



# 정보처리기사 실기

길벗알앤디 지음



독자의 1초까지 아껴주는  
정성을 만나 보세요.



## 지은이

김정준 - 한국산업기술대학교 컴퓨터공학부 교수

길벗알앤디 - 강윤석, 김용갑, 김우경, 김유희

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 실기 - 시나공 시리즈 34

초판 발행 · 2020년 6월 26일

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](http://www.gilbut.co.kr) 이메일 · [gilbut@gilbut.co.kr](mailto:gilbut@gilbut.co.kr)

기획 및 책임 편집 · 강윤석([kys@gilbut.co.kr](mailto:kys@gilbut.co.kr)), 김미정([kongkong@gilbut.co.kr](mailto: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, 2020

- 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 100986

비매품

독자의 1초를 아껴주는 정성 길벗출판사

길벗 | IT실용, IT/일반 수험서, IT전문서, 경제실용서, 취미실용서, 건강실용서, 자녀교육서

더퀘스트 | 인문교양서, 비즈니스서

길벗이지독 | 어학단행본, 어학수험서

길벗스쿨 | 국어학습서, 수학학습서, 유아학습서, 어학학습서, 어린이교양서, 교과서

페이스북 · [www.facebook.com/gilbutzigu](https://www.facebook.com/gilbutzigu)

커뮤니티 · <http://cafe.naver.com/gilbutit>




2020년 2회 정보처리기사 실기 시험, 이렇게 준비하세요.

 4

## 1 핵심요약

1장 프로그래밍 언어 활용	16
2장 요구사항 확인	69
3장 데이터 입·출력 구현	75
4장 통합 구현	82
5장 서버 프로그램 구현	91
6장 화면 설계	98
7장 애플리케이션 테스트 관리	102
8장 SQL 응용	111
9장 소프트웨어 개발 보안 구축	123
10장 응용 SW 기초 기술 활용	134
11장 제품 소프트웨어 패키징	155

## 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** 시나공440101

## 2 최신기출문제

2020년 1회 정보처리기사 실기

 162

## 3 실전 모의고사

1회 실전 모의고사	178
2회 실전 모의고사	192
3회 실전 모의고사	204
4회 실전 모의고사	217
5회 실전 모의고사	230



## 2020년 2회 정보처리기사 실기 시험, 이렇게 준비하세요.

### 이왕 공부하는 거 조금만 더 열심히 해서 한 번에 딱 합격합시다!!

440121



지난 5월 24일 치러진 2020년 1회 정보처리기사 실기 시험에는 예상하지 못한 필기 도서의 내용이 출제되어 수험생은 물론 저희도 적잖이 당황했습니다. 아시다시피 개편된 정보처리 실기 시험과 필기 시험은 출제 범위가 거의 같기 때문에 2020 시나공 정보처리기사 실기 교재는 필기 교재에서 다뤘던 이론적인 내용은 제외하고 좀 더 실무적인 내용 위주로 교재 내용을 구성했기 때문입니다. 1회 실기 시험을 대비해서 2020 시나공 정보처리기사 필기 교재로 공부하고 실기 교재를 학습했다면 최대 18문제를 맞혔겠지만 2020 시나공 정보처리기사 실기 교재만으로 준비했다면 아쉽게도 14문제만 맞혔을 것으로 예상됩니다. 물론 합격 점수는 초과했지만 좀 더 높은 적중률을 보이지 못한 것에 대한 안타까움이 큼니다. 그래서 준비했습니다. 필기 교재에서 나올만한 내용을 짝 정리하고, 실기 교재의 방대한 내용도 효율적으로 학습할 수 있도록 핵심만 뽑아 186개로 요약 정리했습니다. 아울러 학습 전략을 세우고 저자들이 공을 들여 예상 모의고사도 5회분을 제작했습니다. 아직 시험이 완전히 정착되지 않았기 때문에 시험대비 자료만 보는 것은 위험할 수 있으니 꼭 2020 정보처리기사 실기 교재와 병행해서 공부해야 합니다.

이왕 공부하는 거 조금만 더 열심히 해서 한 번에 딱 합격합시다!! 시나공도 같이 노력하겠습니다.





## 학습전략

아직 시험이 한 번밖에 시행되지 않아 특정 부분만 꼭 짚어서 공부할 수는 없지만 1회 기사 실기 시험과 1, 2회 기능사 실기 시험을 분석한 결과 거칠지만 2회 실기 시험의 방향을 대충은 파악할 수 있을 것 같아 장별로 학습 방법을 정리해 봤습니다. 꼭 읽어보시고 전략을 세워 효율적으로 공부하시기 바랍니다.

### 1장 프로그래밍 언어 활용

C와 JAVA로 된 코드를 읽고 해석할 수 있어야 합니다.

1회 실기 시험에서는 연산자, 배열, 반복문 등 기본 문법 위주의 코드를 분석하여 결과를 적는 기초적인 문제가 출제되었지만, 2020년 2회 기능사 실기 시험에 재귀 함수가 출제된 것으로 보아 프로그램 코드의 난이도는 더 높아질 것으로 예상됩니다. 변경된 정보처리기사 시험에서 중점을 둔 것이 실무 개발능력강화라는 것을 감안할 때 이 부분은 앞으로 난이도가 더 높아질 것을 예상할 수 있게 합니다. 프로그램과 관련된 문제가 기사에서 3문제, 기능사에서 6문제가 출제되었는데 이런 문제들은 문법 학습만으로는 해결할 수 없다는 것을 명심하고, 교재에 수록된 다양한 예제를 풀어보세요. 프로그램 코드들은 반드시 코드를 디버깅하여 결과를 꼭 확인해야 합니다.

파이썬 관련 문제가 아직 출제되지는 않았지만 시험 범위에 포함되어 있기 때문에 무시할 수는 없습니다. 다만 아직은 출제비중이 낮으니 다른 장을 다 공부하고 여유가 된다면 학습하시기 바랍니다.

[교재] 정보처리기사 실기 7색선 2필드(1권 72쪽)

[1회 실기 기출문제 4번] 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.)

```
public class Test {
    static int[] arr() {
        int a[] = new int[4];
        int b = a.length;
        for(int i = 0; i < b; i++)
            a[i] = i;
        return a;
    }

    public static void main(String[] args) {
        int a[] = arr();
        for(int i = 0; i < a.length; i++)
            System.out.print(a[i] + " ");
    }
}
```

[정답] 0 1 2 3

[교재] 정보처리기사 실기 5색선 4필드(1권 58쪽)

[1회 실기 기출문제 14번] 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.)

```
#include <stdio.h>
main() {
    int c = 1;
    switch (3) {
        case 1: c += 3;
        case 2: c++;
        case 3: c = 0;
        case 4: c += 3;
        case 5: c -= 10;
        default: c--;
    }
    printf("%d", c);
}
```

[정답] -8



## 2020년 2회 정보처리기사 실기 시험, 이렇게 준비하세요.

[교재] 정보처리기사 실기 10섹션 1필드(1권 102쪽)

[1회 실기 기출문제 20번] 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.)

```
#include <stdio.h>
void align(int a[]) {
    int temp;
    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 4 - i; j++)
            if (a[j] > a[j+1]) {
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
}

main() {
    int a[] = { 85, 75, 50, 100, 95 };
    align(a);
    for (int i = 0; i < 5; i++)
        printf("%d ", a[i]);
}
```

[정답] 50 75 85 95 100

## 2장 요구사항 확인

요구사항에 대한 개념, 원리, 과정을 숙지하고, 다이어그램을 해석할 수 있어야 합니다.

1회 실기 시험에서는 변경 전 출제 범위였던 비용 산정 문제 하나만 출제되었습니다. 출제 비중은 낮지만 무시할 수 없는 부분입니다. 소프트웨어 개발에 있어 고객의 요구사항을 파악하는 일이 빠져서는 안 되기 때문이죠. 실기 시험의 특성상 세부적인 특징들을 모두 암기하기 보다는 핵심적인 단어나 원리, 과정 위주로 학습하는 것이 효율적입니다. 또한 문제로 제시된 다이어그램의 명칭을 말할 수 있고 의미를 해석할 수 있어야 합니다.

[교재] 정보처리기사 필기 160섹션 2필드(704쪽)

[1회 실기 실기 기출문제 6번] LOC 기법에 의하여 예측된 총 라인 수가 30,000라인, 개발에 참여할 프로그래머가 5명, 프로그래머들의 평균 생산성이 월간 300라인일 때 개발에 소요되는 기간을 쓰시오.

[정답]

- 계산과정 :  $(30,000/300)/5$
- 답 : 20개월

※ LOC 기법은 정보처리기사 필기 교재에 수록된 내용입니다. LOC 기법은 물론 관련 내용을 모두 정리하여 본 합격 전략집에 수록 했습니다.



### 3장 데이터 입·출력 구현

용어들의 개념은 물론 특성과 종류까지 모두 암기해야 합니다.

1회 실기 시험에서 2문제가 출제되었는데 모두 개념을 묻는 문제였습니다. 아직 시행 초기라 개념적인 문제가 고정된 출제 형태라고 단정 지을 수는 없지만 교재에 수록된 실무적인 내용 외에 이론적인 내용도 함께 학습할 필요가 있습니다. 실무적인 내용이 아니라서 실기 교재에 수록하지 않은 내용들은 2회 실기 시험 대비용으로 제공하는 본 합격 전략집에 모두 정리해 놓았으니 꼭 학습하시기 바랍니다.

[교재] 정보처리기사 필기 87색션 2필드(37쪽)

[1회 실기 기출문제 7번] 다음은 트랜잭션의 특징이다. 괄호(①, ②)에 들어갈 알맞은 특징을 쓰시오. (5점)

( ① )	트랜잭션의 연산은 데이터베이스에 모두 반영되도록 완료(Commit)되든지 아니면 전혀 반영되지 않도록 복구(Rollback)되어야 한다. (All or Nothing)
일관성	트랜잭션이 그 실행을 성공적으로 완료하면 언제나 일관성 있는 데이터베이스 상태로 변환한다.
( ② )	둘 이상의 트랜잭션이 동시에 병행 실행되는 경우 어느 하나의 트랜잭션 실행중에 다른 트랜잭션의 연산이 끼어들 수 없다.
지속성	성공적으로 완료된 트랜잭션의 결과는 시스템이 고장나더라도 영구적으로 반영되어야 한다.

[정답] ① 원자성 ② 독립성

※ 트랜잭션의 특징은 정보처리기사 필기 교재에 수록된 내용입니다. 트랜잭션의 특징은 물론 관련 내용을 모두 정리하여 본 합격 전략집에 수록 했습니다.

[교재] 정보처리기사 실기 34색션 1필드(권 377쪽)

[1회 실기 기출문제 9번] 데이터베이스에서 비정규화(Denormalization)의 개념을 설명하시오.

[정답] 비정규화는 정규화된 데이터 모델을 통합, 중복, 분리하는 과정으로, 의도적으로 정규화 원칙을 위배 하는 행위이다.





## 2020년 2회 정보처리기사 실기 시험, 이렇게 준비하세요.

### 4장 통합 구현

**통합 구현과 관련된 용어의 의미를 확실히 기억해 두세요.**

통합 구현은 환경에 따라 과정이나 방법이 지속적으로 변하기 때문에 이와 관련된 문제는 제한적일 수 있습니다. 그래서 서인지 1회 실기 시험에서도 실무적인 과정이나 방법을 묻는 문제가 아닌 용어의 의미를 묻는 문제가 출제되었습니다. 이 장에서는 연계 방식의 종류, 연계 솔루션, 연계를 위한 프로토콜, 언어 등과 같이 변화에 관계없이 답이 명확한 용어의 의미를 확실히 숙지해 두세요.

[교재] 정보처리기사 실기 46섹션 1필드(권 447쪽)

[1회 실기 기출문제 1번] SW 인터페이스 구현에 관련된 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오.

웹 페이지의 기본 형식인 HTML의 문법이 각 웹브라우저에서 상호 호환적이지 못하다는 문제와 SGML의 복잡함을 해결하기 위하여 개발된 ( )은 다른 특수한 목적을 갖는 마크업 언어이다. 원활한 데이터의 연계를 위해 송·수신 시스템 간에 전송되는 데이터가 동일한 구조로 구성될 수 있도록 형태를 정의하는 역할을 수행하며, 다음과 같은 특징이 있다.

유니코드 문자(Unicode Text)	텍스트 데이터 형식으로 유니코드를 사용하여 전 세계 언어를 지원한다.
( ) 파서(Parser)	대다수의 웹 브라우저가 해석을 위한 번역기(Parser)를 내장하고 있다.
마크업(Markup)과 내용(Content)	<ul style="list-style-type: none"> <li>• ( ) 문서의 문자들은 마크업과 내용으로 구분된다.</li> <li>• 일반적으로 마크업은 "&lt;"로 시작하여 "&gt;"로 끝나는 태그(Tag)를 의미하고, 그 외의 문자열은 내용에 해당한다.</li> </ul>
엘리먼트(Element)	마크업과 내용으로 이루어지는 하나의 요소를 의미한다.

[정답] XML(eXtensible Markup Language)

[교재] 정보처리기사 실기 64섹션 2필드(276쪽)

[1회 실기 기출문제 2번] 소프트웨어 인터페이스 구현에 관련된 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오.

( )은 속성-값 쌍(Attribute-Value Pairs)으로 이루어진 데이터 객체를 전달하기 위해 사람이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷이다. 비동기 처리에 사용되는 AJAX에서 XML을 대체하여 사용되고 있다. 다음은 ( )이 가질 수 있는 자료 기본형이다.

숫자	정수와 실수를 표현한다.
문자열	유니코드 문자로 표현하며, 큰따옴표(" ")로 묶는다.
참/거짓	참(True) 또는 거짓(False)을 표현한다.
배열	다양한 요소들을 침표로 구분하여 표현하며, 대괄호([ ])로 묶는다.
객체	이름/값 쌍으로 표현하며, 중괄호({ })로 묶는다.
NULL	아직 알려지지 않거나 모르는 값을 표현하기 위한 자료형이다.

[정답] JSON(JavaScript Object Notation)

※ JSON은 정보처리기사 실기 교재에 수록된 내용입니다. JSON 기법은 물론 관련 내용을 모두 정리하여 본 합격 전략집에 수록 했습니다.





## 5장 서버 프로그램 구현

프로그래밍 언어 부분과 중복되므로 이 장에서는 개념과 특징만 확실히 기억하면 됩니다.

1회 실기 시험에서는 모듈에 대해서만 2문제가 출제되었지만 1회 필기 시험에서는 모듈과 함께 객체지향의 개념과 특성, 아키텍처 패턴, 디자인 패턴 등이 출제되었습니다. 구현과 관련된 내용은 프로그래밍 언어 부분과 중복되므로 이 장에서 학습할 내용은 개념적인 내용 위주라고 할 수 있습니다. 필기 시험에 출제되었던 아키텍처 패턴이나 디자인 패턴과 같은 이론들도 충분히 출제될 수 있습니다. 이 부분도 실무적인 내용이 아니라서 실기 교재에 수록하지 않은 내용은 2회 실기 시험 대비용으로 제공하는 본 합격 전략집에 모두 정리해 놓았으니 꼭 학습하시기 바랍니다.

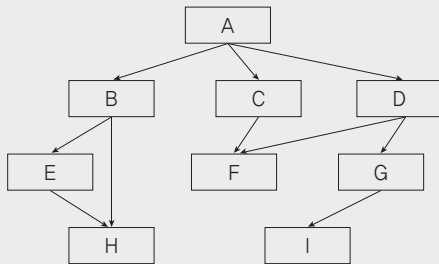
[교재 정보처리기사 실기 49섹션 1필드(1권 472쪽)]

[1회 실기 기출문제 5번] 소프트웨어 패키징이란 모듈별로 생성한 실행 파일들을 묶어 배포용 설치 파일을 만드는 것을 말한다. 소스 코드는 향후 관리를 고려하여 모듈화하여 패키징한다. 모듈화는 모듈 간 ( ① )의 최소화화 모듈 내 요소들의 ( ② )를 최대화 하는 것이 목표이다. 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오.

[정답] ① 결합도 ② 응집도

[교재 정보처리기사 실기 49섹션 4필드(1권 474쪽)]

[1회 실기 기출문제 16번] 다음의 모듈 관계를 표현한 시스템 구조도를 참고하여 팬인(Fan-In)이 2 이상인 모듈을 모두 쓰시오.



[정답] F, H

## 6장 화면 설계

UI의 종류, 설계 원칙, 도구, 품질 요구사항을 학습한 후, 관련 용어의 의미를 암기하세요.

1회 실기 시험에서는 한 문제도 출제되지 않았지만, 1회 필기 시험에서는 UI 설계 원칙이나 품질에 관한 문제가 출제되었습니다. 지난 1회 실기 시험에 이론적인 개념을 묻는 문제들이 출제되었던 것을 감안하면 실기 교재에 수록된 실무적인 내용 외에 개념적인 이론도 함께 학습할 필요가 있습니다. 용어의 개념이나 특징을 학습할 때는 세부적인 내용까지는 아니더라도 핵심적인 단어나 원리, 과정만큼은 확실히 기억해 두세요.



## 2020년 2회 정보처리기사 실기 시험, 이렇게 준비하세요.

### 7장 애플리케이션 테스트 관리

테스트들의 개념과 분류, 애플리케이션 성능까지 빠뜨리지 말고 모두 암기하세요.

출제 비중이 높지는 않지만 1회 기사 실기뿐만 아니라 1회 기사 필기, 1, 2회 기능사 실기 시험에서 모두 출제된 것으로 보아 이 부분은 향후에도 꾸준히 출제될 것을 예상할 수 있습니다. 성능 분석, 테스트 분류 및 분류별 종류, 결함 관리 등을 중점적으로 학습하세요.

[교재] 정보처리기사 실기 70색선 1필드(2권 7쪽)

[1회 실기 기출문제 8번] 애플리케이션 성능이란 사용자가 요구한 기능을 최소한의 자원을 사용하여 최대한 많은 기능을 신속하게 처리하는 정도를 나타낸다. 애플리케이션 성능 특성의 지표에 대한 설명에서 괄호(①~③)에 들어갈 알맞은 용어를 쓰시오.

( ① )	일정 시간 내에 애플리케이션이 처리하는 일의 양을 의미한다.
( ② )	애플리케이션에 요청을 전달한 시간부터 응답이 도착할 때까지 걸린 시간을 의미한다.
( ③ )	애플리케이션에 작업을 의뢰한 시간부터 처리가 완료될 때까지 걸린 시간을 의미한다.
자원 활용률	애플리케이션이 의뢰한 작업을 처리하는 동안의 CPU 사용량, 메모리 사용량, 네트워크 사용량 등 자원 사용률지 걸린 시간을 의미한다.

[정답] ① 처리량 ② 응답시간 ③ 경과시간

[교재] 정보처리기사 실기 60색선 3필드(2권 37쪽)

[1회 실기 기출문제 13번] 애플리케이션 테스트에서 살충제 패러독스(Pesticide Paradox)의 개념을 설명하시오.

[정답] 살충제 패러독스는 동일한 테스트 케이스로 반복 실행하면 더 이상 새로운 결함을 발견할 수 없는 현상이다.





## 8장 SQL 응용

DDL과 DCL은 문법을 확실하게 암기해야 하고, DML은 SQL문의 실행 결과까지 파악할 수 있어야 합니다. 1회 실기 시험에서는 DISTINCT를 활용한 SQL문의 실행 결과를 적는 문제가 출제되었습니다. 문제의 난이도는 높지 않았지만 2020년 2회 기능사 실기 시험에서 하위 질의나 Left Join 문제가 출제된 것을 감안할 때 향후 기사 실기 시험에서도 하위 질의나 Join 관련 문제들이 출제될 것을 충분히 예상할 수 있습니다. DDL과 DCL의 기본 문법을 확실히 암기했으면, DML에 집중해서 학습하세요. 2회 실기 시험 대비용으로 제공하는 본 합격 전략집에 Join과 관련된 다양한 예제를 수록했으니 잘 정리해 두세요.

[교재] 정보처리기사 실기 75섹션 1필드(2권 120쪽)

[1회 실기 기출문제 17번] 학생(STUDENT) 테이블에 전기과 학생이 50명, 전산과 학생이 100명, 전자과 학생이 50명 있다고 할 때, 다음 SQL 문 ①, ②, ③의 실행 결과로 표시되는 튜플의 수를 쓰시오.(단, DEPT 필드는 학과를 의미한다)

- ① SELECT DEPT FROM STUDENT;
- ② SELECT DISTINCT DEPT FROM STUDENT;
- ③ SELECT COUNT(DISTINCT DEPT) FROM STUDENT WHERE DEPT = '전산과';

[정답] ① 200 ② 3 ③ 1

## 9장 소프트웨어 개발 보안 구축

보안 약점의 발생 원인과 관련 용어들의 개념을 잘 정리하세요.

서비스 공격 유형, 암호 알고리즘이 1회 실기 시험에 출제되었습니다. 보안 관련 용어의 개념이나 공격 유형의 특징을 학습할 때는 세부적인 내용까지는 아니더라도 핵심적인 개념이나 특징만큼은 확실히 기억해둬야 합니다. 실기 교재에 수록된 내용 외에 추가적인 보안 관련 용어와 공격 유형들을 모두 정리하여 2회 실기 시험 대비용으로 제공하는 본 합격 전략집에 정리해 놓았으니 꼭 학습하시기 바랍니다.

[교재] 정보처리기사 실기 92섹션 4필드(2권 231쪽)

[1회 실기 기출문제 10번] 1991년 Rivest가 MD4를 개선한 암호화 알고리즘으로, 각각의 512 비트짜리 입력 메시지 블록에 대해 차례로 동작한다. 각 512 비트 입력 메시지 블록을 처리하고 나면 128 비트 스테이트(state)의 값이 변하는 암호화 알고리즘을 쓰시오.

[정답] MD5(Message Digest Algorithm 5)

※ MD5와 관련된 내용을 좀 더 보완하여 본 합격 전략집에 수록 했습니다.



## 2020년 2회 정보처리기사 실기 시험, 이렇게 준비하세요.

[교재] 정보처리기사 필기 189섹션 6필드(802쪽)

[1회 실기 기출문제 12번] 다음은 네트워크 공격에 대한 패킷 로그를 표현한 것이다. 아래의 패킷 로그와 같이 공격자가 패킷의 출발지 주소(Address) 또는 포트(Port)를 임의로 변경하여 송신측 IP 주소 또는 포트를 동일하게 함으로써 송신 IP 주소가 자신이므로 자신에게 응답을 수행하게 된다. 이처럼 자신에 대해 무한히 응답하는 패킷을 계속 전송하여 컴퓨터의 실행 속도를 느리게 하거나 동작을 마비시켜 서비스 거부 상태에 빠지도록 하는 네트워크 공격 유형이 무엇인지 쓰시오.

```
source : 192.168.1.200
destination : 192.168.1.200
protocol : 6
src port : 21845
dst port : 21845
```

[정답] LAND Attack

※ LAND Attack은 정보처리기사 필기 교재에 수록된 내용입니다. LAND Attack은 물론 관련된 내용을 모두 정리하여 본 합격 전략집에 수록 했습니다.

### 10장 응용 SW 기초 기술 활용

주요 용어의 개념과 구성요소, 특징, 계산 공식 등을 확실히 암기하세요.

1회 실기 시험에서 가장 출제 비중이 높았으며, 시험 범위가 변경되기 이전 내용들도 출제되어 혼란을 주었던 부분입니다. 대부분 용어의 의미나 구성요소, 특징, 계산 공식 등에 관련된 내용인데, 용어는 핵심적인 단어 위주로 기억해 두는 것이 효과적입니다. 추가로 나올만한 내용들은 2회 실기 시험 대비용으로 제공하는 본 합격 전략집에 모두 정리해 놓았으니 꼭 학습하시기 바랍니다.

[교재] 정보처리기사 실기 106섹션 1필드(2권 314쪽)

[1회 실기 기출문제 11번] 프로토콜은 서로 다른 기기들 간의 데이터 교환을 원활하게 수행할 수 있도록 표준화시켜 놓은 통신 규약이다. 프로토콜의 기본 요소 3가지를 쓰시오.

[정답] 구문(Syntax), 의미(Semantics), 시간(Timing)

[교재] 정보처리기사 실기 106섹션 1필드(2권 314쪽)

[1회 실기 기출문제 18번] 데이터 마이닝(Data Mining)의 개념을 설명하시오.

[정답] 데이터 마이닝은 데이터 웨어하우스에 저장된 데이터 집합에서 사용자의 요구에 따라 유용하고 가능성 있는 정보를 발견하기 위한 기법이다.



[교재] 2019 정보처리기사 필기 86색션 4필드(1권 521쪽)

[1회 실기 기출문제 19번] HRN 비선점형 스케줄링의 우선순위를 구하는 계산식을 쓰시오.

[정답] (대기 시간 + 서비스 시간) / 서비스 시간

※ HRN 스케줄링은 정보처리기사 필기 교재에 수록된 내용입니다. HRN 스케줄링 물론 관련 내용을 모두 정리하여 본 합격 전략집에 수록 했습니다.

[교재] 정보처리기사 실기 104색션 4필드(2권 304쪽)

[1회 실기 기출문제 15번] 다음이 설명하는 OSI 참조 모델의 계층을 쓰시오.

- 다양한 전송매체를 통해 비트 스트림을 전송한다.
- 전송에 필요한 두 장치 간의 실제 접속과 절단 등 기계적, 전기적, 기능적, 절차적 특성에 대한 규칙을 정의한다.

[정답] 물리 계층(Physical Layer)

## 11장 제품 소프트웨어 패키징

릴리즈 노트, 디지털 저작권 관리, 형상 관리에 집중하세요.

1회 실기 시험에 릴리즈 노트 작성 시 포함되는 항목 중 머릿말(Header)에 대한 문제가 출제되었고, 1회 필기 시험에 디지털 저작권 관리(DRM)와 형상(버전) 관리에 대한 문제가 출제되었습니다. 릴리즈 노트, DRM, 형상 관리의 개념과 요소들을 암기하고 형상 관리 방식에 따른 종류에는 어떤 것들이 있는지 확실히 기억해 두세요.

[교재] 정보처리기사 실기 109색션 2필드(2권 355쪽)

[1회 실기 기출문제 3번] 릴리즈 노트는 개발 과정에서 정리된 릴리즈 정보를 소프트웨어의 최종 사용자인 고객과 공유하기 위한 문서이다. 릴리즈 노트는 정확하고 완전한 정보를 기반으로 개발팀에서 직접 현재 시제로 작성해야 한다. 릴리즈 노트 작성 시 릴리즈 노트 이름, 소프트웨어 이름, 릴리즈 버전, 릴리즈 날짜, 릴리즈 노트 날짜, 릴리즈 노트 버전 등이 포함된 항목을 쓰시오.

[정답] 머릿말(Header)



m · e · m · o



# 핵심요약

1장 프로그래밍 언어 활용

2장 요구사항 확인

3장 데이터 입·출력 구현

4장 통합 구현

5장 서버 프로그램 구현

6장 화면 설계

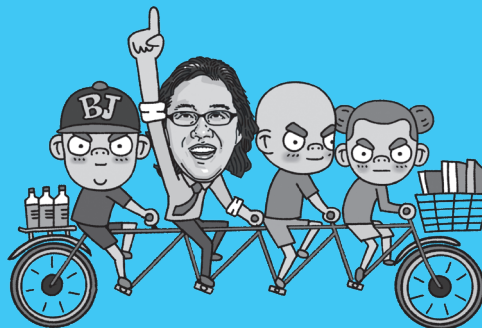
7장 애플리케이션 테스트 관리

8장 SQL 응용

9장 소프트웨어 개발 보안 구축

10장 응용 SW 기초 기술 활용

11장 제품 소프트웨어 패키징





# 정보처리기사 실기 핵심요약

## 1장 | 프로그래밍 언어 활용

### [핵심001] 데이터 타입(Data Type)

- 변수(Variable)에 저장될 데이터의 형식을 나타내는 것으로, 변수에 값을 저장하기 전에 문자형, 정수형, 실수형 등 어떤 형식의 값을 저장할지 데이터 타입을 지정하여 변수를 선언해야 한다.
- 데이터 타입의 유형

유형	기능	예
정수 타입 (Integer Type)	정수, 즉 소수점이 없는 숫자를 저장할 때 사용한다.	1, -1, 10, -100
부동 소수점 타입 (Floating Point Type)	소수점 이하가 있는 실수를 저장할 때 사용한다.	$0.123 \times 10^2$ , $-1.6 \times 2^3$
문자 타입 (Character Type)	<ul style="list-style-type: none"><li>한 문자를 저장할 때 사용한다.</li><li>작은따옴표(' ') 안에 표시한다.</li></ul>	'A', 'a', '1', '*'
문자열 타입 (Character String Type)	<ul style="list-style-type: none"><li>문자열을 저장할 때 사용한다.</li><li>큰따옴표(" ") 안에 표시한다.</li></ul>	"Hello!", "1+2=3"
불린 타입 (Boolean Type)	<ul style="list-style-type: none"><li>조건이 참(True), 거짓(False) 여부를 판단하여 저장할 때 사용한다.</li><li>기본값은 거짓(False)이다.</li></ul>	true, false
배열 타입 (Array Type)	<ul style="list-style-type: none"><li>같은 타입의 데이터 집합을 만들어 저장할 때 사용한다.</li><li>데이터는 중괄호({ }) 안에 콤마(,)로 구분하여 값들을 나열한다.</li></ul>	{1, 2, 3, 4, 5}

1. 다음 설명에 가장 부합하는 자료형(Data Type)을 쓰시오.

- 같은 타입의 데이터 집합을 만들어 저장할 때 사용한다.
- 데이터는 중괄호({ }) 안에 콤마(,)로 구분하여 값들을 나열한다.

답 :

정답 1. 배열(Array)

### [핵심002] 변수와 기억 클래스

#### 변수(Variable)

- 컴퓨터가 명령을 처리하는 도중 발생하는 값을 저장하기 위한 공간으로, 변할 수 있는 값을 의미한다.
- 변수는 저장하는 값에 따라 정수형, 실수형, 문자형, 포인터형 등으로 구분한다.



## 기억 클래스

- 변수 선언 시 메모리 내에 변수의 값을 저장하기 위한 기억영역이 할당되는데, 할당되는 기억영역에 따라 사용 범위에 제한이 있다. 이러한 기억영역을 결정하는 작업을 기억 클래스(Storage Class)라 한다.
- C언어에서 제공하는 기억 클래스의 종류

자동 변수(Automatic Variable)	함수나 코드의 범위를 한정하는 블록 내에서 선언되는 변수이다.
외부 변수(External Variable)	현재 파일이나 다른 파일에서 선언된 변수나 함수를 참조(reference)하기 위한 변수이다.
정적 변수(Static Variable)	함수나 블록 내에서 선언하는 내부 정적 변수와 함수 외부에서 선언하는 외부 정적 변수가 있다.
레지스터 변수(Register Variable)	메모리가 아닌 CPU 내부의 레지스터에 기억영역을 할당받는 변수이다.

### 1. 다음이 설명하는 기억 클래스를 쓰시오.

- 함수나 블록 내에서 선언하는 변수와 함수 외부에서 선언하는 변수가 있다.
- 함수나 블록이 종료된 뒤에도 값이 소멸되지 않는다.
- 초기화는 변수 선언 시 한 번만 할 수 있으며, 초기화를 생략하면 자동으로 0으로 초기화 된다.

답 :

**정답** 1. 정적 변수(Static Variable)

## [핵심003] scanf( ) 함수 / printf( ) 함수

### scanf( ) 함수

- C언어의 표준 입력 함수로, 키보드로 입력받아 변수에 저장하는 함수이다.
- 형식

#### scanf(서식 문자열, 변수의 주소)

- 서식 문자열 : 입력받을 데이터의 자료형을 지정한다.
- 변수의 주소 : 데이터를 입력받을 변수를 적는다. 변수의 주소로 입력받아야 하기 때문에 변수에 주소연산자 &를 붙인다.

예) `scanf("%3d", &a);`

- ▶ % : 서식 문자임을 지정
- ▶ 3 : 입력 자릿수를 3자리로 지정
- ▶ d : 10진수로 입력
- ▶ &a : 입력받은 데이터를 변수 a의 주소에 저장

- 서식 문자열 : printf( ) 함수로 출력할 때도 동일하게 적용됨

서식 문자열	의미
%d	정수형 10진수를 입 · 출력하기 위해 지정한다.

%u	부호없는 정수형 10진수를 입 · 출력하기 위해 지정한다.
%o	정수형 8진수를 입 · 출력하기 위해 지정한다.
%x	정수형 16진수를 입 · 출력하기 위해 지정한다.
%c	문자를 입 · 출력하기 위해 지정한다.
%s	문자열을 입 · 출력하기 위해 지정한다.
%f	소수점을 포함하는 실수를 입 · 출력하기 위해 지정한다.
%e	지수형 실수를 입 · 출력하기 위해 지정한다.
%ld	long형 10진수를 입 · 출력하기 위해 지정한다.
%lo	long형 8진수를 입 · 출력하기 위해 지정한다.
%lx	long형 16진수를 입 · 출력하기 위해 지정한다.
%p	주소를 16진수로 입 · 출력하기 위해 지정한다.

### printf ( ) 함수

- C언어의 표준 출력 함수로, 인수로 주어진 값을 화면에 출력하는 함수이다.
- 형식

#### printf(서식 문자열, 변수)

- 서식 문자열 : 변수의 자료형에 맞는 서식 문자열을 입력한다.
- 변수 : 서식 문자열의 순서에 맞게 출력할 변수를 적는다. scanf( )와 달리 주소 연산자 &를 붙이지 않는다.

예) printf("%-8.2f", 200.2); → 200.20

- ▶ % : 서식 문자임을 지정
- ▶ - : 왼쪽부터 출력
- ▶ 8 : 출력 자릿수를 8자리로 지정
- ▶ 2 : 소수점 이하를 2자리로 지정
- ▶ f : 실수로 출력

- 주요 제어문자 : 입력 혹은 출력 내용을 제어하는 문자

문자	의미	기능	문자	의미	기능
\n	new line	커서를 다음 줄 앞으로 이동한다.	\'	single quote	작은따옴표를 출력한다.
\b	backspace	커서를 왼쪽으로 한 칸 이동한다.	\"	double quote	큰따옴표를 출력한다.
\t	tab	커서를 일정 간격 띄운다.	\a	alert	스피커로 벨 소리를 출력한다.
\r	carriage return	커서를 현재 줄의 처음으로 이동한다.	\\	backslash	역 슬래시를 출력한다.
\0	null	널 문자를 출력한다.	\f	form feed	한 페이지를 넘긴다.

1. 다음 C언어의 <코드>와 <입력>을 보고 프로그램을 분석하여 그 실행 결과를 쓰시오.

<코드>

```
#include <stdio.h>
main() {
    char input_Value[50];
    scanf("%s", input_Value);
    printf("%.5s", input_Value);
    printf("%5.3s", input_Value);
}
```

<입력>

Sinagong.gilbut.co.kr Connect!

답:

```
#include <stdio.h>
main() {
{
    char input_Value[50];
    scanf("%s", input_Value);
    printf("%.5s", input_Value);
    printf("%5.3s", input_Value);
}
```

50자리 크기의 문자 배열 input\_Value를 선언한다.  
 입력한 데이터를 input\_Value에 저장한다.  
 input\_Value에 저장된 문자열을 5글자만 출력한다.  
 결과 Sinag  
 전체 5자리를 확보한 후 input\_Value에 저장된 문자열 중 앞에 3글자만을 오른쪽부터 출력한다.  
 결과 Sinag Sin

2. 다음 C언어의 <코드>와 <입력>을 보고 프로그램을 분석하여 그 실행 결과를 쓰시오.

<코드>

```
#include <stdio.h>
main() {
    int i, j;
    scanf("%o%lx", &i, &j);
    printf("%d %d", i, j);
}
```

<입력>

20@20

답:

```
#include <stdio.h>
```

```
main() {
```

```
    int i, j;
```

```
    scanf("%o%x", &i, &j);
```

'@'를 기준으로 입력값을 구분하여 앞에 입력된 수는 정수형 8진수로, 뒤에 입력된 수는 정수형 16진수로 각각 변수 i와 j에 저장한다.

```
    printf("%d %d", i, j);
```

8진수 20을 10진수로 변환하여 출력한 다음 공백을 한 칸 띄우고, 16진수 20을 10진수로 변환하여 출력한다.

결과 16 32

```
}
```

**정답** 1. Sinag Sin 2. 16 32

### [핵심004] Java에서의 표준 입 · 출력

#### Java에서의 표준 입력

- Java에서 키보드로 입력받은 값을 변수에 저장하려면 먼저 Scanner 클래스를 이용해 키보드로부터 값을 입력받는 객체 변수를 생성한 후 이를 사용해야 한다.
- 형식

```
① Scanner scan01 = new Scanner(System.in);
② inNum = scan01.nextInt();
```

#### ① 객체 변수 생성

- Scanner : 입력에 사용할 객체 변수를 생성할 때 사용하는 클래스 이름이다. 그대로 적어준다.
- scan01 : 객체 변수명이다. 사용자 임의로 적어준다.
- new : 객체 생성 예약어이다. 그대로 적어준다.
- Scanner() : 클래스의 이름이다. ()를 붙여 그대로 적어준다.
- System.in : 표준 입력장치, 즉 키보드를 의미한다. 키보드로부터 값을 입력받는 객체 변수를 생성할 것이므로 그대로 적어준다.

#### ② 객체 변수 활용

- inNum : 입력받은 값을 저장할 변수이다. 이 변수는 미리 선언되어 있어야 한다.
- scan01.nextInt()
  - ▶ scan01 : 입력에 사용할 객체 변수 이름이다. 객체 변수 생성 시 사용한 객체 변수 이름과 동일해야 한다.
  - ▶ nextInt() : 입력받은 값을 정수형으로 반환한다.

#### 참관만요 ① Scanner 클래스의 입력 메소드

- next() : 입력값을 문자열로 반환
- nextLine() : 입력받은 라인 전체를 문자열로 반환
- nextInt() : 입력값을 정수형으로 반환
- nextFloat() : 입력값을 실수형으로 반환

## Java에서의 표준 출력

- Java에서 값을 화면에 출력할 때는 `printf()` 메소드를 이용한다.
- 형식

`System.out.printf(서식 문자열, 변수)`

- `System.out.printf()` : `System` 클래스의 서브 클래스인 `out` 클래스의 메소드 `printf()`를 사용해서 출력한다는 의미이다. `printf()` 메소드는 C언어의 `printf()` 함수와 사용법이 동일하다.

예) `System.out.printf("%-8.2f", 200.2);` → 

2	0	0	.	2	0		
---	---	---	---	---	---	--	--

- ▶ % : 서식 문자임을 지정
- ▶ - : 왼쪽부터 출력
- ▶ 8 : 출력 자릿수를 8자리로 지정
- ▶ 2 : 소수점 이하를 2자리로 지정
- ▶ f : 실수로 출력

1. 다음은 2개의 정수를 입력받아 합을 출력하는 Java 프로그램이다. 괄호에 공통으로 들어갈 가장 적합한 코드를 쓰시오.

```
import java.util.(      );
public class Test {
    public static void main(String args[]) {
        (      ) scan = new (      )(System.in);
        int a = scan.nextInt();
        int b = scan.nextInt();
        System.out.printf("%d", a + b);
    }
}
```

답 :

정답 1. Scanner

## [핵심005] 연산자

### • 산술 연산자

연산자	의미
+	덧셈
-	뺄셈
*	곱셈
/	나눗셈
%	나머지
++	증가 연산자
--	감소 연산자

### • 관계 연산자

연산자	의미
==	같다
!=	같지 않다
>	크다
>=	크거나 같다
<	작다
<=	작거나 같다

### • 논리 연산자

연산자	의미	비고
!	not	부정
&&	and	모두 참이면 참
	or	하나라도 참이면 참

※ 증가/감소 연산자는 변수의 앞(전치) 또는 변수의 뒤(후치)에 붙여 사용한다.

- 전치 : 변수 앞에 증감 연산자가 오는 형태로 먼저 변수의 값을 증감시킨 후 변수를 연산에 사용한다(++a, --a).
- 후치 : 변수 뒤에 증감 연산자가 오는 형태로 먼저 변수를 연산에 사용한 후 변수의 값을 증감시킨다(a++, a--).

#### • 비트 연산자

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

1. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오.

```
#include <stdio.h>
main() {
    int a = 4, b = 3, c = 5, d = 7;
    int r1, r2, r3, r4;
    r1 = 10 % a++;
    r2 = b > 3 && b > 2;
    r3 = c & d;
    r4 = d << 3;
    printf("%d, %d, %d, %d", r1, r2, r3, r4);
}
```

답:

```
#include <stdio.h>
main() {
    int a = 4, b = 3, c = 5, d = 7;
    int r1, r2, r3, r4;
    ① r1 = 10 % a++;
    ② r2 = b > 3 && b > 2;
    ③ r3 = c & d;
    ④ r4 = d << 3;
    ⑤ printf("%d, %d, %d, %d", r1, r2, r3, r4);
}
```



[핵심 006] 연산자2

• 대입 연산자

연산자	예	의미
<code>+=</code>	<code>a += 1</code>	<code>a = a + 1</code>
<code>-=</code>	<code>a -= 1</code>	<code>a = a - 1</code>
<code>*=</code>	<code>a *= 1</code>	<code>a = a * 1</code>
<code>/=</code>	<code>a /= 1</code>	<code>a = a / 1</code>
<code>%=</code>	<code>a %= 1</code>	<code>a = a % 1</code>
<code>&lt;&lt;=</code>	<code>a &lt;&lt;= 1</code>	<code>a = a &lt;&lt; 1</code>
<code>&gt;&gt;=</code>	<code>a &gt;&gt;= 1</code>	<code>a = a &gt;&gt; 1</code>

• 조건 연산자

형식

**조건 ? 수식1 : 수식2;**

'조건'의 수식이 참이면 '수식1'을, 거짓이면 '수식2'를 실행한다.

• 기타 연산자

연산자	의미
<code>sizeof</code>	자료형의 크기를 표시한다.
<code>,</code> (콤마)	<ul style="list-style-type: none"> <li>• 콤마로 구분하여 한 줄에 두 개 이상의 수식을 작성하거나 변수를 정의한다.</li> <li>• 왼쪽에서 오른쪽으로 순서대로 수행되며, 순서 연산자라 부르기도 한다.</li> </ul>
(자료형)	<ul style="list-style-type: none"> <li>• 사용자가 자료형을 다른 자료형으로 변환할 때 사용하는 것으로, <code>cast</code>(캐스트) 연산자라고 부른다.</li> <li>• 변환할 자료형을 괄호 안에 넣어서 변환할 값이나 변수명 앞에 놓는다.  <b>예</b> <code>a = (int)1.3 + (int)1.4;</code>  1.3을 정수형으로 변환한 값 1과 1.4를 정수형으로 변환한 값 1이 더해진 2가 a에 저장된다. </li> </ul>



1. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오.

```
#include <stdio.h>
main() {
    int a = 2, b = 5, p = 10, q = 7;
    b -= a--;
    p %= a < b ? a++ : b++;
    q /= b % 3 ? a * b : b % a;
    printf("%d %d", p, q);
}
```

답 :

```
#include <stdio.h>
main() {
    int a = 2, b = 5, p = 10, q = 7;
    ① b -= a--;
    ② p %= a < b ? a++ : b++;
    ③ q /= b % 3 ? a * b : b % a;
    ④ printf("%d %d", p, q);
}
```

① 'b = b - a--;'와 같다. a--는 후치 연산이므로 연산에 사용되는 a는 2다. b에는 5 - 2의 결과인 3이 저장되고, a는 1이 된다.

② p %= a < b ? a++ : b++;

㉠  
㉡

- ㉠: a는 1이고 b는 3이므로 조건(a < b)이 참이 되어 a++의 결과인 2가 사용된다.
- ㉡: p = p % 1 = 10 % 1
- 10을 1로 나눈 나머지는 0이므로 p에는 0이 저장되고, 후치 연산으로 a는 3이 된다.

③ q /= b % 3 ? a \* b : b % a;

㉢  
㉣

- ㉢: b는 3이므로 '3 % 3'은 0이 된다. 조건에서 0은 거짓과 같으므로 'b % a'의 결과인 2가 사용된다.
- ㉣: q = q / 1 = 7

④ p와 q의 값을 화면에 출력한다.

정답 1.07

### [핵심007] 연산자 우선순위

- 한 개의 수식에 여러 개의 연산자가 사용되면 기본적으로 다음 표의 순서대로 처리된다.
- 다음 표의 한 줄에 가로로 나열된 연산자는 우선순위가 같기 때문에 결합규칙에 따라 ←는 오른쪽에 있는 연산자부터, →는 왼쪽에 있는 연산자부터 차례로 계산된다.

대분류	중분류	연산자	결합규칙	우선 순위
단항 연산자	단항 연산자	!(논리 not) ~ (비트 not) ++ (증가) -- (감소) sizeof (기타)	←	높음 
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	조건 연산자	? :	→	
대입 연산자	대입 연산자	= += -= *= /= %= <<= >>= 등	←	
순서 연산자	순서 연산자	.(콤마)	→	

1. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오.

```
#include <stdio.h>
main() {
    int c = 3, d = 6, e = 3, x, y;
    x = c > 1 || d != 0;
    y = d <= 4 && e > 1;
    printf("%d, %d", x, y);
}
```

답 :

```
#include <stdio.h>
main() {
    int c = 3, d = 6, e = 3, x, y;
    ❶ x = c > 1 || d != 0;
    ❷ y = d <= 4 && e > 1;
    ❸ printf("%d, %d", x, y);
}
```

- ❶ 관계 연산자와 논리 연산자가 있으면 관계 연산자를 먼저 수행한다. 'c > 1'은 참이고 'd != 0'은 참이므로 1(참) || 1(참) = 1(참)이 되어 x에 1이 저장된다.
- ❷ 'd <= 4'는 거짓이고 'e > 1'은 참이므로 0(거짓) && 1(참) = 0(거짓)이 되어 y에 0이 저장된다.
- ❸ x와 y의 값을 출력한다.

## 2. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오.

```
#include <stdio.h>
main() {
    int a = 3, b = 4, c = 5, d = 6, p, q;
    p = a * b + c >= d && d / a - b != 0;
    q = d % b + ++a * c-- || c - --a >= 10;
    printf("%d", p && q);
}
```

답 :

```
#include <stdio.h>
main() {
    int a = 3, b = 4, c = 5, d = 6, p, q;
    ❶ p = a * b + c >= d && d / a - b != 0;
    ❷ q = d % b + ++a * c-- || c - --a >= 10;
    ❸ printf("%d", p && q);
}
```

❶  $p = a * b + c >= d \ \&\& \ d / a - b \neq 0;$

$$\begin{array}{rcl} \textcircled{1}(12) & & \textcircled{2}(2) \\ \hline \textcircled{3}(17) & & \textcircled{4}(-2) \\ \hline \textcircled{5}(1) & & \textcircled{6}(1) \\ \hline & & \textcircled{7}(1) \end{array}$$

❷  $q = d \% b + ++a * c-- \ || \ c - --a >= 10;$

$$\begin{array}{rcl} \textcircled{4}(2) & \textcircled{3}(3) & \textcircled{2}(5) & \textcircled{1}(3) \\ \hline & \textcircled{5}(15) & & \\ \hline \textcircled{6}(17) & & \textcircled{7}(2) & \\ \hline & & \textcircled{8}(0) & \\ \hline & & & \textcircled{9}(1) \end{array}$$

※ ❷의 ①은 --a에 의해 처음에는 2를 갖지만 ③의 전치 증가 연산이 적용되어 계산에 사용될 때는 3이 된다.

❸ p와 q를 &&(논리 and) 연산한 결과 1을 출력한다.

정답 1. 1, 0 2. 1

**[핵심008] if문****단순 if문**

- if문은 조건에 따라서 실행할 문장을 달리하는 제어문이며, 단순 if문은 조건이 한 개 일 때 사용하는 제어문이다.
- 조건이 참일 때만 실행할 문장을 지정할 수도 있고, 참과 거짓에 대해 각각 다른 실행문을 지정할 수도 있다.
- 형식1 : 조건이 참일 때만 실행한다.
  - 조건이 참일 때 실행할 문장이 하나인 경우

**if(조건)**

if는 조건 판단문에 사용되는 예약어이므로 그대로 적는다.  
조건은 참(1) 또는 거짓(0)이 결과로 나올 수 있는 수식을 ( ) 안에 입력한다.

**실행할 문장;**

조건이 참일 경우 실행할 문장을 적는다.

- 조건이 참일 때 실행할 문장이 두 문장 이상인 경우

```
{
    실행할 문장1;
    실행할 문장2;
    :
}
```

{ } 사이에 조건이 참일 경우 실행할 문장을 적는다.

- 형식2 : 조건이 참일 때와 거짓일 때 실행할 문장이 다르다.

**if(조건)****실행할 문장1;**

조건이 참일 경우 실행할 문장을 적는다. 참일 경우 실행할 문장이 두 문장 이상이면 { }를 입력하고 그 사이에 문장을 적는다.

**else****실행할 문장2;**

조건이 거짓일 경우 실행할 문장을 적는다. 두 문장 이상인 경우 { }를 입력하고 그 사이에 문장을 적는다.

**다중 if문**

- 조건이 여러 개 일 때 사용하는 제어문이다.
- 형식1

**if(조건1)****실행할 문장1;**

조건1이 참일 경우 실행할 문장을 적는다.

**else if(조건2)****실행할 문장2;**

조건2가 참일 경우 실행할 문장을 적는다.

**else if(조건3)****실행할 문장3;**

조건3이 참일 경우 실행할 문장을 적는다.

:

**else****실행할 문장4;**

앞의 조건이 모두 거짓일 경우 실행할 문장을 적는다.

- 형식2 : if문 안에 if문이 포함된다.

```

if(조건1)
{
    if(조건2)
        실행할 문장1;    조건2가 참일 경우 실행할 문장을 적는다.
    else
        실행할 문장2;    조건2가 거짓일 경우 실행할 문장을 적는다.
}
else
    실행할 문장3;    조건1이 거짓일 경우 실행할 문장을 적는다.

```

1. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오.

```

#include <stdio.h>
main() {
    int a = 15, b = 22;
    if (a % 2 == 0)
        if (b % 2 == 0)
            printf("A");
        else
            printf("B");
    else if (b % 2 == 0)
        printf("C");
    else
        printf("D");
}

```

답 :

```

#include <stdio.h>
main() {
    int a = 15, b = 22;
    ① if (a % 2 == 0)
    ②     if (b % 2 == 0)
    ③         printf("A");
    ④     else
    ⑤         printf("B");
    ⑥ else if (b % 2 == 0)
    ⑦     printf("C");
    ⑧ else
    ⑨     printf("D");
    } ⑩

```

- ① a를 2로 나눈 나머지가 0이면 ②번을 실행하고, 아니면 ⑥번으로 이동한다.
- ② b를 2로 나눈 나머지가 0이면 ③번을 실행하고, 아니면 ④번으로 이동한다.
- ③ "A"를 출력하고, ⑩번으로 이동하여 프로그램을 종료한다.
- ④ ②번의 조건식이 거짓일 경우 ⑤번을 실행한다.
- ⑤ "B"를 출력하고, ⑩번으로 이동하여 프로그램을 종료한다.
- ⑥ b를 2로 나눈 나머지가 0이면 ⑦번을 실행하고, 아니면 ⑧번으로 이동한다.
- ⑦ "C"를 출력하고, ⑩번으로 이동하여 프로그램을 종료한다.
- ⑧ ⑥번의 조건식이 거짓일 경우 ⑨번을 실행한다.
- ⑨ "D"를 출력하고, ⑩번으로 이동하여 프로그램을 종료한다.

정답 1. C

## [핵심009] switch문

- 조건에 따라 분기할 곳이 여러 곳인 경우 간단하게 처리할 수 있는 제어문이다.
- 형식

## switch(수식) ①

{ ②

case 레이블1: ③

실행할 문장1;

break;

case 레이블2: ④

실행할 문장2;

break;

⋮

default:

실행할 문장3;

} ⑤

- switch는 switch문에 사용되는 예약어로 그대로 입력한다.
- 수식 : '레이블1' ~ '레이블n'의 값 중 하나를 도출하는 변수나 수식을 입력한다.
- ②~⑤번이 switch문의 범위이다. '{'로 시작해서 '}'로 끝난다. 반드시 입력해야 한다.
- case는 switch문에서 레이블을 지정하기 위한 예약어로 그대로 입력해야 한다.
- 레이블1 : ①번 식의 결과가 될 만한 값 중 하나를 입력한다. 결과가 '레이블1'과 일치하면 이곳으로 찾아온다. 식의 결과가 5종류로 나타나면 case문이 5번 나와야 한다.
- ①번 식의 결과가 ③번의 '레이블1'과 일치할 때 실행할 문장이다.
- switch문을 탈출하여 ⑤번으로 간다.
- ①번의 식의 결과가 '레이블2'와 일치하면 찾아오는 곳이다.
- ①번의 식의 결과가 ④번의 '레이블2'와 일치할 때 실행할 문장이다.
- switch문을 탈출하여 ⑤번으로 간다.
- ①번의 식의 결과가 '레이블1' ~ '레이블n'에 해당하지 않는 경우 찾아오는 곳이다.

- case문의 레이블에는 한 개의 상수만 지정할 수 있으며, int, char, enum형의 상수만 가능하다.
- case문의 레이블에는 변수를 지정할 수 없다.
- break문은 생략이 가능하지만 break문이 생략되면 수식과 레이블이 일치할 때 실행할 문장부터 break문 또는 switch문이 종료될 때까지 모든 문장이 실행된다.

2020년 1회 기사 실기

1. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오.

```
#include <stdio.h>
main() {
    int c = 1;
    switch (3) {
        case 1: c += 3;
        case 2: c++;
        case 3: c = 0;
        case 4: c += 3;
        case 5: c -= 10;
        default: c--;
    }
    printf("%d", c);
}
```

답 :

```
#include <stdio.h>
main() {
    ① int c = 1;
    ② switch (3) {
        case 1: c += 3;
        case 2: c++;
    ③ case 3: c = 0;
    ④ case 4: c += 3;
    ⑤ case 5: c -= 10;
    ⑥ default: c--;
    }
    ⑦ printf("%d", c);
}
```

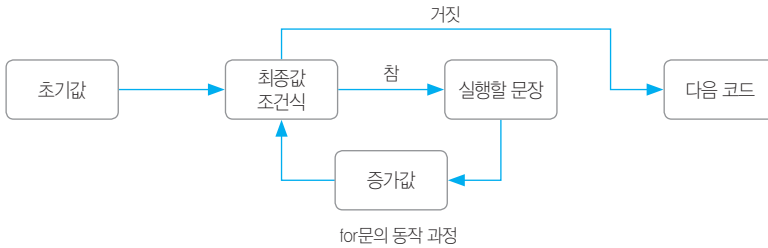
모든 case문에 break문이 생략되었으므로, switch문의 인수와 일치하는 'case 3' 문장부터 switch문이 종료될 때까지 모든 문장이 실행된다.

- ① 정수형 변수 c를 선언하고 1로 초기화한다. →  $c=1$
- ② 3에 해당하는 숫자를 찾아간다. 'case 3' 문장으로 이동한다.
- ③ c의 값을 0으로 치환한다. →  $c=0$
- ④ ' $c=c+3$ '과 동일하다. c의 값에 3을 더한다. →  $c=3$
- ⑤ ' $c=c-10$ '과 동일하다. c의 값에서 10을 뺀다. →  $c=-7$
- ⑥ ' $c=c-1$ '과 동일하다. c의 값에서 1을 뺀다. →  $c=-8$
- ⑦ c의 값을 출력한다.

정답 1. -8

**[핵심010] for문**

- 초기값, 최종값, 증가값을 지정하는 수식을 이용해 정해진 횟수를 반복하는 제어문이다.
- 초기값을 정한 다음 최종값에 대한 조건이 참이면 실행할 문장을 실행한 후 초기값을 증가값 만큼 증가시키면서 최종값에 대한 조건이 참인 동안 실행할 문장을 반복 수행한다.



- 형식

**for(식1; 식2; 식3)**

- for는 반복문을 의미하는 예약어로 그대로 입력한다.
- 식1 : 초기값을 지정할 수식을 입력한다.
- 식2 : 최종값을 지정할 수식을 입력한다.
- 식3 : 증가값으로 사용할 수식을 입력한다.

**실행할 문장;**

식2가 참일 동안 실행할 문장을 입력한다. 실행할 문장이 두 문장 이상일 경우 { }를 입력하고 그 사이에 처리할 문장들을 입력한다.

- 초기값, 최종값, 증가값 중 하나 이상을 생략할 수 있고, 각각의 요소에 여러 개의 수식을 지정할 수도 있다.
- for문은 처음부터 최종값에 대한 조건식을 만족하지 못하면 한 번도 수행하지 않는다.

2020년 2회 가능사 실기

1. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오.

```

#include <stdio.h>
main() {
    int i, j;
    for (i = 2; i <= 4; i++) {
        for (j = 5; j <= 7; j++) {
            printf("%d × %d = %d", j, i, i * j);
        }
    }
}
  
```

답:



```
#include <stdio.h>
main() {
    int i, j;
    ❶ for (i = 2; i <= 4; i++) {
    ❷         for (j = 5; j <= 7; j++) {
    ❸             }
    ❹     }
    ❺ printf("%d × %d = %d", j, i, i * j);
}
```

- ❶ 반복 변수 `i`가 2에서 시작하여 1씩 증가하면서 4보다 작거나 같은 동안 ❷번 문장을 반복하여 수행한다. 결국 ❷번을 3번 반복하여 수행한다.
- ❷ 반복 변수 `j`가 5에서 시작하여 1씩 증가하면서 7보다 작거나 같은 동안 반복하는데 실행할 문장 없이 for문만 반복 수행한다.
- ❸ ❷번 반복문의 끝이다.
- ❹ ❶번 반복문의 끝이다.
- ❺ "`j × i = j * i`"의 형태로 변수의 값을 출력한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

i	j
2	5
	6
	7
	8
3	5
	6
	7
	8
4	5
	6
	7
	8
5	

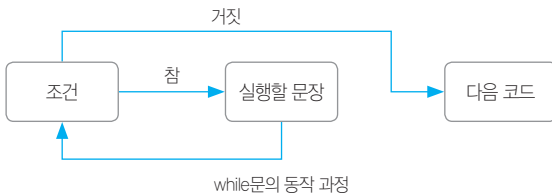
**정답** 1.  $8 \times 5 = 40$

## [핵심011] 기타 반복문

### while문

- 조건이 참인 동안 실행할 문장을 반복 수행하는 제어문이다.
- while문은 조건이 참인동안 실행할 문장을 반복 수행하다가 조건이 거짓이면 while문을 끝낸 후 다음 코드를 실행한다.

- while문은 조건이 처음부터 거짓이면 한 번도 수행하지 않는다.



- 형식

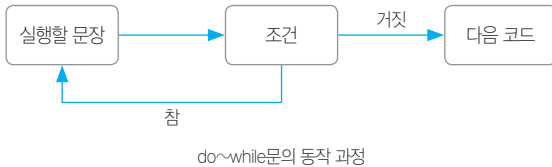
### while(조건)

#### 실행할 문장;

- while은 반복문에 사용되는 예약어로 그대로 입력한다.
- (조건) : 참이나 거짓을 결과로 갖는 수식을 '조건'에 입력한다. 참(!)을 직접 입력할 수도 있다.
- 조건이 참인 동안 실행할 문장을 입력한다. 문장이 두 문장 이상인 경우 { }를 입력하고 그 사이에 처리할 문장들을 입력한다.

### do~while문

- 조건이 참인 동안 정해진 문장을 반복 수행하다가 조건이 거짓이면 반복문을 벗어나는 while문과 같은 동작을 하는데, 다른 점은 do~while문은 실행할 문장을 무조건 한 번 실행한 다음 조건을 판단하여 탈출 여부를 결정한다는 것이다.
- do~while문은 실행할 문장을 우선 실행한 후 조건을 판별하여 조건이 참이면 실행할 문장을 계속 반복 수행하고, 거짓이면 do~while문을 끝낸 후 다음 코드를 실행한다.



- 형식

### do

#### 실행할 문장;

### while(조건);

- do는 do~while문에 사용되는 예약어로, do~while의 시작 부분에 그대로 입력한다.
- 조건이 참인 동안 실행할 문장을 입력한다. 문장이 두 문장 이상인 경우 { }를 입력하고 그 사이에 실행할 문장들을 입력한다.
- while은 do~while문에 사용되는 예약어로, do~while의 끝 부분에 그대로 입력한다.
- (조건) : 참이나 거짓을 결과로 갖는 수식을 '조건'에 입력한다. 참(!)을 직접 입력할 수도 있다.

### break, continue

- switch문이나 반복문의 실행을 제어하기 위해 사용되는 예약어이다.
- break : switch문이나 반복문 안에서 break가 나오면 블록을 벗어난다.
- continue
  - continue 이후의 문장을 실행하지 않고 제어를 반복문의 처음으로 옮긴다.
  - 반복문에서만 사용된다.

1. 다음 C언어로 구현된 프로그램을 실행했을 때 최종적으로 출력되는 결과를 쓰시오.

```
#include <stdio.h>
main() {
    int x = 100, y = 10;
    do {
        x -= y;
        printf("%d, %d\n", x, y++);
    } while (x > 30);
}
```

답 :

```
#include <stdio.h>
main() {
    ① int x = 100, y = 10;
    ② do {
    ③     x -= y;
    ④     printf("%d, %d\n", x, y++);
    ⑤ } while (x > 30);
}
```

- ① 정수형 변수 x와 y를 선언한 후 각각 100과 10으로 초기화한다.
- ② do~while 반복문의 시작점이다. ③, ④번 문장을 반복하여 수행한다.
- ③ 'x = x - y;'와 동일하다. x의 값에서 y의 값을 뺀 후 x에 저장한다.
- ④ x와 y를 정수형 10진수로 출력한 후 커서를 다음 줄 맨 처음으로 이동한다. y++로 인해 ④번 문장 종료 후 y의 값이 1 증가한다.
- ⑤ x가 30보다 큰 동안 ③, ④번 문장을 반복하여 수행한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

X	Y
100	10
90	11
79	12
67	13
54	14
40	15
25	16

정답 1. 25, 15

## [핵심012] 배열과 문자열

### 배열의 개념

- 동일한 데이터 유형을 여러 개 사용해야 할 경우 이를 손쉽게 처리하기 위해 여러 개의 변수들을 조합해서 하나의 이름으로 정의해 사용하는 것을 말한다.

- 배열은 변수명 뒤에 대괄호 [ ]를 붙이고 그 안에 사용할 개수를 지정한다.
- C언어에서 배열의 위치는 0부터 시작된다.
- C언어에서 배열 위치를 나타내는 첨자 없이 배열 이름을 사용하면 배열의 첫 번째 요소의 주소를 지정하는 것과 같다.

## 1차원 배열

- 변수들을 일직선상의 개념으로 조합한 배열이다.
- 형식

### 자료형 변수명[개수];

- 자료형 : 배열에 저장할 자료의 형을 지정한다.
- 변수명 : 사용할 배열의 이름으로 사용자가 임의로 지정한다.
- 개수 : 배열의 크기를 지정하는 것으로 생략할 수 있다.

예 int a[5] : 5개의 요소를 갖는 정수형 배열 a

	첫 번째	두 번째	세 번째	네 번째	다섯 번째
배열 a	a[0]	a[1]	a[2]	a[3]	a[4]

※ a[3] : a는 배열의 이름이고, 3은 첨자로서 배열 a에서의 위치를 나타낸다.

a[3]에 4를 저장시키려면 'a[3] = 4'와 같이 작성한다.

## 2차원 배열

- 변수들을 평면, 즉 행과 열로 조합한 배열이다.
- 형식

### 자료형 변수명[행개수][열개수]

- 자료형 : 배열에 저장할 자료의 형을 지정한다.
- 변수명 : 사용할 배열의 이름으로 사용자가 임의로 지정한다.
- 행개수 : 배열의 행 크기를 지정한다.
- 열개수 : 배열의 열 크기를 지정한다.

예 int b[3][3] : 3개의 행과 3개의 열을 갖는 정수형 배열 b

	열			
	0, 0	0, 1	0, 2	b[0][2]
행	1, 0	1, 1	1, 2	
	2, 0	2, 1	2, 2	

※ b[0][2] : b는 배열의 이름이고, 0은 행 첨자, 2는 열 첨자로서 배열 b에서의 위치를 나타낸다.

## 배열 형태의 문자열 변수

- C언어에서는 큰따옴표(" ")로 묶인 글자는 글자 수에 관계없이 문자열로 처리된다.
- C언어에는 문자열을 저장하는 자료형이 없기 때문에 배열 또는 포인터를 이용하여 처리한다.

## • 형식

```
char 배열이름[크기] = "문자열"
```

- 배열에 문자열을 저장할 때는 배열 선언 시 초기값으로 지정해야 하며, 이미 선언된 배열에는 문자열을 저장할 수 없다.
- 문자열 끝에 자동으로 널 문자(\\0)가 삽입되므로, 널 문자까지 고려하여 배열 크기를 지정해야 한다.

예 char a[5] = "love" → 

l	o	v	e	\\0
---	---	---	---	-----

1. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오.

```
#include <stdio.h>
main() {
    int exint[ ] = { 4,9,3,7,6,4,24,4,8,4,10 };
    int len = sizeof(exint) / sizeof(int);
    int value = 0;
    for (int i = 0; i < len; i++) {
        if (exint[i] == 4) {
            value++;
        }
    }
    printf("%d", value);
}
```

답 :

```
#include <stdio.h>
main() {
    ① int exint[] = { 4,9,3,7,6,4,24,4,8,4,10 };
    ② int len = sizeof(exint) / sizeof(int);
    ③ int value = 0;
    ④ for (int i = 0; i < len; i++) {
        ⑤     if (exint[i] == 4) {
        ⑥         value++;
        }
    }
    ⑦ printf("%d", value);
}
```

- ① 배열을 선언할 때 사용할 개수를 생략하고 초기값을 지정하면, 초기값으로 지정된 값의 수와 같은 크기의 배열이 선언된다.

exint[11] 

4	9	3	7	6	4	24	4	8	4	10
---	---	---	---	---	---	----	---	---	---	----

- ② 정수형 변수 len을 선언하고 초기값으로 배열 exint의 크기에서 int 자료형의 크기를 나눈 값을 할당한다. int 자료형은 4Byte이고, exint는 int 자료형의 요소 11개를 가지고 있으므로 44Byte이다. 즉 len은 exint의 요소 개수인 11을 저장한다.
- ③ 정수형 변수 value를 선언하고 초기값으로 0을 할당한다.

- ④ 반복 변수 `i`가 0에서 시작하여 1씩 증가하면서 `len`보다 작은 동안 ⑤번을 반복하여 수행한다.  
 ⑤ `exint` 배열의 `i`번째 값이 4이면 ⑥번을 수행한다.  
 ⑥ `value = value + 1`;과 동일하다. `value`의 값을 1씩 누적시킨다.

⑦ 결과 4

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

`exint[11]`    4   9   3   7   6   4   24   4   8   4   10

value	i	exint[i]	출력
0	0	4	4
1	1	9	
2	2	3	
3	3	7	
4	4	6	
	5	4	
	6	24	
	7	4	
	8	8	
	9	4	
	10	10	
	11		

정답 1. 4

### [핵심013] Java에서의 배열과 문자열

#### Java에서의 배열 처리

- Java에서는 향상된 for문을 사용할 수 있는데, 향상된 for문은 객체를 대상으로만 가능하다.
- Java에서는 배열을 객체로 취급하며, 배열을 이용하여 작업할 때 필요할 만한 내용은 이미 API로 만들어 두었기 때문에 잘 골라서 사용하면 된다.

**예제** 다음은 Java에서 향상된 for문을 사용한 예제이다. 결과를 확인하시오.

```
public class Example {
    public static void main(String[] args) {
        ① int[] a = {90,100,80,70,60,50,30};
        ② int hap = 0;
        ③ float avg;
        ④ for (int i : a)
        ⑤     hap = hap + i;
        ⑥     avg = (float)hap / a.length;
        ⑦     System.out.printf("%d, %.2f", hap, avg);
    }
}
```

### 코드 해설

① 배열을 선언하면서 초기값을 지정한다. 개수를 지정하지 않으면 초기값으로 지정된 수만큼 배열의 요소가 만들어진다. 이건 C언어와 같다.

	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
배열 a	90	100	80	70	60	50	30

② 총점을 구하기 위해 정수형 변수 hap을 선언하고 초기값으로 0을 할당한다.

③ 평균을 구하기 위해 실수형 변수 avg를 선언한다.

④ 향상된 반복문이다. a 배열의 요소 수만큼 ⑤번을 반복 수행한다.

- int i : a 배열의 각 요소가 일시적으로 저장될 변수를 선언한다. a 배열과 형이 같아야 한다. 즉 a 배열이 정수면 정수, 문자면 문자여야 한다.
- a : 배열의 이름을 입력한다. a 배열이 7개의 요소를 가지므로 각 요소를 i에 저장하면서 ⑥번을 7번 수행한다.

⑤ i의 값을 hap에 누적한다. i는 a 배열 각 요소의 값을 차례로 받는다.

변수의 변화는 다음과 같다.

첫 번째 수행 : a 배열의 첫 번째 값이 i를 거쳐 hap에 누적된다.

hap	i	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
90	90	90	100	80	70	60	50	30

두 번째 수행 : a 배열의 두 번째 값이 i를 거쳐 hap에 누적된다.

hap	i	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
190	100	90	100	80	70	60	50	30

세 번째 수행 : a 배열의 세 번째 값이 i를 거쳐 hap에 누적된다.

hap	i	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
270	80	90	100	80	70	60	50	30

이런 방식으로 a 배열의 요소 수만큼 반복한다.

⑥ 총점이 저장되어 있는 hap을 실수형으로 변환한 후 a 배열의 요소 수로 나눠 평균을 구한다.

- length : 클래스에는 클래스의 속성과 수행해야 할 메소드가 포함되어 있다. length는 배열 클래스의 속성으로 배열 요소의 개수가 저장되어 있다. a 배열은 7개의 요소를 가지므로 a.length는 7을 가지고 있다.
- a.length : 개체 변수의 이름과 메소드는 .(마침표)로 연결하여 사용한다.

⑦ 결과 480, 68.57

## Java의 문자열

Java에서는 문자열을 처리할 수 있도록 클래스를 제공하므로 그에 따른 속성과 메소드도 당연히 지원하는데 여기서는 문제 풀이에 꼭 필요한 속성과 메소드만 학습한다.

**예제** 다음은 문자열을 거꾸로 출력하는 Java 프로그램이다. 결과를 확인하시오.

```
public class Example {
    public static void main(String[] args){
        ① String str = "Information!";
        ② int n = str.length();
        ③ char[] st = new char[n];
        ④ n--;
        ⑤ for (int k = n; k >= 0; k--) {
        ⑥     st[n-k] = str.charAt(k);
        ⑦ }
    }
}
```

```

8   for (char k : st) {
9       System.out.printf("%c", k);
    }
    }
}

```

### 코드 해설

- 1 문자열 변수 `st`를 선언하면서 초기값으로 "Information!"을 할당한다. 객체 변수를 생성할 때는 예약어 `new`를 입력해야 하지만 문자열 변수는 초기값을 이용해 `new` 없이 바로 생성할 수 있다.
- 2 문자열 클래스에서 `length()` 메소드는 해당 문자열의 길이를 반환한다. 즉 정수형 변수 `n`에 `st`에 저장된 문자열의 길이 12가 저장된다.
- 3 Java에서는 배열도 클래스이므로 생성할 때는 `new`를 사용해야 한다. `n`에 12가 저장되어 있으므로 `st`는 12개의 요소를 갖는 문자 배열로 생성된다.
  - `char[] st`: 문자 배열이고 배열 이름은 `st`이다. `char st[]`처럼 입력해도 된다. `st`만 임의로 입력하고 나머지는 그대로 적어준다.
  - `new`: 객체 변수를 생성하는 예약어다. 그대로 입력한다.
  - `char[n]`: 문자 배열의 크기를 지정하므로 12개 요소를 갖는 문자 배열이 생성된다. `n`을 제외한 나머지는 항상 그대로 입력한다.
- 4 12개짜리 배열이지만 배열의 첨자는 0부터 시작하여 11까지 사용하기 때문에 첨자로 사용할 변수의 값을 1 감소시킨다( $n = 11$ ).

