 함 • OSI 7계층 중 하위 3계층과 상위 3계층의 인터페이스(Interface)를 담당함 • 종단 시스템(End-to-End) 간의 전송 연결 설정, 데이터 전송, 연결 해제 기능을 함 • 주소 설정, 다중화, 오류 제어, 흐름 제어를 수행함 • TCP, UDP 등의 표준이 있음
세션 계층	• 송 · 수신측 간의 관련성을 유지하고 대화 제어를 담당하는 계층 • 대화(회화) 구성 및 동기 제어, 데이터 교환 관리 기능을 함 • 송 · 수신측의 대화(회화) 동기를 위해 전송하는 정보의 일정한 부분에 체크점을 두어 정보의 수신 상태를 체크하며, 이때의 체크점을 동기점이라고 함 • 동기점은 오류가 있는 데이터의 회복을 위해 사용하는 것으로, 종류에는 소동기점과 대동기점이 있음
표현 계층	• 응용 계층으로부터 받은 데이터를 세션 계층에 보내기 전에 통신에 적당한 형태로 변환하고, 세션 계층에서 받은 데이터는 응용 계층에 맞게 변환하는 기능을 함 • 서로 다른 데이터 표현 형태를 갖는 시스템 간의 상호 접속을 위해 필요한 계층 • 코드 변환, 데이터 암호화, 데이터 압축, 구문 검색, 정보 형식(포맷) 변환, 문맥 관리 기능을 함
응용 계층	• 사용자(응용 프로그램)가 OSI 환경에 접근할 수 있도록 서비스를 제공함 • 응용 프로세스 간의 정보 교환, 전자 사서함, 파일 전송 등의 서비스를 제공함

78 문제의 지문에 제시된 기능을 가지고 있는 프로토콜은 TCP(Transmission Control Protocol)입니다.

- RTCP(Real-Time Control Protocol) : RTP(Real-time Transport Protocol) 패킷의 전송 품질을 제어하기 위한 제어 프로토콜
- FTP(File Transfer Protocol) : 컴퓨터와 컴퓨터 또는 컴퓨터와 인터넷 사이에서 파일을 주고받을 수 있도록 하는 원격 파일 전송 프로토콜
- UDP(User Datagram Protocol) : 데이터 전송 전에 연결을 설정하지 않는 비연결형 서비스를 제공하는 프로토콜

TCP/IP(Transmission Control Protocol/Internet Protocol)

- 인터넷에 연결된 서로 다른 기종의 컴퓨터들이 데이터를 주고받을 수 있도록 하는 표준 프로토콜이다.
- 1960년대 말 ARPA에서 개발하여 ARPANET(1972)에서 사용하기 시작했다.
- UNIX의 기본 프로토콜로 사용되었고, 현재 인터넷 범용 프로토콜로 사용된다.
- 다음과 같은 기능을 수행하는 TCP 프로토콜과 IP 프로토콜이 결합된 것을 의미한다.
- TCP(Transmission Control Protocol)
  - OSI 7계층의 전송 계층에 해당
  - 신뢰성 있는 연결형 서비스를 제공
  - 패킷의 다중화, 순서 제어, 오류 제어, 흐름 제어 기능을 제공
  - 스트림(Stream) 전송 기능 제공
  - TCP 헤더에는 Source/Destination Port Number, Sequence Number, Acknowledgment Number, Checksum 등이 포함된다.
- IP(Internet Protocol)
  - OSI 7계층의 네트워크 계층에 해당
  - 데이터그램을 기반으로 하는 비연결형 서비스를 제공
  - 패킷의 분해/조립, 주소 지정, 경로 선택 기능을 제공
  - 헤더의 길이는 최소 20Byte에서 최대 60Byte이다.
  - IP 헤더에는 Version, Header Length, Total Packet Length, Header Checksum, Source IP Address, Destination IP Address 등이 포함된다.

79 제어의 역흐름은 객체들의 제어를 개발자가 아닌 프레임워크가 담당합니다. ①번은 모듈화(Modularity), ②번은 재사용성(Reusability), ③번은 확장성(Extensibility)에 대한 내용입니다.

프레임워크의 특성

- 모듈화(Modularity) : 캡슐화를 통해 모듈화를 강화하고 설계 및 구현의 변경에 따른 영향을 최소화함으로서 소프트웨어의 품질을 향상시킴
- 재사용성(Reusability) : 재사용 가능한 모듈들을 제공함으로써 개발자의 생산성을 향상시킴
- 확장성(Extensibility) : 다형성(Polymorphism)을 통한 인터페이스 확장이 가능하여 다양한 형태와 기능을 가진 애플리케이션 개발이 가능함
- 제어의 역흐름(Inversion of Control) : 개발자가 관리하고 통제해야 하는 객체들의 제어를 프레임워크에 넘김으로써 생산성을 향상시킴

80 외부 소프트웨어를 구매하여 사용하는 패키지 소프트웨어에 비해 직접 개발하는 전용 개발 소프트웨어가 많은 인력과 시간이 소요됩니다.

패키지 소프트웨어

- 기능 요구사항 : 70% 이상 충족시키는 패키지 소프트웨어가 있는 경우 이용
- 안정성 : 품질이 검증되었고, 업계 표준 준용
- 라이선스 : 판매자
- 생산성 : 개발을 위한 인력과 시간이 절약됨
- 호환성 : 보장이 안 됨
- 유지보수 : 결함 발생 시 즉시 대응이 어려움

전용 개발 소프트웨어

- 기능 요구사항 : 모든 기능 요구사항 반영 가능
- 안정성 : 개발자의 역량에 따라 달라짐
- 라이선스 : 회사
- 생산성 : 개발을 위한 인력과 시간이 필요함
- 호환성 : 설계 단계부터 고려하여 개발
- 유지보수 : 결함 발생 시 즉시 대응이 가능

81 SPICE(ISO/IEC 15504)의 프로세스 범주에는 고객-공급자(Customer-Supplier), 공학(Engineering), 지원(Support), 관리(Management), 조직(Organization)이 있습니다.

SPICE(ISO/IEC 15504)의 프로세스 범주

- 고객-공급자(Customer-Supplier) 프로세스
  - 소프트웨어를 개발하여 고객에게 전달하는 것을 지원하고, 소프트웨어의 정확한 운용 및 사용을 위한 프로세스로 구성된다.
  - 구성 요소 : 인수, 공급, 요구 도출, 운영
  - 프로세스 수 : 10개
- 공학(Engineering) 프로세스
  - 시스템과 소프트웨어 제품의 명세화, 구현, 유지보수를 하는데 사용되는 프로세스로 구성된다.
  - 구성 요소 : 개발, 소프트웨어 유지보수
  - 프로세스 수 : 9개
- 지원(Support) 프로세스
  - 소프트웨어 생명 주기에서 다른 프로세스에 의해 이용되는 프로세스로 구성된다.
  - 구성 요소 : 문서화, 형상, 품질 보증, 검증, 확인, 리뷰, 감사, 품질 문제 해결
  - 프로세스 수 : 8개
- 관리(Management) 프로세스
  - 소프트웨어 생명 주기에서 프로젝트 관리자에 의해 사용되는 프로세스로 구성된다.
  - 구성 요소 : 관리, 프로젝트 관리, 품질 및 위험 관리
  - 프로세스 수 : 4개
- 조직(Organization) 프로세스
  - 조직의 업무 목적 수립과 조직의 업무 목표 달성을 위한 프로세스로 구성된다.
  - 구성 요소 : 조직 배치, 개선 활동 프로세스, 인력 관리, 기반 관리, 측정 도구, 재사용
  - 프로세스 수 : 9개

82 소프트웨어 개발 방법론의 테일러링 시 고려해야 할 외부적 요건에는 법적 제약사항과 표준 품질 기준이 있습니다. 보유 기술과 프로젝트 규모는 내부적 요건에 해당합니다.

소프트웨어 개발 방법론 테일러링

- 프로젝트 상황 및 특성에 맞도록 정의된 소프트웨어 개발 방법론의 절차, 사용기법 등을 수정 및 보완하는 작업이다.
- 고려해야 할 내부적 요건
  - 목표 환경 : 시스템의 개발 환경과 유형이 서로 다른 경우
  - 요구사항 : 프로젝트의 생명 주기 활동에서 개발, 운영, 유지보수 등 프로젝트에서 우선적으로 고려할 요구사항이 서로 다른 경우
  - 프로젝트 규모 : 비용, 구성원(인력), 기간 등 프로젝트의 규모가 서로 다른 경우
  - 보유 기술 : 프로세스, 개발 방법론, 산출물 등이 서로 다른 경우
- 고려해야 할 외부적 요건
  - 법적 제약사항 : 프로젝트별로 적용될 IT Compliance가 서로 다른 경우
  - 표준 품질 기준 : 금융, 제도 등 분야별 표준 품질 기준이 서로 다른 경우

- 테일러링 기법의 종류
  - 프로젝트 규모와 복잡도에 따른 테일러링 기법
  - 프로젝트 구성원에 따른 테일러링 기법
  - 팀내 방법론 지원에 따른 테일러링 기법
  - 자동화에 따른 테일러링 기법

83 개발 기간은 소프트웨어 비용 결정 요소 중 생산성 요소에 해당합니다.

소프트웨어 비용 결정 요소

- 프로젝트 요소
  - 제품 복잡도 : 소프트웨어의 종류에 따라 발생할 수 있는 문제점들의 난이도를 의미
  - 시스템 크기 : 소프트웨어의 규모에 따라 개발해야 할 시스템의 크기를 의미
  - 요구되는 신뢰도 : 일정 기간 내 주어진 조건하에서 프로그램이 필요한 기능을 수행하는 정도를 의미
- 자원 요소
  - 인적 자원 : 소프트웨어 개발 관련자들이 갖춘 능력 혹은 자질을 의미
  - 하드웨어 자원 : 소프트웨어 개발 시 필요한 장비와 워드프로세서, 프린터 등의 보조 장비를 의미
  - 소프트웨어 자원 : 소프트웨어 개발 시 필요한 언어 분석기, 문서화 도구 등의 개발 지원 도구를 의미
- 생산성 요소
  - 개발자 능력 : 개발자들이 갖춘 전문지식, 경험, 이해도, 책임감, 창의력 등을 의미
  - 개발 기간 : 소프트웨어를 개발하는 기간을 의미

84 전문가 감정 기법의 주관적인 편견을 보완하기 위해 많은 전문가의 의견을 종합하여 산정하는 비용 산정 기법은 델파이 기법입니다. 나머지 보기로 제시된 기법들의 개념도 확인하고 넘어가세요.

- 기능 점수(FP) 모형 : 총 기능 점수와 영향도를 이용하여 기능 점수(FP)를 구한 후 이를 이용해서 비용을 산정하는 기법
- LOC 기법 : 소프트웨어 각 기능의 원시 코드 라인 수의 비관치, 낙관치, 기대치를 측정하여 예측치를 구하고 이를 이용하여 비용을 산정하는 기법
- COCOMO 모형 : 원시 프로그램의 규모인 LOC(원시 코드 라인 수)에 의한 비용 산정 기법

하향식 비용 산정 기법

- 과거의 유사한 경험을 바탕으로 전문 지식이 많은 개발자들이 참여한 회의를 통해 비용을 산정하는 비과학적인 방법이다.
- 프로젝트의 전체 비용을 산정한 후 각 작업별로 비용을 세분화한다.
- 하향식 비용 산정 기법에는 전문가 감정 기법, 델파이 기법 등이 있다.
- 전문가 감정 기법
  - 조직 내에 있는 경험이 많은 두 명 이상의 전문가에게 비용 산정을 의뢰하는 기법이다.
  - 가장 편리하고 신속하게 비용을 산정할 수 있으며, 의뢰자로부터 믿음을 얻을 수 있다.
  - 새로운 프로젝트에는 과거의 프로젝트와 다른 요소들이 있다는 것을 간과할 수 있다.
  - 새로운 프로젝트와 유사한 프로젝트에 대한 경험이 없을 수 있다.
  - 개인적이고 주관적일 수 있다.
- 델파이 기법
  - 전문가 감정 기법의 주관적인 편견을 보완하기 위해 많은 전문가의 의견을 종합하여 산정하는 기법이다.
  - 전문가들의 편견이나 분위기에 지배되지 않도록 한 명의 조정자와 여러 전문가로 구성된다.

85
- 노력 = LOC/1인당 월평균 생산 라인 수 = 25000/500 = 50
- 개발 기간 = 노력/투입인원 = 50/5명 = 10개월

※ 단순하게 '25,000라인 / (5명 * 500라인) = 10'으로 계산해도 됩니다.

86 생산성은 '원시 코드 라인 수/노력'이며, 노력은 소프트웨어를 한 달 간 개발하는 데 소요되는 총 인원 또는 한 사람을 기준으로 몇 개월에 걸쳐 개발했느냐를 나타냅니다. 이를 계산하는 방법은 '투입 인원 × 개발 기간'입니다.

87 NFC(Near Field Communication, 근거리 무선 통신)는 고주파(HF)를 이용한 근거리 무선 통신 기술입니다. ④번의 내용은 클라우드 컴퓨팅(Cloud Computing)에 대한 설명입니다.

88 LAN에서는 주로 버스(Bus)형이나 링(Ring)형 구조를 사용합니다. 망(Mesh)형은 주로 공중 통신망에서 사용됩니다.

통신망의 구성 형태

- 성형(Star, 중앙 집중형) : 중앙에 중앙 컴퓨터가 있고, 이를 중심으로 단말장치들이 연결되는 중앙 집중식의 네트워크 구성 형태
- 링형(Ring, 루프형) : 컴퓨터와 단말장치들을 서로 이웃하는 것끼리 포인트 투 포인트(Point-to-Point) 방식으로 연결시킨 형태
- 버스형(Bus) : 한 개의 통신 회선에 여러 대의 단말장치가 연결되어 있는 형태
- 계층형(Tree, 분산형) : 중앙 컴퓨터와 일정 지역의 단말장치까지는 하나의 통신 회선으로 연결시키고, 이웃하는 단말장치는 일정 지역 내에 설치된 중간 단말장치로부터 다시 연결시키는 형태
- 망형(Mesh) : 모든 지점의 컴퓨터와 단말장치를 서로 연결한 형

태로, 노드의 연결성이 높음

89 문제의 지문에서 설명하고 있는 소프트웨어는 그레이웨어(Grayware)입니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- 매시업(Mashup) : 다수의 정보원이 제공하는 콘텐츠를 조합하여 하나의 서비스로 제공하는 웹 사이트 또는 애플리케이션
- RIA(Rich Internet Application) : 역동적이고 인터랙티브한 웹페이지를 제공하는 신개념의 플래시 웹페이지 제작 기술
- 뉴럴링크(Neuralink) : 사람의 뇌와 컴퓨터를 결합하는 기술

90 현실의 사물을 소프트웨어로 가상화한 모델로, 현실의 사물을 대신해 다양한 상황을 모의 실험하기 위한 용도로 사용되는 것은 디지털 트윈(Digital Twin)입니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- 소프트웨어 에스크로(Software Escrow) : 소프트웨어를 제3 기관에 임치하여 지식재산권을 보호하는 것
- 서비스형 소프트웨어(SaaS) : 소프트웨어의 여러 기능 중에서 사용자가 필요로 하는 서비스만 이용할 수 있도록 한 소프트웨어
- 오픈 그리드 서비스 아키텍처(OGSA) : 애플리케이션 공유를 위한 웹 서비스를 그리드 상에서 제공하기 위해 만든 개방형 표준

91 보안 가이드의 개발은 행정안전부가 아닌 한국인터넷진흥원(KISA)에서 수행해야 합니다.

활동 주체별 개발 보안 역할

- 행정안전부
  - 소프트웨어 개발 보안 정책을 총괄한다.
  - 소프트웨어 개발 보안 관련 법규, 지침, 제도를 정비한다.
  - 소프트웨어 보안 약점을 진단하는 사람의 양성 및 관련 업무를 수행한다.
- 한국인터넷진흥원(KISA)
  - 소프트웨어 개발 보안 정책 및 가이드를 개발한다.
  - 소프트웨어 개발 보안에 대한 기술을 지원하고, 교육과정 및 자격제도를 운영한다.
- 발주기관
  - 소프트웨어 개발 보안의 계획을 수립한다.
  - 소프트웨어 개발 보안 사업자 및 감리법인을 선정한다.
  - 소프트웨어 개발 보안의 준수 여부를 점검한다.
- 사업자
  - 소프트웨어 개발 보안 관련 기술 수준 및 적용 계획을 명시한다.
  - 소프트웨어 개발 보안 관련 인력을 대상으로 교육을 실시한다.
  - 소프트웨어 개발 보안 가이드를 참조하여 개발한다.
  - 자체적으로 보안 약점을 진단하고 제거한다.
  - 소프트웨어 보안 약점과 관련된 시정 요구사항을 이행한다.
- 감리법인
  - 감리 계획을 수립하고 협의한다.
  - 소프트웨어 보안 약점의 제거 여부 및 조치 결과를 확인한다.