`st[0] st[1] st[2] st[3] st[4] st[5] st[6] st[7] st[8] st[9] st[10] st[11]`  
`st`

--	--	--	--	--	--	--	--	--	--	--	--

- 5 정수형 변수 `k`를 반복 변수로 선언하면서 초기값으로 11을 갖고, 1씩 감소시키면서 0보다 크거나 같은 동안 ⑥번을 반복 실행한다. 실행할 문장이 한 개이므로 ⑤, ⑦번의 종결호는 없어도 된다. 반복 변수 `k`는 `for`문 안에서 선언한 지역 변수이기 때문에 `for`문을 벗어나면 소멸된다.
- 6 `charAt()` 메소드는 해당 문자열에서 인수에 해당하는 위치의 문자열을 반환한다.

#### 첫 번째 수행

`k = 11, n = 11`이므로 `st.charAt(k)`는 '!'을 반환한다. '!'을 `st[n-k]`번째, 즉 `st[0]`번째에 저장한다.

`st[0] st[1] st[2] st[3] st[4] st[5] st[6] st[7] st[8] st[9] st[10] st[11]`  

!											
---	--	--	--	--	--	--	--	--	--	--	--

#### 두 번째 수행

`k = 10, n = 11`이므로 `st.charAt(k)`는 'n'을 반환한다. 'n'을 `st[1]`에 저장한다.

`st[0] st[1] st[2] st[3] st[4] st[5] st[6] st[7] st[8] st[9] st[10] st[11]`  

!	n										
---	---	--	--	--	--	--	--	--	--	--	--

위와 같은 작업을 반복수행하다가 `k = 0`일 때, '!'를 `st[11]`에 저장한 다음 `k`는 -1이 되어 반복문을 벗어난다.

`st[0] st[1] st[2] st[3] st[4] st[5] st[6] st[7] st[8] st[9] st[10] st[11]`  

!	n	o	i	t	a	m	r	o	f	n	!
---	---	---	---	---	---	---	---	---	---	---	---

- 8 향상된 반복문이다. `st` 배열의 각 요소를 처음부터 차례대로 문자 변수 `k`에 옮기면서 `st` 배열의 개수, 즉 ④번을 12회 반복 수행한다.
- 9 `k`에는 `st` 배열 각 요소의 값이 할당되므로 12회 수행을 마치면 출력 결과는 다음과 같다. 서식 문자열에 `'\n'`이 없으므로 한 줄에 붙여 출력한다.

결과    `!noitamrofni!`



2020년 1회 기사 실기

1. 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.)

```
public class Test {
    static int[] arr() {
        int a[] = new int[4];
        int b = a.length;
        for(int i = 0; i < b; i++)
            a[i] = i;
        return a;
    }

    public static void main(String[] args) {
        int a[] = arr();
        for(int i = 0; i < a.length; i++)
            System.out.print(a[i] + " ");
    }
}
```

답:

모든 Java 프로그램은 반드시 main() 메소드부터 시작해야 한다.

```
public static void main(String[] args) {
    ❶ int a[] = arr();
    for(int i = 0; i < a.length; i++)
        System.out.print(a[i] + " ");
}
```

❶ 정수형 배열 a를 선언하고, 인수 없이 arr() 메소드를 호출한 다음 돌려받은 값을 배열 a에 저장한다.

```
    ❷ static int[] arr() {
    ❸         int a[] = new int[4];
    ❹         int b = a.length;
    ❺         for(int i = 0; i < b; i++)
    ❻         a[i] = i;
    ❼         return a;
    }
```

❷ 메소드의 리턴값이 정수형 배열인 arr() 메소드의 시작점이다.

❸ 4개의 요소를 갖는 정수형 배열 a를 선언한다.

❹ 정수형 변수 b를 선언하고 배열 a의 길이인 4로 초기화한다.

• length : 배열 클래스의 속성으로 배열 요소의 개수가 저장되어 있다. a 배열은 4개의 요소를 가지므로 a.length는 4를 가지고 있다.

❺ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 b보다 작은 동안 ❻번 문장을 반복 수행한다.

❻ a[i]에 반복 변수 i의 값을 저장한다.

	a[0]	a[1]	a[2]	a[3]
배열 a	0	1	2	3

⑦ 배열 a에 저장된 값들을 호출한 곳(main() 메소드)으로 반환한다.

```
public static void main(String[] args) {
    ⑧ int a[] = arr();
    ⑨ for(int i = 0; i < a.length; i++)
    ⑩     System.out.print(a[i] + " ");
}
```

⑧ arr() 메소드로부터 반환받은 값들을 main() 메소드의 배열 a에 저장한다.

⑨ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 a 배열의 길이인 4보다 작은 동안 ⑩번 문장을 반복 수행한다.

⑩ a[i]의 값과 공백 한 칸을 출력한다.

정답 1.0123

## [핵심014] 포인터

### 포인터와 포인터 변수

- 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;).

### 포인터와 배열

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

예 int a[5], \*b;

b = a → 배열의 대표명을 적었으므로 a 배열의 시작 주소인 a[0]의 주소를 b에 저장한다.

b = &a[0] → a 배열의 첫 번째 요소인 a[0]의 주소(&)를 b에 저장한다.

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

예제 다음 C언어로 구현된 프로그램의 출력 결과를 확인하시오.

main( )

{

int a = 50; ① 정수형 변수 a를 선언하고 50으로 초기화한다.

int \*b; ② 정수형 변수가 저장된 곳의 주소를 기억할 포인터 변수 b를 선언한다.

b = &a; ③ 정수형 변수 a의 주소를 포인터 변수 b에 기억시킨다. b에는 a의 주소가 저장된다.

`*b = *b+20;` ④ b가 가리키는 곳의 값에 20을 더한다. b가 가리키는 곳이 a이므로 결국 a의 값도 바뀌는 것이다.

`printf("%d, %d", a, *b);` ⑤ 결과 70, 70

- ②번과 같이 선언할 때 \*는 해당 변수가 포인터 변수라는 것을 의미한다.
- ④, ⑤번과 같이 사용할 때 \*를 붙이면 그 포인터 변수가 가리키는 곳의 값을 의미한다.

}

1. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오.

```
#include <stdio.h>
main() {
    int a[5], b = 1, sum = 0;
    for (int i = 4; i > -1; i--) {
        a[i] = b;
        b *= 3;
    }
    for (int i = 4; i > -1; i -= 2)
        sum += *(a + i);
    printf("%d", sum);
}
```

답 :

```
#include <stdio.h>
main() {
    int a[5], b = 1, sum = 0;

    for (int i = 4; i > -1; i--) { ①

        a[i] = b; ②
        b *= 3; ③
    }
    for (int i = 4; i > -1; i -= 2) ④

        sum += *(a + i); ⑤
    printf("%d", sum); ⑥
}
```

5개의 요소를 갖는 정수형 배열 a와 각각 초기값 1, 0을 갖는 변수 b, sum을 선언한다.

반복 변수 i가 4에서 시작하여 1씩 감소하면서 -1보다 큰 동안 ②, ③번을 반복 수행한다.

a[i]에 b의 값을 저장한다.

'b = b \* 3'과 동일하다. b에 3을 곱한 값을 저장한다.

반복 변수 i가 4에서 시작하여 2씩 감소하면서 -1보다 큰 동안 ⑤번을 반복 수행한다.

sum에 (a + i)가 가리키는 곳의 값을 누적한다.

결과 91

※ Java와 Python에서는 포인터 변수를 사용할 수 없다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

i	b	sum	*(a+i)	a[]					출력
				a[0]	a[1]	a[2]	a[3]	a[4]	
	1	0							
4	3							1	
3	9						3	1	
2	27					9	3	1	
1	81				27	9	3	1	
0	243			81	27	9	3	1	
-1									
4		1	1						
2		10	9						
0		91	81						
-2									

정답 1. 91

### [핵심015] 사용자 정의 함수

사용자가 필요한 기능을 취향대로 만들어 사용할 수 있는 함수이다. 사용자 정의 함수를 이용하면 프로그램 구조가 간단해지고 이해하기가 쉬워진다. 무엇보다 동일한 코드를 반복 입력하는 수고를 줄일 수 있다.

**예제** 다음 프로그램의 실행 결과를 확인하시오.

```
#include <stdio.h>
```

```
① void func(int i, int j);
```

```
main( )
```

```
{
  ② int a = 3, b = 12;
  ③ func(a, b);
  ④ printf("%d, %d\n", a, b);
}
```

사용할 사용자 정의 함수를 선언하는 곳이다. ④번에서 작성하는 사용자 정의 함수를 이곳에서 정의하는 것이다. 이런 함수를 이 프로그램에서 만들어 사용하겠다는 의미이다.

- void : 사용할 함수의 리턴 값이 없음을 알려준다. 그대로 적어준다.
- func : 사용할 함수의 이름이다. ④번에서 정의한 이름과 일치해야 한다.
- (int i, int j) : 함수에서 사용할 인수이다. 호출하는 곳에서 보내준 인수의 순서와 자료형이 일치해야 한다. 인수로 사용하는 변수의 이름이 같을 필요는 없다.

정수형 변수 a와 b를 선언하고, 초기값으로 3과 12를 각각 할당한다.

a, b, 즉 3과 12를 인수로 하여 func 함수를 호출한다. 'func(3,12);'라는 의미이다. ④번으로 이동한다.

결과 3, 12

func( ) 함수에서 돌려받은 값이 없으므로 원래의 a와 b의 값인 3과 12를 그대로 출력한다.

#### 4 void func(i, j)

- void : 함수의 리턴 값이 없을 때 적어준다.
- func : 함수의 이름이다. 사용자가 임의로 지정하면 된다.
- (i, j) : 함수에서 사용할 인수이다. 호출하는 곳에서 보내준 인수의 순서와 자료형이 일치해야 한다. ③번에서 'func(a, b)'라고 했으므로 i는 a의 값 3을 받고, j는 b의 값 12를 받는다.

#### 5 int i, j;

인수로 받은 i와 j가 정수형 변수임을 선언한다. 꼭 해야 한다.

#### 6 {

⑥~⑩번이 func 함수의 범위이다.

#### 7 i \*= 3;

i = i \* 3이므로 i는 9가 된다.

#### 8 j /= 3;

j = j / 3이므로 j는 4가 된다.

#### 9 printf("%d, %d\n", i, j);

결과 9, 4

#### 10 }

함수를 마치고 ⑩번으로 이동한다.

2020년 1회 기사 실기

1. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오.

```
#include <stdio.h>
void align(int a[]) {
    int temp;
    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 4 - i; j++)
            if (a[j] > a[j+1]) {
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
}

main() {
    int a[] = { 85, 75, 50, 100, 95 };
    align(a);
    for (int i = 0; i < 5; i++)
        printf("%d ", a[i]);
}
```

답:

모든 C 프로그램은 반드시 main() 함수부터 시작해야 한다.

```
main() {
    ① int a[] = { 85, 75, 50, 100, 95 };
    ② align(a);
    for (int i = 0; i < 5; i++)
        printf("%d ", a[i]);
}
```

- ① 배열을 선언할 때 사용할 개수를 생략하고 초기값을 지정하면, 초기값으로 지정된 값의 수와 같은 크기의 배열이 선언된다.

	a[0]	a[1]	a[2]	a[3]	a[4]
배열 a	85	75	50	100	95

- ② a를 인수로 하여 함수 align( )을 호출한다. 인수로 배열의 이름을 지정하면 배열의 시작 주소가 인수로 전달된다. 그러니까 align(a)는 align(&a[0])과 같은 의미다.

```

3 void align(int a[]) {
4     int temp;
5     for (int i = 0; i < 4; i++)
6         for (int j = 0; j < 4 - i; j++)
7             if (a[j] > a[j+1]) {
8                 temp = a[j];
9                 a[j] = a[j+1];
10                a[j+1] = temp;
11            }
12 }

```

- ③ ②번에서 'align(a)'라고 했으므로 정수형 배열 a는 main( ) 함수의 a 배열의 시작 주소를 받는다.

	a[0]	a[1]	a[2]	a[3]	a[4]
배열 a	85	75	50	100	95

- ④ 정수형 변수 temp를 선언한다.  
 ⑤ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 4보다 작은 동안 ⑥번을 반복하여 수행한다.  
 ⑥ 반복 변수 j가 0에서 시작하여 1씩 증가하면서 4-i보다 작은 동안 ⑦번을 반복하여 수행한다.  
 ⑦ a[j]가 a[j+1]보다 크면 ⑧~⑩번 사이의 문장을 실행한다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

i	j	a[j]	a[j+1]	temp	배열 a
0	0	85	75	85	
	1	75	85	85	
	2	85	50	100	
	3	50	85		
	4	85	100		
1	0	75	50	75	
	1	50	75		
	2	75	85		
	3	85	95		
2	0	50	75		
	1	75	85		
	2				
3	0	50	75		
	1				
4					

- ⑪ #문의 끝이다.  
 ⑫ 함수를 마치고 align(a) 함수를 호출했던 main( ) 함수로 제어를 옮긴다.

```
main() {
    int a[] = { 85, 75, 50, 100, 95 };
    align(a);
    ⑬ for (int i = 0; i < 5; i++)
    ⑭     printf("%d ", a[i]);
}
```

- ⑬ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 5보다 작은 동안 ⑭번을 반복 수행한 후 프로그램을 종료한다.  
⑭ a[i]의 값을 정수로 출력한 후 한 칸을 띄운다.

결과 50 75 85 95 100

정답 1. 50 75 85 95 100

### [핵심016] 재귀 함수

자기가 자기를 호출하는 순환 프로그램을 의미한다. 재귀 함수는 순환하는 만큼 반복하여 실행하면서 변수에 저장된 값을 추적하면 결과를 이해하기 쉽다.

**예제** 다음은 재귀 함수를 이용해 팩토리얼(Factorial)을 구하는 프로그램이다. 실행 결과를 확인하시오.

```
#include <stdio.h>
main()
{
    printf("%d", factorial(5));
}

factorial(int n) {
    if ( n <= 1 )
        return 1;
    else
        return n * factorial(n-1);
}
```

#### 코드 해설

```
main()
{
    ① printf("%d", factorial(5));
}
```

- ① 5를 인수로 하여 factorial 함수를 호출한 다음 돌려받은 값을 정수형으로 출력한다.

```

factorial(int n) {
  ❷ if ( n <= 1 )
    return 1;
  else
    ❸ return n * factorial(n-1);
}

```

①회

factorial 함수가 호출될 때 5를 전달받았으므로 n은 5이다. ❷의 조건을 만족하지 않으므로 ❸을 수행한다.

❸을 수행하기 위해 factorial() 함수를 호출하는데, 호출할 때 전달되는 값은 factorial(n-1)이므로 factorial(4)인 상태로 호출된다.

```

factorial(int n) {
  ❹ if ( n <= 1 )
    return 1;
  else
    ❺ return n * factorial(n-1);
}

```

②회

factorial 함수가 호출될 때 4를 전달받았으므로 n은 4이다. ❹의 조건을 만족하지 않으므로 ❺을 수행한다.

❺을 수행하기 위해 factorial() 함수를 호출하는데, 호출할 때 전달되는 값은 factorial(n-1)이므로 factorial(3)인 상태로 호출된다.

```

factorial(int n) {
  ❻ if ( n <= 1 )
    return 1;
  else
    ❼ return n * factorial(n-1);
}

```

③회

factorial 함수가 호출될 때 3을 전달받았으므로 n은 3이다. ❻의 조건을 만족하지 않으므로 ❼을 수행한다.

❼을 수행하기 위해 factorial() 함수를 호출하는데, 호출할 때 전달되는 값은 factorial(n-1)이므로 factorial(2)인 상태로 호출된다.

```

factorial(int n) {
  ❽ if ( n <= 1 )
    return 1;
  else
    ❾ return n * factorial(n-1);
}

```

④회

factorial 함수가 호출될 때 2를 전달받았으므로 n은 2이다. ❽의 조건을 만족하지 않으므로 ❾을 수행한다.

❾을 수행하기 위해 factorial() 함수를 호출하는데, 호출할 때 전달되는 값은 factorial(n-1)이므로 factorial(1)인 상태로 호출된다.

```

factorial(int n) {
  ❿ if ( n <= 1 )
    ⓫ return 1;
  else
    return n * factorial(n-1);
}

```

⑤회

factorial 함수가 호출될 때 1을 전달받았으므로 n은 1이다. ❿의 조건을 만족하므로 ⓫을 수행한다.

'return 1;'이므로 함수의 실행을 종료하고 1을 반환하면서 제어를 ④회 factorial(n-1) 함수를 호출했던 곳으로 옮긴다.



```
factorial(int n) {
    if ( n <= 1 )
        return 1;
    else
        ⑫ return n * factorial(n-1);
}
```

④회

⑤회 수행 과정에서 10이 반환되었으므로

⑫를 반환하면서 제어를 ③회 factorial(n-1) 함수로 옮긴다.

return  $n * \underline{\text{factorial}(n-1)}$

③      ⑤

- ③ : 2 ('factorial(n-1)'을 호출할 때 n은 2였으므로)

- ⑤ : 1 (⑤회 수행 과정에서 10이 반환되었으므로)

```
factorial(int n) {
    if ( n <= 1 )
        return 1;
    else
        ⑬ return n * factorial(n-1);
}
```

③회

④회 수행 과정에서 2가 반환되었으므로

⑬을 반환하면서 제어를 ②회 factorial(n-1) 함수로 옮긴다.

return  $n * \underline{\text{factorial}(n-1)}$

③      ②

```
factorial(int n) {
    if ( n <= 1 )
        return 1;
    else
        ⑭ return n * factorial(n-1);
}
```

②회

③회 수행 과정에서 60이 반환되었으므로

⑭를 반환하면서 제어를 ①회 factorial(n-1) 함수로 옮긴다.

return  $n * \underline{\text{factorial}(n-1)}$

④      ⑥

```
factorial(int n) {
    if ( n <= 1 )
        return 1;
    else
        ⑮ return n * factorial(n-1);
}
```

①회

②회 수행 과정에서 24가 반환되었으므로

⑮를 반환하면서 제어를 처음 factorial(5) 함수로 옮긴다.

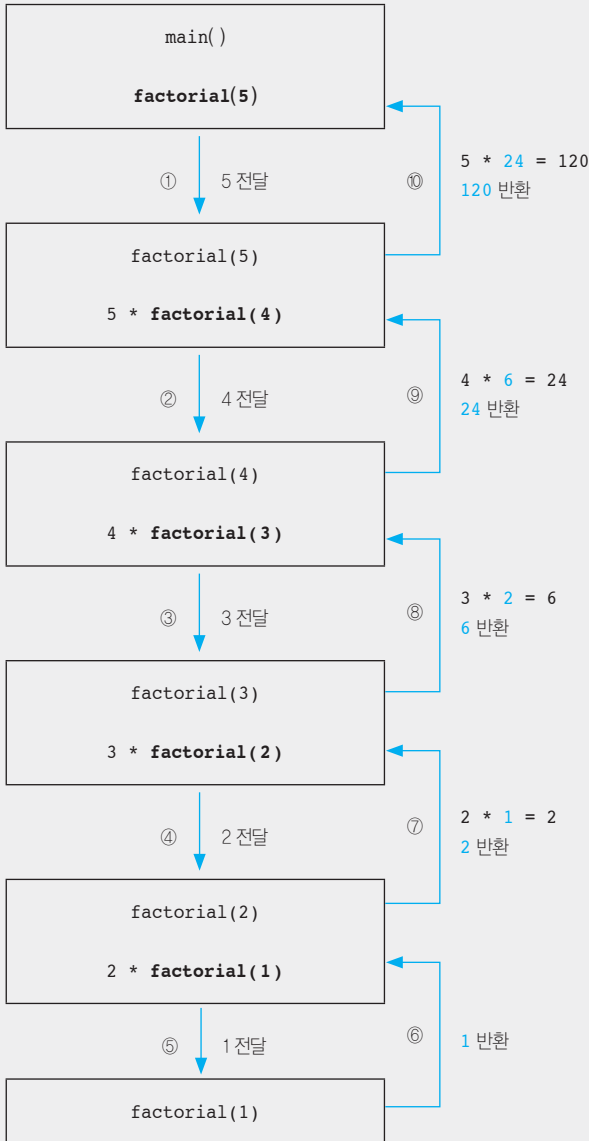
return  $n * \underline{\text{factorial}(n-1)}$

⑤      ②④

```
main()
{
    ⑩ printf("%d", factorial(5));
}
```

⑩ 회 수행 과정에서 120이 반환되었으므로 돌려받은 값 120을 정수형으로 출력하고 프로그램을 종료한다.

지금까지의 재귀 함수 과정을 개괄적인 그림을 통해 살펴보자.



2020년 2회 가능사 실기

1. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오.

```
#include <stdio.h>
hrd(num) {
    if (num <= 0)
        return;
    printf("%d ", num);
    hrd(num-1);
}
main( ) {
    hrd(5);
    return 0;
}
```

답 :

C 언어는 main( ) 함수부터 시작해야 한다.

```
main()
{
    ❶ hrd(5);
    return 0;
}
```

❶ 5를 인수로 하여 hrd 함수를 호출한다.

```
hrd(num) {
    ❷ if (num <= 0)
        return;
    ❸ printf("%d ", num);
    ❹ hrd(num-1);
    ①회 }
```

hrd 함수가 호출될 때 5를 전달받았으므로 num은 5이다. ❷의 조건을 만족하지 않으므로 ❸을 수행한다.

❸ num의 값 5를 출력하고 한 칸을 띄운다.

결과 5

❹를 수행하기 위해 hrd( ) 함수를 호출하는데, 호출할 때 전달되는 값은 num-1이므로 hrd(4)인 상태로 호출된다.

```
hrd(num) {
    ❺ if (num <= 0)
        return;
    ❻ printf("%d ", num);
    ❼ hrd(num-1);
    ②회 }
```

hrd 함수가 호출될 때 4를 전달받았으므로 num은 4이다. ❺의 조건을 만족하지 않으므로 ❻을 수행한다.

❻ num의 값 4를 출력하고 한 칸을 띄운다.

결과 5 4

⑦을 수행하기 위해 hrd( ) 함수를 호출하는데, 호출할 때 전달되는 값은 num-1이므로 hrd(3)인 상태로 호출된다.

```
hrd(num) {
  ⑧ if (num <= 0)
    return;
  ⑨ printf("%d ", num);
  ⑩ hrd(num-1);
}
```

③회

hrd 함수가 호출될 때 3을 전달받았으므로 num은 3이다. ⑧의 조건을 만족하지 않으므로 ⑨를 수행한다.

⑨ num의 값 3을 출력하고 한 칸을 띄운다.

결과 5 4 3

⑩을 수행하기 위해 hrd( ) 함수를 호출하는데, 호출할 때 전달되는 값은 num-1이므로 hrd(2)인 상태로 호출된다.

```
hrd(num) {
  ⑪ if (num <= 0)
    return;
  ⑫ printf("%d ", num);
  ⑬ hrd(num-1);
}
```

④회

hrd 함수가 호출될 때 2를 전달받았으므로 num은 2이다. ⑪의 조건을 만족하지 않으므로 ⑫를 수행한다.

⑫ num의 값 2를 출력하고 한 칸을 띄운다.

결과 5 4 3 2

⑬을 수행하기 위해 hrd( ) 함수를 호출하는데, 호출할 때 전달되는 값은 num-1이므로 hrd(1)인 상태로 호출된다.

```
hrd(num) {
  ⑭ if (num <= 0)
    return;
  ⑮ printf("%d ", num);
  ⑯ hrd(num-1);
}
```

⑤회

hrd 함수가 호출될 때 1을 전달받았으므로 num은 1이다. ⑭의 조건을 만족하지 않으므로 ⑮를 수행한다.

⑮ num의 값 1을 출력하고 한 칸을 띄운다.

결과 5 4 3 2 1

⑯을 수행하기 위해 hrd( ) 함수를 호출하는데, 호출할 때 전달되는 값은 num-1이므로 hrd(0)인 상태로 호출된다.

```
hrd(num) {
  ⑰ if (num <= 0)
    return;
  ⑱ printf("%d ", num);
  hrd(num-1);
}
```

⑥회

hrd 함수가 호출될 때 0을 전달받았으므로 num은 0이다. ⑰의 조건을 만족하므로 ⑱를 수행한다.

'return'이므로 함수의 실행을 종료하고 반환값 없이 제어를 ⑤회 hrd(0) 함수를 호출했던 곳으로 옮긴다.

```
hrd(num) {
    if (num <= 0)
        return;
    printf("%d ", num);
    hrd(num-1);
} ⑩
```

⑤회

⑩ 함수의 실행을 종료하고 반환값 없이 제어를 ④회 hrd(1) 함수를 호출했던 곳으로 옮긴다.

```
hrd(num) {
    if (num <= 0)
        return;
    printf("%d ", num);
    hrd(num-1);
} ⑪
```

④회

⑪ 함수의 실행을 종료하고 반환값 없이 제어를 ③회 hrd(2) 함수를 호출했던 곳으로 옮긴다.

```
hrd(num) {
    if (num <= 0)
        return;
    printf("%d ", num);
    hrd(num-1);
} ⑫
```

③회

⑫ 함수의 실행을 종료하고 반환값 없이 제어를 ②회 hrd(3) 함수를 호출했던 곳으로 옮긴다.

```
hrd(num) {
    if (num <= 0)
        return;
    printf("%d ", num);
    hrd(num-1);
} ⑬
```

②회

⑬ 함수의 실행을 종료하고 반환값 없이 제어를 ①회 hrd(4) 함수를 호출했던 곳으로 옮긴다.

```
hrd(num) {
    if (num <= 0)
        return;
    printf("%d ", num);
    hrd(num-1);
} ⑭
```

①회

⑭ 함수의 실행을 종료하고 반환값 없이 제어를 처음 hrd(5) 함수를 호출했던 곳으로 옮긴다.

```
main()
{
    hrd(5);
    ㉔ return 0;
}
```

㉔ main() 함수에서 'return 0;'는 프로그램 종료를 의미한다.

### [핵심017] Java의 클래스와 메소드

클래스는 객체 생성을 위한 필드(속성)와 메소드(함수)를 정의하는 설계도로, Java는 아무리 작은 프로그램이라도 클래스를 만들어서 사용해야 한다.

**예제** 다음은 두 수를 교환하는 Java 프로그램이다. 결과를 확인하시오.

```
public class Example {
    static class AAclass { ❶
        int i;
        int j;
    }
    public static void main(String[] args) {
        AAclass myVal = new AAclass( ); ❷
        myVal.i = 10;
        myVal.j = 20;
        myVal = change(myVal); ❸❹
        System.out.printf("i=%d, j=%d\n", myVal.i, myVal.j); ❺
    }
    static AAclass change(AAclass myVal) { ❻
        int temp;
        temp = myVal.i; ❼
        myVal.i = myVal.j; ❽
        myVal.j = temp; ❻
        return myVal; ❼
    }
}
```

AAclass 클래스를 정의한다. 클래스 안에 클래스를 정의할 때는 static을 붙인다.

- class : 클래스를 정의하는 명령어로, 꼭 써야하는 예약어이다.
- AAclass : 클래스 이름으로, 사용자가 원하는 이름을 임의로 지정할 수 있다. 단 첫 글자는 관례상 대문자로 지정한다.

#### 코드 해설

❷ AAclass라는 클래스 설계도를 이용하여 객체 변수 myVal을 생성한다.

- AAclass : 클래스의 이름이다. 그대로 적어준다.
- myVal : 사용자 정의 변수다. 사용자 마음대로 적어준다.
- new : 객체 생성 예약어다. 그대로 적어준다.
- AAclass( ) : 클래스의 이름이다. ( )를 붙여 그대로 적어준다.
- myVal은 다음과 같이 2개의 속성을 갖는다.

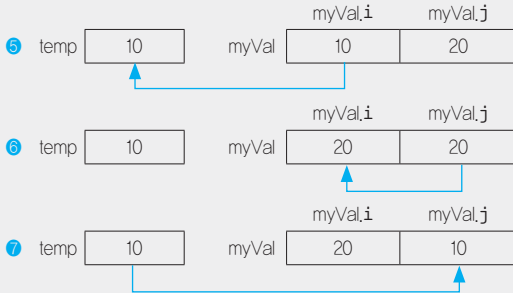
	myVal.i	myVal.j
myVal		

❸ 객체 변수 myVal을 인수로 하여 change( ) 메소드를 호출한다. 다음과 같이 객체의 모든 속성이 change( ) 메소드로 전달된다. 실제로는 myVal의 주소가 전달된다. 즉 리턴값을 돌려받지 않아도 변경된 결과가 적용되므로 myVal에 할당하지 않고 'change(myVal)'와 같이 입력해도 결과는 같다. 메모리의 주소를 그려서 설명하지 않는 이유는, Java에서는 주소를 컨트롤 할 수 있는 기능을 제공하지 않으며, 주소의 개념 없이 변수의 개념만 가지고 충분히 이해할 수 있기 때문이다.

	myVal.i	myVal.j
myVal	10	20

- **static** : 실행 클래스 안에 메소드를 정의할 때는 static을 붙여야 한다. 실행 클래스는 main( ) 메소드가 속해있는 클래스를 말한다.
- **Aclass : change( )** 메소드의 리턴값이 Aclass 클래스의 객체 변수라는 의미이다. 리턴값이 정수일 때 int를 적는 것과 같은 이치다. 꼭 적어야 한다.
- **change** : 메소드의 이름이다. 사용자가 원하는 이름을 적으면 된다.
- **(Aclass myVal) : change( )** 메소드의 인수로 Aclass 클래스의 객체 변수 myVal을 받는다. ④번에서 호출할 때 사용한 객체 변수명과 동일해도 되고 그렇지 않아도 된다.

⑤~⑦ : myVal.i와 myVal.j의 값을 교환하는 명령으로 실행 과정은 다음과 같다.



⑧ change( ) 메소드는 myVal을 반환하고 제어를 ⑨번으로 옮긴다.

⑨ change( ) 메소드의 리턴값을 myVal에 저장한다.



⑩ 결과 i=20, j=10

1. 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오.

```
public class Test {
    static int power(int data, int exp) {
        int i, result = 1;
        for(i = 0; i < exp; i++)
            result = result * data;
        return result;
    }
    public static void main(String args[]) {
        System.out.print(power(2, 10));
    }
}
```

답 :

모든 Java 프로그램은 반드시 main( ) 메소드부터 시작해야 한다.

```
public static void main(String args[]) {
    ① System.out.print(power(2, 10));
}
```

- ① 2와 10을 인수로 하여 power() 메소드를 호출한 다음 돌려받은 값을 출력한다. power(2, 10)로 호출한다.

```
static int power(int data, int exp) {
    ② int i, result = 1;
    ③ for(i = 0; i < exp; i++)
    ④     result = result * data;
    ⑤ return result;
}
```

power() 메소드가 호출될 때 2와 10을 전달받았으므로 data는 2이고 exp는 10이다.

- ② 정수형 변수 i와 result를 선언하고, result의 초기값으로 1을 할당한다.  
 ③ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 exp보다 작은 동안 ④번 문장을 반복한다. exp가 10을 가지고 있으므로 총 10회 반복 수행한다.  
 ④ result \* data의 결과를 result에 저장한다.  
 ⑤ ④번을 10회 반복하는 과정은 다음과 같다.

i	result	data	exp
	1	2	10
0	2		
1	4		
2	8		
3	16		
4	32		
5	64		
6	128		
7	256		
8	512		
9	1024		
10			

- ⑤ result의 값이 1024이므로 반환값 1024를 가지고 power(2, 10) 메소드를 호출했던 main() 메소드로 제어를 옮긴다.

```
public static void main(String args[]) {
    ⑥ System.out.print(power(2, 10));
}
```

- ⑥ power(2, 10) 메소드에서 돌려받은 1024를 출력한다.

결과 1024

정답 1. 1024

## [핵심018] Python의 기초

### input() 함수

- Python의 표준 입력 함수로, 키보드로 입력받아 변수에 저장하는 함수이다.
- 입력되는 값은 문자열로 취급되어 저장된다.



• 형식1

**변수 = input(출력문자)**

- '출력문자'는 생략이 가능하며, '변수'는 사용자가 임의로 지정할 수 있다.
- 값을 입력하고 **Enter**를 누르면, 입력한 값이 '변수'에 저장된다.

• 형식2

**변수1, 변수2, ... = input(출력문자).split(분리문자)**

- 화면에 '출력문자'가 표시되고 입력받은 값을 '분리문자'로 구분하여 각각 변수1, 변수2, ...에 저장한다.
- '분리문자'를 생략하면 공백으로 값들을 구분한다.

**print( ) 함수**

- Python의 표준 출력 함수로, 값을 화면에 출력하는 함수이다.
- 형식1

**print(출력값1, 출력값2, ..., sep = 분리문자, end = 종료문자)**

- '출력값'에는 숫자, 문자, 문자열, 변수 등 다양한 값이나 식이 올 수 있다.
- 'sep'는 여러 값을 출력할 때 값과 값 사이를 구분하기 위해 출력하는 문자로, 생략할 경우 기본값은 공백 한 칸(' ')이다.
- 'end'는 맨 마지막에 표시할 문자로, 생략할 경우 기본값은 줄 나눔이다.

• 형식2

**print(서식 문자열 % (출력값1, 출력값2, ...))**

- C와 Java에서 사용했던 서식 문자열이 동일하게 적용된다.
- 출력값이 한 개일 경우 출력값에 대한 괄호를 생략할 수 있다.

**리스트(List)**

- C와 Java에서는 여러 요소들을 한 개의 이름으로 처리할 때 배열을 사용하지만 Python에서는 리스트를 사용한다.
- 1차원 리스트 형식

**리스트명 = [ 값1, 값2, ... ]**

리스트명은 사용자가 임의로 지정하며, 리스트를 의미하는 대괄호 사이에 저장할 값들을 쉼표(,)로 구분하여 입력한다.

**리스트명 = list([ 값1, 값2, ... ])**

• 2차원 리스트 형식1

**리스트명 = [ [ 값1, 값2, 값3 ],  
[ 값4, 값5, 값6 ] ]**

- 2차원 리스트 형식2

```
리스트A = [ 값1, 값2, 값3 ]
리스트B = [ 값4, 값5, 값6 ]
리스트명 = [ 리스트A, 리스트B ]
```

## Range

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

<b>range(최종값)</b>	0에서 '최종값'-1까지 연속된 숫자를 생성한다.
<b>range(초기값, 최종값)</b>	'초기값'에서 '최종값'-1까지 연속된 숫자를 생성한다.
<b>range(초기값, 최종값, 증가값)</b>	<ul style="list-style-type: none"> <li>• '초기값'에서 '최종값'-1까지 '증가값'만큼 증가하면서 숫자를 생성한다.</li> <li>• '증가값'이 음수인 경우 '초기값'에서 '최종값'+1까지 '증가값'만큼 감소하면서 숫자를 생성한다.</li> </ul>

## 슬라이스(Slice)

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

<b>객체명[초기위치:최종위치]</b>	'초기위치'에서 '최종위치'-1까지의 요소들을 가져온다.
<b>객체명[초기위치:최종위치:증가값]</b>	<ul style="list-style-type: none"> <li>• '초기위치'에서 '최종위치'-1까지 '증가값'만큼 증가하면서 해당 위치의 요소들을 가져온다.</li> <li>• '증가값'이 음수인 경우 '초기위치'에서 '최종위치'+1까지 '증가값' 만큼 감소하면서 해당 위치의 요소들을 가져온다.</li> </ul>

- 슬라이스는 일부 인수를 생략하여 사용할 수 있다.

<b>객체명[:] 또는 객체명[::]</b>	객체의 모든 요소를 반환한다.
<b>객체명[초기위치:]</b>	객체의 '초기위치'에서 마지막 위치까지의 요소들을 반환한다.
<b>객체명[:최종위치]</b>	객체의 0번째 위치에서 '최종위치'-1까지의 요소들을 반환한다.
<b>객체명[:증가값]</b>	객체의 0번째 위치에서 마지막 위치까지 '증가값'만큼 증가하면서 해당 위치의 요소들을 반환한다.

1. 다음 Python으로 구현된 <코드>와 <입력>을 보고 프로그램을 분석하여 그 실행 결과를 쓰시오.

<코드>

```
x = input().split('//')
y = int(input())
val = list(range(0, y, 3))
val.remove(6)
print(x[1].find('i') + val[2])
```

<입력>

```
abcde//fghij
13
```

답 :

```
① x = input().split('//')
② y = int(input())
③ val = list(range(0, y, 3))
④ val.remove(6)
⑤ print(x[1].find('i') + val[2])
```

① 입력받은 데이터를 '/'로 구분하여 변수 x에 리스트로 저장한다. '/'는 저장하지 않는다.

	x[0]	x[1]
x	abcde	fghij

② 입력받은 데이터를 y에 정수로 저장한다.

③ 0에서 y-1까지 3씩 증가하는 숫자를 val에 리스트로 저장한다.

	val[0]	val[1]	val[2]	val[3]	val[4]
val	0	3	6	9	12

④ 리스트 val에서 6이 저장된 요소를 삭제하고, 뒤의 요소들을 한 칸씩 앞으로 당긴다.

	val[0]	val[1]	val[2]	val[3]
val	0	3	9	12

• remove() : 값을 기준으로 요소를 삭제할 때 사용하는 메소드이다.

⑤ 리스트 x[1]에서 처음 검색되는 'i'의 위치값과 val[2]의 값을 더하여 출력한다. x[1]에서 'i'의 위치값은 3이고, val[2]의 값은 9이므로 12(3+9)가 출력된다.

• find(값) : 문자열에서 처음 검색되는 '값'의 위치를 반환하는 메소드이다.

정답 1. 12

**[핵심019] 라이브러리 – C 언어**

- 라이브러리는 프로그램을 효율적으로 개발할 수 있도록 자주 사용하는 함수나 데이터들을 미리 만들어 모아 놓은 집합체이다.
- 표준 라이브러리 : 프로그래밍 언어에 기본적으로 포함되어 있는 라이브러리로, 여러 종류의 모듈이나 패키지 형태이다.
- 외부 라이브러리 : 개발자들이 필요한 기능들을 만들어 인터넷 등에 공유해 놓은 것으로, 외부 라이브러리를 다운받아 설치한 후 사용한다.

**C언어의 대표적인 표준 라이브러리**

헤더 파일	기능
stdio.h	<ul style="list-style-type: none"> <li>• 데이터의 입·출력에 사용되는 기능들을 제공한다.</li> <li>• 주요 함수 : printf, scanf, fprintf, fscanf, fclose, fopen 등</li> </ul>
math.h	<ul style="list-style-type: none"> <li>• 수학 함수들을 제공한다.</li> <li>• 주요 함수 : sqrt, pow, abs 등</li> </ul>
string.h	<ul style="list-style-type: none"> <li>• 문자열 처리에 사용되는 기능들을 제공한다.</li> <li>• 주요 함수 : strlen, strcpy, strcmp 등</li> </ul>
stdlib.h	<ul style="list-style-type: none"> <li>• 자료형 변환, 난수 발생, 메모리 할당에 사용되는 기능들을 제공한다.</li> <li>• 주요 함수 : atoi, atof, srand, rand, malloc, free 등</li> </ul>
time.h	<ul style="list-style-type: none"> <li>• 시간 처리에 사용되는 기능들을 제공한다.</li> <li>• 주요 함수 : time, clock 등</li> </ul>

1. C언어에서 자료형 변환, 난수 발생, 메모리 할당에 사용되는 기능들을 제공하는 표준 라이브러리로, atoi, atof, srand, rand, malloc, free 등의 다양한 함수가 포함되어 있는 헤더 파일의 이름을 쓰시오.

답 :

정답 1. stdlib.h

**[핵심020] 라이브러리 – Java****Java의 대표적인 표준 라이브러리**

패키지	기능
java.lang	<ul style="list-style-type: none"> <li>• Java에 기본적으로 필요한 인터페이스, 자료형, 예외 처리 등에 관련된 기능을 제공한다.</li> <li>• import문 없이도 사용할 수 있다.</li> <li>• 주요 클래스 : String, System, Process, Runtime, Math, Error 등</li> </ul>

java.util	<ul style="list-style-type: none"> <li>날짜 처리, 난수 발생, 복잡한 문자열 처리 등에 관련된 기능을 제공한다.</li> <li>주요 클래스 : Date, Calendar, Random, StringTokenizer 등</li> </ul>
java.io	<ul style="list-style-type: none"> <li>파일 입·출력과 관련된 기능 및 프로토콜을 제공한다.</li> <li>주요 클래스 : InputStream, OutputStream, Reader, Writer 등</li> </ul>
java.net	<ul style="list-style-type: none"> <li>네트워크와 관련된 기능을 제공한다.</li> <li>주요 클래스 : Socket, URL, InetAddress 등</li> </ul>
java.awt	<ul style="list-style-type: none"> <li>사용자 인터페이스(UI)와 관련된 기능을 제공한다.</li> <li>주요 클래스 : Frame, Panel, Dialog, Button, Checkbox 등</li> </ul>

1. Java에서 사용자 인터페이스(UI)와 관련된 기능을 제공하는 표준 라이브러리로, Frame, Panel, Dialog, Button, Checkbox 등의 다양한 클래스가 포함되어 있는 패키지의 이름을 쓰시오.

답 :

정답 1. java.awt

## [핵심021] 절차적 프로그래밍 언어

일련의 처리 절차를 정해진 문법에 따라 순서대로 기술해 나가는 언어이다.

### 절차적 프로그래밍 언어의 종류

언어	특징
C	<ul style="list-style-type: none"> <li>1972년 미국 벨 연구소의 데니스 리치에 의해 개발되었다.</li> <li>시스템 소프트웨어를 개발하기 편리하여 시스템 프로그래밍 언어로 널리 사용된다.</li> <li>자료의 주소를 조작할 수 있는 포인터를 제공한다.</li> <li>고급 프로그래밍 언어이면서 저급 프로그램 언어의 특징을 모두 갖췄다.</li> <li>UNIX의 일부가 C언어로 구현되었다.</li> <li>컴파일러 방식의 언어이다.</li> <li>이식성이 좋아 컴퓨터 기종에 관계없이 프로그램을 작성할 수 있다.</li> </ul>
ALGOL	<ul style="list-style-type: none"> <li>수치 계산이나 논리 연산을 위한 과학 기술 계산용 언어이다.</li> <li>PASCAL과 C언어의 모체가 되었다.</li> </ul>
COBOL	<ul style="list-style-type: none"> <li>사무 처리용 언어이다.</li> <li>영어 문장 형식으로 구성되어 있어 이해와 사용이 쉽다.</li> <li>4개의 DIVISION으로 구성되어 있다.</li> </ul>
FORTRAN	<ul style="list-style-type: none"> <li>과학 기술 계산용 언어이다.</li> <li>수학과 공학 분야의 공식이나 수식과 같은 형태로 프로그래밍 할 수 있다.</li> </ul>

1. 다음의 설명에 가장 적합한 프로그래밍 언어를 쓰시오.

- 수치 계산이나 논리 연산을 위한 과학 기술 계산용 언어이다.
- PASCAL과 C언어의 모체가 되었다.

답 :

2. 절차적 프로그래밍 언어의 한 종류로 시스템 소프트웨어를 개발하기 편리하여 시스템 프로그래밍 언어로 널리 사용된다. 컴파일러 방식의 언어이며, 이식성이 좋아 컴퓨터 기종에 관계없이 프로그램을 작성할 수 있다. 이 설명에 해당하는 프로그래밍 언어를 쓰시오.

답 :

**정답** 1. ALGOL 2. C

### [핵심022] 객체지향 프로그래밍 언어

현실 세계의 개체(Entity)를 기계의 부품처럼 하나의 객체로 만들어, 기계적인 부품들을 조립하여 제품을 만들 듯이 소프트웨어를 개발할 때도 객체들을 조립해서 프로그램을 작성할 수 있도록 한 프로그래밍 기법이다.

#### 객체지향 프로그래밍 언어의 종류

언어	특징
Java	<ul style="list-style-type: none"> <li>• 분산 네트워크 환경에 적용이 가능하며, 멀티스레드 기능을 제공하므로 여러 작업을 동시에 처리할 수 있다.</li> <li>• 운영체제 및 하드웨어에 독립적이며, 이식성이 강하다.</li> <li>• 캡슐화가 가능하고 재사용성 높다.</li> </ul>
C++	<ul style="list-style-type: none"> <li>• C언어에 객체지향 개념을 적용한 언어이다.</li> <li>• 모든 문제를 객체로 모델링하여 표현한다.</li> </ul>
Smalltalk	<ul style="list-style-type: none"> <li>• 1세대 객체지향 프로그래밍 언어 중 하나로 순수한 객체지향 프로그래밍 언어이다.</li> <li>• 최초로 GUI를 제공한 언어이다.</li> </ul>

1. 다음 설명에 가장 적합한 프로그래밍 언어를 쓰시오.

- 분산 네트워크 환경에 적용이 가능하며, 멀티스레드 기능을 제공하므로 여러 작업을 동시에 처리할 수 있다.
- 운영체제 및 하드웨어에 독립적이며, 이식성이 강하다.
- 캡슐화가 가능하고 재사용성 높다.

답 :

**정답** 1. Java

### [핵심 023] 객체지향 프로그래밍 언어의 구성 요소

객체(Object)	<ul style="list-style-type: none"> <li>• 데이터(속성)과 이를 처리하기 위한 연산(메소드)을 결합시킨 실체이다.</li> <li>• 데이터 구조와 그 위에서 수행되는 연산들을 가지고 있는 소프트웨어 모듈이다.</li> <li>• 속성(Attribute) : 한 클래스 내에 속한 객체들이 가지고 있는 데이터 값들을 단위별로 정의하는 것으로서 성질, 분류, 식별, 수량 또는 현재 상태 등을 표현한다.</li> <li>• 메소드(Method) : 객체가 메시지를 받아 실행해야 할 때 구체적인 연산을 정의하는 것으로, 객체의 상태를 참조하거나 변경하는 수단이 된다.</li> </ul>
클래스(Class)	<ul style="list-style-type: none"> <li>• 두 개 이상의 유사한 객체들을 묶어서 하나의 공통된 특성을 표현하는 요소이다. 즉 공통된 특성과 행위를 갖는 객체의 집합이라고 할 수 있다.</li> <li>• 객체의 유형 또는 타입(Object Type)을 의미한다.</li> <li>• 데이터를 추상화하는 단위이다.</li> </ul>
메시지(Message)	<ul style="list-style-type: none"> <li>• 객체들 간에 상호작용을 하는데 사용되는 수단으로 객체의 메소드(동작, 연산)를 일으키는 외부의 요구 사항이다.</li> <li>• 메시지를 받은 객체는 대응하는 연산을 수행하여 예상된 결과를 반환하게 된다.</li> </ul>

2020년 1. 2회 기사 필기

1. 객체지향 프로그램에서 데이터를 추상화하는 단위가 무엇인지 쓰시오.

답 :

**정답** 1. 클래스(Class)

### [핵심 024] 객체지향 프로그래밍 언어의 특징

캡슐화 (Encapsulation)	<ul style="list-style-type: none"> <li>• 데이터(속성)와 데이터를 처리하는 함수를 하나로 묶는 것을 의미한다.</li> <li>• 캡슐화된 객체의 세부 내용이 외부에 은폐(정보 은닉)되어, 변경이 발생할 때 오류의 파급 효과가 적다.</li> <li>• 캡슐화된 객체들은 재사용이 용이하다.</li> </ul>
정보 은닉 (Information Hiding)	캡슐화에서 가장 중요한 개념으로, 다른 객체에게 자신의 정보를 숨기고 자신의 연산만을 통하여 접근을 허용하는 것이다.
추상화 (Abstraction)	<ul style="list-style-type: none"> <li>• 불필요한 부분을 생략하고 객체의 속성 중 가장 중요한 것에만 중점을 두어 개략화하는 것, 즉 모델화하는 것이다.</li> <li>• 데이터의 공통된 성질을 추출하여 슈퍼 클래스를 선정하는 개념이다.</li> </ul>
상속성 (Inheritance)	<ul style="list-style-type: none"> <li>• 이미 정의된 상위 클래스(부모 클래스)의 모든 속성과 연산을 하위 클래스가 물려받는 것이다.</li> <li>• 상속성을 이용하면 하위 클래스는 상위 클래스의 모든 속성과 연산을 자신의 클래스 내에서 다시 정의하지 않고서도 즉시 사용할 수 있다.</li> </ul>
다형성 (Polymorphism)	<ul style="list-style-type: none"> <li>• 메시지에 의해 객체(클래스)가 연산을 수행하게 될 때 하나의 메시지에 대해 각 객체(클래스)가 가지고 있는 고유한 방법(특성)으로 응답할 수 있는 능력을 의미한다.</li> <li>• 객체(클래스)들은 동일한 메소드명을 사용하며 같은 의미의 응답을 한다.</li> </ul>

1. 다음은 객체지향 프로그래밍 언어의 특징에 대한 설명이다. 괄호에 들어갈 가장 알맞은 용어를 쓰시오.

( )은 메시지에 의해 객체가 연산을 수행하게 될 때 하나의 메시지에 대해 각 객체가 가지고 있는 고유한 방법으로 응답할 수 있는 능력으로 객체들은 동일한 메소드명을 사용하며 같은 의미의 응답을 한다.

답 :

**정답** 1. 다형성(Polymorphism)

### [핵심025] 스크립트 언어

- HTML 문서 안에 직접 프로그래밍 언어를 삽입하여 사용하는 것으로, 기계어로 컴파일 되지 않고 별도의 번역기가 소스를 분석하여 동작하게 하는 언어이다.
- 스크립트 언어의 분류

서버용 스크립트 언어	<ul style="list-style-type: none"> <li>• 서버에서 해석되어 실행된 후 결과만 클라이언트로 보내는 스크립트 언어이다.</li> <li>• 종류 : ASP, JSP, PHP, 파이썬(Python), 펄(Perl), 루비(Ruby) 등</li> </ul>
클라이언트용 스크립트 언어	<ul style="list-style-type: none"> <li>• 클라이언트의 웹 브라우저에서 해석되어 실행되는 스크립트 언어이다.</li> <li>• 종류 : 자바 스크립트(Java Script), VB 스크립트(Visual Basic Script) 등</li> </ul>

2020년 1,2회 기사 필기

1. 다음 중 스크립트 언어가 아닌 것을 모두 골라 쓰시오.

Java, JSP, ASP, Cobol, PHP, Smalltalk, Python

답 :

**정답** 1. Java, Cobol, Smalltalk

### [핵심026] 스크립트 언어의 종류

자바 스크립트(Java Script)	<ul style="list-style-type: none"> <li>• 넷스케이프(Netscape)의 브렌던 아이크(Brendan Eich)가 개발한 언어이다.</li> <li>• 웹 페이지의 동작을 제어하며, 변수 선언이 필요 없다.</li> </ul>
ASP(Active Server Page)	<ul style="list-style-type: none"> <li>• 서버 측에서 동적으로 수행되는 페이지를 만들기 위한 언어이다.</li> <li>• 마이크로 소프트 사에서 제작하였으며, Windows 계열에서만 수행 가능한 프로그래밍 언어이다.</li> </ul>
JSP(Java Server Page)	<ul style="list-style-type: none"> <li>• Java로 만들어진 서버용 스크립트 언어이다.</li> <li>• 다양한 운영체제에서 사용이 가능하다.</li> </ul>



PHP(Professional Hypertext Preprocessor)	<ul style="list-style-type: none"> <li>• 서버용 스크립트 언어로, Linux, Unix, Windows 운영체제에서 사용이 가능하다.</li> <li>• C, Java 등과 문법이 유사하여 배우기 쉽고 웹 페이지 제작에 많이 사용된다.</li> </ul>
파이썬(Python)	<ul style="list-style-type: none"> <li>• 객체지향 기능을 지원하는 대화형 인터프리터 언어이다.</li> <li>• 플랫폼에 독립적이고 문법이 간단하여 배우기 쉽다.</li> </ul>

1. Windows 또는 Linux 기반의 운영체제에서 사용하는 서버용 스크립트 언어의 하나로, C, Java 등의 대중적인 언어와 문법이 유사하여 배우기 쉽고 웹 페이지 제작에 많이 사용되는 프로그래밍 언어를 쓰시오.

답 :

정답 1. PHP

### [핵심027] 선언형 언어

- 프로그램이 수행해야 할 문제를 기술하는 언어로, 목표를 명시하고 알고리즘은 명시하지 않는다.
- 선언형 언어의 분류

함수형 언어	<ul style="list-style-type: none"> <li>• 수학적 함수를 조합하여 문제를 해결하는 언어로, 알려진 값을 함수에 적용하는 것을 기반으로 한다.</li> <li>• 대표적인 언어 : LISP</li> </ul>
논리형 언어	<ul style="list-style-type: none"> <li>• 기호 논리학에 기반을 둔 언어로, 논리 문장을 이용하여 프로그램을 표현하고 계산을 수행한다.</li> <li>• 대표적인 언어 : PROLOG</li> </ul>

- 선언형 언어의 종류

HTML	<ul style="list-style-type: none"> <li>• 인터넷의 표준 문서인 하이퍼텍스트 문서를 만들기 위해 사용하는 언어이다.</li> <li>• 특별한 데이터 타입이 없는 단순한 텍스트이므로 호환성이 좋고 사용이 편리하다.</li> </ul>
LISP	인공지능 분야에 사용되는 언어로, 기본 자료 구조가 연결 리스트 구조이며 재귀(Recursion) 호출을 많이 사용한다.
PROLOG	논리학을 기초로 한 고급 언어로, 인공지능 분야에서의 논리적인 추론이나 리스트 처리 등에 주로 사용된다.
XML	<ul style="list-style-type: none"> <li>• 기존 HTML의 단점을 보완하여 웹에서 구조화된 폭넓고 다양한 문서들을 상호 교환할 수 있도록 설계된 언어이다.</li> <li>• 사용자가 HTML에 새로운 태그(Tag)를 정의할 수 있다.</li> </ul>
Haskell	함수형 프로그래밍 언어로, 부작용(Side Effect)이 없고 코드가 간결하여 에러 발생 가능성이 낮다.

1. 다음 설명에 가장 적합한 프로그래밍 언어를 쓰시오.

기존 HTML의 단점을 보완하여 웹에서 구조화된 폭넓고 다양한 문서들을 상호 교환할 수 있도록 설계된 언어로, 사용자가 HTML에 새로운 태그(Tag)를 정의할 수 있다.

답 :

2. 다음은 함수형 언어에 대한 설명이다. 괄호에 들어갈 가장 적합한 용어를 쓰시오.

- 함수형 언어는 수학적 함수를 조합하여 문제를 해결하는 언어로, 알려진 값을 함수에 적용하는 것을 기반으로 한다.
- ( )은 인공지능 분야에서 사용되는 가장 대표적인 함수형 언어로, 재귀 호출을 많이 사용하는 것이 특징이다.

답 :

**정답** 1. XML(eXtensible Markup Language) 2. LISP

### [핵심028] 예외 처리

- 프로그램의 정상적인 실행을 방해하는 조건이나 상태를 예외(Exception)라고 하며, 이러한 예외가 발생했을 때 프로그래머가 해당 문제에 대비해 작성해 놓은 처리 루틴을 수행하도록 하는 것을 예외 처리(Exception Handling)라고 한다.
- C++, Ada, Java, 자바 스크립트와 같은 언어에는 예외 처리 기능이 내장되어 있으며, 그 외의 언어에서는 필요한 경우 조건문을 이용해 예외 처리 루틴을 작성한다.
- 예외의 원인 : 하드웨어 문제, 운영체제의 설정 실수, 라이브러리 손상, 사용자의 입력 실수, 받아들이지 못하는 연산, 할당하지 못하는 기억장치 접근 등

#### Java의 예외 처리

- 기본 형식

```
try {
    예외가 발생할 가능성이 있는 코드;
}
catch ( 예외객체1 매개변수 ) {
    예외객체1에 해당하는 예외 발생 시 처리 코드;
}
catch ( 예외객체2 매개변수 ) {
    예외객체2에 해당하는 예외 발생 시 처리 코드;
}
```

```
catch ( 예외객체n 매개변수) {
    예외객체n에 해당하는 예외 발생 시 처리 코드;
}
catch (Exception 매개변수) {
    예외객체1~n에 해당하지 않는 예외 발생 시 처리 코드;
}
finally {
    예외의 발생 여부와 관계없이 무조건 처리되는 코드;
}
```

1. 다음 설명에서 괄호에 공통으로 들어갈 가장 적합한 용어를 쓰시오.

프로그램의 정상적인 실행을 방해하는 조건이나 상태가 발생했을 때, 프로그래머가 해당 문제에 대비해 작성해 놓은 처리 루틴을 수행하도록 하는 것을 ( )라고 한다. C++, Ada, Java, 자바 스크립트와 같은 언어에는 ( )를 위한 기능이 내장되어 있으며, Java의 경우 try ~ catch문을 이용하여 ( )를 구현한다.

답 :

**정답** 1. 예외 처리(Exception Handling)

### [핵심 029] Java의 주요 예외 객체

예외 객체	발생 원인
ClassNotFoundException	클래스를 찾지 못한 경우
NoSuchMethodException	메소드를 찾지 못한 경우
FileNotFoundException	파일을 찾지 못한 경우
InterruptedException	입 · 출력 처리가 중단된 경우
ArithmeticException	0으로 나누는 등의 산술 연산에 대한 예외가 발생한 경우
IllegalArgumentException	잘못된 인자를 전달한 경우
NumberFormatException	숫자 형식으로 변환할 수 없는 문자열을 숫자 형식으로 변환한 경우
ArrayIndexOutOfBoundsException	배열의 범위를 벗어난 접근을 시도한 경우
NegativeArraySizeException	0보다 작은 값으로 배열의 크기를 지정한 경우
NullPointerException	존재하지 않는 객체를 참조한 경우

1. Java에서는 예외 처리(Exception Handling)를 위해 다양한 예외 객체를 활용한다. 다음 괄호(①~③)에 들어갈 알맞은 Java의 예외 객체를 <보기>에서 찾아 쓰시오.

( ① )	숫자 형식으로 변환할 수 없는 문자열을 숫자 형식으로 변환한 경우
NoSuchMethodException	메소드를 찾지 못한 경우
ArithmeticException	0으로 나누는 등의 산술 연산에 대한 예외가 발생한 경우
( ② )	입 · 출력 처리가 중단된 경우
( ③ )	잘못된 인자를 전달한 경우

<보기>

ClassNotFoundException, FileNotFoundException, InterruptedIOException, IllegalArgumentException, NumberFormatException, ArrayIndexOutOfBoundsException, NegativeArraySizeException

답

- ① :  
② :  
③ :

**정답** 1. ① NumberFormatException ② InterruptedIOException ③ IllegalArgumentException



## 2장 | 요구사항 확인

### [핵심 030] 현행 시스템 파악 절차

- 시스템 구성 파악 : 현행 시스템의 구성은 조직의 주요 업무를 담당하는 기간 업무와 이를 지원하는 지원 업무로 구분하여 기술
- 시스템 기능 파악 : 현행 시스템의 기능은 단위 업무 시스템이 현재 제공하는 기능들을 주요 기능과 하부 기능, 세부 기능으로 구분하여 계층형으로 표시
- 시스템 인터페이스 파악 : 현행 시스템의 인터페이스에는 단위 업무 시스템 간에 주고받는 데이터의 종류, 형식, 프로토콜, 연계 유형, 주기 등을 명시
- 아키텍처 구성 파악 : 현행 시스템의 아키텍처 구성은 기간 업무 수행에 어떠한 기술 요소들이 사용되는지 최상위 수준에서 계층별로 표현한 아키텍처 구성도로 작성
- 소프트웨어 구성 파악 : 소프트웨어 구성에는 단위 업무 시스템별로 업무 처리를 위해 설치되어 있는 소프트웨어들의 제품명, 용도, 라이선스 적용 방식, 라이선스 수 등을 명시
- 하드웨어 구성 파악 : 하드웨어 구성에는 단위 업무 시스템들이 운용되는 서버의 주요 사양과 수량, 그리고 이중화의 적용 여부를 명시
- 네트워크 구성 파악 : 네트워크 구성은 업무 시스템들의 네트워크 구성을 파악할 수 있도록 서버의 위치, 서버 간의 네트워크 연결 방식을 네트워크 구성도로 작성

#### 1. 다음 내용 중 괄호에 들어갈 가장 적합한 용어를 쓰시오.

현행 시스템의 ( )에는 단위 업무 시스템 간에 주고받는 데이터의 종류, 형식, 프로토콜, 연계 유형, 주기 등을 명시한다. 데이터를 어떤 형식으로 주고받는지, 통신규약은 무엇을 사용하는지, 연계 유형은 무엇인지 등을 반드시 고려해야 한다.

답 :

**정답** 1. 인터페이스

### [핵심 031] 개발 기술 환경

개발하고자 하는 소프트웨어와 관련된 운영체제 (Operating System), 데이터베이스 관리 시스템 (DataBase Management System), 미들웨어(Middle Ware) 등을 선정할 때 고려해야 할 사항을 기술하고, 오픈 소스 사용 시 주의해야 할 내용을 제시한다.

운영체제 (OS, Operating System)	<ul style="list-style-type: none"> <li>• 컴퓨터 시스템의 자원들을 효율적으로 관리하며, 사용자가 컴퓨터를 편리하고 효율적으로 사용할 수 있도록 환경을 제공하는 소프트웨어</li> <li>• 운영체제 관련 요구사항 식별 시 고려사항 : 가용성, 성능, 기술 지원, 주변 기기, 구축 비용</li> </ul>
데이터베이스 관리 시스템 (DBMS)	<ul style="list-style-type: none"> <li>• 사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성해 주고, 데이터베이스를 관리해 주는 소프트웨어</li> <li>• DBMS 관련 요구사항 식별 시 고려사항 : 가용성, 성능, 기술 지원, 상호 호환성, 구축 비용</li> </ul>
웹 애플리케이션 서버(WAS; Web Application Server)	<ul style="list-style-type: none"> <li>• 정적인 콘텐츠 처리를 하는 웹 서버와 달리 사용자의 요구에 따라 변하는 동적인 콘텐츠를 처리하기 위해 사용되는 미들웨어</li> <li>• WAS 관련 요구사항 식별 시 고려사항 : 가용성, 성능, 기술 지원, 구축 비용</li> </ul>
오픈 소스 (Open Source)	<ul style="list-style-type: none"> <li>• 누구나 별다른 제한 없이 사용할 수 있도록 소스 코드를 공개한 것으로 오픈 소스 라이선스를 만족하는 소프트웨어</li> <li>• 오픈 소스 사용에 따른 고려사항 : 라이선스의 종류, 사용자 수, 기술의 지속 가능성</li> </ul>

#### 2020년 1, 2회 기사 필기

##### 1. DBMS 관련 요구사항 식별시 고려 사항을 쓰시오.

답 :

##### 2. 웹 애플리케이션 서버(WAS; Web Application Server)의 개념을 쓰시오.

답 :

#### **정답** 1. 가용성, 성능, 기술 지원, 상호 호환성, 구축 비용

2. 웹 애플리케이션 서버는 정적인 콘텐츠 처리를 하는 웹 서버와 달리 사용자의 요구에 따라 변하는 동적인 콘텐츠를 처리하기 위해 사용되는 미들웨어이다.

**[핵심 032] 요구사항 유형**

기능 요구사항	시스템이 갖춰야 할 필수적인 기능에 대한 요구사항
비기능 요구사항	필수 기능 외의 품질이나 제약사항에 관한 요구사항
사용자 요구사항	사용자 관점에서 본 시스템이 제공해야 할 요구사항
시스템 요구사항	개발자 관점에서 본 시스템 전체가 사용자 와 다른 시스템에 제공해야 할 요구사항

1. 소프트웨어가 어떤 문제를 해결하기 위해 제공하는 서비스에 대한 설명과 정상적으로 운영되는데 필요한 제약조건 등을 나타내는 요구사항의 유형 4가지를 쓰시오.

답 :

**정답** 1. 기능 요구사항, 비기능 요구사항, 사용자 요구사항, 시스템 요구사항

**[핵심 033] 요구사항 개발 프로세스**

요구사항 도출 (Requirement Elicitation)	<ul style="list-style-type: none"> <li>시스템, 사용자, 그리고 시스템 개발에 관련된 사람들이 서로 의견을 교환하여 요구사항이 어디에 있는지, 어떻게 수집할 것인지를 식별하고 이해하는 과정</li> <li>주요 기법 : 인터뷰, 설문, 브레인스토밍, 워크샵, 프로토타이핑, 유스케이스</li> </ul>
요구사항 분석 (Requirement Analysis)	<ul style="list-style-type: none"> <li>개발 대상에 대한 사용자의 요구사항 중 명확하지 않거나 모호하여 이해되지 않는 부분을 발견하고 이를 걸러내기 위한 과정</li> <li>비용과 일정에 대한 제약 설정, 타당성 조사</li> </ul>
요구사항 명세 (Requirement Specification)	요구사항을 체계적으로 분석한 후 승인될 수 있도록 문서화하는 것
요구사항 확인 (Requirement Validation)	개발 자원을 요구사항에 할당하기 전에 요구사항 명세서가 정확하고 완전하게 작성되었는지를 검토하는 활동

1. 요구사항 개발 프로세스 중 시스템, 사용자, 그리고 시스템 개발에 관련된 사람들이 서로 의견을 교환하여 요구사항이 어디에 있는지, 어떻게 수집할 것인지를 식별하고 이해하는 요구사항 개발 프로세스를 쓰시오.

답 :

**정답** 1. 요구사항 도출(Requirement Elicitation)

**[핵심 034] 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 요구사항을 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 한다.
  - 개념 모델 종류 : 유스케이스 다이어그램, 데이터 흐름 모델, 상태 모델, 목표기반 모델, 사용자 인터랙션, 객체 모델, 데이터 모델 등
- 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

**잠깐만요 ! 자료 흐름도(DFD; Data Flow Diagram)**

- 요구사항 분석에서 자료의 흐름 및 변환 과정과 기능을 도형 중심으로 기술하는 방법으로 자료 흐름 그래프, 버블 차트라고도 합니다.
- 자료 흐름도 구성 요소 표기법

**프로세스  
(Process)**

- 자료를 변환시키는 시스템의 한 부분(처리 과정)을 나타내며 처리, 기능, 변환, 버블이라고 함
- 원이나 둥근 사각형으로 표시하고 그 안에 프로세스 이름을 기입함

자료 흐름 (Data Flow)	<ul style="list-style-type: none"> <li>• 자료의 이동(흐름)이나 연관관계를 나타냄</li> <li>• 화살표 위에 자료의 이름을 기입함</li> </ul>
자료 저장소 (Data Store)	<ul style="list-style-type: none"> <li>• 시스템에서의 자료 저장소(파일, 데이터베이스)를 나타냄</li> <li>• 도형 안에 자료 저장소 이름을 기입함</li> </ul>
단말 (Terminator)	<ul style="list-style-type: none"> <li>• 시스템과 교신하는 외부 개체로, 입력 데이터가 만들어지고 출력 데이터를 받음(정보의 생산자와 소비자)</li> <li>• 도형 안에 이름을 기입함</li> </ul>

1. 개발 대상에 대한 사용자의 요구사항 중 명확하지 않거나 모호한 부분을 걸러내기 위한 요구사항 분석 기법 중 정형 분석(Formal Analysis)의 개념을 설명하시오.

답 :

2. 유스케이스 다이어그램, 사용자 인터랙션, 데이터 흐름 모델과 관계있는 요구사항 분석 기법을 쓰시오.

답 :

2020년 1, 2회 기사 필기

3. 데이터 흐름도(DFD)의 구성 요소를 쓰시오.

답 :

**정답** 1. 정형 분석은 구문(Syntax)과 의미(Semantics)를 갖는 정형화된 언어를 이용해 요구사항을 수학적 기호로 표현한 후 이를 분석하는 과정이다.

2. 개념 모델링(Conceptual Modeling)

3. 프로세스(Process), 자료 흐름(Data Flow), 자료 저장소(Data Store), 단말(Terminator)

### [핵심 035] 요구사항 확인 기법

- 요구사항 개발 과정을 거쳐 문서화된 요구사항 관련 내용을 확인하고 검증하는 방법이다.
- 요구사항 검토(Requirement Reviews) : 문서화된 요구사항을 훑어보면서 확인하는 것으로 가장 일반적인 요구사항 검증 방법

### • 프로토타이핑(Prototyping)

- 초기 도출된 요구사항을 토대로 프로토타입(Prototype)을 만든 후 대상 시스템의 개발이 진행되는 동안 도출되는 요구사항을 반영하면서 지속적으로 프로토타입을 재작성하는 과정
- 프로토타입 : 상품이나 서비스가 출시되기 전에 개발 대상 시스템 또는 그 일부분을 개략적으로 만든 원형

• 모델 검증(Model Verification) : 요구사항 분석 단계에서 개발된 모델이 요구사항을 충족시키는지 검증하는 것

• 인수 테스트(Acceptance Test) : 사용자가 실제로 사용될 환경에서 요구사항들이 모두 충족되는지 사용자 입장에서 확인하는 과정

1. 요구사항 개발 과정을 거쳐 문서화된 요구사항 관련 내용을 확인하고 검증하는 방법인 요구사항 확인 기법 중 사용자가 실제로 사용될 환경에서 요구사항들이 모두 충족되는지 사용자 입장에서 확인하는 과정이 무엇인지 쓰시오.

답 :

2. 상품이나 서비스가 출시되기 전에 개발 대상 시스템 또는 그 일부분을 개략적으로 만든 원형을 무엇이라고 하는지 쓰시오.

답 :

**정답** 1. 인수 테스트(Acceptance Test) 2. 프로토타입(Prototype)



### [핵심 036] UML(Unified Modeling Language)

- 시스템 분석, 설계, 구현 등 시스템 개발 과정에서 시스템 개발자와 고객 또는 개발자 상호 간의 의사소통이 원활하게 이루어지도록 표준화한 대표적인 객체지향 모델링 언어이다.

## • UML의 구성 요소

사물(Things)	<ul style="list-style-type: none"> <li>모델을 구성하는 가장 중요한 기본 요소로, 다이어그램 안에서 관계가 형성될 수 있는 대상들</li> <li>구조(Structural) 사물 : 시스템의 개념적, 물리적 요소를 표현</li> <li>행동(Behavioral) 사물 : 시간과 공간에 따른 요소들의 행위를 표현</li> <li>그룹(Grouping) 사물 : 요소들을 그룹으로 묶어서 표현</li> <li>주해(Annotation) 사물 : 부가적인 설명이나 제약조건 등을 표현</li> </ul>
관계(Relationships)	<ul style="list-style-type: none"> <li>사물과 사물 사이의 연관성을 표현하는 것</li> <li>연관(Association) 관계 : 2개 이상의 사물이 서로 관련되어 있음을 표현함</li> <li>집합(Aggregation) 관계 : 하나의 사물이 다른 사물에 포함되어 있는 관계를 표현함</li> <li>포함(Composition) 관계 : 집합 관계의 특수한 형태로, 포함하는 사물의 변화가 포함되는 사물에게 영향을 미치는 관계를 표현함</li> <li>일반화(Generalization) 관계 : 하나의 사물이 다른 사물에 비해 더 일반적인지 구체적인지를 표현함</li> <li>의존(Dependency) 관계 : 연관 관계와 같이 사물 사이에 서로 연관은 있으나 필요에 의해 서로에게 영향을 주는 짧은 시간 동안만 연관을 유지하는 관계를 표현함</li> <li>실체화(Realization) 관계 : 사물이 할 수 있거나 해야 하는 기능(행위, 인터페이스)으로 서로를 그룹화 할 수 있는 관계를 표현함</li> </ul>
다이어그램(Diagram)	<ul style="list-style-type: none"> <li>사물과 관계를 도형으로 표현한 것</li> <li>정적 모델링에서는 주로 구조적 다이어그램을 사용하고 동적 모델링에서는 주로 행위 다이어그램을 사용함               <ul style="list-style-type: none"> <li>정적 모델링 : 사용자가 요구한 기능을 구현하는데 필요한 자료들의 논리적인 구조를 표현한 것</li> <li>동적 모델링 : 시스템의 내부 구성 요소들의 상태가 시간의 흐름에 따라 변화하는 과정과 변화하는 과정에서 발생하는 상호 작용을 표현한 것</li> </ul> </li> </ul>

### 1. UML의 구성 요소 3가지를 쓰시오.

답 :

2. 다음은 UML의 구성 요소들에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 구성 요소를 쓰시오.

관계(Relationships)	사물과 사물 사이의 연관성을 표현하는 것이다.
( ① )	모델을 구성하는 가장 중요한 기본 요소로, 다이어그램 안에서 관계가 형성될 수 있는 대상들이다.
( ② )	사물과 관계를 도형으로 표현한 것이다.

답

① :

② :

3. UML 구성 요소 중 사물과 사물 사이의 연관성을 표현하는 것은 관계이다. 다음은 관계 종류 중 무엇에 대한 설명인지 쓰시오.

사물이 할 수 있거나 해야 하는 기능(행위, 인터페이스)으로 서로를 그룹화 할 수 있는 관계를 표현한다.

답 :

**정답** 1. 사물(Things), 관계(Relationships), 다이어그램(Diagram)

2. ① 사물(Things) ② 다이어그램(Diagram)

3. 실체화(Realization) 관계

### [핵심 037] 구조적(Structural) 다이어그램

#### • 클래스(Class) 다이어그램

- 클래스와 클래스가 가지는 속성, 클래스 사이의 관계를 표현한다.
- UML을 이용한 정적 모델링의 대표적인 것이 클래스 다이어그램이다.
- 클래스 다이어그램의 구성 요소



클래스(Class)	<ul style="list-style-type: none"> <li>• 각각의 객체들이 갖는 속성과 오퍼레이션(동작)을 표현</li> <li>• 일반적으로 3개의 구획(Compartment)으로 나뉘 클래스의 이름, 속성, 오퍼레이션을 표기함</li> <li>• 속성(Attribute) : 클래스의 상태나 정보를 표현</li> <li>• 오퍼레이션(Operation, 연산) : 클래스가 수행할 수 있는 동작으로, 함수(메소드, Method)라고도 함</li> </ul>
제약조건	속성에 입력될 값에 대한 제약조건이나 오퍼레이션 수행 전후에 지정해야 할 조건이 있다면 이를 적음
관계(Relationships)	클래스와 클래스 사이의 연관성을 표현

- 객체(Object) 다이어그램 : 클래스에 속한 사물(객체)들, 즉 인스턴스(Instance)를 특정 시점의 객체와 객체 사이의 관계로 표현
- 컴포넌트(Component) 다이어그램 : 실제 구현 모듈인 컴포넌트 간의 관계나 컴포넌트 간의 인터페이스를 표현
- 배치(Deployment) 다이어그램 : 결과물, 프로세스, 컴포넌트 등 물리적 요소들의 위치를 표현
- 복합체 구조(Composite Structure) 다이어그램 : 클래스나 컴포넌트가 복합 구조를 갖는 경우 그 내부 구조를 표현
- 패키지(Package) 다이어그램 : 유스케이스나 클래스 등의 모델 요소들을 그룹화한 패키지들의 관계를 표현

2020년 1, 2회 기사 필기

1. Class Diagram, Object Diagram, Component Diagram, Activity Diagram 중 구조적 다이어그램이 아닌 것을 쓰시오.

답 :

2. 다음은 클래스 다이어그램의 클래스에 대한 설명이다. 괄호에 들어갈 가장 적합한 용어를 쓰시오.

클래스 다이어그램의 클래스는 각각의 객체들이 갖는 ( ① ) 과 ( ② )을 표현한다. ( ① )은 클래스의 상태나 정보를 표현하고, ( ② )은 클래스가 수행할 수 있는 동작으로, 함수(메소드, Method)라고도 한다.

답

- ① :  
② :

정답 1. Activity Diagram

2. ① 속성(Attribute) ② 오퍼레이션(Operation)

### [핵심 038] 시퀀스 다이어그램

- 행위(Behavioral) 다이어그램에는 동적 모델링인 시퀀스 다이어그램, 커뮤니케이션 다이어그램, 상태 다이어그램, 기능 모델링인 유스케이스(Use Case) 다이어그램, 활동(Activity) 다이어그램 등이 있다.
- 시퀀스(Sequence) 다이어그램
  - 시스템이나 객체들이 메시지를 주고받으며 시간의 흐름에 따라 상호 작용하는 과정을 그림으로 표현한 것이다.
  - 시퀀스 다이어그램의 구성 요소 : 액터, 객체, 라이프라인, 활성 상자, 메시지, 객체 소멸, 프레임 등

액터(Actor)	시스템으로부터 서비스를 요청하는 외부 요소로, 사람이나 외부 시스템
객체(Object)	<ul style="list-style-type: none"> <li>• 메시지를 주고받는 주체</li> <li>• 콜론(:)을 기준으로 앞쪽에는 객체명을 뒤 쪽에는 클래스명 기술</li> </ul>
라이프라인(Lifeline)	객체가 메모리에 존재하는 기간으로, 객체 아래쪽에 점선을 그어 표현
활성 상자(Activation Box)	객체가 메시지를 주고받으며 구동되고 있음을 라이프라인 상에 겹쳐 직사각형 형태로 표현
메시지(Message)	객체가 상호 작용을 위해 주고받는 메시지
객체 소멸	라이프라인 상에서 객체 소멸 표시를 만나면 해당 객체는 더 이상 메모리에 존재하지 않음을 의미
프레임(Frame)	다이어그램의 전체 또는 일부를 묶어 표현

1. 행위 다이어그램 중 동적 모델링에 해당하는 다이어그램 3가지를 쓰시오.

답 :

2. 시스템이나 객체들이 메시지를 주고 받으며 상호 작용하는 과정을 표현하는 다이어그램이 무엇인지 쓰시오.

답 :

**정답 1.** 시퀀스 다이어그램(Sequence Diagram), 커뮤니케이션 다이어그램(Communication Diagram), 상태 다이어그램(State Diagram)

2. 시퀀스 다이어그램(Sequence Diagram)

**[핵심 039] 커뮤니케이션 다이어그램 / 상태 다이어그램**

**커뮤니케이션(Communication) 다이어그램**

- 시퀀스 다이어그램과 같이 동작에 참여하는 객체들이 주고받는 메시지를 표현하는데, 메시지뿐만 아니라 객체들 간의 연관까지 표현한다.
- 커뮤니케이션 다이어그램의 구성 요소 : 액터, 객체, 링크, 메시지 등

링크(Link)	<ul style="list-style-type: none"> <li>• 객체들 간의 관계를 표현하는 데 사용</li> <li>• 액터와 객체, 객체와 객체 간에 실선을 그어 표현</li> </ul>
----------	---

**상태(State) 다이어그램**

- 하나의 객체가 자신이 속한 클래스의 상태 변화 혹은 다른 객체와의 상호 작용에 따라 상태가 어떻게 변화하는지를 표현한다.
- 상태 다이어그램의 구성 요소 : 상태, 이벤트, 상태 전환 등

상태(State)	<ul style="list-style-type: none"> <li>• 객체의 상태를 둥근 사각형 안에 기술</li> <li>• 시작 상태 : 상태의 시작을 속이 채워진 원(●)으로 표현</li> <li>• 종료 상태 : 상태의 종료를 속이 채워진 원을 둘러싼 원(◎)으로 표현</li> </ul>
이벤트(Event)	조건, 외부 신호, 시간의 흐름 등 상태에 변화를 주는 현상
상태 전환	상태 사이의 흐름, 변화를 화살표로 표현
프레임(Frame)	상태 다이어그램의 범위를 표현

**1.** 동작에 참여하는 객체들이 주고받는 메시지뿐만 아니라 객체들 간의 연관까지 표현하는 다이어그램을 쓰시오.

**답 :**

**정답 1.** 커뮤니케이션 다이어그램(Communication Diagram)

**[핵심 040] 유스케이스 다이어그램**

- 개발될 시스템과 관련된 외부 요소들, 즉 사용자와 다른 외부 시스템들이 개발될 시스템을 이용해 수행할 수 있는 기능을 사용자의 관점(View)에서 표현한 것이다.
- 유스케이스 다이어그램의 구성 요소 : 시스템 범위, 액터, 유스케이스, 관계

시스템 범위(System Scope)	시스템 내부에서 수행되는 기능들을 외부 시스템과 구분하기 위해 시스템 내부의 유스케이스들을 사각형으로 묶어 시스템의 범위를 표현
액터(Actor)	시스템과 상호작용을 하는 모든 외부 요소로, 사람이나 외부 시스템을 의미
유스케이스(Use Case)	사용자가 보는 관점에서 시스템이 액터에게 제공하는 서비스 또는 기능을 표현한 것
관계(Relationship)	유스케이스 다이어그램에서 관계는 액터와 유스케이스, 유스케이스와 유스케이스 사이에서 나타날 수 있으며, 포함 관계, 확장 관계, 일반화 관계의 3종류가 있음

**1.** 유스케이스 다이어그램의 구성 요소 4가지를 쓰시오.

**답 :**

**정답 1.** 시스템 범위(System Scope), 액터(Actor), 유스케이스(Use Case), 관계(Relationship)

**[핵심 041] 활동 다이어그램**

- 자료 흐름도와 유사한 것으로, 사용자의 관점(View)에서 시스템이 수행하는 기능을 처리 흐름에 따라 순서대로 표현한 것이다.
- 활동 다이어그램의 구성 요소 : 액션, 액티비티, 노드, 스윔레인 등

액션(Action)/ 액티비티(Activity)	<ul style="list-style-type: none"> <li>• 액션(Action)은 더 이상 분해할 수 없는 단일 작업</li> <li>• 액티비티(Activity)는 몇 개의 액션으로 분리될 수 있는 작업</li> </ul>
-------------------------------	--

노드	<ul style="list-style-type: none"> <li>• 시작 노드 : 액션이나 액티비티가 시작됨을 의미</li> <li>• 종료 노드 : 액티비티 안의 모든 흐름이 종료됨을 의미</li> <li>• 조건(판단) 노드 : 조건에 따라 제어의 흐름이 분리됨을 표현</li> <li>• 병합 노드 : 여러 경로의 흐름이 하나로 합쳐짐을 표현</li> <li>• 포크(Fork) 노드 : 액티비티의 흐름이 분리되어 수행됨을 표현</li> <li>• 조인(Join) 노드 : 분리되어 수행되던 액티비티의 흐름이 다시 합쳐짐을 표현</li> </ul>
스윔레인 (Swim Lane)	액티비티 수행을 담당하는 주체를 구분

1. 시스템이 어떤 기능을 수행하는지 객체의 처리 로직이나 조건에 따른 처리의 흐름을 순서에 따라 표현하는 다이어그램이 무엇인지 쓰시오.

답 :

**정답** 1. 활동 다이어그램(Activity Diagram)

### 3장 | 데이터 입 · 출력 구현

#### [핵심 042] 데이터 모델

- 현실 세계의 정보들을 컴퓨터에 표현하기 위해서 단순화, 추상화하여 체계적으로 표현한 개념적 모형이다.
- 데이터 모델의 구성 요소
  - 개체(Entity) : 데이터베이스에 표현하려는 것으로, 사람이 생각하는 개념이나 정보 단위 같은 현실 세계의 대상체
  - 속성(Attribute) : 데이터의 가장 작은 논리적 단위로서 파일 구조상의 데이터 항목 또는 데이터 필드에 해당함
  - 관계(Relationship) : 개체 간의 관계 또는 속성 간의 논리적인 연결을 의미함

1. 다음은 데이터 모델의 구성 요소들에 대한 설명이다. 괄호 ①~③에 들어갈 알맞은 구성 요소를 쓰시오.

- ( ① )은 데이터의 가장 작은 논리적 단위로서 파일 구조상의 데이터 항목 또는 데이터 필드에 해당한다.
- ( ② )는 개체 간의 관계 또는 속성 간의 논리적인 연결을 의미한다.
- ( ③ )는 데이터베이스에 표현하려는 것으로, 사람이 생각하는 개념이나 정보 단위 같은 현실 세계의 대상체이다.

답

- ① :  
② :  
③ :

**정답** 1. ① 속성(Attribute) ② 관계(Relationship) ③ 개체(Entity)

#### [핵심 043] 데이터 모델의 종류

- 개념적 데이터 모델 : 현실 세계에 대한 인간의 이해를 돕기 위해 현실 세계에 대한 인식을 추상적 개념으로 표현하는 과정
- 논리적 데이터 모델 : 개념적 모델링 과정에서 얻은 개념적 구조를 컴퓨터가 이해하고 처리할 수 있는 컴퓨터 세계의 환경에 맞도록 변환하는 과정



- 물리적 데이터 모델 : 실제 컴퓨터에 데이터가 저장되는 방법을 정의하는 물리 데이터베이스 설계 과정

1. 데이터 모델의 종류 중 개념적 모델링 과정에서 얻은 개념적 구조를 컴퓨터가 이해하고 처리할 수 있는 컴퓨터 세계의 환경에 맞추도록 변환하는 과정을 의미하는 것은 무엇인지 쓰시오.

답 :

정답 1. 논리적 데이터 모델

#### [핵심 044] 데이터 모델에 표시할 요소

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

1. 데이터 모델은 현실 세계의 정보들을 컴퓨터에 표현하기 위해서 단순화, 추상화하여 체계적으로 표현한 개념적 모형이다. 데이터 모델에 표시할 요소 세 가지를 쓰시오.

답 :

정답 1. 구조(Structure), 연산(Operation), 제약 조건(Constraint)

#### [핵심 045] 이상(Anomaly)

- 테이블에서 일부 속성들의 종속으로 인해 데이터의 중복(Redundancy)이 발생하고, 이 중복으로 인해 테이블 조작 시 문제가 발생하는 현상을 의미한다.
- 이상의 종류
  - 삽입 이상(Insertion Anomaly) : 테이블에 데이터를 삽입할 때 의도와는 상관없이 원하지 않은 값들로 인해 삽입할 수 없게 되는 현상
  - 삭제 이상(Deletion Anomaly) : 테이블에서 한 튜플을 삭제할 때 의도와는 상관없는 값들도 함께 삭제되는, 즉 연쇄 삭제가 발생하는 현상

- 갱신 이상(Update Anomaly) : 테이블에서 튜플에 있는 속성 값을 갱신할 때 일부 튜플의 정보만 갱신되어 정보에 불일치성(Inconsistency)이 생기는 현상

1. 다음은 <출고> 테이블을 분석한 결과이다. <분석 결과>에 해당하는 이상(Anomaly)의 명칭을 쓰시오.

<출고>

출고번호	제품번호	출고가	출고량
1001	11	15,000	150
1002	22	16,000	200
1003	33	12,000	100
1004	44	18,000	150

<분석 결과>

- ① 어느 한 제품에 대한 출고를 삭제하면 제품 정보도 함께 삭제된다. 예를 들어 출고번호 1002를 삭제하면 제품번호 22에 대한 제품 정보까지 모두 삭제된다.
- ② 제품번호 33의 출고가를 변경하려면 그 제품번호가 포함된 모든 출고들의 출고가를 변경해야 한다.
- ③ 제품번호가 55인 새로운 출고 내용을 입력하기 위해서는 반드시 출고번호가 있어야 한다.

답

- ① :
- ② :
- ③ :

2. 이상(Anomaly) 현상이 발생하는 원인을 설명하시오.

답 :

정답 1. ① 삭제 이상(Deletion Anomaly) ② 갱신 이상(Update Anomaly) ③ 삽입 이상(Insertion Anomaly)

2. 이상은 테이블에서 일부 속성들의 종속으로 인해 데이터의 중복이 발생하기 때문에 발생한다.

### [핵심 046] 함수적 종속(Functional Dependency)

- 어떤 테이블 R에서 X와 Y를 각각 R의 속성 집합의 부분 집합이라 할 때 속성 X의 값 각각에 대해 시간에 관계없이 항상 속성 Y의 값이 오직 하나만 연관되어 있을 때 Y는 X에 함수적 종속 또는 X가 Y를 함수적으로 결정한다고 하고,  $X \rightarrow Y$ 로 표기한다.
- $X \rightarrow Y$ 의 관계를 갖는 속성 X와 Y에서 X를 결정자(Determinant)라 하고, Y를 종속자(Dependent)라고 한다.
- 완전 함수적 종속 : 어떤 테이블 R에서 속성 Y가 다른 속성 집합 X 전체에 대해 함수적 종속이면서 속성 집합 X의 어떠한 진부분 집합 Z(즉,  $Z \subset X$ )에도 함수적 종속이 아닐 때 속성 Y는 속성 집합 X에 완전 함수적 종속이라고 함
- 부분 함수적 종속 : 어떤 테이블 R에서 속성 Y가 다른 속성 집합 X 전체에 대해 함수적 종속이면서 속성 집합 X의 임의의 진부분 집합에 대해 함수적 종속일 때, 속성 Y는 속성 집합 X에 부분 함수적 종속이라고 함

1. 데이터베이스와 관련된 다음 설명에서 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오.

어떤 테이블 R에서 X와 Y를 각각 R의 속성 집합의 부분 집합이라 할 때 속성 X의 값 각각에 대해 시간에 관계없이 항상 속성 Y의 값이 오직 하나만 연관되어 있을 때 Y는 X에 ( ① )이라고 하고, ( ② )로 표기한다.

답

- ① :  
② :

2. 데이터베이스와 관련된 다음 설명에서 괄호(①, ②)에 들어갈 알맞은 함수적 종속의 종류를 쓰시오.

( ① )이라는 말은 어떤 속성이 기본키에 대해 완전히 종속적일 때를 말한다. 예를 들어 <수강> 테이블은 (학번, 과목번호)가 기본키이고, 성적은 학번과 과목번호에 의해서만 결정된다면 성적은 기본키(학번, 과목번호)에 ( ① )이라고 한다. 반면 학년은 과목번호에 관계없이 학년이 같으면 항상 같은 학년이 온다면, 즉 기본키의 일부인 학년에 의해서 학년이 결정된다면 학년은 ( ② )이라고 한다.

답

- ① :  
② :

정답 1. ① 함수적 종속 ②  $X \rightarrow Y$

2. ① 완전 함수적 종속 ② 부분 함수적 종속



### [핵심 047] 정규화(Normalization)

- 테이블의 속성들이 상호 종속적인 관계를 갖는 특성을 이용하여 테이블을 무손실 분해하는 과정이다.

#### 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

4NF

↓ 조인 종속성 이용

5NF

#### 정규화 단계 암기 요령

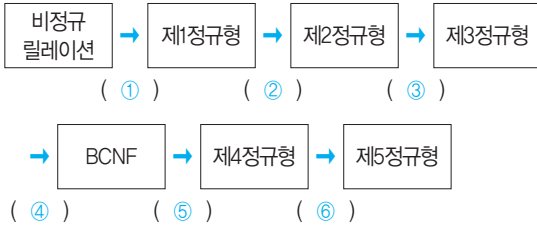
두부를 좋아하는 정규화가 두부가 가게에 가서 가게에 있는 두부를 다 달라고 말하니 주인이 깜짝 놀라며 말했다.

#### 두부이걸다줘? ≡ 도부이걸다조

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

※ 이행적 함수적 종속 :  $A \rightarrow B$ 이고  $B \rightarrow C$ 일 때  $A \rightarrow C$ 를 만족하는 관계를 이행적 함수적 종속이라고 함

1. 다음은 정규화 과정을 나열한 것이다. 괄호(①~⑥)에 각 정규화 과정별로 필요한 작업을 쓰시오.



답

- ① :  
② :  
③ :  
④ :  
⑤ :  
⑥ :

2020년 1, 2회 기사 필기

2. 다음은 이행적 함수 종속에 대한 설명이다. 괄호에 들어갈 알맞은 기호를 쓰시오.

이행적 함수 종속은  $A \rightarrow B$ 이고  $B \rightarrow C$ 일 때 ( )를 만족하는 관계를 의미한다.

답 :

**정답** 1. ① 도메인을 원자 값만으로 구성 ② 부분적 함수적 종속 제거  
③ 이행적 함수적 종속 제거 ④ 결정자이면서 후보키가 아닌 것 제거  
⑤ 다치 종속 ⑥ 조인 종속성 이용

2.  $A \rightarrow C$

#### [핵심 048] 논리 데이터 모델의 물리 데이터 모델로 변환

- 엔티티(Entity)를 테이블로 변환 : 논리 데이터 모델에서 정의된 엔티티를 물리 데이터 모델의 테이블로 변환
- 슈퍼타입 기준 테이블 변환 : 서브타입을 슈퍼타입에 통합하여 하나의 테이블로 변환
- 서브타입 기준 테이블 변환 : 슈퍼타입 속성들을 각각의 서브타입에 추가하여 서브타입들을 개별적인 테이블로 변환
- 개별타입 기준 테이블 변환 : 슈퍼타입과 서브타입들을 각각의 개별적인 테이블로 변환

- 속성을 컬럼으로 변환 : 논리 데이터 모델에서 정의한 속성을 물리 데이터 모델의 컬럼으로 변환
- 관계를 외래키로 변환 : 논리 데이터 모델에서 정의된 관계는 기본키와 이를 참조하는 외래키로 변환

1. 다음 설명의 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오.

- 개별타입 기준의 테이블 변환은 ( ① )과 ( ② )들을 각각의 개별적인 테이블로 변환하는 것으로, 두 테이블들 사이에는 각각 1:1 관계가 형성된다.
- ( ① ) 기준의 테이블 변환은 ( ② )을 ( ① )에 통합하여 하나의 테이블로 만드는 것이고, ( ② ) 기준의 테이블 변환은 ( ① ) 속성들을 각각의 ( ② )에 추가하여 ( ② )들을 개별적인 테이블로 만드는 것이다.

답

- ① :  
② :

**정답** 1. ① 슈퍼타입 ② 서브타입

#### [핵심 049] 반정규화(Denormalization)

- 시스템의 성능 향상, 개발 및 운영의 편의성 등을 위해 정규화된 데이터 모델을 통합, 중복, 분리하는 과정으로, 의도적으로 정규화 원칙을 위배하는 행위이다.
- 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	<ul style="list-style-type: none"> <li>수평 분할(Horizontal Partitioning) : 레코드(Record)를 기준으로 테이블을 분할하는 것으로, 레코드별로 사용 빈도의 차이가 큰 경우 사용 빈도에 따라 테이블을 분할함</li> <li>수직 분할(Vertical Partitioning) : 하나의 테이블에 속성이 너무 많을 경우 속성을 기준으로 테이블을 분할하는 것</li> </ul>

중복 테이블 추가	<ul style="list-style-type: none"> <li>여러 테이블에서 데이터를 추출해서 사용해야 하거나 다른 서버에 저장된 테이블을 이용해야 하는 경우 중복 테이블을 추가하여 작업의 효율성을 향상시킬 수 있음</li> <li>추가 방법 : 집계 테이블의 추가, 진행 테이블의 추가, 특정 부분만을 포함하는 테이블의 추가</li> </ul>
중복 속성 추가	조인해서 데이터를 처리할 때 데이터를 조회하는 경로를 단축하기 위해 자주 사용하는 속성을 하나 더 추가하는 것

2020년 1회 기사 실기

### 1. 비정규화(Denormalization)의 개념을 설명하시오.

답 :

2020년 1, 2회 기사 필기

### 2. 반정규화(Denormalization) 유형 중 중복 테이블을 추가하는 방법 3가지를 쓰시오.

답 :

**정답 1.** 비정규화는 시스템의 성능 향상, 개발 및 운영의 편의성 등을 위해 정규화된 데이터 모델을 통한, 중복, 분리하는 과정이다.

2. 집계 테이블의 추가, 진행 테이블의 추가, 특정 부분만을 포함하는 테이블의 추가

## [핵심 050] 인덱스(Index)

- 데이터 레코드를 빠르게 접근하기 위해 <키 값, 포인터> 쌍으로 구성되는 데이터 구조이다.
- 인덱스 키의 순서에 따라 데이터가 정렬되어 저장되는 방식인 클러스터드 인덱스(Clustered Index)와 인덱스의 키 값만 정렬되어 있을 뿐 실제 데이터는 정렬되지 않는 방식인 언클러스터드 인덱스(Non-Clustered Index)가 있다.
- 인덱스는 인덱스를 구성하는 구조나 특징에 따라 다음과 같이 구분한다.

트리 기반 인덱스	인덱스를 저장하는 블록들이 트리 구조를 이루고 있는 것으로, 상용 DBMS에서는 트리 구조 기반의 B+ 트리 인덱스를 주로 활용함
비트맵 인덱스	인덱스 컬럼의 데이터를 Bit 값인 0 또는 1로 변환하여 인덱스 키로 사용하는 방법

함수 기반 인덱스	컬럼의 값 대신 컬럼에 특정 함수(Function)나 수식(Expression)을 적용하여 산출된 값을 사용하는 것으로, B+ 트리 인덱스 또는 비트맵 인덱스를 생성하여 사용함
비트맵 조인 인덱스	다수의 조인된 객체로 구성된 인덱스로, 단일 객체로 구성된 일반적인 인덱스와 액세스 방법이 다름
도메인 인덱스	개발자가 필요한 인덱스를 직접 만들어 사용하는 것으로, 확장형 인덱스(Extensible Index)라고도 함

1. 인덱스(Index)는 인덱스를 구성하는 특징에 따라 다음과 같이 구분한다. 다음 괄호(①~③)에 들어갈 알맞은 인덱스 방법을 쓰시오.

함수 기반 인덱스	컬럼의 값 대신 컬럼에 특정 Function이나 Expression을 적용하여 산출된 값을 사용하는 것이다.
( ① )	다수의 조인된 객체로 구성된 인덱스로, 단일 객체로 구성된 일반적인 인덱스와 액세스 방법이 다르다.
( ② )	인덱스 컬럼의 데이터를 Bit 값인 0 또는 1로 변환하여 인덱스 키로 사용하는 방법이다.
트리 기반 인덱스	인덱스를 저장하는 블록들이 트리 구조를 이루고 있는 것이다.
( ③ )	개발자가 필요한 인덱스를 직접 만들어 사용하는 것이다.

답

① :

② :

③ :

2. 다음 괄호에 공통적으로 들어갈 데이터베이스 관련 용어를 쓰시오.

- ( )는 데이터 레코드를 빠르게 접근하기 위해 <키 값, 포인터> 쌍으로 구성되는 데이터 구조이다.
- 클러스터드 ( ) : ( ) 키의 순서에 따라 데이터가 정렬되어 저장되는 방식이다.
- 언클러스터드 ( ) : ( )의 키 값만 정렬되어 있을 뿐 실제 데이터는 정렬되지 않는 방식이다.

답 :



**정답** 1. ① 비트맵 조인 인덱스 ② 비트맵 인덱스 ③ 도메인 인덱스  
2. 인덱스(Index)

### [핵심 051] 뷰(View)

- 사용자에게 접근이 허용된 자료만을 제한적으로 보여 주기 위해 하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블이다.
- 저장장치 내에 물리적으로 존재하지 않지만, 사용자에게는 있는 것처럼 간주된다.
- CREATE문으로 정의하고, DROP문으로 제거한다.
- 데이터의 논리적 독립성을 제공한다.

2020년 1, 2회 기사 필기

1. 다음이 설명하는 데이터베이스 관련 용어를 쓰시오.

- 하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블이다.
- 저장장치 내에 물리적으로 존재하지 않는다.
- CREATE문을 사용하여 정의한다.
- 제거할 때는 DROP문을 사용한다.
- 데이터의 논리적 독립성을 제공한다.

답 :

**정답** 1. 뷰(View)

### [핵심 052] 트랜잭션(Transaction)

- 데이터베이스의 상태를 변환시키는 하나의 논리적 기능을 수행하기 위한 작업의 단위 또는 한꺼번에 모두 수행되어야 할 일련의 연산들을 의미한다.
- 특징

Atomicity (원자성)	트랜잭션의 연산은 데이터베이스에 모두 반영되도록 완료(Commit)되든지 아니면 전혀 반영되지 않도록 복구(Rollback)되어야 함
Consistency (일관성)	트랜잭션이 그 실행을 성공적으로 완료하면 언제나 일관성 있는 데이터베이스 상태로 변환함

Isolation (독립성)	둘 이상의 트랜잭션이 동시에 병행 실행되는 경우 어느 하나의 트랜잭션 실행중에 다른 트랜잭션의 연산이 끼어들 수 없음
Durability (지속성)	성공적으로 완료된 트랜잭션의 결과는 시스템이 고장나더라도 영구적으로 반영되어야 함

2020년 1회 기사 실기

1. 다음은 트랜잭션의 특징이다. 괄호(①, ②)에 들어갈 알맞은 특징을 쓰시오.

- 일관성 : 트랜잭션이 그 실행을 성공적으로 완료하면 언제나 일관성 있는 데이터베이스 상태로 변환한다.
- ( ① ) : 트랜잭션의 연산은 데이터베이스에 모두 반영되도록 완료(Commit)되든지 아니면 전혀 반영되지 않도록 복구(Rollback)되어야 한다(All or Nothing).
- 지속성 : 성공적으로 완료된 트랜잭션의 결과는 시스템이 고장나더라도 영구적으로 반영되어야 한다.
- ( ② ) : 둘 이상의 트랜잭션이 동시에 병행 실행되는 경우 어느 하나의 트랜잭션 실행중에 다른 트랜잭션의 연산이 끼어들 수 없다.

답

- ① :  
② :

**정답** 1. ① 원자성(Atomicity) ② 독립성(Isolation)

### [핵심 053] 클러스터(Cluster)

- 데이터 저장 시 데이터 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법이다.
- 클러스터링으로 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.
- 처리 범위가 넓은 경우에는 단일 테이블 클러스터링을, 조인이 많이 발생하는 경우에는 다중 테이블 클러스터링을 사용한다.



1. 데이터 저장 시 데이터 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법이 무엇인지 쓰시오.

답 :

정답 1. 클러스터(Cluster)

### [핵심 054] 파티션(Partition)

- 대용량의 테이블이나 인덱스를 작은 논리적 단위로 나누는 것을 말한다.
- 파티셔닝 방식에 따른 파티션의 종류

범위 분할 (Range Partitioning)	지정한 열의 값을 기준으로 분할
해시 분할 (Hash Partitioning)	해시 함수를 적용한 결과 값에 따라 데이터를 분할
조합 분할 (Composite Partitioning)	범위 분할로 분할한 다음 해시 함수를 적용하여 다시 분할

- 인덱스 파티션 : 파티션된 테이블의 데이터를 관리하기 위해 인덱스를 나눈 것

1. ( )은 데이터베이스에서 테이블이나 인덱스를 작은 논리적 단위로 나누는 것으로, ( ) 별로 데이터가 분산되어 저장되므로 디스크의 성능이 향상된다. 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오.

답 :

2. 데이터베이스에서 파티션은 대용량의 테이블이나 인덱스를 작은 논리적 단위로 나누는 것을 말한다. 다음 괄호(①~③)에 들어갈 알맞은 파티션 종류를 쓰시오.

- ( ① )은 범위 분할로 분할한 다음 해시 함수를 적용하여 다시 분할하는 방식이다.
- ( ② )은 일별, 월별, 분기별 등 지정한 열의 값을 기준으로 분할하는 방식이다.
- ( ③ )은 해시 함수를 적용한 결과 값에 따라 데이터를 분할하는 방식이다.

답

- ① :
- ② :
- ③ :

정답 1. 파티션(Partition)

2. ① 조합 분할(Composite Partitioning) ② 범위 분할(Range Partitioning) ③ 해시 분할(Hash Partitioning)

### [핵심 055] 데이터베이스 용량 설계

- 데이터가 저장될 공간을 정의하는 것이다.
- 데이터베이스 용량을 설계할 때는 테이블에 저장할 데이터양과 인덱스, 클러스터 등이 차지하는 공간 등을 예측하여 반영해야 한다.
- 데이터베이스의 용량을 정확히 산정하여 디스크의 저장 공간을 효과적으로 사용하고 확장성 및 가용성을 높인다.
- 데이터 접근성을 향상시키는 설계 방법
  - 테이블의 테이블스페이스와 인덱스의 테이블스페이스를 분리하여 구성한다.
  - 테이블스페이스와 임시 테이블스페이스를 분리하여 구성한다.
  - 테이블을 마스터 테이블과 트랜잭션 테이블로 분류한다.

1. 다음은 데이터베이스 용량 설계 시 데이터 접근성을 향상시키기 위한 설계 방법이다. 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오.

- 테이블의 ( )와 인덱스의 ( )를 분리하여 구성한다.
- ( )와 임시 ( )를 분리하여 구성한다.
- 테이블을 마스터 테이블과 트랜잭션 테이블로 분류한다.

답 :

정답 1. 테이블스페이스

## 4장 | 통합 구현

### [핵심 056] 통합 구현

- 사용자의 요구사항에 맞춰 송·수신 모듈과 중계 모듈 간의 연계를 구현하는 것을 의미한다.
- 일반적인 통합 구현은 송·수신 시스템과 모듈, 중계 시스템, 연계 데이터, 네트워크로 구성된다.

송신 시스템과 모듈	<ul style="list-style-type: none"> <li>• 송신 시스템은 데이터를 생성 및 변환하여 전송하는 시스템으로, 송신 모듈과 모니터링(Monitoring) 기능으로 구성된다.</li> <li>• 송신 모듈은 전송 데이터를 생성하고 필요에 따라 전송 데이터의 변환 등을 수행한다.</li> <li>• 모니터링 기능은 전송 데이터의 생성부터 송신까지의 과정과 송신 상태 등을 확인한다.</li> </ul>
수신 시스템과 모듈	<ul style="list-style-type: none"> <li>• 수신 시스템은 수신 받은 데이터를 정제 및 변환하는 시스템으로, 수신 모듈과 모니터링(Monitoring) 기능으로 구성된다.</li> <li>• 수신 모듈은 수신 데이터를 정제하고 애플리케이션이나 데이터베이스(DB) 테이블에 적합한 데이터로 변환하는 작업 등을 수행한다.</li> </ul>
중계 시스템	내·외부 시스템 간 또는 내부 시스템 간의 연계 시 사용되는 아키텍처이다.
연계 데이터	<ul style="list-style-type: none"> <li>• 송·수신 시스템 간 송·수신되는 데이터이다.</li> <li>• 연계 데이터에는 속성, 길이(Size), 타입(Type) 등이 포함된다.</li> </ul>
네트워크	송신 시스템, 수신 시스템, 중계 시스템을 연결해주는 통신망이다.

1. 다음은 사용자의 요구사항에 맞춰 송·수신 모듈과 중계 모듈 간의 연계를 구현하는 통합 구현에 대한 설명이다. 괄호(①~③)에 들어갈 알맞은 구성 요소를 쓰시오.

통합 구현은 일반적으로 송신 시스템과 모듈, 수신 시스템과 모듈, ( ① ), ( ② ), ( ③ )로 구성된다. 데이터를 생성 및 변환하여 전송하는 시스템은 송신 시스템, 수신 받은 데이터를 정제 및 변환하는 시스템은 수신 시스템, 내·외부 시스템 간 또는 내부 시스템 간의 연계 시 사용되는 아키텍처는 ( ① ), 송·수신 시스템 간 송·수신되는 데이터는 ( ② ), 송신 시스템, 수신 시스템, 중계 시스템을 연결해주는 통신망은 ( ③ )이다.

답

- ① :  
② :  
③ :

정답 1. ① 중계 시스템 ② 연계 데이터 ③ 네트워크

### [핵심 057] 연계 요구사항 분석

- 통합 구현을 위해 사용자 요구사항을 분석하고 연계 데이터를 식별 및 표준화하여 연계 데이터를 정의하는 것이다.

#### • 연계 요구사항 분석 절차

- ① 시스템 구성도, 응용 애플리케이션 구성 등을 통해 송·수신 시스템의 하드웨어 및 소프트웨어 구성, 네트워크 현황을 확인한다.
  - 시스템 구성도 : 송·수신 시스템의 하드웨어, 소프트웨어, 네트워크 구성 등을 정의한 문서
  - 응용 애플리케이션 구성 : 메뉴 구조, 화면 설계, 데이터의 발생 시점, 발생 유형, 발생 주기 등을 정의한 문서
- ② 송·수신 시스템과 연결할 데이터와 관련된 테이블 정의서, 코드 정의서 등의 문서를 확인한다.
  - 테이블 정의서 : 테이블에서 관리되는 컬럼(Column)들의 특징, 인덱스, 업무 규칙 등을 정의한 문서
  - 코드 정의서 : 데이터베이스에서 코드성 속성을 정의한 문서
- ③ 확인된 시스템의 구성과 데이터 현황 정보 등을 기반으로 체크리스트를 작성한다.
- ④ 시스템 구성도, 응용 애플리케이션 구성, 테이블 정의서, 코드 정의서, 체크리스트 등을 사용자, 시스템 관리자 등과 공유하고 인터뷰나 설문조사를 실시한다.
- ⑤ 인터뷰나 설문조사를 통해 확인된 결과를 기반으로 요구사항의 ID, 이름, 유형, 설명, 해결 방안 등이 포함된 연계 요구사항 분석서(정의서)를 작성한다.

1. 다음은 연계 요구사항 분석시 확인하거나 작성하는 문서에 대한 설명이다. 괄호(①~③)에 들어갈 알맞은 문서를 쓰시오.

시스템 구성도	송·수신 시스템의 하드웨어, 소프트웨어, 네트워크 구성 등을 정의한 문서
코드 정의서	데이터베이스에서 코드성 속성을 정의한 문서
( ① )	메뉴 구조, 화면 설계, 데이터의 발생 시점, 발생 유형, 발생 주기 등을 정의한 문서
( ② )	테이블에서 관리되는 컬럼(Column)들의 특징, 인덱스, 업무 규칙 등을 정의한 문서
( ③ )	인터뷰나 설문조사를 통해 확인된 결과를 기반으로 요구사항의 ID, 이름, 유형, 설명, 해결 방안 등을 정의한 문서

답

- ① :  
② :  
③ :

**정답** 1. ① 응용 애플리케이션 구성 ② 테이블 정의서 ③ 연계 요구사항 분석서

### [핵심 058] 연계 데이터 식별 및 표준화

- 연계 데이터를 구성하기 위해 연계 데이터를 식별하고 식별된 연계 데이터를 표준화한 후 이를 기반으로 연계(인터페이스) 정의서를 작성한다.
- 연계 데이터 식별 및 표준화 절차
  - ① 연계 범위 및 항목 정의 : 데이터 타입, 길이, 코드화 여부 등 시스템 간 연계할 정보를 상세화하여 정의
  - ② 연계 코드 변환 및 매핑 : 정보로 전환하거나 검색 조건으로 활용하기 위해, 연계 정보 중 코드로 관리되어야 할 항목을 찾아 코드로 변환
  - ③ 연계 데이터 식별자와 변경 구분 추가 : 수신 시스템의 연계 정보에 송신 시스템의 식별자(PK;Primary Key), 추가, 수정, 삭제 중 어떤 조작을 수행하는지 식별해주는 구분 정보를 추가하고, 연계 데이터의 확인 및 모니터링을 위해 인터페이스 테이블 또는 파일에 관리 정보를 추가함

④ 연계 데이터 표현 방법 정의 : 연계 대상 범위 및 항목과 코드 변환 및 매핑 방식을 정의한 이후에는 연계 데이터를 구성함

⑤ 연계 정의서 및 명세서 작성

- 연계 정의서는 송·수신 시스템에서 연계할 항목과 연계 데이터의 타입, 길이 등을 구성하고 형식을 정의한 결과를 기반으로 작성한다.
- 연계 정의서에는 송·수신 시스템의 시스템명과 인터페이스 ID, 인터페이스 방식 및 주기 등 송·수신 시스템 간의 인터페이스 현황을 기재한다.
- 연계 명세서는 연계 정의서에 작성한 인터페이스 ID별로 인터페이스 주기 및 방식, DB 및 파일 형식, 송·수신 시스템의 타입, 길이, 식별자, 코드화 여부 등의 세부 사항을 작성한다.

1. 다음 괄호에 공통으로 들어갈 가장 적합한 문서를 쓰시오.

연계 데이터를 구성하기 위해 연계 데이터를 식별하고 식별된 연계 데이터를 표준화한 후 이를 기반으로 ( )를 작성한다. ( )에는 송·수신 시스템의 시스템명과 인터페이스 ID, 인터페이스 방식 및 주기 등 송·수신 시스템 간의 인터페이스 현황을 기재한다.

답 :

**정답** 1. 연계 정의서

### [핵심 059] 연계 메커니즘

- 연계 메커니즘은 데이터의 생성 및 전송을 담당하는 송신 체계와 데이터 수신 및 운영 DB 반영을 담당하는 수신 체계로 구성된다.
- 송신 시스템은 운영 DB로부터 인터페이스 테이블이나 파일(xml, text, csv 등) 형식으로 연계 데이터를 생성하여 송신한다.
- 수신 시스템은 송신 시스템으로부터 전송된 데이터를 받아 수신 시스템에 맞는 데이터로 변환한 후 운영 DB에 반영한다.
- 송·수신 시스템 사이에는 데이터의 송·수신과 송·수신 시스템 현황을 모니터링하는 중계 시스템을 설치할 수 있다.

- 송·수신 시스템과 중계 시스템은 제각기 역할이 중복되지 않도록 아키텍처를 설계한 후 인터페이스 테스트와 통합 테스트를 통해 기능을 검증한다.
- 연계 메커니즘의 연계 방식에는 직접 연계 방식과 간접 연계 방식이 있다.

**1. 데이터의 생성 및 전송을 담당하는 송신 체계와 데이터 수신 및 운영 DB 반영을 담당하는 수신 체계로 구성된 연계 메커니즘의 연결 방식 두 가지를 쓰시오.**

**답 :**

**정답** 1. 직접 연계 방식, 간접 연계 방식

### [핵심 060] 연계 메커니즘의 직접 연계 방식

- 중간 매개체 없이 송·수신 시스템이 직접 연계하는 방식이다.
- 장/단점

장점	단점
<ul style="list-style-type: none"> <li>• 연계 및 통합 구현이 단순하고 용이하다.</li> <li>• 중간 매개체 없이 직접 연계되기 때문에 데이터 연계 처리 성능이 뛰어나다.</li> <li>• 개발 비용이 저렴하다.</li> <li>• 개발 기간이 짧다.</li> </ul>	<ul style="list-style-type: none"> <li>• 송·수신 시스템 간 결합도(Coupling)가 높아 시스템 변경 시 오류가 발생할 수 있다.</li> <li>• 보안을 위한 암호화, 복호화 처리와 비즈니스 로직(Business Logic)을 적용하기 어렵다.</li> <li>• 연계 및 통합이 가능한 시스템 환경이 제한적이다.</li> </ul>

#### • 종류

DB Link	DB에서 제공하는 DB Link 객체를 이용하는 방식
API (Application Programming Interface)/ Open API	데이터를 송신 시스템의 DB에서 읽어와 제공하는 애플리케이션 프로그래밍 인터페이스 ※ API(Application Programming Interface) : 운영체제나 프로그래밍 언어 등에 있는 라이브러리를 응용프로그램 개발 시 이용할 수 있도록 규칙 등에 대해 정의해 놓은 인터페이스 ※ Open API : API의 기능을 누구나 무료로 사용하여 프로그램을 개발하거나 Open API에 새로운 API를 추가할 수 있도록 공개된 API

DB Connection	수신 시스템의 WAS(웹 애플리케이션 서버)에서 송신 시스템의 DB로 연결해주는 방식 ※ WAS : 정적인 콘텐츠를 처리하는 웹 서버와 달리 사용자의 요구에 따라 변하는 동적인 콘텐츠를 처리하기 위해 사용되는 미들웨어
JDBC(Java DataBase Connectivity)	Java에서 DB에 접근하여 데이터를 삽입, 삭제, 수정, 조회할 수 있도록 Java와 DB를 연결해주는 방식

**1. 다음은 연계 메커니즘의 직접 연계 방식의 종류에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 연계 방식을 쓰시오.**

( ① )	수신 시스템의 WAS(웹 애플리케이션 서버)에서 송신 시스템의 DB로 연결해주는 방식
API/ Open AP	데이터를 송신 시스템의 DB에서 읽어와 제공하는 애플리케이션 프로그래밍 인터페이스
( ② )	Java에서 DB에 접근하여 데이터를 삽입, 삭제, 수정, 조회할 수 있도록 Java와 DB를 연결해 주는 방식

**답**

- ① :  
② :

**2. 다음 괄호(①, ②)에 들어갈 가장 적합한 용어를 쓰시오**

데이터를 송신 시스템의 DB에서 읽어와 제공하는 ( ① )는 운영체제나 프로그래밍 언어 등에 있는 라이브러리를 응용프로그램 개발 시 이용할 수 있도록 규칙 등에 대해 정의해 놓은 인터페이스이다. ( ② )는 ( ① )의 기능을 누구나 무료로 사용하여 프로그램을 개발하거나 ( ② )에 새로운 ( ① )를 추가할 수 있도록 공개한 ( ① )이다.

**답**

- ① :  
② :

**정답** 1. ① DB Connection ② JDBC(Java DataBase Connectivity)

2. ① API(Application Programming Interface) ② Open API

### [핵심 061] 연계 메커니즘의 간접 연계 방식

- 송·수신 시스템 사이에 중간 매개체를 두어 연계하는 방식이다.
- 장/단점

장점	단점
<ul style="list-style-type: none"> <li>• 서로 다른 네트워크, 프로토콜 등 다양한 환경의 연계 및 통합이 가능하다.</li> <li>• 송·수신 시스템 간 인터페이스가 변경되어도 오류 없이 서비스가 가능하다.</li> <li>• 보안 품질 보장과 비즈니스 처리를 위한 로직(Logic)을 쉽게 반영할 수 있다.</li> </ul>	<ul style="list-style-type: none"> <li>• 연계 메커니즘과 아키텍처가 복잡하다.</li> <li>• 중간 매개체로 인해 성능이 저하될 수 있다.</li> <li>• 개발 및 적용 기간이 비교적 길다.</li> </ul>

#### • 종류

연계 솔루션	EAI 서버와 송·수신 시스템에 설치되는 클라이언트(Client)를 이용하는 방식 ※ EAI : 송·수신 데이터를 식별하기 위해 송·수신 처리 및 진행 현황을 모니터링하고 통제하는 시스템
ESB (Enterprise Service Bus)	애플리케이션 간 연계, 데이터 변환, 웹 서비스 지원 등 표준 기반의 인터페이스를 제공하는 방식
Socket	서버는 통신을 위한 소켓(Socket)을 생성하여 포트를 할당하고 클라이언트의 통신 요청 시 클라이언트와 연결하여 통신하는 네트워크 기술
Web Service	웹 서비스(Web Service)에서 WSDL과 UDDI, SOAP 프로토콜을 이용하여 연계하는 방식

1. 다음은 송·수신 시스템 사이에 중간 매개체를 두어 연계하는 간접 연계 방식의 종류에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 연계 방식을 쓰시오.

Socket	서버는 통신을 위한 소켓(Socket)을 생성하여 포트를 할당하고 클라이언트의 통신 요청 시 클라이언트와 연결하여 통신하는 네트워크 기술
( ① )	EAI 서버와 송·수신 시스템에 설치되는 클라이언트(Client)를 이용하는 방식

Web Service	웹 서비스(Web Service)에서 WSDL과 UDDI, SOAP 프로토콜을 이용하여 연계하는 방식
( ② )	애플리케이션 간 연계, 데이터 변환, 웹 서비스 지원 등 표준 기반의 인터페이스를 제공하는 방식

#### 답

- ① :  
② :

정답 1. ① 연계 솔루션 ② ESB(Enterprise Service Bus)

### [핵심 062] 연계 메커니즘의 구성

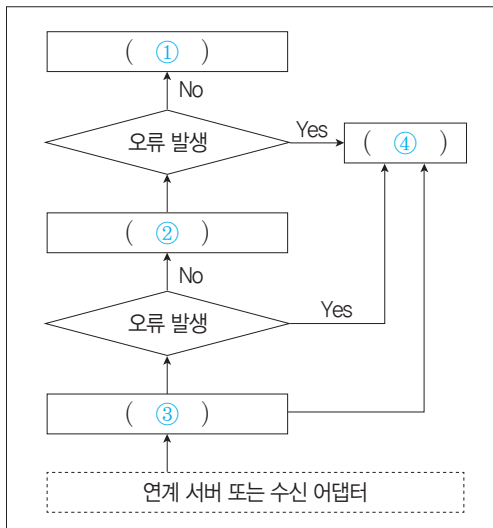


연계 데이터 생성 및 추출	연계 솔루션과 관계없이 응용 프로그램이나 DB 등 응용 시스템에서 연계 데이터를 생성하고 추출하는 것
코드 매핑 및 데이터 변환	송신 시스템에서 사용하는 코드를 수신 시스템에서 사용하는 코드로 매핑 및 변환하는 것
인터페이스 테이블 또는 파일 생성	연계 데이터를 인터페이스 테이블이나 파일 형식으로 생성하는 것

로그(Log) 기록	송·수신 시스템에서 수행되는 모든 과정에 대한 결과 및 오류에 대한 정보를 로그 테이블이나 파일에 기록하는 것
연계 서버 또는 송·수신 어댑터	<ul style="list-style-type: none"> <li>• 연계 서버 : 송·수신 시스템 중 한 곳에 설치하며 인터페이스 테이블 또는 파일의 데이터를 전송 형식에 맞게 변환하고 송·수신을 수행하는 등 송·수신과 관련된 모든 처리 수행</li> <li>• 송신 어댑터 : 송신 시스템에 설치하며, 인터페이스 테이블 또는 파일의 데이터를 전송 형식에 맞도록 변환하고 송신 수행</li> <li>• 수신 어댑터 : 수신 시스템에 설치하며, 송신 시스템으로부터 수신한 데이터를 인터페이스 테이블이나 파일로 생성</li> </ul>
전송	송신 시스템에서 생성된 연계 데이터를 네트워크 환경에 맞는 데이터로 변환한 후 수신 시스템으로 보내는 것
운영 DB에 연계 데이터 반영	수신된 인터페이스 테이블 또는 파일 구조의 데이터를 변환 프로그램을 이용하여 수신 시스템의 운영 DB에 반영하는 것

1. 다음 그림은 연계 메커니즘의 수신 시스템의 주요 기능 및 역할을 표현한 것이다. 괄호(①~④)에 들어갈 알맞은 내용을 쓰시오.

〈수신 시스템 및 수신 모듈〉



답

- ① :  
② :  
③ :  
④ :

**정답 1.** ① 운영 DB에 연계 데이터 반영 ② 코딩 매핑 및 데이터 변환  
③ 인터페이스 테이블 또는 파일 생성 ④ 로그 기록



**[핵심 063] 연계 메커니즘 구간별 장애 및 오류 모니터링 현황**

구간	오류 발생 시점	오류 로그 기록 장소	오류 로그 기록 주체
송신 시스템	<ul style="list-style-type: none"> <li>• 데이터 생성 및 추출 시</li> <li>• 코드 매핑 및 데이터 변환 시</li> <li>• 인터페이스 테이블 또는 파일 등록 시</li> </ul>	송신 연계 프로그램에서 설정한 로그 및 오류 로그 테이블	별도로 구현한 송신용 연계 프로그램
수신 시스템	<ul style="list-style-type: none"> <li>• 연계 데이터 로드(Load) 시</li> <li>• 코드 매핑 및 데이터 변환 시</li> <li>• 운영 DB에 반영 시</li> </ul>	수신 연계 프로그램에서 설정한 로그 및 오류 로그 테이블	별도로 구현한 수신용 연계 프로그램
연계 서버	<ul style="list-style-type: none"> <li>• 연계 데이터 로드(Load) 및 전송 형식으로 변환 시</li> <li>• 연계 데이터 송·수신 시</li> <li>• 수신 시스템의 데이터 형식으로 변환 및 로드(Load) 시</li> </ul>	연계 서버에 설정된 로그 및 오류 로그 테이블 또는 파일	연계 서버

1. 연계 메커니즘에서는 오류 발생 시점을 크게 세 구간으로 구분한다. 괄호(①~③)에 들어갈 알맞은 구간을 쓰시오.

구간	오류 발생 시점
( ① )	<ul style="list-style-type: none"> <li>• 연계 데이터 로드(Load) 시</li> <li>• 코드 매핑 및 데이터 변환 시</li> <li>• 운영 DB에 반영 시</li> </ul>
( ② )	<ul style="list-style-type: none"> <li>• 코드 매핑 및 데이터 변환 시</li> <li>• 인터페이스 테이블 또는 파일 등록 시</li> </ul>
( ③ )	<ul style="list-style-type: none"> <li>• 연계 데이터 송 · 수신 시</li> <li>• 수신 시스템의 데이터 형식으로 변환 및 로드(Load) 시</li> </ul>

답

- ① :  
② :  
③ :

정답 1. ① 수신 시스템 ② 송신 시스템 ③ 연계 서버

#### [핵심 064] 장애 및 오류 유형과 처리 방안

오류 유형	내용	처리 방안
송신 시스템의 연계 프로그램	데이터 생성 및 추출 권한이 없거나 데이터 변환 시 예외(Exception) 미처리 등 연계 프로그램 구현상의 오류	송 · 수신 연계 프로그램의 로그를 확인하여 원인을 분석한 후 처리하고, 데이터 전송이나 반영을 재작업함
수신 시스템의 연계 프로그램	운영 DB 접근 권한이 없거나 데이터 변환 및 반영 시 예외(Exception) 미처리 등 연계 프로그램 구현상의 오류	
연계 서버	연계 서버 실행, 송 · 수신, 전송 형식 변환 등 서버의 기능과 관련된 오류	<ul style="list-style-type: none"> <li>• 연계 서버를 재시작함</li> <li>• 연계 서버의 로그를 확인하여 원인을 분석한 후 처리함</li> </ul>

연계 데이터	송신 시스템에서 생성 및 추출된 데이터의 값이 유효하지 않은 오류	송신 연계 프로그램의 로그를 확인하여 데이터를 보정한 후 재전송함
--------	--------------------------------------	--------------------------------------

1. 장애 및 오류 유형 중 송신 시스템에서 생성 및 추출된 데이터의 값이 유효하지 않은 오류가 발생할 시 처리 방안을 간략히 서술하시오.

답 :

정답 1. 송신 연계 프로그램의 로그를 확인하여 데이터를 보정한 후 재전송한다.

#### [핵심 065] 장애 및 오류의 확인과 처리 절차 / 정의와 설계

##### 장애 및 오류의 확인과 처리 절차

- 장애 및 오류는 1차적으로 연계 서버에서 제공하는 장애 및 오류 현황 모니터링 화면을 통해 오류 원인 및 발생 현황을 확인한다.
- 1차에서 확인이 불가능한 경우 송 · 수신 시스템의 연계 프로그램과 연계 서버에서 기록한 오류 로그 테이블 또는 파일을 확인하여 오류 원인을 분석한다.
- 발생한 오류에 대한 원인이 확인되면 원인에 따른 적절한 조치를 취한다.

##### 장애 및 오류의 정의와 설계

- 장애 및 오류 관리 대상 : 송 · 수신 시스템의 연계 프로그램에서 관리하는 장애 및 오류를 관리 대상으로 정의
- 관리 대상의 장애 및 오류 코드와 메시지 : 관리 대상에서 식별한 오류 내용을 주제별로 분류한 후 각 오류 내용에 오류 코드를 부여하고 오류 메시지를 정의
- 장애 및 오류 코드와 메시지 관리 방식 : 오류 코드와 오류 메시지를 관리하는 방식에는 테이블 관리 방식과 파일 관리 방식이 있음
  - 테이블 관리 방식은 관리 대상 오류 코드와 오류 메시지가 많은 경우, 파일 관리 방식은 관리 대상 오류 코드와 오류 메시지가 적은 경우 사용한다.
- 장애 및 오류 기록 방식 : 오류 로그 테이블 또는 파일은 기록 단위에 따라 인터페이스 테이블 또는 파일에 대한 로그, 연계 데이터에 대한 로그로 설계함



1. 다음은 장애 및 오류 현황의 기록과 확인에 대한 내용이다. 다음 괄호(①, ②)에 들어갈 가장 적합한 용어를 쓰시오.

오류 코드와 오류 메시지를 관리하는 방식에는 ( ① ) 방식과 ( ② ) 방식이 있다. ( ① ) 방식은 관리 대상 오류 코드와 오류 메시지가 많은 경우, ( ② ) 방식은 관리 대상 오류 코드와 오류 메시지가 적은 경우 사용한다.

답

- ① :  
② :

**정답** 1. ① 테이블 관리 ② 파일 관리



### [핵심 066] 연계 데이터 보안

- 송신 시스템에서 수신 시스템으로 전송되는 연계 데이터는 보안에 취약할 수 있으므로 데이터의 중요성을 고려하여 보안을 적용해야 한다.
- 일반적으로 연계 데이터의 보안은 전송 구간에서의 암호·복호화와 데이터의 암호·복호화로 구현한다.

전송 구간 보안	<ul style="list-style-type: none"> <li>전송되는 데이터나 패킷(Packet)을 쉽게 가로챌 수 없도록 암호화 기능이 포함된 프로토콜을 사용하거나 데이터나 패킷을 가로채더라도 내용을 확인할 수 없게 데이터나 패킷을 암호화 한다.</li> <li>전송 구간 암호화를 지원하는 VPN(가상 사설망)이나 연계 솔루션을 적용해 전송 구간 암호화를 수행한다.</li> </ul>
데이터 보안	<ul style="list-style-type: none"> <li>송신 시스템에서 연계 데이터를 추출할 때와 수신 시스템에서 데이터를 운영 DB에 반영할 때 데이터를 암호·복호화 하는 것이다.</li> <li>데이터 보안을 위해서는 암호·복호화 적용 대상 선정, 암호화 알고리즘, 암호·복호화 적용을 위한 환경 설정을 설계 및 구현해야 한다. <ul style="list-style-type: none"> <li>암호·복호화 적용 대상 선정 : 연계 데이터의 중요도에 따라 선정</li> </ul> </li> </ul>

데이터  
보안

- ※ 중요도 : 송·수신 시스템에 정의된 기준에 따라 다르지만 일반적인 적용 대상은 주민등록번호, 운전면허번호, 장애인관리번호, 은행계좌번호, 신용카드번호 등 '개인 정보 보호법'에 근거한 개인 정보
- 암호화 알고리즘 : 주민등록번호, 은행계좌번호와 같은 중요 정보를 보호하기 위해 평문을 암호화된 문장으로 만드는 절차 또는 방법

1. 다음은 연계 데이터 보안에 대한 설명이다. 괄호(①, ②)에 들어갈 가장 적합한 용어를 쓰시오.

전송 구간의 보안은 전송 구간 암호화를 지원하는 ( ① )이나 ( ② )를 적용해 전송 구간 암호화를 수행한다. ( ① )은 인터넷 등 통신 사업자의 공중 네트워크와 암호화 기술을 이용하여 사용자가 마치 자신의 전용 회선을 사용하는 것처럼 해주는 보안 솔루션이고, ( ② )은 EAI 서버와 송·수신 시스템에 설치되는 클라이언트(Client)를 이용하는 간접 연결 방식이다.

답

- ① :  
② :

**정답** 1. ① VPN(Virtual Private Network, 가상 사설망) ② 연계 솔루션

### [핵심 067] 연계 모듈의 구현 환경

- 연계 메커니즘에 따라 구현된 연계 시스템 환경 및 모듈 구현 환경을 의미한다.
- 연계 모듈의 구현 환경은 트리거(Trigger) 또는 프로그래밍 언어를 이용한 EAI, ESB 방식과, 배치 프로그램 및 인터페이스 파일을 이용한 웹 서비스(Web Service) 방식으로 구분된다.

EAI  
(Enterprise  
Application  
Integration)

- 기업 내 각종 애플리케이션 및 플랫폼 간의 정보 전달, 연계, 통합 등 상호 연동이 가능하게 해주는 솔루션이다.
- EAI의 구축 유형
  - Point-to-Point : 가장 기본적인 애플리케이션 통합 방식으로, 애플리케이션을 1:1로 연결
  - Hub & Spoke : 단일 접점인 허브 시스템을 통해 데이터를 전송하는 중앙 집중형 방식



EAI (Enterprise Application Integration)	<ul style="list-style-type: none"> <li>- Message Bus(ESB 방식) : 애플리케이션 사이 에 미들웨어를 두어 처리하는 방식</li> <li>- Bus Hybrid : Hub &amp; Spoke와 Message Bus 의 혼합 방식</li> </ul>
ESB (Enterprise Service Bus)	<ul style="list-style-type: none"> <li>• 애플리케이션 간 연계, 데이터 변환, 웹 서비스 지원 등 표준 기반의 인터페이스를 제공하는 솔 루션이다.</li> <li>• ESB는 애플리케이션 통합 측면에서 EAI와 유사 하지만 애플리케이션 보다는 서비스 중심의 통 합을 지향한다.</li> </ul>
웹 서비스 (Web Service)	<ul style="list-style-type: none"> <li>• 네트워크의 정보를 표준화된 서비스 형태로 만 들어 공유하는 기술로, 서비스 지향 아키텍처 (SOA) 개념을 실현하는 대표적인 기술이다.</li> <li>• 웹 서비스의 구성 <ul style="list-style-type: none"> <li>- SOAP(Simple Object Access Protocol) : HTTP, HTTPS, SMTP 등을 활용하여 XML 기 반의 메시지를 네트워크 상에서 교환하는 프 로토콜</li> <li>- UDDI(Universal Description, Discovery and Integration) : WSDL을 등록하여 서비스와 서 비스 제공자를 검색하고 접근하는데 사용됨</li> <li>- WSDL(Web Services Description Language) : 웹 서비스명, 서비스 제공 위치, 프로토콜 등 웹 서비스에 대한 상세 정보를 XML 형식으로 구현</li> </ul> </li> </ul>

1. 연계 모듈 구현 환경은 이용하는 언어나 인터페이스 등에 따라 다음과 같은 방식으로 구분된다. 괄호(①, ②)에 들어갈 가장 적합한 방식을 쓰시오.

( ① )	네트워크의 정보를 표준화된 서비스 형태로 만들어 공유하는 기술로, 서비스 지향 아키텍처(SOA) 개념을 실현하는 대표적인 기술
( ② )	애플리케이션 간 연계, 데이터 변환, 웹 서비스 지원 등 표준 기반의 인터페이스를 제공하는 솔루션
( ③ )	기업 내 각종 애플리케이션 및 플랫폼 간의 정보 전달, 연계, 통합 등 상호 연동이 가능하게 해주는 솔루션

답 :

- ① :  
② :  
③ :

2. 모듈의 연계 환경 구축 방식인 웹 서비스의 구성 요소 중 HTTP, HTTPS, SMTP 등을 활용하여 XML 기반의 메시지를 네트워크 상에서 교환하는 프로토콜을 무엇이라고 하는지 쓰시오.

답 :

정답 1. ① 웹 서비스(Web Service) ② ESB(Enterprise Service Bus)  
③ EAI(Enterprise Application Integration)

2. SOAP(Simple Object Access Protocol)



#### [핵심 068] XML(eXtensible Markup Language)

- 웹브라우저 간 HTML 문법이 호환되지 않는 문제와 SGML의 복잡함을 해결하기 위하여 개발된 다목적 마크업 언어이다.
- 유니코드를 기반으로 다국어를 지원한다.
- 사용자가 직접 문서의 태그를 정의할 수 있으며, 다른 사용자가 정의한 태그를 사용할 수 있다.
- XML의 구성
  - XML의 첫 문단

```
<?xml version="버전" encoding="언어셋" standalone="yes/no"?>
```

- XML 요소(Element)의 구성

```
<요소이름 속성1="속성값1" 속성2="속성값2"... > 내용 </요소이름>
```

2020 사나공 정보처리기사 필기 수록 내용  
잠깐만요 ! JSON(JavaScript Object Notation)

- 속성-값 쌍(Attribute-Value Pairs)으로 이루어진 데이터 객체를 전달하기 위해 사람이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷입니다.
- 비동기 처리에 사용되는 AJAX에서 XML을 대체하여 사용되고 있습니다.

2020년 1회 기사 실기

1. SW 인터페이스 구현에 관련된 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오.

웹 페이지의 기본 형식인 HTML의 문법이 각 웹브라우저에서 상호 호환적이지 못하다는 문제와 SGML의 복잡함을 해결하기 위하여 개발된 ( )은 다른 특수한 목적을 갖는 마크업 언어이다. 원활한 데이터의 연계를 위해 송수신 시스템 간에 전송되는 데이터가 동일한 구조로 구성될 수 있도록 형태를 정의하는 역할을 수행하며, 다음과 같은 특징이 있다.

유니코드 문자 (Unicode Text)	텍스트 데이터 형식으로 유니코드를 사용하여 전 세계 언어를 지원한다.
( ) 파서 (Parser)	대다수의 웹 브라우저가 해석을 위한 번역기(Parser)를 내장하고 있다.
마크업 (Markup)과 내용(Content)	<ul style="list-style-type: none"> <li>( ) 문서의 문자들은 마크업과 내용으로 구분된다.</li> <li>일반적으로 마크업은 "&lt;"로 시작하여 "&gt;"로 끝나는 태그(Tag)를 의미하고, 그 외의 문자열은 내용에 해당한다.</li> </ul>
엘리먼트 (Element)	마크업과 내용으로 이루어지는 하나의 요소를 의미한다.

답 :

2020년 1회 기사 실기

2. 다음은 소프트웨어 인터페이스 구현에 관련된 설명이다. 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오.

( )은 속성-값 쌍(Attribute-Value Pairs)으로 이루어진 데이터 객체를 전달하기 위해 사람이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷이다. 비동기 처리에 사용되는 AJAX에서 XML을 대체하여 사용되고 있다. 다음은 ( )이 가질 수 있는 자료형이다.

숫자	정수와 실수를 표현
문자열	유니코드 문자로 표현하며, 큰따옴표(" ")로 묶임
참/거짓	참(True) 또는 거짓(False)을 표현
배열	다양한 요소들을 심표로 구분하여 표현하며, 대괄호([ ])로 묶임
객체	이름/값 쌍으로 표현하며, 중괄호({ })로 묶임
NULL	아직 알려지지 않거나 모르는 값을 표현하기 위한 자료형

답 :

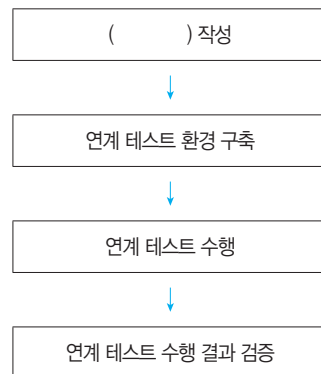
정답 1. XML(Extensible Markup Language)

2. JSON(JavaScript Object Notation)

## [핵심 069] 연계 테스트

- 구축된 연계 시스템과 연계 시스템의 구성 요소가 정상적으로 동작하는지 확인하는 활동이다.
- 연계 테스트 절차
  - 연계 테스트 케이스 작성 : 연계 시스템 간의 데이터 및 프로세스의 흐름을 분석하여 필요한 테스트 항목을 도출하는 과정
  - 연계 테스트 환경 구축 : 테스트의 일정, 방법, 절차, 소요 시간 등을 송·수신 기관과의 협의를 통해 결정
  - 연계 테스트 수행 : 연계 응용 프로그램을 실행하여 연계 테스트 케이스의 시험 항목 및 처리 절차 등을 실제로 진행
  - 연계 테스트 수행 결과 검증 : 연계 테스트 케이스의 시험 항목 및 처리 절차를 수행한 결과가 예상 결과와 동일한지를 확인

1. 다음은 구축된 연계 시스템과 연계 시스템의 구성 요소가 정상적으로 동작하는지 확인하는 연계 테스트의 수행 과정이다. 괄호에 들어갈 가장 적합한 용어를 쓰시오.



답 :

정답 1. 연계 테스트 케이스

## 5장 | 서버 프로그램 구현

### [핵심 070] 개발 환경 구축 - 하드웨어 환경

- 개발 환경 구축은 응용 소프트웨어 개발을 위해 개발 프로젝트를 이해하고 소프트웨어 및 하드웨어 장비를 구축하는 것을 의미한다.
- 하드웨어 환경은 사용자와의 인터페이스 역할을 하는 클라이언트(Client) 그리고 클라이언트와 통신하여 서비스를 제공하는 서버(Server)로 구성된다.
- 서버는 사용 목적에 따라 웹 서버, 웹 애플리케이션 서버, 데이터베이스 서버, 파일 서버 등으로 나뉜다.

웹 서버 (Web Server)	클라이언트로부터 직접 요청을 받아 처리하는 서버로, 저장량의 정적 파일들을 제공
웹 애플리케이션 서버(WAS; Web Application Server)	사용자에게 동적 서비스를 제공하기 위해 웹 서버로부터 요청을 받아 데이터 가공 작업을 수행하거나, 웹 서버와 데이터베이스 서버 또는 웹 서버와 파일 서버 사이에서 인터페이스 역할을 수행하는 서버
데이터베이스 서버(DB Server)	데이터베이스와 이를 관리하는 DBMS를 운영하는 서버
파일 서버 (File Server)	데이터베이스에 저장하기에는 비효율적이거나, 서비스 제공을 목적으로 유지하는 파일들을 저장하는 서버

1. 다음은 클라이언트에게 서비스를 제공하는 서버에 대한 설명이다. 괄호(①, ②)에 들어갈 가장 적합한 서버를 쓰시오.

데이터베이스 서버	데이터베이스와 이를 관리하는 DBMS를 운영하는 서버
( ① )	클라이언트로부터 직접 요청을 받아 처리하는 서버로, 저장량의 정적 파일들을 제공
파일 서버	데이터베이스에 저장하기에는 비효율적이거나, 서비스 제공을 목적으로 유지하는 파일들을 저장하는 서버
( ② )	사용자에게 동적 서비스를 제공하기 위해 웹 서버로부터 요청을 받아 데이터 가공 작업을 수행하거나, 웹 서버와 데이터베이스 서버 또는 웹 서버와 파일 서버 사이에서 인터페이스 역할을 수행하는 서버

답

- ① :  
② :

정답 1. ① 웹 서버(Web Server)

② 웹 애플리케이션 서버(WAS; Web Application Server)

### [핵심 071] 개발 환경 구축 - 소프트웨어 환경

- 소프트웨어 환경은 클라이언트와 서버 운영을 위한 시스템 소프트웨어와 개발에 사용되는 개발 소프트웨어로 구성된다.
- 시스템 소프트웨어에는 운영체제(OS), 웹 서버 및 WAS 운영을 위한 서버 프로그램, DBMS 등이 있다.
- 개발 소프트웨어에는 요구사항 관리 도구, 설계/모델링 도구, 구현 도구, 빌드 도구, 테스트 도구, 형상 관리 도구 등이 있다.

요구사항 관리 도구	요구사항의 수집과 분석, 추적 등을 편리하게 도와주는 소프트웨어
설계/모델링 도구	UML(통합 모델링 언어)을 지원하며, 개발의 전 과정에서 설계 및 모델링을 도와주는 소프트웨어
구현 도구	개발 언어를 통해 애플리케이션의 실제 구현을 지원하는 소프트웨어
빌드 도구	구현 도구를 통해 작성된 소스의 빌드 및 배포, 라이브러리 관리를 지원하는 소프트웨어
테스트 도구	모듈들이 요구사항에 적합하게 구현되었는지 테스트하는 소프트웨어
형상 관리 도구	산출물들을 버전별로 관리하여 품질 향상을 지원하는 소프트웨어

1. 소프트웨어 개발에 사용되는 개발 소프트웨어 중 다음은 어떤 도구에 대한 설명인지 쓰시오.