92 소프트웨어의 안전 여부를 확인할 수 있도록 문서화하는 것은 구현 개발자의 역할입니다.

설계자(Designer)가 수행해야 할 보안활동

- 특정 기술에 대해 보안 요구사항의 만족성 여부를 확인한다.
- 문제 발생 시 최선의 문제 해결 방법을 결정한다.
- 애플리케이션 보안 수준에 대한 품질 측정을 지원한다.
- 많은 비용이 필요한 수정 요구사항을 최소화하기 위한 방법을 제공한다.
- 다른 소프트웨어와 통합할 때 발생할 수 있는 보안 위험에 대해 이해해야 한다.
- 소프트웨어에서 발견된 보안 위협에 대해 적절히 대응한다.

93 문제의 지문에서 괄호 안에 공통적으로 들어갈 용어는 멤리스터(Memristor)입니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- 엠디스크(Millennial DISC) : 한 번의 기록만으로 자료를 영구 보관할 수 있는 광 저장장치
- 멤스(MEMS) : 기계 구조를 다양한 기술로 미세 가공하여 전기 기계적 동작을 할 수 있도록 한 초미세 장치
- RAID(Redundant Array of Inexpensive Disk) : 오류 검출 및 복구를 위해 디스크 배열에 오류 검출 및 복구를 위한 여분의 디스크들을 추가하는 기술

94
- 내부 정보를 암호화 하는 방법은 암호화 분리이며, 암호화 분리는 Secure OS에서 구현되는 자원 보호 방법 중 구현이 가장 복잡합니다.
- ①번은 논리적 분리, ②번은 시간적 분리, ③번은 물리적 분리에 대한 내용입니다.

Secure OS

- 기존의 운영체제(OS)에 내재된 보안 취약점을 해소하기 위해 보안 기능을 갖춘 커널을 이식하여 외부의 침입으로부터 시스템 자원을 보호하는 운영체제를 의미한다.
- 보안 커널은 보안 기능을 갖춘 커널을 의미하며, TCB를 기반으로 참조 모니터의 개념을 구현하고 집행한다.
- 보안 커널의 보호 대상에는 메모리와 보조기억장치, 그리고 그곳에 저장된 데이터, 하드웨어 장치, 자료 구조, 명령어, 각종 보호 메커니즘 등이 있다.
- 보호 방법(복잡한 것부터)
  - 암호적 분리(Cryptographic Separation) : 내부 정보를 암호화하는 방법

- 논리적 분리(Logical Separation) : 프로세스의 논리적 구역을 지정하여 구역을 벗어나는 행위를 제한하는 방법
- 시간적 분리(Temporal Separation) : 동일 시간에 하나의 프로세스만 수행되도록 하여 동시 실행으로 발생하는 보안 취약점을 제거하는 방법
- 물리적 분리(Physical Separation) : 사용자별로 특정 장비만 사용하도록 제한하는 방법

95 병행제어 기법에서 병행수행되던 트랜잭션들 중 어느 하나에 문제가 생겨 Rollback하는 경우 다른 트랜잭션도 함께 Rollback되는 현상을 연쇄 복귀(Cascading Rollback)라고 합니다.

병행수행의 문제점

- 갱신 분실(Lost Update) : 두 개 이상의 트랜잭션이 같은 자료를 공유하여 갱신할 때 갱신 결과의 일부가 없어지는 현상
- 비완료 의존성(Uncommitted Dependency) : 하나의 트랜잭션 수행이 실패한 후 회복되기 전에 다른 트랜잭션이 실패한 갱신 결과를 참조하는 현상
- 모순성(Inconsistency) : 두 개의 트랜잭션이 병행 수행될 때 원치 않는 자료를 이용함으로써 발생하는 문제
- 연쇄 복귀(Cascading Rollback) : 병행수행되던 트랜잭션들 중 어느 하나에 문제가 생겨 Rollback하는 경우 다른 트랜잭션도 함께 Rollback되는 현상

96 소프트웨어 개발에 있어 충족시켜야 할 3대 주요 보안 요소에는 기밀성(Confidentiality), 무결성(Integrity), 가용성(Availability)이 있습니다.

보안 요소

- 소프트웨어 개발에 있어 충족시켜야할 요소 및 요건을 의미한다.
- 기밀성(Confidentiality) : 시스템 내의 정보와 자원은 인가된 사용자에게만 접근이 허용됨
- 무결성(Integrity) : 시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있음
- 가용성(Availability) : 인가받은 사용자는 언제라도 사용할 수 있음
- 인증(Authentication) : 시스템 내의 정보와 자원을 사용하려는 사용자가 합법적인 사용자인지를 확인하는 모든 행위
- 부인 방지(NonRepudiation) : 데이터를 송 · 수신한 자가 송 · 수신 사실을 부인할 수 없도록 송 · 수신 증거를 제공함

97 데이터 표준화는 데이터 모델이나 DB에서 정의할 수 있는 모든 오브젝트를 대상으로 합니다.

데이터 표준화

- 시스템을 구성하는 데이터 요소의 명칭, 정의, 형식, 규칙에 대한 원칙을 수립하고 적용하는 것을 의미한다.
- 데이터 모델이나 DB에서 정의할 수 있는 모든 오브젝트를 대상으로 데이터 표준화를 수행해야 한다.

- 데이터 표준의 종류

표준 단어	업무에서 사용하고 일정한 의미를 갖고 있는 최소 단위의 단어를 의미
표준 도메인	문자형, 숫자형, 날짜형, 시간형과 같이 컬럼을 성질에 따라 그룹핑한 개념
표준 코드	선택할 수 있는 값을 정형화하기 위해 기준에 맞게 이미 정의된 코드 값으로, 도메인의 한 유형
표준 용어	단어, 도메인, 코드 표준이 정의되면 이를 바탕으로 표준 용어를 구성

- 데이터 표준화의 대상

데이터 명칭	데이터를 유일하게 구별할 수 있는 유일성, 의미 전달의 충분성, 그리고 업무적 보편성을 갖는 이름으로 정의
데이터 정의	데이터를 제3자의 입장에서도 쉽게 이해할 수 있도록 해당 데이터가 의미하는 범위 및 자격 요건 등을 규정
데이터 형식	업무 규칙 및 사용 목적과 유사한 데이터에 대해 일관되게 데이터 형식을 정의함으로써 데이터 입력 오류, 통제 위험 등을 최소화
데이터 규칙	기본 값, 허용 값, 허용 범위 등과 같이 발생할 수 있는 데이터 값을 사전에 지정함으로써 데이터의 정합성 및 완전성을 향상

98 임의의 길이의 입력 데이터나 메시지를 고정된 길이의 값이나 키(key)로 변환하는 알고리즘은 해시 알고리즘(Hash Algorithm)입니다. 나머지 보기로 제시된 알고리즘들의 개념도 확인하고 넘어가세요.

- RSA(Rivest Shamir Adleman) : 큰 숫자는 소인수 분해하기 어렵다는 점을 기반으로 만들어진 공개키 암호화 알고리즘
- DES(Data Encryption Standard) : 미국 NBS에서 발표한 개인키 암호화 알고리즘
- PKI(Public Key Infrastructure, 공개키 기반 구조) : 공개키 암호 시스템을 안전하게 사용하고 관리하기 위한 정보 보호 표준 방식

해시(Hash)

- 임의의 길이의 입력 데이터나 메시지를 고정된 길이의 값이나 키로 변환하는 것을 의미한다.
- 해시 알고리즘을 해시 함수라고 부르며, 해시 함수로 변환된 값이나 키를 해시값 또는 해시키라고 부른다.
- 데이터의 암호화, 무결성 검증을 위해 사용될 뿐만 아니라 정보보호의 다양한 분야에서 활용된다.
- 종류 : SHA 시리즈, MD5, N-NASH, SNEFRU 등

99 문제의 지문에 제시된 네트워크 공격 방법은 APT(Advanced Persistent Threats, 지능형 지속 위협)입니다. 나머지 보기로 제시된 네트워크 공격 방법들의 개념도 확인하고 넘어가세요.

- 스미싱(Smishing) : 문자 메시지(SMS)를 이용해 사용자의 개인 신용 정보를 빼내는 수법
- 큐싱(Qshing) : QR코드를 통해 악성 앱의 다운로드를 유도하거나 악성 프로그램을 설치하도록 하는 금융사기 기법

- 분산 서비스 거부 공격(DDoS; Distributed Denial of Service Attack) : 여러 곳에 분산된 공격 지점에서 한 곳의 서버에 대해 대량의 트래픽을 집중하여 자원을 고갈시키는 기법

100 ②번의 내용은 침입 방지 시스템에 대한 설명입니다.

※ 침입 방지 시스템(IPS; Intrusion Prevention System) : 방화벽과 침입 탐지 시스템을 결합한 보안 솔루션

침입 탐지 시스템(IDS; Intrusion Detection System)

- 컴퓨터 시스템의 비정상적인 사용, 오용, 남용 등을 실시간으로 탐지하는 시스템이다.
- 방화벽과 같은 침입 차단 시스템만으로는 내부 사용자의 불법적인 행동과 외부 해킹에 100% 완벽하게 대처할 수는 없다.
- 문제가 발생한 경우 모든 내 · 외부 정보의 흐름을 실시간으로 차단하기 위해 해커 침입 패턴에 대한 추적과 유해 정보 감시가 필요하다.
- 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
- 침입 탐지 시스템의 위치
  - 패킷이 라우터로 들어오기 전 : 네트워크에 시도되는 모든 공격을 탐지할 수 있음
  - 라우터 뒤 : 라우터에 의해 패킷 필터링을 통과한 공격을 탐지할 수 있음
  - 방화벽 뒤 : 내부에서 외부로 향하는 공격을 탐지할 수 있음
  - 내부 네트워크 : 내부에서 내부 네트워크의 해킹 공격을 탐지할 수 있음
  - DMZ : 외부 인터넷에 서비스를 제공하는 서버가 위치하는 네트워크로, 강력한 외부 공격이나 내부 공격으로부터 중요 데이터를 보호하거나 서버의 서비스 중단을 방지할 수 있음

정보처리기사 필기 최종점검 모의고사 2회 해설

1 애자일 모형(Agile Model)은 변화에 기민하고 유연하게 대응하는 것이 특징인 모형으로, 몇 개월이 아닌 몇 주 단위로 실행되는 소프트웨어를 제공합니다.

애자일 개발 12가지 실행 지침

- 유용한 소프트웨어를 빠르고, 지속적으로 제공하여 고객을 만족시킨다.
- 개발 막바지라도 요구사항 변경을 적극 수용한다.
- 몇 개월이 아닌 몇 주 단위로 실행되는 소프트웨어를 제공한다.
- 고객과 개발자가 프로젝트 기간에 함께 일한다.
- 개발에 대한 참여 의지가 확실한 사람들로 팀을 구성하고, 필요한 개발 환경과 지원을 제공하며, 일을 잘 끝낼 수 있도록 신뢰한다.
- 같은 사무실에서 얼굴을 맞대고 의견을 나눈다.
- 개발의 진척도를 확인하는 1차 기준은 작동하는 소프트웨어이다.
- 지속 가능한 개발을 장려하고 일정한 속도로 개발을 진행한다.
- 기술적 우수성과 좋은 설계에 지속적인 관심을 기울이면 민첩성이 향상된다.
- 단순화를 추구한다.
- 최상의 아키텍처, 명확한 요구사항, 최상의 설계는 자기 스스로 일을 주도하는 조직적인 팀으로부터 나온다.
- 더 효과적인 팀이 될 수 있는 방안을 정기적으로 깊이 고민하고 그에 따라 팀의 행동을 조정한다.

2 나선형 모형(Spiral Model)은 보헴(Boehm)이 제안한 것으로, 폭포수 모형과 프로토타입 모형의 장점에 위험 분석 기능을 추가한 모형입니다. 나머지 보기로 제시된 소프트웨어 생명주기 모델의 개념도 확인하고 넘어가세요.

- 폭포수 모형(Waterfall Model) : 폭포에서 한번 떨어진 물은 거슬러 올라갈 수 없듯이 소프트웨어 개발도 이전 단계로 돌아갈 수 없다는 전제하에 각 단계를 확실히 매듭짓고 그 결과를 철저하게 검토하여 승인 과정을 거친 후에 다음 단계를 진행하는 개발 방법론
- 프로토타입 모형(Prototype Model, 원형 모형) : 사용자의 요구사항을 정확히 파악하기 위해 실제 개발될 소프트웨어에 대한 견본(시제)품(Prototype)을 만들어 최종 결과물을 예측하는 모형
- 애자일 모형(Agile Model) : 애자일은 '민첩한', '기민한'이라는 의미로, 고객의 요구사항 변화에 유연하게 대응할 수 있도록 일정한 주기를 반복하면서 개발하는 방법론

3 ④번의 내용은 스프린트 회고(Retrospective)에 대한 설명입니다. 스프린트 검토 회의(Sprint Review)에서는 부분 또는 전체 완성 제품이 요구사항에 잘 부합되는지 사용자가 포함된 참석자 앞에서 테스팅을 수행합니다.

스크럼 개발 프로세스

- 제품 백로그(Product Backlog) : 제품 개발에 필요한 모든 요구사항(User Story)을 우선순위에 따라 나열한 목록
- 스프린트 계획 회의(Sprint Planning Meeting) : 제품 백로그 중 이번 스프린트에서 수행할 작업을 대상으로 단기 일정을 수립하는 회의
- 스프린트(Sprint) : 실제 개발 작업을 진행하는 과정으로, 보통 2~4주 정도의 기간 내에서 진행
- 일일 스크럼 회의(Daily Scrum Meeting) : 모든 팀원이 매일 약속된 시간에 약 15분 정도의 짧은 시간동안 진행 상황을 점검하는 회의
- 스프린트 검토 회의(Sprint Review) : 부분 또는 전체 완성 제품이 요구사항에 잘 부합되는지 사용자가 포함된 참석자 앞에서 테스팅을 수행하는 회의
- 스프린트 회고(Sprint Retrospective) : 스프린트 주기를 되돌아보며 정해놓은 규칙을 잘 준수했는지, 개선할 점은 없는지 등을 확인하고 기록하는 것

4 XP(eXtreme Programming)는 수시로 발생하는 고객의 요구사항에 유연하게 대응하기 위해 고객의 참여와 개발 과정의 반복을 극대화하여 개발 생산성을 향상시키는 방법으로, 고객의 요구 변화에 신속하게 대응하기 위해서는 릴리즈 기간을 가능한 짧게 반복해야 합니다.

XP의 주요 실천 방법(Practice)

- Pair Programming(짝 프로그래밍) : 다른 사람과 함께 프로그래밍을 수행함으로써 개발에 대한 책임을 공동으로 나눠 갖는 환경을 조성함
- Test-Driven Development(테스트 주도 개발)
  - 개발자가 실제 코드를 작성하기 전에 테스트 케이스를 먼저 작성하므로 자신이 무엇을 해야 할지를 정확히 파악함
  - 테스트가 지속적으로 진행될 수 있도록 자동화된 테스팅 도구(구조, 프레임워크)를 사용함
- Whole Team(전체 팀) : 개발에 참여하는 모든 구성원(고객 포함)들은 각자 자신의 역할이 있고 그 역할에 대한 책임을 가져야 함
- Continuous Integration(계속적인 통합) : 모듈 단위로 나눠서 개발된 코드들은 하나의 작업이 마무리될 때마다 지속적으로 통합함
- Design Improvement(디자인 개선) 또는 Refactoring(리팩토링) : 프로그램 기능의 변경 없이, 단순화, 유연성 강화 등을 통해 시스템을 재구성함
- Small Releases(소규모 릴리즈) : 릴리즈 기간을 짧게 반복함으로써 고객의 요구 변화에 신속히 대응함

5 포함(Composition) 관계는 집합 관계의 특수한 형태로, 포함하는 사물의 변화가 포함되는 사물에게 영향을 미치는 관계를 의미합니다. ①번은 집합(Aggregation) 관계에 대한 설명입니다.