- 산출물들을 버전별로 관리하여 품질 향상을 지원하는 소프트웨어이다.
- 종류에는 GIT, CVS, Subversion, Mercurial 등이 있다.

답 :

정답 1. 형상 관리 도구

**[핵심 072] 모듈(Module)**

- 모듈화를 통해 분리된 시스템의 각 기능들로, 서브루틴, 서브시스템, 소프트웨어 내의 프로그램, 작업 단위 등과 같은 의미로 사용된다.
  - 모듈화 : 소프트웨어의 성능을 향상시키거나 시스템의 수정 및 재사용, 유지 관리 등이 용이하도록 시스템의 기능들을 모듈 단위로 분해하는 것으로 모듈 간 결합도의 최소화, 응집도의 최대화가 목표임
- 모듈의 기능적 독립성은 소프트웨어를 구성하는 각 모듈의 기능이 서로 독립됨을 의미하는 것으로, 모듈이 하나의 기능만을 수행하고 다른 모듈과의 과도한 상호작용을 배제함으로써 이루어진다.
- 독립성이 높은 모듈일수록 모듈을 수정하더라도 다른 모듈들에게는 거의 영향을 미치지 않으며, 오류가 발생해도 쉽게 발견하고 해결할 수 있다.
- 모듈의 독립성은 결합도(Coupling)와 응집도(Cohesion)에 의해 측정되며, 독립성을 높이려면 모듈의 결합도는 약하게, 응집도는 강하게, 모듈의 크기는 작게 만들어야 한다.

2020년 1회 기사 실기

1. 소프트웨어 패키징이란 모듈별로 생성한 실행 파일들을 묶어 배포용 설치 파일을 만드는 것을 말한다. 괄호(①, ②)에 들어갈 가장 적합한 용어를 쓰시오.

소스 코드는 향후 관리를 고려하여 모듈화하여 패키징한다. 모듈화는 모듈 간 ( ① )의 최소화와 모듈 내 요소들의 ( ② )는 최대화 하는 것이 목표이다.

**답**

- ① :  
② :

**정답** 1. ① 결합도(Coupling) ② 응집도(Cohesion)**[핵심 073] 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 종류에는 자료 결합도, 스탬프 결합도, 제어 결합도, 외부 결합도, 공통 결합도, 내용 결합도가 있으며 결합도의 정도는 다음과 같다.

자료 결합도	스탬프 결합도	제어 결합도	외부 결합도	공통 결합도	내용 결합도
결합도 약함 ← → 결합도 강함					
자료 결합도 (Data Coupling)	<ul style="list-style-type: none"> <li>• 모듈 간의 인터페이스가 자료 요소로만 구성될 때의 결합도</li> <li>• 어떤 모듈이 다른 모듈을 호출하면서 매개 변수나 인수로 데이터를 넘겨주고, 호출 받은 모듈은 받은 데이터에 대한 처리 결과를 다시 돌려주는 방식</li> </ul>				
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 요소(Function Code, Switch, Tag, Flag)를 전달하는 결합도				
외부 결합도 (External Coupling)	어떤 모듈에서 선언한 데이터(변수)를 외부의 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				

2020년 1, 2회 기사 필기

1. 다음의 설명에서 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

- ( )는 모듈 간에 상호 의존하는 정도 또는 두 모듈 사이의 연관 관계를 의미한다.
- ( )가 높으면 시스템 구현 및 유지보수 작업이 어렵다.
- 모듈 간의 ( )를 약하게 하면 모듈 독립성이 향상된다.

**답 :**

2. 다음은 두 모듈 사이의 연관 관계를 나타내는 결합도의 종류에 대한 설명이다. ①~③의 설명에 해당하는 결합도를 쓰시오.

- ① ( ) 결합도 : 모듈 간의 인터페이스로 배열이나 레코드 등의 자료 구조가 전달될 때의 결합도  
 ② ( ) 결합도 : 한 모듈이 다른 모듈의 내부 기능 및 그 내부 자료를 직접 참조하거나 수정할 때의 결합도  
 ③ ( ) 결합도 : 모듈 간의 인터페이스가 자료 요소로만 구성될 때의 결합도

답

- ① :  
 ② :  
 ③ :

정답 1. 결합도(Coupling)

2. ① 스탬프(Stapm) ② 내용(Content) ③ 자료(Data)

### [핵심 074] 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 종류에는 기능적 응집도, 순차적 응집도, 교환(통신)적 응집도, 절차적 응집도, 시간적 응집도, 논리적 응집도, 우연적 응집도가 있으며 응집도의 정도는 다음과 같다.

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

응집도 강함 ← → 응집도 약함

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 모여 있을 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

2020년 1, 2회 기사 필기

1. 다음 지문에 제시된 응집도를 높은 응집도에서 낮은 응집도 순으로 나열하시오. 단, 기호로 쓰시오.

- ㉠ 기능적 응집도    ㉡ 시간적 응집도    ㉢ 절차적 응집도  
 ㉣ 우연적 응집도

답 : ( ) → ( ) → ( ) → ( )

2. 다음은 응집도의 종류에 대한 설명이다. 괄호(①~③)에 들어갈 가장 적합한 응집도를 쓰시오.

( ① ) 응집도	모듈 내 하나의 활동으로부터 나온 출력 데이터를 그 다음 활동의 입력 데이터로 사용할 경우의 응집도
기능적 응집도	모듈 내부의 모든 기능 요소들이 단일 문제와 연관되어 수행될 경우의 응집도
( ② ) 응집도	특정 시간에 처리되는 몇 개의 기능을 모아 하나의 모듈로 작성할 경우의 응집도
논리적 응집도	유사한 성격을 갖거나 특정 형태로 분류되는 처리 요소들로 하나의 모듈이 형성되는 경우의 응집도
( ③ ) 응집도	모듈이 다수의 관련 기능을 가질 때 모듈 안의 구성 요소들이 그 기능을 순차적으로 수행할 경우의 응집도

답

- ① :  
 ② :  
 ③ :

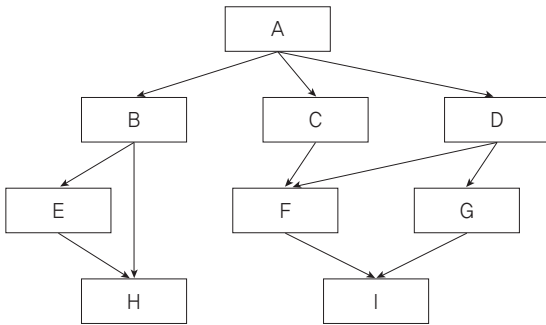
정답 1. ㉠ → ㉢ → ㉡ → ㉣

2. ① 순차적(Sequential) ② 시간적(Temporal) ③ 절차적(Procedural)

### [핵심 075] 팬인(Fan-In) / 팬아웃(Fan-Out)

- 팬인은 어떤 모듈을 제어(호출)하는 모듈의 수를 나타낸다.
- 팬아웃은 어떤 모듈에 의해 제어(호출)되는 모듈의 수를 나타낸다.
- 팬인과 팬아웃을 분석하여 시스템의 복잡도를 알 수 있다.
- 팬인이 높다는 것은 재사용 측면에서 설계가 잘 되어있다고 볼 수 있으나, 단일 장애점이 발생할 수 있으므로 중점적인 관리 및 테스트가 필요하다.

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



**해설**

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

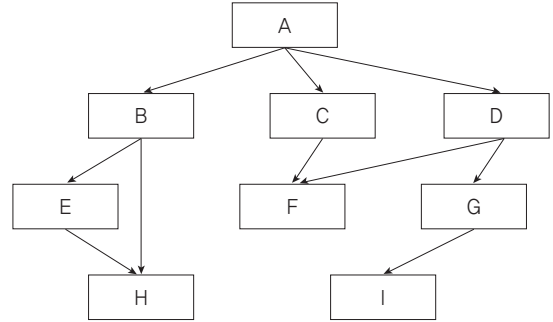
1. ( ㉠ )은 어떤 모듈을 제어하는 모듈의 수를 나타내고, ( ㉡ )은 어떤 모듈에 의해 제어되는 모듈의 수를 나타낸다. 괄호 (㉠, ㉡)에 들어갈 가장 적합한 용어를 쓰시오.

**답**

- ① :  
② :

2020년 1회 기사 실기

2. 다음의 모듈 관계를 표현한 시스템 구조도를 참고하여 팬인(Fan-In)이 2 이상인 모듈을 모두 쓰시오.



**답 :**

**해설** 2. 팬인(Fan-In)은 위에서 특정 모듈로 들어오는(In) 선(Line)의 수라고 생각하면 됩니다. 그러므로 2개의 선이 들어오는 F, H 모듈이 정답이 됩니다.

정답 1. ① 팬인(Fan-In) ② 팬아웃(Fan-Out) 2. F, H

### [핵심 076] 공통 모듈

- 여러 프로그램에서 공통적으로 사용할 수 있는 모듈을 의미한다.
- 자주 사용되는 계산식이나 매번 필요한 사용자 인증과 같은 기능들이 공통 모듈로 구성될 수 있다.
- 공통 모듈을 구현할 때는 다른 개발자들이 해당 기능을 명확히 이해할 수 있도록 다음의 명세 기법을 준수해야 한다.

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

1. 다음 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

- ( )는 자주 사용되는 계산식이나 매번 필요한 사용자 인증과 같은 기능들에 주로 사용된다.
- 인터넷 쇼핑몰 사이트에서 사이트의 첫 페이지에서 '로그인'을 할 수도 있고, 상품 구매 버튼을 누른 후에 나오는 '로그인' 창에서 할 수도 있다. 두 개의 '로그인'은 발생하는 위치는 다르지만 동일한 기능을 갖고 있어 ( )로 구성하기에 적합하다.

답 :

2020년 1.2회 기사 필기

2. 공통 모듈에 대한 명세 기법 중 해당 기능에 대해 일관되게 이해되고 한 가지로 해석될 수 있도록 작성하는 원칙을 쓰시오.

답 :

**정답** 1. 공통 모듈 2. 명확성(Clarity)

### [핵심 077] 재사용(Reuse)

- 비용과 개발 시간을 절약하기 위해 이미 개발된 기능들을 파악하고 재구성하여 새로운 시스템 또는 기능 개발에 사용하기 적합하도록 최적화 시키는 작업이다.
- 재사용되는 대상은 외부 모듈과의 결합도는 낮고, 응집도는 높아야 한다.
- 재사용 규모에 따른 분류

함수와 객체	클래스나 메소드 단위의 소스 코드를 재사용함
컴포넌트	컴포넌트 자체에 대한 수정 없이 인터페이스를 통해 통신하는 방식으로 재사용함
애플리케이션	공통된 기능들을 제공하는 애플리케이션을 공유하는 방식으로 재사용함

1. 다음은 재사용에 대한 설명이다. 괄호(①~③)에 들어갈 가장 적합한 용어를 쓰시오.

재사용은 비용과 개발 시간을 절약하기 위해 이미 개발된 기능들을 파악하고 재구성하여 새로운 시스템 또는 기능 개발에 사용하기 적합하도록 최적화 시키는 작업으로 재사용되는 대상은 외부 모듈과의 ( ① )는 낮고, ( ② )는 높아야 한다. ( ① )는 모듈 간에 상호 의존하는 정도를, ( ② )는 내부요소들의 서로 관련되어 있는 정도를 나타낸다.

답

- ① :  
② :

**정답** 1. ① 결합도 ② 응집도

### [핵심 078] DBMS 접속(Connection)

- 사용자가 데이터를 사용하기 위해 응용 시스템을 이용하여 DBMS에 접근하는 것을 의미한다.
- 응용 시스템은 사용자로부터 매개 변수를 전달받아 SQL을 실행하고 DBMS로부터 전달받은 결과를 사용자에게 전달하는 매개체 역할을 수행한다.
- 인터넷을 통해 구동되는 웹 응용 프로그램은 웹 응용 시스템을 통해 DBMS에 접근한다.
- DBMS 접속 기술 : DBMS에 접근하기 위해 사용하는 API 또는 API의 사용을 편리하게 도와주는 프레임워크 등을 의미

JDBC(Java DataBase Connectivity)	Java 언어로 다양한 종류의 데이터베이스에 접속하고 SQL문을 수행할 때 사용되는 표준 API
ODBC(Open DataBase Connectivity)	데이터베이스에 접근하기 위한 표준 개방형 API로, 개발 언어에 관계없이 사용할 수 있음
MyBatis	<ul style="list-style-type: none"> <li>• JDBC 코드를 단순화하여 사용할 수 있는 SQL Mapping 기반 오픈 소스 접속 프레임워크</li> <li>• SQL을 거의 그대로 사용할 수 있어 SQL 친화적인 국내 환경에 적합하여 많이 사용됨</li> </ul>

※ 프레임워크(Framework) : 본래 '뼈대', '골조'를 의미하는 용어로, 소프트웨어에서는 특정 기능을 수행하기 위해 필요한 클래스나 인터페이스 등을 모아둔 집합체를 가리킴

- 동적 SQL(Dynamic SQL)
  - 다양한 조건에 따라 SQL 구문을 동적으로 변경하여 처리할 수 있는 SQL 처리 방식이다.
  - 동적 SQL은 사용자로부터 SQL문의 일부 또는 전부를 입력받아 실행할 수 있다.



- 동적 SQL은 정적 SQL에 비해 속도가 느리지만, 상황에 따라 다양한 조건을 첨가하는 등 유연한 개발이 가능하다.

※ 정적 SQL : SQL 코드를 변수에 담지 않고 코드 사이에 직접 기술한 SQL문

**1. 다음의 설명에 가장 부합하는 DBMS 접속 기술을 쓰시오.**

- 데이터베이스에 접근하기 위한 표준 개방형 API로, 개발 언어에 관계없이 사용할 수 있다.
- 프로그램 내 이 기술의 문장을 사용하여 MS-Access, DBase, DB2, Excel, Text 등 다양한 데이터베이스에 접근할 수 있다.

답 :

**2. JDBC 코드를 단순화하여 사용할 수 있는 SQL Mapping 기반 오픈 소스 접속 프레임워크로 SQL을 거의 그대로 사용할 수 있어 SQL 친화적인 국내 환경에 적합하여 많이 사용되는 DBMS 접속 기술을 쓰시오.**

답 :

**정답** 1. ODBC(Open DataBase Connectivity) 2. MyBatis

**[핵심 079] 서버 개발**

- 웹 애플리케이션의 로직을 구현할 서버 프로그램을 제작하여 웹 애플리케이션 서버(WAS)에 탑재하는 것을 의미한다.
- 웹 애플리케이션 서버에 구현된 서버 프로그램은 웹 서버로부터 받은 요청을 처리하여 결과를 반환하는 역할을 수행한다.
- 서버 개발에 사용되는 프로그래밍 언어에는 Java, JavaScript, Python, PHP, Ruby 등이 있다.
- 각 프로그래밍 언어에는 해당 언어로 서버 프로그램을 개발할 수 있도록 지원하는 프레임워크가 있다.

**• 서버 개발 프레임워크**

프레임워크	특징
spring	JAVA를 기반으로 만들어진 프레임워크로, 전자정부 표준 프레임워크의 기반 기술로 사용되고 있다.
Node.js	JavaScript를 기반으로 만들어진 프레임워크로, 비동기 입·출력 처리와 이벤트 위주의 높은 처리 성능을 갖고 있어 실시간으로 입·출력이 빈번한 애플리케이션에 적합하다.
Django	Python을 기반으로 만들어진 프레임워크로, 컴포넌트의 재사용과 플러그인화를 강조하여 신속한 개발이 가능하도록 지원한다.
Codeigniter	PHP를 기반으로 만들어진 프레임워크로, 인터페이스가 간편하며 서버 자원을 적게 사용한다.
Ruby on Rails	Ruby를 기반으로 만들어진 프레임워크로, 테스트를 위한 웹 서버를 지원하며 데이터베이스 작업을 단순화, 자동화시켜 개발 코드의 길이가 짧아 신속한 개발이 가능하다.

**• 프레임워크의 특징**

모듈화 (Modularity)	프레임워크는 캡슐화를 통해 모듈화를 강화하고 설계 및 구현의 변경에 따른 영향을 최소화함으로써 소프트웨어의 품질을 향상시킨다.
재사용성 (Reusability)	프레임워크는 재사용 가능한 모듈들을 제공함으로써 개발자의 생산성을 향상시킨다.
확장성 (Extensibility)	프레임워크는 다형성(Polymorphism)을 통한 인터페이스 확장이 가능하여 다양한 형태와 기능을 가진 애플리케이션 개발이 가능하다.
제어의 역흐름 (Inversion of Control)	개발자가 관리하고 통제해야 하는 객체들의 제어 권한을 프레임워크에 넘김으로써 생산성을 향상시킨다.

**1. 다음은 서버 개발 프레임워크들의 특징을 설명한 것이다. 제시된 특징에 부합되도록 괄호(①, ②)에 들어갈 가장 알맞은 프레임워크를 <보기>에서 골라 쓰시오.**

- ( ① ) : JavaScript를 기반으로 만들어진 프레임워크로, 비동기 입·출력 처리와 이벤트 위주의 높은 처리 성능을 갖고 있어 실시간으로 입·출력이 빈번한 애플리케이션에 적합하다.



- ( ② ) : 테스트를 위한 웹 서버를 지원하며 데이터베이스 작업을 단순화, 자동화시켜 개발 코드의 길이가 짧아 신속한 개발이 가능하다.

〈보기〉

스프링(Spring), 이클립스(Eclipse), 엑스 코드(Xcode), 메이븐(Maven), 장고(Django), 루비 온 레일즈(Ruby on Rails), 마이바티스(MyBatis), 코드이그나이터(Codeigniter), Node.js

답

- ① :  
② :

2. 프레임워크(Framework)의 특성 중 제어의 역흐름(Inversion of Control)에 대해 간략히 서술하시오.

답 :

정답 1. ① Node.js ② 루비 온 레일즈(Ruby on Rails)

2. 제어의 역흐름은 개발자가 관리하고 통제해야 하는 객체들의 제어 권한을 프레임워크에 넘김으로써 생산성을 향상시키는 특성이다.

[핵심 080] 서버 개발 과정

- 서버 개발은 DTO/VO, SQL, DAO, Service, Controller를 각각 구현하는 과정을 통해 이루어진다.
- 구현 순서는 개발자가 임의로 변경할 수 있다.
- 개발하려는 서버 프로그램의 목적, 개발 언어, 규모 등의 이유로 통합되거나 세분화될 수 있다.
- 모든 과정에서 보안 약점이 발생하지 않도록 소프트웨어 개발 보안 가이드를 참고한다.

<p>DTO (Data Transfer Object)/ VO(Value Object) 구현</p>	<ul style="list-style-type: none"> <li>• 데이터 교환을 위해 사용할 객체를 만드는 과정이다.</li> <li>• 변수 및 객체를 송·수신할 데이터의 자료형(Data Type)에 알맞게 생성한다.</li> <li>• 알고리즘 등의 로직은 구현하지 않고, 변수와 데이터를 저장하고 반환하는 메소드만 구현한다.</li> </ul>
--	---

<p>SQL 구현</p>	<ul style="list-style-type: none"> <li>• 데이터의 삽입, 변경, 삭제 등의 작업을 수행할 SQL문을 생성하는 과정이다.</li> <li>• SQL문은 소스 코드 내에 직접 입력하거나, 별도의 XML 파일로 저장하여 관리한다.</li> <li>• XML 파일로 SQL문을 관리하는 경우 중복되는 SQL문을 최소화할 수 있고, 유지보수가 간편해진다.</li> </ul>
<p>DAO (Data Access Object) 구현</p>	<p>데이터베이스에 접근하고, SQL을 활용하여 데이터를 실제로 조작하는 코드를 구현하는 과정이다.</p>
<p>Service 구현</p>	<p>사용자의 요청에 응답하기 위한 로직을 구현하는 과정이다.</p>
<p>Controller 구현</p>	<p>사용자의 요청에 적절한 서비스를 호출하여, 그 결과를 사용자에게 반환하는 코드를 구현하는 과정이다.</p>

1. 서버 개발은 DTO/VO, SQL, DAO, Service, Controller를 각각 구현하는 과정을 통해 이루어지는데, 서버 개발 구현 과정 중 데이터베이스에 접근하고, SQL을 활용하여 데이터를 실제로 조작하는 코드를 구현하는 과정은 무엇인지 쓰시오.

답 :

정답 1. DAO(Data Access Object) 구현

[핵심 081] 배치 프로그램(Batch Program)

- 사용자와의 상호 작용 없이 여러 작업들을 미리 정해진 일련의 순서에 따라 일괄적으로 처리하는 것을 의미한다.
- 배치 프로그램이 자동으로 수행되는 주기에 따라 정기 배치, 이벤트성 배치, On-Demand 배치로 구분된다.
- 배치 프로그램의 필수 요소 : 대용량 데이터, 자동화, 견고성, 안정성/신뢰성, 성능
- 배치 스케줄러(Batch Scheduler) : 일괄 처리(Batch Processing) 작업이 설정된 주기에 맞춰 자동으로 수행되도록 지원해주는 도구

• 주요 배치 스케줄러

스프링 배치 (Spring Batch)	<ul style="list-style-type: none"> <li>• Spring Source 사와 Accenture 사가 2007년 공동 개발한 오픈소스 프레임워크이다.</li> <li>• 로그 관리, 추적, 트랜잭션 관리, 작업 처리 통계, 작업 재시작 등의 다양한 기능을 제공한다.</li> <li>• 주요 구성 요소 : Job, Job Launcher, Step, Job Repository</li> </ul>
Quartz	<ul style="list-style-type: none"> <li>• 스프링 프레임워크로 개발되는 응용 프로그램들의 일괄 처리를 위한 다양한 기능을 제공하는 오픈 소스 라이브러리이다.</li> <li>• 주요 구성 요소 : Scheduler, Job, JobDetail, Trigger</li> </ul>
Cron	리눅스의 스케줄러 도구로 crontab 명령어를 통해 작업을 예약할 수 있다.

1. 다음 설명에 가장 부합하는 용어를 쓰시오.

사용자와의 상호 작용 없이 여러 작업들을 미리 정해진 일련의 순서에 따라 일괄적으로 처리하는 것을 의미하며, 수행하는 주기에 따라 정기, 이벤트성, On-Demand로 구분할 수 있다.

답 :

2. 배치 프로그램과 관련된 소프트웨어 중 다음과 같은 특징을 갖는 소프트웨어를 쓰시오.

- 데이터베이스나 파일의 데이터를 교환하는데 필요한 컴포넌트들을 제공한다.
- 로그 관리, 추적, 트랜잭션 관리, 작업 처리 통계, 작업 재시작 등의 다양한 기능을 제공한다.
- 주요 구성 요소에는 Job, Job Launcher, Step, Job Repository가 있다.

답 :

**정답** 1. 배치 프로그램(Batch Program)

2. 스프링 배치(Spring Batch)

## 6장 | 화면 설계

### [핵심 082] 사용자 인터페이스(UI; User Interface)

- 사용자와 시스템 간의 상호작용이 원활하게 이뤄지도록 도와주는 장치나 소프트웨어를 의미한다.
- 사용자 인터페이스의 구분

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

2020년 2회 기공사 실기

1. 사용자 인터페이스 중 아이콘이나 메뉴를 마우스로 선택하여 작업을 수행하는 그래픽 환경의 인터페이스는 무엇인지 약어로 쓰시오.

답 :

**정답** 1. GUI

### [핵심 083] 사용자 인터페이스의 기본 원칙

직관성	누구나 쉽게 이해하고 사용할 수 있어야 한다.
유효성	사용자의 목적을 정확하고 완벽하게 달성해야 한다.
학습성	누구나 쉽게 배우고 익힐 수 있어야 한다.
유연성	사용자의 요구사항을 최대한 수용하고 실수를 최소화해야 한다.

2020년 1, 2회 기사 필기

1. 다음은 사용자 인터페이스의 기본 원칙들에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 원칙을 쓰시오.

학습성	누구나 쉽게 배우고 익힐 수 있어야 한다.
( ① )	사용자의 목적을 정확하고 완벽하게 달성해야 한다.
유연성	사용자의 요구사항을 최대한 수용하고 실수를 최소화해야 한다.
( ② )	누구나 쉽게 이해하고 사용할 수 있어야 한다.

답

- ① :  
② :

정답 1. ① 유효성 ② 직관성

### [핵심 084] 사용자 인터페이스의 설계 지침

사용자 중심	사용자가 쉽게 이해하고 편리하게 사용할 수 있는 환경을 제공하며, 실사용자에 대한 이해가 바탕이 되어야 한다.
일관성	버튼이나 조작 방법 등을 일관성 있게 제공하므로 사용자가 쉽게 기억하고 습득할 수 있게 설계해야 한다.
단순성	조작 방법을 단순화시켜 인지적 부담을 감소시켜야 한다.
결과 예측 가능	작동시킬 기능만 보고도 결과를 미리 예측할 수 있게 설계해야 한다.
가시성	메인 화면에 주요 기능을 노출시켜 최대한 조작이 쉽도록 설계해야 한다.
표준화	기능 구조와 디자인을 표준화하여 한 번 학습한 이후에는 쉽게 사용할 수 있도록 설계해야 한다.
접근성	사용자의 연령, 성별, 인증 등 다양한 계층이 사용할 수 있도록 설계해야 한다.
명확성	사용자가 개념적으로 쉽게 인지할 수 있도록 설계해야 한다.
오류 발생 해결	오류가 발생하면 사용자가 쉽게 인지할 수 있도록 설계해야 한다.

1. 다음은 사용자 인터페이스의 설계 지침 중 일부를 설명한 것이다. 괄호(①~③)에 들어갈 알맞은 지침을 쓰시오.

( ① )	메인 화면에 주요 기능을 노출시켜 최대한 조작이 쉽도록 설계해야 한다.
( ② )	사용자의 연령, 성별, 인증 등 다양한 계층이 사용할 있도록 설계해야 한다.
( ③ )	사용자가 개념적으로 쉽게 인지할 수 있도록 설계해야 한다.

답

- ① :  
② :  
③ :

정답 1. ① 가시성 ② 접근성 ③ 명확성

### [핵심 085] UI 설계 도구

와이어프레임 (Wireframe)	<ul style="list-style-type: none"> <li>• 기획 단계의 초기에 제작하는 것으로, 페이지에 대한 개략적인 레이아웃이나 UI 요소 등에 대한 뼈대를 설계하는 단계이다.</li> <li>• 와이어프레임을 제작할 때는 각 페이지의 영역 구분, 콘텐츠, 텍스트 배치 등을 화면 단위로 설계한다.</li> </ul>
모업 (Mockup)	<ul style="list-style-type: none"> <li>• 디자인, 사용 방법 설명, 평가 등을 위해 와이어프레임보다 좀 더 실제 화면과 유사하게 만든 정적인 형태의 모형이다.</li> <li>• 시각적으로만 구성 요소를 배치하는 것으로 실제로 구현되지는 않는다.</li> </ul>
스토리보드 (Story Board)	<ul style="list-style-type: none"> <li>• 와이어프레임에 콘텐츠에 대한 설명, 페이지 간 이동 흐름 등을 추가한 문서이다.</li> <li>• 스토리보드의 상단이나 우측에는 제목, 작성자 등을 입력하고, 좌측에는 UI 화면, 우측에는 디스크립션(Description)을 기입한다.</li> </ul>
프로토타입 (Prototype)	<ul style="list-style-type: none"> <li>• 와이어프레임이나 스토리보드 등에 인터랙션을 적용함으로써 실제 구현된 것처럼 테스트가 가능한 동적인 형태의 모형이다.</li> <li>• 프로토타입은 사용성 테스트나 작업자 간 서비스 이해를 위해 작성하는 샘플이다.</li> <li>• 페이퍼 프로토타입 : 아날로그적인 방법으로, 스케치, 그림, 글 등을 이용하여 손으로 직접 작성하는 방법</li> <li>• 디지털 프로토타입 : 파워포인트, 아크로벳, 비지오, 옴니그래플 등과 같은 프로그램을 사용하여 작성하는 방법</li> </ul>
유스케이스 (Use Case)	<ul style="list-style-type: none"> <li>• 사용자 측면에서의 요구사항으로, 사용자가 원하는 목표를 달성하기 위해 수행할 내용을 기술한다.</li> <li>• 사용자의 요구사항을 빠르게 파악함으로써 프로젝트의 초기에 시스템의 기능적인 요구를 결정하고 그 결과를 문서화할 수 있다.</li> </ul>

1. 다음은 사용자의 요구사항에 맞게 UI의 화면 구조나 화면 배치 등을 설계할 때 사용하는 UI 설계 도구에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 도구를 쓰시오.

- ( ① )은 기획 단계의 초기에 제작하는 것으로, 페이지에 대한 개략적인 레이아웃이나 UI 요소 등에 대한 뼈대를 설계하는 단계이다.
- ( ② )은 디자인, 사용 방법 설명, 평가 등을 위해 ( ① )보다 좀 더 실제 화면과 유사하게 만든 정적인 형태의 모형이다.

답 :

- ① :  
② :

2. 사용자 요구사항을 기반으로 실제 동작하는 것처럼 만든 동적인 형태의 모형을 무엇이라고 하는지 쓰시오.

답 :

**정답** 1. ① 와이어프레임(Wireframe) ② 목업(Mockup)  
2. 프로토타입(Prototype)

### [핵심 086] UI 스타일 가이드 작성

UI 스타일 가이드는 개발자나 디자이너들이 UI를 작성할 때 기준이 되는 규칙들로, 구동 환경, 레이아웃, 네비게이션 등을 정의한다.

구동 환경 정의	컴퓨터 OS, 웹 브라우저, 모니터 해상도, 프레임 세트 등을 사용 환경에 적합하도록 규정하는 단계이다.
레이아웃 정의	<ul style="list-style-type: none"> <li>• 화면 구조를 정의하고 각 영역의 메뉴를 구성하는 단계이다.</li> <li>• 레이아웃 영역에는 Top, Left, Contents, Footer Area가 있으며, 기본적으로 Top, Left, Contents 영역으로 구성한다. <ul style="list-style-type: none"> <li>- 상단 메뉴(Top Area) : 필수 영역으로 시스템 전체 페이지에 동일하게 적용됨</li> <li>- 좌측 메뉴(Left Area) : 선택 영역으로 시스템별 서브 페이지에 선택적으로 적용됨</li> </ul> </li> </ul>

레이아웃 정의

- 내용 구성(Contents Area) : 필수 영역으로 시스템의 전체 개념을 나타내는 메인 이미지와 시스템별로 필요한 콘텐츠를 표시함
- 하단 메뉴(Footer Area) : 선택 영역으로 회사 상황에 따라 표시 여부를 결정할 수 있음

네비게이션 정의

- 네비게이션의 메뉴 타입을 선택하여 적용하는 단계이다.
- 네비게이션은 사용자가 원하는 정보를 빠르게 찾을 수 있도록 안내하는 것으로 메뉴, 버튼, 링크 등으로 구성된다.

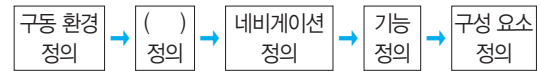
기능 정의

시스템에 적용할 업무 과정에서 일어나는 모든 활동이나 필요한 데이터 간의 관계 등을 논리적인 모델로 상세화하는 단계이다.

구성 요소 정의

화면에 표시할 그리드나 버튼 등을 정의하는 단계이다.

1. 다음은 개발자나 디자이너들이 UI를 작성할 때 기준이 되는 규칙인 UI 스타일 가이드의 작성 순서이다. 괄호에 들어갈 알맞은 용어를 쓰시오.



답 :

2. 레이아웃 영역 중 필수 영역으로 시스템의 전체 개념을 나타내는 메인 이미지와 시스템별로 필요한 콘텐츠를 표시하는 영역을 쓰시오.

답 :

**정답** 1. 레이아웃 2. 내용 구성(Contents Area)



### [핵심 087] UI 요구사항 확인

- 새로 개발할 시스템에 적용할 UI 관련 요구사항을 조사해서 작성하는 단계이다.
- UI 요구사항 확인 순서

목표 정의	<ul style="list-style-type: none"> <li>• 사용자들을 대상으로 인터뷰를 진행한 후 사용자들의 의견이 수렴된 비즈니스 요구사항을 정의한다.</li> <li>• 인터뷰 진행 시 유의사항 <ul style="list-style-type: none"> <li>- 인터뷰는 가능하면 개별적으로 진행한다.</li> <li>- 인터뷰는 한 시간을 넘지 않도록 한다.</li> <li>- 인터뷰 진행은 반드시 사용자 리서치를 시작하기 전에 해야 한다.</li> </ul> </li> </ul>
활동 사항 정의	조사한 요구사항을 토대로 앞으로 해야 할 활동 사항을 정의한다.
UI 요구사항 작성	<ul style="list-style-type: none"> <li>• 여러 경로를 통해 수집된 사용자들의 요구사항을 검토하고 분석하여 UI 개발 목적에 맞게 작성한다.</li> <li>• UI 요구사항은 반드시 실사용자 중심으로 작성되어야 한다.</li> <li>• UI 요구사항 작성 순서 <ul style="list-style-type: none"> <li>- 요구사항 요소 확인 : 파악된 요구사항 요소의 종류와 각각의 표현 방식 등을 검토 ※ 요구사항 요소 : 데이터 요구, 기능 요구, 제품/서비스의 품질, 제약 사항</li> <li>- 정황 시나리오 작성 : 정황 시나리오는 사용자의 요구사항을 도출하기 위해 작성하는 것으로, 사용자가 목표를 달성하기 위해 수행하는 방법을 순차적으로 묘사함</li> <li>- 요구사항 작성 : 정황 시나리오를 토대로 작성</li> </ul> </li> </ul>

#### 1. 다음 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

( )는 사용자가 목표를 달성하기 위해 수행하는 방법을 순차적으로 묘사한 것으로 UI 요구사항은 ( )를 토대로 작성한다.

답 :

**정답 1.** 정황 시나리오

### [핵심 088] UI 흐름 설계

- 업무의 진행 과정이나 수행 절차에 따른 흐름을 파악하여 화면과 폼을 설계하는 단계이다.
- UI 흐름 설계 순서

기능 작성	화면에 표현할 기능을 작성하는 단계
입력 요소 확인	화면에 표현되어야 할 기능을 확인한 후 화면에 입력할 요소를 확인하는 단계
유스케이스 설계	<ul style="list-style-type: none"> <li>• UI 요구사항을 기반으로 UI 유스케이스를 설계하는 단계</li> <li>• 유스케이스는 화면에 표현할 입력 요소들의 형태나 입력 방법, 배치 등을 고려해서 설계한다.</li> </ul>
기능 및 양식 확인	<ul style="list-style-type: none"> <li>• 분석한 기능을 토대로 텍스트 박스, 콤보 박스, 라디오 박스, 체크 박스 등을 확인하고 규칙을 정의한다.</li> <li>• 텍스트 박스(Text Box) : 입력이 가능함을 표시</li> <li>• 콤보 박스(Combo Box) : 목록에서 항목을 선택하거나 입력할 수 있음</li> <li>• 라디오 박스(Radio Box) : 여러 개의 값 중 하나만을 선택할 수 있음</li> <li>• 체크 박스(Check Box) : 여러 개의 값 중 하나 이상을 선택할 수 있음</li> </ul>

#### 1. 다음 괄호(①~③)에 들어갈 가장 적합한 컨트롤의 이름을 쓰시오.

( ① )	여러 개의 값 중 하나 이상을 선택할 수 있다.
( ② )	여러 개의 값 중 하나만을 선택할 수 있다.
( ③ )	목록에서 항목을 선택하거나 입력할 수 있다.

답

- ① :  
② :  
③ :

**정답 1.** ① 체크 박스(Check Box) ② 라디오 박스(Radio Box)  
③ 콤보 박스(Combo Box)

**[핵심 089] UI 상세 설계**

- 실제 설계 및 구현을 위해 모든 화면에 대해 자세하게 설계를 진행하는 단계이다.
- UI 상세 설계 순서

요구사항 확인	UI 상세 설계를 위한 요구사항을 최종적으로 확인하는 단계이다.
UI 설계서 표지 및 개정 이력 작성	<ul style="list-style-type: none"> <li>• UI 설계서 표지는 다른 문서와 혼동되지 않도록 프로젝트명이나 시스템명을 포함시켜 작성한다.</li> <li>• UI 설계서 개정 이력은 UI 설계서가 수정될 때마다 어떤 부분이 어떻게 수정되었는지를 정리해 놓은 문서이다.             <ul style="list-style-type: none"> <li>- 처음 작성 시 첫 번째 항목을 '초안 작성', 버전(Version)을 1.0으로 설정한다.</li> <li>- UI 설계서에 변경 사항이 있을 때마다 변경 내용을 적고 버전을 0.1씩 높인다.</li> </ul> </li> </ul>
UI 구조 설계	UI 요구사항과 UI 프로토타입에 기초하여 UI 구조를 설계하는 단계이다.
메뉴 구조 설계	사이트 맵 구조를 통해 사용자 기반 메뉴 구조를 설계 하는 단계이다.
화면 설계	UI 프로토타입과 UI 프로세스를 참고하여 필요한 화면을 페이지별로 설계하는 단계이다.

**1. 다음 괄호(①, ②)에 들어갈 가장 적합한 내용을 쓰시오.**

UI 설계서가 수정될 때마다 어떤 부분이 어떻게 수정되었는지를 알수 있도록 UI 설계서 개정 이력을 작성해야 한다. UI 설계서를 처음 작성할 때 첫 번째 항목을 '초안 작성', 버전(Version)을 ( ① )으로 설정하고, 설계서에 변경 사항이 있을 때마다 변경 내용을 적고 버전을 ( ② )씩 높인다.

**답**

① :

② :

**정답** 1. ① 1.0 ② 0.1**7장 | 애플리케이션 테스트 관리****[핵심 090] 애플리케이션 테스트**

- 애플리케이션에 잠재되어 있는 결함을 찾아내는 일련의 행위 또는 절차이다.
- 애플리케이션 테스트는 개발된 소프트웨어가 고객의 요구사항을 만족시키는지 확인(Validation)하고 소프트웨어가 기능을 정확히 수행하는지 검증(Verification)한다.
- 애플리케이션 테스트의 기본 원리
  - 완벽한 테스트 불가능 : 애플리케이션 테스트는 소프트웨어의 잠재적인 결함을 줄일 수 있지만 소프트웨어에 결함이 없다고 증명할 수는 없음. 즉 완벽한 소프트웨어 테스트는 불가능함
  - 결함 집중(Defect Clustering) : 애플리케이션의 결함은 대부분 개발자의 특성이나 애플리케이션의 기능적 특징 때문에 특정 모듈에 집중되어 있으며, 애플리케이션의 20%에 해당하는 코드에서 전체 결함의 80%가 발견된다고 하여 파레토 법칙(Pareto Principle)을 적용하기도 함
  - 살충제 패러독스(Pesticide Paradox) : 애플리케이션 테스트에서는 동일한 테스트 케이스로 동일한 테스트를 반복하면 더 이상 결함이 발견되지 않는 '살충제 패러독스(Pesticide Paradox) 현상이 발생하며, 살충제 패러독스를 방지하기 위해서 테스트 케이스를 지속적으로 보완 및 개선해야 함
  - 테스트는 정황(Context) 의존 : 애플리케이션 테스트는 소프트웨어 특징, 테스트 환경, 테스터 역량 등 정황(Context)에 따라 테스트 결과가 달라질 수 있으므로, 정황에 따라 테스트를 다르게 수행해야 함
  - 오류-부재의 궤변(Absence of Errors Fallacy) : 소프트웨어의 결함을 모두 제거해도 사용자의 요구사항을 만족시키지 못하면 해당 소프트웨어는 품질이 높다고 말할 수 없음을 의미함

2020년 1회 기사 실기

1. 애플리케이션 테스트에서 살충제 패러독스(Pesticide Paradox)의 개념을 설명하시오.

답 :

2. 소프트웨어 테스트에서 오류의 80%는 전체 모듈의 20% 내에서 발견된다는 법칙이 무엇인지 쓰시오.

답 :

**정답** 1. 살충제 패러독스는 동일한 테스트 케이스를 반복하면 더 이상 새로운 결함이 발견되지 않는 현상이다.

2. 파레토 법칙(Pareto Principle)

### [핵심091] 애플리케이션 테스트의 분류

#### • 프로그램 실행 여부에 따른 테스트

정적 테스트	프로그램을 실행하지 않고 명세서나 소스 코드를 대상으로 분석하는 테스트
동적 테스트	프로그램을 실행하여 오류를 찾는 테스트로, 소프트웨어 개발의 모든 단계에서 테스트를 수행할 수 있음

#### • 테스트 기반(Test Bases)에 따른 테스트

명세 기반 테스트	사용자의 요구사항에 대한 명세를 빠짐없이 테스트 케이스로 만들어 구현하고 있는지 확인하는 테스트
구조 기반 테스트	소프트웨어 내부의 논리 흐름에 따라 테스트 케이스를 작성하고 확인하는 테스트
경험 기반 테스트	유사 소프트웨어나 기술 등에 대한 테스트의 경험을 기반으로 수행하는 테스트

#### • 시각에 따른 테스트

검증(Verification) 테스트	개발자의 시각에서 제품의 생산 과정을 테스트하는 것으로, 제품이 명세서대로 완성됐는지를 테스트함
확인(Validation) 테스트	사용자의 시각에서 생산된 제품의 결과를 테스트하는 것으로, 사용자가 요구한대로 제품이 완성됐는지, 제품이 정상적으로 동작하는지를 테스트함

#### • 목적에 따른 테스트

회복(Recovery) 테스트	시스템에 여러 가지 결함을 주어 실패하도록 한 후 올바르게 복구되는지를 확인하는 테스트
안전(Security) 테스트	시스템에 설치된 시스템 보호 도구가 불법적인 침입으로부터 시스템을 보호할 수 있는지를 확인하는 테스트
강도(Stress) 테스트	시스템에 과도한 정보량이나 빈도 등을 부과하여 과부하 시에도 소프트웨어가 정상적으로 실행되는지를 확인하는 테스트
성능(Performance) 테스트	소프트웨어의 실시간 성능이나 전체적인 효율성을 진단하는 테스트로, 소프트웨어의 응답 시간, 처리량 등을 테스트
구조(Structure) 테스트	소프트웨어 내부의 논리적인 경로, 소스 코드의 복잡도 등을 평가하는 테스트
회귀(Regression) 테스트	소프트웨어의 변경 또는 수정된 코드에 새로운 결함이 없음을 확인하는 테스트
병행(Parallel) 테스트	변경된 소프트웨어와 기존 소프트웨어에 동일한 데이터를 입력하여 결과를 비교하는 테스트

1. 다음은 애플리케이션 테스트의 종류를 설명한 것이다. 서로 관련 있는 것끼리 연결하시오.

- |             |   |  |
|-------------|---|--|
| ① 구조 기반 테스트 | • | ⑥ 프로그램을 실행하지 않고 명세서나 소스 코드를 대상으로 분석하는 테스트          |
| ② 동적 테스트    | • | ⑦ 소프트웨어의 실시간 성능이나 전체적인 효율성을 진단하는 테스트               |
| ③ 회복 테스트    | • | ⑧ 프로그램을 실행하여 오류를 찾는 테스트                            |
| ④ 정적 테스트    | • | ⑨ 소프트웨어 내부의 논리 흐름에 따라 테스트 케이스를 작성하고 확인하는 테스트       |
| ⑤ 성능 테스트    | • | ⑩ 시스템에 여러 가지 결함을 주어 실패하도록 한 후 올바르게 복구되는지를 확인하는 테스트 |



답

- ① :  
② :  
③ :  
④ :  
⑤ :

2. 다음은 애플리케이션 테스트 중 시각에 따른 테스트에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 테스트를 쓰시오.

( ① ) 테스트는 개발자의 시각에서 제품의 생산 과정을 테스트하는 것으로, 제품이 명세서대로 완성됐는지를 테스트하는 것이고, ( ② ) 테스트는 사용자의 시각에서 생산된 제품의 결과를 테스트하는 것으로, 사용자가 요구한대로 제품이 완성됐는지, 제품이 정상적으로 동작하는지를 테스트한다.

답

- ① :  
② :

**정답** 1. ①-㉔ ②-㉓ ③-㉒ ④-㉑ ⑤-㉐  
2. ① 검증(Verification) ② 확인(Validation)



**[핵심 092] 화이트박스 테스트(White Box Test)**

- 모듈의 원시 코드를 오픈시킨 상태에서 원시 코드의 논리적인 모든 경로를 테스트하여 테스트 케이스를 설계하는 방법이다.
- 원시 코드(Source Code)의 모든 문장을 한 번 이상 실행함으로써 수행된다.
- 모듈 안의 작동을 직접 관찰할 수 있다.
- 화이트박스 테스트의 종류

기초 경로 검사	<ul style="list-style-type: none"> <li>• 테스트 케이스 설계자가 절차적 설계의 논리적 복잡성을 측정할 수 있게 해주는 테스트 기법</li> <li>• 테스트 측정 결과는 실행 경로의 기초를 정의하는 데 지침으로 사용됨</li> </ul>
----------	--

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

- 화이트박스 테스트 검증 기준 : 테스트 케이스들이 테스트에 얼마나 적정한지를 판단하는 기준으로, 문장 검증 기준(Statement Coverage), 분기 검증 기준(Branch Coverage), 조건 검증 기준(Condition Coverage), 분기/조건 기준(Branch/Condition Coverage) 등이 있음

1. 다음이 설명하는 애플리케이션 테스트는 무엇인지 쓰시오.

- 원시 코드의 논리적인 모든 경로를 테스트하는 방법이다.
- Source Code의 모든 문장을 한 번 이상 실행함으로써 수행된다.
- 모듈 안의 작동을 직접 관찰할 수 있다.
- 대표적인 기법에는 기초 경로 검사와 제어 구조 검사가 있다.

답 :

2020년 1회 기공사 실기

2. 다음 <보기>에서 내부 소스 코드를 테스트하는 화이트박스 테스트에 속하지 않는 테스트 기법을 모두 골라 쓰시오.

<보기>

- |              |          |
|--------------|----------|
| • 기초 경로 테스트  | • 조건 테스트 |
| • 한계값 분석     | • 루프 테스트 |
| • 데이터 흐름 테스트 | • 비교 테스트 |

답 :

**해설** 2. 한계값 분석과 비교 테스트는 블랙박스 테스트에 해당합니다.

**정답** 1. 화이트박스 테스트(White Box Test)  
2. 한계값 분석, 비교 테스트



### [핵심 093] 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.
- 블랙박스 테스트의 종류

동치 분할 검사	입력 자료에 초점을 맞춰 테스트 케이스를 만들고 검사하는 기법
경계값 분석	입력 자료에만 치중한 동치 분할 기법을 보완하기 위한 기법으로, 입력 조건의 경계값을 테스트 케이스로 선정하여 검사함
원인 - 효과 그래프 검사	입력 데이터 간의 관계와 출력에 영향을 미치는 상황을 체계적으로 분석한 다음 효용성이 높은 테스트 케이스를 선정하여 검사하는 기법
오류 예측 검사	과거의 경험이나 확인자의 감각으로 테스트하는 기법
비교 검사	여러 버전의 프로그램에 동일한 테스트 자료를 제공하여 동일한 결과가 출력되는지 테스트하는 기법

1. 애플리케이션 테스트 중 블랙박스 테스트(Black Box Test)의 개념을 설명하시오.

답 :

2. 다음은 블랙박스 테스트(Black Box Test)의 종류별 특징이다. 괄호(①~③)에 들어갈 알맞은 종류를 쓰시오.

- ( ① ) : 여러 버전의 프로그램에 동일한 테스트 자료를 제공하여 동일한 결과가 출력되는지 테스트하는 기법이다.
- 원인-효과 그래프 검사 : 입력 데이터 간의 관계와 출력에 영향을 미치는 상황을 체계적으로 분석한 다음 효용성이 높은 테스트 케이스를 선정하여 검사하는 기법이다.
- ( ② ) : 입력 자료에만 치중한 동치 분할 기법을 보완하기 위한 기법이다.
- 오류 예측 검사 : 과거의 경험이나 확인자의 감각으로 테스트하는 기법이다.
- ( ③ ) : 입력 자료에 초점을 맞춰 테스트 케이스를 만들고 검사하는 기법이다.

답 :

**정답** 1. 블랙박스 테스트는 소프트웨어가 수행할 특정 기능을 알기 위해서 각 기능이 완전히 작동되는 것을 입증하는 테스트이다.

2. ① 비교 검사 ② 경계값 분석 ③ 동치 분할 검사



### [핵심 094] 개발 단계에 따른 애플리케이션 테스트

- 단위 테스트(Unit Test)
  - 코딩 직후 소프트웨어 설계의 최소 단위인 모듈이나 컴포넌트에 초점을 맞춰 테스트하는 것
  - 구조 기반 테스트 : 프로그램 내부 구조 및 복잡도를 검증하는 화이트박스(White Box) 테스트 시행
  - 명세 기반 테스트 : 목적 및 실행 코드 기반의 블랙박스(Black Box) 테스트 시행
- 통합 테스트(Integration Test) : 단위 테스트가 완료된 모듈들을 결합하여 하나의 시스템으로 완성시키는 과정에서의 테스트를 의미함
- 시스템 테스트(System Test) : 개발된 소프트웨어가 해당 컴퓨터 시스템에서 완벽하게 수행되는가를 점검하는 테스트
- 인수 테스트(Acceptance Test)
  - 개발한 소프트웨어가 사용자의 요구사항을 충족하는지에 중점을 두고 테스트하는 것
  - 알파 테스트 : 개발자의 장소에서 사용자가 개발자 앞에서 행하는 테스트 기법으로, 테스트는 통제된 환경에서 행해지며, 오류와 사용상의 문제점을 사용자와 개발자가 함께 확인하면서 기록함
  - 베타 테스트 : 선정된 최종 사용자가 여러 명의 사용자 앞에서 행하는 테스트 기법으로, 개발자에 의해 제어되지 않은 상태에서 테스트가 행해지며, 발견된 오류와 사용상의 문제점을 기록하고 개발자에게 주기적으로 보고함

2020년 1, 2회 기사 필기

1. 검증 검사 기법 중 개발자의 장소에서 사용자가 개발자 앞에서 행하는 기법으로, 일반적으로 통제된 환경에서 사용자와 개발자가 함께 확인하면서 수행되는 테스트는 무엇인지 쓰시오.

답 :

2. 애플리케이션 테스트는 소프트웨어의 개발 단계에 따라 통합, 시스템, 인수 테스트 등으로 분류된다. 다음 괄호에 공통적으로 들어갈 알맞은 테스트를 쓰시오.

- (     )는 코딩 직후 소프트웨어 설계의 최소 단위인 모듈이나 컴포넌트에 초점을 맞춰 테스트하는 것으로, 인터페이스, 외부적 I/O, 자료 구조, 독립적 기초 경로, 오류 처리 경로, 경계 조건 등을 검사한다.
- (     )는 구조 기반 테스트와 명세 기반 테스트로 나뉘지만 주로 구조 기반 테스트를 시행한다.

답 :

정답 1. 알파 테스트 2. 단위 테스트(Unit Test)

### [핵심 095] 통합 테스트(Integration Test)

- 단위 테스트가 끝난 모듈을 통합하는 과정에서 발생하는 오류 및 결함을 찾는 테스트 기법이다.
- 비점진적 통합 방식
  - 단계적으로 통합하는 절차 없이 모든 모듈이 미리 결합되어 있는 프로그램 전체를 테스트하는 방법
  - 규모가 작은 소프트웨어에 유리하며 단시간 내에 테스트가 가능함
  - 오류 발견 및 장애 위치 파악 및 수정이 어려움
  - 빅뱅 통합 테스트 : 모듈 간의 상호 인터페이스를 고려하지 않고 단위 테스트가 끝난 모듈을 한꺼번에 결합시켜 테스트하는 방법
- 점진적 통합 방식
  - 모듈 단위로 단계적으로 통합하면서 테스트하는 방법

- 오류 수정이 용이하고, 인터페이스와 연관된 오류를 완전히 테스트할 가능성이 높음
- 하향식 통합 테스트(Top Down Integration Test) : 프로그램의 상위 모듈에서 하위 모듈 방향으로 통합하면서 테스트하는 기법
- 상향식 통합 테스트(Bottom Up Integration Test) : 프로그램의 하위 모듈에서 상위 모듈 방향으로 통합하면서 테스트하는 기법
- 혼합식 통합 테스트 : 하위 수준에서는 상향식 통합, 상위 수준에서는 하향식 통합을 사용하여 최적의 테스트를 지원하는 방식으로, 샌드위치(Sandwich)식 통합 테스트 방법이라고도 함
- 회귀 테스트(Regression Test) : 이미 테스트된 프로그램의 테스트를 반복하는 것으로, 통합 테스트로 인해 변경된 모듈이나 컴포넌트에 새로운 오류가 있는지 확인하는 테스트

1. 다음 설명에 해당하는 테스트를 쓰시오.

- 단위 테스트가 끝난 모듈을 통합하는 과정에서 발생하는 오류 및 결함을 찾는 테스트 기법이다.
- 테스트 방법에는 비점진적 통합 방식과 점진적 통합 방식이 있다.

답 :

2. 통합 테스트 중 점진적 통합 방식은 모듈 단위로 단계적으로 통합하면서 테스트하는 방법으로 다음과 같은 종류가 있다. 괄호(①, ②)에 들어갈 알맞은 테스트를 쓰시오.

( ① ) 통합 테스트	프로그램의 하위 모듈에서 상위 모듈 방향으로 통합하면서 테스트하는 기법이다.
( ② ) 통합 테스트	프로그램의 상위 모듈에서 하위 모듈 방향으로 통합하면서 테스트하는 기법이다.
혼합식 통합 테스트	하위 수준에서는 ( ① ) 통합, 상위 수준에서는 ( ② ) 통합을 사용하여 최적의 테스트를 지원하는 방식이다.

답

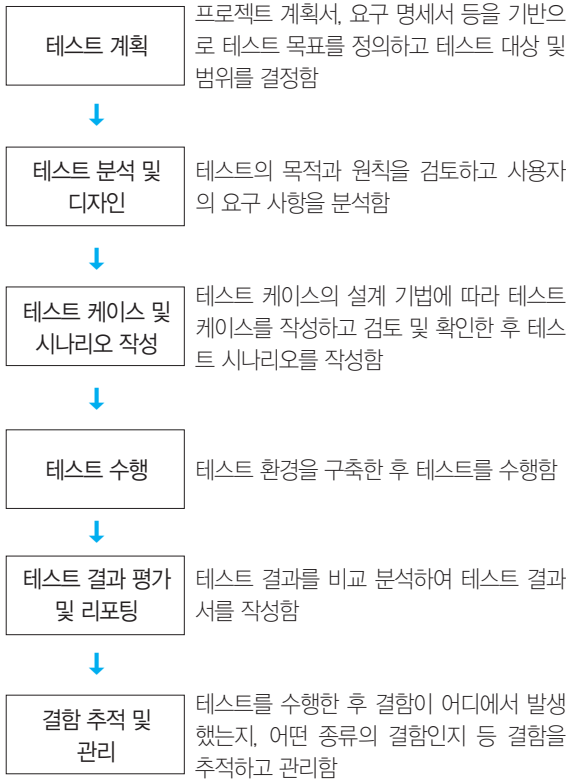
- ① :  
② :

정답 1. 통합 테스트(Integration Test)

2. ① 상향식(Bottom Up) ② 하향식(Top Down)

### [핵심 096] 애플리케이션 테스트 프로세스

개발된 소프트웨어가 사용자의 요구대로 만들어졌는지, 결함은 없는지 등을 테스트하는 절차이다.



1. 애플리케이션 테스트 프로세스는 개발된 소프트웨어가 사용자의 요구대로 만들어졌는지, 결함은 없는지 등을 테스트하는 절차이다. 다음 <보기>로 제시된 애플리케이션 테스트 프로세스 절차를 진행 순서대로 기호(㉠~㉥)로 나열하시오.

<보기>

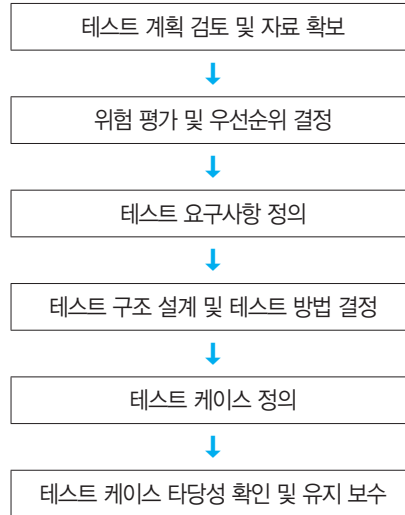
- ㉠ 테스트 분석 및 디자인
- ㉡ 테스트 결과 평가 및 리포팅
- ㉢ 테스트 수행
- ㉣ 테스트 계획
- ㉤ 테스트 케이스 및 시나리오 작성
- ㉥ 결함 추적 및 관리

답 :

정답 1. ㉣, ㉠, ㉤, ㉢, ㉡, ㉥

### [핵심 097] 테스트 케이스(Test Case)

- 구현된 소프트웨어가 사용자의 요구사항을 정확하게 준수했는지를 확인하기 위해 설계된 입력 값, 실행 조건, 기대 결과 등으로 구성된 테스트 항목에 대한 명세서로, 명세 기반 테스트의 설계 산출물에 해당된다.
- 테스트 케이스 작성 순서



1. 구현된 소프트웨어가 사용자의 요구사항을 정확하게 준수했는지를 확인하기 위해 설계된 입력 값, 실행 조건, 기대 결과 등으로 구성된 테스트 항목에 대한 명세서로, 명세 기반 테스트의 설계 산출물에 해당되는 것은 무엇인지 쓰시오.

답 :

2. 테스트 케이스는 테스트 전략이나 테스트 계획서 등을 기반으로 하여 다음과 같은 순서로 작성된다. 괄호(①, ②)에 들어갈 알맞은 작업 내용을 쓰시오.

테스트 계획 검토 및 자료 확보 → ( ① ) → 테스트 요구사항 정의 → 테스트 구조 설계 및 테스트 방법 결정 → ( ② ) → 테스트 케이스 타당성 확인 및 유지 보수

답

- ① :
- ② :

정답 1. 테스트 케이스(Test Case)

2. ① 위험 평가 및 우선순위 결정 ② 테스트 케이스 정의

**[핵심 098] 테스트 시나리오(Test Scenario)**

- 테스트 케이스를 적용하는 순서에 따라 여러 개의 테스트 케이스들을 묶은 집합으로, 테스트 케이스들을 적용하는 구체적인 절차를 명세한 문서이다.
- 테스트 시나리오에는 테스트 순서에 대한 구체적인 절차, 사전 조건, 입력 데이터 등이 설정되어 있다.
- 테스트 시나리오는 시스템별, 모듈별, 항목별 등과 같이 여러 개의 시나리오로 분리하여 작성해야 한다.
- 각각의 테스트 항목은 식별자 번호, 순서 번호, 테스트 데이터, 테스트 케이스, 예상 결과, 확인 등을 포함해서 작성해야 한다.
- 테스트 시나리오는 유스케이스(Use Case) 간 업무 흐름이 정상적인지를 테스트할 수 있도록 작성해야 한다.

1. 테스트 케이스를 적용하는 순서에 따라 여러 개의 테스트 케이스들을 묶은 집합으로, 테스트 케이스들을 적용하는 구체적인 절차를 명세한 문서는 무엇인지 쓰시오.

답 :

**정답** 1. 테스트 시나리오(Test Scenario)

**[핵심 099] 테스트 오라클(Test Oracle)**

- 테스트 결과가 올바른지 판단하기 위해 사전에 정의된 참 값을 대입하여 비교하는 기법 및 활동이다.
- 참(True) 오라클 : 모든 테스트 케이스의 입력 값에 대해 기대하는 결과를 제공하는 오라클로, 발생한 모든 오류를 검출할 수 있음
- 샘플링(Sampling) 오라클 : 특정한 몇몇 테스트 케이스의 입력 값들에 대해서만 기대하는 결과를 제공하는 오라클
- 추정(Heuristic) 오라클 : 샘플링 오라클을 개선한 오라클로, 특정 테스트 케이스의 입력값에 대해 기대하는 결과를 제공하고, 나머지 입력 값들에 대해서는 추정으로 처리하는 오라클
- 일관성 검사(Consistent) 오라클 : 애플리케이션의 변경이 있을 때, 테스트 케이스의 수행 전과 후의 결과 값이 동일한지를 확인하는 오라클

1. 다음은 테스트 오라클(Test Oracle)의 종류에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 테스트 오라클을 쓰시오.

일관성 검사 오라클	애플리케이션의 변경이 있을 때, 테스트 케이스의 수행 전과 후의 결과 값이 동일 한지를 확인하는 오라클이다.
참 오라클	모든 테스트 케이스의 입력 값에 대해 기대하는 결과를 제공하는 오라클로, 발생한 모든 오류를 검출할 수 있다.
( ① ) 오라클	샘플링 오라클을 개선한 오라클로, 특정 테스트 케이스의 입력값에 대해 기대하는 결과를 제공하고, 나머지 입력 값들에 대해서는 추정으로 처리하는 오라클이다.
( ② ) 오라클	특정한 몇몇 테스트 케이스의 입력 값들에 대해서만 기대하는 결과를 제공하는 오라클이다.

**답**

① :

② :

**정답** 1. ① 추정(Heuristic) ② 샘플링(Sampling)

**[핵심 100] 테스트 자동화 도구 유형**

- 사람이 반복적으로 수행하던 테스트 절차를 테스트 자동화 도구를 사용함으로써 휴먼 에러(Human Error)를 줄이고 테스트의 정확성을 유지하면서 테스트의 품질을 향상시킬 수 있다.
- ※ 휴먼 에러(Human Error) : 사람의 판단 실수나 조작 실수 등으로 인해 발생하는 에러
- 정적 분석 도구(Static Analysis Tools) : 프로그램을 실행하지 않고 분석하는 도구로, 소스 코드에 대한 코딩 표준, 코딩 스타일, 코드 복잡도 및 남은 결함 등을 발견하기 위해 사용됨
- 테스트 실행 도구(Test Execution Tools) : 스크립트 언어를 사용하여 테스트를 실행하는 방법으로, 테스트 데이터와 테스트 수행 방법 등이 포함된 스크립트를 작성한 후 실행함

- 성능 테스트 도구(Performance Test Tools) : 애플리케이션의 처리량, 응답 시간, 경과 시간, 자원 사용률 등을 인위적으로 적용한 가상의 사용자를 만들어 테스트를 수행함으로써 성능의 목표 달성 여부를 확인함
- 테스트 통제 도구(Test Control Tools) : 테스트 계획 및 관리, 테스트 수행, 결함 관리 등을 수행하는 도구로, 종류에는 형상 관리 도구, 결함 추적/관리 도구 등이 있음
- 테스트 하네스 도구(Test Harness Tools) : 테스트가 실행될 환경을 시뮬레이션 하여 컴포넌트 및 모듈이 정상적으로 테스트 되도록 하는 도구

2020년 1회 기능사 실기

1. 인간의 실수 등을 통해 원래의 의도와 다르게 소프트웨어가 예정된 설계를 벗어나 발생하는 오류를 의미하는 용어를 쓰시오.

답 :

2. 프로그램을 실행하지 않고 분석하는 도구로, 소스 코드에 대한 코딩 표준, 코딩 스타일, 코드 복잡도 및 남은 결함 등을 발견하기 위해 사용되는 테스트 자동화 도구는 무엇인지 쓰시오.

답 :

정답 1. 휴먼 에러(Human Error)

2. 정적 분석 도구(Static Analysis Tools)

[핵심101] 결함 관리(Fault)

- 결함은 오류 발생, 작동 실패 등과 같이 소프트웨어가 개발자가 설계한 것과 다르게 동작하거나 다른 결과가 발생하는 것을 의미한다.
- 결함 관리 프로세스 처리 순서 : 결함 관리 계획 → 결함 기록 → 결함 검토 → 결함 수정 → 결함 재확인 → 결함 상태 추적 및 모니터링 활동 → 최종 결함 분석 및 보고서 작성
- 결함 관리 측정 지표

결함 분포	모듈 또는 컴포넌트의 특정 속성에 해당하는 결함 수 측정
결함 추세	테스트 진행 시간에 따른 결함 수의 추이 분석
결함 에이징	특정 결함 상태로 지속되는 시간 측정

- 결함 추적 순서 : 결함 등록(Open) → 결함 검토(Reviewed) → 결함 할당(Assigned) → 결함 수정(Resolved) → 결함 종료(Closed) → 결함 해제(Clarified)

※ Fixed(고정) : 개발자가 필요한 변경 작업을 수행하여 결함 수정 작업을 완료한 상태

• 결함 분류

시스템 결함	애플리케이션 환경이나 데이터베이스 처리에서 발생한 결함
기능 결함	애플리케이션의 기획, 설계, 업무 시나리오 등의 단계에서 유입된 결함
GUI 결함	사용자 화면 설계에서 발생한 결함
문서 결함	기획자, 사용자, 개발자 간의 의사소통 및 기록이 원활하지 않아 발생한 결함

• 결함 심각도

High	더 이상 프로세스를 진행할 수 없도록 만드는 결함
Medium	시스템 흐름에 영향을 미치는 결함
Low	시스템 흐름에는 영향을 미치지 않는 결함

- 결함 우선순위 : 결정적(Critical), 높음(High), 보통(Medium), 낮음(Low) 또는 즉시 해결, 주의 요망, 대기, 개선 권고 등

1. 다음은 테스트에서 발견되는 결함을 유형별로 분류한 것이다. 괄호(①, ②)에 들어갈 알맞은 결함을 쓰시오.

문서 결함	사용자의 온라인/오프라인 매뉴얼의 불일치 등 기획자, 사용자, 개발자 간의 의사소통 및 기록이 원활하지 않아 발생한 결함이다.
( ① ) 결함	UI 비밀관선, 데이터 타입의 표시 오류, 부정확한 커서/메시지 오류 등 사용자 화면 설계에서 발생한 결함이다.
기능 결함	사용자의 요구사항 미반영/불일치, 스크립트 오류, 타시스템 연동 시 오류 등 애플리케이션의 기획, 설계, 업무 시나리오 등의 단계에서 유입된 결함이다.
( ② ) 결함	시스템 다운, 애플리케이션의 작동 정지, 종료, 응답 시간 지연, 데이터베이스 에러 등 주로 애플리케이션 환경이나 데이터베이스 처리에서 발생한 결함이다.

답

- ① :  
② :

2. 결함 추적은 결함이 발견된 때부터 결함이 해결될 때까지 전 과정을 추적하는 것이다. 다음에 제시된 결함 추적 단계들을 순서대로 기호(㉠~㉢)로 나열하시오..

- |                    |                   |
|--------------------|-------------------|
| ㉠ 결함 수정(Resolved)  | ㉡ 결함 등록(Open)     |
| ㉢ 결함 해제(Clarified) | ㉣ 결함 검토(Reviewed) |
| ㉤ 결함 할당(Assigned)  | ㉥ 결함 종료(Closed)   |

답 :

2020년 2회 기사 실기

3. 결함 관리 과정에서 사용되는 용어인 Fixed의 의미를 쓰시오.

답 :

정답 1. ① GUI ② 시스템 2. ㉡, ㉢, ㉣, ㉤, ㉥

3. Fixed는 개발자가 필요한 변경 작업을 수행하여 결함 수정 작업을 완료한 상태이다.

### [핵심102] 애플리케이션 성능

- 사용자가 요구한 기능을 최소한의 자원을 사용하여 최대한 많은 기능을 신속하게 처리하는 정도를 나타낸다.
- 애플리케이션 성능 측정 지표

처리량 (Throughput)	일정 시간 내에 애플리케이션이 처리하는 일의 양
응답 시간 (Response Time)	애플리케이션에 요청을 전달한 시간부터 응답이 도착할 때까지 걸린 시간
경과 시간 (Turn Around Time)	애플리케이션에 작업을 의뢰한 시간부터 처리가 완료될 때까지 걸린 시간
자원 사용률 (Resource Usage)	애플리케이션이 의뢰한 작업을 처리하는 동안의 CPU 사용량, 메모리 사용량, 네트워크 사용량 등 자원 사용률

2020년 1회 기사 실기

1. 애플리케이션 성능이란 사용자가 요구한 기능을 최소한의 자원을 사용하여 최대한 많은 기능을 신속하게 처리하는 정도를 나타낸다. 애플리케이션 성능 측정의 지표에 대한 설명에서 괄호(①~③)에 들어갈 알맞은 지표를 쓰시오.

자원 사용률	애플리케이션이 의뢰한 작업을 처리하는 동안의 CPU 사용량, 메모리 사용량, 네트워크 사용량 등 자원 사용률
( ① )	애플리케이션에 요청을 전달한 시간부터 응답이 도착할 때까지 걸린 시간
( ② )	일정 시간 내에 애플리케이션이 처리하는 일의 양
( ③ )	애플리케이션에 작업을 의뢰한 시간부터 처리가 완료될 때까지 걸린 시간

답

- ① :  
② :  
③ :

정답 ① 응답 시간(Response Time) ② 처리량(Throughput)  
③ 경과 시간(Turn Around Time)

### [핵심103] 소스 코드 최적화

- 나쁜 코드(Bad Code)를 배제하고, 클린 코드(Clean Code)로 작성하는 것이다.
- 클린 코드(Clean Code) : 누구나 쉽게 이해하고 수정 및 추가할 수 있는 단순, 명료한 코드
- 나쁜 코드(Bad Code) : 코드의 로직이 서로 얽혀 있는 스파게티 코드 등 프로그램의 로직(Logic)이 복잡하고 이해하기 어려운 코드
- 나쁜 코드로 작성된 애플리케이션의 코드를 클린 코드로 수정하면 애플리케이션의 성능이 개선된다.
- 클린 코드 작성 원칙 : 가독성, 단순성, 의존성 배제, 중복성 최소화, 추상화



## • 소스 코드 최적화 유형

- 클래스 분할 배치 : 하나의 클래스는 하나의 역할만 수행하도록 응집도를 높이고, 크기를 작게 작성함
- Loosely Coupled(느슨한 결합) : 인터페이스 클래스를 이용하여 추상화된 자료 구조와 메소드를 구현함으로써 클래스 간의 의존성을 최소화함
- 코딩 형식 준수, 좋은 이름 사용, 적절한 주석문 사용

1. 소스 코드 최적화와 관련된 다음 설명에서 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오.

- 소스 코드 최적화는 ( ① )를 배제하고, ( ② )로 작성하는 것이다.
- ( ② )는 누구나 쉽게 이해하고 수정 및 추가할 수 있는 단순, 명료한 코드를, ( ① )는 프로그램의 Logic이 복잡하고 이해하기 어려운 코드를 의미한다.

답 :

① :

② :

2. 소스 코드 최적화 유형 중 인터페이스 클래스를 이용하여 추상화된 자료 구조와 메소드를 구현함으로써 클래스 간의 의존성을 최소화 하는 것을 의미하는 용어를 쓰시오.

답 :

정답 1. ① 나쁜 코드(Bad Code) ② 클린 코드(Clean Code)

2. Loosely Coupled(느슨한 결합)



## 8장 | SQL 응용

### [핵심104] DDL – CREATE TABLE

#### DDL(Data Define Language, 데이터 정의어)

- DB 구조, 데이터 형식, 접근 방식 등 DB를 구축하거나 수정할 목적으로 사용하는 언어이다.
- DDL은 번역한 결과가 데이터 사전(Data Dictionary)이라는 특별한 파일에 여러 개의 테이블로서 저장된다.
- DDL에는 CREATE SCHEMA, CREATE DOMAIN, CREATE TABLE, CREATE VIEW, CREATE INDEX, ALTER TABLE, DROP 등이 있다.

#### CREATE TABLE

- 테이블을 정의하는 명령문이다.
- 표기 형식

CREATE TABLE 테이블명

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

- PRIMARY KEY : 기본키로 사용할 속성 또는 속성의 집합을 지정함
- UNIQUE : 대체키로 사용할 속성 또는 속성의 집합을 지정하는 것으로 UNIQUE로 지정한 속성은 중복된 값을 가질 수 없음
- FOREIGN KEY ~ REFERENCES ~
  - ON DELETE 옵션 : 참조 테이블의 튜플이 삭제되었을 때 기본 테이블에 취해야 할 사항을 지정함. 옵션에는 NO ACTION, CASCADE, SET NULL, SET DEFAULT가 있음
  - ON UPDATE 옵션 : 참조 테이블의 참조 속성 값이 변경되었을 때 기본 테이블에 취해야 할 사항을 지

정함. 옵션에는 NO ACTION, CASCADE, SET NULL, SET DEFAULT가 있음

- CONSTRAINT : 제약 조건의 이름을 지정함. 이름을 지정할 필요가 없으면 CHECK절만 사용하여 속성 값에 대한 제약 조건을 명시함
- CHECK : 속성 값에 대한 제약 조건을 정의함

### 다른 테이블을 이용한 테이블 정의

- 기존 테이블의 정보를 이용해 새로운 테이블을 정의할 수 있다.
- 표기 형식

```
CREATE TABLE 신규테이블명 AS SELECT 속성명[, 속성명, ...] FROM 기존테이블명;
```

1. 아래의 <요구사항>을 만족하는 테이블 <Instructor>를 정의하는 SQL의 괄호(① ~ ③)에 알맞은 명령을 쓰시오.

#### <요구사항>

- 'id(문자 5)', 'name(문자 15)', 'dept(문자 15)' 속성을 가진다.
- 'id' 속성은 기본키이다.
- 'name' 속성은 Null이 올 수 없다.
- 'dept' 속성은 <Department> 테이블의 'name' 속성을 참조하는 외래키이다.
  - <Department> 테이블에서 튜플이 삭제되면 관련된 모든 튜플의 'dept' 속성의 값은 NULL로 변경되어야 한다.
  - <Department> 테이블의 'name' 속성이 변경되면 <Instructor> 테이블의 관련된 모든 속성 값도 같은 값으로 변경되어야 한다.

#### <SQL문>

```
CREATE TABLE Instructor
(
  id CHAR(5),
  name CHAR(15) ( ① ),
  dept CHAR(15),
  PRIMARY KEY(id),
  FOREIGN KEY(dept) ( ② )
    Department(name)
      ON DELETE SET NULL
      ON UPDATE ( ③ )
);
```

#### 답

- ① :  
② :  
③ :

2. <사원> 테이블의 '사번', '성명', '직급' 속성을 이용하여 <진급사원> 테이블을 정의하는 SQL문을 작성하시오.

#### 답

##### 해설 1.

```
① CREATE TABLE Instructor
② ( id CHAR(5),
③ name CHAR(15) NOT NULL,
④ dept CHAR(15),
⑤ PRIMARY KEY(id),
⑥ FOREIGN KEY(dept) REFERENCES
  Department(name)
⑦ ON DELETE SET NULL
⑧ ON UPDATE CASCADE
);
```

- ① 생성한 테이블 이름은 <Instructor>이다.
- ② 'id' 속성은 문자 5자리이다.
- ③ 'name' 속성은 문자 15자리이며, NULL 값을 갖지 않는다.
- ④ 'dept' 속성은 문자 15자리이다.
- ⑤ 'id' 속성은 기본키이다.
- ⑥ 'dept' 속성은 <Department> 테이블의 기본키인 'name' 속성을 참조하는 외래키이다.
- ⑦ <Department> 테이블에서 튜플이 삭제되면 관련된 모든 튜플의 'dept' 속성의 값을 NULL로 변경한다.
- ⑧ <Department> 테이블에서 튜플이 변경되면 관련된 모든 튜플의 'dept' 속성의 값도 같은 값으로 변경한다.

**정답** 1. ① Not Null ② References ③ Cascade

2. CREATE TABLE 진급사원 AS SELECT 사번, 성명, 직급 FROM 사원;

### [핵심105] DDL – CREATE VIEW

- 뷰(View)를 정의하는 명령문이다.
- 표기 형식

```
CREATE VIEW 뷰명[속성명[, 속성명, ...]]
AS SELECT문;
```



- SELECT문을 서브 쿼리로 사용하여 SELECT문의 결과로서 뷰를 생성한다.
- 서브 쿼리인 SELECT문에는 UNION이나 ORDER BY 절을 사용할 수 없다.
- 속성명을 기술하지 않으면 SELECT문의 속성명이 자동으로 사용된다.

2020년 1회 기능사 실기

1. <학생> 테이블에서 2학년 학생들의 성명, 사진, 학년으로 구성된 뷰를 '출석부'라는 이름으로 생성하려고 한다. <출석부> 뷰를 생성하는 SQL문의 괄호에 알맞은 명령을 쓰시오.

<SQL문>

```
CREATE VIEW 출석부
(      ) 성명, 사진, 학년
FROM 학생
WHERE 학년 = 2;
```

답 :

해설 1.

- CREATE VIEW 출석부 : 생성한 뷰의 이름은 <출석부>이다.
- AS SELECT 성명, 사진, 학년 : '성명', '사진', '학년' 속성을 가져온다.
- FROM 학생 : <학생> 테이블에서 속성을 가져와 뷰를 생성한다.
- WHERE 학년 = 2 : '학년' 속성의 값이 2인 자료만을 뷰로 생성한다.

정답 1. AS SELECT

[핵심106] DDL – ALTER TABLE

- 테이블에 대한 정의를 변경하는 명령문이다.
- 표기 형식

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

- ADD : 새로운 속성(열)을 추가할 때 사용
- ALTER | MODIFY : 특정 속성의 정의를 변경할 때 사용
- DROP COLUMN : 특정 속성을 삭제할 때 사용

2020년 2회 기능사 실기

1. 테이블의 구조를 변경할 때 사용하는 DDL 명령어를 쓰시오.  
답 :

2020년 2회 기능사 실기

2. <HRD> 테이블의 '주민번호' 속성에 "900351"을 입력할 때 오류가 발생하여 <HRD> 테이블의 '주민번호' 속성의 크기를 6으로 변경하는 SQL문을 작성하였다. 괄호에 알맞은 명령을 쓰시오.

<SQL문>

```
ALTER TABLE HRD (      ) 주민번호 NUMBER(6);
```

답 :

정답 1. ALTER 2. ALTER 또는 MODIFY

[핵심107] DDL – DROP TABLE

- 기본 테이블을 제거하는 명령문이다.
- 표기 형식

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

- 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

2020년 1회 기능사 실기

1. SQL 중 데이터 정의어(DDL)에서 테이블 구조를 제거할 때 사용하는 명령어를 쓰시오.

답 :

2. 다음 <처리 조건>에 부합하는 SQL문을 작성하시오.  
<처리 조건>

- <사원> 테이블을 제거한다.
- <사원> 테이블을 참조하는 모든 데이터도 함께 제거한다.

답 :

정답 1. DROP 2. DROP TABLE 사원 CASCADE;

**[핵심 108] DCL – GRANT / REVOKE****DCL(Data Control Language, 데이터 제어어)**

- 데이터의 보안, 무결성, 회복, 병행 제어 등을 정의하는 데 사용하는 언어이다.
- DCL은 데이터베이스 관리자(DBA)가 데이터 관리를 목적으로 사용한다.
- DCL에는 GRANT, REVOKE, COMMIT, ROLLBACK, SAVEPOINT 등이 있다.

**GRANT / REVOKE**

- 데이터베이스 관리자가 데이터베이스 사용자에게 권한을 부여하거나 취소하기 위한 명령어이다.
- GRANT : 권한 부여를 위한 명령어
- REVOKE : 권한 취소를 위한 명령어
- 사용자등급 지정 및 해제

– GRANT 사용자등급 TO 사용자\_ID\_리스트 [IDENTIFIED BY 암호];  
 – REVOKE 사용자등급 FROM 사용자\_ID\_리스트;

- 테이블 및 속성에 대한 권한 부여 및 취소

– GRANT 권한\_리스트 ON 개체 TO 사용자 [WITH GRANT OPTION];  
 – REVOKE [GRANT OPTION FOR] 권한\_리스트 ON 개체 FROM 사용자 [CASCADE];

- 권한 종류 : ALL, SELECT, INSERT, DELETE, UPDATE, ALTER 등
- WITH GRANT OPTION : 부여받은 권한을 다른 사용자에게 다시 부여할 수 있는 권한을 부여함
- GRANT OPTION FOR : 다른 사용자에게 권한을 부여할 수 있는 권한을 취소함
- CASCADE : 권한 취소 시 권한을 부여받았던 사용자가 다른 사용자에게 부여한 권한도 연쇄적으로 취소함

2020년 1회 기능사 실기, 2020년 1, 2회 기사 필기

1. 관리자가 사용자에게 데이터베이스에 대한 권한을 부여 또는 회수하거나 데이터 보안, 무결성 유지, 병행수행 제어 등의 기능을 수행하기 위해 사용하는 언어를 영문 약어로 쓰시오.

**답 :**

2. 김영웅에게 <사원> 테이블에 대한 접근 및 조작에 관한 모든 권한을 부여하는 SQL문을 작성하시오.

**답 :**

3. 이민지에게 부여된 <고객> 테이블에 대한 SELECT, INSERT, DELETE 권한을 취소하는 SQL문을 작성하시오.

**답 :****정답** 1. DCL 2. GRANT ALL ON 사원 TO 김영웅;

3. REVOKE SELECT, INSERT, DELETE ON 고객 FROM 이민지;

**[핵심 109] COMMIT / ROLLBACK / SAVEPOINT**

COMMIT	트랜잭션이 성공적으로 끝나면 데이터베이스가 새로운 일관성(Consistency) 상태를 가지기 위해 변경된 모든 내용을 데이터베이스에 반영하여야 하는데, 이때 사용하는 명령어
ROLLBACK	아직 COMMIT되지 않은 변경된 모든 내용들을 취소하고 데이터베이스를 이전 상태로 되돌리는 명령어
SAVEPOINT	트랜잭션 내에 ROLLBACK 할 위치인 저장점을 지정하는 명령어로, 저장점을 지정할 때는 이름을 부여하며, ROLLBACK 시 지정된 저장점까지의 트랜잭션 처리 내용이 취소됨

1. 트랜잭션은 데이터베이스에서 하나의 논리적 기능을 수행하기 위한 일련의 연산 집합으로서 작업의 단위가 된다. 트랜잭션의 연산 ROLLBACK과 COMMIT 중 ROLLBACK의 개념을 간략히 서술하시오.

**답 :**



핵심요약 115

2.

- DELETE : 삭제한다.
- FROM 사원 : (사원) 테이블을 대상으로 한다.
- WHERE 이름 = '홍길동' : '이름'이 '홍길동'인 사원만을 대상으로 한다.

3.

- UPDATE 고객 : (고객) 테이블을 갱신한다.
- SET 등급 = 'VIP' : '등급'을 'VIP'로 갱신한다.
- WHERE 구매금액 >= 1000000 : '구매금액'이 1,000,000 이상인 고객만을 대상으로 한다.

정답 1. ① INSERT INTO ② VALUES

2. DELETE FROM 사원 WHERE 이름 = '홍길동';

3. ① SET ② WHERE

**[핵심 111] DML – SELECT**

```
SELECT [PREDICATE] [테이블명].[속성명] [AS 별칭], [테이블명].[속성명, ...] [, 그룹함수(속성명) [AS 별칭]]
FROM 테이블명, 테이블명, ...
[WHERE 조건]
[GROUP BY 속성명, 속성명, ...]
[HAVING 조건]
[ORDER BY 속성명 [ASC | DESC]];
```

## • SELECT절

– PREDICATE : 불러올 튜플 수를 제한할 명령어를 기술함

▶ ALL : 모든 튜플을 검색할 때 지정하는 것으로, 주로 생략함

▶ DISTINCT : 중복된 튜플이 있으면 그 중 첫 번째만 검색함

▶ DISTINCTROW : 중복된 튜플을 검색하지만 선택된 속성의 값이 아닌, 튜플 전체를 대상으로 함

– 속성명 : 검색하여 불러올 속성(열) 및 수식들을 지정함

▶ 기본 테이블을 구성하는 모든 속성을 지정할 때는 '\*'를 기술함

▶ 두 개 이상의 테이블을 대상으로 검색할 때는 '테이블명.속성명'으로 표현함

– AS : 속성 및 연산의 이름을 다른 제목으로 표시하기 위해 사용함

• FROM절 : 질의에 의해 검색될 데이터들을 포함하는 테이블명을 기술함

• WHERE절 : 검색할 조건 기술함

• GROUP BY절 : 특정 속성을 기준으로 그룹화하여 검색할 때 그룹화 할 속성을 지정함

– 일반적으로 GROUP BY절은 그룹 함수와 함께 사용됨

– 그룹 함수의 종류

COUNT(속성명)	그룹별 튜플 수를 구하는 함수
MAX(속성명)	그룹별 최대값을 구하는 함수
MIN(속성명)	그룹별 최소값을 구하는 함수
SUM(속성명)	그룹별 합계를 구하는 함수
AVG(속성명)	그룹별 평균을 구하는 함수

• HAVING절 : GROUP BY와 함께 사용되며, 그룹에 대한 조건을 지정함

• ORDER BY절 : 특정 속성을 기준으로 정렬하여 검색할 때 사용함

– 속성명 : 정렬의 기준이 되는 속성명을 기술함

– [ASC | DESC] : 정렬 방식으로서 'ASC'는 오름차순, 'DESC'는 내림차순임. 생략하면 오름차순으로 지정됨

**참고만요 ! SELECT문의 실행 작동 순서**

FROM → WHERE → GROUP BY → HAVING → SELECT → DISTINCT → ORDER BY

2020년 1회 기사 실기, 2020년 1, 2회 기사 필기

1. 학생(STUDENT) 테이블에 전기과 학생이 50명, 전산과 학생이 100명, 전자과 학생이 50명 있다고 할 때, 다음 SQL문 ①, ②, ③의 실행 결과로 표시되는 튜플의 수를 쓰시오. (단, DEPT 필드는 학과를 의미한다)

- ```
① SELECT DEPT FROM STUDENT;
② SELECT DISTINCT DEPT FROM STUDENT;
③ SELECT COUNT(DISTINCT DEPT) FROM STUDENT
WHERE DEPT = '전산과';
```

**답**

- ① :  
② :  
③ :

2020년 2회 기능사 실기

2. 다음은 SELECT문의 실행 작동 순서를 나열한 것이다. 괄호 (① ~ ③)에 알맞은 명령을 보기에서 찾아 쓰시오.

FROM → ( ① ) → ( ② ) → ( ③ ) → SELECT  
→ ORDER BY

**<보기>**

GROUP BY, HAVING, WHERE

**답**

- ① :  
② :  
③ :

**해설 1.**

①

- SELECT DEPT : 'DEPT'를 표시한다.
- FROM STUDENT; : (STUDENT) 테이블을 대상으로 검색한다.

※ (STUDENT) 테이블에서 'DEPT'를 검색합니다. 총 200개의 튜플이 들어 있고 검색 조건이 없으므로 튜플의 수는 200입니다.

②

- SELECT DISTINCT DEPT : 'DEPT'를 표시하되, 같은 'DEPT' 속성의 값은 한 번만 표시한다.
- FROM STUDENT; : (STUDENT) 테이블을 대상으로 검색한다.

※ (STUDENT) 테이블에서 'DEPT'를 검색하는 데 중복된 결과는 처음의 한 개만 검색에 포함시킵니다. 전기과 50개 튜플의 'DEPT' 속성의 값이 같으므로 1개, 전산과 100개 튜플의 'DEPT' 속성의 값이 같으므로 1개, 전자과 50개 튜플의 'DEPT' 속성의 값이 같으므로 1개를 검색에 포함시키므로 3개의 튜플이 검색됩니다.

③

- SELECT COUNT(DISTINCT DEPT) : 'DEPT'의 개수를 표시하되, 같은 'DEPT' 속성의 값은 한 번만 계산한다.
- FROM STUDENT; : (STUDENT) 테이블을 대상으로 검색한다.
- WHERE DEPT = '전산과'; : 'DEPT'가 "전산과"인 자료만을 대상으로 검색한다.

※ (STUDENT) 테이블에서 'DEPT' 속성의 값이 '전산과'인 튜플에 대해 중복을 제거하고 개수를 세므로 1이 검색 결과로 표시됩니다.

**정답** 1. ① 200 ② 3 ③ 1

2. ① WHERE ② GROUP BY ③ HAVING

**[핵심112] 하위 질의**

조건절에 주어진 질의를 먼저 수행하여 그 검색 결과를 조건절의 피연산자로 사용한다.

**<사원>**

| 이름  | 부서  | 생일       | 주소  | 기본급 |
|-----|-----|----------|-----|-----|
| 홍길동 | 기획  | 04/05/61 | 망원동 | 120 |
| 임꺽정 | 인터넷 | 01/09/69 | 서교동 | 80  |
| 황진이 | 편집  | 07/21/75 | 합정동 | 100 |
| 김선달 | 편집  | 10/22/73 | 망원동 | 90  |
| 성춘향 | 기획  | 02/20/64 | 대흥동 | 100 |
| 장길산 | 편집  | 03/11/67 | 상암동 | 120 |
| 일지매 | 기획  | 04/29/78 | 연남동 | 110 |
| 강건달 | 인터넷 | 12/11/80 |     | 90  |

**<여가활동>**

| 이름  | 취미    | 경력 |
|-----|-------|----|
| 김선달 | 당구    | 10 |
| 성춘향 | 나이트댄스 | 5  |
| 일지매 | 태권    | 15 |
| 임꺽정 | 씨름    | 8  |

**예제** '취미'가 "나이트댄스"인 사원의 '이름'과 '주소'를 검색하시오.

SELECT 이름, 주소

FROM 사원

WHERE 이름 = (SELECT 이름 FROM 여가활동 WHERE 취미 = '나이트댄스');

- 먼저 "SELECT 이름 FROM 여가활동 WHERE 취미 = '나이트댄스'"를 수행하여 <여가활동> 테이블에서 '성춘향'을 찾는다. 그런 다음 하위 질의에 해당하는 피연산자의 자리에 '성춘향'을 대입하면 질의문은 "SELECT 이름, 주소 FROM 사원 WHERE 이름 = '성춘향'"과 같다.
- WHERE 절의 조건에 IN 연산자를 이용하여 다음과 같이 조건을 지정해도 된다.
- WHERE 이름 IN (SELECT 이름 FROM 여가활동 WHERE 취미 = '나이트댄스');

**<결과>**

| 이름  | 주소  |
|-----|-----|
| 성춘향 | 대흥동 |

2020년 2회 기능사 실기, 2020년 1.2회 기사 필기

1. 다음 SQL문의 실행 결과를 쓰시오.

SELECT 가격 FROM 도서가격

WHERE 책번호 = (SELECT 책번호 FROM 도서 WHERE 책명 = '데이터베이스');

## 〈도서〉

| 책번호  | 책명     |
|------|--------|
| 1111 | 데이터베이스 |
| 2222 | 운영체제   |
| 3333 | 자료구조   |

## 〈도서가격〉

| 책번호  | 가격    |
|------|-------|
| 1111 | 18000 |
| 2222 | 20000 |
| 3333 | 10000 |
| 4444 | 15000 |

답 :

## 해설 1.

- ② SELECT 가격 FROM 도서가격 WHERE 책번호 =  
 ① (SELECT 책번호 FROM 도서 WHERE 책명 = '데이터베이스');

①

- SELECT 책번호 : '책번호'를 표시한다.
  - FROM 도서 : 〈도서〉 테이블을 대상으로 검색한다.
  - WHERE 책명 = '데이터베이스' : '책명'이 "데이터베이스"인 자료만을 대상으로 검색한다.
- ※ '책명'이 "데이터베이스"인 도서의 책번호 "1111"이 검색되므로 ②의 질의 문은 SELECT 가격 FROM 도서가격 WHERE 책번호 = '1111'이 됩니다.

②

- SELECT 가격 : '가격'을 표시한다.
  - FROM 도서가격 : 〈도서가격〉 테이블을 대상으로 검색한다.
  - WHERE 책번호 = '1111' : '책번호'가 "1111"인 자료만을 대상으로 검색한다.
- ※ '책번호'가 "1111"인 도서의 가격은 18000입니다.

정답 1. 18000

2020 시나공 정보처리기사 필기 수록 내용

## [핵심113] DML - JOIN

## JOIN의 개념

- 2개의 테이블에 대해 연관된 튜플들을 결합하여, 하나의 새로운 릴레이션을 반환한다.
- JOIN은 크게 INNER JOIN과 OUTER JOIN으로 구분된다.
- JOIN은 일반적으로 FROM절에 기술하지만, 릴레이션이 사용되는 어느 곳에서나 사용할 수 있다.

## INNER JOIN

- 가장 일반적인 JOIN의 형태로, 관계가 설정된 두 테이블에서 조인된 필드가 일치하는 행만을 표시한다.
- WHERE절을 이용한 표기 형식

```
SELECT [테이블명1.]속성명, [테이블명2.]속성명, ...
FROM 테이블명1, 테이블명2, ...
WHERE 테이블명1.속성명 = 테이블명2.속성명;
```

- NATURAL JOIN을 이용한 표기 형식

```
SELECT [테이블명1.]속성명, [테이블명2.]속성명, ...
FROM 테이블명1 NATURAL JOIN 테이블명2;
```

- JOIN ~ USING절을 이용한 표기 형식

```
SELECT [테이블명1.]속성명, [테이블명2.]속성명, ...
FROM 테이블명1 JOIN 테이블명2 USING(속성명);
```

## 〈학생〉

| 학번 | 이름  | 학과코드 | 선배 | 성적 |
|----|-----|------|----|----|
| 15 | 고길동 | com  |    | 83 |
| 16 | 이순신 | han  |    | 96 |
| 17 | 김선달 | com  | 15 | 95 |
| 19 | 아무개 | han  | 16 | 75 |
| 37 | 박치민 |      | 17 | 55 |

## 〈학과〉

| 학과코드 | 학과명 |
|------|-----|
| com  | 컴퓨터 |
| han  | 국어  |
| eng  | 영어  |

**예제** 〈학생〉 테이블과 〈학과〉 테이블에서 학과코드 값이 같은 튜플을 JOIN하여 학번, 이름, 학과코드, 학과명을 출력하는 SQL문을 작성하시오.

- SELECT 학번, 이름, 학생.학과코드, 학과명  
FROM 학생, 학과  
WHERE 학생.학과코드 = 학과.학과코드;
- SELECT 학번, 이름, 학생.학과코드, 학과명  
FROM 학생 NATURAL JOIN 학과;
- SELECT 학번, 이름, 학생.학과코드, 학과명  
FROM 학생 JOIN 학과 USING(학과코드);

## 〈결과〉

| 학번 | 이름  | 학과코드 | 학과명 |
|----|-----|------|-----|
| 15 | 고길동 | com  | 컴퓨터 |
| 16 | 이순신 | han  | 국어  |
| 17 | 김선달 | com  | 컴퓨터 |
| 19 | 아무개 | han  | 국어  |

1. 다음은 〈학생〉 테이블과 〈학과〉 테이블에서 '학과코드' 속성의 값이 같은 자료 중 '이름'과 '학과명'을 검색하는 SQL문이다. 괄호(①~③)에 알맞은 명령어나 연산자를 넣어 SQL문을 완성하시오.

### 〈학생〉

| 학번 | 이름  | 학과코드 | 성적 |
|----|-----|------|----|
| 15 | 고길동 | han  | 73 |
| 16 | 이순신 | com  | 96 |
| 37 | 전산오 |      | 83 |

### 〈학과〉

| 학과코드 | 학과명 |
|------|-----|
| com  | 컴퓨터 |
| han  | 국문  |
| eng  | 영어  |

### 〈SQL문〉

|     |                                                                     |
|-----|---------------------------------------------------------------------|
| 방법1 | SELECT 이름, 학과명 AS 학과<br>FROM 학생, 학과<br>WHERE 학생.학과코드 ( ① ) 학과.학과코드; |
| 방법2 | SELECT 이름, 학과명 AS 학과<br>FROM 학생 ( ② ) 학과;                           |
| 방법3 | SELECT 이름, 학과명 AS 학과<br>FROM 학생 JOIN 학과 ( ③ )(학과코드);                |

### 답

- ① :  
② :  
③ :

#### 해설 1.

##### • 방법1

- SELECT 이름, 학과명 AS 학과 : '이름'과 '학과명'을 표시하되, '학과명'은 '학과'로 표시한다.
- FROM 학생, 학과 : 〈학생〉 테이블과 〈학과〉 테이블을 대상으로 한다.
- WHERE 학생.학과코드 = 학과.학과코드; : 〈학생〉 테이블의 '학과코드'와 〈학과〉 테이블의 '학과코드'가 같은 튜플을 JOIN한다.

##### • 방법2

- SELECT 이름, 학과명 AS 학과 : '이름'과 '학과명'을 표시하되, '학과명'은 '학과'로 표시한다.

- FROM 학생 NATURAL JOIN 학과; : 〈학생〉 테이블과 〈학과〉 테이블에서 같은 속성을 기준으로 JOIN한다.

##### • 방법3

- SELECT 이름, 학과명 AS 학과 : '이름'과 '학과명'을 표시하되, '학과명'은 '학과'로 표시한다.
- FROM 학생 JOIN 학과 USING(학과코드); : 〈학생〉 테이블과 〈학과〉 테이블에서 '학과코드'를 기준으로 JOIN한다.

정답 1. ① = ② NATURAL JOIN ③ USING

2020 시나공 정보처리기사 필기 수록 내용

#### [핵심114] DML - OUTER JOIN

- 릴레이션에서 JOIN 조건에 만족하지 않는 튜플도 결과로 출력하기 위한 JOIN 방법으로, LEFT OUTER JOIN, RIGHT OUTER JOIN 등이 있다.
- LEFT OUTER JOIN : INNER JOIN의 결과를 구한 후, 우측 항 릴레이션의 어떤 튜플과도 맞지 않는 좌측 항의 릴레이션에 있는 튜플들에 NULL 값을 붙여서 INNER JOIN의 결과에 추가함
- 표기 형식

```
SELECT [테이블명1].속성명, [테이블명2].속성명, ...
FROM 테이블명1 LEFT OUTER JOIN 테이블명2
ON 테이블명1.속성명 = 테이블명2.속성명;
```

- RIGHT OUTER JOIN : INNER JOIN의 결과를 구한 후, 좌측 항 릴레이션의 어떤 튜플과도 맞지 않는 우측 항의 릴레이션에 있는 튜플들에 NULL 값을 붙여서 INNER JOIN의 결과에 추가함
- 표기 형식

```
SELECT [테이블명1].속성명, [테이블명2].속성명, ...
FROM 테이블명1 RIGHT OUTER JOIN 테이블명2
ON 테이블명1.속성명 = 테이블명2.속성명;
```

## 〈학생〉

| 학번 | 이름  | 학과코드 | 선배 | 성적 |
|----|-----|------|----|----|
| 15 | 고길동 | com  |    | 83 |
| 16 | 이순신 | han  |    | 96 |
| 17 | 김선달 | com  | 15 | 95 |
| 19 | 아무개 | han  | 16 | 75 |
| 37 | 박치민 |      | 17 | 55 |

## 〈학과〉

| 학과코드 | 학과명 |
|------|-----|
| com  | 컴퓨터 |
| han  | 국어  |
| eng  | 영어  |

**예제** 〈학생〉 테이블과 〈학과〉 테이블에서 학과코드 값이 같은 튜플을 JOIN하여 학번, 이름, 학과코드, 학과명을 출력하는 SQL문을 작성하시오. 이때, 학과코드가 입력되지 않은 학생도 출력하시오.

```
SELECT 학번, 이름, 학생.학과코드, 학과명
FROM 학생 LEFT OUTER JOIN 학과
ON 학생.학과코드 = 학과.학과코드;
```

**해설** INNER JOIN을 하면 학과코드가 입력되지 않은 박치민은 출력되지 않는다. 그러므로 JOIN 구문을 기준으로 왼쪽 테이블, 즉 〈학생〉의 자료는 모두 출력되는 LEFT JOIN을 사용한 것이다. 다음과 같이 JOIN 구문을 기준으로 테이블의 위치를 교환하여 RIGHT JOIN을 사용해도 결과는 같다.

## 〈결과〉

| 학번 | 이름  | 학과코드 | 학과명 |
|----|-----|------|-----|
| 15 | 고길동 | com  | 컴퓨터 |
| 16 | 이순신 | han  | 국어  |
| 17 | 김선달 | com  | 컴퓨터 |
| 19 | 아무개 | han  | 국어  |
| 37 | 박치민 |      |     |

2020년 2회 기능사 실기

1. 다음은 〈topic〉 테이블과 〈auth〉 테이블을 결합하여 검색하는 SQL문이다. 괄호에 알맞은 명령을 쓰시오.

## 〈topic〉

| id | title1     | title2        | auth_id |
|----|------------|---------------|---------|
| 1  | sql        | sql is        | 1       |
| 2  | oracle     | oracle is     | 1       |
| 3  | sql-server | sql-server is | 3       |
| 4  | progress   | progress is   | 2       |
| 5  | mongoDB    | mongoDB is    | 1       |

## 〈auth〉

| id | name | job                    |
|----|------|------------------------|
| 1  | kim  | develope               |
| 2  | shin | database administrator |
| 3  | park | scientist, develope    |

## 〈SQL문〉

```
SELECT * FROM topic LEFT ( ) auth ON topic.auth_
id = auth.id;
```

## 〈결과〉

| id | title1     | title2        | auth_id | id | name | job                    |
|----|------------|---------------|---------|----|------|------------------------|
| 1  | sql        | sql is        | 1       | 1  | kim  | develope               |
| 2  | oracle     | oracle is     | 1       | 1  | kim  | develope               |
| 3  | sql-server | sql-server is | 3       | 3  | park | scientist, develope    |
| 4  | progress   | progress is   | 2       | 2  | shin | database administrator |
| 5  | mongoDB    | mongoDB is    | 1       | 1  | kim  | develope               |

## 답 :

## 해설 1.

- SELECT \* : 모든 속성을 표시한다.
- FROM topic LEFT OUTER JOIN auth : 〈topic〉 테이블의 튜플들은 〈auth〉 테이블의 연관성과 관계없이 모두 표시하는 LEFT OUTER JOIN 이다.
- ON topic.auth\_id = auth.id; : 〈topic〉 테이블의 'auth\_id'와 〈auth〉 테이블의 'id'로 JOIN한다.

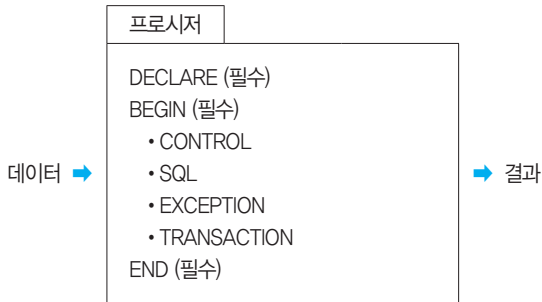
정답 1. OUTER JOIN

## [핵심 115] 프로시저(Procedure)

- 절차형 SQL을 활용하여 특정 기능을 수행하는 일종의 트랜잭션 언어로, 호출을 통해 실행되어 미리 저장해 놓은 SQL 작업을 수행한다.
- 프로시저는 데이터베이스에 저장되어 수행되기 때문에 스토어드(Stored) 프로시저라고도 불린다.
- 프로시저는 시스템의 일일 마감 작업, 일괄(Batch) 작업 등에 주로 사용된다.



## 프로시저 구성도



## 프로시저 생성

```

CREATE [OR REPLACE] PROCEDURE 프로시저명(파라미터)
[지역변수 선언]
BEGIN
    프로시저 BODY;
END;
  
```

## 프로시저 실행

```

EXECUTE 프로시저명;
EXEC 프로시저명;
CALL 프로시저명;
  
```

## 프로시저 제거

```
DROP PROCEDURE 프로시저명;
```

1. DBMS에서 시스템의 주간 마감, 일일 마감 작업 등 주로 일괄 작업에 사용되며, 데이터베이스에 저장되어 수행된다. DECLARE, CONTROL, SQL, EXCEPTION 등의 구성 요소로 이루어져 있고, EXECUTE 또는 CALL 명령어로 실행되는 절차형 SQL은 무엇인지 쓰시오.

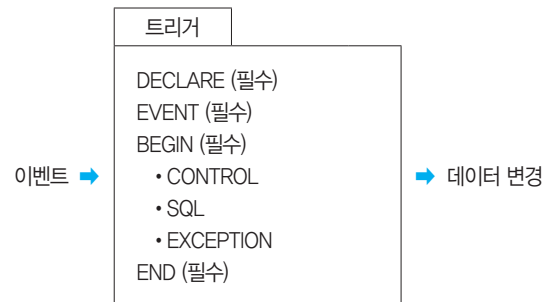
답 :

**정답** 1. 프로시저(Procedure) 또는 스토어드(Stored) 프로시저

## [핵심116] SQL – 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.

## 트리거 구성도



## 트리거의 생성

```

CREATE [OR REPLACE] TRIGGER 트리거명 [동작시기 옵션]
[동작 옵션] ON 테이블명
REFERENCING [NEW | OLD] AS 테이블명
FOR EACH ROW
[WHEN 조건식]
BEGIN
    트리거 BODY;
END;
  
```

## 트리거의 제거

```
DROP TRIGGER 트리거명;
```

1. 데이터베이스 시스템에서 삽입, 갱신, 삭제 등의 이벤트가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL을 쓰시오.

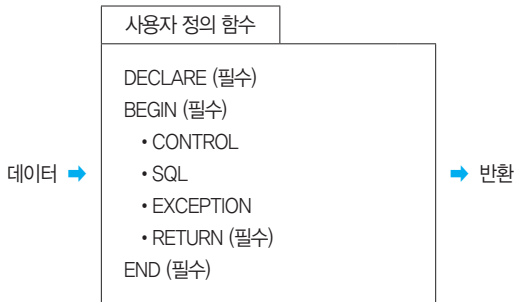
답 :

**정답** 1. 트리거(Trigger)

### [핵심117] SQL – 사용자 정의 함수

- 프로시저와 유사하게 SQL을 사용하여 일련의 작업을 연속적으로 처리하며, 종료 시 처리 결과를 단일 값으로 반환하는 절차형 SQL이다.
- 사용자 정의 함수는 데이터베이스에 저장되어 SELECT, INSERT, DELETE, UPDATE 등 DML문의 호출에 의해 실행된다.
- 사용자 정의 함수는 예약어 RETURN을 통해 값을 반환하기 때문에 출력 파라미터가 없다.

#### 사용자 정의 함수의 구성도



#### 사용자 정의 함수 생성

```

CREATE [OR REPLACE] FUNCTION 사용자 정의 함수명(파라미터)
[지역변수 선언]
BEGIN
    사용자 정의 함수 BODY;
    RETURN 반환값;
END;
  
```

#### 사용자 정의 함수 실행

```

SELECT 사용자 정의 함수명 FROM 테이블명;
INSERT INTO 테이블명(속성명) VALUES (사용자 정의 함수명);
DELETE FROM 테이블명 WHERE 속성명 = 사용자 정의 함수명;
UPDATE 테이블명 SET 속성명 = 사용자 정의 함수명;
  
```

#### 사용자 정의 함수 제거

```

DROP FUNCTION 사용자 정의 함수명;
  
```

#### 1. 다음 설명에 가장 부합하는 절차형 SQL의 종류를 쓰시오.

- 일련의 작업을 연속적으로 처리하며, 종료 시 처리 결과를 단일값으로 반환한다.
- 데이터베이스에 저장되어 DML문의 호출에 의해 실행된다.
- RETURN을 통해 값을 반환하기 때문에 출력 파라미터가 존재하지 않는다.

답 :

정답 1. 사용자 정의 함수

### [핵심118] SQL – 커서(Cursor)

#### 커서(Cursor)

- 쿼리문의 처리 결과가 저장되어 있는 메모리 공간을 가리키는 포인터(Pointer)이다.
- 커서는 내부에서 자동으로 생성되어 사용되는 묵시적 커서와, 사용자가 직접 정의해서 사용하는 명시적 커서가 있다.
- 커서의 수행은 열기(Open), 패치(Fetch), 닫기(Close)의 세 단계로 진행된다.

#### 묵시적 커서(Implicit Cursor)

- DBMS 자체적으로 열리고(Open) 패치(Fetch)되어 사용이 끝나면 닫히지만(Close) 커서의 속성을 조회하여 사용된 쿼리 정보를 열람하는 것이 가능하다.
- 커서의 속성

|              |                                                                                                                                |
|--------------|--------------------------------------------------------------------------------------------------------------------------------|
| SQL%FOUND    | 쿼리 수행의 결과로 패치(Fetch)된 튜플 수가 1개 이상이면 TRUE                                                                                       |
| SQL%NOTFOUND | 쿼리 수행의 결과로 패치(Fetch)된 튜플 수가 0개이면 TRUE                                                                                          |
| SQL%ROWCOUNT | 쿼리 수행의 결과로 패치(Fetch)된 튜플 수를 반환                                                                                                 |
| SQL%ISOPEN   | <ul style="list-style-type: none"> <li>• 커서가 열린(Open) 상태이면 TRUE</li> <li>• 묵시적 커서는 자동으로 생성된 후 자동으로 닫히기 때문에 항상 FALSE</li> </ul> |

## 명시적 커서(Explicit Cursor)

- 사용자가 직접 정의해서 사용하는 커서로, 주로 절차형 SQL에서 SELECT문의 결과로 반환되는 여러 튜플들을 제어하기 위해 사용된다.
- 커서는 기본적으로 '열기(Open) - 패치(Fetch) - 닫기(Close)' 순으로 이루어지며, 명시적 커서로 사용하기 위해서는 열기 전에 선언(Declare)을 해야 한다.

### 1. 다음 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

- ( )는 쿼리문의 처리 결과가 저장되어 있는 메모리 공간을 가리키는 포인터(Pointer)이다.
- ( )는 묵시적 ( )와 명시적 ( )가 있다.
- ( )의 수행은 열기(Open), 패치(Fetch), 닫기(Close)의 세 단계로 진행된다.

답 :

2. 절차형 SQL에서 튜플들에 순차적으로 접근하는데 사용되는 커서(Cursor) 중 명시적 커서와 묵시적 커서에 대해 간략히 서술하시오.

답

- ① 명시적 커서 :
- ② 묵시적 커서 :

### 정답 1. 커서(Cursor)

2. ① 사용자가 직접 정의해서 사용하는 커서이다. ② 내부에서 자동으로 생성하여 사용하는 커서이다.



## 9장 | 소프트웨어 개발 보안 구축

2020 시나공 정보처리기사 필기 수록 내용

### [핵심119] 소프트웨어 개발의 비용 산정 기법

- 소프트웨어 비용 산정은 소프트웨어의 개발 규모를 소요되는 인원, 자원, 기간 등으로 확인하여 실행 가능한 계획을 수립하기 위해 필요한 비용을 산정하는 것이다.

#### • 하향식 비용 산정 기법

– 과거의 유사한 경험을 바탕으로 전문 지식이 많은 개발자들이 참여한 회의를 통해 비용을 산정하는 비과학적인 방법이다.

– 종류

|           |                                                      |
|-----------|------------------------------------------------------|
| 전문가 감정 기법 | 조직 내에 있는 경험이 많은 두 명 이상의 전문가에게 비용 산정을 의뢰하는 기법         |
| 델파이 기법    | 전문가 감정 기법의 주관적인 편견을 보완하기 위해 많은 전문가의 의견을 종합하여 산정하는 기법 |

#### • 상향식 비용 산정 기법

– 프로젝트의 세부적인 작업 단위별로 비용을 산정한 후 집계하여 전체 비용을 산정하는 방법이다.

– 종류

|                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LOC(원시 코드 라인 수, source Line Of Code) 기법 | <ul style="list-style-type: none"> <li>• 소프트웨어 각 기능의 원시 코드 라인 수의 비관치, 낙관치, 기대치를 측정하여 예측치를 구하고 이를 이용하여 비용을 산정하는 기법이다</li> <li>• 산정 공식 : <ul style="list-style-type: none"> <li>– 노력(인월) = 개발 기간 × 투입 인원</li> <li>= LOC / 1인당 월평균 생산 코드 라인 수</li> <li>– 개발 비용 = 노력(인월) × 단위 비용 (1인당 월평균 인건비)</li> <li>– 개발 기간 = 노력(인월) / 투입 인원</li> <li>– 생산성 = LOC / 노력(인월)</li> </ul> </li> <li>예 LOC 기법에 의하여 예측된 총 라인 수가 40,000라인, 개발에 참여할 프로그래머가 10명, 프로그래머들의 평균 생산성이 월간 400라인일 때 개발에 소요되는 기간은?</li> <li>• 노력(인월) = LOC / 1인당 월평균 생산 코드 라인 수 = 40000 / 400 = 100명</li> <li>• 개발 기간 = 노력(인월) / 투입 인원 = 100 / 10 = 10개월</li> </ul> |
|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                         |                                                                |
|-----------------------------------------|----------------------------------------------------------------|
| 개발 단계별<br>인월수(Effort<br>Per Task)<br>기법 | LOC 기법을 보완하기 위한 기법으로, 각 기능을 구현시키는 데 필요한 노력을 생명 주기의 각 단계별로 산정한다 |
|-----------------------------------------|----------------------------------------------------------------|

## 2020년 1회 기사 실기

1. LOC 기법에 의하여 예측된 총 라인 수가 30,000라인, 개발에 참여할 프로그래머가 5명, 프로그래머들의 평균 생산성이 월간 300라인일 때 개발에 소요되는 기간을 쓰시오.

답

- 계산 공식 :
- 결과 :

2. 어떤 소프트웨어 개발을 위해 5명의 개발자가 5개월 동안 참여되었다. 그런데 그 중 3명은 5개월 동안 계속 참여했지만 2명은 2개월 동안만 부분적으로 참여했다. 이 소프트웨어 개발을 위한 인월(Man Month)은 얼마인지 쓰시오.

답 :

**해설** 2. 5명이 5개월 동안 개발한 것의 노력(인월)은 25이고, 2명이 2개월 동안 개발한 것의 노력은 4이므로  $25 + 4 = 29$ 입니다.

**정답** 1. 계산 공식 :  $(30,000 / 300) / 5 = 20$ , 결과 : 20개월 2. 29

## 2020 사나공 정보처리기사 필기 수록 내용

**[핵심120]** 서비스 거부 공격

- 표적이 되는 서버의 자원을 고갈시킬 목적으로 다수의 공격자 또는 시스템에서 대량의 데이터를 한 곳의 서버에 집중적으로 전송함으로써, 표적이 되는 서버의 정상적인 기능을 방해하는 것이다.
- 서비스 거부 공격의 유형

|                          |                                                                                             |
|--------------------------|---------------------------------------------------------------------------------------------|
| Ping of Death<br>(죽음의 핑) | Ping 명령을 전송할 때 패킷의 크기를 인터넷 프로토콜 허용 범위(65,536 바이트) 이상으로 전송하여 공격 대상의 네트워크를 마비시키는 서비스 거부 공격 방법 |
| Smurfing(스머핑)            | IP나 ICMP의 특성을 악용하여 엄청난 양의 데이터를 한 사이트에 집중적으로 보냄으로써 네트워크 또는 시스템의 상태를 불능으로 만드는 공격 방법           |

|                                                   |                                                                                                                                                                                                                 |
|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SYN Flooding                                      | TCP(Transmission Control Protocol)는 신뢰성 있는 전송을 위해 3-way-handshake를 거친 후에 데이터를 전송하게 되는데, SYN Flooding은 공격자가 가상의 클라이언트로 위장하여 3-way-handshake 과정을 의도적으로 중단시킴으로써 공격 대상 지인 서버가 대기 상태에 놓여 정상적인 서비스를 수행하지 못하게 하는 공격 방법 |
| TearDrop                                          | 데이터의 송·수신 과정에서 패킷의 크기가 커 여러 개로 분할되어 전송될 때 분할 순서를 알 수 있도록 Fragment Offset 값을 함께 전송하는데, TearDrop은 이 Offset 값을 변경시켜 수신 측에서 패킷을 재조립할 때 오류로 인한 과부하를 발생시킴으로써 시스템이 다운되도록 하는 공격 방법                                      |
| Land                                              | 패킷을 전송할 때 송신 IP 주소와 수신 IP 주소를 모두 공격 대상의 IP 주소로 하여 공격 대상에게 전송하는 것으로, 이 패킷을 받은 공격 대상은 송신 IP 주소가 자신이므로 자신에게 응답을 수행하게 되는데, 이러한 패킷이 계속해서 전송될 경우 자신에 대해 무한히 응답하게 하는 공격                                                |
| DDoS(Distributed Denial of Service, 분산 서비스 거부) 공격 | 여러 곳에 분산된 공격 지점에서 한 곳의 서버에 대해 분산 서비스 공격을 수행하는 것으로, 네트워크에서 취약점이 있는 호스트들을 탐색한 후 이들 호스트들에 분산 서비스 공격용 툴을 설치하여 에이전트(Agent)로 만든 후 DDoS 공격에 이용함                                                                        |

## 2020년 1회 기사 실기

1. 다음은 네트워크 공격에 대한 패킷 로그를 표현한 것이다. 아래의 패킷 로그와 같이 공격자가 패킷의 출발지 주소(Address) 또는 포트(Port)를 임의로 변경하여 송신측 IP 주소 또는 포트를 동일하게 함으로써 송신 IP 주소가 자신이므로 자신에게 응답을 수행하게 되는데, 이러한 패킷을 계속 전송하면 자신에 대해 무한히 응답하게 되므로 컴퓨터의 실행 속도를 느리게 하거나 동작을 마비시켜 서비스 거부 상태에 빠지도록 하는 네트워크 공격이 무엇인지 쓰시오.

```
source : 192.168.1.200
destination : 192.168.1.200
protocol : 6
src port : 21845
dst port : 21845
```

답 :

2020년 1, 2회 기사 필기

2. IP나 ICMP의 특성을 악용하여 특정 사이트에 집중적으로 데이터를 보내 네트워크 또는 시스템의 상태를 불능으로 만드는 공격 방법을 쓰시오.

답 :

정답 1. Land 2. Smurfing



2020 사나공 정보처리기사 필기 수록 내용

### [핵심121] 네트워크 침해 공격 관련 용어

|                                             |                                                                                                                             |
|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| 스미싱<br>(Smishing)                           | 각종 행사 안내, 경품 안내 등의 문자 메시지(SMS)를 이용해 사용자의 개인 신용 정보를 빼내는 수법                                                                   |
| 스피어 피싱<br>(Spear Phishing)                  | 사회 공학의 한 기법으로, 특정 대상을 선정한 후 그 대상에게 일반적인 이메일로 위장한 메일을 지속적으로 발송하여, 발송 메일의 본문 링크나 첨부된 파일을 클릭하도록 유도해 사용자의 개인 정보를 탈취 함           |
| APT(Advanced Persistent Threats, 지능형 지속 위협) | 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 공격                  |
| 무작위 대입 공격<br>(Brute Force Attack)           | 암호화된 문서의 암호키를 찾아내기 위해 적용 가능한 모든 값을 대입하여 공격하는 방식                                                                             |
| 큐싱(Qshing)                                  | QR코드(Quick Response Code)를 통해 악성 앱의 다운로드를 유도하거나 악성 프로그램을 설치하도록 하는 금융사기 기법의 하나로, QR코드와 개인정보 및 금융정보를 낚는다(Fishing)는 의미의 합성 신조어 |

SQL  
삽입(Injection)  
공격

전문 스캐너 프로그램 혹은 봇넷 등을 이용해 웹사이트를 무차별적으로 공격하는 과정에서 취약한 사이트가 발견되면 데이터베이스 등의 데이터를 조작하는 일련의 공격 방식

1. 다음은 네트워크 침해 공격과 관련된 용어 설명이다. 괄호(①~④)에 들어갈 가장 알맞은 용어를 쓰시오.

|       |                                                                                                                            |
|-------|----------------------------------------------------------------------------------------------------------------------------|
| ( ① ) | QR코드(Quick Response Code)를 통해 악성 앱의 다운로드를 유도하거나 악성 프로그램을 설치하도록 하는 금융사기 기법이다.                                               |
| ( ② ) | 각종 행사 안내, 경품 안내 등의 문자 메시지(SMS)를 이용해 사용자의 개인 신용 정보를 빼내는 수법이다.                                                               |
| ( ③ ) | 전문 스캐너 프로그램 혹은 봇넷 등을 이용해 웹사이트를 무차별적으로 공격하는 과정에서 취약한 사이트가 발견되면 데이터베이스 등의 데이터를 조작하는 일련의 공격 방식이다.                             |
| ( ④ ) | 사회 공학의 한 기법으로, 특정 대상을 선정한 후 그 대상에게 일반적인 이메일로 위장한 메일을 지속적으로 발송하여, 발송 메일의 본문 링크나 첨부된 파일을 클릭하도록 유도해 사용자의 개인 정보를 탈취하는 공격 방식이다. |

답

- ① :  
② :  
③ :  
④ :

정답 1. ① 큐싱(Qshing) ② 스미싱(Smishing)

③ SQL 삽입(Injection) 공격 ④ 스피어 피싱(Spear Phishing)



2020 시나공 정보처리기사 필기 수록 내용

**[핵심 122] 정보 보안 침해 공격 관련 용어**

|                               |                                                                                                                                                                                                        |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 좀비(Zombie)<br>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 말함                                                                                                                                             |
| 봇넷(Botnet)                    | 악성 프로그램에 감염되어 악의적인 의도로 사용될 수 있는 다수의 컴퓨터들이 네트워크로 연결된 형태를 말함                                                                                                                                             |
| 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<br>(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<br>(Key Logger Attack) | 컴퓨터 사용자의 키보드 움직임을 탐지해 ID, 패스워드, 계좌번호, 카드번호 등과 같은 개인의 중요한 정보를 몰래 빼가는 해킹 공격                                                                                                                              |
| 랜섬웨어<br>(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<br>(Back Door, Trap Door) | <ul style="list-style-type: none"> <li>시스템 설계자가 서비스 기술자나 유지 보수 프로그램 작성자(Programmer)의 액세스 편의를 위해 시스템 보안을 제거하여 만들어놓은 비밀 통로로, 컴퓨터 범죄에 악용되기도 함</li> <li>백도어 탐지 방법 : 무결성 검사, 로그 분석, SetUID 파일 검사</li> </ul> |
| 트로이 목마<br>(Trojan Horse)      | 정상적인 기능을 하는 프로그램으로 위장하여 프로그램 내에 숨어 있다가 해당 프로그램이 동작할 때 활성화되어 부작용을 일으키는 것으로, 자기 복제 능력은 없음                                                                                                                |

2020년 1, 2회 기사 필기

**1.** 컴퓨터 사용자의 키보드 움직임을 탐지해 ID, 패스워드 등 개인의 중요한 정보를 몰래 빼가는 해킹 공격을 무엇이라고 하는지 쓰시오.

답 :

2020년 1, 2회 기사 필기

**2.** 다음 설명의 정보 보안 침해 공격 관련 용어가 무엇인지 쓰시오.

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

답 :

2020년 1, 2회 기사 필기

**3.** 시스템 설계자가 서비스 기술자나 유지 보수 프로그램 작성자(Programmer)의 액세스 편의를 위해 시스템 보안을 제거하여 만들어놓은 비밀 통로를 백도어(Back Door)라고 한다. 백도어 탐지 방법 3가지를 쓰시오.

답 :

**정답** 1. 키로거 공격(Key Logger Attack) 2. 랜섬웨어(Ransomware)  
3. 무결성 검사, 로그 분석, SetUID 파일 검사

**[핵심 123] 소프트웨어 개발 보안**

- 소프트웨어 개발 과정에서 발생할 수 있는 보안 취약점을 최소화하여 보안 위협으로부터 안전한 소프트웨어를 개발하기 위한 일련의 보안 활동을 의미한다.
- 소프트웨어 개발 보안은 데이터의 기밀성(Confidentiality), 무결성(Integrity), 가용성(Availability)을 유지하는 것을 목표로 한다.
- 소프트웨어 개발 보안 관련 기관

|                |                                     |
|----------------|-------------------------------------|
| 행정안전부          | 소프트웨어 개발 보안 정책을 총괄한다.               |
| 한국인터넷진흥원(KISA) | 소프트웨어 개발 보안 정책 및 가이드를 개발한다.         |
| 발주기관           | 소프트웨어 개발 보안의 계획을 수립한다.              |
| 사업자            | 소프트웨어 개발 보안 관련 기술 수준 및 적용 계획을 명시한다. |
| 감리법인           | 감리 계획을 수립하고 협의한다.                   |

1. 다음은 소프트웨어 개발 보안 관련 기관에 대한 설명이다. 괄호(①~②)에 들어갈 알맞은 기관을 쓰시오.

|       |                                     |
|-------|-------------------------------------|
| 발주기관  | 소프트웨어 개발 보안의 계획을 수립한다.              |
| ( ① ) | 소프트웨어 개발 보안 정책 및 가이드를 개발한다.         |
| 사업자   | 소프트웨어 개발 보안 관련 기술 수준 및 적용 계획을 명시한다. |
| ( ② ) | 소프트웨어 개발 보안 정책을 총괄한다.               |
| 감리법인  | 감리 계획을 수립하고 협의한다.                   |

답

① :

② :

정답 1. ① 한국인터넷진흥원(KISA) ② 행정안전부

#### [핵심 124] 소프트웨어 개발 보안 활동 관련 법령

|                            |                                                       |
|----------------------------|-------------------------------------------------------|
| 개인정보 보호법                   | 개인정보의 처리 및 보호에 관한 사항을 정함으로써 개인의 자유와 권리를 보호한다.         |
| 정보통신망 이용촉진 및 정보보호 등에 관한 법률 | 정보통신망의 이용 촉진 및 정보통신 서비스를 이용하는 이용자들의 개인 정보를 보호한다.      |
| 신용정보의 이용 및 보호에 관한 법률       | 개인 신용정보의 효율적 이용과 체계적인 관리를 통해 정보의 오남용을 방지한다.           |
| 위치정보의 보호 및 이용 등에 관한 법률     | 개인 위치정보의 안전한 이용 환경을 조성하여 정보의 유출이나 오남용을 방지한다.          |
| 표준 개인정보 보호 지침              | 개인정보의 처리에 관한 기준, 개인정보 침해의 유형 및 예방조치 등에 관한 세부사항을 규정한다. |

|                    |                                                                                                               |
|--------------------|---------------------------------------------------------------------------------------------------------------|
| 개인정보의 안전성 확보 조치 기준 | 개인정보 처리자가 개인정보를 처리하는데 있어 개인정보가 분실, 도난, 유출, 위조, 변조, 훼손되지 않도록 안전성 확보에 필요한 기술적, 관리적, 물리적 안전조치에 관한 최소한의 기준을 규정한다. |
| 개인정보 영향평가에 관한 고시   | 개인정보 영향평가를 위한 평가기관의 지정, 영향평가의 절차 등에 관한 세부기준을 규정한다.                                                            |

1. 소프트웨어 개발 보안 활동 관련 법령 중 개인정보 보호법이 무엇인지 간략히 서술하시오.

답 :

정답 1. 개인정보 보호법은 개인정보의 처리 및 보호에 관한 사항을 정함으로써 개인의 자유와 권리를 보호하는 법령이다.

#### [핵심 125] Secure SDLC

- 보안상 안전한 소프트웨어를 개발하기 위해 SDLC에 보안 강화를 위한 프로세스를 포함한 것을 의미한다.
- SDLC(Software Development Life Cycle, 소프트웨어 개발 생명주기) : 소프트웨어 개발 방법론의 바탕이 되는 것으로, 소프트웨어를 개발하기 위해 정의하고 운용, 유지보수 등의 전 과정을 각 단계별로 나눈 것
- Secure SDLC는 요구사항 분석, 설계, 구현, 테스트, 유지 보수 등 SDLC 전체 단계에 걸쳐 수행되어야 할 보안 활동을 제시한다.

| 단계         | 보안 활동                                                                                                                                          |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| 요구사항 분석 단계 | <ul style="list-style-type: none"> <li>• 보안 항목에 해당하는 요구사항을 식별하는 작업을 수행한다.</li> <li>• 전산화되는 정보가 가지고 있는 보안 수준을 보안 요소별로 등급을 구분하여 분류한다.</li> </ul> |
| 설계 단계      | 식별된 보안 요구사항들을 소프트웨어 설계서에 반영하고, 보안 설계서를 작성한다.                                                                                                   |



|          |                                                                                                                                                                                                              |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 구현 단계    | <ul style="list-style-type: none"> <li>표준 코딩 정의서 및 소프트웨어 개발 보안 가이드를 준수하며, 설계서에 따라 보안 요구 사항들을 구현한다.</li> <li>시큐어 코딩(Secure Coding) : 소프트웨어의 구현 단계에서 발생할 수 있는 보안 취약점들을 최소화하기 위해 보안 요소들을 고려하며 코딩하는 것</li> </ul> |
| 테스트 단계   | 설계 단계에서 작성한 보안 설계서를 바탕으로 보안 사항들이 정확히 반영되고 동작되는지 점검한다.                                                                                                                                                        |
| 유지 보수 단계 | 이전 과정을 모두 수행하였음에도 발생할 수 있는 보안 사고들을 식별하고, 사고 발생 시 이를 해결하고 보안 패치를 실시한다.                                                                                                                                        |

1. 다음에 제시된 내용은 Secure SDLC의 수행 과정 중 어느 단계에서의 보안 활동인지 쓰시오.

- 표준 코딩 정의서 및 소프트웨어 개발 보안 가이드를 준수하며, 설계서에 따라 보안 요구사항들을 구현한다.
- 개발 과정 중에는 지속적인 단위 테스트를 통해 소프트웨어에 발생할 수 있는 보안 취약점을 최소화해야 한다.

답 :

2. 다음은 무엇에 대한 설명인지 가장 적합한 용어를 쓰시오.

- 소프트웨어의 구현 단계에서 발생할 수 있는 보안 취약점들을 최소화하기 위해 보안 요소들을 고려하며 코딩하는 것을 의미한다.
- 보안 취약점을 사전에 대응하여 안정성과 신뢰성을 확보하기 위해 사용된다.

답 :

**정답** 1. 구현 단계 2. 시큐어 코딩(Secure Coding)

### [핵심126] 보안 요소

- 소프트웨어 개발에 있어 충족시켜야 할 요소 및 요건을 의미한다.
- 주요 보안 요소에는 기밀성(Confidentiality), 무결성(Integrity), 가용성(Availability)이 있으며, 그 외에도 인증(Authentication), 부인 방지(NonRepudiation) 등이 있다.

|       |                                                        |
|-------|--------------------------------------------------------|
| 기밀성   | 시스템 내의 정보와 자원은 인가된 사용자에게만 접근이 허용된다.                    |
| 무결성   | 시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있다.                       |
| 가용성   | 인가받은 사용자는 언제라도 사용할 수 있다.                               |
| 인증    | 시스템 내의 정보와 자원을 사용하려는 사용자가 합법적인 사용자인지를 확인하는 모든 행위를 말한다. |
| 부인 방지 | 데이터를 송·수신한 자가 송·수신 사실을 부인할 수 없도록 송·수신 증거를 제공한다.        |

2020년 1, 2회 기사 필기

1. 다음은 소프트웨어 개발에 있어 충족시켜야 할 보안 요소에 대한 설명이다. 괄호(①~③)에 들어갈 알맞은 보안 요소를 쓰시오.

|       |                                                        |
|-------|--------------------------------------------------------|
| ( ① ) | 인가받은 사용자는 언제라도 사용할 수 있다.                               |
| 인증    | 시스템 내의 정보와 자원을 사용하려는 사용자가 합법적인 사용자인지를 확인하는 모든 행위를 말한다. |
| ( ② ) | 시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있다.                       |
| 부인 방지 | 데이터를 송·수신한 자가 송·수신 사실을 부인할 수 없도록 송·수신 증거를 제공한다.        |
| ( ③ ) | 시스템 내의 정보와 자원은 인가된 사용자에게만 접근이 허용된다.                    |

답

- ① :  
② :  
③ :

**정답** 1. ① 가용성 ② 무결성 ③ 기밀성



### [핵심127] 보안 점검 - 세션 통제

- 세션은 서버와 클라이언트의 연결을 의미하고, 세션 통제는 세션의 연결과 연결로 인해 발생하는 정보를 관리하는 것을 의미한다.
- 세션 통제는 소프트웨어 개발 과정 중 요구사항 분석 및 설계 단계에서 진전해야 하는 보안 점검 내용이다.
- 세션 통제의 보안 약점

|                  |                                                                                                                                                                                                                             |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 불충분한 세션 관리       | <ul style="list-style-type: none"> <li>• 일정한 규칙이 존재하는 세션ID가 발급되거나 타임아웃이 너무 길게 설정되어 있는 경우 발생할 수 있는 보안 약점이다.</li> <li>• 세션ID(SessionID) : 서버가 클라이언트들을 구분하기 위해 부여하는 키(Key)로, 클라이언트가 서버에 요청을 보낼 때마다 세션ID를 통해 인증이 수행됨</li> </ul> |
| 잘못된 세션에 의한 정보 노출 | <ul style="list-style-type: none"> <li>• 다중 스레드(Multi-Thread) 환경에서 멤버 변수에 정보를 저장할 때 발생하는 보안 약점이다.</li> <li>• 멤버 변수보다 지역 변수를 활용하여 변수의 범위를 제한함으로써 방지할 수 있다.</li> </ul>                                                        |

1. 다음은 세션 통제의 보안 약점에 대한 설명이다. 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

다중 스레드(Multi-Thread) 환경에서 ( )에 정보를 저장할 때 정보 노출을 방지하려면 ( )보다 지역 변수를 활용하여 변수의 범위를 제한해야 한다. ( )는 객체와 연결된 변수로, 클래스 내에 선언되어 클래스의 모든 메소드들이 접근 가능한 변수이다.

답 :

정답 1. 멤버 변수



### [핵심128] 보안 점검 - 입력 데이터 검증 및 표현

- 입력 데이터 검증 및 표현은 입력 데이터로 인해 발생하는 문제들을 예방하기 위해 구현 단계에서 검증해야 하는 보안 점검 항목들이다.
- 입력 데이터 검증 및 표현의 보안 약점

|                         |                                                                                                                                                                                                                           |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SQL 삽입                  | <ul style="list-style-type: none"> <li>• 입력란에 SQL을 삽입하여 무단으로 DB를 조회하거나 조작하는 보안 약점이다.</li> <li>• 동적 쿼리에 사용되는 입력 데이터에 예약어 및 특수문자가 입력되지 않게 필터링 되도록 설정하여 방지할 수 있다.</li> </ul>                                                 |
| 경로 조작 및 자원 삽입           | <ul style="list-style-type: none"> <li>• 데이터 입출력 경로를 조작하여 서버 자원을 수정 · 삭제할 수 있는 보안 약점이다.</li> <li>• 사용자 입력값을 식별자로 사용하는 경우, 경로 순회 공격을 막는 필터를 사용하여 방지할 수 있다.</li> </ul>                                                      |
| 크로스사이트 스크립팅(XSS)        | <ul style="list-style-type: none"> <li>• 웹페이지에 악의적인 스크립트를 삽입하여 방문자들의 정보를 탈취하거나, 비정상적인 기능 수행을 유발하는 보안 약점이다.</li> <li>• HTML 태그의 사용을 제한하거나 스크립트에 삽입되지 않도록 '&lt;', '&gt;', '&amp;' 등의 문자를 다른 문자로 치환함으로써 방지할 수 있다.</li> </ul> |
| 운영체제 명령어 삽입             | <ul style="list-style-type: none"> <li>• 외부 입력값을 통해 시스템 명령어의 실행을 유도함으로써 권한을 탈취하거나 시스템 장애를 유발하는 보안 약점이다.</li> <li>• 웹 인터페이스를 통해 시스템 명령어가 전달되지 않도록 하고, 외부 입력값을 검증 없이 내부 명령어로 사용하지 않음으로써 방지할 수 있다.</li> </ul>                |
| 위험한 형식 파일 업로드           | <ul style="list-style-type: none"> <li>• 악의적인 명령어가 포함된 스크립트 파일을 업로드함으로써 시스템에 손상을 주거나, 시스템을 제어할 수 있는 보안 약점이다.</li> <li>• 업로드 되는 파일의 확장자 제한, 파일명의 암호화, 웹사이트와 파일 서버의 경로 분리, 실행 속성을 제거하는 등의 방법으로 방지할 수 있다.</li> </ul>         |
| 신뢰되지 않는 URL 주소로 자동접속 연결 | <ul style="list-style-type: none"> <li>• 입력 값으로 사이트 주소를 받는 경우 이를 조작하여 방문자를 피싱 사이트로 유도하는 보안 약점이다.</li> <li>• 연결되는 외부 사이트의 주소를 화이트 리스트로 관리함으로써 방지할 수 있다.</li> </ul>                                                         |

1. 다음은 입력 데이터로 인해 발생할 수 있는 보안 약점에 대한 설명이다. 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

입력란에 SQL을 삽입하여 무단으로 DB를 조회하거나 조작할 경우 발생할 수 있는 문제를 방지하려면 ( )에 사용되는 입력 데이터에 예약어 및 특수문자가 입력되지 않게 필터링 되도록 설정하면 된다. ( )는 질의어 코드를 문자열 변수에 넣어 조건에 따라 질의를 동적으로 변경하여 처리하는 방식이다.

답 :

2. 다음은 입력 데이터로 인해 발생할 수 있는 문제를 예방하기 위한 방법에 대한 설명이다. 괄호에 들어갈 가장 적합한 용어를 쓰시오.

입력 값으로 사이트 주소를 받는 경우 이를 조작하여 방문자를 피싱 사이트로 유도할 수 있다. 이를 방지하려면 연결되는 외부 사이트의 주소를 ( )로 관리해야 한다.

답 :

정답 1. 동적 쿼리 2. 화이트 리스트

### [핵심129] 보안 점검 - 보안 기능

- 소프트웨어 개발의 구현 단계에서 코딩하는 기능인 인증, 접근제어, 기밀성, 암호화 등을 올바르게 구현하기 위한 보안 점검 항목들이다.
- 보안 기능에 대한 점검을 수행하지 않을 경우 발생할 수 있는 보안 약점

|                      |                                                                                                                                                                          |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 적절한 인증 없이<br>중요기능 허용 | <ul style="list-style-type: none"> <li>• 보안검사를 우회하여 인증과정 없이 중요한 정보 또는 기능에 접근 및 변경이 가능하다.</li> <li>• 중요정보나 기능을 수행하는 페이지에서는 재인증 기능을 수행하도록 하여 방지할 수 있다.</li> </ul>          |
| 부적절한 인가              | <ul style="list-style-type: none"> <li>• 접근제어 기능이 없는 실행경로를 통해 정보 또는 권한을 탈취할 수 있다.</li> <li>• 모든 실행경로에 대해 접근제어 검사를 수행하고, 사용자에게는 반드시 필요한 접근 권한만을 부여하여 방지할 수 있다.</li> </ul> |

|                      |                                                                                                                                                                                                  |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 중요한 자원에 대한 잘못된 권한 설정 | <ul style="list-style-type: none"> <li>• 권한 설정이 잘못된 자원에 접근하여 해당 자원을 임의로 사용할 수 있다.</li> <li>• 소프트웨어 관리자만 자원들을 읽고 쓸 수 있도록 설정하고, 인가되지 않은 사용자의 중요 자원에 대한 접근 여부를 검사함으로써 방지할 수 있다.</li> </ul>          |
| 취약한 암호화 알고리즘 사용      | <ul style="list-style-type: none"> <li>• 암호화된 환경설정 파일을 해독하여 비밀번호 등의 중요정보를 탈취할 수 있다.</li> <li>• 안전한 암호화 알고리즘을 이용하고, 업무 관련 내용이나 개인정보 등에 대해서는 IT보안인증사무국이 안정성을 확인한 암호모듈을 이용함으로써 방지할 수 있다.</li> </ul> |
| 중요 정보 평문 저장 및 전송     | <ul style="list-style-type: none"> <li>• 암호화되지 않은 평문 데이터를 탈취하여 중요한 정보를 획득할 수 있다.</li> <li>• 중요한 정보를 저장하거나 전송할 때는 반드시 암호화 과정을 거치도록 하고, HTTPS 또는 SSL과 같은 보안 채널을 이용함으로써 방지할 수 있다.</li> </ul>        |
| 하드코딩된 비밀번호           | <ul style="list-style-type: none"> <li>• 소스코드 유출 시 내부에 하드코딩된 패스워드를 이용하여 관리자 권한을 탈취할 수 있다.</li> <li>• 패스워드는 암호화하여 별도의 파일에 저장하고, 디폴트 패스워드나 디폴트 키의 사용을 피함으로써 방지할 수 있다.</li> </ul>                   |

1. 다음은 보안 기능에 대한 점검을 수행하지 않을 경우 발생할 수 있는 보안 약점에 대한 설명이다. 괄호(①~②)에 들어갈 알맞은 용어를 쓰시오.

개인정보나 인증정보와 같은 중요한 정보를 저장하거나 전송할 때는 반드시 암호화 과정을 거치도록 하고, ( ① ) 또는 ( ② )과 같은 보안 채널을 이용함으로써 방지할 수 있다. ( ① )는 웹브라우저와 서버 간의 안전한 통신을 위해 암호 통신규약을 결합한 것이다. ( ② )는 데이터를 송·수신하는 두 컴퓨터 사이에 위치하여 인증, 암호화, 무결성을 보장하는 업계 표준 프로토콜이다.

답

① :

② :

정답 1. ① HTTPS(Hypertext Transfer Protocol Secure)

② SSL(Secure Sockets Layer)

### [핵심 130] 보안 점검 - 시간 및 상태

- 동시 수행을 지원하는 병렬 처리 시스템이나 다수의 프로세스가 동작하는 환경에서 시간과 실행 상태를 관리하여 시스템이 원활하게 동작되도록 하기 위한 보안 검증 항목들이다.
- 시간 및 상태를 점검하지 않은 코딩이 유발하는 보안 약점

|                     |                                                                                                                                                                                                                   |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TOCTOU 경쟁 조건        | <ul style="list-style-type: none"> <li>• 검사 시점(Time Of Check)과 사용 시점(Time Of Use)을 고려하지 않고 코딩하는 경우 발생하는 보안 약점이다.</li> <li>• 코드 내에 동기화 구문을 사용하여 해당 자원에는 한 번에 하나의 프로세스만 접근 가능하도록 구성함으로써 방지할 수 있다.</li> </ul>        |
| 종료되지 않는 반복문 또는 재귀함수 | <ul style="list-style-type: none"> <li>• 반복문이나 재귀함수에서 종료 조건을 정의하지 않았거나 논리 구조상 종료될 수 없는 경우 발생하는 보안 약점이다.</li> <li>• 모든 반복문이나 재귀함수의 수행 횟수를 제한하는 설정을 추가하거나, 종료 조건을 점검하여 반복 또는 호출의 종료 여부를 확인함으로써 방지할 수 있다.</li> </ul> |

1. 다음은 시간 및 상태를 점검하지 않은 코딩이 유발하는 보안 약점에 대한 설명이다. 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

검사 시점(Time Of Check)과 사용 시점(Time Of Use)을 고려하지 않고 코딩하는 경우 프로세스가 가진 자원 정보와 실제 자원 상태가 일치하지 않는 오류가 발생할 수 있다. 이를 방지하려면 코드 내에 ( ) 구문을 사용하여 해당 자원에는 한 번에 하나의 프로세스만 접근 가능하도록 구성해야 한다. ( ) 구문은 공유 자원에 대해 둘 이상의 프로세스가 접근하는 것을 막는 구문이다.