UML의 관계(Relationships)

- 연관(Association) 관계 : 2개 이상의 사물이 서로 관련되어 있음
- 집합(Aggregation) 관계 : 하나의 사물이 다른 사물에 포함되어 있는 관계
- 포함(Composition) 관계 : 집합 관계의 특수한 형태로, 포함하는 사물의 변화가 포함되는 사물에게 영향을 미치는 관계
- 일반화(Generalization) 관계 : 하나의 사물이 다른 사물에 비해 더 일반적인지 구체적인지를 표현하는 관계
- 의존(Dependency) 관계 : 연관 관계와 같이 사물 사이에 서로 연관은 있으나 필요에 의해 서로에게 영향을 주는 짧은 시간 동안만 연관을 유지하는 관계
- 실체화(Realization) 관계 : 사물이 할 수 있거나 해야 하는 기능(행위, 인터페이스)으로 서로를 그룹화 할 수 있는 관계

6 배치 다이어그램은 구조적 다이어그램의 종류 중 하나로, 결과물, 프로세스, 컴포넌트 등 물리적 요소들의 위치를 표현하고 노드와 의사소통(통신) 경로로 표현하는 것이 특징입니다.

UML의 구조적(Structural) 다이어그램

- 클래스 다이어그램(Class Diagram)
  - 클래스와 클래스가 가지는 속성, 클래스 사이의 관계를 표현한다.
  - 시스템의 구조를 파악하고 구조상의 문제점을 도출할 수 있다.
- 객체 다이어그램(Object Diagram) : 클래스에 속한 사물(객체)들, 즉 인스턴스(Instance)를 특정 시점의 객체와 객체 사이의 관계로 표현함
- 컴포넌트 다이어그램(Component Diagram)
  - 실제 구현 모듈인 컴포넌트 간의 관계나 컴포넌트 간의 인터페이스를 표현한다.
  - 구현 단계에서 사용되는 다이어그램이다.
- 배치 다이어그램(Deployment Diagram)
  - 결과물, 프로세스, 컴포넌트 등 물리적 요소들의 위치를 표현한다.
  - 노드와 의사소통(통신) 경로로 표현한다.
  - 구현 단계에서 사용되는 다이어그램이다.
- 복합체 구조 다이어그램(Composite Structure Diagram) : 클래스나 컴포넌트가 복합 구조를 갖는 경우 그 내부 구조를 표현함
- 패키지 다이어그램(Package Diagram) : 유스케이스나 클래스 등의 모델 요소들을 그룹화한 패키지들의 관계를 표현함

7 사용자 인터페이스(UI)를 설계할 때 버튼이나 조작 방법 등은 새로운 방식 보다는 일관성 있고 표준화된 형태로 제공하여 사용자가 쉽게 습득하고 사용할 수 있도록 설계해야 합니다.

사용자 인터페이스(UI; User Interface)의 설계 지침

- 사용자 중심 : 사용자가 쉽게 이해하고 편리하게 사용할 수 있는 환경을 제공하며, 실사용자에 대한 이해가 바탕이 되어야 함
- 일관성 : 버튼이나 조작 방법 등을 일관성 있게 제공하므로 사용자가 쉽게 기억하고 습득할 수 있게 설계해야 함
- 단순성 : 조작 방법을 단순화시켜 인지적 부담을 감소시켜야 함
- 결과 예측 가능 : 작동시킬 기능만 보고도 결과를 미리 예측할 수 있게 설계해야 함
- 가시성 : 메인 화면에 주요 기능을 노출시켜 최대한 조작이 쉽도록 설계해야 함
- 표준화 : 기능 구조와 디자인을 표준화하여 한 번 학습한 이후에는 쉽게 사용할 수 있도록 설계해야 함
- 접근성 : 사용자의 연령, 성별, 인종 등 다양한 계층이 사용할 있도록 설계해야 함
- 명확성 : 사용자가 개념적으로 쉽게 인지할 수 있도록 설계해야 함
- 오류 발생 해결 : 오류가 발생하면 사용자가 쉽게 인지할 수 있도록 설계해야 함

8 이클립스(Eclipse)는 개발에 필요한 환경을 모두 제공하는 통합 개발 환경(IDE) 도구입니다.

UI 설계 도구의 종류

- 와이어프레임(Wireframe) : 페이지에 대한 개략적인 레이아웃이나 UI 요소 등에 대한 뼈대를 설계하는 단계
- 목업(Mockup) : 디자인, 사용 방법 설명, 평가 등을 위해 와이어프레임보다 좀 더 실제 화면과 유사하게 만든 정적인 형태의 모형
- 스토리보드(Story Board) : 와이어프레임에 콘텐츠에 대한 설명, 페이지 간 이동 흐름 등을 추가한 문서
- 프로토타입(Prototype) : 와이어프레임나 스토리보드 등에 인터랙션을 적용함으로써 실제 구현된 것처럼 테스트가 가능한 동적인 형태의 모형
- 유스케이스(Use Case) : 사용자가 원하는 목표를 달성하기 위해 수행할 내용을 기술한 것

9 유스케이스(Use Case)는 사용자의 요구사항을 정리하고 기록하기 위한 도구로, 현재 진행 상태 등을 공유하는 데에는 적합하지 않습니다. 현재 진행 상태 등을 공유하는 데 적합한 도구는 와이어프레임(Wireframe)입니다.

10 데이터 추상화는 데이터의 세부적인 속성이나 용도를 정의하지 않고, 데이터 구조를 대표할 수 있는 표현으로 대체하는 방법입니다. ③번의 내용은 과정 추상화에 대한 설명입니다.

추상화(Abstraction)

- 추상화는 문제의 전체적이고 포괄적인 개념을 설계한 후 차례로 세분화하여 구체화시켜 나가는 것이다.
- 인간이 복잡한 문제를 다룰 때 가장 기본적으로 사용하는 방법으로, 완전한 시스템을 구축하기 전에 그 시스템과 유사한 모델을 만들어서 여러 가지 요인들을 테스트할 수 있다.
- 추상화는 최소의 비용으로 실제 상황에 대처할 수 있고, 시스템의 구조 및 구성을 대략적으로 파악할 수 있게 해준다.
- 추상화의 유형
  - 과정 추상화 : 자세한 수행 과정을 정의하지 않고, 전반적인 흐름만 파악할 수 있게 설계하는 방법
  - 데이터 추상화 : 데이터의 세부적인 속성이나 용도를 정의하지 않고, 데이터 구조를 대표할 수 있는 표현으로 대체하는 방법
  - 제어 추상화 : 이벤트 발생의 정확한 절차나 방법을 정의하지 않고, 대표할 수 있는 표현으로 대체하는 방법

11 UI 프로토타입을 사용할 경우 프로토타입에 사용자의 모든 요구사항을 반영하기 위해 반복적인 개선 및 보완 작업을 수행합니다. 이로 인해 작업 시간이 증가할 수 있고, 필요 이상의 자원이 소모될 수 있습니다.

UI 프로토타입의 장 · 단점

장점	• 사용자를 설득하고 이해시키기 쉬움 • 요구사항과 기능의 불일치 등으로 인한 혼선을 예방할 수 있어 개발 시간을 줄일 수 있음 • 사전에 오류를 발견할 수 있음
단점	• 프로토타입에 사용자의 모든 요구사항을 반영하기 위한 반복적인 개선 및 보완 작업 때문에 작업 시간을 증가 시킬 수 있고, 필요 이상으로 자원을 소모할 수 있음 • 부분적으로 프로토타이핑을 진행하다보면 중요한 작업이 생략될 수 있음

12 소프트웨어 아키텍처의 설계 과정은 설계 목표 설정, 시스템 타입 결정, 아키텍처 패턴 적용, 서브시스템 구체화, 검토 순입니다.

소프트웨어 아키텍처의 설계 과정

❶ 설계 목표 설정 : 시스템의 개발 방향을 명확히 하기 위해 설계에 영향을 주는 비즈니스 목표, 우선순위 등의 요구사항을 분석하여 전체 시스템의 설계 목표를 설정함

❷ 시스템 타입 결정 : 시스템과 서브시스템의 타입을 결정하고, 설계 목표와 함께 고려하여 아키텍처 패턴을 선택함

❸ 아키텍처 패턴 적용 : 아키텍처 패턴을 참조하여 시스템의 표준 아키텍처를 설계함

❹ 서브시스템 구체화 : 서브시스템의 기능 및 서브시스템 간의 상호작용을 위한 동작과 인터페이스를 정의함

❺ 검토 : 아키텍처가 설계 목표에 부합하는지, 요구사항이 잘 반영되었는지, 설계의 기본 원리를 만족하는지 등을 검토함

13 아키텍처 패턴은 시스템의 구조를 이해하는 것이 쉬워 개발에 참여하지 않은 사람도 손쉽게 유지보수를 수행할 수 있습니다.

아키텍처 패턴

- 아키텍처를 설계할 때 참조할 수 있는 전형적인 해결 방식 또는 예제를 의미한다.
- 소프트웨어 시스템의 구조를 구성하기 위한 기본적인 윤곽을 제시한다.
- 서브시스템들과 그 역할을 정의하고 있으며, 서브시스템 사이의 관계와 여러 규칙 · 지침 등이 포함되어 있다.
- 아키텍처 스타일 또는 표준 아키텍처라고도 한다.
- 아키텍처 패턴의 장점
  - 시행착오를 줄여 개발 시간을 단축시키고, 고품질의 소프트웨어를 생산할 수 있다.
  - 검증된 구조로 개발하기 때문에 안정적인 개발이 가능하다.
  - 이해관계자들이 공통된 아키텍처를 공유할 수 있어 의사소통이 간편해진다.
  - 시스템의 구조를 이해하는 것이 쉬워 개발에 참여하지 않은 사람도 손쉽게 유지보수를 수행할 수 있다.
  - 시스템의 특성을 개발 전에 예측하는 것이 가능해진다.

14 객체가 반응할 수 있는 메시지(Message)의 집합을 행위라고 하며, 객체는 메시지를 통해 행위의 특징을 나타낼 수 있습니다.

객체(Object)의 특성

- 객체는 독립적으로 식별 가능한 이름을 가지고 있다.
- 객체가 가질 수 있는 조건을 상태(State)라고 하는데, 일반적으로 상태는 시간에 따라 변한다.
- 객체와 객체는 상호 연관성에 의한 관계가 형성된다.
- 객체가 반응할 수 있는 메시지(Message)의 집합을 행위라고 하며, 객체는 메시지를 통해 행위의 특징을 나타낼 수 있다.
- 객체는 일정한 기억장소를 가지고 있다.

15 모듈이 파라미터(매개변수)나 인수로 데이터를 주고받는 것을 데이터 결합도(Data Coupling)라고 합니다.

결합도(Coupling)

- 결합도는 모듈 간에 상호 의존하는 정도 또는 두 모듈 사이의 연관 관계를 의미한다.
- 다양한 결합으로 모듈을 구성할 수 있으나 결합도가 약할수록 품질이 높고, 강할수록 품질이 낮다.
- 결합도가 강하면 시스템 구현 및 유지보수 작업이 어렵다.
- 결합도의 종류
  - 자료 결합도(Data Coupling) : 모듈 간의 인터페이스가 자료 요소로만 구성될 때의 결합도
  - 스탬프(검인) 결합도(Stamp Coupling) : 모듈 간의 인터페이스로 배열이나 레코드 등의 자료 구조가 전달될 때의 결합도

- 제어 결합도(Control Coupling) : 한 모듈에서 다른 모듈로 논리적인 흐름을 제어하는 데 사용하는 제어 요소(Function code, Switch, Tag, Flag)가 전달될 때의 결합도
- 외부 결합도(External Coupling) : 어떤 모듈에서 외부로 선언한 데이터(변수)를 다른 모듈에서 참조할 때의 결합도
- 공통(공유) 결합도(Common Coupling) : 공유되는 공통 데이터 영역을 여러 모듈이 사용할 때의 결합도
- 내용 결합도(Content Coupling) : 한 모듈이 다른 모듈의 내부 기능 및 그 내부 자료를 직접 참조하거나 수정할 때의 결합도

16 우연적 응집도(Coincidental Cohesion)는 모듈 내부의 각 구성 요소들이 서로 관련 없는 요소로만 구성된 경우의 응집도입니다. ②번의 내용은 순차적 응집도(Sequential Cohesion)에 대한 설명입니다.

응집도(Cohesion)

- 정보 은닉 개념을 확장한 것으로, 명령어나 호출문 등 모듈의 내부 요소들의 서로 관련되어 있는 정도, 즉 모듈이 독립적인 기능으로 정의되어 있는 정도를 의미한다.
- 다양한 기준으로 모듈을 구성할 수 있으나 응집도가 강할수록 품질이 높고, 약할수록 품질이 낮다.
- 응집도의 종류
  - 기능적 응집도(Functional Cohesion) : 모듈 내부의 모든 기능 요소들이 단일 문제와 연관되어 수행될 경우의 응집도
  - 순차적 응집도(Sequential Cohesion) : 모듈 내 하나의 활동으로부터 나온 출력 데이터를 그 다음 활동의 입력 데이터로 사용할 경우의 응집도
  - 교환(통신)적 응집도(Communication Cohesion) : 동일한 입력과 출력을 사용하여 서로 다른 기능을 수행하는 구성 요소들이 모였을 경우의 응집도
  - 절차적 응집도(Procedural Cohesion) : 모듈이 다수의 관련 기능을 가질 때 모듈 안의 구성 요소들이 그 기능을 순차적으로 수행할 경우의 응집도
  - 시간적 응집도(Temporal Cohesion) : 특정 시간에 처리되는 몇 개의 기능을 모아 하나의 모듈로 작성할 경우의 응집도
  - 논리적 응집도(Logical Cohesion) : 유사한 성격을 갖거나 특정 형태로 분류되는 처리 요소들로 하나의 모듈이 형성되는 경우의 응집도
  - 우연적 응집도(Coincidental Cohesion) : 모듈 내부의 각 구성 요소들이 서로 관련 없는 요소로만 구성된 경우의 응집도

17 미들웨어(Middleware)는 미들(Middle)과 소프트웨어(Software)의 합성어로, 운영체제와 해당 운영체제에서 실행되는 응용 프로그램 사이에서 운영체제가 제공하는 서비스 이외에 추가적인 서비스를 제공하는 소프트웨어입니다. 나머지 보기로 제시된 용어의 개념도 확인하고 넘어가세요.

- ORM(Object-Relational Mapping) : 객체지향 프로그래밍의 객체(Object)와 관계형(Relational) 데이터베이스의 데이터를 연결(Mapping)하는 기술을 의미함
- 객체지향 프로그래밍(OOP; Object Oriented Programming) : 현실 세계의 개체(Entity)를 기계의 부품처럼 하나의 객체(Object)로 만들어, 기계적인 부품들을 조립하여 제품을 만들듯이 소프트웨어를 개발할 때에도 객체들을 조립해서 작성할 수 있는 기법을 의미함
- 인터페이스(Interface) : 서로 다른 두 시스템이나 소프트웨어 등을 서로 이어주는 부분 또는 접속 장치를 의미함

18 대량의 송 · 수신 데이터를 처리할 때 사용하는 방식은 배치 방식입니다.

인터페이스 처리 유형

- 실시간 방식 : 사용자가 요청한 내용을 바로 처리해야 할 때 사용하는 방식
- 지연 처리 방식 : 데이터를 매건 단위로 처리할 경우 비용이 많이 발생할 때 사용하는 방식
- 배치 방식 : 대량의 데이터를 처리할 때 사용하는 방식

19 RPC(Remote Procedure Call)는 응용 프로그램의 프로시저를 사용하여 원격 프로시저를 마치 로컬 프로시저처럼 호출하는 방식의 미들웨어입니다. ①번은 TP-Monitor(Transaction Processing Monitor)에 대한 설명입니다.

미들웨어(Middleware)의 종류

- DB(DataBase) : 데이터베이스 벤더(Vendor)에서 제공하는 클라이언트에서 원격의 데이터베이스와 연결하기 위한 미들웨어
- RPC(Remote Procedure Call) : 응용 프로그램의 프로시저를 사용하여 원격 프로시저를 마치 로컬 프로시저처럼 호출하는 방식의 미들웨어
- MOM(Message Oriented Middleware) : 메시지 기반의 비동기형 메시지를 전달하는 방식의 미들웨어
- TP-Monitor(Transaction Processing Monitor) : 항공기나 철도 예약 업무 등과 같은 온라인 트랜잭션 업무에서 트랜잭션을 처리 및 감시하는 미들웨어
- ORB(Object Request Broker) : 객체 지향 미들웨어로 코바(CORBA) 표준 스펙을 구현한 미들웨어
- WAS(Web Application Server) : 사용자의 요구에 따라 변하는 동적인 콘텐츠를 처리하기 위해 사용되는 미들웨어

20 아키텍처 패턴이 전체 시스템의 구조를 설계하기 위한 참조 모델이라면, 디자인 패턴은 서브시스템에 속하는 컴포넌트들과 그 관계를 설계하기 위한 참조 모델이기 때문에 아키텍처 패턴보다 하위 수준의 설계에 사용됩니다.

디자인 패턴(Design Pattern)

- 각 모듈의 세분화된 역할이나 모듈들 간의 인터페이스와 같은 코드를 작성하는 수준의 세부적인 구현 방안을 설계할 때 참조할 수 있는 전형적인 해결 방식 또는 예제를 의미한다.
- 재사용할 수 있는 기본형 코드들이 포함되어 있다.
- '바퀴를 다시 발명하지 마라(Don't reinvent the wheel)'라는 말과 같이, 개발 과정 중에 문제가 발생하면 새로 해결책을 구상하는 것보다 문제에 해당하는 디자인 패턴을 참고하여 적용하는 것이 더 효율적이다.
- 한 패턴에 변형을 가하거나 특정 요구사항을 반영하면 유사한 형태의 다른 패턴으로 변화되는 특징이 있다.
- 1995년 GoF(Gang of Four)라고 불리는 에릭 감마(Erich Gamma), 리차드 헬름(Richard Helm), 랄프 존슨(Ralph Johnson), 존 블리시디스(John Vlissides)가 처음으로 구체화 및 체계화하였다.
- GoF의 디자인 패턴은 수많은 디자인 패턴들 중 가장 일반적인 사례에 적용될 수 있는 패턴들을 분류하여 정리함으로써, 지금까지도 소프트웨어 공학이나 현업에서 가장 많이 사용되는 디자인 패턴이다.
- GoF의 디자인 패턴은 유형에 따라 생성 패턴 5개, 구조 패턴 7개, 행위 패턴 11개 총 23개의 패턴으로 구성된다.

21 연결 리스트(Linked List)는 선형 리스트(Linear List)의 한 종류로, 포인터(Pointer)를 이용하여 서로 연결시킨 형태의 자료 구조입니다.

자료 구조의 종류

배열(Array)	동일한 자료형의 데이터들이 같은 크기로 나열되어 순서를 갖고 있는 집합
선형 리스트(Linear List)	일정한 순서에 의해 나열된 자료 구조로, 배열을 이용하는 연속 리스트(Contiguous List)와 포인터를 이용하는 연결 리스트(Linked List)로 구분됨
스택(Stack)	리스트의 한쪽 끝으로만 자료의 삽입, 삭제 작업이 이루어지는 자료 구조
큐(Queue)	리스트의 한쪽에서는 삽입 작업이 이루어지고 다른 한쪽에서는 삭제 작업이 이루어지도록 구성한 자료 구조
트리(Tree)	정점(Node, 노드)과 선분(Branch, 가지)을 이용하여 사이클을 이루지 않도록 구성한 그래프(Graph)의 특수한 형태

22 ①번은 저장된 데이터(Stored Data), ③번은 운영 데이터(Operational Data), ④번은 공용 데이터(Shared Data)에 대한 설명입니다.

23 전위식은 연산자를 해당 피연산자 두 개의 앞(왼쪽)으로 이동시킨 것입니다. 그러므로 연산자와 피연자 2개를 묶은 후 연산자를 피연산자 두 개의 뒤(오른쪽)로 옮겨 놓으면 후위식이 됩니다.

❶ 왼쪽으로 인접한 연산자 1개와 피연산자 2개를 묶습니다.

− / * A + B C D E

↓

(− (/ (* A (+ B C)) D) E)

❷ 연산자를 피연산자 두 개의 뒤로 옮깁니다.

(− (/ (* A (+ B C)) D) E)

↓

(((A (B C +) *) D /) E −)

❸ 괄호를 제거합니다.

A B C + * D / E −

24 인수 테스트는 개발된 소프트웨어를 사용자가 직접 수행하는 테스트입니다.

인수 테스트(Acceptance Test)

- 개발한 소프트웨어가 사용자의 요구사항을 충족하는지에 중점을 두고 테스트하는 방법이다.
- 개발한 소프트웨어를 사용자가 직접 테스트한다.
- 테스트에 문제가 없으면 사용자는 소프트웨어를 인수하게 되고, 프로젝트는 종료된다.
- 인수 테스트의 종류
  - 사용자 인수 테스트 : 사용자가 시스템 사용의 적절성 여부를 확인
  - 운영상의 인수 테스트 : 시스템 관리자가 시스템 인수 시 수행하는 테스트 기법으로, 백업/복원 시스템, 재난 복구, 사용자 관리, 정기 점검 등을 확인
  - 계약 인수 테스트 : 계약상의 인수/검수 조건을 준수하는지 여부를 확인
  - 규정 인수 테스트 : 소프트웨어가 정부 지침, 법규, 규정 등 규정에 맞게 개발되었는지 확인
  - 알파 테스트 : 개발자의 장소에서 사용자가 개발자 앞에서 행하는 테스트 기법
  - 베타 테스트 : 선정된 최종 사용자가 여러 명의 사용자 앞에서 행하는 테스트 기법

25 테스트 시나리오는 시스템별, 모듈별, 항목별 등과 같이 여러 개의 시나리오로 분리하여 작성해야 합니다.

테스트 시나리오 작성 시 유의 사항

- 시스템별, 모듈별, 항목별 등과 같이 여러 개의 시나리오로 분

리하여 작성해야 한다.

- 사용자의 요구사항과 설계 문서 등을 토대로 작성해야 한다.
- 각각의 테스트 항목은 식별자 번호, 순서 번호, 테스트 데이터, 테스트 케이스, 예상 결과, 확인 등을 포함해서 작성해야 한다.
- 유스케이스(Use Case) 간 업무 흐름이 정상적인지를 테스트할 수 있도록 작성해야 한다.
- 개발된 모듈 또는 프로그램 간의 연계가 정상적으로 동작하는지 테스트할 수 있도록 작성해야 한다.

26 결함 관리의 처리 순서는 '기록 → 검토 → 수정 → 재확인'입니다.

결함 관리 프로세스

❶ 결함 관리 계획 : 전체 프로세스에 대한 결함 관리 일정, 인력, 업무 프로세스 등을 확보하여 계획을 수립

❷ 결함 기록 : 발견된 결함을 결함 관리 DB에 등록

❸ 결함 검토 : 등록된 결함을 검토하고 수정할 개발자에게 전달

❹ 결함 수정 : 개발자가 결함 수정

❺ 결함 재확인 : 개발자가 수정한 내용을 확인하고 다시 테스트를 수행

❻ 결함 상태 추적 및 모니터링 활동 : 결함 관리 상태를 한 눈에 확인할 수 있도록 대시보드나 게시판 형태로 제공

❼ 최종 결함 분석 및 보고서 작성 : 발견된 결함에 대한 정보와 이해관계자들의 의견이 반영된 보고서를 작성하고 결함 관리를 종료

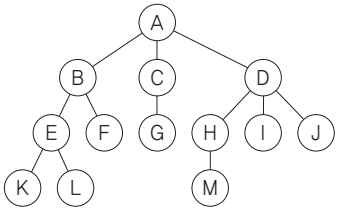
27 애플리케이션이 과부하 상태에서 어떻게 작동하는지 확인하는 테스트는 스트레스 테스트(Stress Test)입니다.

성능 테스트 도구

- 애플리케이션의 성능을 테스트하기 위해 애플리케이션에 부하나 스트레스를 가하면서 애플리케이션의 성능 측정 지표를 점검하는 도구이다.
- 부하(Load) 테스트 : 애플리케이션에 일정 시간 동안 부하를 가하면서 반응을 측정하는 테스트
- 스트레스(Stress) 테스트 : 부하 테스트를 확장한 테스트로, 애플리케이션이 과부하 상태에서 어떻게 작동하는지 테스트
- 종류
  - JMeter : HTTP, FTP 등 다양한 프로토콜을 지원하는 부하 테스트 도구
  - LoadUI : 서버 모니터링, Drag&Drop 등 사용자의 편리성이 강화된 부하 테스트 도구
  - OpenSTA : HTTP, HTTPS 프로토콜에 대한 부하 테스트 및 생산품 모니터링 도구

28 트리(Tree)에서 차수(Degree)는 가장 차수가 많은 노드의 차수입니다. 문제에 주어진 트리(Tree)에서 각 노드의 차수는 A = 2, B = 3, C = 1, D = 1이므로 B노드의 차수인 3이 트리의 차수입니다.

트리 관련 용어



- 노드(Node) : 트리의 기본 요소로서 자료 항목과 다른 항목에 대한 가지(Branch)를 합친 것
  예 A, B, C, D, E, F, G, H, I, J, K, L, M
- 근 노드(Root Node) : 트리의 맨 위에 있는 노드
  예 A
- 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
- 자식 노드(Son Node) : 어떤 노드에 연결된 다음 레벨의 노드들
  예 D의 자식 노드 : H, I, J
- 부모 노드(Parent Node) : 어떤 노드에 연결된 이전 레벨의 노드들
  예 E, F의 부모 노드 : B
- 형제 노드(Brother Node, Sibling) : 동일한 부모를 갖는 노드들
  예 H의 형제 노드 : I, J
- 트리의 디그리 : 노드들의 디그리 중에서 가장 많은 수
  예 노드 A나 D가 3개의 디그리를 가지므로 앞 트리의 디그리는 3이다.

29 애플리케이션의 성능 관리를 위해 접속자, 자원 현황, 트랜잭션 수행 내역, 장애 진단 등 다양한 모니터링 기능을 제공하는 소프트웨어는 APM(Application Performance Management /Monitoring)입니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- ESB(Enterprise Service Bus) : 애플리케이션 간 연계, 데이터 변환, 웹 서비스 지원 등 표준 기반의 인터페이스를 제공하는 솔루션
- EAI(Enterprise Application Integration) : 기업 내 각종 애플리케이션 및 플랫폼 간의 정보 전달, 연계, 통합 등 상호 연동이 가능하게 해주는 솔루션
- 옵티마이저(Optimizer) : SQL문을 수행할 가장 효율적이고 빠른 방법을 선택해주는 DBMS의 핵심 엔진

30 SQL은 관계형 데이터베이스(RDB)를 지원하는 언어로, 데이터 정의어(DDL), 데이터 조작어(DML), 데이터 제어어(DCL)로 구분되며, 데이터의 보안, 무결성, 회복, 병행 수행 제어 등을 정의하는 데 사용되는 SQL 언어는 데이터 제어어(DCL)입니다.

SQL의 종류

- 데이터 정의어(DDL) : SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의하거나 변경 또는 삭제할 때 사용하는 언어
- 데이터 조작어(DML) : 데이터베이스 사용자가 응용 프로그램이나 질의어를 통하여 저장된 데이터를 실질적으로 처리하는 데 사용되는 언어
- 데이터 제어어(DCL) : 데이터의 보안, 무결성, 회복, 병행 수행 제어 등을 정의하는 데 사용되는 언어

31 클리어링 하우스(Clearing House)는 디지털 저작권 라이선스의 중개 및 발급을 수행하는 곳으로, 디지털 저작물의 이용 내역을 근거로 저작권료의 정산 및 분배가 수행됩니다. 'Clearing'에는 '결제', '청산'이라는 의미도 있으므로 클리어링 하우스란 '결제가 이루어지는 곳'으로 해석할 수도 있습니다.

디지털 저작권 관리(DRM) 관련 용어

- 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
- 콘텐츠 소비자(Customer) : 콘텐츠를 구매해서 사용하는 주체
- DRM 컨트롤러(DRM Controller) : 배포된 콘텐츠의 이용 권한을 통제하는 프로그램
- 보안 컨테이너(Security Container) : 콘텐츠 원본을 안전하게 유통하기 위한 전자적 보안 장치

32 ③번의 내용은 공유 폴더 방식에 대한 설명입니다.

소프트웨어 버전 관리 도구

- 공유 폴더 방식
  - 버전 관리 자료가 로컬 컴퓨터의 공유 폴더에 저장되어 관리되는 방식이다.
  - 종류 : SCCS, RCS, PVCS, QVCS 등
- 클라이언트/서버 방식
  - 버전 관리 자료가 중앙 시스템(서버)에 저장되어 관리되는 방식이다.
  - 종류 : CVS, SVN(Subversion), CVSNT, Clear Case, CMVC, Perforce 등
- 분산 저장소 방식
  - 버전 관리 자료가 하나의 원격 저장소와 분산된 개발자 PC의 로컬 저장소에 함께 저장되어 관리되는 방식이다.
  - 종류 : Git, GNU arch, DCVS, Bazaar, Mercurial, TeamWare, Bitkeeper, Plastic SCM 등

33 ③번의 내용은 화이트박스(White Box) 테스트에 대한 설명입니다.

화이트박스 테스트(White Box Test)

- 모듈의 원시 코드를 오픈시킨 상태에서 원시 코드의 논리적인 모든 경로를 테스트하여 테스트 케이스를 설계하는 방법이다.
- 설계된 절차에 초점을 둔 구조적 테스트로 프로시저 설계의 제어 구조를 사용하여 테스트 케이스를 설계하며, 테스트 과정의 초기에 적용된다.
- 모듈 안의 작동을 직접 관찰한다.
- 원시 코드(모듈)의 모든 문장을 한 번 이상 실행함으로써 수행된다.
- 프로그램의 제어 구조에 따라 선택, 반복 등의 분기점 부분들을 수행함으로써 논리적 경로를 제어한다.
- 종류 : 기초 경로 검사, 제어 구조 검사 등

블랙박스 테스트(Black Box Test)

- 소프트웨어가 수행할 특정 기능을 알기 위해서 각 기능이 완전히 작동되는 것을 입증하는 테스트로, 기능 테스트라고도 한다.
- 사용자의 요구사항 명세를 보면서 테스트하는 것으로, 주로 구현된 기능을 테스트한다.
- 소프트웨어 인터페이스에서 실시되는 테스트이다.
- 부정확하거나 누락된 기능, 인터페이스 오류, 자료 구조나 외부 데이터베이스 접근에 따른 오류, 행위나 성능 오류, 초기화와 종료 오류 등을 발견하기 위해 사용되며, 테스트 과정의 후반부에 적용된다.
- 종류 : 동치 분할 검사, 경계값 분석, 원인-효과 그래프 검사, 오류 예측 검사, 비교 검사 등

34 테스트 케이스 설계자가 절차적 설계의 논리적 복잡성을 측정할 수 있게 해주고, 결과가 실행 경로의 기초를 정의하는 지침으로 사용되는 테스트는 기초 경로 검사입니다.

화이트 박스 테스트의 종류

- 기초 경로 검사
  - 대표적인 화이트박스 테스트 기법으로, 테스트 케이스 설계자가 절차적 설계의 논리적 복잡성을 측정할 수 있게 해준다.
  - 테스트 측정 결과는 실행 경로의 기초를 정의하는 데 지침으로 사용된다.
- 제어 구조 검사

- 조건 검사(Condition Testing) : 프로그램 모듈 내에 있는 논리적 조건을 테스트하는 테스트 케이스 설계 기법
- 루프 검사(Loop Testing) : 프로그램의 반복(Loop) 구조에 초점을 맞춰 실시하는 테스트 케이스 설계 기법
- 데이터 흐름 검사(Data Flow Testing) : 프로그램에서 변수의 정의와 변수 사용의 위치에 초점을 맞춰 실시하는 테스트 케이스 설계 기법

블랙 박스 테스트의 종류

- 동치 분할 검사(Equivalence Partitioning Testing) : 입력 자료에 초점을 맞춰 테스트 케이스를 만들고 검사하는 방법으로 동등 분할 기법이라고도 함
- 경계값 분석(Boundary Value Analysis) : 입력 자료에만 치중한 동치 분할 기법을 보완하기 위한 기법
- 원인-효과 그래프 검사(Cause-Effect Graphing Testing) : 입력 데이터 간의 관계와 출력에 영향을 미치는 상황을 체계적으로 분석한 다음 효용성이 높은 테스트 케이스를 선정하여 검사하는 기법
- 오류 예측 검사(Error Guessing) : 과거의 경험이나 확인자의 감각으로 테스트하는 기법
- 비교 검사(Comparison Testing) : 여러 버전의 프로그램에 동일한 테스트 자료를 제공하여 동일한 결과가 출력되는지 테스트하는 기법

35 하향식 통합 테스트를 수행할 때 주요 제어 모듈의 종속 모듈들은 스텁(Stub)으로 대체합니다.

하향식 통합 테스트(Top Down Integration Test)

- 프로그램의 상위 모듈에서 하위 모듈 방향으로 통합하면서 테스트하는 기법이다.
- 주요 제어 모듈을 기준으로 하여 아래 단계로 이동하면서 통합하는데, 이때 깊이 우선 통합법이나 넓이 우선 통합법을 사용한다.
- 테스트 초기부터 사용자에게 시스템 구조를 보여줄 수 있다.
- 상위 모듈에서는 테스트 케이스를 사용하기 어렵다.
- 하향식 통합 테스트 수행 절차
  ❶ 주요 제어 모듈은 작성된 프로그램을 사용하고, 주요 제어 모듈의 종속 모듈들은 스텁(Stub)으로 대체한다.
  ❷ 깊이 우선 또는 넓이 우선 등의 통합 방식에 따라 하위 모듈인 스텁들이 한 번에 하나씩 실제 모듈로 교체한다.
  ❸ 모듈이 통합될 때마다 테스트를 실시한다.
  ❹ 새로운 오류가 발생하지 않음을 보증하기 위해 회귀 테스트를 실시한다.