답 :

정답 1. 동기화

### [핵심 131] 보안 점검 - 에러처리

- 소프트웨어 실행 중 발생할 수 있는 오류(Error)들을 사전에 정의하여 오류로 인해 발생할 수 있는 문제들을 예방하기 위한 보안 점검 항목들이다.
- 에러처리의 미비로 인한 코딩이 유발하는 보안 약점

|                 |                                                                                                                                                                                                                                                                                                              |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 오류 메시지를 통한 정보노출 | <ul style="list-style-type: none"> <li>• 오류 발생으로 실행 환경, 사용자 정보, 디버깅 정보 등의 중요 정보를 소프트웨어가 메시지로 외부에 노출하는 보안 약점이다.</li> <li>• 예외처리 구문에 예외의 이름이나 스택 트레이스를 출력하도록 코딩한 경우 해커는 소프트웨어의 내부구조를 쉽게 파악할 수 있다.</li> <li>• 오류 발생 시 가능한 한 내부에서만 처리되도록 하거나 메시지를 출력할 경우 최소한의 정보 또는 사전에 준비된 메시지만 출력되도록 함으로써 방지할 수 있다.</li> </ul> |
| 오류 상황 대응 부재     | <ul style="list-style-type: none"> <li>• 소프트웨어 개발 중 예외처리를 하지 않았거나 미비로 인해 발생하는 보안 약점이다.</li> <li>• 오류가 발생할 수 있는 부분에 예외처리 구문을 작성하고, 제어문을 활용하여 오류가 악용되지 않도록 코딩함으로써 방지할 수 있다.</li> </ul>                                                                                                                         |
| 부적절한 예외처리       | <ul style="list-style-type: none"> <li>• 함수의 반환값 또는 오류들을 세분화하여 처리하지 않고 광범위하게 묶어 한 번에 처리하거나, 누락된 예외가 존재할 때 발생하는 보안 약점이다.</li> <li>• 모든 함수의 반환값이 의도대로 출력되는지 확인하고, 세분화된 예외처리를 수행함으로써 방지할 수 있다.</li> </ul>                                                                                                       |

1. 다음은 에러처리의 미비로 인한 코딩이 유발하는 보안 약점에 대한 설명이다. 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

( )는 오류가 발생한 위치를 추적하기 위해 소프트웨어가 실행 중에 호출한 메소드의 리스트를 기록한 것으로 예외처리 구문에 ( )를 출력하도록 코딩한 경우 해커는 소프트웨어의 내부구조를 쉽게 파악할 수 있다. 이를 방지하려면 오류 발생 시 가능한 한 내부에서만 처리되도록 하거나 메시지를 출력할 경우 최소한의 정보 또는 사전에 준비된 메시지만 출력되도록 해야 한다.

답 :

정답 1. 스택 트레이스(Stack Trace)

**[핵심 132] 보안 점검 - 코드 오류**

- 소프트웨어 구현 단계에서 개발자들이 코딩 중 실수하기 쉬운 형(Type) 변환, 자원 반환 등의 오류를 예방하기 위한 보안 점검 항목들이다.
- 코드 오류로 발생할 수 있는 보안 약점

|                         |                                                                                                                                                                                                                                                                                                                                    |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 널 포인터(Null Pointer) 역참조 | <ul style="list-style-type: none"> <li>• 널 포인터가 가리키는 메모리에 어떠한 값을 저장할 때 발생하는 보안 약점이다.</li> <li>• 널 포인터(Null Pointer) : 널(Null)은 값이 없음을 의미하며, 포인터(Pointer)는 메모리의 위치를 가리키는 요소이다. 널 포인터(Null Pointer)는 포인터에 널(Null)이 저장되어 어떠한 곳도 가리키지 못하는 상태의 요소를 말한다.</li> <li>• 널(Null)이 될 수 있는 포인터를 이용하기 전에 널 값을 갖고 있는지 검사함으로써 방지할 수 있다.</li> </ul> |
| 부적절한 자원 해제              | <ul style="list-style-type: none"> <li>• 자원을 반환하는 코드를 누락하거나 프로그램 오류로 할당된 자원을 반환하지 못했을 때 발생하는 보안 약점이다.</li> <li>• 프로그램 내에 자원 반환 코드가 누락되었는지 확인하고, 오류로 인해 함수가 중간에 종료되었을 때 예외처리에 관계없이 자원이 반환되도록 코딩함으로써 방지할 수 있다.</li> </ul>                                                                                                            |
| 해제된 자원 사용               | <ul style="list-style-type: none"> <li>• 이미 사용이 종료되어 반환된 메모리를 참조하는 경우 발생하는 보안 약점이다.</li> <li>• 반환된 메모리에 접근할 수 없도록 주소를 저장하고 있는 포인터를 초기화함으로써 방지할 수 있다.</li> </ul>                                                                                                                                                                    |
| 초기화되지 않은 변수 사용          | <ul style="list-style-type: none"> <li>• 변수 선언 후 값이 부여되지 않은 변수를 사용할 때 발생하는 보안 약점이다.</li> <li>• 변수 선언 시 할당된 메모리를 초기화함으로써 방지할 수 있다.</li> </ul>                                                                                                                                                                                       |

1. 다음은 코드 오류로 발생할 수 있는 보안 약점에 대한 설명이다. 괄호에 공통적으로 들어갈 가장 적합한 용어를 쓰시오.

( )는 포인터에 널(Null)이 저장되어 어떠한 곳도 가리키지 못하는 상태의 요소를 말하는 것으로 ( )가 가리키는 메모리에 어떠한 값을 저장할 경우 ( ) 역참조가 발생한다.

답 :

**정답** 1. 널 포인터(Null Pointer)

**[핵심 133] 보안 점검 - 캡슐화**

- 정보 은닉이 필요한 중요한 데이터와 기능을 불충분하게 캡슐화하거나 잘못 사용함으로써 발생할 수 있는 문제를 예방하기 위한 보안 점검 항목들이다.
- 캡슐화로 인해 발생할 수 있는 보안 약점

|                              |                                                                                                                                                                                                                              |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 잘못된 세션에 의한 정보 노출             | <ul style="list-style-type: none"> <li>• 다중 스레드(Multi-Thread) 환경에서 멤버 변수에 정보를 저장할 때 발생하는 보안 약점이다.</li> <li>• 멤버 변수보다 지역 변수를 활용하여 변수의 범위를 제한함으로써 방지할 수 있다.</li> </ul>                                                         |
| 제거되지 않고 남은 디버그 코드            | <ul style="list-style-type: none"> <li>• 개발 중에 버그 수정이나 결과값 확인을 위해 남겨둔 코드들로 인해 발생하는 보안 약점이다.</li> <li>• 소프트웨어를 배포하기 전에 코드 검사를 통해 남아있는 디버그 코드를 삭제함으로써 방지할 수 있다.</li> </ul>                                                     |
| 시스템 데이터 정보 노출                | <ul style="list-style-type: none"> <li>• 시스템의 내부 정보를 시스템 메시지 등을 통해 외부로 출력하도록 코딩했을 때 발생하는 보안 약점이다.</li> <li>• 시스템 메시지를 통해 노출되는 메시지는 최소한의 정보만을 제공함으로써 방지할 수 있다.</li> </ul>                                                     |
| Public 메소드로부터 반환된 Private 배열 | <ul style="list-style-type: none"> <li>• 선언된 클래스 내에서만 접근이 가능한 Private 배열을 모든 클래스에서 접근이 가능한 Public 메소드에서 반환할 때 발생하는 보안 약점이다.</li> <li>• Private 배열을 별도의 메소드를 통해 조작하거나, 동일한 형태의 복제본으로 반환받은 후 값을 전달하는 방식으로 방지할 수 있다.</li> </ul> |
| Private 배열에 Public 데이터 할당    | <ul style="list-style-type: none"> <li>• Private 배열에 Public으로 선언된 데이터 또는 메소드의 파라미터를 저장할 때 발생하는 보안 약점이다.</li> <li>• Public으로 선언된 데이터를 Private 배열에 저장할 때, 레퍼런스가 아닌 값을 직접 저장함으로써 방지할 수 있다.</li> </ul>                           |

**잠깐만요 ① Java의 접근 제어자(제한자)**

Java에서 특정 개체를 선언할 때 외부로부터의 접근을 제한하기 위해 사용되는 예약어로 Public, Private, Protected, Default가 있습니다.

2020년 1, 2회 기사 필기

1. Java에서 특정 개체를 선언할 때 외부로부터의 접근을 제한하기 위해 사용되는 접근 제한자의 종류 4가지를 쓰시오.

답 :

정답 1. Public, Private, Protected, Default

### [핵심 134] 보안 점검 - API 오용

- 소프트웨어 구현 단계에서 API를 잘못 사용하거나 보안에 취약한 API를 사용하지 않도록 하기 위한 보안 검증 항목들이다.
- API 오용으로 발생할 수 있는 보안 약점

|                             |                                                                                                                                                                                                                                                                                                                                    |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DNS Lookup에<br>의존한 보안<br>결정 | <ul style="list-style-type: none"> <li>• 도메인명에 의존하여 인증이나 접근 통제 등의 보안 결정을 내리는 경우 발생하는 보안 약점이다.</li> <li>• DNS 검색을 통해 도메인 이름을 비교하지 않고 IP 주소를 직접 입력하여 접근함으로써 방지할 수 있다.</li> </ul>                                                                                                                                                     |
| 취약한 API<br>사용               | <ul style="list-style-type: none"> <li>• 보안 문제로 사용이 금지된 API를 사용하거나, 잘못된 방식으로 API를 사용했을 때 발생하는 보안 약점이다.</li> <li>• 보안 문제로 금지된 대표적인 API에는 C언어의 문자열 함수 <code>strcat()</code>, <code>strcpy()</code>, <code>sprintf()</code> 등이 있다.</li> <li>• 보안 문제로 금지된 함수는 안전한 함수로 대체하고, API의 매뉴얼을 참고하여 보안이 보장되는 인터페이스를 사용함으로써 방지할 수 있다.</li> </ul> |

1. 다음은 API 오용으로 발생할 수 있는 보안 약점에 대한 설명이다. 괄호에 들어갈 가장 적합한 용어를 쓰시오.

도메인명에 의존하여 인증이나 접근 통제 등의 보안 결정을 내리는 경우 공격자는 ( )을 악용하여 인증이나 접근 통제를 우회하는 방법으로 권한을 탈취한다. 이를 방지하려면 IP 주소를 직접 입력하여 접근해야 한다.

답 :

정답 1. DNS Lookup

### [핵심 135] 암호 알고리즘

- 패스워드, 주민번호, 은행계좌와 같은 중요정보를 보호하기 위해 평문을 암호화된 문장으로 만드는 절차 또는 방법을 의미한다.
- 암호 알고리즘은 해시(Hash)를 사용하는 단방향 암호화 방식과, 개인키 및 공개키로 분류되는 양방향 암호화 방식이 있다.

|                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 개인키<br>암호화(Private<br>Key Encryption)<br>기법 | <ul style="list-style-type: none"> <li>• 동일한 키로 데이터를 암호화하고 복호화한다.</li> <li>• 개인키 암호화 기법은 대칭 암호 기법 또는 단일키 암호화 기법이라고도 한다.</li> </ul>                                                                                                                                                                                                                                                                                                                  |
| 공개키<br>암호화(Public<br>Key Encryption)<br>기법  | <ul style="list-style-type: none"> <li>• 데이터를 암호화할 때 사용하는 공개키(Public Key)는 데이터베이스 사용자에게 공개하고, 복호화할 때의 비밀키(Secret Key)는 관리자가 비밀리에 관리한다.</li> <li>• 공개키 암호화 기법은 비대칭 암호 기법이라고도 함</li> </ul>                                                                                                                                                                                                                                                            |
| 해시(Hash)                                    | <ul style="list-style-type: none"> <li>• 임의의 길이의 입력 데이터나 메시지를 고정된 길이의 값이나 키로 변환하는 것을 의미한다.</li> <li>• 해시 함수의 종류 : SHA 시리즈, MD5, SNEFRU 등 <ul style="list-style-type: none"> <li>- SHA 시리즈 : 1993년에 미국 NSA가 제작하고 미국 국립표준기술연구소(NIST)에서 표준으로 채택한 암호화 알고리즘으로 가장 많이 사용됨</li> <li>- MD5 : 1991년 Rivest가 MD4를 개선한 암호화 알고리즘으로 각각의 512 비트 짜리 입력 메시지 블록에 대해 차례로 동작함</li> <li>- SNEFRU : 1990년에 R.C. Merkle에 의해 제안된 128, 254비트 암호화 알고리즘</li> </ul> </li> </ul> |

2020년 1회 기사 필기

1. 다음 설명에 가장 적합한 알고리즘을 쓰시오.

1991년 R.rivest가 MD4를 개선한 암호화 알고리즘으로 각각의 512 비트짜리 입력 메시지 블록에 대해 차례로 동작한다. 각 512 비트 입력 메시지 블록을 처리하고 나면 128 비트 스테이트(State)의 값이 변하게 된다.

답 :

정답 1. MD5

### [핵심136] 개인키 암호화 방식과 공개키 암호화 방식에서 사용되는 주요 암호화 알고리즘

|                                           |                                                                                                                                                                                 |
|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SEED                                      | <ul style="list-style-type: none"> <li>1999년 한국인터넷진흥원(KISA)에서 개발한 블록 암호화 알고리즘</li> <li>블록 크기는 128비트이며, 키 길이에 따라 128, 256으로 분류</li> </ul>                                        |
| ARIA(Academy, Research Institute, Agency) | <ul style="list-style-type: none"> <li>2004년 국가정보원과 산학연협회가 개발한 블록 암호화 알고리즘</li> <li>블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류</li> </ul>                                      |
| DES(Data Encryption Standard)             | <ul style="list-style-type: none"> <li>1975년 미국 NBS에서 발표한 개인키 암호화 알고리즘</li> <li>블록 크기는 64비트이며, 키 길이는 56비트</li> </ul>                                                            |
| AES(Advanced Encryption Standard)         | <ul style="list-style-type: none"> <li>2001년 미국 표준 기술 연구소(NIST)에서 발표한 개인키 암호화 알고리즘</li> <li>블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류</li> </ul>                              |
| RSA(Rivest Shamir Adleman)                | <ul style="list-style-type: none"> <li>1978년 MIT의 라이베스트(Rivest), 샤미르(Shamir), 애들먼(Adelman)에 의해 제안된 공개키 암호화 알고리즘</li> <li>소인수 분해 문제를 이용한 공개키 암호화 기법에 널리 사용되는 암호화 알고리즘</li> </ul> |

2020년 1, 2회 기사 필기

1. 소인수 분해 문제를 이용한 공개키 암호화 기법에 널리 사용되는 암호화 알고리즘이 무엇인지 쓰시오.

답 :

2. 다음 설명에 가장 적합한 알고리즘을 쓰시오.

2001년 미국 표준 기술 연구소(NIST)에서 발표한 개인키 암호화 알고리즘으로 블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류된다.

답 :

정답 1. RSA(Rivest Shamir Adleman)

2. AES(Advanced Encryption Standard)

## 10장 | 응용 SW 기초 기술 활용

### [핵심137] 운영체제(OS; Operating System)

- 컴퓨터 시스템의 자원들을 효율적으로 관리하며, 사용자가 컴퓨터를 편리하고 효과적으로 사용할 수 있도록 환경을 제공하는 여러 프로그램의 모임이다.
- 목적 : 처리 능력 향상, 신뢰도 향상, 사용 가능도 향상, 반환 시간 단축
- 성능 평가 기준

|                            |                                     |
|----------------------------|-------------------------------------|
| 처리 능력<br>(Throughput)      | 일정 시간 내에 시스템이 처리하는 일의 양             |
| 반환 시간(Turn<br>Around Time) | 시스템에 작업을 의뢰한 시간부터 처리가 완료될 때까지 걸린 시간 |
| 사용 가능도<br>(Availability)   | 시스템을 사용할 필요가 있을 때 즉시 사용 가능한 정도      |
| 신뢰도(Reliability)           | 시스템이 주어진 문제를 정확하게 해결하는 정도           |

#### • 기능

- 프로세서, 기억장치, 입·출력장치, 파일 및 정보 등의 자원 관리
- 자원의 효과적인 경영 스케줄링 기능 제공
- 사용자와 시스템 간의 편리한 인터페이스 제공
- 시스템의 각종 하드웨어와 네트워크 관리·제어
- 시스템의 오류 검사 및 복구, 데이터 관리, 데이터 및 자원 공유

1. 다음은 운영체제(OS)의 성능 평가 기준에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 성능 평가 기준을 쓰시오.

|          |                                            |
|----------|--------------------------------------------|
| ( ① )    | 시스템을 사용할 필요가 있을 때 즉시 사용 가능한 정도를 의미한다.      |
| 처리<br>능력 | 일정 시간 내에 시스템이 처리하는 일의 양을 의미한다.             |
| ( ② )    | 시스템이 주어진 문제를 정확하게 해결하는 정도를 의미한다.           |
| 반환<br>시간 | 시스템에 작업을 의뢰한 시간부터 처리가 완료될 때까지 걸린 시간을 의미한다. |



답

- ① :  
② :

2. 운영체제(OS)는 컴퓨터 시스템의 자원들을 효율적으로 관리하며, 사용자가 컴퓨터를 편리하고 효과적으로 사용할 수 있도록 환경을 제공하는 여러 프로그램의 모임이다. 운영체제의 목적 4가지를 쓰시오.

답 :

정답 1. ① 사용 가능성(Availability) ② 신뢰도(Reliability)

2. 처리 능력 향상, 신뢰도 향상, 사용 가능성 향상, 반환 시간 단축

### [핵심138] Windows의 특징

- 그래픽 사용자 인터페이스(GUI; Graphic User Interface) : 키보드로 명령어를 직접 입력하지 않고, 마우스로 아이콘이나 메뉴를 선택하여 모든 작업을 수행하는 방식
- 선점형 멀티태스킹(Premptive Multi-Tasking) : 동시에 여러 개의 프로그램을 실행하는 멀티태스킹을 하면서 운영체제가 각 작업의 CPU 이용 시간을 제어하여 응용 프로그램 실행중 문제가 발생하면 해당 프로그램을 강제 종료시키고 모든 시스템 자원을 반환하는 방식
- PnP(Plug and Play, 자동 감지 기능) : 컴퓨터 시스템에 프린터나 사운드 카드 등의 하드웨어를 설치했을 때, 해당 하드웨어를 사용하는 데 필요한 시스템 환경을 운영체제가 자동으로 구성해 주는 기능
- OLE(Object Linking and Embedding) : 다른 여러 응용 프로그램에서 작성된 문자나 그림 등의 개체(Object)를 현재 작성 중인 문서에 자유롭게 연결(Linking)하거나 삽입(Embedding)하여 편집할 수 있게 하는 기능

2020년 2회 기능사 실기

1. CUI(Command User Interface)는 명령어를 직접 입력해서 명령을 수행하는 방식으로 DOS에서 주로 사용한다. 반면 마우스로 아이콘이나 메뉴를 선택하여 명령을 수행하는 방식으로 주로 Windows에서 사용하는 것이 있다. 무엇이라고 하는지 쓰시오.

답 :

2. 컴퓨터 시스템에 프린터나 사운드 카드 등의 하드웨어를 설치했을 때, 해당 하드웨어를 사용하는 데 필요한 시스템 환경을 운영체제가 자동으로 구성해 주는 Windows의 기능을 쓰시오.

답 :

정답 1. GUI(Graphic User Interface, 그래픽 사용자 인터페이스)

2. PnP(Plug and Play, 자동 감지 기능)

### [핵심139] UNIX / LINUX

#### UNIX

- 1960년대 AT&T 벨(Bell) 연구소, MIT, General Electric이 공동 개발한 운영체제이다.
- 시분할 시스템(Time Sharing System)을 위해 설계된 대화식 운영체제로, 소스가 공개된 개방형 시스템(Open System)이다.
- 대부분 C 언어로 작성되어 있어 이식성이 높으며 장치, 프로세스 간의 호환성이 높다.
- 다중 사용자(Multi-User), 다중 작업(Multi-Tasking)을 지원한다.
- 트리(Tree) 구조의 파일 시스템을 갖는다.

#### LINUX

- 1991년 리누스 토발즈(Linus Torvalds)가 UNIX를 기반으로 개발한 운영체제이다.
- UNIX와 완벽하게 호환되고, 대부분의 특징이 UNIX와 동일하다.
- 프로그램 소스 코드가 무료로 공개되어 있다.

1. 1960년대 AT&T 벨(Bell) 연구소, MIT, General Electric이 공동 개발한 운영체제이다. 대부분 C 언어로 작성되었고, 멀티유저와 멀티태스킹을 지원하며, Tree 구조의 파일 시스템을 갖는 운영체제는 무엇인지 쓰시오.

답 :

정답 1. UNIX

**[핵심 140]** UNIX 시스템의 구성**커널(Kernel)**

- UNIX의 가장 핵심적인 부분이다.
- 하드웨어를 보호(캡슐화)하고, 프로그램들과 하드웨어 간의 인터페이스 역할을 담당한다.
- 프로세스, 기억장치, 파일, 입·출력 관리, 프로세스 간 통신, 데이터 전송 및 변환 등 여러 가지 기능을 수행한다.
- 컴퓨터 부팅 시 주기억장치에 적재되어 상주하면서 실행된다.

**셸(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 보조기억장치에서 교체 처리가 가능하다.
- 종류 : Bourne Shell, C Shell, Korn Shell 등

1. 컴퓨터 부팅 시 주기억장치에 적재되어 상주하면서 실행되며, 프로세스, 기억장치, 파일, 입·출력 관리, 프로세스 간 통신, 데이터 전송 및 변환 등 여러 가지 기능을 수행하는 UNIX의 가장 핵심적인 구성 요소를 쓰시오.

답 :

2. UNIX 시스템에서 명령어 해석기로 사용자의 명령어를 인식하여 필요한 프로그램을 호출하고 그 명령을 수행하며, DOS의 COMMAND.COM과 같은 기능을 수행하는 것은 무엇인지 쓰시오.

답 :

**정답** 1. 커널(Kernel) 2. 셸(Shell)

**[핵심 141]** Windows의 CLI 기본 명령어

- dir : 현재 디렉터리의 파일 목록 표시
- copy : 파일 복사
- del : 파일 삭제
- type : 파일 내용 표시
- ren : 파일 이름 변경
- md : 디렉터리 생성
- cd : 동일한 드라이브에서 디렉터리 위치 변경
- cls : 화면 내용 지움
- attrib : 파일 속성 변경
- find : 파일에서 문자열 찾기
- chkdsk : 디스크 상태 점검
- format : 디스크 표면을 트랙과 섹터로 나누어 초기화
- move : 파일 이동

**2020년 1회 기능사 실기**

1. Windows의 명령 프롬프트 창에서 파일 전체를 나열하는 명령어로, UNIX 명령어의 ls와 동일한 기능을 하는 명령어를 쓰시오.

답 :

2. Windows의 명령 프롬프트 창에서 현재 드라이브의 상태를 확인할 때 사용하는 명령어를 쓰시오.

답 :

**정답** 1. dir 2. chkdsk

**[핵심 142]** UNIX / LINUX의 CLI 기본 명령어

- cat : 파일 내용을 화면에 표시
- cd : 디렉터리 위치 변경
- chmod : 파일의 보호 모드를 설정하여 파일 사용 허가 지정
- chown : 파일 소유자와 그룹 변경
- cp : 파일 복사
- rm : 파일 삭제
- find : 파일 찾기



- **fsck** : 파일 시스템 검사 및 보수
- **kill** : PID(프로세스 고유 번호)를 이용하여 프로세스 종료
- **killall** : 프로세스의 이름을 이용하여 프로세스 종료
- **ls** : 현재 디렉터리의 파일 목록 표시
- **mkdir** : 디렉터리 생성
- **rmdir** : 디렉터리 삭제
- **mv** : 파일 이동
- **ps** : 현재 실행중인 프로세스 표시
- **pwd** : 현재 작업중인 디렉터리 경로를 화면에 표시
- **top** : 시스템의 프로세스와 메모리 사용 현황 표시
- **who** : 현재 시스템에 접속해 있는 사용자 표시

1. 파일에 대한 액세스(읽기, 쓰기, 실행) 권한을 설정하여 사용자에게 제한적인 권한을 주려고 할 때 사용하는 UNIX 명령어를 쓰시오.

답 :

2. 다음은 UNIX의 주요 명령어에 대한 설명이다. 괄호(①~③)에 들어갈 알맞은 명령어를 쓰시오.

|       |                                    |
|-------|------------------------------------|
| ( ① ) | 파일 내용을 화면에 표시하는 명령어이다.             |
| rm    | 파일을 삭제하는 명령어이다.                    |
| ( ② ) | 파일 시스템을 검사 및 보수하여 무결성을 검사하는 명령어이다. |
| ls    | 현재 디렉터리 내의 파일 목록을 확인하는 명령어이다.      |
| ( ③ ) | 파일의 소유자를 변경하는 명령어이다.               |

답

- ① :  
② :  
③ :

정답 1. chmod 2. ① cat ② fsck ③ chown



### [핵심 143] 기억장치 관리 전략

보조기억장치의 프로그램이나 데이터를 주기억장치에 적재시키는 시기, 적재 위치 등을 지정하여 한정된 주기억장치의 공간을 효율적으로 사용하기 위한 것이다.

#### 반입(Fetch) 전략

보조기억장치에 보관중인 프로그램이나 데이터를 언제 주기억장치로 적재할 것인지를 결정하는 전략으로, 요구 반입(Demand Fetch)과 예상반입(Anticipatory Fetch)이 있다.

#### 배치(Placement) 전략

- 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

#### 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 전략으로, FIFO, OPT, LRU, LFU, NUR, SCR 등이 있다.

1. 아래 그림을 참조하여 주기억장치 관리 기법인 First-Fit, Best-Fit, Worst-Fit 방법을 각각 적용할 경우 15K의 프로그램이 할당될 영역을 차례로 표시하시오.

| 할당영역 | 운영체제 |
|------|------|
| 1    | 10k  |
| 2    | 20k  |
| 3    | 사용중  |
| 4    | 15k  |
| 5    | 30k  |

답

- ① First-Fit :  
 ② Best-Fit :  
 ③ Worst-Fit :

2. 기억장치의 관리 전략은 보조기억장치의 프로그램이나 데이터를 주기억장치에 적재시키는 시기, 적재 위치 등을 지정하여 한정된 주기억장치의 공간을 효율적으로 사용하기 위한 것이다. 기억장치 관리 전략의 종류 3가지를 쓰시오.

답 :

정답 1. ① 2 ② 4 ③ 5

2. 반입(Fetch) 전략, 배치(Placement) 전략, 교체(Replacement) 전략

2020 시나공 정보처리기사 필기 수록 내용

#### [핵심144] 가상 기억장치(Virtual Memory)

- 보조기억장치(하드디스크)의 일부를 주기억장치처럼 사용하는 것으로, 용량이 작은 주기억장치를 마치 큰 용량을 가진 것처럼 사용하는 것으로, 현재 사용되는 운영체제에서 흔히 사용되는 기법이다.
- 가상 기억장치 구현 기법
  - 페이지징(Paging) 기법 : 가상 기억장치에 보관되어 있는 프로그램과 주기억장치의 영역을 동일한 크기로 나눈 후 나뉜 프로그램(페이지)을 동일하게 나뉜 주기억장치의 영역에 적재시켜 실행하는 기법으로, 외부 단편화는 발생하지 않으나 내부 단편화는 발생할 수 있음

- 세그먼테이션(Segmentation) 기법 : 가상 기억장치에 보관되어 있는 프로그램을 다양한 크기의 논리적인 단위로 나눈 후 주기억장치에 적재시켜 실행시키는 기법으로, 내부 단편화는 발생하지 않으나 외부 단편화는 발생할 수 있음

1. 다음이 설명하는 가상 기억장치(Virtual Memory)의 구현 기법을 쓰시오.

- 가상기억장치에 보관되어 있는 프로그램을 다양한 크기의 논리적인 단위로 나눈 후 주기억장치에 적재시켜 실행시키는 기법이다.
- 내부 단편화는 발생하지 않으나 외부 단편화는 발생할 수 있다.

답 :

정답 1. 세그먼테이션(Segmentation) 기법

2020 시나공 정보처리기사 필기 수록 내용

#### [핵심145] 가상기억장치 기타 관리 사항

- Locality(국부성)
  - 프로세스가 실행되는 동안 주기억장치를 참조할 때 일부 페이지만 집중적으로 참조하는 성질이 있다는 이론
  - 시간 구역성(Temporal Locality) : 프로세스가 실행되면서 하나의 페이지를 일정 시간 동안 집중적으로 액세스하는 현상
  - 공간 구역성(Spatial Locality) : 프로세스 실행 시 일정 위치의 페이지를 집중적으로 액세스하는 현상
- 워킹 셋(Working Set) : 프로세스가 일정 시간 동안 자주 참조하는 페이지들의 집합
- 페이지 부재(Page Fault) : 프로세스 실행 시 참조할 페이지가 주기억장치에 없는 현상으로, 페이지 부재가 일어나는 횟수를 페이지 부재 빈도(Page Fault Frequency)라고 함
- 스래싱(Thrashing) : 프로세스의 처리 시간보다 페이지 교체에 소요되는 시간이 더 많아지는 현상

1. 다음이 설명하는 용어를 쓰시오.

- 프로세스가 일정 시간 동안 자주 참조하는 페이지들의 집합이다.
- 데닝(Denning)이 제안한 프로그램의 움직임에 대한 모델로, 프로그램의 Locality 특징을 이용한다.

답 :

2. 프로세스가 실행되는 동안 주기억장치를 참조할 때 일부 페이지만 집중적으로 참조하는 성질이 있다는 이론으로, 스래싱을 방지하기 위한 워킹 셋 이론의 기반이 되는 것은 무엇인지 쓰시오.

답 :

정답 1. 워킹 셋(Working Set) 2. Locality(국부성)

2020 시나공 정보처리기사 필기 수록 내용

[핵심 146] 페이지 교체 알고리즘

- OPT(OPTimal replacement, 최적 교체) : 앞으로 가장 오랫동안 사용하지 않을 페이지를 교체하는 기법
- FIFO(First In First Out) : 각 페이지가 주기억장치에 적재될 때마다 그때의 시간을 기억시켜 가장 먼저 들어와서 가장 오래 있었던 페이지를 교체하는 기법
- LRU(Least Recently Used) : 최근에 가장 오랫동안 사용하지 않은 페이지를 교체하는 기법
- LFU(Least Frequently Used) : 사용 빈도가 가장 적은 페이지를 교체하는 기법
- NUR(Not Used Recently) : 최근에 사용하지 않은 페이지를 교체하는 기법으로, 참조 비트(Reference Bit)와 변형 비트(Modified Bit)가 사용됨
- SCR(Second Chance Replacement, 2차 기회 교체) : 가장 오랫동안 주기억장치에 있던 페이지 중 자주 사용되는 페이지의 교체를 방지하기 위한 것으로, FIFO 기법의 단점을 보완하는 기법

1. 페이지 교체 알고리즘 중 최근에 가장 오랫동안 사용하지 않은 페이지를 교체하는 기법을 쓰시오.

답 :

2. 페이지 교체 알고리즘 중 최근에 사용하지 않은 페이지를 교체하며, 참조 비트(Reference Bit)와 변형 비트(Modified Bit)가 사용되는 기법은 무엇인지 쓰시오.

답 :

정답 1. LRU(Least Recently Used) 2. NUR(Not Used Recently)

[핵심 147] 프로세스(Process) 관리

- 프로세스는 프로세서(처리기, CPU)에 의해 처리되는 사용자 프로그램, 시스템 프로그램, 즉 실행중인 프로그램을 의미하며, 작업(Job), 태스크(Task)라고도 한다.
- 프로세스 상태 전이

|                      |                                                                      |
|----------------------|----------------------------------------------------------------------|
| 제출(Submit)           | 작업을 처리하기 위해 사용자가 작업을 시스템에 제출한 상태                                     |
| 접수(Hold)             | 제출된 작업이 스톱 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 |
|----------|-----------------------------------------------------------------------------------------------------------------------------------|

### 1. 다음이 설명하는 용어를 쓰시오.

프로세서에 의해 처리되는 사용자 프로그램, 시스템 프로그램, 즉 실행중인 프로그램을 의미하며, 다음과 같이 여러 형태로 정의할 수 있다.

- PCB를 가진 프로그램
- 실기억장치에 저장된 프로그램
- 프로세서가 할당되는 실체로서, 디스패치가 가능한 단위
- 프로시저가 활동중인 것
- 비동기적 행위를 일으키는 주체

답 :

2020년 1, 2회 기사 필기

### 2. 다음은 프로세스 상태 전이 관련 용어에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오.

- ( ① ) : 입·출력 작업이 완료되어 프로세스가 대기 상태에서 준비 상태로 전이 되는 과정이다.
- ( ② ) : 준비 상태에서 대기하고 있는 프로세스 중 하나가 프로세서를 할당받아 실행 상태로 전이되는 과정이다.
- Spooling : 입·출력할 데이터를 직접 입·출력장치에 보내지 않고 나중에 한꺼번에 입·출력하기 위해 디스크에 저장하는 과정이다.

답

① :

② :

**정답** 1. 프로세스(Process) 2. ① Wake Up ② Dispatch



### 전문가의 조언

핵심 148~핵심 151, 핵심 155의 내용은 2020년 2회 실기시험에 대비하여 새롭게 정리한 내용입니다. 하나도 빼놓지 말고 공부하세요.

### [핵심148] 비선점(Non-preemptive) 스케줄링

- 이미 할당된 CPU를 다른 프로세스가 강제로 빼앗아 사용할 수 없는 스케줄링 기법이다.
- FCFS(First Come First Service, 선입 선출) = FIFO(First In First Out) : 준비상태 큐에 도착한 순서에 따라 차례로 CPU를 할당하는 기법으로, 가장 간단한 알고리즘임
- SJF(Shortest Job First, 단기 작업 우선) : 준비상태 큐에서 기다리고 있는 프로세스들 중에서 실행 시간이 가장 짧은 프로세스에게 먼저 CPU를 할당하는 기법
- HRN(Highest Response-ratio Next)
  - 실행 시간이 긴 프로세스에 불리한 SJF 기법을 보완하기 위한 것으로, 대기 시간과 서비스(실행) 시간을 이용하는 기법
  - 우선순위 계산식 : (대기 시간 + 서비스 시간) / 서비스 시간
- 기한부(Deadline) : 프로세스에게 일정한 시간을 주어 그 시간 안에 프로세스를 완료하도록 하는 기법
- 우선순위(Priority) : 준비상태 큐에서 기다리는 각 프로세스마다 우선순위를 부여하여 그 중 가장 높은 프로세스에게 먼저 CPU를 할당하는 기법

### 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.

### 1. 다음이 설명하는 비선점 스케줄링 기법을 쓰시오.

대기 큐, 준비 완료 리스트, 작업준비 큐, 스케줄링 큐에 도착한 순서에 따라 차례로 CPU를 할당하는 기법으로, 먼저 도착한 것이 먼저 처리되어 공정성은 유지되지만 짧은 작업이 긴 작업을, 중요한 작업이 중요하지 않은 작업을 기다리게 된다.

답 :

2020년 1회 기사 실기

2. HRN 비선점형 스케줄링의 우선순위를 구하는 계산식을 쓰시오.

답 :

정답 1. FCFS(FIFO) 2. (대기 시간 + 서비스 시간) / 서비스 시간

### [핵심 149] 선점(Preemptive) 스케줄링

- 하나의 프로세스가 CPU를 할당받아 실행하고 있을 때 우선순위가 높은 다른 프로세스가 CPU를 강제로 빼앗아 사용할 수 있는 스케줄링 기법이다.
- 선점 우선순위 : 준비상태 큐의 프로세스들 중에서 우선순위가 가장 높은 프로세스에게 먼저 CPU를 할당하는 기법
- SRT(Shortest Remaining Time) : 비선점 스케줄링인 SJF 기법을 선점 형태로 변경한 기법으로, 선점 SJF 기법이라고도 함
- 라운드 로빈(RR; Round Robin) : 시분할 시스템(Time Sharing System)을 위해 고안된 방식으로, 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 배치됨
- 다단계 큐(MQ; Multi-level Queue) : 프로세스를 특정 그룹으로 분류할 수 있을 경우 그룹에 따라 각기 다른 준비상태 큐를 사용하는 기법
- 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 기법

1. 비선점 스케줄링인 SJF 기법을 선점 형태로 변경한 기법으로, 현재 실행중인 프로세스의 남은 시간과 준비상태 큐에 새로 도착한 프로세스의 실행 시간을 비교하여 가장 짧은 실행 시간을 요구하는 프로세스에게 CPU를 할당하는 선점 스케줄링 기법은 무엇인지 쓰시오.

답 :

2. 시분할 시스템(Time Sharing System)을 위해 고안된 방식으로, FCFS(FIFO) 알고리즘을 선점 형태로 변형한 기법은 무엇인지 쓰시오.

답 :

정답 1. SRT(Shortest Remaining Time)  
2. 라운드 로빈(RR; Round Robin)

### [핵심 150] 교착상태(Dead Lock)

- 상호 배제에 의해 나타나는 문제점으로, 둘 이상의 프로세스들이 자원을 점유한 상태에서 서로 다른 프로세스가 점유하고 있는 자원을 요구하며 무한정 기다리는 현상이다.
- 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

1. 교착상태(Dead Lock)의 개념을 설명하시오.

답 :

2020년 1, 2회 기사 필기

2. 교착상태(Dead Lock) 발생의 필요 충분 조건 4가지를 쓰시오.

답 :

정답 1. 교착상태는 둘 이상의 프로세스들이 자원을 점유한 상태에서 서로 다른 프로세스가 점유하고 있는 자원을 요구하며 무한정 기다리는 현상이다.

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

**[핵심151] 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 사용된다.

**발견(Detection) 기법**

시스템에 교착 상태가 발생했는지 점검하여 교착 상태에 있는 프로세스와 자원을 발견하는 것으로, 자원 할당 그래프 등을 사용한다.

**회복(Recovery) 기법**

교착 상태를 일으킨 프로세스를 종료하거나 교착 상태의 프로세스에 할당된 자원을 선점하여 프로세스나 자원을 회복하는 것이다.

1. 교착상태(Dead Lock)의 해결 방법 4가지를 쓰시오.

답 :

2020년 1, 2회 기사 필기

2. 은행원 알고리즘(Banker's Algorithm)은 교착상태(Dead Lock) 해결 방법 중 어떤 기법에 해당하는지 쓰시오.

답 :

**정답** 1. 예방 기법(Prevention), 회피 기법(Avoidance), 발견(Detection) 기법, 회복(Recovery) 기법

2. 회피 기법(Avoidance)

**[핵심152] 데이터베이스(Database)**

- 특정 조직의 업무를 수행하는 데 필요한 상호 관련된 데이터들의 모임이다.
- 데이터베이스의 정의
  - 통합된 데이터(Integrated Data) : 검색의 효율성을 위해 중복이 최소화된 데이터의 모임
  - 저장된 데이터(Stored Data) : 컴퓨터가 접근할 수 있는 저장 매체에 저장된 데이터
  - 운영 데이터(Operational Data) : 조직의 고유한 업무를 수행하는 데 존재 가치가 확실하고 없어서는 안 될 반드시 필요한 데이터
  - 공용 데이터(Shared Data) : 여러 응용 시스템들이 공동으로 소유하고 유지하는 데이터

1. 데이터베이스(Database)는 특정 조직의 업무를 수행하는 데 필요한 상호 관련된 데이터들의 모임으로 다음과 같이 정의할 수 있다. 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오.

|         |                                                        |
|---------|--------------------------------------------------------|
| 운영 데이터  | 조직의 고유한 업무를 수행하는 데 존재 가치가 확실하고 없어서는 안 될 반드시 필요한 데이터이다. |
| ( ① )   | 검색의 효율성을 위해 중복이 최소화된 데이터의 모임이다.                        |
| 저장된 데이터 | 컴퓨터가 접근할 수 있는 저장 매체에 저장된 데이터이다.                        |
| ( ② )   | 여러 응용 시스템들이 공동으로 소유하고 유지하는 데이터이다.                      |

**답**

① :

② :

**정답** 1. ① 통합된 데이터(Integrated Data)  
② 공용 데이터(Shared Data)



### [핵심 153] DBMS(DataBase Management System, 데이터베이스 관리 시스템)

사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성해주고, 데이터베이스를 관리해 주는 소프트웨어이다.

#### DBMS의 필수 기능

- 정의(Definition) 기능 : 모든 응용 프로그램들이 요구하는 데이터 구조를 지원하기 위해 데이터베이스에 저장될 데이터의 형(Type)과 구조에 대한 정의, 이용방식, 제약 조건 등을 명시하는 기능
- 조작(Manipulation) 기능 : 데이터 검색, 갱신, 삽입, 삭제 등을 체계적으로 처리하기 위해 사용자와 데이터베이스 사이의 인터페이스 수단을 제공하는 기능
- 제어(Control) 기능 : 데이터베이스를 접근하는 갱신, 삽입, 삭제 작업이 정확하게 수행되어 데이터의 무결성이 유지되도록 제어하는 기능

#### DBMS의 종류

- 계층형 DBMS : 트리(Tree) 구조를 이용해서 데이터의 상호관계를 계층적으로 정의한 DBMS로, 개체 타입 간에는 상위(Owner)와 하위(Member) 관계가 존재하며, 일 대 다(1:N) 대응 관계만 존재함
- 망형 DBMS : 그래프를 이용해서 데이터 논리 구조를 표현한 DBMS로, 상위(Owner)와 하위(Member) 레코드 사이에서 1:1, 1:N, N:M(다 대 다) 대응 관계를 모두 지원함
- 관계형 DBMS : 계층형과 망형 DBMS의 복잡한 구조를 단순화시킨 가장 널리 사용되는 DBMS로, 파일 구조처럼 구성된 2차원적인 표(Table)를 하나의 DB로 묶어서 테이블 내에 있는 속성들 간의 관계(Relationship)를 설정하거나 테이블 간의 관계를 설정하여 이용함

1. DBMS는 사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성해주고, 데이터베이스를 관리해 주는 소프트웨어이다. DBMS의 필수 기능 3가지를 쓰시오.

답 :

2. 계층형과 망형 DBMS의 복잡한 구조를 단순화시켰으며, 파일 구조처럼 구성된 2차원적인 Table을 하나의 DB로 묶어서 테이블 내에 있는 속성들 간의 Relationship을 설정하거나 테이블 간의 관계를 설정하여 이용하는 DBMS는 무엇인지 쓰시오.

답 :

정답 1. 정의(Definition), 조작(Manipulation), 제어(Control) 기능  
2. 관계형 DBMS

2020 시나공 정보처리기사 필기 수록 내용

### [핵심 154] 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

2020년 1, 2회 기사 필기

1. 데이터베이스의 분산된 물리적 환경에서 특정 지역의 컴퓨터 시스템이나 네트워크 장애가 발생해도 데이터 무결성을 보장하는 분산 데이터베이스의 목표를 쓰시오.

답 :

정답 1. 장애 투명성(Failure Transparency)

## [핵심155] 고급 데이터베이스

- 데이터 웨어하우스(Data Warehouse) : 급증하는 다량의 데이터를 효과적으로 분석하여 정보화하고 이를 여러 계층의 사용자들이 효율적으로 사용할 수 있도록 한 데이터베이스
- 데이터 마트(Data Mart) : 전사적으로 구축된 데이터 웨어하우스로부터 특정 주제나 부서 중심으로 구축된 소규모 단일 주제의 데이터 웨어하우스를 말함
- 데이터 마이닝(Data Mining) : 데이터 웨어하우스에 저장된 데이터 집합에서 사용자의 요구에 따라 유용하고 가능성 있는 정보를 발견하기 위한 기법
- OLAP(Online Analytical Processing) : 다차원으로 이루어진 데이터로부터 통계적인 요약 정보를 분석하여 의사결정에 활용하는 방식
- OLTP(Online Transaction Processing) : 온라인 업무 처리 형태의 하나로 네트워크상의 여러 이용자가 실시간으로 데이터베이스의 데이터를 갱신하거나 검색하는 등의 단위 작업을 처리하는 방식

2020년 1회 기사 실기

1. 데이터 마이닝(Data Mining)의 개념을 설명하시오.

답 :

2. 급증하는 다량의 데이터를 효과적으로 분석하여 정보화하고 이를 여러 계층의 사용자들이 효율적으로 사용할 수 있도록 한 데이터베이스는 무엇인지 쓰시오.

답 :

**정답** 1. 데이터 마이닝은 데이터 웨어하우스에 저장된 데이터 집합에서 사용자의 요구에 따라 유용하고 가능성 있는 정보를 발견하기 위한 기법이다.

2. 데이터 웨어하우스(Data Warehouse)



## [핵심156] ER(Entity Relationship) 모델 - ER 도형

ER 모델은 개념적 데이터 모델의 가장 대표적인 것으로, 1976년 피터 첸(Peter Chen)에 의해 제안되었으며, 다음과 같은 도형들을 이용하여 현실 세계를 표현한다.

| 도형 | 의미                       |
|----|--------------------------|
|    | 개체 타입(Entity Type)       |
|    | 관계 타입(Relationship Type) |
|    | 속성                       |
|    | 기본키(Primary Key) 속성      |
|    | 개체 타입과 속성을 연결            |
|    | 다중값 속성                   |
|    | 복합 속성                    |
|    | 개체 타입 간의 연관성             |

2020년 1,2회 기사 필기

1. 다음은 ER(Entity Relationship) 도형에 대한 설명이다. 괄호(①~③)에 들어갈 알맞은 용어를 쓰시오.

데이터베이스의 전체적인 논리적 구조는 다음과 같이 Peter Chen이 제안한 구성 요소들을 사용하여 개체 관계도(E-R Diagram)로 표현할 수 있다.

- 다이아몬드(◇) : ( ① )을 나타낸다.
- 사각형(□) : ( ② )을 나타낸다.
- 타원(○) : ( ③ )을 나타낸다.
- 선(—) : 속성과 개체 집합을 연결시키며, 개체 집합과 관계를 연결시킨다.

답

- ① :  
② :  
③ :

**정답** 1. ① 관계 타입(Relationship Type) ② 개체 타입(Entity Type)  
③ 속성



### [핵심 157] ER(Entity Relationship) 모델 - 관계 및 관계 타입

- 관계는 2개 이상의 개체 사이에 존재하는 연관성을 뜻하며, 관계 타입은 같은 관계들의 집합 또는 틀을 뜻한다.
- 관계는 관계에 참여하는 개체 타입의 개수에 대한 차수(Degree)와 관계에 참여하는 개체 어커런스의 개수에 대한 대응 카디널리티(Mapping Cardinality)를 갖는다.
- 차수에 따른 관계의 종류
  - 단항(Unary) 관계 : 관계에 참여하고 있는 개체 타입이 1개인 관계
  - 이항(Binary) 관계 : 관계에 참여하고 있는 개체 타입이 2개인 관계
  - 삼항(Ternary) 관계 : 관계에 참여하고 있는 개체 타입이 3개인 관계
  - n항(n-ary) 관계 : 관계에 참여하고 있는 개체 타입이 n개인 관계
- 대응 카디널리티에 따른 관계의 종류
  - 1:1 관계 : 관계에 참여하고 있는 두 개체 타입이 모두 하나씩의 개체 어커런스를 갖는 관계
  - 1:N 관계 : 관계에 참여하고 있는 개체 타입 중 한 개체 타입은 여러 개의 개체 어커런스를 가질 수 있고, 다른 한 개체 타입은 하나의 개체 어커런스를 갖는 관계
  - N:M 관계 : 관계에 참여하고 있는 두 개체 타입 모두 여러 개의 개체 어커런스를 가질 수 있는 관계

1. 관계는 2개 이상의 개체 사이에 존재하는 연관성을 의미하는 것으로, 다음은 대응 카디널리티에 따른 관계의 종류에 대한 설명이다. 괄호(①~③)에 들어갈 알맞은 관계를 쓰시오.

|       |                                                                                        |
|-------|----------------------------------------------------------------------------------------|
| ( ① ) | 관계에 참여하고 있는 두 개체 타입 모두 여러 개의 개체 어커런스를 가질 수 있는 관계이다.                                    |
| ( ② ) | 관계에 참여하고 있는 개체 타입 중 한 개체 타입은 여러 개의 개체 어커런스를 가질 수 있고, 다른 한 개체 타입은 하나의 개체 어커런스를 갖는 관계이다. |
| ( ③ ) | 관계에 참여하고 있는 두 개체 타입이 모두 하나씩의 개체 어커런스를 갖는 관계이다.                                         |

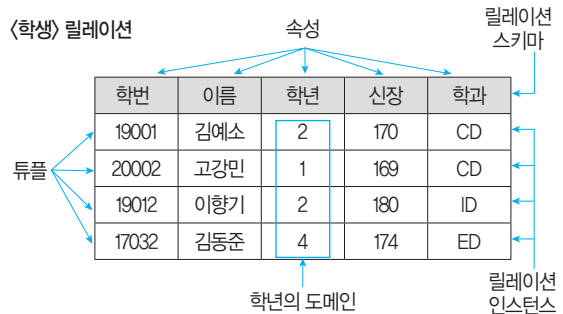
답

- ① :  
② :  
③ :

정답 1. ① N:M 관계 ② 1:N 관계 ③ 1:1 관계

### [핵심 158] 관계 데이터베이스의 Relation 구조

릴레이션은 데이터들을 표(Table)의 형태로 표현한 것으로, 구조를 나타내는 릴레이션 스키마와 실제 값들인 릴레이션 인스턴스로 구성된다.



#### 튜플(Tuple)

- 릴레이션을 구성하는 각각의 행이다.
- 속성의 모임으로 구성된다.
- 파일 구조에서 레코드(Record)와 같은 의미이다.
- 튜플의 수 = 카디널리티(Cardinality) = 기수 = 대응수

#### 속성(Attribute)

- 릴레이션을 구성하는 각각의 열이다.
- 데이터베이스를 구성하는 가장 작은 논리적 단위이다.
- 파일 구조 상의 데이터 항목 또는 데이터 필드에 해당된다.
- 개체의 특성을 기술한다.
- 속성의 수 = 디그리(Degree) = 차수

### 도메인(Domain)

- 하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자(Atomic)값들의 집합이다.
- 실제 애트리뷰트 값이 나타날 때 그 값의 합법 여부를 시스템이 검사하는 데에도 이용된다.

### 릴레이션 인스턴스(Relation Instance)

데이터 개체를 구성하고 있는 속성들에 데이터 타입이 정의되어 구체적인 데이터 값을 갖고 있는 것을 말한다.

2020년 2회 기공사 실기

1. 다음 <사원> 릴레이션에 대한 각 물음(①~④)에 답하시오.  
<사원>

| 사번    | 부서명  | 직급 | 호봉 | 기본급       |
|-------|------|----|----|-----------|
| SA001 | 총무부  | 과장 | 5  | 3,600,000 |
| SA002 | 기획부  | 과장 | 3  | 3,200,000 |
| SA003 | 생산부  | 대리 | 3  | 2,700,000 |
| SA004 | 영업부  | 대리 | 2  | 2,600,000 |
| SA005 | 마케팅부 | 사원 | 2  | 2,200,000 |

- ① 카디널리티(Cardinality) 수와 디그리(Degree) 수는 각각 몇 개인가?
- ② 데이터베이스를 구성하는 가장 작은 논리적 단위로, <사원> 릴레이션의 '사번', '부서명', '직급', '호봉', '기본급' 각 열을 의미하는 것은 무엇인가?
- ③ 릴레이션을 구성하는 각각의 행으로, <사원> 릴레이션의 'SA001', '총무부', '과장', '5', '3,600,000' 한 행을 의미하는 것은 무엇인가?
- ④ <사원> 릴레이션에는 표시되지 않았으나 '호봉'에 1~6의 숫자 값만 저장하도록 제한했다. 이처럼 릴레이션에서 하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자(Atomic) 값들의 집합을 무엇이라 하는가?

답

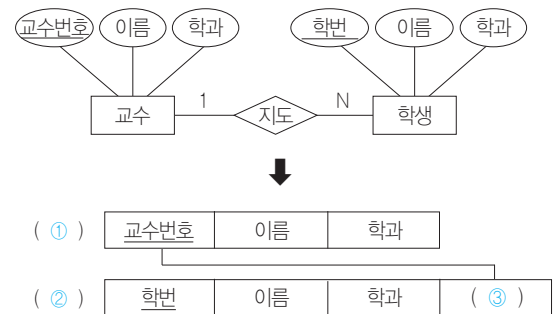
- ① :
- ② :
- ③ :
- ④ :

정답 1. ① 5, 5    ② 속성(Attribute)    ③ 튜플(Tuple)    ④ 도메인(Domain)

### [핵심 159] ER 모델을 관계형 데이터 모델로 변환

- 개념적 데이터 모델인 ER 모델을 논리적 데이터 모델인 릴레이션 스키마로 변환하는 것으로, 매핑 룰(Mapping Rule)이라고도 한다.
- 개체 A, B와 관계 Y로 이루어진 ER 모델을 관계형 데이터 모델에서의 릴레이션 스키마로 변환하는 과정은 다음과 같다.
  - 개체 A, B는 각각 독립적인 릴레이션 A, B로 표현된다. 또한, ER 도형에서의 각 개체의 속성들은 각 릴레이션의 속성들로 정의되고, 기본키는 밑줄을 그어 표시한다.
  - 관계 Y가 1:1 관계 : 릴레이션 A의 기본키를 릴레이션 B의 외래키로 추가하거나 릴레이션 B의 기본키를 릴레이션 A의 외래키로 추가하여 표현함
  - 관계 Y가 1:N 관계 : 릴레이션 A의 기본키를 릴레이션 B의 외래키로 추가하여 표현함
  - 관계 Y가 N:M 관계 : 릴레이션 A와 B의 기본키를 모두 포함한 별도의 릴레이션으로 표현함

1. <교수>, <학생> 개체 간의 1:N 관계를 표현한 다음의 ER 도형을 릴레이션 스키마로 변환한 것이다. 괄호(①~③)에 들어갈 알맞은 용어를 쓰시오.



답

- ① :
- ② :
- ③ :

**해설** 1. 교수 개체와 학생 개체는 1:N의 지도 관계를 가지므로 <교수> 릴레이션의 기본키인 '교수번호'를 <학생> 릴레이션에 외래키로 추가해야 합니다.

정답 1. ① 교수 ② 학생 ③ 교수번호

### [핵심160] 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

2020년 1, 2회 기사 필기

1. 다음 두 릴레이션에서 외래키(Foreign Key)로 사용된 것은 무엇인지 쓰시오(밑줄 친 속성은 기본키임).

과목(과목번호, 과목명)  
수강(수강번호, 학번, 과목번호, 학기)

답 :

2020년 1, 2회 기사 필기

2. 다음은 후보키(Candidate Key)에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오.

- 후보키는 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용되는 속성들의 부분집합이다.
- 후보키는 릴레이션에 있는 모든 튜플에 대해 ( ① )과 ( ② )을 모두 만족시켜야 한다.

답

① :

② :

해설 1. <수강> 릴레이션의 '과목번호'는 <과목> 릴레이션의 기본키인 '과목번호'를 참조하고 있으므로 <수강> 릴레이션에서 '과목번호'는 외래키가 됩니다.

정답 1. 과목번호 2. ① 유일성(Unique) ② 최소성(Minimality)



### [핵심161] 무결성(Integrity)

- 데이터베이스에 저장된 데이터 값과 그것이 표현하는 현실 세계의 실제값이 일치하는 정확성을 의미한다.
- 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 함. 즉 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다는 규정
- 사용자 정의 무결성(User-Defined Integrity) : 속성 값들이 사용자가 정의한 제약조건에 만족해야 한다는 규정
- NULL 무결성 : 릴레이션의 특정 속성 값이 NULL이 될 수 없도록 하는 규정
- 고유(Unique) 무결성 : 릴레이션의 특정 속성에 대해 각 튜플이 갖는 속성값들이 서로 달라야 한다는 규정
- 키(Key) 무결성 : 하나의 릴레이션에는 적어도 하나의 키가 존재해야 한다는 규정
- 관계(Relationship) 무결성 : 릴레이션에 어느 한 튜플의 삽입 가능 여부 또는 한 릴레이션과 다른 릴레이션의 튜플들 사이의 관계에 대한 적절성 여부를 지정한 규정

2020년 1, 2회 기사 필기

1. 데이터 무결성 제약 조건 중 '개체 무결성 제약' 조건에 대해 설명하시오.

답 :

2. 다음은 무결성(Integrity)의 종류별 특징이다. 괄호(①, ②)에 들어갈 알맞은 무결성 종류를 쓰시오.

|             |                                                                              |
|-------------|------------------------------------------------------------------------------|
| NULL<br>무결성 | 릴레이션의 특정 속성 값이 NULL이 될 수 없도록 하는 규정이다.                                        |
| ( ① )       | 외래키 값은 Null이거나 참조 릴레이션의 기본키 값과 동일해야 한다.                                      |
| 관계<br>무결성   | 릴레이션에 어느 한 튜플의 삽입 가능 여부 또는 한 릴레이션과 다른 릴레이션의 튜플들 사이의 관계에 대한 적절성 여부를 지정한 규정이다. |
| ( ② )       | 릴레이션의 특정 속성에 대해 각 튜플이 갖는 속성값들이 서로 달라야 한다는 규정이다.                              |
| 키<br>무결성    | 하나의 릴레이션에는 적어도 하나의 키가 존재해야 한다는 규정이다.                                         |

답

① :

② :

**정답** 1. 개체 무결성 제약 조건은 기본 테이블의 기본키를 구성하는 어떤 속성도 Null 값이나 중복값을 가질 수 없다는 규정이다.

2. ① 참조 무결성(Referential Integrity) ② 고유(UUnique) 무결성

### [핵심162] 네트워크(Network)

두 대 이상의 컴퓨터를 전화선이나 케이블 등으로 연결하여 자원을 공유하는 것을 말한다.

#### 근거리 통신망(LAN; Local Area Network)

- 회사, 학교, 연구소 등에서 비교적 가까운 거리에 있는 컴퓨터, 프린터, 저장장치 등과 같은 자원을 연결하여 구성한다.
- 사이트 간의 거리가 짧아 데이터의 전송 속도가 빠르고, 에러 발생률이 낮다.
- 근거리 통신망에서는 주로 버스형이나 링형 구조를 사용한다.

#### 광역 통신망(WAN; Wide Area Network)

- 국가와 국가 혹은 대륙과 대륙 등과 같이 멀리 떨어진 사이트들을 연결하여 구성한다.
- 사이트 간의 거리가 멀기 때문에 통신 속도가 느리고, 에러 발생률이 높다.

1. 다음은 네트워크(Network)에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오.

- 네트워크는 두 대 이상의 컴퓨터를 전화선이나 케이블 등으로 연결하여 자원을 공유하는 것을 말하며, 각 사이트들이 분포되어 있는 지리적 범위에 따라 ( ① )과 ( ② )으로 분류된다.
- ( ① )은 회사, 학교, 연구소 등에서 비교적 가까운 거리에 있는 컴퓨터, 프린터, 저장장치 등과 같은 자원을 연결하여 구성하며, ( ② )은 국가와 국가 혹은 대륙과 대륙 등과 같이 멀리 떨어진 사이트들을 연결하여 구성한다.

답

① :

② :

**정답** 1. ① 근거리 통신망(LAN; Local Area Network)  
② 광대역 통신망(WAN; Wide Area Network)

### [핵심163] IP 주소(Internet Protocol Address)

- 인터넷에 연결된 모든 컴퓨터 자원을 구분하기 위한 고유한 주소이다.
- 숫자로 8비트씩 4부분, 총 32비트로 구성되어 있다.
- 네트워크 부분의 길이에 따라 다음과 같이 A~E 클래스까지 총 5단계로 구성되어 있다.

|         |                             |
|---------|-----------------------------|
| Class A | 국가나 대형 통신망(16,777,216개 호스트) |
| Class B | 중대형 통신망(65,536개 호스트)        |
| Class C | 소규모 통신망(256개 호스트)           |
| Class D | 멀티캐스트용                      |
| Class E | 실험용으로 공용되지 않음               |

- 서브넷 마스크 : 4바이트의 IP 주소 중 네트워크 주소와 호스트 주소를 구분하기 위한 비트

1. IP 주소(Internet Protocol Address)의 개념을 설명하시오.

답 :

**정답** 1. IP 주소는 인터넷에 연결된 모든 컴퓨터 자원을 구분하기 위한 고유한 주소이다.

### [핵심 164] 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)

2020년 1, 2회 기사 필기

1. 다음이 설명하는 것은 무엇인지 쓰시오.

- 현재 사용하고 있는 IP 주소 체계의 주소 부족 문제를 해결하기 위해 개발되었다.
- IPv6은 16비트씩 8부분, 총 128비트로 구성되어 있다.
- 각 부분을 16진수로 표현하고, 콜론(:)으로 구분한다.
- 확장 헤더를 통해 네트워크 기능 확장이 용이하다.
- 인증 및 보안 기능을 포함하고 있다.

답 :

2020년 1, 2회 기사 필기

2. IPv6(Internet Protocol version 6)의 세 가지 주소 체계를 쓰시오.

답 :

**정답** 1. IPv6(Internet Protocol version 6)

2. 유니캐스트(Unicast), 멀티캐스트(Multicast), 애니캐스트(Anycast)

### [핵심 165] 도메인 네임(Domain Name)

- 숫자로 된 IP 주소를 사람이 이해하기 쉬운 문자 형태로 표현한 것이다.
- 호스트 컴퓨터 이름, 소속 기관 이름, 소속 기관의 종류, 소속 국가명 순으로 구성되며, 왼쪽에서 오른쪽으로 갈수록 상위 도메인을 의미한다.
- 문자로 된 도메인 네임을 컴퓨터가 이해할 수 있는 IP 주소로 변환하는 역할을 하는 시스템을 DNS(Domain Name System)라고 하며 이런 역할을 하는 서버를 DNS 서버라고 한다.

1. 도메인 네임은 숫자로 된 IP 주소를 사람이 이해하기 쉬운 문자 형태로 표현한 것이다. 문자로 된 도메인 네임을 컴퓨터가 이해할 수 있는 IP 주소로 변환하는 역할을 하는 시스템은 무엇인지 쓰시오.

답 :

**정답** 1. DNS(Domain Name System)

### [핵심 166] OSI(Open System Interconnection) 참조 모델

- 다른 시스템 간의 원활한 통신을 위해 ISO(국제표준화 기구)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

## 2020년 1, 2회 기사 필기

1. OSI 참조 모델은 다른 시스템 간의 원활한 통신을 위해 ISO에서 제안한 통신 규약으로, 개방형 시스템 간의 데이터 통신 시 필요한 장비 및 처리 방법 등을 다음과 같이 7단계로 표준화하여 규정했다. 괄호(①, ②)에 들어갈 알맞은 계층을 쓰시오.

|           |                                                                                          |
|-----------|------------------------------------------------------------------------------------------|
| 데이터 링크 계층 | 두 개의 인접한 개방 시스템들 간에 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 하고, 흐름 제어, 프레임 동기화, 오류 제어, 순서 제어 등을 수행한다. |
| ( ① )     | 종단간 신뢰성 있고 효율적인 데이터를 전송하기 위해 오류 검출과 복구, 흐름 제어 등을 수행한다.                                   |
| ( ② )     | 송·수신 측 간의 관련성을 유지하고 대화 제어를 담당하고, 대화(회화) 구성 및 동기 제어, 데이터 교환 관리 기능을 수행한다.                  |
| 표현 계층     | 응용 계층으로부터 받은 데이터를 세션 계층에 맞게, 세션 계층에서 받은 데이터는 응용 계층에 맞게 변환한다.                             |

|         |                                                                                             |
|---------|---------------------------------------------------------------------------------------------|
| 응용 계층   | 사용자가 OSI 환경에 접근할 수 있도록 서비스를 제공한다.                                                           |
| ( ③ )   | 다양한 전송매체를 통해 비트 스트림을 전송하고, 전송에 필요한 두 장치 간의 실제 접속과 절단 등 기계적, 전기적, 기능적, 절차적 특성에 대한 규칙을 정의한다.  |
| 네트워크 계층 | 개방 시스템들 간의 네트워크 연결을 관리하는 기능과 데이터의 교환 및 중계 기능을 하고, 경로 설정(Routing), 트래픽 제어, 패킷 정보 전송 등을 수행한다. |

## 답

- ① :  
② :  
③ :

정답 1. ① 전송 계층(Transport Layer) ② 세션 계층(Session Layer)  
③ 물리 계층(Physical Layer)

## [핵심 167] 네트워크 관련 장비

## • 허브(Hub)

- 한 사무실이나 가까운 거리의 컴퓨터들을 연결하는 장치로, 각 회선을 통합적으로 관리하며, 신호 증폭 기능을 하는 리피터의 역할도 포함함
- 더미 허브(Dummy Hub) : 네트워크에 흐르는 모든 데이터를 단순히 연결하는 기능만을 제공함
- 스위칭 허브(Switching Hub) : 네트워크상에 흐르는 데이터의 유무 및 흐름을 제어하여 각각의 노드가 허브의 최대 대역폭을 사용할 수 있는 지능형 허브임

- 리피터(Repeater) : 물리 계층의 장비로, 전송되는 신호가 왜곡되거나 약해질 경우 원래의 신호 형태로 재생함
- 브리지(Bridge) : 데이터 링크 계층의 장비로, LAN과 LAN을 연결하거나 LAN 안에서의 컴퓨터 그룹을 연결함
- 라우터(Router) : 네트워크 계층의 장비로, LAN과 LAN의 연결 및 경로 선택, 서로 다른 LAN이나 LAN과 WAN을 연결함



- 게이트웨이(Gateway) : 전 계층(1~7계층)의 프로토콜 구조가 전혀 다른 네트워크의 연결을 수행함
- 스위치(Switch) : 브리지와 같이 LAN과 LAN을 연결하여 훨씬 더 큰 LAN을 만드는 장치

#### 잠깐만요 ① 스위치 분류

- L2 스위치 : OSI의 2계층에 속하며, MAC 주소를 기반으로 프레임을 전송함
- L3 스위치 : OSI의 3계층에 속하며, L2 스위치에 라우터 기능이 추가되었고, IP 주소를 기반으로 패킷을 전송함
- L4 스위치 : OSI 4계층에 속하며, IP 주소 및 TCP/UDP를 기반으로 사용자들의 요구를 서버의 부하가 적은 곳에 배분하는 로드밸런싱 기능을 제공함
- L7 스위치 : OSI 7계층에 속하며, IP 주소, TCP/UDP 포트 정보에 패킷 내용까지 참조하여 세밀하게 로드밸런싱함

1. 프로토콜 구조가 전혀 다른 네트워크의 연결을 수행하는 장비로, 세션 계층, 표현 계층, 응용 계층 간을 연결하여 데이터 형식 변환, 주소 변환, 프로토콜 변환 등을 수행하는 네트워크 장비를 쓰시오.

답 :

2. OSI 참조 모델의 네트워크 계층에서 동작하는 장비로, 브리지와 같이 LAN과 LAN의 연결 기능에 데이터 전송의 최적 경로를 선택할 수 있는 기능이 추가되었으며, 서로 다른 LAN이나 LAN과 WAN의 연결도 수행하는 네트워크 장비를 쓰시오.

답 :

정답 1. 게이트웨이(Gateway) 2. 라우터(Router)

### [핵심 168] TCP/IP(Transmission Control Protocol/Internet Protocol)

인터넷에 연결된 서로 다른 기종의 컴퓨터들이 데이터를 주고받을 수 있도록 하는 표준 프로토콜이다.

#### TCP(Transmission Control Protocol)

- 신뢰성 있는 연결형 서비스를 제공한다.
- 패킷의 다중화, 순서 제어, 오류 제어, 흐름 제어 기능을 제공한다.

- 스트림(Stream) 전송 기능을 제공한다.

#### IP(Internet Protocol)

- 데이터그램을 기반으로 하는 비연결형 서비스를 제공한다.
- 패킷의 분해/조립, 주소 지정, 경로 선택 기능을 제공한다.
- 헤더의 길이는 최소 20Byte에서 최대 60Byte이다.

1. 인터넷에 연결된 서로 다른 기종의 컴퓨터들이 데이터를 주고받을 수 있도록 하는 표준 프로토콜은 무엇인지 쓰시오.

답 :

2. 다음이 설명하는 인터넷 표준 프로토콜을 쓰시오.

- 패킷의 분해/조립, 주소 지정, 경로 선택 기능을 제공한다.
- 비연결형인 데이터그램 방식을 사용하는 것으로 신뢰성이 보장되지 않는다.
- 헤더의 길이는 최소 20Byte에서 최대 60Byte이다.

답 :

정답 1. TCP/IP(Transmission Control Protocol/Internet Protocol)  
2. IP(Internet Protocol)

### [핵심 169] 프로토콜(Protocol)

- 서로 다른 기기들 간의 데이터 교환을 원활하게 수행할 수 있도록 표준화시켜 놓은 통신 규약이다.
- 프로토콜의 기본 요소

|               |                                                         |
|---------------|---------------------------------------------------------|
| 구문(Syntax)    | 전송하고자 하는 데이터의 형식, 부호화, 신호 레벨 등을 규정함                     |
| 의미(Semantics) | 두 기기 간의 효율적이고 정확한 정보 전송을 위한 협조 사항과 오류 관리를 위한 제어 정보를 규정함 |
| 시간(Timing)    | 두 기기 간의 통신 속도, 메시지의 순서 제어 등을 규정함                        |

## 2020년 1회 기사 실기

1. 프로토콜은 서로 다른 기기들 간의 데이터 교환을 원활하게 수행할 수 있도록 표준화시켜 놓은 통신 규약이다. 프로토콜의 기본 요소 3가지를 쓰시오.

답 :

정답 1. 구문(Syntax), 의미(Semantics), 시간(Timing)

## [핵심170] TCP/IP의 구조

| OSI                     | TCP/IP      | 기능                                                                                                                                   |
|-------------------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------|
| 응용 계층<br>표현 계층<br>세션 계층 | 응용 계층       | <ul style="list-style-type: none"> <li>• 응용 프로그램 간의 데이터 송·수신을 제공</li> <li>• TELNET, FTP, SMTP, SNMP, DNS, HTTP 등</li> </ul>          |
| 전송 계층                   | 전송 계층       | <ul style="list-style-type: none"> <li>• 호스트들 간의 신뢰성 있는 통신을 제공</li> <li>• TCP, UDP</li> </ul>                                        |
| 네트워크 계층                 | 인터넷 계층      | <ul style="list-style-type: none"> <li>• 데이터 전송을 위한 주소 지정, 경로 설정을 제공</li> <li>• IP, ICMP, IGMP, ARP, RARP</li> </ul>                 |
| 데이터 링크 계층<br>물리 계층      | 네트워크 액세스 계층 | <ul style="list-style-type: none"> <li>• 실제 데이터(프레임)를 송·수신하는 역할</li> <li>• Ethernet, IEEE 802, HDLC, X.25, RS-232C, ARQ 등</li> </ul> |

## 2020년 1회 기능사 실기

1. 다음은 TCP/IP 계층을 나타낸 것이다. 괄호에 들어갈 알맞은 계층을 쓰시오.

|             |
|-------------|
| 응용 계층       |
| (      )    |
| 인터넷 계층      |
| 네트워크 액세스 계층 |

답 :

정답 1. 전송 계층

## [핵심171] TCP/IP의 응용 계층 프로토콜

- FTP(File Transfer Protocol) : 컴퓨터와 컴퓨터 또는 컴퓨터와 인터넷 사이에서 파일을 주고받을 수 있도록 하는 원격 파일 전송 프로토콜
- SMTP(Simple Mail Transfer Protocol) : 전자 우편을 교환하는 서비스
- TELNET : 멀리 떨어져 있는 컴퓨터에 접속하여 자신의 컴퓨터처럼 사용할 수 있도록 해주는 서비스
- SNMP(Simple Network Management Protocol) : TCP/IP의 네트워크 관리 프로토콜로, 라우터나 허브 등 네트워크 기기의 네트워크 정보를 네트워크 관리 시스템에 보내는 데 사용되는 표준 통신 규약
- DNS(Domain Name System) : 도메인 이름을 IP 주소로 매핑(Mapping)하는 시스템
- HTTP(HyperText Transfer Protocol) : 월드 와이드 웹(WWW)에서 HTML 문서를 송수신 하기 위한 표준 프로토콜

## 2020년 2회 기능사 실기

1. 웹 서버와 웹 클라이언트 사이에서 정보를 주고 받기 위한 표준 프로토콜을 쓰시오.