36 모듈 연계 방법에는 EAI와 ESB 두 가지가 있으며, 애플리케이션 중심의 모듈 연계 방법은 EAI, 서비스 중심의 모듈 연계 방법은 ESB입니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- IDE(Integrated Development Environment) : 개발에 필요한 환경, 즉 편집기(Editor), 컴파일러(Compiler), 디버거(Debugger) 등의 다양한 툴을 하나의 인터페이스로 통합하여 제공하는 것
- 미들웨어(Middleware) : 미들(Middle)과 소프트웨어(Software)의 합성어로, 운영체제와 해당 운영체제에서 실행되는 응용 프로그램 사이에서 운영체제가 제공하는 서비스 이외에 추가적인 서비스를 제공하는 소프트웨어

37 소프트웨어 패키징은 사용자를 중심으로 작업이 진행되므로 사용자의 편의성 및 실행 환경을 우선적으로 고려해야 합니다.

소프트웨어 패키징

- 모듈별로 생성한 실행 파일들을 묶어 배포용 설치 파일을 만드는 것을 말한다.
- 개발자가 아니라 사용자를 중심으로 진행한다.
- 소스 코드는 향후 관리를 고려하여 모듈화하여 패키징한다.
- 사용자가 소프트웨어를 사용하게 될 환경을 이해하여, 다양한 환경에서 소프트웨어를 손쉽게 사용할 수 있도록 일반적인 배포 형태로 패키징한다.
- 사용자를 중심으로 진행되는 작업이므로 사용자의 편의성 및 실행 환경을 우선적으로 고려해야 한다.
- 사용자의 시스템 환경, 즉 운영체제(OS), CPU, 메모리 등에 필요한 최소 환경을 정의한다.
- UI(User Interface)는 사용자가 눈으로 직접 확인할 수 있도록 시각적인 자료와 함께 제공하고 매뉴얼과 일치시켜 패키징한다.
- 소프트웨어는 단순히 패키징하여 배포하는 것으로 끝나는 것이 아니라 하드웨어와 함께 관리될 수 있도록 Managed Service 형태로 제공하는 것이 좋다.
- 고객의 편의성을 고려한 안정적인 배포가 중요하다.
- 다양한 사용자의 요구사항을 반영할 수 있도록 패키징의 변경 및 개선에 대한 관리를 항상 고려한다.

38 살충제 패러독스(Pesticide Paradox)는 살충제를 지속적으로 뿌리면 벌레가 내성이 생겨서 죽지 않는 현상을 의미하는 것으로, 애플리케이션 테스트에서는 동일한 테스트를 반복하면 더 이상 결함이 발견되지 않는 현상을 가리킵니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- 결함 집중(Defect Clustering) : 대부분의 결함이 소수의 특정 모듈에 집중해서 발생하는 현상
- 파레토 법칙(Pareto Principle) : 테스트로 발견된 80%의 오류는 20%의 모듈에서 발견된다는 법칙
- 오류-부재의 궤변(Absence of Errors Fallacy) : 결함을 모두 제거

해도 사용자의 요구사항을 만족시키지 못하면 해당 소프트웨어는 품질이 높다고 말할 수 없음

39 IPSec, SSL, S-HTTP는 네트워크 영역에 보안 기능을 적용하기 위해 사용하는 암호화 방식입니다. 데이터베이스 영역에서 사용되는 암호화 방식에는 DES, RSA 등이 있습니다.

인터페이스 보안 기능

- 네트워크 영역
  - 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 등의 다양한 방식으로 적용한다.
- 애플리케이션 영역 : 소프트웨어 개발 보안 가이드를 참조하여 애플리케이션 코드 상의 보안 취약점을 보완하는 방향으로 애플리케이션 보안 기능을 적용함
- 데이터베이스 영역
  - 데이터베이스, 스키마, 엔티티의 접근 권한과 프로시저(Procedure), 트리거(Trigger) 등 데이터베이스 동작 객체의 보안 취약점에 보안 기능을 적용한다.
  - 개인 정보나 업무상 민감한 데이터의 경우 암호화나 익명화 등 데이터 자체의 보안 방안도 고려한다.

40 통합 개발 환경(IDE; Integrated Development Environment)은 개발에 필요한 환경, 즉 편집기(Editor), 컴파일러(Compiler), 디버거(Debugger) 등의 다양한 툴을 하나의 인터페이스로 통합하여 제공하는 것을 의미합니다.

통합 개발 환경(IDE)

- 개발에 필요한 환경, 즉 편집기(Editor), 컴파일러(Compiler), 디버거(Debugger) 등의 다양한 툴을 하나의 인터페이스로 통합하여 제공하는 것을 의미한다.
- 통합 개발 환경 도구는 통합 개발 환경을 제공하는 소프트웨어를 의미한다.
- 통합 개발 환경 도구는 코드의 자동 생성 및 컴파일이 가능하고 추가 기능을 위한 도구들을 다운로드하여 추가할 수 있다.
- 통합 개발 환경 도구는 코드를 실행하거나 테스트할 때 오류가 발생한 부분을 시각화하므로 수정이 용이하다.
- 통합 개발 환경 도구는 외부의 다양한 서비스와 연동하여 개발에 편의를 제공하고 필요한 정보를 공유할 수 있다.
- 통합 개발 환경을 지원하는 도구는 플랫폼, 운영체제, 언어별로 다양하게 존재한다.
- 통합 개발 환경 지원 도구 : 이클립스(Eclipse), 비주얼 스튜디오(Visual Studio), 엑스 코드(Scode), IDEA 등

41 트랜잭션의 인터페이스와 테이블을 설계하는 단계는 데이터 모델링(논리적 설계) 단계입니다. 데이터 구조화(물리적 설계) 단계에서는 논리적 구조로 표현된 데이터를 디스크 등의 물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 작업을 수행합니다.

논리적 설계(데이터 모델링)

- 현실 세계에서 발생하는 자료를 컴퓨터가 이해하고 처리할 수 있는 물리적 저장장치에 저장할 수 있도록 변환하기 위해 특정 DBMS가 지원하는 논리적 자료 구조로 변환시키는 과정이다.
- 개념 세계의 데이터를 필드로 기술된 데이터 타입과 이 데이터 타입들 간의 관계로 표현되는 논리적 구조의 데이터로 모델화한다.
- 개념적 설계가 개념 스키마를 설계하는 단계라면 논리적 설계에서는 개념 스키마를 평가 및 정제하고 DBMS에 따라 서로 다른 논리적 스키마를 설계하는 단계이다.
- 트랜잭션의 인터페이스를 설계한다.
- 관계형 데이터베이스라면 테이블을 설계하는 단계이다.

물리적 설계(데이터 구조화)

- 논리적 설계 단계에서 논리적 구조로 표현된 데이터를 디스크 등의 물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 과정이다.
- 물리적 설계 단계에서는 다양한 데이터베이스 응용에 대해 처리 성능을 얻기 위해 데이터베이스 파일의 저장 구조 및 액세스 경로를 결정한다.
- 저장 레코드의 형식, 순서, 접근 경로와 같은 정보를 사용하여 데이터가 컴퓨터에 저장되는 방법을 묘사한다.

42 개체 어커런스(Occurrence)는 개체 인스턴스를 부르는 다른 말입니다.

개체의 구성 요소

- 속성 : 개체가 가지고 있는 특성
- 개체 타입 : 속성으로만 기술된 개체의 정의
- 개체 인스턴스 : 개체를 구성하고 있는 각 속성들이 값을 가져 하나의 개체를 나타내는 것으로 개체 어커런스(Occurrence)라고도 함
- 개체 세트 : 개체 인스턴스의 집합

43 하나의 릴레이션에는 동일한 튜플이 존재할 수 없습니다.

릴레이션의 특징

- 한 릴레이션에는 똑같은 튜플이 포함될 수 없으므로 릴레이션에 포함된 튜플들은 모두 상이하다.
- 한 릴레이션에 포함된 튜플 사이에는 순서가 없다.
- 튜플들의 삽입, 삭제 등의 작업으로 인해 릴레이션은 시간에 따라 변한다.

- 릴레이션 스키마를 구성하는 속성들 간의 순서는 중요하지 않다.
- 속성의 유일한 식별을 위해 속성의 명칭은 유일해야 하지만, 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.

44 외래키(Foreign Key)는 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미합니다.

키(Key)의 종류

- 후보키(Candidate Key) : 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용하는 속성들의 부분집합, 즉 기본키로 사용할 수 있는 속성들
- 기본키(Primary Key) : 후보키 중에서 특별히 선정된 주키(Main Key)로, 한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성
- 대체키(Alternate Key) : 대체키는 후보키가 둘 이상일 때 기본키를 제외한 나머지 후보키
- 슈퍼키(Super Key) : 한 릴레이션 내에 있는 속성들의 집합으로 구성된 키로서 릴레이션을 구성하는 모든 튜플들 중 슈퍼키로 구성된 속성의 집합과 동일한 값은 나타나지 않음
- 외래키(Foreign Key) : 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미하며, 릴레이션 간에 참조 관계를 표현하는데 중요한 도구

45 공통 속성을 중심으로 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 연산은 Join으로, 기호는 ⋈를 사용합니다.

Join

- 공통 속성을 중심으로 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 연산이다.
- Join의 결과로 만들어진 릴레이션의 차수는 조인된 두 릴레이션의 차수를 합한 것과 같다.
- Join의 결과는 Cartesian Product(교차곱)를 수행한 다음 Select를 수행한 것과 같다.
- 연산자의 기호는 ⋈를 사용한다.
- 표기 형식
  - R ⋈ 키속성r = 키속성s S
  - 키 속성 r은 릴레이션 R의 속성이고, 키 속성 s는 릴레이션 S의 속성이다.

46 BCNF는 결정자가 모두 후보키인 정규형으로, 제 3정규형에서 모든 결정자가 후보키가 되도록 분해하면 BCNF가 됩니다.

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

정규화 단계 암기 요령
정규화가 두부가게에 가서 모든 두부를 달라고 하니 주인이 깜짝 놀라 말했다.
두부이걸다줘? ≒ 도부이결다조
도메인이 원자값
부분적 함수 종속 제거
이행적 함수 종속 제거
결정자이면서 후보키가 아닌 것 제거
다치 종속 제거
조인 종속성 이용

47 시스템의 성능 향상, 개발 및 운영의 편의성 등의 목적으로 정규화된 데이터 모델을 통합, 중복, 분리하는 과정을 반정규화(Denormalization)라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- 정규화(Normalization) : 함수적 종속성 등의 종속성 이론을 이용하여 잘못 설계된 관계형 스키마를 더 작은 속성의 세트로 쪼개어 바람직한 스키마로 만들어 가는 과정
- 조인(Join) : 2개의 테이블에 대해 연관된 튜플들을 결합하여, 하나의 새로운 릴레이션을 반환하는 것
- 분할(Partitioning) : 반정규화를 위해 테이블을 수직 또는 수평으로 분할하는 것

48 인덱스(Index)는 〈키 값과 포인터〉 쌍으로 구성되는 데이터 구조로, 인덱스 설계 시에는 인덱스가 가리키는 데이터와는 별개로 추가적인 저장 공간이 필요하다는 것을 염두에 두어야 합니다.

인덱스 설계 시 고려사항

- 새로 추가되는 인덱스는 기존 액세스 경로에 영향을 미칠 수 있다.
- 인덱스를 지나치게 많이 만들면 오버헤드가 발생한다.
- 넓은 범위를 인덱스로 처리하면 많은 오버헤드가 발생한다.
- 인덱스를 만들면 추가적인 저장 공간이 필요하다.

- 인덱스와 테이블 데이터의 저장 공간이 분리되도록 설계한다.

49 사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블을 뷰(View)라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- 인덱스(Index) : 데이터 레코드를 빠르게 접근하기 위해 〈키 값, 포인터〉 쌍으로 구성되는 데이터 구조
- 클러스터(Cluster) : 데이터 저장 시 데이터 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법
- 파티션(Partition) : 대용량의 테이블이나 인덱스를 작은 논리적 단위인 파티션으로 나누는 것

50 클러스터(Cluster)는 데이터 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법입니다.

클러스터(Cluster)

- 데이터 저장 시 데이터 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법이다.
- 클러스터링키로 지정된 컬럼 값의 순서대로 저장되고, 여러 개의 테이블이 하나의 클러스터에 저장된다.
- 클러스터링 된 테이블은 데이터 조회 속도는 향상시키지만 데이터 입력, 수정, 삭제에 대한 성능은 저하시킨다.
- 클러스터는 데이터의 분포도가 넓을수록 유리하다.
- 데이터 분포도가 넓은 테이블을 클러스터링 하면 저장 공간을 절약할 수 있다.
- 클러스터링된 테이블은 클러스터링키 열을 공유하므로 저장 공간이 줄어든다.
- 대용량을 처리하는 트랜잭션은 전체 테이블을 스캔하는 일이 자주 발생하므로 클러스터링을 하지 않는 것이 좋다.
- 처리 범위가 넓은 경우에는 단일 테이블 클러스터링을, 조인이 많이 발생하는 경우에는 다중 테이블 클러스터링을 사용한다.
- 파티셔닝된 테이블에는 클러스터링을 할 수 없다.
- 클러스터링을 하면 비슷한 데이터가 동일한 데이터 블록에 저장되기 때문에 디스크 I/O가 줄어든다.
- 클러스터링된 테이블에 클러스터드 인덱스를 생성하면 접근 성능이 향상된다.

51 범위 분할로 분할한 다음 해시 함수를 적용하여 다시 분할하는 방식을 조합 분할(Composite Partitioning)이라고 합니다.

파티션의 종류

- 범위 분할(Range Partitioning)
  - 지정한 열의 값을 기준으로 분할한다.
  - 예 일별, 월별, 분기별 등
- 해시 분할(Hash Partitioning)
  - 해시 함수를 적용한 결과 값에 따라 데이터를 분할한다.
  - 특정 파티션에 데이터가 집중되는 범위 분할의 단점을 보완한 것으로, 데이터를 고르게 분산할 때 유용하다.
  - 특정 데이터가 어디에 있는지 판단할 수 없다.
  - 고객번호, 주민번호 등과 같이 데이터가 고른 컬럼에 효과적이다.
- 조합 분할(Composite Partitioning)
  - 범위 분할로 분할한 다음 해시 함수를 적용하여 다시 분할하는 방식이다.
  - 범위 분할한 파티션이 너무 커서 관리가 어려울 때 유용하다.

52 분산 데이터베이스를 이용할 경우 DBMS가 수행할 기능이 복잡하고, 소프트웨어 개발 비용 및 처리 비용이 증가한다는 단점이 있습니다.

분산 데이터베이스의 장 · 단점

장점	• 지역 자치성이 높음 • 자료의 공유성이 향상됨 • 분산 제어가 가능함 • 시스템 성능이 향상됨 • 중앙 컴퓨터의 장애가 전체 시스템에 영향을 끼치지 않음 • 효용성과 융통성이 높음 • 신뢰성 및 가용성이 높음 • 점진적 시스템 용량 확장이 용이함
단점	• DBMS가 수행할 기능이 복잡함 • 데이터베이스 설계가 어려움 • 소프트웨어 개발 비용이 증가함 • 처리 비용이 증가함 • 잠재적 오류가 증가함

53 '가입방식' 속성을 기준으로 그룹화하고 내림차순으로 정렬한 한 후, 그룹별로 튜플의 개수를 구하는 COUNT 함수의 결과를 출력합니다.

- SELECT COUNT(*) : 튜플의 개수를 표시한다.
- FROM 회원 : 〈회원〉 테이블을 대상으로 검색한다.
- GROUP BY 가입방식 : '가입방식'을 기준으로 그룹을 지정한다.
- ORDER BY 가입방식 DESC; : '가입방식' 속성의 값을 기준으로 내림차순 정렬한다.

54 GRANT문의 기본 형식은 'GRANT 권한_리스트 ON 개체 TO 사용자 [WITH GRANT OPTION];'이지만, 부여받을 권한을 다른 사용자에게 다시 부여할 수 있는 권한에 대한 언급이 없으므로 '[WITH GRANT OPTION]'을 생략하고 작성하면 됩니다.

- GRANT SELECT : 조회(SELECT) 권한을 부여한다.
- ON EMPLIST : 〈EMPLIST〉 테이블에 대한 권한을 부여한다.
- TO guest03; : 'guest03'이라는 사용자에게 부여한다.

55 참조 대상이 되는 튜플이 삭제될 때 취해야 할 사항을 지정하는

옵션이 'ON DELETE SET NULL'이므로, 〈제조처〉 테이블에서 튜플이 삭제되면 관련된 모든 튜플의 '제조사' 속성의 값을 NULL로 변경합니다.

```
❶ CREATE TABLE 상품(
❷    상품명 VARCHAR(10) PRIMARY KEY,
❸    단가 INT,
❹    재고량 INT,
❺    제조사 VARCHAR(20),
❻    FOREIGN KEY(제조사) REFERENCES 제조처(이름)
❼        ON DELETE SET NULL);
```

❶ 〈상품〉 테이블을 생성한다.
❷ '상품명' 속성은 최대 문자 10자로, 기본키로 정의한다.
❸ '단가' 속성은 정수이다.
❹ '재고량' 속성은 정수이다.
❺ '제조사' 속성은 최대 문자 20자이다.
❻ '제조사' 속성은 〈제조처〉 테이블의 '이름' 속성을 참조하는 외래키이다.
❼ 〈제조처〉 테이블에서 튜플이 삭제되면 관련된 모든 튜플의 '제조사' 속성의 값을 NULL로 변경한다.

56 RETURN 명령어가 반드시 사용되어야 하는 절차형 SQL은 사용자 정의 함수입니다.

트리거(Trigger)

- 트리거는 데이터베이스 시스템에서 데이터의 삽입(Insert), 갱신(Update), 삭제(Delete) 등의 이벤트(Event)가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL이다.
- 트리거는 데이터베이스에 저장되며, 데이터 변경 및 무결성 유지, 로그 메시지 출력 등의 목적으로 사용된다.
- 트리거의 구문에는 DCL(데이터 제어어)을 사용할 수 없으며, DCL이 포함된 프로시저나 함수를 호출하는 경우에도 오류가 발생한다.
- 트리거에 오류가 있는 경우 트리거가 처리하는 데이터에도 영향을 미치므로 트리거를 생성할 때 세심한 주의가 필요하다.

57 문제의 그림에서 '방문객'과 '상담사'의 '상담' 관계는 다 대 일의 관계로, 한 사람의 상담사가 다수의 방문객을 상담할 수 있음을 나타냅니다.

피터 첸 표기법

사각형( )	개체(Entity) 타입
마름모( )	관계(Relationship) 타입
타원( )	속성(Attribute)
이중 타원( )	다중값 속성(복합 속성)
밑줄 타원( )	기본키 속성
복수 타원( )	복합 속성
관계(N M)	1:1, 1:N, N:M 등의 개체 간 관계에 대한 대응수를 선 위에 기술함
선, 링크( )	개체 타입과 속성을 연결

58 Java 언어로 다양한 종류의 데이터베이스에 접속하고 SQL문을 수행할 때 사용되는 표준 API는 JDBC(Java DataBase Connectivity)입니다. 나머지 보기로 제시된 DBMS 접속 기술들의 개념도 확인하고 넘어가세요.

- ODBC(Open DataBase Connectivity) : 데이터베이스에 접근하기 위한 표준 개방형 API로, 개발 언어에 관계없이 사용할 수 있음
- MyBatis : JDBC 코드를 단순화하여 사용할 수 있는 SQL Mapping 기반 오픈 소스 접속 프레임워크
- ORM(Object-Relational Mapping) : 객체지향 프로그래밍의 객체(Object)와 관계형 데이터베이스(Relational Database)의 데이터를 연결(Mapping)하는 기술

59 쿼리문의 성능을 최적화하기 위해 IN보다 EXISTS를 사용하는 것이 좋습니다.

쿼리 성능 최적화

- 실행 계획에 표시된 연산 순서, 조인 방식, 테이블 조회 방법 등을 참고하여 SQL문이 더 빠르고 효율적으로 작동하도록 SQL 코드와 인덱스를 재구성하는 것을 의미한다.
- SQL 코드 재구성
  - WHERE 절을 추가하여 일부 레코드만 조회하게 함으로써 조회에 들어가는 비용을 줄인다.
  - WHERE 절에 연산자가 포함되면 INDEX를 활용하지 못하므로 가능한 한 연산자 사용을 자제한다.
  - 서브 쿼리에 특정 데이터가 존재하는지 확인할 때는 IN보다 EXISTS를 활용한다.
  - 옵티마이저의 실행 계획이 잘못되었다고 판단되는 경우 힌트를 활용하여 실행 계획의 액세스 경로 및 조인 순서를 변경한다.
- 인덱스 재구성
  - SQL 코드에서 조회되는 속성과 조건들을 고려하여 인덱스를 구성한다.
  - 실행 계획을 참고하여 인덱스를 추가하거나 기존 인덱스의 열 순서를 변경한다.
  - 인덱스의 추가 및 변경은 해당 테이블을 참조하는 다른 SQL문에도 영향을 줄 수 있으므로 신중히 결정한다.
  - 단일 인덱스로 쓰기나 수정 없이 읽기로만 사용되는 테이블의 경우 IOT(Index-Organized Table)로 구성하는 것을 고려한다.
  - 불필요한 인덱스를 제거한다.

60 데이터 전환 계획서의 항목 중 비상 계획은 데이터 전환 당일에 구성하게 될 종합상황실에 대한 계획을 수립하고 데이터 전환 시 종합상황실과 전환 관리 그룹과의 실시간 정보 공유 방법에 대한 계획을 수립하는 것을 의미합니다. ④번의 내용은 데이터 복구 대책에 대한 설명입니다.

데이터 전환

- 운영 중인 기존 정보 시스템에 축적되어 있는 데이터를 추출(Extraction)하여 새로 개발할 정보 시스템에서 운영 가능하도록 변환(Transformation)한 후, 적재(Loading)하는 일련의 과정을 말한다.
- 데이터 전환을 ETL(Extraction, Transformation, Load), 즉 추출, 변환, 적재 과정이라고 한다.
- 데이터 전환을 데이터 이행(Data Migration) 또는 데이터 이관이라고도 한다.
- 데이터 전환을 성공적으로 수행하기 위해 데이터 전환이 필요한 대상을 분석하여 데이터 전환 작업에 필요한 모든 계획을 기록하는 데이터 전환 계획서를 작성하고 이를 토대로 진행한다.
- 데이터 전환 계획서의 주요 항목
  - 데이터 전환 개요 : 데이터 전환 목표, 주요 성공 요인, 전제조건 및 제약조건
  - 데이터 전환 대상 및 범위
  - 데이터 전환 환경 구성 : 원천 시스템 구성도, 목적 시스템 구성도, 전환 단계별 DISK 사용량
  - 데이터 전환 조직 및 역할 : 데이터 전환 조직도, 조직별 역할
  - 데이터 전환 일정
  - 데이터 전환 방안 : 데이터 전환 규칙, 데이터 전환 절차, 데이터 전환 방법, 데이터 전환 설계 등
  - 데이터 정비 방안 : 데이터 정비 대상 및 방법, 데이터 정비 일정 및 조직
  - 비상 계획 : 종합상황실 및 의사소통 체계
  - 데이터 복구 대책 : 백업 및 복구 방안

61 네트워크 트래픽의 포화를 방지하기 위해 응답 속도를 제한하는 기능은 대역폭 제한(Bandwidth Throttling)입니다.

웹 서버(Web Server)의 기능

- HTTP/HTTPS : 지원 브라우저로부터 요청을 받아 응답할 때 사용되는 프로토콜
- 통신 기록(Communication Log) : 처리한 요청들을 로그 파일로 기록하는 기능
- 정적 파일 관리(Managing Static Files) : HTML, CSS, 이미지 등의 정적 파일들을 저장하고 관리하는 기능
- 대역폭 제한(Bandwidth Throttling) : 네트워크 트래픽의 포화를 방지하기 위해 응답 속도를 제한하는 기능
- 가상 호스팅(Virtual Hosting) : 하나의 서버로 여러 개의 도메인 이름을 연결하는 기능
- 인증(Authentication) : 사용자가 합법적인 사용자인지를 확인하는 기능

62 프레임워크는 추상화가 아니라 캡슐화를 통해 모듈화를 강화하고 설계 및 구현의 변경에 따른 영향을 최소화함으로서 소프트웨어의 품질을 향상시킵니다.

프레임워크의 특성

- 모듈화(Modularity) : 캡슐화를 통해 모듈화를 강화하고 설계 및 구현의 변경에 따른 영향을 최소화함으로서 소프트웨어의 품질을 향상시킴
- 재사용성(Reusability) : 재사용 가능한 모듈들을 제공함으로써 개발자의 생산성을 향상시킴
- 확장성(Extensibility) : 다형성(Polymorphism)을 통한 인터페이스 확장이 가능하여 다양한 형태와 기능을 가진 애플리케이션 개발이 가능함
- 제어의 역흐름(Inversion of Control) : 개발자가 관리하고 통제해야 하는 객체들의 제어를 프레임워크에 넘김으로써 생산성을 향상시킴

63 boolean을 기본 자료형으로 갖고 있는 프로그래밍 언어는 JAVA입니다. C 언어에서 사용하기 위해서는 라이브러리 파일을 불러와야 합니다.

C/JAVA의 기본 자료형

- C 언어
  - 문자 : char
  - 정수 : short, int, long, long long
  - 실수 : float, double, long double
- JAVA
  - 문자 : char
  - 정수 : byte, short, int, long
  - 실수 : float double
  - 논리 : boolean

64 메모리가 아닌 CPU 내부의 레지스터에 기억영역을 할당받는 변수는 레지스터 변수입니다.

기억 클래스의 종류

- 자동 변수(Automatic Variable) : 함수나 코드의 범위를 한정하는 블록 내에서 선언되는 변수
- 외부 변수(External Variable) : 현재 파일이나 다른 파일에서 선언된 변수나 함수를 참조(Reference)하기 위한 변수
- 정적 변수(Static Variable) : 함수나 블록 내에서 선언하는 내부 정적 변수와 함수 외부에서 선언하는 외부 정적 변수
- 레지스터 변수(Register Variable) : 메모리가 아닌 CPU 내부의 레지스터에 기억영역을 할당받는 변수

65 사용된 코드의 의미는 다음과 같습니다.

```
#include <stdio.h>
main( ) {
❶ int a = 3, b = 5;
❷ int *c = &a;
❸ *c += ++b;
❹ printf("%d", a + b + *c);
}
```

❶ 정수형 변수 a와 b를 선언하고, 각각 3과 5로 초기화한다.
❷ 포인터 변수 c를 선언하고, 정수형 변수 a의 주소로 초기화한다.
❸ c가 가리키는 곳에 증가한 b의 값을 누적한다. c가 가리키는 곳이 a이므로 a의 값이 변경된다.
*c += ++b → *c = *c + ++b → *c = 3+6
❹ 정수형 변수 a, b와 포인터 변수 c가 가리키는 곳의 값을 합하여 출력한다. 9+6+9 = 24

66 scanf( ) 함수에서 5개의 문자를 입력받기 위해서는 문자를 입력받는 서식 지정자 '%s' 또는 '%c'를 사용하여 '%5s' 또는 '%5c'로 작성해야 합니다. 또한 배열명 앞에는 '&'를 붙이지 않아야 합니다.

67 코드에 사용된 Random 클래스를 포함하고 있는 패키지는 java.util입니다.

```
❶ import java.util.*;
  public class Test {
      public static void main(String[ ] args) {
❷         Random rand = new Random( );
❸         int a[ ] = new int[6];
❹         for(int i = 0; i < 6; i++)
❺             a[i] = rand.nextInt(10);
  }
}
```

❶ Random 클래스가 정의되어 있는 라이브러리이다.
❷ Random 클래스의 객체 변수 rand를 선언한다.
❸ 6개의 요소를 갖는 정수형 배열 a를 선언한다.
❹ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 6보다 작은 동안 ❺번을 반복 수행한다.
❺ 객체 변수 rand의 메소드 nextInt( )를 수행하여 0~9 사이의 정수 난수를 a[i]에 저장한다.

68 반복문에서 이후의 문장을 실행하지 않고 제어를 반복문의 처음으로 옮기는 명령어(예약어)는 continue입니다. break는 반복문의 처음이 아닌 반복문을 벗어나는 명령어이며, next와 release는 존재하지 않는 명령어입니다.

69 ③번의 내용은 MOM(Message Oriented Middleware)에 대한 설명입니다.

운영체제의 기능
- 프로세서(처리기, Processor), 기억장치(주기억장치, 보조기억장치), 입 · 출력장치, 파일 및 정보 등의 자원을 관리한다.
- 자원을 효율적으로 관리하기 위해 자원의 스케줄링 기능을 제공한다.
- 사용자와 시스템 간의 편리한 인터페이스를 제공한다.
- 시스템의 각종 하드웨어와 네트워크를 관리 · 제어한다.
- 데이터를 관리하고, 데이터 및 자원의 공유 기능을 제공한다.
- 시스템의 오류를 검사하고 복구한다.
- 자원 보호 기능을 제공한다.
- 입 · 출력에 대한 보조 기능을 제공한다.
- 가상 계산기 기능을 제공한다.

70 사용자와 시스템 간의 인터페이스를 제공하는 것은 쉘(Shell)입니다.

UNIX 시스템의 구성

커널(Kernel)
- UNIX의 가장 핵심적인 부분이다.
- 컴퓨터가 부팅될 때 주기억장치에 적재된 후 상주하면서 실행된다.
- 하드웨어를 보호하고, 프로그램과 하드웨어 간의 인터페이스 역할을 담당한다.
- 프로세스(CPU 스케줄링) 관리, 기억장치 관리, 파일 관리, 입 · 출력 관리, 프로세스간 통신, 데이터 전송 및 변환 등 여러 가지 기능을 수행한다.

쉘(Shell)
- 사용자의 명령어를 인식하여 프로그램을 호출하고 명령을 수행하는 명령어 해석기이다.
- 시스템과 사용자 간의 인터페이스를 담당한다.
- DOS의 COMMAND.COM과 같은 기능을 수행한다.
- 주기억장치에 상주하지 않고, 명령어가 포함된 파일 형태로 존재하며 보조 기억장치에서 교체 처리가 가능하다.
- 파이프라인 기능을 지원하고 입 · 출력 재지정을 통해 출력과 입력의 방향을 변경할 수 있다.
- 공용 Shell(Bourne Shell, C Shell, Korn Shell)이나 사용자 자신이 만든 Shell을 사용할 수 있다.

유틸리티(Utility Program)
- 일반 사용자가 작성한 응용 프로그램을 처리하는 데 사용한다.
- DOS에서의 외부 명령어에 해당된다.
- 유틸리티 프로그램에는 에디터, 컴파일러, 인터프리터, 디버거 등이 있다.

71 3개의 페이지를 수용할 수 있는 주기억장치이므로 아래 그림과 같이 3개의 페이지 프레임으로 표현할 수 있습니다.

참조 페이지	0	1	2	0	3	1	0	2
페이지 프레임	0	0	0	0	0	0	0	0
		1	1	1	3	3	3	2
			2	2	2	1	1	1
부재 발생	●	●	●		●	●		●

※ ● : 페이지 부재 발생

참조 페이지가 페이지 테이블에 없을 경우 페이지 결함(부재)이 발생됩니다. 초기에는 모든 페이지가 비어 있으므로 처음 0, 1, 2 페이지 적재 시 페이지 결함이 발생됩니다. LRU는 최근에 가장 오랫동안 사용하지 않은 페이지를 교체하는 기법이므로 참조 페이지 3을 참조할 때에는 1을 제거한 후 3을 가져오게 됩니다. 이러한 과정으로 모든 페이지에 대한 요구를 처리하고 나면 총 페이지 결함 발생 횟수는 6회입니다.

72 최적 적합(Best Fit)은 프로그램이나 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 단편화를 가장 작게 남기는 영역에 배치시키는 방법이므로, 사용중이 아닌 영역 중 9K의 작업이 들어갈 수 있는 가장 작은 영역은 10K의 공간이 있는 영역C입니다.

배치(Placement) 전략

- 배치 전략은 새로 반입되는 프로그램이나 데이터를 주기억장치의 어디에 위치시킬 것인지를 결정하는 전략이다.
- 최초 적합(First Fit) : 프로그램이나 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 첫 번째 분할 영역에 배치시키는 방법
- 최적 적합(Best Fit) : 프로그램이나 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 단편화를 가장 작게 남기는 분할 영역에 배치시키는 방법
- 최악 적합(Worst Fit) : 프로그램이나 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 단편화를 가장 많이 남기는 분할 영역에 배치시키는 방법

73 프로세스의 처리 시간보다 페이지 교체에 소요되는 시간이 더 많아지는 현상을 스래싱(Thrashing)이라고 합니다.