답 :

2. TCP/IP 응용 계층의 주요 프로토콜로, 컴퓨터와 컴퓨터 또는 컴퓨터와 인터넷 사이에서 파일을 주고받을 수 있도록 하는 원격 파일 전송 프로토콜을 쓰시오.

답 :

정답 1. HTTP(HyperText Transfer Protocol)

2. FTP(File Transfer Protocol)

## [핵심172] TCP/IP의 전송 계층 프로토콜

- TCP(Transmission Control Protocol)
  - 양방향 연결(Full Duplex Connection)형 서비스를 제공한다.
  - 가상 회선 연결(Virtual Circuit Connection) 형태의 서비스를 제공한다.
  - 스트림 위주의 전달(패킷 단위)을 한다.
  - 신뢰성 있는 경로를 확립하고 메시지 전송을 감독한다.



- 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.
- RTCP(Real-Time Control Protocol)
  - RTP(Real-time Transport Protocol) 패킷의 전송 품질을 제어하기 위한 제어 프로토콜이다.
  - 세션(Session)에 참여한 각 참여자들에게 주기적으로 제어 정보를 전송한다.
  - 하위 프로토콜은 데이터 패킷과 제어 패킷의 다중화(Multiplexing)를 제공한다.

2020년 1회 기공사 실기

1. OSI 7계층 중 TCP(Transmission Control Protocol)와 UDP(User Datagram Protocol)가 속한 계층을 쓰시오.

답 :

2. 다음이 설명하는 TCP/IP의 전송 계층 프로토콜을 쓰시오.

- 데이터 전송 전에 연결을 설정하지 않는 비연결형 서비스를 제공한다.
- 데이터를 전송만 할 뿐 확인 과정을 거치지 않기 때문에 신뢰성보다는 빠른 데이터 전송 작업에 적합하다.

답 :

정답 1. 전송 계층(Transport Layer) 2. UDP(User Datagram Protocol)

### [핵심173] TCP/IP의 인터넷 계층 프로토콜

- IP(Internet Protocol) : 전송할 데이터에 주소 지정 및 경로 설정 등의 기능을 하며, 비연결형인 데이터그램 방식을 사용하므로 신뢰성이 보장되지 않음
- ICMP(Internet Control Message Protocol) : IP와 조합하여 통신중에 발생하는 오류의 처리와 전송 경로 변경 등을 위한 제어 메시지를 관리하는 역할을 하며, 헤더는 8Byte로 구성됨
- IGMP(Internet Group Management Protocol) : 멀티캐스트를 지원하는 호스트나 라우터 사이에서 멀티캐스트 그룹 유지를 위해 사용됨
- ARP(Address Resolution Protocol) : 호스트의 IP 주소를 호스트와 연결된 네트워크 접속 장치의 물리적 주소(MAC Address)로 바꿈
- RARP(Reverse Address Resolution Protocol) : ARP와 반대로 물리적 주소를 IP 주소로 변환하는 기능을 함

2020년 1,2회 기사 필기

1. TCP/IP 네트워크에서 호스트의 IP 주소를 호스트와 연결된 네트워크 접속 장치의 MAC Address로 변환하는 프로토콜을 쓰시오.

답 :

2. IP 프로토콜과 조합하여 통신중에 발생하는 오류의 처리와 전송 경로 변경 등을 위한 제어 메시지를 관리하는 역할을 하는 TCP/IP의 인터넷 계층 프로토콜을 쓰시오.

답 :

정답 1. ARP(Address Resolution Protocol)

2. ICMP(Internet Control Message Protocol)

### [핵심174] TCP/IP의 네트워크 액세스 계층 프로토콜

- Ethernet(IEEE 802.3) : CSMA/CD 방식의 LAN
- IEEE 802 : LAN을 위한 표준 프로토콜
- HDLC : 비트 위주의 데이터 링크 제어 프로토콜
- X.25 : 패킷 교환망을 통한 DTE와 DCE 간의 인터페이스를 제공하는 프로토콜
- RS-232C : 공중 전화 교환망(PSTN)을 통한 DTE와 DCE 간의 인터페이스를 제공하는 프로토콜

1. TCP/IP 네트워크에서 패킷 교환망을 통한 DTE와 DCE 간의 인터페이스를 제공하는 프로토콜을 쓰시오.

답 :

2. TCP/IP 네트워크에서 LAN을 위한 표준 프로토콜을 쓰시오.

답 :

정답 1. X.25 2. IEEE 802

### [핵심175] 회선 교환 방식(Circuit Switching)

- 통신을 원하는 두 지점을 교환기를 이용하여 물리적으로 접속시키는 방식으로, 기존의 음성 전화망이 대표적이다.
- 접속에는 긴 시간이 소요되나 일단 접속되면 전송 지연이 거의 없어 실시간 전송이 가능하다.
- 데이터 전송에 필요한 전체 시간이 축적 교환 방식에 비해 길다.
- 일정한 데이터 전송률을 제공하므로 동일한 전송 속도가 유지된다.
- 회선 교환 방식의 종류
  - 공간 분할 교환 방식(SDS; Space Division Switching) : 기계식 접점과 전자 교환기의 전자식 접점 등을 이용하여 교환을 수행하는 방식으로, 음성 전화용 교환기기에 속함
  - 시분할 교환 방식(TDS; Time Division Switching) : 전자 부품이 갖는 고속성과 디지털 교환 기술을 이용하여 다수의 디지털 신호를 시분할적으로 동작시켜 다중화하는 방식

1. 다음이 설명하는 데이터 교환 방식을 쓰시오.

- 통신을 원하는 두 지점을 교환기를 이용하여 물리적으로 접속시키는 방식으로, 기존의 음성 전화망이 대표적이다.
- 기계식 접점과 전자 교환기의 전자식 접점 등을 이용하여 교환을 수행하는 방식인 공간 분할 교환 방식과 전자 부품이 갖는 고속성과 디지털 교환 기술을 이용하여 다수의 디지털 신호를 시분할적으로 동작시켜 다중화하는 시분할 교환 방식이 있다.

답 :

정답 1. 회선 교환 방식(Circuit Switching)

### [핵심176] 패킷 교환 방식(Packet Switching)

- 메시지를 일정한 길이의 패킷으로 잘라서 전송하는 방식이다.
- 패킷은 장애 발생 시의 재전송을 위해 패킷 교환기에 일시 저장되었다가 곧 전송되며 전송이 끝난 후 폐기된다.
- 전송 시 교환기, 회선 등에 장애가 발생하더라도 다른 정상적인 경로를 선택해서 우회할 수 있다.
- 음성 전송보다 데이터 전송에 더 적합하다.
- 패킷 교환망은 OSI 7계층의 네트워크 계층에 해당한다.
- 패킷 교환 방식의 종류
  - 가상 회선 방식 : 단말장치 상호간에 논리적인 가상 통신 회선을 미리 설정하여 송신지와 수신지 사이의 연결을 확립한 후에 설정된 경로를 따라 패킷들을 순서적으로 운반하는 방식
  - 데이터그램 방식 : 연결 경로를 설정하지 않고 인접한 노드들의 트래픽(전송량) 상황을 감안하여 각각의 패킷들을 순서에 상관없이 독립적으로 운반하는 방식

1. 다음이 설명하는 데이터 교환 방식을 쓰시오.

- 메시지를 일정한 길이의 패킷으로 잘라서 전송하는 방식으로, 전송 시 교환기, 회선 등에 장애가 발생하더라도 다른 정상적인 경로를 선택해서 우회할 수 있다.
- 이 교환 방식의 종류로는 가상 회선 방식과 데이터그램 방식이 있다.

답 :

정답 1. 패킷 교환 방식(Packet Switching)

### [핵심177] 라우팅(Routing, 경로 제어)

- 송·수신 측 간의 전송 경로 중에서 최적 패킷 교환 경로를 결정하는 기능이다.
- 경로 제어표(Routing Table)를 참조해서 이루어지며, 라우터에 의해 수행된다.

## • 라우팅 프로토콜

- RIP(Routing Information Protocol) : 현재 가장 널리 사용되는 라우팅 프로토콜로, 소규모 동종의 네트워크 내에서 효율적인 방법이며, 최대 홉수를 15로 제한함
- IGRP(Interior Gateway Routing Protocol) : RIP의 단점을 보완하기 위해 만들어 개발된 것으로, 네트워크 상태를 고려하여 라우팅하며, 중규모 네트워크에 적합함
- OSPF(Open Shortest Path First Protocol) : 대규모 네트워크에서 많이 사용되는 라우팅 프로토콜로, 라우팅 정보에 변화가 생길 경우 변화된 정보만 네트워크 내의 모든 라우터에 알리며, RIP에 비해 홉수에 제한이 없음
- BGP(Border Gateway Protocol) : 자율 시스템(AS) 간의 라우팅 프로토콜로, EGP의 단점을 보완하기 위해 개발되었음

## • 라우팅 알고리즘

- 거리 벡터 알고리즘(Distance Vector Algorithm) : 인접해 있는 라우터 간의 거리(Distance)와 방향(Vector)에 대한 정보를 이용하여 최적의 경로를 찾고 그 최적 경로를 이용할 수 없을 경우 다른 경로를 찾는 알고리즘으로, RIP와 IGRP가 있음
- 링크 상태 알고리즘(Link State Algorithm) : 라우터와 라우터 간의 모든 경로를 파악하여 미리 대체 경로를 마련해 두는 알고리즘으로, 거리 벡터 알고리즘의 단점을 보완하기 위해 개발되었으며, OSPF가 있음

2020년 1, 2회 기사 필기

1. 소규모 동종의 네트워크 내에서 효율적인 방법이며, 최대 홉수를 15로 제한한 현재 가장 널리 사용되는 라우팅 프로토콜을 쓰시오.

답 :

2. 인접해 있는 라우터 간의 Distance와 Vector에 대한 정보를 이용하여 최적의 경로를 찾고 그 최적 경로를 이용할 수 없을 경우 다른 경로를 찾는 라우팅 알고리즘을 쓰시오.

답 :

정답 1. RIP(Routing Information Protocol)

2. 거리 벡터 알고리즘(Distance Vector Algorithm)

## 11장 | 제품 소프트웨어 패키징

### [핵심178] 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.
- 소프트웨어 패키징 작업 순서 : 기능 식별 → 모듈화(Modularity) → 빌드(Build) 진행 → 사용자 환경 분석 → 패키징 및 적용 시험 → 패키징 변경 개선 → 배포

1. 다음은 소프트웨어 패키징 시 수행하는 작업들이다. 작업 순서에 맞게 기호(㉠~㉤)를 나열하시오.

- |                   |             |
|-------------------|-------------|
| ㉠ 모듈화(Modularity) | ㉣ 패키징 변경 개선 |
| ㉡ 빌드(Build) 진행    | ㉢ 기능 식별     |
| ㉢ 패키징 및 적용 시험     | ㉤ 사용자 환경 분석 |
| ㉣ 배포              |             |

답 :

정답 1. ㉢, ㉠, ㉡, ㉢, ㉢, ㉢, ㉣, ㉤

### [핵심179] 릴리즈 노트(Release Note)

- 개발 과정에서 정리된 릴리즈 정보를 소프트웨어의 최종 사용자인 고객과 공유하기 위한 문서이다.
- 릴리즈 노트 작성 순서
  - 1 모듈 식별 : 모듈별 빌드 수행 후 릴리즈 노트에 작성될 내용 확인
  - 2 릴리즈 정보 확인 : 릴리즈 노트 및 소프트웨어 이름, 릴리즈 버전 및 날짜, 노트 날짜 및 버전 등 확인
  - 3 릴리즈 노트 개요 작성 : 소프트웨어 및 변경사항 전체에 대한 간략한 내용 작성
  - 4 영향도 체크 : 버그나 이슈 관련 내용 또는 해당 릴리즈 버전에서의 기능 변화가 다른 소프트웨어나 기능을 사용하는데 미칠 수 있는 영향 기술
  - 5 정식 릴리즈 노트 작성 : Header(머릿말), 개요, 영향도 체크 항목을 포함하여 정식 릴리즈 노트에 작성될 기본 사항 작성

- ⑥ 추가 개선 항목 식별 : 추가 버전 릴리즈 노트 작성이 필요한 경우 추가 릴리즈 노트 작성

※ Header(머릿말) : 릴리즈 노트 이름, 소프트웨어 이름, 릴리즈 버전, 릴리즈 날짜, 릴리즈 노트 날짜, 릴리즈 노트 버전 등을 표시함

2020년 1회 기사 실기

1. 릴리즈 노트는 개발 과정에서 정리된 릴리즈 정보를 소프트웨어의 최종 사용자인 고객과 공유하기 위한 문서이다. 릴리즈 노트 작성 시 릴리즈 노트 이름, 소프트웨어 이름, 릴리즈 버전, 릴리즈 날짜, 릴리즈 노트 날짜, 릴리즈 노트 버전 등이 포함된 항목을 쓰시오.

답 :

2. 다음은 릴리즈 노트의 작성 순서를 나열한 것이다. 괄호(①, ②)에 들어갈 알맞은 작업을 쓰시오.

모듈 식별 → ( ① ) → 릴리즈 노트 개요 작성 → 영향도 체크 → ( ② ) → 추가 개선 항목 식별

답

① :

② :

정답 1. Header(머릿말)

2. ① 릴리즈 정보 확인 ② 정식 릴리즈 노트 작성

### [핵심180] 디지털 저작권 관리(DRM)

- 저작권자가 배포한 디지털 콘텐츠가 저작권자가 의도한 용도로만 사용되도록 디지털 콘텐츠의 생성, 유통, 이용까지의 전 과정에 걸쳐 사용되는 디지털 콘텐츠 관리 및 보호 기술이다.
- 디지털 저작권 관리의 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

1. 저작권자가 배포한 디지털 콘텐츠가 저작권자가 의도한 용도로만 사용되도록 디지털 콘텐츠의 생성, 유통, 이용까지의 전 과정에 걸쳐 사용되는 디지털 콘텐츠 관리 및 보호 기술을 쓰시오.

답 :

정답 1. 디지털 저작권 관리(DRM)

### [핵심181] 소프트웨어 설치 매뉴얼 작성

- 소프트웨어 설치 매뉴얼은 개발 초기에서부터 적용된 기준이나 사용자가 소프트웨어를 설치하는 과정에 필요한 내용을 기록한 설명서와 안내서이다.
- 설치 매뉴얼 작성 순서
  - ① 기능 식별 : 소프트웨어의 개발 목적과 주요 기능을 흐름 순으로 정리하여 기록
  - ② UI 분류 : 설치 매뉴얼을 작성할 순서대로 UI를 분류한 후 기록
  - ③ 설치 파일/백업 파일 확인 : 폴더 위치, 설치 파일, 백업 파일 등의 개별적인 기능을 확인하여 기록
  - ④ Uninstall 절차 확인 : 직접 Uninstall을 수행하면서 그 순서를 단계별로 자세히 기록
  - ⑤ 이상 Case 확인 : 설치 과정에서 발생할 수 있는 다양한 Case를 만들어 확인하고 해당 Case에 대한 대처법을 자세하게 기록
  - ⑥ 최종 매뉴얼 적용 : 설치가 완료된 화면과 메시지를 캡처하여 추가한 후 완성된 매뉴얼을 검토하고 고객 지원에 대한 내용 기록

1. 다음은 소프트웨어 설치 매뉴얼의 작성 순서를 나열한 것이다. 괄호(①, ②)에 들어갈 알맞은 작업을 쓰시오.

기능 식별 → ( ① ) → 설치 파일/백업 파일 확인 → ( ② )  
→ 이상 Case 확인 → 최종 매뉴얼 적용

답

- ① :  
② :

정답 1. ① UI 분류 ② Uninstall 절차 확인



### [핵심182] 소프트웨어 사용자 매뉴얼 작성

- 소프트웨어 사용자 매뉴얼은 사용자가 소프트웨어를 사용하는 과정에서 필요한 내용을 문서로 기록한 설명서와 안내서이다.
- 사용자 매뉴얼 작성 순서
  - ① 기능 식별 : 소프트웨어의 개발 목적과 사용자 활용 기능을 흐름 순으로 정리하여 기록
  - ② 사용자 화면 분류 : 사용자 화면을 메뉴별로 분류하여 기록
  - ③ 사용자 환경 파일 확인 : 폴더 위치, 사용자 로그 파일, 백업 파일 등의 개별적인 기능을 확인하여 기록
  - ④ 초기화 절차 확인 : 프로그램을 사용하기 위한 초기화 절차를 확인하고 그 단계를 순서대로 기록
  - ⑤ 이상 Case 확인 : 소프트웨어 사용 과정에서 발생할 수 있는 다양한 이상 Case를 만들어 확인하고 해당 Case에 대한 대처법을 자세하게 기록
  - ⑥ 최종 매뉴얼 적용 : 사용과 관련된 문의 답변(FAQ)을 기록한 후 완성된 매뉴얼을 검토하고 고객 지원에 대한 내용 기록

1. 다음은 소프트웨어 사용자 매뉴얼의 작성 순서를 나열한 것이다. 괄호(①, ②)에 들어갈 알맞은 작업을 쓰시오.

기능 식별 → ( ① ) → 사용자 환경 파일 확인 → ( ② ) →  
이상 Case 확인 → 최종 매뉴얼 적용

답

- ① :  
② :

정답 1. ① 사용자 화면 분류 ② 초기화 절차 확인

### [핵심183] 소프트웨어 패키징의 형상 관리(SCM)

- 형상 관리(SCM; Software Configuration Management)는 소프트웨어의 개발 과정에서 소프트웨어의 변경 사항을 관리하기 위해 개발된 일련의 활동이다.
- 형상 관리 기능
  - 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

#### 2020년 1회 기능사 실기

1. 다음은 결함 조치 관리에 대한 설명이다. 괄호에 들어갈 적합한 용어를 쓰시오.

( )는 소프트웨어 개발 중 소프트웨어 개발 생명주기 전반에 걸쳐 생성되는 모든 산출물의 종합 및 변경 과정을 체계적으로 관리하고 유지하는 일련의 개발 관리 활동으로, 소프트웨어에 가시성과 추적 가능성을 부여하여 제품의 품질과 안정성을 높인다.

답 :

**정답 1.** 형상 관리(SCM; Software Configuration Management)

### [핵심184] 소프트웨어 버전 등록 과정

- ① 가져오기(Import) : 개발자가 저장소에 신규로 파일을 추가함
- ② 인출(Check-Out) : 수정 작업을 진행할 개발자가 저장소에 추가된 파일을 자신의 작업 공간으로 인출함
- ③ 예치(Commit) : 인출한 파일을 수정한 후 설명을 붙여 저장소에 예치함
- ④ 동기화(Update) : 커밋(Commit) 후 새로운 개발자가 자신의 작업 공간을 동기화(Update)함
- ⑤ 차이(Diff) : 새로운 개발자가 추가된 파일의 수정 기록(Change Log)을 확인하면서 이전 개발자가 처음 추가한 파일과 이후 변경된 파일의 차이를 확인함

1. 다음은 소프트웨어 버전 등록 과정을 나열한 것이다. 괄호에 들어갈 알맞은 작업을 쓰시오.

가져오기(Import) → 인출(Check-Out) → (     ) → 동기화(Update) → 차이(Diff)

답 :

**정답 1.** 예치(Commit)

### [핵심185] 소프트웨어 버전 관리 도구

- 공유 폴더 방식 : 버전 관리 자료가 로컬 컴퓨터의 공유 폴더에 저장되어 관리되는 방식
- 클라이언트/서버 방식 : 버전 관리 자료가 중앙 시스템(서버)에 저장되어 관리되는 방식
- 분산 저장소 방식 : 버전 관리 자료가 하나의 원격 저장소와 분산된 개발자 PC의 로컬 저장소에 함께 저장되어 관리되는 방식
- Subversion(SVN) : CVS(Concurrent Version System)를 개선한 것으로, 클라이언트/서버 구조이며, 아파치 소프트웨어 재단에서 2000년에 발표하였음
- Git : 리누스 토발즈(Linus Torvalds)가 2005년 리눅스 커널 개발에 사용할 관리 도구로 개발한 이후 주니오 하마노(Junio Hamano)에 의해 유지 보수되고 있음

#### 잠깐만요 ! Git 명령어

- add : 작업 내역을 지역 저장소에 저장하기 위해 스테이징 영역(Staging Area)에 추가
- commit : 작업 내역을 지역 저장소에 저장
- branch : 새로운 브랜치 생성
- checkout : 지정한 브랜치로 이동
- merge : 지정한 브랜치의 변경 내역을 현재 HEAD 포인터가 가리키는 브랜치에 반영함으로써 두 브랜치를 병합
- init : 지역 저장소 생성
- remote add : 원격 저장소에 연결
- push : 로컬 저장소의 변경 내역을 원격 저장소에 반영
- fetch : 원격 저장소의 변경 이력만을 지역 저장소로 가져와 반영
- clone : 원격 저장소의 전체 내용을 지역 저장소로 복제
- fork : 지정한 원격 저장소의 내용을 자신의 원격 저장소로 복제

1. 리누스 토발즈(Linus Torvalds)가 2005년 리눅스 커널 개발에 사용할 관리 도구로 개발하였으며, 분산 버전 관리 시스템으로 지역 저장소와 원격 저장소가 존재하는 소프트웨어 버전 관리 도구는 무엇인지 쓰시오.

답 :

2. Git 명령어 중 지정한 브랜치의 변경 내역을 현재 HEAD 포인터가 가리키는 브랜치에 반영함으로써 두 브랜치를 합칠 때 사용하는 명령어를 쓰시오.

답 :

**정답 1.** Git(깃) **2.** merge





### [핵심186] 빌드 자동화 도구

빌드는 소스 코드 파일들을 컴파일한 후 여러 개의 모듈을 묶어 실행 파일로 만드는 과정이며, 이러한 빌드를 포함하여 테스트 및 배포를 자동화하는 도구를 빌드 자동화 도구라고 한다.

#### Jenkins

- Java 기반의 오픈 소스 형태로, 가장 많이 사용되는 빌드 자동화 도구이다.
- 서버리스 컨테이너에서 실행되는 서버 기반 도구이다.
- SVN, Git 등 대부분의 형상 관리 도구와 연동이 가능하다.
- 여러 대의 컴퓨터를 이용한 분산 빌드나 테스트가 가능하다.

#### Gradle

- Groovy를 기반으로 한 오픈 소스 형태의 자동화 도구로, 안드로이드 앱 개발 환경에서 사용된다.
- 안드로이드 뿐만 아니라 플러그인을 설정하면, Java, C/C++, Python 등의 언어도 빌드가 가능하다.
- Groovy를 사용해서 만든 DSL(Domain Specific Language)을 스크립트 언어로 사용한다.
- Gradle은 실행할 처리 명령들을 모아 태스크(Task)로 만든 후 태스크 단위로 실행한다.

1. 다음이 설명하는 빌드 자동화 도구가 무엇인지 쓰시오.

- Java 기반의 오픈 소스 형태로, 가장 많이 사용되는 빌드 자동화 도구이다.
- SVN, Git 등 대부분의 형상 관리 도구와 연동이 가능하다.

답 :

2. Groovy를 기반으로 한 오픈 소스 형태의 자동화 도구로, 기본적으로 안드로이드 앱 개발 환경에서 사용되지만 플러그인을 설정하면 Java, C/C++, Python 등의 언어도 빌드가 가능한 빌드 자동화 도구를 쓰시오.

답 :

**정답** 1. Jenkins 2. Gradle



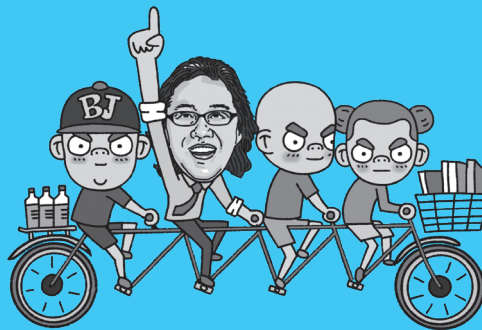
m · e · m · o





# 최신기출문제

2020년 1회 정보처리기사 실기





## 수험자 유의사항

1. 시험 문제지를 받는 즉시 응시하고자 하는 종목의 문제지가 맞는지를 확인하여야 합니다.
2. 시험 문제지 총면수 · 문제번호 순서 · 인쇄상태 등을 확인하고, 수험번호 및 성명을 답안지에 기재하여야 합니다.
3. 문제 및 답안(지), 채점기준은 일절 공개하지 않으며 자신이 작성한 답안, 문제 내용 등을 수험표 등에 이기(옮겨 적는 행위) 등은 관련 법 등에 의거 불이익 조치 될 수 있으니 유의하시기 바랍니다.
4. 수험자 인적사항 및 답안작성(계산식 포함)은 흑색 필기구만 사용하되, 흑색을 제외한 유색 필기구 또는 연필류를 사용하였을 경우 그 문항은 0점 처리됩니다.
5. 답란(답안 기재란)에는 문제와 관련 없는 불필요한 낙서나 특이한 기록사항 등을 기재하여서는 안되며 부정의 목적으로 특이한 표식을 하였다고 판단될 경우에는 모든 문항이 0점 처리됩니다.
6. 답안을 정정할 때에는 반드시 정정부분을 두 줄(=)로 그어 표시하여야 하며, 두 줄로 굿지 않은 답안은 정정하지 않은 것으로 간주합니다. (수정테이프, 수정액 사용불가)
7. 답안의 한글 또는 영문의 오타자는 오답으로 처리됩니다. 단, 답안에서 영문의 대 · 소문자 구분, 띄어쓰기는 여부에 관계 없이 채점합니다.
8. 계산 또는 디버깅 등 계산 연습이 필요한 경우는 <문제> 아래의 연습란을 사용하시기 바라며, 연습란은 채점대상이 아닙니다.
9. 문제에서 요구한 가지 수(항수) 이상을 답란에 표기한 경우에는 답안기재 순으로 요구한 가지 수(항수)만 채점하고 한 항에 여러 가지를 기재하더라도 한 가지로 보며 그 중 정답과 오답이 함께 기재란에 있을 경우 오답으로 처리됩니다.
10. 한 문제에서 소문제로 파생되는 문제나, 가지수를 요구하는 문제는 대부분의 경우 부분채점을 적용합니다. 그러나 소문제로 파생되는 문제 내에서의 부분 배점은 적용하지 않습니다.
11. 답안은 문제의 마지막에 있는 답란에 작성하여야 합니다.
12. 부정 또는 불공정한 방법(시험문제 내용과 관련된 메모지 사용 등)으로 시험을 치른 자는 부정행위자로 처리되어 당해 시험을 중지 또는 무효로 하고, 2년간 국가기술자격검정의 응시자격이 정지됩니다.
13. 시험위원이 시험 중 신분확인을 위하여 신분증과 수험표를 요구할 경우 반드시 제시하여야 합니다.
14. 시험 중에는 통신기기 및 전자기기(휴대용 전화기 등)를 지참하거나 사용할 수 없습니다.
15. 국가기술자격 시험문제는 일부 또는 전부가 저작권법상 보호되는 저작물이고, 저작권자는 한국산업인력공단입니다. 문제의 일부 또는 전부를 무단 복제, 배포, 출판, 전자출판 하는 등 저작권을 침해하는 일체의 행위를 금합니다.

※ 수험자 유의사항 미준수로 인한 채점상의 불이익은 수험자 본인에게 전적으로 책임이 있음

440101



**문제 1** SW 인터페이스 구현에 관련된 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오. (5점)

웹 페이지의 기본 형식인 HTML의 문법이 각 웹브라우저에서 상호 호환적이지 못하다는 문제와 SGML의 복잡함을 해결하기 위하여 개발된 ( )은 다른 특수한 목적을 갖는 마크업 언어이다. 원활한 데이터의 연계를 위해 송·수신 시스템 간에 전송되는 데이터가 동일한 구조로 구성될 수 있도록 형태를 정의하는 역할을 수행하며, 다음과 같은 특징이 있다.

|                             |                                                                                                                                                                    |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 유니코드 문자<br>(Unicode Text)   | 텍스트 데이터 형식으로 유니코드를 사용하여 전 세계 언어를 지원한다.                                                                                                                             |
| ( ) 파서(Parser)              | 대다수의 웹 브라우저가 해석을 위한 번역기(Parser)를 내장하고 있다.                                                                                                                          |
| 마크업(Markup)과<br>내용(Content) | <ul style="list-style-type: none"> <li>• ( ) 문서의 문자들은 마크업과 내용으로 구분된다.</li> <li>• 일반적으로 마크업은 "&lt;"로 시작하여 "&gt;"로 끝나는 태그(Tag)를 의미하고, 그 외의 문자열은 내용에 해당한다.</li> </ul> |
| 엘리먼트(Element)               | 마크업과 내용으로 이루어지는 하나의 요소를 의미한다.                                                                                                                                      |

답 :

440102



**문제 2** 소프트웨어 인터페이스 구현에 관련된 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오. (5점)

( )은 속성-값 쌍(Attribute-Value Pairs)으로 이루어진 데이터 객체를 전달하기 위해 사람이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷이다. 비동기 처리에 사용되는 AJAX에서 XML을 대체하여 사용되고 있다. 다음은 ( )이 가질 수 있는 자료 기본형이다.

|      |                                        |
|------|----------------------------------------|
| 숫자   | 정수와 실수를 표현한다.                          |
| 문자열  | 유니코드 문자로 표현하며, 큰따옴표(" ")로 묶는다.         |
| 참/거짓 | 참(True) 또는 거짓(False)을 표현한다.            |
| 배열   | 다양한 요소들을 심표로 구분하여 표현하며, 대괄호([ ])로 묶는다. |
| 객체   | 이름/값 쌍으로 표현하며, 중괄호({ })로 묶는다.          |
| NULL | 아직 알려지지 않거나 모르는 값을 표현하기 위한 자료형이다.      |

답 :

440103



**문제 3** 릴리즈 노트는 개발 과정에서 정리된 릴리즈 정보를 소프트웨어의 최종 사용자인 고객과 공유하기 위한 문서이다. 릴리즈 노트는 정확하고 완전한 정보를 기반으로 개발팀에서 직접 현재 시제로 작성해야 한다. 릴리즈 노트 작성 시 릴리즈 노트 이름, 소프트웨어 이름, 릴리즈 버전, 릴리즈 날짜, 릴리즈 노트 날짜, 릴리즈 노트 버전 등이 포함된 항목을 쓰시오. (5점)

**답 :**

440104



**문제 4** 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
public class Test {
    static int[] arr() {
        int a[] = new int[4];
        int b = a.length;
        for(int i = 0; i < b; i++)
            a[i] = i;
        return a;
    }

    public static void main(String[] args) {
        int a[] = arr();
        for(int i = 0; i < a.length; i++)
            System.out.print(a[i] + " ");
    }
}
```

**답 :**

440105



**문제 5** 소프트웨어 패키징이란 모듈별로 생성한 실행 파일들을 묶어 배포용 설치 파일을 만드는 것을 말한다. 소스 코드는 향후 관리를 고려하여 모듈화하여 패키징한다. 모듈화는 모듈 간 ( ① )의 최소화화 모듈 내 요소들의 ( ② )를 최대화 하는 것이 목표이다. 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오. (5점)

**답**

- ① :
- ② :

**문제 6** LOC 기법에 의하여 예측된 총 라인 수가 30,000라인, 개발에 참여할 프로그래머가 5명, 프로그래머들의 평균 생산성이 월간 300라인일 때 개발에 소요되는 기간을 쓰시오. (5점)

440106



**답**

- 계산과정 :
- 답 :

**문제 7** 다음은 트랜잭션의 특징이다. 괄호(①, ②)에 들어갈 알맞은 특징을 쓰시오. (5점)

440107



|       |                                                                                               |
|-------|-----------------------------------------------------------------------------------------------|
| ( ① ) | 트랜잭션의 연산은 데이터베이스에 모두 반영되도록 완료(Commit)되든지 아니면 전혀 반영되지 않도록 복구(Rollback)되어야 한다. (All or Nothing) |
| 일관성   | 트랜잭션이 그 실행을 성공적으로 완료하면 언제나 일관성 있는 데이터베이스 상태로 변환한다.                                            |
| ( ② ) | 둘 이상의 트랜잭션이 동시에 병행 실행되는 경우 어느 하나의 트랜잭션 실행중에 다른 트랜잭션의 연산이 끼어들 수 없다.                            |
| 지속성   | 성공적으로 완료된 트랜잭션의 결과는 시스템이 고장나더라도 영구적으로 반영되어야 한다.                                               |

**답**

- ① :
- ② :

**문제 8** 애플리케이션 성능이란 사용자가 요구한 기능을 최소한의 자원을 사용하여 최대한 많은 기능을 신속하게 처리하는 정도를 나타낸다. 애플리케이션 성능 측정의 지표에 대한 다음 설명에서 괄호(①~③)에 들어갈 알맞은 용어를 쓰시오. (5점)

440108



|        |                                                          |
|--------|----------------------------------------------------------|
| ( ① )  | 일정 시간 내에 애플리케이션이 처리하는 일의 양을 의미한다.                        |
| ( ② )  | 애플리케이션에 요청을 전달한 시간부터 응답이 도착할 때까지 걸린 시간을 의미한다.            |
| ( ③ )  | 애플리케이션에 작업을 의뢰한 시간부터 처리가 완료될 때까지 걸린 시간을 의미한다.            |
| 자원 활용률 | 애플리케이션이 의뢰한 작업을 처리하는 동안의 CPU, 메모리, 네트워크 등의 자원 사용률을 의미한다. |

**답**

- ① :
- ② :
- ③ :

**문제 9** 데이터베이스에서 비정규화(Denormalization)의 개념을 설명하십시오. (5점)

440109



답 :

**문제 10** 1991년 R.rivest가 MD4를 개선한 암호화 알고리즘으로, 각각의 512 비트짜리 입력 메시지 블록에 대해 차례로 동작한다. 각 512 비트 입력 메시지 블록을 처리하고 나면 128 비트 스테이트(state)의 값이 변하는 암호화 알고리즘을 쓰시오. (5점)

440110



답 :

**문제 11** 프로토콜은 서로 다른 기기들 간의 데이터 교환을 원활하게 수행할 수 있도록 표준화시켜 놓은 통신 규약이다. 프로토콜의 기본 요소 3가지를 쓰시오. (5점)

440111



답 :

**문제 12** 다음은 네트워크 공격에 대한 패킷 로그를 표현한 것이다. 아래의 패킷 로그와 같이 공격자가 패킷의 출발지 주소(Address) 또는 포트(Port)를 임의로 변경하여 송신측 IP 주소 또는 포트를 동일하게 함으로써 송신 IP 주소가 자신이므로 자신에게 응답을 수행하게 된다. 이처럼 자신에 대해 무한히 응답하는 패킷을 계속 전송하여 컴퓨터의 실행 속도를 느리게 하거나 동작을 마비 시켜 서비스 거부 상태에 빠지도록 하는 네트워크 공격 유형이 무엇인지 쓰시오. (5점)

440112



```
source : 192.168.1.200
destination : 192.168.1.200
protocol : 6
src port : 21845
dst port : 21845
```

답 :

**문제 13** 애플리케이션 테스트에서 사용되는 살충제 패러독스(Pesticide Paradox)의 개념을 설명하시오. (5점)

440113



답 :

**문제 14** 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

440114



```
#include <stdio.h>
main() {
    int c = 1;
    switch (3) {
        case 1: c += 3;
        case 2: c++;
        case 3: c = 0;
        case 4: c += 3;
        case 5: c -= 10;
        default: c--;
    }
    printf("%d", c);
}
```

답 :

**문제 15** 다음이 설명하는 OSI 참조 모델의 계층을 쓰시오. (5점)

440115

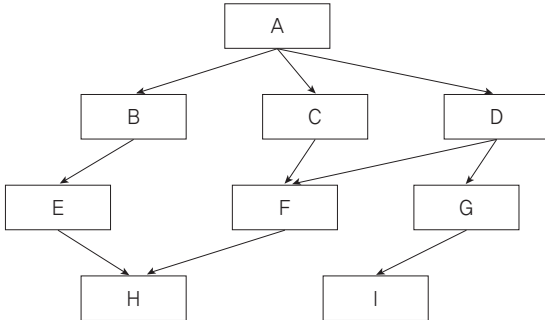


- 다양한 전송매체를 통해 비트 스트림을 전송한다.
- 전송에 필요한 두 장치 간의 실제 접속과 절단 등 기계적, 전기적, 기능적, 절차적 특성에 대한 규칙을 정의한다.

답 :



**문제 16** 다음의 모듈 관계를 표현한 시스템 구조도를 참고하여 팬인(Fan-In)이 2 이상인 모듈을 모두 쓰시오. (5점)



답 :



**문제 17** 학생(STUDENT) 테이블에 전기과 학생이 50명, 전산과 학생이 100명, 전자과 학생이 50명 있다고 할 때, 다음 SQL문 ①, ②, ③의 실행 결과로 표시되는 튜플의 수를 쓰시오. (단, DEPT 필드는 학과를 의미한다) (5점)

- ① SELECT DEPT FROM STUDENT;
- ② SELECT DISTINCT DEPT FROM STUDENT;
- ③ SELECT COUNT(DISTINCT DEPT) FROM STUDENT WHERE DEPT = '전산과';

답

- ① :
- ② :
- ③ :



**문제 18** 데이터 마이닝(Data Mining)의 개념을 설명하시오. (5점)

답 :



**문제 19** HRN 비선점형 스케줄링의 우선순위를 구하는 계산식을 쓰시오. (5점)

답 :

440119



**문제 20** 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

440120



```
#include <stdio.h>
void align(int a[]) {
    int temp;
    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 4 - i; j++)
            if (a[j] > a[j+1]) {
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
}

main() {
    int a[] = { 85, 75, 50, 100, 95 };
    align(a);
    for (int i = 0; i < 5; i++)
        printf("%d ", a[i]);
}
```

답 :



## 2020년 1회 정보처리기사 실기 정답 및 해설

### [답안 작성 방법 안내]

‘운영체제(OS; Operation System)’처럼 한글과 영문으로 제시되어 있는 경우 ‘운영체제’, ‘OS’, ‘Operation System’ 중 1가지만 쓰면 됩니다.

### [문제 1]

XML(eXtensible Markup Language)

### [문제 2]

JSON(JavaScript Object Notation)

### [문제 3]

머릿말(Header)

### [문제 4]

0 1 2 3

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 콤마를 넣어 0, 1, 2, 3으로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

### 해설

모든 Java 프로그램은 반드시 main() 메소드부터 시작해야 한다.

```
public static void main(String[] args) {  
    ❶ int a[] = arr();  
        for(int i = 0; i < a.length; i++)  
            System.out.print(a[i] + " ");  
}
```

❶ 정수형 배열 a를 선언하고, 인수 없이 arr() 메소드를 호출한 다음 돌려받은 값을 배열 a에 저장한다.

```
    ❷ static int[] arr() {  
        ❸ int a[] = new int[4];  
        ❹ int b = a.length;  
        ❺ for(int i = 0; i < b; i++)  
            ❻ a[i] = i;  
        ❼ return a;  
    }
```

❷ 메소드의 리턴값이 정수형 배열인 arr() 메소드의 시작점이다.

❸ 4개의 요소를 갖는 정수형 배열 a를 선언한다.

❹ 정수형 변수 b를 선언하고 배열 a의 길이인 4로 초기화한다.

• length : length는 배열 클래스의 속성으로 배열 요소의 개수가 저장되어 있다. a 배열은 4개의 요소를 가지므로 a.length는 4를 가지고 있다.

❺ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 b보다 작은 동안 ❻번 문장을 반복 수행한다.

⑥ a[i]에 반복 변수 i의 값을 저장한다.

|     | a[0] | a[1] | a[2] | a[3] |
|-----|------|------|------|------|
| a[] | 0    | 1    | 2    | 3    |

⑦ 배열 a에 저장된 값들을 호출한 곳(main() 메소드)으로 반환한다.

```
public static void main(String[] args) {
    ⑧ int a[] = arr();
    ⑨ for(int i = 0; i < a.length; i++)
    ⑩     System.out.print(a[i] + " ");
}
```

⑧ arr() 메소드로부터 반환받은 값들을 main() 메소드의 배열 a에 저장한다.

⑨ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 a 배열의 길이 4보다 작은 동안 ⑩번 문장을 반복 수행한다.

⑩ a[i]의 값과 공백 한 칸을 출력한다.

#### [문제 5]

① 결합도(Coupling) ② 응집도(Cohesion)

#### [문제 6]

• 계산과정 :  $(30,000 / 300) / 5 = 20$

• 답 : 20개월

#### 해설

LOC 기법에서 개발 기간은 '노력(인원) / 투입 인원'이고, 노력(인원)은 'LOC(총 라인 수) / 1인당 월평균 생산 코드 라인 수'이므로 'LOC / 1인당 월평균 생산 코드 라인 수 / 투입 인원'에 값을 대입하여 답을 구할 수 있습니다.

#### [문제 7]

① 원자성(Atomicity) ② 독립성(Isolation)

#### [문제 8]

① 처리량(Throughput) ② 응답 시간(Response Time) ③ 경과 시간(Turn Around Time)

#### [문제 9]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

비정규화란 정규화된 데이터 모델을 통합, 중복, 분리하는 과정으로, 의도적으로 정규화 원칙을 위배하는 행위이다.

#### [문제 10]

MD5(Message-Digest algorithm 5)

#### [문제 11]

구문(Syntax), 의미(Semantics), 시간(Timing)

#### [문제 12]

LAND Attack(Local Area Network Denial Attack)

**[답안 작성 방법 안내]**

'운영체제(OS; Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

**[문제 13]**

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

살충제 패러독스는 동일한 테스트 케이스로 동일한 테스트를 반복하면 더 이상 결함이 발견되지 않는 현상을 의미한다.

**[문제 14]**

-8

**해설**

```
#include <stdio.h>
main() {
  ① int c = 1;
  ② switch (3) {
    case 1: c += 3;
    case 2: c++;
  ③ case 3: c = 0;
  ④ case 4: c += 3;
  ⑤ case 5: c -= 10;
  ⑥ default: c--;
  }
  ⑦ printf("%d", c);
}
```

모든 case문에 break문이 생략되었으므로, switch문의 인수와 일치하는 'case 3' 문장부터 switch문이 종료될 때까지 모든 문장이 실행된다.

- ① 정수형 변수 c를 선언하고 1로 초기화한다. →  $c = 1$
- ② 3에 해당하는 숫자를 찾아간다. 'case 3' 문장으로 이동한다.
- ③ c의 값을 0으로 치환한다. →  $c = 0$
- ④ ' $c = c + 3$ '과 동일하다. c의 값에 3을 더한다. →  $c = 3$
- ⑤ ' $c = c - 10$ '과 동일하다. c의 값에서 10을 뺀다. →  $c = -7$
- ⑥ ' $c = c - 1$ '과 동일하다. c의 값에서 1을 뺀다. →  $c = -8$
- ⑦ c의 값을 출력한다. 결과 **-8**

**[문제 15]**

물리 계층(Physical Layer)

**[문제 16]**

F, H

**해설**

팬인(Fan-in)은 모듈로 들어오는(n) 선(Line)의 수라고 생각하면 됩니다. 그러므로 2개의 선이 들어오는 F, H 모듈이 답이 됩니다.

[문제 17]

① 200    ② 3    ③ 1

**해설**

① <SQL문>

```
SELECT DEPT
FROM STUDENT;
```

'DEPT'를 표시한다.  
<STUDENT> 테이블을 대상으로 검색한다.

※ <STUDENT> 테이블에서 'DEPT'를 검색합니다. 총 200개의 튜플이 들어 있고 검색 조건이 없으므로 튜플의 수는 200개 입니다.

② <SQL문>

```
SELECT DISTINCT DEPT
FROM STUDENT;
```

'DEPT'를 표시하되, 같은 'DEPT' 속성의 값은 한 번만 표시한다.  
<STUDENT> 테이블을 대상으로 검색한다.

※ <STUDENT> 테이블에서 'DEPT'를 검색하는 데 중복된 결과는 처음의 한 개만 검색에 포함시킵니다. 전기과 50개 튜플의 'DEPT' 속성의 값이 같으므로 1개, 전산과 100개 튜플의 'DEPT' 속성의 값이 같으므로 1개, 전자과 50개 튜플의 'DEPT' 속성의 값이 같으므로 1개를 검색에 포함시키므로 3개의 튜플이 검색됩니다.

③ <SQL문>

```
SELECT COUNT(DISTINCT DEPT)
FROM STUDENT
WHERE DEPT = '전산과';
```

'DEPT'의 개수를 표시하되, 같은 'DEPT' 속성의 값은 한 번만 계산한다.  
<STUDENT> 테이블을 대상으로 검색한다.  
'DEPT'가 "전산과"인 자료만을 대상으로 검색한다.

※ <STUDENT> 테이블에서 'DEPT' 속성의 값이 '전산과'인 튜플에 대해 중복을 제거하고 개수를 세므로 1개의 튜플이 검색됩니다.

[문제 18]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

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

[문제 19]

(대기 시간 + 서비스 시간) / 서비스 시간

[문제 20]

50 75 85 95 100

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 콤마를 넣어 50, 75, 85, 95, 100으로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

**해설**

문제의 코드는 버블 정렬 알고리즘을 이용하여 배열 `a`에 저장된 값을 오름차순으로 정렬한 후 출력하는 프로그램이다. 버블 정렬 알고리즘은 첫 번째 자료와 두 번째 자료를, 두 번째 자료와 세 번째 자료를, 세 번째와 네 번째를, ... 이런 식으로 (마지막-1)번째 자료와 마지막 자료를 비교하여 교환하면서 자료를 정렬한다. 1회전을 수행하고 나면 가장 큰 자료가 맨 뒤로 이동하므로 2회전에서는 맨 끝에 있는 자료는 정렬에서 제외되고, 2회전을 수행하고 나면 끝에서 두 번째 자료까지는 정렬에서 제외된다. 이렇게 정렬을 1회전 수행할 때마다 정렬에서 제외되는 데이터가 하나씩 늘어난다.

모든 C 프로그램은 반드시 `main()` 함수부터 시작해야 한다.

```
main() {
  ❶ int a[] = { 85, 75, 50, 100, 95 };
  ❷ align(a);
    for (int i = 0; i < 5; i++)
      printf("%d ", a[i]);
}
```

❶ 배열을 선언할 때 사용할 개수를 생략하고 초기값을 지정하면, 초기값으로 지정된 값의 수와 같은 크기의 배열이 선언된다.

|      | a[0] | a[1] | a[2] | a[3] | a[4] |
|------|------|------|------|------|------|
| 배열 a | 85   | 75   | 50   | 100  | 95   |

❷ `a`를 인수로 하여 함수 `align()`을 호출한다. 인수로 배열의 이름을 지정하면 배열의 시작 주소가 인수로 전달된다. 그러니까 `align(a)`는 `align(&a[0])`과 같은 의미이다.

```
  ❸ void align(int a[]) {
  ❹ int temp;
  ❺ for (int i = 0; i < 4; i++)
  ❻ for (int j = 0; j < 4 - i; j++)
  ❼ if (a[j] > a[j+1]) {
  ❽     temp = a[j];
  ❾     a[j] = a[j+1];
  ❿     a[j+1] = temp;
  ⓫ }
  ⓬ }
```

❸ ❷에서 '`align(a)`'라고 했으므로 정수형 배열 `a`는 `main()` 함수의 `a` 배열의 시작 주소를 받는다.

|      | a[0] | a[1] | a[2] | a[3] | a[4] |
|------|------|------|------|------|------|
| 배열 a | 85   | 75   | 50   | 100  | 95   |

❹ 정수형 변수 `temp`를 선언한다.

❺ 반복 변수 `i`가 0에서 시작하여 1씩 증가하면서 4보다 작은 동안 ❻번을 반복하여 수행한다.

❻ 반복 변수 `j`가 0에서 시작하여 1씩 증가하면서 4-`i`보다 작은 동안 ❼번을 반복하여 수행한다.

❼ `a[j]`가 `a[j+1]`보다 크면 ❽~❿ 사이의 문장을 실행한다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

| i | j | a[j] | a[j+1] | temp | 배열 a |
|---|---|------|--------|------|------|
| 0 | 0 | 85   | 75     | 85   |      |
|   | 1 | 75   | 85     | 85   |      |
|   | 2 | 85   | 50     | 100  |      |
|   | 3 | 50   | 85     |      |      |
|   | 4 | 85   | 100    |      |      |
| 1 | 0 | 75   | 50     | 75   |      |
|   | 1 | 50   | 75     |      |      |
|   | 2 | 75   | 85     |      |      |
|   | 3 | 85   | 95     |      |      |
| 2 | 0 | 50   | 75     |      |      |
|   | 1 | 75   | 85     |      |      |
|   | 2 |      |        |      |      |
| 3 | 0 | 50   | 75     |      |      |
|   | 1 |      |        |      |      |
| 4 |   |      |        |      |      |

- ⑪ #문의 끝이다.  
 ⑫ 함수를 마치고 align(a) 함수를 호출했던 main( ) 함수로 제어를 옮긴다.

```
main( ) {
    int a[] = { 85, 75, 50, 100, 95 };
    align(a);
    ⑬ for (int i = 0; i < 5; i++)
    ⑭ printf("%d ", a[i]);
}
```

- ⑬ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 5보다 작은 동안 ⑭번을 반복 수행한 후 프로그램을 종료한다.  
 ⑭ a[i]의 값을 정수로 출력한 후 한 칸을 띄운다.

결과 50 75 85 95 100

m · e · m · o





# 실전 모의고사

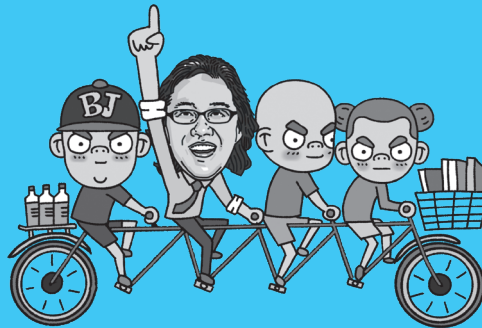
1회 실전 모의고사

2회 실전 모의고사

3회 실전 모의고사

4회 실전 모의고사

5회 실전 모의고사





**문제 1** 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
#include <stdio.h>
main() {
    int k = 2, j;
    while(1) {
        j = 2;
        while (k % j != 0)
            j++;
        if (k == j)
            printf("%d ", k);
        if (k < 7)
            k++;
        else
            break;
    }
}
```

**답 :**

**문제 2** 시스템이나 객체들이 메시지를 주고받으며 시간의 흐름에 따라 상호 작용하는 과정을 액터, 객체, 메시지 등의 요소를 사용하여 그림으로 표현한 UML의 다이어그램은 무엇인지 쓰시오. (5점)

**답 :**

**문제 3** 데이터베이스와 관련된 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오. (5점)

테이블에서 일부 속성들의 중복으로 인해 데이터의 중복(Redundancy)이 발생하고, 이 중복으로 인해 테이블 조작 시 문제가 발생하는 현상을 ( )이라고 하며, 종류는 다음과 같다.

|        |                                                                             |
|--------|-----------------------------------------------------------------------------|
| 삽입 ( ) | 테이블에 데이터를 삽입할 때 의도와는 상관없이 원하지 않은 값들로 인해 삽입할 수 없게 되는 현상이다.                   |
| 삭제 ( ) | 테이블에서 한 튜플을 삭제할 때 의도와는 상관없는 값들도 함께 삭제되는, 즉 연쇄 삭제가 발생하는 현상이다.                |
| 갱신 ( ) | 테이블에서 튜플에 있는 속성 값을 갱신할 때 일부 튜플의 정보만 갱신되어 정보에 불일치성(Inconsistency)이 생기는 현상이다. |

**답 :**

**문제 4** 사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블이다. 저장장치 내에 물리적으로 존재하지 않으며, 조인문의 사용이 줄어 접근 시간을 단축할 수 있고, 필요한 데이터만 노출하는 것이 가능해 보안상의 이유로도 사용되는 데이터베이스의 개체를 쓰시오. (5점)

**답 :**

**문제 5** 데이터베이스에서 파티션(Partition)은 대용량의 테이블이나 인덱스를 작은 논리적 단위인 파티션으로 나누는 것을 말한다. 파티션의 종류에 대한 다음 설명에서 괄호(①~③)에 들어갈 알맞은 종류를 쓰시오. (5점)

|       |                                            |
|-------|--------------------------------------------|
| ( ① ) | 지정한 열의 값을 기준으로 분할한다.                       |
| ( ② ) | 해시 함수를 적용한 결과 값에 따라 데이터를 분할한다.             |
| ( ③ ) | 지정한 열의 값을 기준으로 분할한 다음 해시 함수를 적용하여 다시 분할한다. |

**답**

- ① :
- ② :
- ③ :

**문제 6** 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
public class Test {
    public static void main(String[] args) {
        int i = 0, hap = 0;
        String a = "Technology";
        do {
            if(a.charAt(i) == 'o')
                hap += i;
            i++;
        } while (i < a.length());
        System.out.printf("%d", hap);
    }
}
```

**답 :**

**문제 7** 운영체제와 해당 운영체제에서 실행되는 응용 프로그램 사이에서 운영체제가 제공하는 서비스 이외에 추가적인 서비스를 제공하고, 표준화된 인터페이스를 통해 시스템 간의 데이터 교환에 일관성을 보장하는 역할을 수행하는 소프트웨어를 쓰시오. (5점)

**답 :**

**문제 8** 정적인 콘텐츠를 처리하는 웹 서버와 달리 사용자의 요구에 따라 변하는 동적인 콘텐츠를 처리하고, 클라이언트/서버 환경보다는 웹 환경을 구현하기 위해 사용되며, HTTP 세션 처리를 위한 웹 서버 기능뿐만 아니라 업무 수행에 중요한 로직까지 JAVA, EJB 컴포넌트를 기반으로 구현이 가능한 미들웨어는 무엇인지를 쓰시오. (5점)

**답 :**

**문제 9** 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
public class Test {
    public static int a = 5;
    public static void main(String[] args) {
        int b = 2;
        cal(b);
        cal(b);
        System.out.printf("%d, %d\n", a, b);
    }
    static void cal(int b) {
        if (b < a)
            a -= 3;
        else
            b += 3;
    }
}
```

**답 :**

**문제 10** 공통 모듈과 관련된 다음 설명에서 괄호(①~③)에 들어갈 알맞은 용어를 쓰시오. (5점)

|       |                                                |
|-------|------------------------------------------------|
| 정확성   | 시스템 구현 시 해당 기능이 필요하다는 것을 알 수 있도록 작성한다.         |
| ( ① ) | 해당 기능을 이해할 때 중의적으로 해석되지 않도록 작성한다.              |
| ( ② ) | 시스템 구현을 위해 필요한 모든 것을 기술한다.                     |
| ( ③ ) | 공통 기능들 간 상호 충돌이 발생하지 않도록 작성한다.                 |
| 추적성   | 기능에 대한 요구사항의 출처, 관련 시스템 등의 관계를 파악할 수 있도록 작성한다. |

**답**

- ① :
- ② :
- ③ :

**문제 11** 다음은 5개의 정수를 입력받아 그 중 짝수의 개수를 구하여 출력하는 알고리즘을 C언어로 구현한 것이다. 프로그램을 분석하여 괄호에 가장 적합한 답을 쓰시오. (5점)

```
#include <stdio.h>
main() {
    int i, a[5], cnt = 0;
    for (i = 0; i < 5; i++)
        scanf("%d", &a[i]);
    for (i = 0; i < 5; i++)
        if (a[i] % 2 (      ) 0)
            cnt = cnt + 1;
    printf("짝수의 개수 : %d개", cnt);
}
```

답 :

**문제 12** 프레임워크(Framework)의 특성 중 제어의 역흐름(Inversion of Control)의 개념을 설명하시오. (5점)

답 :

**문제 13** 사용자 인터페이스(UI; User Interface)는 사용자와 시스템 간의 상호작용이 원활하게 이뤄지도록 도와주는 장치나 소프트웨어를 의미한다. 사용자 인터페이스의 기본 원칙 4가지를 쓰시오. (5점)

답 :

**문제 14** 입력 자료에 초점을 맞춰 테스트 케이스를 만들고 검사하는 방법으로, 프로그램의 입력 조건에 타당한 입력 자료와 타당하지 않은 입력 자료의 개수를 균등하게 하여 테스트 케이스를 정하고, 해당 입력 자료에 맞는 결과가 출력 되는지 확인하는 테스트 기법을 쓰시오. (5점)

답 :

**문제 15** 애플리케이션 테스트에서 사용되는 오류-부재의 궤변(Absence of Errors Fallacy)의 개념을 설명하시오. (5점)

**답 :**

**문제 16** 다음 〈처리조건〉에 유의하여 〈상품목록〉 테이블에서 상품명이 'DR-725F'인 레코드의 제조가를 100000으로 바꾸고, 생산지를 'Australia'로 변경하는 SQL문을 작성하시오. (5점)

〈상품목록〉

| 상품코드  | 상품명     | 제조가    | 생산지       | 출고일        |
|-------|---------|--------|-----------|------------|
| 87495 | CK-85M  | 56000  | Hong Kong | 2020-07-20 |
| 81323 | DR-725F | 123200 | China     | 2020-06-24 |
| 51453 | ED-623  | 41800  | China     | 2020-07-09 |
| 35485 | RA-9503 | 96000  | Hong Kong | 2020-07-13 |
| 72695 | OE-132R | 160000 | Germany   | 2020-08-02 |
| 65484 | UM-111  | 28000  | China     | 2020-07-26 |
| 21638 | GQ-580  | 74500  | Australia | 2020-06-10 |

〈처리조건〉

1. 명령질의 마지막은 ';'으로 끝낸다.
2. 참조의 경우 '('(작은따옴표)를 이용하여 작성한다.

**답 :**

**문제 17** 소프트웨어 개발에 있어 충족시켜야 할 보안 요소 중 패스워드, 인증용 카드, 지문 검사 등의 방법으로 시스템 내의 정보와 자원을 사용하려는 사용자가 합법적인 사용자인지를 확인하는 모든 행위를 가리키는 용어를 쓰시오. (5점)

**답 :**

**문제 18** 학계, 연구기관, 정부의 영문 앞 글자를 따 명명한 양방향 알고리즘의 한 종류로 국가정보원과 산학연협회가 2004년 개발하였다. 128비트의 블록 크기와 키 길이에 따라 128, 192, 256으로 분류되는 블록 암호화 알고리즘의 명칭을 영문 약어 4글자로 쓰시오. (5점)

**답 :**

**문제 19** Windows, MacOS에서 사용하는 인터페이스로, 키보드를 이용하여 명령어를 직접 입력하지 않고, 마우스를 이용하여 아이콘이나 메뉴를 선택하여 모든 작업을 수행하는 사용자 인터페이스를 쓰시오. (5점)

**답 :**

**문제 20** 기억장치의 배치(Placement) 전략과 관련된 다음 설명에서 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오. (5점)

기억장치의 관리 전략 중 보조기억장치의 프로그램이나 데이터를 주기억장치에 적재 시키는 위치를 지정하여 한정된 주기억장치의 공간을 효율적으로 사용하기 위한 방법을 배치(Placement) 전략이라고 한다. 배치 전략에는 새로 반입되는 프로그램이나 데이터를 주기억장치의 어디에 위치시킬 것인지에 따라 최초 적합, ( ① ), ( ② )으로 구분되며, 각 방법의 설명은 다음과 같다.

|       |                                                               |
|-------|---------------------------------------------------------------|
| 최초 적합 | 프로그램이나 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 첫 번째 분할 영역에 배치한다.           |
| ( ① ) | 프로그램이나 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 단편화를 가장 작게 남기는 분할 영역에 배치한다. |
| ( ② ) | 프로그램이나 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 단편화를 가장 많이 남기는 분할 영역에 배치한다. |

**답**

- ① :
- ② :





## 1회 실전 모의고사 정답 및 해설

### [문제 1]

2 3 5 7

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 콤마를 넣어 2, 3, 5, 7로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

#### 해설

문제의 코드는 7 이하의 소수들을 출력하는 프로그램입니다. 숫자를 2부터 7까지 증가시키면서 각각의 숫자가 소수인지를 판별하기 위해 2부터 1씩 증가시키면서 나누어 떨어지는지 검사합니다. 예를 들어 5는 2, 3, 4, 5로 나누었을 때 나머지가 0이 되는 순간이 자기수(5)이므로 소수이고, 6은 2, 3, 4, 5, 6으로 나누었을 때 나머지가 0이 되는 순간이 자기수(6)가 아닌 2이므로 소수가 아닙니다.

```
#include <stdio.h>
main() {
    int k = 2, j;
    ① while(1) {
    ②         j = 2;
    ③         while (k % j != 0)
    ④             j++;
    ⑤         if (k == j)
    ⑥             printf("%d ", k);
    ⑦         if (k < 7)
    ⑧             k++;
    ⑨         else
                break;
    }
}
```

- ① 조건이 참(1)이므로 ⑨번의 break를 만날 때까지 while문 내의 문장을 무한 반복한다.
- ② j의 값을 2로 치환한다.
- ③ k를 j로 나눈 나머지가 0이 아니면 ④번 문장을 실행하고, 0이면 ⑤번으로 이동한다.
- ④ 'j = j + 1;'과 동일하다. j에 1씩 누적한다.
- ⑤ k와 j의 값이 같으면 ⑥번 문장을 실행하고, 아니면 ⑦번으로 이동한다.
- ⑥ k의 값을 출력하고 이어서 공백을 한 칸 출력한다.
- ⑦ k의 값이 7보다 작으면 ⑧번 문장을 실행하고, 아니면 ⑨번 문장을 실행한다.
- ⑧ 'k = k + 1;'과 동일하다. k에 1씩 누적한다.
- ⑨ while문을 벗어나 프로그램을 종료한다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

| k | j | k % j | 출력   |
|---|---|-------|------|
| 2 | 2 | 0     | 2357 |
| 3 | 2 | 1     |      |
| 4 | 3 | 0     |      |
| 5 | 2 | 0     |      |
| 6 | 2 | 1     |      |
| 7 | 3 | 2     |      |
|   | 4 | 1     |      |
|   | 5 | 0     |      |
|   | 2 | 0     |      |
|   | 2 | 1     |      |
|   | 3 | 1     |      |
|   | 4 | 3     |      |
|   | 5 | 2     |      |
|   | 6 | 1     |      |
|   | 7 | 0     |      |

**[답안 작성 방법 안내]**

'운영체제(OS; Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

**[문제 2]**

시퀀스 다이어그램(Sequence Diagram)

**[문제 3]**

이상(Anomaly)

**[문제 4]**

뷰(View)

**[문제 5]**

① 범위 분할(Range Partitioning) ② 해시 분할(Hash Partitioning) ③ 조합 분할(Composite Partitioning)

[문제 6]

12

해설

```
public class Test {
    public static void main(String[ ] args) {
        int i = 0, hap = 0;
        ① String a = "Technology";
        ② do {
            ③     if(a.charAt(i) == 'o')
            ④         hap += i;
            ⑤     i++;
        ⑥ } while (i < a.length( ));
        ⑦ System.out.printf("%d", hap);
    }
}
```

① 문자열 변수 *a*를 선언하고 "Technology"로 초기화한다.

|   | a[0] | a[1] | a[2] | a[3] | a[4] | a[5] | a[6] | a[7] | a[8] | a[9] |
|---|------|------|------|------|------|------|------|------|------|------|
| a | T    | e    | c    | h    | n    | o    | l    | o    | g    | y    |

② do~while 반복문의 시작점이다.

③ *a*에서 *i*번째에 있는 문자가 'o'이면 ④번 문장을 실행하고, 아니면 ⑤번으로 이동한다.

• charAt( ) : 해당 문자열에서 인수에 해당하는 위치의 문자를 반환하는 메소드

④ hap에 *i*의 값을 누적한다.

⑤ 'i = i + 1;'과 동일하다. *i*에 1씩 누적한다.

⑥ *i*가 문자열 변수 *a*의 크기인 10보다 작은 동안 ③~⑤번을 반복 수행한다.

• length( ) : 변수의 크기를 반환하는 메소드

⑦ hap의 값을 출력한다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

| a          | i  | a.charAt(i) | hap | a.length() | 출력 |
|------------|----|-------------|-----|------------|----|
| Technology | 0  | T           | 0   | 10         | 12 |
|            | 1  | e           | 5   |            |    |
|            | 2  | c           | 12  |            |    |
|            | 3  | h           |     |            |    |
|            | 4  | n           |     |            |    |
|            | 5  | o           |     |            |    |
|            | 6  | l           |     |            |    |
|            | 7  | o           |     |            |    |
|            | 8  | g           |     |            |    |
|            | 9  | y           |     |            |    |
|            | 10 |             |     |            |    |

**[답안 작성 방법 안내]**

‘운영체제(OS; Operation System)’처럼 한글과 영문으로 제시되어 있는 경우 ‘운영체제’, ‘OS’, ‘Operation System’ 중 1가지만 쓰면 됩니다.

**[문제 7]**

미들웨어(Middleware)

**[문제 8]**

WAS(Web Application Server, 웹 애플리케이션 서버)

**[문제 9]**

2, 2

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 콤마(,) 없이 2 2로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

**해설**

```
public class Test {
  ① public static int a = 5;
  ② public static void main(String[] args) {
    ③ int b = 2;
    ④ cal(b);
    cal(b);
    System.out.printf("%d, %d\n", a, b);
  }
}
```

- ① 정수형 전역 변수 `a`를 선언하고, 초기값으로 5를 할당한다. `a`는 `main()` 메소드 밖에서 선언했기 때문에 이 클래스에 속한 모든 메소드에서 사용할 수 있다.
- ② `main()` 메소드의 시작이다.
- ③ 정수형 변수 `b`를 선언하고 2로 초기화한다.
- ④ `b`의 값 2를 인수로 하여 `cal()` 메소드를 호출한다.

```
  ⑤ static void cal(int b) {
    ⑥ if (b < a)
    ⑦ a -= 3;
    else
      b += 3;
  }
```

- ⑤ `cal()` 메소드의 시작점이다. `cal()` 메소드가 호출될 때 2를 전달받았으므로 `b`는 2이다.
- ⑥ `b`의 값 2가 `a`의 값 5보다 작으므로 ⑦번 문장을 실행한다.
- ⑦ '`a = a - 3`'과 같다. `a`의 값에서 3을 뺀 값을 `a`에 저장하고, 메소드가 종료되었으므로 `main()` 메소드로 돌아간다. `a`는 전역 변수이므로 값의 변화가 클래스에 속한 모든 메소드에서 유지된다.

```
public static void main(String[ ] args) {
    int b = 2;
    cal(b);
    ⑧ cal(b);
    System.out.printf("%d, %d\n", a, b);
}
```

⑧ b의 값 2를 인수로 하여 cal() 메소드를 호출한다.

```
⑨ static void cal(int b) {
    ⑩ if (b < a)
        a -= 3;
    else
    ⑪ b += 3;
}
```

⑨ cal() 메소드가 호출될 때 2를 전달받았으므로 b는 2이다.

⑩ b의 값 2가 a의 값 2보다 작지 않으므로 ⑪번 문장을 실행한다.

⑪ 'b = b + 3;'과 같다. b에 3을 누적하고, 메소드가 종료되었으므로 main() 메소드로 돌아간다. b는 메소드 안에서 선언된 지역 변수이므로 cal() 메소드를 벗어나면 소멸한다.

```
public static void main(String[ ] args) {
    int b = 2;
    cal(b);
    cal(b);
    ⑫ System.out.printf("%d, %d\n", a, b);
}
```

⑫ a와 b의 값을 출력한다.  
결과 2 2

사용자 정의 함수(메소드)에 따른 변수들의 값의 변화는 다음과 같다.

| a | main() 메소드 | cal() 메소드 | 출력   |
|---|------------|-----------|------|
|   | b          | b         |      |
| 5 | 2          | 2         | 2, 2 |
| 2 |            | 5         |      |

※ main() 메소드에서 선언한 b와 cal() 메소드에서 선언한 b는 해당 영역 내에서만 유효합니다. 그러므로 cal() 메소드에서 b의 값이 변경되어도 이 변경된 값이 main() 메소드의 b 변수에 전달되지 않으므로 main() 메소드에서 b 값을 출력할 때는 main() 메소드에서 선언한 b의 초기값인 2가 그대로 출력됩니다.

#### [문제 10]

① 명확성(Clarify) ② 완전성(Completeness) ③ 일관성(Consistency)

#### [문제 11]

==

※ 답안 작성 시 주의 사항

C언어에서 사용하는 형식에 맞게 정확히 작성해야 합니다. '같다'는 표현을 '='으로 하지 않도록 주의하세요.

## 해설

```
#include <stdio.h>
main() {
    int i, a[5], cnt = 0;
    ❶ for (i = 0; i < 5; i++)
    ❷     scanf("%d", &a[i]);
    ❸ for (i = 0; i < 5; i++)
    ❹     if (a[i] % 2 == 0)
    ❺         cnt = cnt + 1;
    ❻ printf("짝수의 개수 : %d개", cnt);
}
```

- ❶ 반복 변수  $i$ 가 0에서 시작하여 1씩 증가하면서 5보다 작은 동안 ❷번을 반복하여 수행한다.  
 ❷  $a[i]$ 에 입력받은 값을 저장한다.  
 ❸ 반복 변수  $i$ 가 0에서 시작하여 1씩 증가하면서 5보다 작은 동안 ❹, ❺번을 반복하여 수행한다.  
 ❹  $a[i]$ 를 2로 나눈 나머지가 0이면 ❺번 문장을 실행한다.  
 ❺  $cnt$ 에 1씩 누적한다.  
 ❻ 화면에 짝수의 개수 : 를 출력하고, 이어서  $cnt$ 의 값을 출력한 다음, 개를 출력한다.

❶번 반복문에서 배열  $a$ 에 차례대로 1, 2, 3, 4, 5가 입력되었다고 가정하고, ❸번 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

| i | a[i] | a[i]%2 | cnt | 배열 a                                                                                                                                                              |
|---|------|--------|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 | 1    | 1      | 0   | <div><div><div>a[i]</div><div>a[0]a[1]a[2]a[3]a[4]</div><div><div>1</div><div>2</div><div>3</div><div>4</div><div>5</div></div></div><div>짝수의 개수 : 2개</div></div> |
| 1 | 2    | 0      | 1   |                                                                                                                                                                   |
| 2 | 3    | 1      | 2   |                                                                                                                                                                   |
| 3 | 4    | 0      |     |                                                                                                                                                                   |
| 4 | 5    | 1      |     |                                                                                                                                                                   |
| 5 |      |        |     |                                                                                                                                                                   |

## [답안 작성 방법 안내]

‘운영체제(OS; Operation System)’처럼 한글과 영문으로 제시되어 있는 경우 ‘운영체제’, ‘OS’, ‘Operation System’ 중 1가지만 쓰면 됩니다.

## [문제 12]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

제어의 역흐름은 개발자가 관리하고 통제해야 하는 객체들의 제어 권한을 프레임워크에 넘겨 생산성을 향상시키는 것이다.

## [문제 13]

직관성, 유효성, 학습성, 유연성

## [문제 14]

동치 분할 검사(Equivalence Partitioning Testing)

## [문제 15]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

오류-부재의 궤변은 소프트웨어의 결함을 모두 제거해도 사용자의 요구사항을 만족시키지 못하면 해당 소프트웨어는 품질이 높다고 말할 수 없는 것을 의미한다.

[문제 16]

UPDATE 상품목록 SET 제조가=100000, 생산지='Australia' WHERE 상품명='DR-725F';

해설

UPDATE 상품목록  
SET 제조가=100000, 생산지='Australia'  
WHERE 상품명='DR-725F';

〈상품목록〉 테이블을 갱신한다.  
'제조사'를 100,000으로 '생산지'를 "Australia"로 갱신한다.  
'상품명'이 "DR-725F"인 튜플만을 대상으로 한다.