스래싱(Thrashing)

- 프로세스의 처리 시간보다 페이지 교체에 소요되는 시간이 더 많아지는 현상이다.
- 다중 프로그래밍 시스템이나 가상기억장치를 사용하는 시스템에서 하나의 프로세스 수행 과정중 자주 페이지 부재가 발생함으로써 나타나는 현상으로, 전체 시스템의 성능이 저하된다.
- 다중 프로그래밍의 정도가 높아짐에 따라 CPU의 이용률은 어느 특정 시점까지는 높아지지만, 다중 프로그래밍의 정도가 더욱 커지면 스래싱이 나타나고, CPU의 이용률은 급격히 감소하게 된다.
- 스래싱 현상 방지 방법
  - 다중 프로그래밍의 정도를 적정 수준으로 유지한다.
  - 페이지 부재 빈도(Page Fault Frequency)를 조절하여 사용한다.
  - 워킹 셋을 유지한다.
  - 부족한 자원을 증설하고, 일부 프로세스를 중단시킨다.
  - CPU 성능에 대한 자료의 지속적 관리 및 분석으로 임계치를 예상하여 운영한다.

74 프로세스(Process)는 일반적으로 프로세서(처리기, CPU)에 의해 처리되는 사용자 프로그램, 시스템 프로그램, 즉 실행중인 프로그램을 의미하며, 작업(Job), 태스크(Task)라고도 합니다. ②번의 내용은 프로시저(Procedure)에 대한 설명입니다.

프로세스의 정의

- PCB를 가진 프로그램
- 실기억장치에 저장된 프로그램
- 프로세서가 할당되는 실체로서, 디스패치가 가능한 단위
- 프로시저가 활동중인 것
- 비동기적 행위를 일으키는 주체
- 지정된 결과를 얻기 위한 일련의 계통적 동작
- 목적 또는 결과에 따라 발생되는 사건들의 과정
- 운영체제가 관리하는 실행 단위

75 D 클래스 주소는 멀티캐스팅(Multicasting)으로 사용하기 위해 예약되어 있습니다.

IP 주소(Internet Protocol Address)

- 인터넷에 연결된 모든 컴퓨터 자원을 구분하기 위한 고유한 주소이다.
- 숫자로 8비트씩 4부분, 총 32비트로 구성되어 있다.
- IP 주소는 네트워크 부분의 길이에 따라 다음과 같이 A 클래스에서 E 클래스까지 총 5단계로 구성되어 있다.
- A Class : 국가나 대형 통신망에 사용(0~127로 시작)
- B Class : 중대형 통신망에 사용(128~191로 시작)
- C Class : 소규모 통신망에 사용(192~223으로 시작)
- D Class : 멀티캐스트용으로 사용(224~239로 시작)
- E Class : 실험적 주소이며 공용되지 않음

76 두 개의 인접한 개방 시스템들 간에 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 하는 계층은 데이터 링크 계층(Data Link Layer)입니다.

OSI 7계층

물리 계층	• 전송에 필요한 두 장치 간의 실제 접속과 절단 등 기계적, 전기적, 기능적, 절차적 특성에 대한 규칙을 정의함 • 물리적 전송 매체와 전송 신호 방식을 정의하며, RS-232C, X.21 등의 표준이 있음

데이터 링크 계층	• 두 개의 인접한 개방 시스템들 간에 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 함 • 송신 측과 수신 측의 속도 차이를 해결하기 위한 흐름 제어 기능을 함 • 프레임의 시작과 끝을 구분하기 위한 프레임의 동기화 기능을 함 • 오류의 검출과 회복을 위한 오류 제어 기능을 함 • 프레임의 순서적 전송을 위한 순서 제어 기능을 함 • HDLC, LAPB, LLC, LAPD, PPP 등의 표준이 있음
네트워크 계층	• 개방 시스템들 간의 네트워크 연결을 관리하는 기능과 데이터의 교환 및 중계 기능을 함 • 네트워크 연결을 설정, 유지, 해제하는 기능을 함 • 경로 설정(Routing), 데이터 교환 및 중계, 트래픽 제어, 패킷 정보 전송을 수행함 • 관련 표준으로는 X.25, IP 등이 있음
전송 (트랜스포트) 계층	• 논리적 안정과 균일한 데이터 전송 서비스를 제공함으로써 종단 시스템(End-to-End) 간에 투명한 데이터 전송을 가능하게 함 • OSI 7계층 중 하위 3계층과 상위 3계층의 인터페이스(Interface)를 담당함 • 종단 시스템(End-to-End) 간의 전송 연결 설정, 데이터 전송, 연결 해제 기능을 함 • 주소 설정, 다중화, 오류 제어, 흐름 제어를 수행함 • TCP, UDP 등의 표준이 있음
세션 계층	• 송 · 수신측 간의 관련성을 유지하고 대화 제어를 담당하는 계층 • 대화(회화) 구성 및 동기 제어, 데이터 교환 관리 기능을 함 • 송 · 수신측의 대화(회화) 동기를 위해 전송하는 정보의 일정한 부분에 체크점을 두어 정보의 수신 상태를 체크하며, 이때의 체크점을 동기점이라고 함 • 동기점은 오류가 있는 데이터의 회복을 위해 사용하는 것으로, 종류에는 소동기점과 대동기점이 있음
표현 계층	• 응용 계층으로부터 받은 데이터를 세션 계층에 보내기 전에 통신에 적당한 형태로 변환하고, 세션 계층에서 받은 데이터는 응용 계층에 맞게 변환하는 기능을 함 • 서로 다른 데이터 표현 형태를 갖는 시스템 간의 상호 접속을 위해 필요한 계층 • 코드 변환, 데이터 암호화, 데이터 압축, 구문 검색, 정보 형식(포맷) 변환, 문맥 관리 기능을 함
응용 계층	• 사용자(응용 프로그램)가 OSI 환경에 접근할 수 있도록 서비스를 제공함 • 응용 프로세스 간의 정보 교환, 전자 사서함, 파일 전송 등의 서비스를 제공함

77 데이터 전송 전에 연결을 설정하지 않는 비연결형 서비스를 제공하며, TCP에 비해 상대적으로 단순한 헤더 구조를 가지고 있어 오버헤드가 적은 전송 계층의 프로토콜은 UDP(User Datagram Protocol)입니다.

UDP(User Datagram Protocol)

- 데이터 전송 전에 연결을 설정하지 않는 비연결형 서비스를 제공한다.
- TCP에 비해 상대적으로 단순한 헤더 구조를 가지므로, 오버헤드가 적다.
- 고속의 안정성 있는 전송 매체를 사용하여 빠른 속도를 필요로 하는 경우, 동시에 여러 사용자에게 데이터를 전달할 경우, 정기적으로 반복해서 전송할 경우에 사용한다.
- 실시간 전송에 유리하며, 신뢰성보다는 속도가 중요시되는 네트워크에서 사용된다.
- UDP 헤더에는 Source Port Number, Destination Port Number, Length, Checksum 등이 포함된다.

78 선점 스케줄링(Preemptive Scheduling)은 선점으로 인해 많은 오버헤드(Overhead)를 초래합니다.

비선점(Non-Preemptive) 스케줄링

- 이미 할당된 CPU를 다른 프로세스가 강제로 빼앗아 사용할 수 없는 스케줄링 기법이다.
- 프로세스가 CPU를 할당받으면 해당 프로세스가 완료될 때까지 CPU를 사용한다.
- 모든 프로세스에 대한 요구를 공정하게 처리할 수 있다.
- 프로세스 응답 시간의 예측이 용이하며, 일괄 처리 방식에 적합하다.
- 중요한 작업(짧은 작업)이 중요하지 않은 작업(긴 작업)을 기다리는 경우가 발생할 수 있다.
- 비선점 스케줄링의 종류에는 FCFS, SJF, 우선순위, HRN, 기한부 등의 알고리즘이 있다.

선점(Preemptive) 스케줄링

- 하나의 프로세스가 CPU를 할당받아 실행하고 있을 때 우선순위가 높은 다른 프로세스가 CPU를 강제로 빼앗아 사용할 수 있는 스케줄링 기법이다.
- 우선순위가 높은 프로세스를 빠르게 처리할 수 있다.
- 주로 빠른 응답 시간을 요구하는 대화식 시분할 시스템에 사용된다.
- 많은 오버헤드(Overhead)를 초래한다.
- 선점이 가능하도록 일정 시간 배당에 대한 인터럽트용 타이머 클록(Clock)이 필요하다.
- 선점 스케줄링의 종류에는 Round Robin, SRT, 선점 우선순위, 다단계 큐, 다단계 피드백 큐 등의 알고리즘이 있다.

79 일괄 처리(Batch Processing) 작업이 설정된 주기에 맞춰 자동으로 수행되도록 지원해주는 도구를 배치 스케줄러(Batch Scheduler)라고 합니다.

배치 스케줄러(Batch Scheduler)

- 일괄 처리(Batch Processing) 작업이 설정된 주기에 맞춰 자동으로 수행되도록 지원해주는 도구이다.
- 배치 스케줄러는 특정 업무(Job)를 원하는 시간에 처리할 수 있도록 지원한다는 특성 때문에 잡 스케줄러(Job Scheduler)

라고도 불린다.

- 주로 사용되는 배치 스케줄러에는 스프링 배치, Quartz 등이 있다.
- 스프링 배치(Spring Batch)
  - Spring Source 사와 Accenture 사가 2007년 공동 개발한 오픈 소스 프레임워크이다.
  - 스프링 프레임워크의 특성을 그대로 가져와 스프링이 가지고 있는 다양한 기능들을 모두 사용할 수 있다.
  - 데이터베이스나 파일의 데이터를 교환하는데 필요한 컴포넌트들을 제공한다.
- Quartz
  - 스프링 프레임워크로 개발되는 응용 프로그램들의 일괄 처리를 위한 다양한 기능을 제공하는 오픈 소스 라이브러리이다.
  - 수행할 작업과 수행 시간을 관리하는 요소들을 분리하여 일괄 처리 작업에 유연성을 제공한다.

80 프로토콜 구조가 다른 네트워크 간의 연결을 위해 프로토콜 변환 기능을 수행하는 장비는 게이트웨이(Gateway)입니다.

네트워크 관련 장비

- 허브(Hub) : 한 사무실이나 가까운 거리의 컴퓨터들을 연결하는 장치로, 각 회선을 통합적으로 관리하며, 신호 증폭 기능을 하는 리피터의 역할도 포함함
- 리피터(Repeater) : 전송되는 신호가 전송 선로의 특성 및 외부 충격 등의 요인으로 인해 원래의 형태와 다르게 왜곡되거나 약해질 경우 원래의 신호 형태로 재생하여 다시 전송하는 역할을 수행함
- 브리지(Bridge) : LAN과 LAN을 연결하거나 LAN 안에서의 컴퓨터 그룹(세그먼트)을 연결하는 기능을 수행함
- 스위치(Switch) : 브리지와 같이 LAN과 LAN을 연결하여 훨씬 더 큰 LAN을 만드는 장치
- 라우터(Router) : 브리지와 같이 LAN과 LAN의 연결 기능에 데이터 전송의 최적 경로를 선택할 수 있는 기능이 추가된 것으로, 서로 다른 LAN이나 LAN과 WAN의 연결도 수행함
- 게이트웨이(Gateway) : 전 계층(1~7계층)의 프로토콜 구조가 다른 네트워크의 연결을 수행함

81 객체지향 방법론의 구성 요소에는 객체(Object), 클래스(Class), 메시지(Message) 등이 있습니다. 패키지(Package)는 JAVA에서 관련 클래스나 인터페이스 등을 하나로 모아둔 것을 가리킵니다.

객체지향 방법론

- 현실 세계의 개체(Entity)를 기계의 부품처럼 하나의 객체(Object)로 만들어, 소프트웨어를 개발할 때 기계의 부품을 조립하듯이 객체들을 조립해서 필요한 소프트웨어를 구현하는 방법론이다.
- 구조적 기법의 문제점으로 인한 소프트웨어 위기의 해결책으로 채택되었다.
- 객체지향 방법론의 구성 요소에는 객체(Object), 클래스(Class), 메시지(Message) 등이 있다.
- 객체지향 방법론의 기본 원칙에는 캡슐화(Encapsulation), 정보 은닉(Information Hiding), 추상화(Abstraction), 상속성(Inheritance), 다형성(Polymorphism) 등이 있다.
- 객체지향 방법론의 절차 : 요구 분석 단계 → 설계 단계 → 구현 단계 → 테스트 및 검증 단계 → 인도 단계

82 기존의 시스템이나 소프트웨어를 구성하는 컴포넌트를 조합하여 하나의 새로운 애플리케이션을 만드는 방법론을 컴포넌트 기반(CBD; Component Based Design) 방법론이라고 합니다. 나머지 보기로 제시된 방법론들의 개념도 정리해 두세요.

- 구조적 방법론 : 정형화된 분석 절차에 따라 사용자 요구사항을 파악하여 문서화하는 처리(Precess) 중심의 방법론
- 정보공학 방법론 : 정보 시스템의 개발을 위해 계획, 분석, 설계, 구축에 정형화된 기법들을 상호 연관성 있게 통합 및 적용하는 자료(Data) 중심의 방법론
- 애자일(Agile) 방법론 : 애자일은 '민첩한', '기민한'이라는 의미로, 고객의 요구사항 변화에 유연하게 대응할 수 있도록 일정한 주기를 반복하면서 개발 과정을 진행하는 방법론

83 소프트웨어 비용 산정 기법에는 하향식 비용 산정 기법과 상향식 비용 산정 기법이 있습니다. 혼합식 비용 산정 기법은 존재하지 않습니다.

소프트웨어 비용 산정

- 소프트웨어의 개발 규모를 소요되는 인원, 자원, 기간 등으로 확인하여 실행 가능한 계획을 수립하기 위해 필요한 비용을 산정하는 것이다.
- 소프트웨어 비용 산정을 너무 높게 산정할 경우 예산 낭비와 일의 효율성 저하를 초래할 수 있고, 너무 낮게 산정할 경우 개발자의 부담이 가중되고 품질문제가 발생할 수 있다.
- 소프트웨어 비용 산정 기법에는 하향식 비용 산정 기법과 상향식 비용 산정 기법이 있다.

84 하향식 비용 산정 기법에는 전문가 감정 기법, 델파이 기법 등이 있습니다.

하향식 비용 산정 기법

- 과거의 유사한 경험을 바탕으로 전문 지식이 많은 개발자들이 참여한 회의를 통해 비용을 산정하는 비과학적인 방법이다.
- 프로젝트의 전체 비용을 산정한 후 각 작업별로 비용을 세분화한다.

- 하향식 비용 산정 기법에는 전문가 감정 기법, 델파이 기법 등이 있다.
- 전문가 감정 기법
  - 조직 내에 있는 경험이 많은 두 명 이상의 전문가에게 비용 산정을 의뢰하는 기법이다.
  - 가장 편리하고 신속하게 비용을 산정할 수 있으며, 의뢰자로부터 믿음을 얻을 수 있다.
  - 새로운 프로젝트에는 과거의 프로젝트와 다른 요소들이 있다는 것을 간과할 수 있다.
  - 새로운 프로젝트와 유사한 프로젝트에 대한 경험이 없을 수 있다.
  - 개인적이고 주관적일 수 있다.
- 델파이 기법
  - 전문가 감정 기법의 주관적인 편견을 보완하기 위해 많은 전문가의 의견을 종합하여 산정하는 기법이다.
  - 전문가들의 편견이나 분위기에 지배되지 않도록 한 명의 조정자와 여러 전문가로 구성된다.

85 • 노력 = LOC/1인당 월평균 생산 라인 수 = 30000/600 = 50
• 투입인원 = 노력/개발 기간 = 50/10개월 = 5명
※ 단순하게 '30,000라인 / (10개월 * 600라인) = 5'로 계산해도 됩니다.

86 Putnam 모형을 기초로 개발된 자동화 추정 도구는 SLIM입니다. ESTIMACS는 FP 모형을 기초로 개발되었습니다.

Putnam 모형
- 소프트웨어 생명 주기의 전 과정 동안에 사용될 노력의 분포를 가정해 주는 모형이다.
- 푸트남(Putnam)이 제안한 것으로 생명 주기 예측 모형이라고도 한다.
- 시간에 따른 함수로 표현되는 Rayleigh-Norden 곡선의 노력 분포도를 기초로 한다.
- 대형 프로젝트의 노력 분포 산정에 이용되는 기법이다.
- 개발 기간이 늘어날수록 프로젝트 적용 인원의 노력이 감소한다.

87 각종 컴퓨팅 자원을 중앙 컴퓨터에 두고 인터넷 기능을 갖는 단말기로 언제 어디서나 인터넷을 통해 컴퓨터 작업을 수행할 수 있는 환경을 클라우드 컴퓨팅 환경이라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.
- 그리드 컴퓨팅(Grid Computing) : 인터넷에 연결된 수많은 컴퓨터를 격자 구조로 모두 연결하여 하나의 컴퓨터처럼 사용할 수 있게 하는 것
- 사물 인터넷(IoT) : 정보 통신 기술을 기반으로 실세계(Physical World)와 가상 세계(Virtual World)의 다양한 사물들을 인터넷으로 서로 연결하여 진보된 서비스를 제공하기 위한 서비스 기반 기술
- 차세대 통신망(NGN) : 유선망뿐만 아니라 이동 사용자를 목표로 하며, 이동통신에서 제공하는 완전한 이동성(Full Mobility) 제공을 목표로 개발되고 있는 차세대 통신망

88 그림은 망형(Mesh)으로, 보통 공중 데이터 통신망에서 사용됩니다. 분산 처리 시스템에서 사용되는 네트워크 구성 방식은 계층형(Tree)입니다.

통신망의 구성 형태
- 성형(Star, 중앙 집중형) : 중앙에 중앙 컴퓨터가 있고, 이를 중심으로 단말장치들이 연결되는 중앙 집중식의 네트워크 구성 형태
- 링형(Ring, 루프형) : 컴퓨터와 단말장치들을 서로 이웃하는 것끼리 포인트 투 포인트(Point-to-Point) 방식으로 연결시킨 형태
- 버스형(Bus) : 한 개의 통신 회선에 여러 대의 단말장치가 연결되어 있는 형태
- 계층형(Tree, 분산형) : 중앙 컴퓨터와 일정 지역의 단말장치까지는 하나의 통신 회선으로 연결시키고, 이웃하는 단말장치는 일정 지역 내에 설치된 중간 단말장치로부터 다시 연결시키는 형태
- 망형(Mesh) : 모든 지점의 컴퓨터와 단말장치를 서로 연결한 형태로, 노드의 연결성이 높음

89 실시간으로 발생하는 많은 사건들 중 의미가 있는 것만을 추출할 수 있도록 사건 발생 조건을 정의하는 데이터 처리 방법을 복잡 이벤트 처리(CEP; Complex Event Processing)라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.
- 소프트웨어 에스크로(Software Escrow) : 소프트웨어 개발자의 지식재산권을 보호하고 사용자는 저렴한 비용으로 소프트웨어를 안정적으로 사용 및 유지보수 받을 수 있도록 소스 프로그램과 기술 정보 등을 제3의 기관에 보관하는 것
- 서비스형 소프트웨어(SaaS) : 소프트웨어의 여러 기능 중에서 사용자가 필요로 하는 서비스만 이용할 수 있도록 한 소프트웨어
- 딥 러닝(Deep Learning) : 인간의 두뇌를 모델로 만들어진 인공 신경망(ANN; Artificial Neural Network)을 기반으로 하는 기계 학습 기술

90 디스크의 속도 향상을 위해 여러 개의 하드디스크로 디스크 배열을 구성하여 파일을 구성하고 있는 데이터 블록들을 서로 다른 디스크들에 분산 저장하는 기술을 RAID(Redundant Array of Inexpensive Disk)라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.
- 고가용성(HA; High Availability) : 긴 시간동안 안정적인 서비스 운영을 위해 장애 발생 시 즉시 다른 시스템으로 대체 가능한 환경을 구축하는 메커니즘

- 멤스(MEMS; Micro-Electro Mechanical Systems) : 초정밀 반도체 제조 기술을 바탕으로 센서, 액추에이터(Actuator) 등 기계 구조를 다양한 기술로 미세 가공하여 전기기계적 동작을 할 수 있도록 한 초미세 장치
- 엠디스크(M-DISC; Millennial DISC) : 한 번의 기록만으로 자료를 영구 보관할 수 있는 광 저장장치

91 일련의 데이터를 정의하고 설명해 주는 데이터를 메타 데이터(Meta Data)라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- 뷰(View) : 사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블
- 브로드 데이터(Broad Data) : 다양한 채널에서 소비자와 상호 작용을 통해 생성된, 기업 마케팅에 있어 효율적이고 다양한 데이터이며, 이전에 사용하지 않거나 알지 못했던 새로운 데이터나, 기존 데이터에 새로운 가치가 더해진 데이터
- 빅데이터(Big Data) : 기존의 관리 방법이나 분석 체계로는 처리하기 어려운 막대한 양의 정형 또는 비정형 데이터 집합

92 Secure SDLC에서 조직의 정보보호 관련 보안 정책을 참고하여 소프트웨어 개발에 적용할 수 있는 보안 정책 항목들의 출처, 요구 수준, 세부 내용 등을 문서화하는 단계는 요구사항 분석 단계입니다.

Secure SDLC의 단계별 주요 보안 활동

- 요구사항 분석 단계
  - 보안 항목에 해당하는 요구사항을 식별하는 작업을 수행
  - 전산화되는 정보가 가지고 있는 보안 수준을 보안 요소별로 등급을 구분하여 분류
  - 조직의 정보보호 관련 보안 정책을 참고하여 소프트웨어 개발에 적용할 수 있는 보안 정책 항목들의 출처, 요구 수준, 세부 내용 등을 문서화
- 설계 단계
  - 식별된 보안 요구사항들을 소프트웨어 설계서에 반영하고, 보안 설계서를 작성
  - 소프트웨어에서 발생할 수 있는 위협을 식별하여 보안대책, 소요예산, 사고 발생 시 영향 범위와 대응책 등을 수립
  - 네트워크, 서버, 물리적 보안, 개발 프로그램 등 환경에 대한 보안통제 기준을 수립하여 설계에 반영
- 구현 단계
  - 표준 코딩 정의서 및 소프트웨어 개발 보안 가이드를 준수하며, 설계서에 따라 보안 요구사항들을 구현
  - 개발 과정 중에는 지속적인 단위 테스트를 통해 소프트웨어에 발생할 수 있는 보안 취약점을 최소화
  - 코드 점검 및 소스 코드 진단 작업을 통해 소스 코드의 안정성을 확보
- 테스트 단계
  - 설계 단계에서 작성한 보안 설계서를 바탕으로 보안 사항들이 정확히 반영되고 동작되는지 점검
  - 동적 분석 도구 또는 모의 침투테스트를 통해 설계 단계에서 식별된 위협들의 해결여부를 검증
  - 설계 단계에서 식별된 위협들 외에도 구현 단계에서 추가로 제시된 위협들과 취약점들을 점검할 수 있도록 테스트 계획을 수립하고 시행
- 유지보수 단계
  - 이전 과정을 모두 수행하였음에도 발생할 수 있는 보안 사고들을 식별하고, 사고 발생 시 이를 해결하고 보안 패치를 실시

93 공개키 암호화 기법은 암호화 할 때는 공개키(Public Key)를, 복호화 할 때는 비밀키(Secret Key)를 사용합니다.

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.
- 비대칭 암호 기법이라고도 하며, 대표적으로는 RSA(Rivest Shamir Adleman) 기법이 있다.
- 장점 : 키의 분배가 용이하고, 관리해야 할 키의 개수가 적음
- 단점 : 암호화/복호화 속도가 느리며, 알고리즘이 복잡하고, 개인키 암호화 기법보다 파일의 크기가 큼

94 여러 곳에 분산된 공격 지점에서 한 곳의 서버에 대해 분산 서비스 공격을 수행하는 것을 DDoS(Distributed Denial of Service, 분산 서비스 거부) 공격이라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- 스미싱(Smishing)
  - 문자 메시지(SMS)를 이용해 사용자의 개인 신용 정보를 빼내는 수법이다.
  - 초기에는 문자 메시지를 이용해 개인 비밀정보나 소액 결제를 유도하는 형태로 시작되었다.
  - 현재는 각종 행사 안내, 경품 안내 등의 문자 메시지에 링크를 걸어 안드로이드 앱 설치 파일인 apk 파일을 설치하도록 유도하여 사용자 정보를 빼가는 수법으로 발전하고 있다.
- 무작위 대입 공격(Brute Force Attack) : 암호화된 문서의 암호키를 찾아내기 위해 적용 가능한 모든 값을 대입하여 공격하는 방식
- 트로이 목마(Trojan Horse) : 정상적인 기능을 하는 프로그램으로 위장하여 프로그램 내에 숨어 있다가 해당 프로그램이 동작할

때 활성화되어 부작용을 일으키는 것으로, 자기 복제 능력은 없음

95 가상 사설 네트워크로서 인터넷 등 통신 사업자의 공중 네트워크와 암호화 기술을 이용하여 사용자가 마치 자신의 전용 회선을 사용하는 것처럼 해주는 보안 솔루션을 VPN(Virtual Private Network, 가상 사설 통신망)이라고 합니다. 나머지 보기로 제시된 용어들의 개념도 확인하고 넘어가세요.

- ESM(Enterprise Security Management) : 다양한 장비에서 발생하는 로그 및 보안 이벤트를 통합하여 관리하는 보안 솔루션
- 침입 방지 시스템(IPS; Intrusion Prevention System) : 방화벽과 침입 탐지 시스템을 결합한 것으로, 비정상적인 트래픽을 능동적으로 차단하고 격리하는 등의 방어 조치를 취하는 보안 솔루션
- NAC(Network Access Control) : 네트워크에 접속하는 내부 PC의 MAC 주소를 IP 관리 시스템에 등록한 후 일관된 보안 관리 기능을 제공하는 보안 솔루션

96 네트워크 내의 원활한 흐름을 위해 송 · 수신 측 사이에 전송되는 패킷의 양이나 속도를 규제하는 흐름 제어(Flow Control) 방식에는 수신 측의 확인 신호(ACK)를 받은 후에 다음 패킷을 전송하는 정지-대기(Stop-and-Wait) 방식과 수신 통지를 이용하여 송신 데이터의 양을 조절하는 슬라이딩 윈도우(Sliding Window) 방식이 있습니다.

흐름 제어(Flow Control)

- 네트워크 내의 원활한 흐름을 위해 송 · 수신 측 사이에 전송되는 패킷의 양이나 속도를 규제하는 기능이다.
- 송신 측과 수신 측 간의 처리 속도 또는 버퍼 크기의 차이에 의해 생길 수 있는 수신 측 버퍼의 오버플로(Overflow)를 방지하기 위한 기능이다.
- 정지-대기(Stop-and-Wait)
  - 수신 측의 확인 신호(ACK)를 받은 후에 다음 패킷을 전송하는 방식이다.
  - 한 번에 하나의 패킷만을 전송할 수 있다.
- 슬라이딩 윈도우(Sliding Window)
  - 확인 신호, 즉 수신 통지를 이용하여 송신 데이터의 양을 조절하는 방식이다.
  - 수신 측의 확인 신호를 받지 않더라도 미리 정해진 패킷의 수만큼 연속적으로 전송하는 방식으로, 한 번에 여러 개의 패킷을 전송할 수 있어 전송 효율이 좋다.
  - 송신 측은 수신 측으로부터 확인 신호(ACK) 없이도 보낼 수 있는 패킷의 최대치를 미리 약속받는데, 이 패킷의 최대치가 윈도우 크기(Window Size)를 의미한다.
  - 윈도우 크기는 상황에 따라 변한다. 즉, 수신 측으로부터 이전에 송신한 패킷에 대한 긍정 수신 응답(ACK)이 전달된 경우 윈도우 크기는 증가하고, 수신 측으로부터 이전에 송신한 패킷에 대한 부정 수신 응답(NAK)이 전달된 경우 윈도우 크기는 감소한다.

97 널 포인터(Null Pointer) 역참조를 방지하기 위해서는 널이 될 수 있는 포인터를 이용하기 전에 널 값을 갖고 있는지 검사해야 합니다.

널 포인터(Null Pointer) 역참조

- 널 포인터가 가리키는 메모리에 어떠한 값을 저장할 때 발생하는 보안 약점이다.
- 많은 라이브러리 함수들이 오류가 발생할 경우 널 값을 반환하는데, 이 반환값을 포인터로 참조하는 경우 발생한다.
- 대부분의 운영체제에서 널 포인터는 메모리의 첫 주소를 가리키며, 해당 주소를 참조할 경우 소프트웨어가 비정상적으로 종료될 수 있다.
- 공격자가 널 포인터 역참조로 발생하는 예외 상황을 악용할 수 있다.
- 널이 될 수 있는 포인터를 이용하기 전에 널 값을 갖고 있는지 검사함으로써 방지할 수 있다.

98 Public 메소드로부터 Private 배열이 반환될 경우 배열의 주소가 외부로 공개되어 외부에서 Private 배열에 접근할 수 있게 됩니다.

Public 메소드로부터 반환된 Private 배열

- 선언된 클래스 내에서만 접근이 가능한 Private 배열을 모든 클래스에서 접근이 가능한 Public 메소드에서 반환할 때 발생하는 보안 약점이다.
- Public 메소드가 Private 배열을 반환하면 배열의 주소가 외부로 공개되어 외부에서 접근할 수 있게 된다.
- Private 배열을 별도의 메소드를 통해 조작하거나, 동일한 형태의 복제본으로 반환받은 후 값을 전달하는 방식으로 방지할 수 있다.

99 고정된 패스워드, 패스 프레이즈, 아이핀 등과 같이 사용자가 기억하고 있는 정보를 기반으로 인증을 수행하는 것을 지식 기반 인증(Something You Know)이라고 합니다.

인증의 주요 유형

- 지식 기반 인증(Something You Know)
  - 사용자가 기억하고 있는 정보를 기반으로 인증을 수행하는 것이다.
  - 지식 기반 인증은 사용자의 기억을 기반으로 하므로 관리 비용이 저렴하다.
  - 사용자가 인증 정보를 기억하지 못하면 본인이라도 인증받지 못한다.
  - 고정된 패스워드(Password) : 사용자가 알고 있는 비밀번

호를 접속할 때 마다 반복해서 입력함
- 패스 프레이즈(Passphrase) : 'iloveyou'와 같이 일반 패스워드보다 길이가 길고 기억하기 쉬운 문장을 활용하여 비밀번호를 구성하는 방법
- 아이핀(i-PIN) : 인터넷에서 주민등록번호 대신 쓸 수 있도록 만든 사이버 주민등록번호로, 사용자에 대한 신원확인을 완료한 후에 본인확인기관에서 온라인으로 발행함

• 소유 기반 인증(Something You Have)
- 사용자가 소유하고 있는 것을 기반으로 인증을 수행하는 것이다.
- 소유 기반 인증은 소유물이 쉽게 도용될 수 있으므로 지식 기반 인증 방식이나 생체 기반 인증 방식과 함께 사용된다.
- 신분증 : 사용자의 사진이 포함된 주민등록증, 운전면허증, 여권 등을 사용하여 사용자의 신분을 확인함
- 메모리 카드(토큰) : 마그네틱 선에 보안 코드를 저장해서 사용하는 것으로, 카드 리더기를 통해서만 읽을 수 있음(예 일반 은행 입출금 카드)
- 스마트 카드 : 마이크로프로세서, 카드 운영체제, 메모리 등으로 구성되어 사용자의 정보뿐만 아니라 특정 업무를 처리할 수 있는 기능이 내장되어 있음(예 IC칩이 내장된 카드)
- OTP(One Time Password) : 사용자가 패스워드를 요청할 때마다 암호 알고리즘을 통해 새롭게 생성된 패스워드를 사용하는 것으로, 한 번 사용된 패스워드는 폐기됨

• 생체 기반 인증(Something You Are)
- 사용자의 고유한 생체 정보를 기반으로 인증을 수행하는 것이다.
- 생체 기반 인증은 사용이 쉽고 도난의 위험도 적으며 위조가 어렵다.
- 생체 인증 대상 : 지문, 홍채/망막, 얼굴, 음성, 정맥 등

• 위치 기반 인증(Somewhere You Are)
- 인증을 시도하는 위치의 적절성을 확인한다.
- 예 콜백, GPS나 IP 주소를 이용한 위치 기반 인증

100 세션ID가 노출되지 않도록 하려면 URL Rewrite 기능을 사용하지 않아야 합니다. URL Rewrite 기능은 URL에 사용자의 세션ID를 포함하는 기능으로, 외부에 사용자의 세션ID가 노출될 수 있습니다.

세션ID의 관리 방법
• 세션ID는 안전한 서버에서 최소 128비트의 길이로 생성한다.
• 세션ID의 예측이 불가능하도록 안전한 난수 알고리즘을 적용한다.
• 세션ID가 노출되지 않도록 URL Rewrite 기능을 사용하지 않는 방향으로 설계한다.
• 로그인 시 로그인 전의 세션ID를 삭제하고 재할당한다.
• 장기간 접속하고 있는 세션ID는 주기적으로 재할당되도록 설계한다.