[문제 17]

인증(Authentication)

[문제 18]

ARIA

[문제 19]

GUI(Graphic User Interface, 그래픽 사용자 인터페이스)

[문제 20]

① 최적 적합(Best Fit) ② 최악 적합(Worst Fit)





**문제 1** 소프트웨어를 설계하기에 앞서 무엇을 개발해야 하는지 요구사항을 정의하고 분석 및 관리하는 프로세스를 연구하는 것이다. 요구사항 변경의 원인과 처리 방법을 이해하고 요구사항 관리 프로세스의 품질을 개선하여 소프트웨어 프로젝트 실패를 최소화하는 것을 목표로 하는 것은 무엇인지 쓰시오. (5점)

답 :

**문제 2** 다음은 <동아리> 릴레이션에서 결정자이면서 후보키가 아닌 속성을 제거하기 위해 <회원> 릴레이션과 <강사> 릴레이션으로 무손실 분해한 것이다. <회원> 릴레이션과 <강사> 릴레이션은 어떤 정규형(Normal Form)인지 쓰시오. (5점)

| <동아리> |      |     | <회원> |     |
|-------|------|-----|------|-----|
| 회원번호  | 소속   | 강사명 | 회원번호 | 강사명 |
| 6255  | 축구   | 김범수 | 6255 | 김범수 |
| 6256  | 테니스  | 나경락 | 6256 | 나경락 |
| 6257  | 클라이밍 | 박인순 | 6257 | 박인순 |
| 6258  | 축구   | 김범수 | 6258 | 김범수 |
| 6256  | 축구   | 이시경 | 6256 | 이시경 |
| 6258  | 테니스  | 나경락 | 6258 | 나경락 |

| <강사> |      |
|------|------|
| 강사명  | 소속   |
| 김범수  | 축구   |
| 이시경  | 축구   |
| 나경락  | 테니스  |
| 박인순  | 클라이밍 |

답 :

**문제 3** 데이터베이스 시스템에서 데이터베이스 내부에 저장되어 데이터 변경 및 무결성 유지, 로그 메시지 출력 등의 목적으로, 데이터의 삽입(Insert), 갱신(Update), 삭제(Delete) 등의 이벤트가 발생할 때마다 지정한 작업이 자동으로 수행되게 하는 절차형 SQL을 쓰시오. (5점)

답 :





**문제 4** 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
#include <stdio.h>
main() {
    int a, b, c, sum;
    a = b = 1;
    sum = a + b;
    for (int i = 3; i <= 5; i++) {
        c = a + b;
        sum += c;
        a = b;
        b = c;
    }
    printf("%d", sum);
}
```

**답 :**

**문제 5** 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
public class Test {
    public static void main(String[] args) {
        int a[] = new int[8];
        int d = 11, n = 0;
        do {
            a[n++] = d % 2;
            d /= 2;
        } while (d > 0);
        for (n = 7; n >= 0; n--)
            System.out.printf("%d", a[n]);
    }
}
```

**답 :**

**문제 6** 인덱스(Index)와 관련된 다음 설명에서 괄호에 그 중 들어갈 알맞은 인덱스 종류를 쓰시오. (5점)

인덱스는 데이터 레코드를 빠르게 접근하기 위해 <키 값, 포인터> 쌍으로 구성되는 데이터 구조로, 인덱스를 구성하는 구조나 특징에 따라 다양한 형태가 존재한다. 그 중 ( )는 인덱스 컬럼의 데이터를 Bit 값인 0 또는 1로 변환하여 인덱스 키로 사용하는 방법으로, 키 값을 포함하는 로우(Row)의 주소를 제공하는 것이 목적이며, 데이터가 Bit로 구성되어 있기 때문에 효율적인 논리 연산이 가능하고 저장 공간이 작다. 또한 동일한 값이 반복되는 경우가 많아 압축 효율이 좋다.

**답 :**

**문제 7** 응용 프로그램 개발 시 운영체제나 프로그래밍 언어 등에 있는 라이브러리를 이용할 수 있도록 규칙 등을 정의해 놓은 인터페이스로, 프로그래밍 언어에서 특정한 작업을 수행하기 위해 사용되거나, 운영체제의 파일 제어, 화상 처리, 문자 제어 등의 기능을 활용하기 위해 사용되는 것은 무엇인지 쓰시오. (5점)

**답 :**

**문제 8** 애플리케이션 테스트는 무엇을 기반으로 수행하느냐에 따라 다음과 같이 분류된다. 괄호(①~③)에 들어갈 알맞은 테스트 종류를 쓰시오. (5점)

|       |                                                          |
|-------|----------------------------------------------------------|
| ( ① ) | 사용자의 요구사항에 대한 명세를 빠짐없이 테스트 케이스로 만들어 구현하고 있는지 확인하는 테스트이다. |
| ( ② ) | 소프트웨어 내부의 논리 흐름에 따라 테스트 케이스를 작성하고 확인하는 테스트이다.            |
| ( ③ ) | 유사 소프트웨어나 기술 등에 대한 테스터의 경험을 기반으로 수행하는 테스트이다.             |

**답**

- ① :
- ② :
- ③ :

**문제 9** 애플리케이션에서 검증 기준(Coverage)에 대한 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오. (5점)

테스트 검증 기준은 테스트가 얼마나 적절한지를 판단하는 기준으로, 종류에는 크게 기능 기반 커버리지, 라인 커버리지, ( )가 있다.

|            |                                                         |
|------------|---------------------------------------------------------|
| 기능 기반 커버리지 | 실제 테스트가 수행된 기능의 수 / 전체 기능의 수                            |
| 라인 커버리지    | 테스트 시나리오가 수행한 소스 코드의 라인 수 / 전체 소스 코드의 라인 수              |
| ( )        | 소스 코드의 구문, 분기, 조건 등의 구조 코드 자체가 얼마나 테스트 되었는지를 측정하는 방법이다. |

**답 :**

**문제 10** 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
public class Test {
    public static void main(String[] args) {
        int numAry[] = { 1, 2, 3, 4, 5 };
        init(numAry);
        prnt(numAry);
    }
    static void init(int a[]) {
        for(int i = a.length - 1; i > 0; i--) {
            a[i] += a[i - 1];
        }
    }
    static void prnt(int a[]) {
        int sum = 0;
        for(int i : a)
            sum += i;
        System.out.print(sum);
    }
}
```

**답 :**

**문제 11** 다음은 <수강> 테이블에 대해 김사랑에게 부여된 UPDATE 권한과 김사랑이 다른 사람에게 UPDATE 권한을 부여할 수 있는 권한, 그리고 김사랑이 다른 사람에게 부여한 UPDATE 권한까지 모두 취소하는 SQL문이다. 괄호(①, ②)에 들어갈 가장 적합한 명령어를 쓰시오. (5점)

( ① ) UPDATE ON 수강 FROM 김사랑 ( ② );

**답**

- ① :
- ② :

**문제 12** Sun Microsystems에서 1997년 2월 개발한 DBMS 접속 기술로, Java 언어로 다양한 종류의 데이터베이스에 접속하고 SQL문을 수행할 때 사용되는 표준 API가 무엇인지 쓰시오. (5점)

**답 :**

**문제 13** 서버 개발 프레임워크에 대한 다음 설명에서 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오. (5점)

서버 개발 프레임워크는 서버 프로그램 개발 시 다양한 네트워크 설정, 요청 및 응답 처리, 아키텍처 모델 구현 등을 손쉽게 처리할 수 있도록 클래스나 인터페이스를 제공하는 소프트웨어로, 대표적인 서버 개발 프레임워크의 종류는 다음과 같다.

|             |                                                                                                  |
|-------------|--------------------------------------------------------------------------------------------------|
| Node.js     | JavaScript를 기반으로 만들어진 프레임워크로, 비동기 입·출력 처리와 이벤트 위주의 높은 처리 성능을 갖고 있어 실시간으로 입·출력이 빈번한 애플리케이션에 적합하다. |
| ( ① )       | JAVA를 기반으로 만들어진 프레임워크로, 전자정부 표준 프레임워크의 기반 기술로 사용되고 있다.                                           |
| ( ② )       | Python을 기반으로 만들어진 프레임워크로, 컴포넌트의 재사용과 플러그인화를 강조하여 신속한 개발이 가능하도록 지원한다.                             |
| Codeigniter | PHP를 기반으로 만들어진 프레임워크로, 인터페이스가 간편하며 서버 자원을 적게 사용한다.                                               |

**답**

- ① :
- ② :

**문제 14** 기획 단계의 초기에 키노트(Keynote), 스케치(Sketch)와 같은 툴을 이용하여 페이지에 대한 개략적인 레이아웃이나 UI 요소 등에 대한 뼈대를 설계하기 위해 제작하는 것으로, 각 페이지의 영역 구분, 콘텐츠, 텍스트 배치 등을 화면 단위로 제작한다. 개발자나 디자이너 등이 레이아웃을 협의하거나 현재 진행 상태 등을 공유하는데 사용되는 UI 설계 도구를 쓰시오. (5점)

**답 :**

**문제 15** 프로세스는 일반적으로 프로세서에 의해 처리되는 사용자 프로그램, 시스템 프로그램 등을 의미한다. 다음의 프로세스 상태에 대한 설명에서 괄호(①~③)에 들어갈 알맞은 용어를 쓰시오. (5점)

|           |                                                                         |
|-----------|-------------------------------------------------------------------------|
| ( ① )     | 작업을 처리하기 위해 사용자가 작업을 시스템에 제출한 상태이다.                                     |
| ( ② )     | 제출된 작업이 스펴 공간인 디스크의 할당 위치에 저장된 상태이다.                                    |
| 준비(Ready) | 프로세스가 프로세서를 할당받기 위해 기다리고 있는 상태이다.                                       |
| 실행(Run)   | 준비상태 큐에 있는 프로세스가 프로세서를 할당받아 실행되는 상태이다.                                  |
| ( ③ )     | 프로세스에 입·출력 처리가 필요하면 현재 실행 중인 프로세스가 중단되고, 입·출력 처리가 완료될 때까지 대기하고 있는 상태이다. |
| 종료(Exit)  | 프로세스의 실행이 끝나고 프로세스 할당이 해제된 상태이다.                                        |

**답**

- ① :
- ② :
- ③ :

**문제 16** 서비스 거부 공격의 한 유형인 스머핑(Smurfing)의 개념을 설명하시오. (5점)

**답 :**

**문제 17** OSI의 네트워크 계층에 대응하며, IP, ICMP, ARP 등이 사용되고 데이터 전송을 위한 주소 지정, 경로 설정 기능 등을 제공하는 TCP/IP의 계층을 쓰시오. (5점)

**답 :**

**문제 18** 다음 설명에 가장 부합하는 용어를 쓰시오. (5점)

소프트웨어의 개발 과정에서 소프트웨어의 변경 사항을 관리하기 위해 개발된 일련의 활동이다.

- 소프트웨어 개발의 전 단계에 적용되는 활동이며, 유지보수 단계에서도 수행된다.
- 소프트웨어 개발의 전체 비용을 줄이고, 개발 과정의 여러 방해 요인이 최소화되도록 보증하는 것을 목적으로 한다.
- 관리 대상에 이름과 관리 번호를 부여하고, 계층(Tree) 구조로 구분하여 수정 및 추적이 용이하게 한다.
- 소프트웨어 업그레이드나 유지 보수 과정에서 생성된 다른 항목을 관리하고, 이를 위해 특정 절차와 도구(Tool)를 결합시킨다.
- 식별된 항목에 대한 변경 요구를 검토하여 현재의 기준선(Base Line)이 잘 반영될 수 있도록 조정한다.
- 기준선의 무결성을 평가하기 위해 확인, 검증, 검열 과정을 통해 공식적으로 승인한다.
- 식별, 통제, 감사 작업의 결과를 기록·관리하고 보고서를 작성한다.

**답 :**

**문제 19** DBMS(DataBase Management System)는 사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성 해주고, 데이터베이스를 관리해 주는 소프트웨어이다. DBMS의 필수 기능 3가지를 쓰시오.

**답 :**

**문제 20** 컴퓨터 시스템의 자원들을 효율적으로 관리하며, 사용자가 컴퓨터를 편리하고 효과적으로 사용할 수 있도록 환경을 제공하는 여러 프로그램의 모임을 운영체제(OS)라고 한다. 이러한 운영체제의 성능을 평가하는 기준 중 신뢰도(Reliability)의 개념을 설명하시오. (5점)

**답 :**



## 2회 실전 모의고사 정답 및 해설

### [답안 작성 방법 안내]

'운영체제(OS: Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

### [문제 1]

요구공학(Requirement Engineering)

### [문제 2]

BCNF(Boyce-Codd Normal Form)

#### 해설

〈동아리〉 릴레이션에는 다음과 같은 함수적 종속이 존재합니다.

- (회원번호, 소속) → 강사명
- 강사명 → 소속

〈동아리〉 릴레이션에는 결정자이지만 후보키가 아닌 속성이 존재합니다. 즉 함수적 종속 '강사명 → 소속'에서 강사명은 결정자이지만 〈동아리〉 릴레이션에서 후보키가 아닙니다. 왜냐하면 강사명 '김범수'는 항상 소속이 '축구'이고, 강사명 '나경락'은 항상 소속이 '테니스'로 강사명이 소속을 종속하지만 강사명 속성에는 중복된 값이 있어 강사명 속성만으로는 레코드를 유일하게 식별할 수 없기 때문입니다. 그러므로 〈동아리〉 릴레이션은 BCNF가 아닙니다. BCNF는 릴레이션의 결정자가 모두 후보키가 되는 정규형입니다.

- 문제에 제시된 내용은 〈동아리〉 릴레이션에서 BCNF를 만족하지 못하게 하는 속성인 결정자이면서 후보키가 아닌 속성(즉 강사명 → 소속)을 분리하여 〈회원〉 릴레이션과 〈강사〉 릴레이션으로 무손실 분해하는 정규화 과정입니다.

### [문제 3]

트리거(Trigger)

### [문제 4]

12

#### 해설

문제의 코드는 피보나치 수를 누적하여 출력하는 프로그램입니다. 피보나치 수는 0과 1로 시작하며, 다음 피보나치 수는 바로 앞의 두 피보나치 수의 합이 되어 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ... 과 같이 나열되는 수열입니다. 코드는 세 번째 피보나치 수까지 누적한 값을 초기값으로 하고, 반복문을 통해 피보나치 수를 3회 더 진행하여 추가로 누적했으므로, 총 6번째 피보나치 수(5)까지의 값들의 합이 결과로 나오게 됩니다.

```

#include <stdio.h>
main() {
    int a, b, c, sum;
    ① a = b = 1;
    ② sum = a + b;
    ③ for (int i = 3; i <= 5; i++) {
    ④     c = a + b;
    ⑤     sum += c;
    ⑥     a = b;
    ⑦     b = c;
    }
}

```

```
⑧ printf("%d", sum);
}
```

- ① b에 1을 저장하고, a에 b의 값을 저장한다. 즉 a와 b에 1을 저장한다.
- ② a와 b의 합을 sum에 저장한다.
- ③ 반복 변수 i가 3에서 시작하여 1씩 증가하면서 5보다 작거나 같은 동안 ④~⑦번을 반복 수행한다.
- ④ a와 b의 합을 c에 저장한다.
- ⑤ 'sum=sum+c;'과 동일하다. sum에 c의 값을 누적한다.
- ⑥ a의 값을 b의 값으로 치환한다.
- ⑦ b의 값을 c의 값으로 치환한다.
- ⑧ sum의 값을 출력한다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

| a | b | c | sum | i | 출력 |
|---|---|---|-----|---|----|
| 1 | 1 | 2 | 2   | 3 | 12 |
| 1 | 2 | 3 | 4   | 4 |    |
| 2 | 3 | 5 | 7   | 5 |    |
| 3 | 5 |   | 12  | 6 |    |

#### [문제 5]

00001011

※ 답안 작성 시 주의 사항

Java에서는 배열 선언 시 모든 요소가 0으로 초기화됩니다. 그러므로 a 배열 출력 시 값이 저장되지 않은 요소에는 0이 출력됩니다.

#### 해설

문제의 코드는 10진수 11을 2진수로 변환하여 배열에 저장한 후 출력하는 프로그램입니다. 10진수를 2진수로 변환하려면 10진수를 2로 나누어 나머지를 구한 후 저장하고, 다시 몫을 2로 나누어 나머지를 구해 저장하는 과정을 반복합니다. 몫이 0이 될 때까지 이 작업을 반복한 후 마지막에 구한 나머지부터 거꾸로 출력합니다.

```
public class Test {
    public static void main(String[] args) {
        ① int a[] = new int[8];
        ② int d = 11, n = 0;
        ③ do {
        ④     a[n++] = d % 2;
        ⑤     d /= 2;
        ⑥ } while (d > 0);
        ⑦ for (n = 7; n >= 0; n--)
        ⑧     System.out.printf("%d", a[n]);
        }
    }
}
```

- ① 8개의 요소를 갖는 정수형 배열 a를 선언한다.
- ② 정수형 변수 d와 n을 선언하고, 각각 11과 0으로 초기화한다.
- ③ do~while 반복문의 시작점이다.
- ④ n++은 후지 연산이므로 a[n]에 d를 2로 나눈 나머지를 저장한 후 n의 값을 1 증가시킨다.



- ⑤ d에 d를 2로 나눈 값을 저장한다.  
 ※ C언어에서 정수 나눗셈은 결과도 정수이다. 예) 정수형 변수 a가 10인 경우 :  $a/3=3$ ,  $a/7=1$   
 ⑥ d가 0보다 큰 동안 ④, ⑤번을 반복 수행한다.  
 ⑦ 반복 변수 n이 7에서 시작하여 1씩 감소하면서 0보다 크거나 같은 동안 ⑥번을 반복하여 수행한다.  
 ⑧ a[n]의 값을 출력한다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

| d  | n  | d%2 | a[n] | 배열 a                                                                                                                         | 출력       |
|----|----|-----|------|------------------------------------------------------------------------------------------------------------------------------|----------|
| 11 | 0  | 1   | 1    | a[0] a[1] a[2] a[3] a[4] a[5] a[6] a[7]<br> |          |
| 5  | 1  | 1   | 1    |                                                                                                                              |          |
| 2  | 2  | 0   | 0    |                                                                                                                              |          |
| 1  | 3  | 1   | 1    |                                                                                                                              |          |
| 0  | 4  |     |      |                                                                                                                              |          |
|    | 7  |     | 0    | a[0] a[1] a[2] a[3] a[4] a[5] a[6] a[7]<br> | 00001011 |
|    | 6  |     | 0    |                                                                                                                              |          |
|    | 5  |     | 0    |                                                                                                                              |          |
|    | 4  |     | 0    |                                                                                                                              |          |
|    | 3  |     | 1    |                                                                                                                              |          |
|    | 2  |     | 0    |                                                                                                                              |          |
|    | 1  |     | 1    |                                                                                                                              |          |
|    | 0  |     | 1    |                                                                                                                              |          |
|    | -1 |     |      |                                                                                                                              |          |

#### [답안 작성 방법 안내]

'운영체제(OS; Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

#### [문제 6]

비트맵 인덱스(Bitmap Index)

#### [문제 7]

API(Application Programming Interface)

#### [문제 8]

① 명세 기반 테스트    ② 구조 기반 테스트    ③ 경험 기반 테스트

#### [문제 9]

코드 커버리지(Code Coverage)

## [문제 10]

25

## 해설

```

public class Test {
    public static void main(String[] args) {
        ① int numArry[] = { 1, 2, 3, 4, 5 };
        ② init(numArry);
        ⑥ prnt(numArry);
    }
    static void init(int a[] ) {
        ③ for(int i = a.length - 1; i > 0; i--) {
        ④     a[i] += a[i - 1];
        ⑤     }
    }
    static void prnt(int a[] ) {
        ⑦ int sum = 0;
        ⑧ for(int i : a)
        ⑨     sum += i;
        ⑩ System.out.print(sum);
    }
}

```

- ① 배열을 선언할 때 사용할 개수를 생략하고 초기값을 지정하면, 초기값으로 지정된 값의 수와 같은 크기의 배열이 선언된다.

|            | numArry[0] | numArry[1] | numArry[2] | numArry[3] | numArry[4] |
|------------|------------|------------|------------|------------|------------|
| 배열 numArry | 1          | 2          | 3          | 4          | 5          |

- ② numArry를 인수로 하여 init( ) 메소드를 호출한다. ③번으로 이동한다. 인수로 배열의 이름을 지정하면 배열의 시작 주소가 인수로 전달된다.  
 ③ init( ) 메소드의 시작점이다. 정수형 배열 a가 main( ) 메소드에서 전달한 배열 numArry의 시작 주소를 받는다.  
 ④ 반복 변수 i가 a의 길이에서 1을 뺀 값인 4에서 시작하여 1씩 감소하면서 0보다 큰 동안 ⑥번을 반복 수행한다.  
 • length : length는 배열 클래스의 속성으로 배열 요소의 개수가 저장되어 있다. 배열 a는 5개의 요소를 가지므로 a.length는 5를 가지고 있다.  
 ⑤ a[i]에 a[i-1]의 값을 누적한다. 반복이 모두 종료되면 ⑥번으로 이동한다.

| i | 배열 a |      |      |      |      |
|---|------|------|------|------|------|
| 4 |      |      |      |      |      |
| 3 | a[0] | a[1] | a[2] | a[3] | a[4] |
| 2 | 1    | 2    | 3    | 4    | 5    |
| 1 |      | 3    | 5    | 7    | 9    |
| 0 |      |      |      |      |      |

- ⑥ numArry를 인수로 하여 prnt( ) 메소드를 호출한다. ⑦번으로 이동한다. 인수로 배열의 이름을 지정하면 배열의 시작 주소가 인수로 전달된다.  
 ⑦ prnt( ) 메소드의 시작점이다. 정수형 배열 a가 main( ) 메소드에서 전달한 배열 numArry의 시작 주소를 받는다.  
 ⑧ 정수형 변수 sum을 선언하고 0으로 초기화한다.  
 ⑨ 배열 a의 요소 수만큼 ⑩번을 반복 수행한다.  
 • int i : 배열 a의 각 요소가 할당될 변수를 선언한다.  
 • a : 배열의 이름을 적어준다. 배열이 5개의 요소를 가지므로 각 요소를 i에 할당하면서 ⑩번을 5회 수행한다.  
 ⑩ sum에 i의 값을 누적한다.

| sum | i | 배열 a |      |      |      |      |
|-----|---|------|------|------|------|------|
| 0   | 1 |      |      |      |      |      |
| 1   | 3 |      |      |      |      |      |
| 4   | 5 |      |      |      |      |      |
| 9   | 7 | a[0] | a[1] | a[2] | a[3] | a[4] |
| 16  | 9 | 1    | 3    | 5    | 7    | 9    |
| 25  |   |      |      |      |      |      |

❶ sum의 값을 출력하고 main() 메소드로 돌아가 프로그램을 종료한다.

[문제 11]

① REVOKE    ② CASCADE

**해설**

<수강> 테이블에 대한 UPDATE 권한을 취소하면 다른 사람에게 UPDATE 권한을 부여할 수 있는 권한도 함께 취소되기 때문에 'GRANT OPTION FOR'는 생략됩니다. 자신에게 권한이 없어지면 해당 권한을 다른 사람에게 부여할 수 없습니다.

[답안 작성 방법 안내]

'운영체제(OS; Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

[문제 12]

JDBC(Java DataBase Connectivity)

[문제 13]

① Spring    ② Django

[문제 14]

와이어프레임(Wireframe)

[문제 15]

① 제출(Submit)    ② 접수(Hold)    ③ 대기(Wait) 또는 블록(Block)

[문제 16]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

IP나 ICMP의 특성을 악용하여 엄청난 양의 데이터를 한 사이트에 집중적으로 보냄으로써 네트워크 또는 시스템의 상태를 불능으로 만드는 공격 방법이다.

[문제 17]

인터넷 계층(Internet Layer)

[문제 18]

형상 관리(SCM; Software Configuration Management)

[문제 19]

정의(Definition) 기능, 조작(Manipulation) 기능, 제어(Control) 기능

[문제 20]

신뢰도는 시스템이 주어진 문제를 정확하게 해결하는 정도를 의미한다.



**문제 1** 요구사항 개발 프로세스는 개발 대상에 대한 요구사항을 분석하여 검증하는 일련의 구조화된 활동이다. 요구사항 개발 프로세스 4단계를 순서대로 나열하시오. (5점)

**답 :**

**문제 2** 데이터베이스(Database)는 특정 조직의 업무를 수행하는 데 필요한 상호 관련된 데이터들의 모임으로 다음과 같이 정의할 수 있다. 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오. (5점)

|         |                                                        |
|---------|--------------------------------------------------------|
| ( ① )   | 조직의 고유한 업무를 수행하는 데 존재 가치가 확실하고 없어서는 안 될 반드시 필요한 데이터이다. |
| 통합된 데이터 | 검색의 효율성을 위해 중복이 최소화된 데이터의 모임이다.                        |
| ( ② )   | 컴퓨터가 접근할 수 있는 저장 매체에 저장된 데이터이다.                        |
| 공용 데이터  | 여러 응용 시스템들이 공동으로 소유하고 유지하는 데이터이다.                      |

**답**

- ① :
- ② :

**문제 3** 다음이 설명하는 데이터베이스 관련 용어를 쓰시오. (5점)

검증하는 다량의 데이터를 분석하여 정보화하고 이를 여러 계층의 사용자들이 효율적으로 사용할 수 있도록 한 데이터베이스이다. 신속 정확한 의사결정으로 경쟁력을 확보하기 위하여 의사결정용 데이터베이스의 필요성으로 인해 등장하였으며, 다양한 원본 데이터베이스로부터 정제되어 추출된 데이터만을 저장하고 필요한 인덱스를 생성한 후 데이터의 다차원 분석 도구로 이를 분석하여 의사결정에 필요한 자료를 얻는다.

**답 :**



**문제 4** 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
#include <stdio.h>
main() {
    int x = 7, y = 10, z;
    z = bitCal(x, y);
    printf("%d", z);
}

int bitCal(x, y) {
    if (x < y)
        return x | y;
    else
        return x & y;
}
```

**답 :**

**문제 5** 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오. (5점)

데이터베이스에서 ( )은 대용량의 테이블이나 인덱스를 작은 논리적 단위인 ( )으로 나누는 것을 말한다. 대용량 DB의 경우 중요한 몇 개의 테이블에만 집중되어 데이터가 증가되므로, 이런 테이블들을 작은 단위로 나눠 분산시키면 성능 저하를 방지할 뿐만 아니라 데이터 관리도 쉬워진다.

**답 :**

**문제 6** 애플리케이션 간 연계, 데이터 변환, 웹 서비스 지원 등 표준 기반의 인터페이스를 제공하는 솔루션으로, 애플리케이션 통합 측면에서 EAI와 유사하지만 애플리케이션 보다는 서비스 중심의 통합을 지향하며, 특정 서비스에 국한되지 않고 범용적으로 사용하기 위하여 애플리케이션과의 결합도(Coupling)를 약하게(Loosely) 유지하는 것은 무엇인지 쓰시오. (5점)

**답 :**

**문제 7** 인터넷에서 전 세계의 비즈니스 업체 목록에 자신의 목록을 등록하기 위한 XML 기반의 규격으로, WSDL을 등록하여 서비스와 서비스 제공자를 검색하고 접근하는데 사용되는 웹 서비스의 구성 요소를 쓰시오. (5점)

**답 :**

**문제 8** 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오. (5점)

( )는 소프트웨어의 성능을 향상시키거나 시스템의 수정 및 재사용, 유지 관리 등이 용이하도록 시스템의 기능들을 모듈 단위로 분해하는 것을 의미한다. ( )의 목표는 모듈 간 결합도의 최소화, 응집도의 최대화이다.

**답 :**

**문제 9** 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
public class Test {
    public static void recursive(int a) {
        if (a <= 1)
            return;
        System.out.print(a + " ");
        recursive(a / 2);
    }
    public static void main(String[] args) {
        recursive(20);
    }
}
```

**답 :**

**문제 10** 운영체제(Operating System)는 컴퓨터 시스템의 자원들을 효율적으로 관리하며, 사용자가 컴퓨터를 편리하고 효과적으로 사용할 수 있도록 환경을 제공하는 여러 프로그램의 모임이다. 운영체제의 목적 4가지를 쓰시오. (5점)

답 :

**문제 11** 다음은 프로토타이핑의 종류를 설명한 것이다. 괄호(①, ②)에 들어갈 가장 적합한 종류를 쓰시오. (5점)

|       |                                                                                                            |
|-------|------------------------------------------------------------------------------------------------------------|
| ( ① ) | 아날로그적인 방법으로, 스케치, 그림, 글 등을 이용하여 손으로 직접 작성하며, 제작 기간이 짧은 경우, 제작 비용이 적을 경우, 업무 협의가 빠를 경우 사용한다.                |
| ( ② ) | 파워포인트, 아크로벳, 비지오, 옴니그래플 등과 같은 프로그램을 사용하여 작성하는 방법으로, 재사용이 필요한 경우, 산출물과 비슷한 효과가 필요한 경우, 숙련된 전문가가 있을 경우 사용한다. |

답

- ① :
- ② :

**문제 12** 애플리케이션 테스트의 종류 중 강도 테스트에 대해 간략히 서술하시오. (5점)

답 :

**문제 13** 웹페이지에 악의적인 스크립트를 삽입하여 방문자들의 정보를 탈취하거나, 비정상적인 기능 수행을 유발하는 공격 기법을 가리키는 용어를 쓰시오. (5점)

답 :

**문제 14** 인수 테스트는 개발한 소프트웨어가 사용자의 요구사항을 충족하는지에 중점을 두고 테스트하는 방법으로, 다음과 같이 6가지 종류로 구분해서 테스트 한다. 괄호(①, ②)에 들어갈 알맞은 테스트를 쓰시오. (5점)

|            |                                                                                                                                              |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 운영상 인수 테스트 | 시스템 관리자가 시스템 인수 시 수행하는 테스트 기법으로, 백업/복원 시스템, 재난 복구, 사용자 관리, 정기 점검 등을 확인한다.                                                                    |
| ( ① )      | 개발자의 장소에서 사용자가 개발자 앞에서 행하는 테스트 기법으로, 테스트는 통제된 환경에서 행해지며, 오류와 사용상의 문제점을 사용자와 개발자가 함께 확인하면서 기록한다.                                              |
| 사용자 인수 테스트 | 사용자가 시스템 사용의 적절성 여부를 확인한다.                                                                                                                   |
| 규정 인수 테스트  | 소프트웨어가 정부 지침, 법규, 규정 등 규정에 맞게 개발되었는지 확인한다.                                                                                                   |
| ( ② )      | 선정된 최종 사용자가 여러 명의 사용자 앞에서 행하는 테스트 기법으로, 실업무를 가지고 사용자가 직접 테스트하는 것으로, 개발자에 의해 제어되지 않은 상태에서 테스트가 행해지며, 발견된 오류와 사용상의 문제점을 기록하고 개발자에게 주기적으로 보고한다. |
| 계약 인수 테스트  | 계약상의 인수/검수 조건을 준수하는지 여부를 확인한다.                                                                                                               |

**답**

- ① :
- ② :

**문제 15** 다음 설명의 괄호에 공통으로 들어갈 암호 알고리즘 관련 용어를 쓰시오. (5점)

- ( )는 임의의 길이의 입력 데이터나 메시지를 고정된 길이의 값이나 키로 변환하는 것으로, 데이터의 암호화, 무결성 검증을 위해 사용될 뿐만 아니라 정보보호의 다양한 분야에서 활용된다.
- ( ) 알고리즘을 ( ) 함수라고 부르며, ( ) 함수로 변환된 값이나 키를 ( )값 또는 ( )키라고 부른다.

**답 :**

**문제 16** UNIX 명령어 중 파일 시스템을 검사 및 보수하여 무결성을 검사하는 명령어를 쓰시오. (5점)

**답 :**



**문제 17** 교착상태 예방 기법(Prevention)은 교착상태가 발생하지 않도록 사전에 시스템을 제어하는 방법으로, 교착상태 발생의 네 가지 조건 중에서 어느 하나를 부정함으로써 수행된다. 다음 괄호(①~③)에 들어갈 알맞은 용어를 쓰시오. (5점)

|          |                                                                                         |
|----------|-----------------------------------------------------------------------------------------|
| ( ① ) 부정 | 한 번에 여러 개의 프로세스가 공유 자원을 사용할 수 있도록 한다.                                                   |
| ( ② ) 부정 | 프로세스가 실행되기 전 필요한 모든 자원을 할당하여 프로세스 대기를 없애거나 자원이 점유되지 않은 상태에서 자원 요구하도록 한다.                |
| 비선점 부정   | 자원을 점유하고 있는 프로세스가 다른 자원을 요구할 때 점유하고 있는 자원을 반납하고, 요구한 자원을 사용하기 위해 기다리게 한다.               |
| ( ③ ) 부정 | 자원을 선형 순서로 분류하여 고유 번호를 할당하고, 각 프로세스는 현재 점유한 자원의 고유 번호보다 앞이나 뒤 어느 한쪽 방향으로만 자원을 요구하도록 한다. |

**답**

- ① :
- ② :
- ③ :

**문제 18** 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
#include <stdio.h>
void func(int, int);
main() {
    int p, q;
    p = 10;
    q = p++ > 10 ? p + 3 : p - 3;
    func(p, q);
}
void func(int x, int y) {
    if (x > y)
        while (x > 0) {
            printf("%d ", x);
            x = x / 2;
        }
    else
        while (y > 0) {
            printf("%d ", y);
            y = y / 2;
        }
}
```

**답 :**

**문제 19** 다음은 가상 기억장치(Virtual Memory)에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오. (5점)

보조기억장치의 일부를 주기억장치 처럼 사용하는 가상 기억장치는 용량이 작은 주기억장치를 마치 큰 용량을 가진 것처럼 사용하는 것으로, 일반적인 구현 방법에는 블록의 종류에 따라 다음과 같이 나눌 수 있다.

|       |                                                                                                                              |
|-------|------------------------------------------------------------------------------------------------------------------------------|
| ( ① ) | 가상 기억장치에 보관되어 있는 프로그램과 주기억장치의 영역을 동일한 크기로 나눈 후 나뉜 페이지를 동일하게 나뉜 주기억장치의 영역에 적재시켜 실행하는 기법으로, 외부 단편화는 발생하지 않으나 내부 단편화는 발생할 수 있다. |
| ( ② ) | 가상 기억장치에 보관되어 있는 프로그램을 다양한 크기의 논리적인 단위로 나눈 후 주기억장치에 적재시켜 실행시키는 기법으로, 내부 단편화는 발생하지 않으나 외부 단편화는 발생할 수 있다.                      |

**답**

- ① :
- ② :

**문제 20** 데이터베이스에 테이블을 생성하고자 한다. 다음 <처리 조건>에 부합하도록 괄호(①~③)를 채워 SQL문을 완성하시오. (5점)

<처리 조건>

- 참가번호, 이름, 국가번호로 구성된 <참가자> 테이블을 정의하는 SQL 문을 작성하시오. 단, 제약 조건은 다음과 같다.
  - 참가번호는 기본키이다.
  - 이름은 중복된 값을 가질 수 없다.
  - 국가번호는 <국가목록> 테이블의 부여번호를 참조하는 외래키로 사용된다.
  - <국가목록> 테이블에서 부여번호가 변경되면 국가번호 값도 같은 값으로 변경한다.
  - 각 속성의 데이터형은 저장될 데이터에 적합하게 지정한다.
- SQL문의 문법은 ISO/IEC 9075 표준을 따른다.

<SQL문>

```
CREATE TABLE 참가자
(참가번호 INT ( ① ),
  이름 CHAR(10) ( ② ),
  국가번호 INT,
  FOREIGN KEY(국가번호) ( ③ ) 국가목록(부여번호)
  ON UPDATE CASCADE);
```

**답**

- ① :
- ② :
- ③ :



## 3회 실전 모의고사 정답 및 해설

### [답안 작성 방법 안내]

‘운영체제(OS: Operation System)’처럼 한글과 영문으로 제시되어 있는 경우 ‘운영체제’, ‘OS’, ‘Operation System’ 중 1가지만 쓰면 됩니다.

### [문제 1]

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

### [문제 2]

① 운영 데이터(Operational Data) ② 저장된 데이터(Stored Data)

### [문제 3]

데이터 웨어하우스(Data Warehouse)

### [문제 4]

15

### 해설

```

#include <stdio.h>
main() {
  ①   int x = 7, y = 10, z;
  ②⑥  z = bitCal(x, y);
  ⑦   printf("%d", z);
}

③ int bitCal(x, y) {
  ④   if (x < y)
  ⑤     return x | y;
    else
    return x & y;
}

```

④번 조건식이 거짓일 경우 실행할 문장의 시작점이다.  
x와 y의 값을 &(비트 and) 연산한 결과를 main 함수로 반환한다.

- ① 정수형 변수 x, y, z를 선언하고, x와 y의 값을 각각 7과 10으로 초기화한다.
- ② x와 y의 값 7과 10을 인수하여 bitCal 함수를 호출한 다음 돌려받은 값을 z에 저장한다.
- ③ 리턴값이 정수인 bitCal 함수의 시작점이다. ②번에서 7과 10을 전달받았으므로 x는 7, y는 10이다.
- ④ x가 y보다 작으면 ⑤번 문장을 수행하고, 아니면 else 다음 문장을 수행한다. x(7)가 y(10)보다 작으므로 ⑤번으로 이동한다.
- ⑤ x와 y의 값을 |(비트 or) 연산한 결과를 호출한 곳(main 함수)으로 반환한다.

C 언어에서 정수형 변수는 4바이트이므로 각 변수의 값을 4바이트 2진수로 변환한 다음 각 비트를 연산한다.

```

  7 = 0000 0000 0000 0000 0000 0000 0000 0111
 10 = 0000 0000 0000 0000 0000 0000 0000 1010
  |
  | 0000 0000 0000 0000 0000 0000 0000 1111 = 10진수로 15이다.

```

- ⑥ bitCal( ) 함수로부터 반환받은 값 15를 z에 저장한다.
- ⑦ z의 값을 출력한다.

**[답안 작성 방법 안내]**

'운영체제(OS; Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

**[문제 5]**

파티션(Patition)

**[문제 6]**

ESB(Enterprise Service Bus)

**[문제 7]**

UDDI(Universal Description, Discovery and Integration)

**[문제 8]**

모듈화(Modularity)

**[문제 9]**

20 10 5 2

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에逗를 넣어 20, 10, 5, 2로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

**해설**

모든 Java 프로그램은 반드시 main() 메소드부터 시작해야 한다.

```
public static void main(String[] args) {
  ❶ recursive(20);
}
```

❶ 20을 인수로 recursive() 메소드를 호출한다.

①회

```
2 public static void recursive(int a) {
3   if (a <= 1)
4     return;
5   System.out.print(a + " ");
6   recursive(a / 2);
7 }
```

❷ recursive() 메소드가 호출될 때 20을 전달받았으므로 a는 20이다.

❸ a가 1보다 작거나 같지 않으므로 ❹번으로 이동한다.

❹ a의 값 20을 출력하고, 이어서 공백 한 칸을 출력한다.

결과 20

⑤ a를 2로 나눈 값 10을 인수로 recursive() 메소드를 호출한다.

②회

```
⑥ public static void recursive(int a) {
⑦     if (a <= 1)
        return;
⑧     System.out.print(a + " ");
⑨     recursive(a / 2);
}
```

⑥ recursive() 메소드가 호출될 때 10을 전달받았으므로 a는 10이다.

⑦ a가 1보다 작거나 같지 않으므로 ⑧번으로 이동한다.

⑧ a의 값 10을 출력하고, 이어서 공백 한 칸을 출력한다.

결과 20 10

⑨ a를 2로 나눈 값 5를 인수로 recursive() 메소드를 호출한다.

③회

```
⑩ public static void recursive(int a) {
⑪     if (a <= 1)
        return;
⑫     System.out.print(a + " ");
⑬     recursive(a / 2);
}
```

⑩ recursive() 메소드가 호출될 때 5를 전달받았으므로 a는 5이다.

⑪ a가 1보다 작거나 같지 않으므로 ⑫번으로 이동한다.

⑫ a의 값 5를 출력하고, 이어서 공백 한 칸을 출력한다.

결과 20 10 5

⑬ a를 2로 나눈 값 2를 인수로 recursive() 메소드를 호출한다.

※ C, Java에서 정수 나눗셈은 결과도 정수이다. ㉠ 정수형 변수 a가 11인 경우 :  $a / 2 = 5$ ,  $a / 7 = 1$

④회

```
⑭ public static void recursive(int a) {
⑮     if (a <= 1)
        return;
⑯     System.out.print(a + " ");
⑰     recursive(a / 2);
}
```

⑭ recursive() 메소드가 호출될 때 2를 전달받았으므로 a는 2이다.

⑮ a가 1보다 작거나 같지 않으므로 ⑯번으로 이동한다.

⑯ a의 값 2를 출력하고, 이어서 공백 한 칸을 출력한다.

결과 20 10 5 2

⑰ a를 2로 나눈 값 1을 인수로 recursive() 메소드를 호출한다.

⑤회

```
⑱ public static void recursive(int a) {
⑲     if (a <= 1)
        return;
⑳     System.out.print(a + " ");
        recursive(a / 2);
}
```

㉓ recursive( ) 메소드가 호출될 때 1을 전달받았으므로 a는 1이다.

㉔ a가 1보다 작거나 같으므로 ㉑번으로 이동한다.

㉑ 메소드를 종료하고 반환값 없이 제어를 ㉔회 recursive(a / 2) 메소드를 호출했던 곳으로 옮긴다.

㉔회

```
public static void recursive(int a) {
    if (a <= 1)
        return;
    System.out.print(a + " ");
    ㉑ recursive(a / 2);
}
```

㉑ 메소드를 종료하고 반환값 없이 제어를 ㉓회 recursive(a / 2) 메소드를 호출했던 곳으로 옮긴다.

㉓회

```
public static void recursive(int a) {
    if (a <= 1)
        return;
    System.out.print(a + " ");
    ㉑ recursive(a / 2);
}
```

㉑ 메소드를 종료하고 반환값 없이 제어를 ㉒회 recursive(a / 2) 메소드를 호출했던 곳으로 옮긴다.

㉒회

```
public static void recursive(int a) {
    if (a <= 1)
        return;
    System.out.print(a + " ");
    ㉑ recursive(a / 2);
}
```

㉑ 메소드를 종료하고 반환값 없이 제어를 ㉑회 recursive(a / 2) 메소드를 호출했던 곳으로 옮긴다.

㉑회

```
public static void recursive(int a) {
    if (a <= 1)
        return;
    System.out.print(a + " ");
    ㉑ recursive(a / 2);
}
```

㉑ 메소드를 종료하고 반환값 없이 제어를 처음 recursive(20) 메소드를 호출했던 main( ) 메소드로 옮긴다.

```
public static void main(String[] args) {
    ㉑ recursive(20);
}
```

㉑ 이후 수행할 코드가 없으므로 프로그램을 종료한다.

[답안 작성 방법 안내]

‘운영체제(OS: Operation System)’처럼 한글과 영문으로 제시되어 있는 경우 ‘운영체제’, ‘OS’, ‘Operation System’ 중 1가지만 쓰면 됩니다.

[문제 10]

처리 능력(Throughput) 향상, 사용 가능도(Availability) 향상, 신뢰도(Reliability) 향상, 반환 시간(Turn Around Time) 단축

[문제 11]

① 페이퍼 프로토타입(Paper Prototype) ② 디지털 프로토타입(Digital Prototype)

[문제 12]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

시스템에 과도한 정보량이나 빈도 등을 부과하여 과부하 시에도 소프트웨어가 정상적으로 실행되는지를 확인하는 테스트이다.

[문제 13]

크로스사이트 스크립팅(XSS)

[문제 14]

① 알파 테스트 ② 베타 테스트

[문제 15]

해시(Hash)

[문제 16]

fsck

[문제 17]

① 상호 배제(Mutual Exclusion) ② 점유 및 대기(Hold and Wait) ③ 환형 대기(Circular Wait)

[문제 18]

11 5 2 1

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 코마를 넣어 11, 5, 2, 1로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

해설

```
#include <stdio.h>
void func(int, int);
main(){
    ① int p, q;
    ② p = 10;
    ③ q = p++ > 10 ? p + 3 : p - 3;
    ④ func(p, q);
}
⑤ void func(int x, int y){
    ⑥ if (x > y)
    ⑦ while (x > 0) {
```

```

8      printf("%d ", x);
9      x = x / 2;
      }
10     else
11         while (y > 0) {
12             printf("%d ", y);
13             y = y / 2;
        }
    }

```

- ① 정수형 변수 p와 q를 선언한다.
- ② p에 10을 저장한다.
- ③ p++가 10보다 크면 p에 3을 더한 값을 저장하고, 아니면 q에 3을 뺀 값을 저장한다. p가 후치 연산이므로 조건식 'p(10) > 10'을 판별한 후 p의 값이 1 증가하여, 거짓인 경우의 연산 'p(11) - 3'을 수행한다.
- ④ p와 q의 값 1과 8을 인수로 func 함수를 호출한다.
- ⑤ 리턴값이 없는 func 함수의 시작점이다. ⑤번에서 11과 8을 전달받았으므로 x는 11, y는 8이다.
- ⑥ x가 y보다 크면 ⑦~⑨번 문장을 수행하고, 아니면 ⑪~⑬번 다음 문장을 수행한다. x(11)가 y(8)보다 크므로 ⑦번으로 이동한다.
- ⑦ x가 0보다 큰 동안 ⑧, ⑨번 문장을 반복 수행한다.
- ⑧ x의 값을 출력하고, 이어서 공백 한 칸을 출력한다.
- ⑨ x에 x를 2로 나눈 값을 저장한다.  
※ C, Java에서 정수 나눗셈은 결과도 정수이다. 예 정수형 변수 a가 11인 경우 :  $a/2=5$ ,  $a/7=1$
- ⑩ ⑥번 조건식이 거짓일 경우 실행할 문장의 시작점이다.
- ⑪ y가 0보다 큰 동안 종괄호({ }) 안의 문장을 반복 수행한다.
- ⑫ y의 값을 출력하고, 이어서 공백 한 칸을 출력한다.
- ⑬ y에 y를 2로 나눈 값을 저장한다.

## [문제 19]

- ① 페이징(Paging) 기법    ② 세그먼테이션(Segmentation) 기법

## [문제 20]

- ① PRIMARY KEY    ② UNIQUE    ③ REFERENCES

※ 답안 작성 시 주의 사항

대 · 소문자를 구분하지 않습니다. 단 스펠링이 하나라도 틀렸을 경우 부분 점수 1도 없는 오답으로 처리된다는 것을 잊지 마세요.

## 해설

```

CREATE TABLE 참가자
(참가번호 INT PRIMARY KEY,
이름 CHAR(10) UNIQUE,
국가번호 INT,
FOREIGN KEY(국가번호) REFERENCES 국가목록(부여번호)

ON UPDATE CASCADE);

```

〈참가자〉 테이블을 생성한다.  
참가번호 속성은 숫자이고, 기본키이다.  
이름 속성은 문자 10자리이고, 중복된 값을 가질 수 없다.  
국가번호 속성은 숫자이다.  
국가번호 속성은 〈국가목록〉 테이블의 부여번호 속성을 참조하는 외래키이다.  
〈국가목록〉 테이블에서 부여번호 속성이 변경되면 관련된 모든 튜플의 부여번호 속성의 값도 같은 값으로 변경한다.





**문제 1** 다음 설명에 가장 부합하는 용어를 쓰시오. (5점)

기업 내 각종 애플리케이션 및 플랫폼 간의 정보 전달, 연계, 통합 등 상호 연동이 가능하게 해주는 솔루션으로, 비즈니스 간 통합 및 연계성을 증대시켜 효율성 및 각 시스템 간의 확정성(Determinacy)을 높여 준다. 대표적인 구축 유형에는 Point-to-Point, Hub & Spoke, Message Bus, Hybrid 등이 있다.

**답 :**

**문제 2** 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

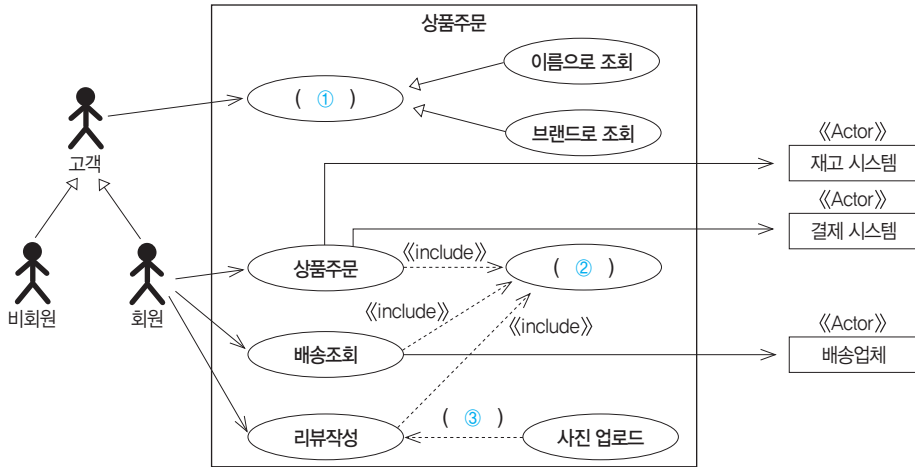
```
#include <stdio.h>
main() {
    int std[] = { 43, 29, 54, 33 };
    int r;
    for (int i = 0; i < 4; i++) {
        r = 1;
        for (int j = 0; j < 4; j++)
            if (std[i] < std[j])
                r++;
        printf("%d, %d\n", std[i], r);
    }
}
```

**답 :**

**문제 3** 데이터베이스에서의 클러스터(Cluster) 개념을 설명하시오. (5점)

**답 :**

**문제 4** 다음의 <요구사항>에 따라 작성된 유스케이스(Use Case) 다이어그램에서 괄호(①~③)에 들어갈 알맞은 용어를 쓰시오. (5점)



#### <요구사항>

- 고객은 회원과 비회원으로 구분된다.
- 고객은 이름이나 브랜드로 상품을 조회할 수 있다.
- 회원은 상품조회, 상품주문, 배송조회, 리뷰작성 기능을 사용할 수 있다.
- 비회원은 상품조회 기능만 사용할 수 있다.
- 회원이 상품주문, 배송조회, 리뷰작성 기능을 사용하려면 반드시 로그인을 수행해야 한다.
- 회원이 리뷰작성을 할 때 필요한 경우 사진 업로드 기능을 사용할 수 있다.
- 회원이 상품주문을 완료하려면 재고 시스템을 통해 재고 여부를 확인해야 한다.
- 회원이 상품주문을 완료하려면 결제 시스템을 통해 승인을 얻어야 한다.
- 회원은 배송업체를 통해 배송조회를 할 수 있다.

#### 답

- ① :
- ② :
- ③ :



**문제 5** OSI 참조 모델은 다른 시스템 간의 원활한 통신을 위해 ISO(국제표준화기구)에서 제안한 통신 규약으로, 개방형 시스템 간의 데이터 통신 시 필요한 장비 및 처리 방법 등을 7단계로 표준화하여 규정했다. OSI 7계층을 하위 계층에서 상위 계층 순으로 나열하시오. (5점)

**답 :**

**문제 6** 다음은 데이터 모델의 구성 요소에 대한 설명이다. 괄호(①~③)에 들어갈 알맞은 구성 요소를 쓰시오. (5점)

|       |                                                         |
|-------|---------------------------------------------------------|
| ( ① ) | 데이터의 가장 작은 논리적 단위로서 파일 구조상의 데이터 항목 또는 데이터 필드에 해당한다.     |
| ( ② ) | 개체 간의 관계 또는 속성 간의 논리적인 연결을 의미한다.                        |
| ( ③ ) | 데이터베이스에 표현하려는 것으로, 사람이 생각하는 개념이나 정보 단위 같은 현실 세계의 대상체이다. |

**답**

- ① :
- ② :
- ③ :

**문제 7** 디자인, 사용 방법 설명, 평가 등을 위해 와이어프레임보다 좀 더 실제 화면과 유사하게 만든 정적인 형태의 모형이며, 시각적으로만 구성 요소를 배치하는 것으로 실제로 구현되지는 않는 UI 설계 도구를 쓰시오. (5점)

**답 :**

**문제 8** 시스템 설계자가 서비스 기술자나 유지 보수 프로그램 작성자(Programmer)의 액세스 편의를 위해 시스템 보안을 제거하여 만들어 놓은 비밀 통로를 무엇이라고 하는지 쓰시오. (5점)

**답 :**

**문제 9** 다음 Java로 구현된 프로그램을 분석하여 6을 입력했을 때 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
import java.util.Scanner;
public class Test {
    public static void main(String[] args) {
        Scanner var = new Scanner(System.in);
        int n, i, sum = 0;
        n = var.nextInt();
        var.close();
        for (i = 1; i <= n; i++)
            sum += i;
        System.out.printf("%d", sum);
    }
}
```

**답 :**

**문제 10** 분산 데이터베이스(Distributed Database)는 논리적으로는 같은 시스템에 속하지만 물리적으로는 컴퓨터 네트워크를 통해 분산되어 있는 데이터베이스로, 다음과 같은 목표가 있다. 괄호(①~③)에 들어갈 알맞은 목표를 쓰시오. (5점)

|        |                                                                                             |
|--------|---------------------------------------------------------------------------------------------|
| ( ① )  | 동일한 데이터가 여러 곳에 중복되어 있더라도 사용자는 마치 하나의 데이터만 존재하는 것처럼 사용할 수 있고, 시스템은 자동으로 여러 데이터에 대한 작업을 수행한다. |
| ( ② )  | 트랜잭션, DBMS, 네트워크, 컴퓨터 장애에도 불구하고 트랜잭션은 정확하게 수행된다.                                            |
| ( ③ )  | 분산 데이터베이스와 관련된 다수의 트랜잭션들이 동시에 실행되더라도 그 트랜잭션들의 수행 결과는 서로 영향을 받지 않는다.                         |
| 위치 투명성 | 접근하려는 데이터베이스의 실제 위치를 알 필요 없이 단지 데이터베이스의 논리적인 명칭만으로 접근할 수 있다.                                |

**답**

- ① :
- ② :
- ③ :

**문제 11** 통합 테스트(Integration Test)와 관련된 다음 설명에서 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오. (5점)

통합 테스트는 단위 테스트가 끝난 모듈을 통합하는 과정에서 발생하는 오류 및 결함을 찾는 테스트 기법으로 종류는 다음과 같다.

- 빅뱅 통합 테스트는 모듈 간의 상호 인터페이스를 고려하지 않고 단위 테스트가 끝난 모듈을 한꺼번에 결합시켜 테스트하는 방법이다.
- ( ① ) 통합 테스트는 프로그램의 상위 모듈에서 하위 모듈 방향으로 통합하면서 테스트하는 기법이다.
- ( ② ) 통합 테스트는 프로그램의 하위 모듈에서 상위 모듈 방향으로 통합하면서 테스트하는 기법이다.
- 혼합식 통합 테스트는 하위 수준에서는 ( ② ) 통합, 상위 수준에서는 ( ① ) 통합을 사용하여 최적의 테스트를 지원하는 방식이다.

**답**

- ① :
- ② :

**문제 12** 보안 요소는 소프트웨어 개발에 있어 충족시켜야 할 요소 및 요건을 의미한다. 주요 보안 요소 3가지를 쓰시오. (5점)

**답 :**

**문제 13** 서버 프로그램 구현 시 서버 개발은 다음 항목들을 구현하는 과정이다. 괄호(①~③)에 들어갈 알맞은 항목을 쓰시오. (5점)

|            |                                                                                             |
|------------|---------------------------------------------------------------------------------------------|
| ( ① ) 구현   | 데이터 교환을 위해 사용할 객체를 만드는 과정으로, 변수 및 객체를 송·수신할 데이터의 자료형(Data Type)에 알맞게 생성한다.                  |
| SQL 구현     | 데이터의 삽입, 변경, 삭제 등의 작업을 수행할 SQL문을 생성하는 과정으로, SQL문은 소스 코드 내에 직접 입력하거나, 별도의 XML 파일로 저장하여 관리한다. |
| ( ② ) 구현   | 데이터베이스에 접근하고, SQL을 활용하여 데이터를 실제로 조작하는 코드를 구현하는 과정이다.                                        |
| Service 구현 | 사용자의 요청에 응답하기 위한 로직을 구현하는 과정이다.                                                             |
| ( ③ ) 구현   | 사용자의 요청에 적절한 서비스를 호출하여, 그 결과를 사용자에게 반환하는 코드를 구현하는 과정이다.                                     |

**답**

- ① :
- ② :
- ③ :

**문제 14** 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
#include <stdio.h>
void res(int a[]) {
    int i = 0;
    while (i < 5) {
        if (a[i] < 3)
            a[i] *= 2;
        else
            a[i] %= 3;
        i++;
    }
}
main() {
    int a[] = { 1, 2, 3, 4, 5 };
    res(a);
    printf("%d, %d", *a, *(a + 2));
}
```

답 :

**문제 15** 애플리케이션 테스트 관리와 관련된 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오. (5점)

- ( )은 테스트 결과가 올바른지 판단하기 위해 사전에 정의된 참 값을 대입하여 비교하는 기법 및 활동이다.
- ( )은 결과를 판단하기 위해 테스트 케이스에 대한 예상 결과를 계산하거나 확인한다.
  - ( )의 특징
    - 제한된 검증 : ( )을 모든 테스트 케이스에 적용할 수 없다.
    - 수학적 기법 : ( )의 값을 수학적 기법을 이용하여 구할 수 있다.
    - 자동화 가능 : 테스트 대상 프로그램의 실행, 결과 비교, 커버리지 측정 등을 자동화 할 수 있다.

답 :



**문제 16** 다음은 UNIX 시스템의 구성 요소에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 구성 요소를 쓰시오. (5점)

( ① )은 UNIX의 가장 핵심적인 부분으로, 프로세스, 기억장치, 파일, 입·출력 관리, 프로세스 간 통신, 데이터 전송 및 변환 등 여러 가지 기능을 수행하고, ( ② )은 사용자의 명령어를 인식하여 프로그램을 호출하고, 명령을 수행하는 명령어 해석기로, 시스템과 사용자 간의 인터페이스를 담당한다.

**답**

- ① :
- ② :

**문제 17** 가상 기억장치 관련 용어 중 스래싱(Thrashing)의 개념을 쓰시오. (5점)

**답 :**

**문제 18** 다음과 같은 특징을 갖고 있는 소프트웨어 버전 관리 도구를 쓰시오. (5점)

- 버전 관리 자료가 하나의 원격 저장소와 분산된 개발자 PC의 로컬 저장소에 함께 저장되어 관리되는 방식이다.
- 개발자별로 원격 저장소의 자료를 자신의 로컬 저장소로 복사하여 작업한 후 변경된 내용을 로컬 저장소에서 우선 반영한 다음 이를 원격 저장소에 반영한다.
- 로컬 저장소에서 버전 관리가 가능하므로 원격 저장소에 문제가 생겨도 로컬 저장소의 자료를 이용하여 작업 할 수 있다.
- 종류에는 Git, GNU arch, DVCs, Bazaar, Mercurial, TeamWare, Bitkeeper, Plastic SCM 등이 있다.

**답 :**

**문제 19** 다음은 페이지 교체 알고리즘의 종류별 특징이다. 괄호(①~③)에 들어갈 알맞은 종류를 쓰시오. (5점)

|       |                                                                              |
|-------|------------------------------------------------------------------------------|
| ( ① ) | 각 페이지가 주기억장치에 적재될 때마다 그때의 시간을 기억시켜 가장 먼저 들어와서 가장 오래 있었던 페이지를 교체하는 기법이다.      |
| OPT   | 앞으로 가장 오랫동안 사용하지 않을 페이지를 교체하는 기법이다.                                          |
| ( ② ) | LRU와 비슷한 알고리즘으로, 최근에 사용하지 않은 페이지를 교체하는 기법이다.                                 |
| SCR   | FIFO 기법의 단점을 보완한 기법으로, 가장 오랫동안 주기억장치에 있던 페이지 중 자주 사용되는 페이지의 교체를 방지하기 위한 것이다. |
| LFU   | 사용 빈도가 가장 적은 페이지를 교체하는 기법이다.                                                 |
| ( ③ ) | 최근에 가장 오랫동안 사용하지 않은 페이지를 교체하는 기법이다.                                          |

**답**

- ① :
- ② :
- ③ :

**문제 20** 다음은 〈사원〉 테이블에 대해 소속지점별 사원들의 성과점수 평균이 30을 넘어가는 지점명과 성과점수 평균을 조회하는 SQL문이다. 괄호(①, ②)에 적합한 명령을 넣어 SQL문을 완성하시오.

SELECT 소속지점, AVG(성과점수) FROM 사원 ( ① ) 소속지점 ( ② ) AVG(성과점수) > 30;

**답**

- ① :
- ② :





## 4회 실전 모의고사 정답 및 해설

### [답안 작성 방법 안내]

‘운영체제(OS: Operation System)’처럼 한글과 영문으로 제시되어 있는 경우 ‘운영체제’, ‘OS’, ‘Operation System’ 중 1가지만 쓰면 됩니다.

### [문제 1]

EA(Enterprise Application Integration)

### [문제 2]

43, 2

29, 4

54, 1

33, 3

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값을 한 줄로 43, 2, 29, 4, ... 혹은 43 2 29 4 ...로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

### 해설

문제의 코드는 배열의 각 요소의 값들을 자기를 포함한 다른 요소들의 값과 비교하여 석차를 구한 후 요소의 값과 석차를 출력하는 프로그램입니다.

```
#include <stdio.h>
main() {
    ① int std[] = { 43, 29, 54, 33 };
    ② int r;
    ③ for (int i = 0; i < 4; i++) {
        ④ r = 1;
        ⑤ for (int j = 0; j < 4; j++)
        ⑥     if (std[i] < std[j])
        ⑦         r++;
        ⑧ printf("%d, %d\n", std[i], r);
    }
}
```

- ① 4개의 요소를 갖는 정수형 배열 std를 선언하고 초기화한다.
  - ② 정수형 변수 r을 선언한다.
  - ③ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 4보다 작은 동안 ④~⑧번을 반복하여 수행한다.
  - ④ r의 값을 1로 치환한다.
  - ⑤ 반복 변수 j가 0에서 시작하여 1씩 증가하면서 4보다 작은 동안 ⑥, ⑦번을 반복하여 수행한다.
  - ⑥ std[i]의 값이 std[j]의 값보다 작으면 ⑦번 문장을 실행하고, 아니면 반복문의 시작인 ⑤번으로 돌아간다.
  - ⑦ 'r = r + 1;'과 동일하다. r에 1을 누적한다.
  - ⑧ std[i]의 값을 출력하고, 심표()와 공백 한 칸을 출력한 다음 r의 값을 출력하고 커서를 다음 줄로 이동한다.
- 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

|        | std[0] | std[1] | std[2] | std[3] |
|--------|--------|--------|--------|--------|
| 배열 std | 43     | 29     | 54     | 33     |

| i | j | std[i] | std[j] | r | 출력    |
|---|---|--------|--------|---|-------|
| 0 | 0 | 43     | 43     | 1 | 43, 2 |
|   | 1 |        | 29     | 2 | 29, 4 |
|   | 2 |        | 54     |   | 54, 1 |
|   | 3 |        | 33     |   | 33, 3 |
|   | 4 |        |        |   |       |
| 1 | 0 | 29     | 43     | 1 |       |
|   | 1 |        | 29     | 2 |       |
|   | 2 |        | 54     | 3 |       |
|   | 3 |        | 33     | 4 |       |
|   | 4 |        |        |   |       |
| 2 | 0 | 54     | 43     | 1 |       |
|   | 1 |        | 29     |   |       |
|   | 2 |        | 54     |   |       |
|   | 3 |        | 33     |   |       |
|   | 4 |        |        |   |       |
| 3 | 0 | 33     | 43     | 1 |       |
|   | 1 |        | 29     | 2 |       |
|   | 2 |        | 54     | 3 |       |
|   | 3 |        | 33     |   |       |
|   | 4 |        |        |   |       |
| 4 |   |        |        |   |       |

**[답안 작성 방법 안내]**

‘운영체제(OS; Operation System)’처럼 한글과 영문으로 제시되어 있는 경우 ‘운영체제’, ‘OS’, ‘Operation System’ 중 1가지만 쓰면 됩니다.

**[문제 3]**

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

클러스터는 데이터 저장 시 데이터 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법이다.

**[문제 4]**

① 상품조회 ② 로그인 ③ <<extends>>

**해설**

- ③ 특정 조건에 부합되어 유스케이스의 기능이 확장될 때 원래의 유스케이스와 확장된 유스케이스와의 관계를 확장(Extends) 관계라고 하며, 확장 관계는 확장될 유스케이스에서 원래의 유스케이스 쪽으로 점선 화살표를 연결한 후 화살표 위에 <<extends>>라고 표기합니다.

**[문제 5]**

물리 계층(Physical Layer), 데이터 링크 계층(Data Link Layer), 네트워크 계층(Network Layer), 전송 계층(Transport Layer), 세션 계층(Session Layer), 표현 계층(Presentation Layer), 응용 계층(Application Layer)

**[문제 6]**

① 속성(Attribute) ② 관계(Relationship) ③ 개체(Entity)

**[문제 7]**

모업(Mockup)

[문제 8]

백도어(Back Door, Trap Door)

[문제 9]

21

해설

```
import java.util.Scanner;
public class Test {
    public static void main(String[] args) {
        ① Scanner var = new Scanner(System.in);
        ② int n, i, sum = 0;
        ③ n = var.nextInt();
        ④ var.close();
        ⑤ for (i = 1; i <= n; i++)
        ⑥     sum += i;
        ⑦ System.out.printf("%d", sum);
    }
}
```

- ① Scanner 클래스의 객체 변수 `var`를 키보드로 입력받을 수 있도록 생성한다. `System.in`은 표준 입력장치, 즉 키보드를 의미한다.
- ② 정수형 변수 `n`, `i`, `sum`을 선언하고, `sum`을 0으로 초기화한다.
- ③ 키보드로부터 정수형 값을 입력받아 `n`에 저장한다.
- ④ Scanner 클래스의 객체 변수는 임의의 메모리 영역을 확보하여 사용하는 것이므로 프로그램 종료 전에 `close()` 메소드를 이용하여 사용하던 메모리 영역을 해제해야 다른 프로그램이 해당 영역을 사용할 수 있다.
- ⑤ 반복 변수 `i`가 1에서 시작하여 1씩 증가하면서 `n`보다 작거나 같은 동안 ⑥번을 반복 수행한다.
- ⑥ `sum`에 `i`의 값을 누적한다.
- ⑦ `sum`의 값을 출력한다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

| n | i | sum | 출력 |
|---|---|-----|----|
| 6 | 1 | 0   | 21 |
|   | 2 | 1   |    |
|   | 3 | 3   |    |
|   | 4 | 6   |    |
|   | 5 | 10  |    |
|   | 6 | 15  |    |
|   | 7 | 21  |    |

[문제 10]

- ① 중복 투명성(Replication Transparency) ② 장애 투명성(Failure Transparency) ③ 병행 투명성(Concurrency Transparency)

[문제 11]

- ① 하향식 ② 상향식

[문제 12]

기밀성(Confidentiality), 무결성(Integrity), 가용성(Availability)

[문제 13]

- ① DTO/VO ② DAO ③ Controller

## [문제 14]

2, 0

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 콤마 없이 2 0으로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

## 해설

모든 C프로그램은 반드시 main() 함수부터 시작해야 한다.

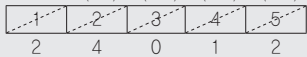
```

#include <stdio.h>
③ void res(int a[]) {
④     int i = 0;
⑤     while (i < 5) {
⑥         if (a[i] < 3)
⑦             a[i] *= 2;
           else
⑧             a[i] %= 3;
⑨         i++;
    }
}
main() {
①     int a[] = { 1, 2, 3, 4, 5 };
②     res(a);
⑩    printf("%d, %d", *a, *(a + 2));
}

```

- ① 5개의 요소를 갖는 정수형 배열 a를 선언하고 초기화한다.
- ② a를 인수로 하여 res 함수를 호출한다. 인수로 배열의 이름을 지정하면 배열의 시작 주소가 인수로 전달된다. 즉 res(a)는 res(&a[0])과 같은 의미이다.
- ③ 리턴값이 없는 res 함수의 시작점이다. ②번에서 보낸 배열 a의 시작 주소를 배열 a가 받는다.
- ④ 정수형 변수 i를 선언하고 0으로 초기화한다.
- ⑤ i가 5보다 작은 동안 ⑥~⑨번 문장을 반복 수행한다.
- ⑥ a[i]가 3보다 작으면 ⑦번 문장을 실행하고, 아니면 ⑧번 문장을 실행한다.
- ⑦ 'a[i] = a[i] \* 2'와 동일하다. a[i]의 값에 2를 곱한 값을 a[i]에 저장한다.
- ⑧ 'a[i] = a[i] % 3'과 동일하다. a[i]의 값을 3으로 나눈 나머지를 a[i]에 저장한다.
- ⑨ 'i = i + 1'과 동일하다. i에 1씩 누적한다.
- ⑩ \*a와 \*(a+2)의 값을 출력한다. \*a는 a[0]과 같은 의미이고 \*(a+2)는 a[0]에서 2번지가 증가한 a[2]와 같은 의미이다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

| i | 배열 a                                                                                | 출력   |
|---|-------------------------------------------------------------------------------------|------|
| 0 |                                                                                     |      |
| 1 | a[0] a[1] a[2] a[3] a[4]                                                            |      |
| 2 | *a *(a+1) *(a+2) *(a+3) *(a+4)                                                      |      |
| 3 |  | 2, 0 |
| 4 |                                                                                     |      |
| 5 |                                                                                     |      |

[답안 작성 방법 안내]

‘운영체제(OS: Operation System)’처럼 한글과 영문으로 제시되어 있는 경우 ‘운영체제’, ‘OS’, ‘Operation System’ 중 1가지만 쓰면 됩니다.

[문제 15]

테스트 오라클(Test Oracle)

[문제 16]

① 커널(Kernel) ② 셸(Shell)

[문제 17]

스래싱은 프로세스의 처리 시간보다 페이지 교체에 소요되는 시간이 더 많아지는 현상이다.

[문제 18]

분산 저장소 방식

[문제 19]

① FIFO(First In First Out) ② NUR(Not Used Recently) ③ LRU(Least Recently Used)

[문제 20]

① GROUP BY ② HAVING

※ 답안 작성 시 주의 사항

대 · 소문자를 구분하지 않습니다. 단 스펠링이 하나라도 틀렸을 경우 부분 점수 1도 없는 오답으로 처리된다는 것을 잊지 마세요.

해설

SELECT 소속지점, AVG(성과점수)  
FROM 사원  
GROUP BY 소속지점  
HAVING AVG(성과점수) > 30;

‘소속지점’, ‘성과점수’의 평균을 표시한다.  
〈사원〉 테이블을 대상으로 검색한다.  
‘소속지점’을 기준으로 그룹을 지정한다.  
‘성과점수’의 평균이 30 초과인 그룹만을 표시한다.



**문제 1** 구문(Syntax)과 의미(Semantics)를 갖는 정형화된 언어를 이용해 요구사항을 수학적 기호로 표현한 후 이를 분석하는 과정으로, 요구사항 분석의 마지막 단계에서 이루어지는 요구사항 분석 기법을 쓰시오. (5점)

**답 :**

**문제 2** 다음 설명에 부합하는 데이터베이스 관련 용어를 쓰시오. (5점)

온라인 업무 처리 형태의 하나로 네트워크상의 여러 이용자가 실시간으로 데이터베이스의 데이터를 갱신하거나 검색하는 등의 단위 작업을 처리하는 방식으로, 여기서 단위 작업은 트랜잭션을 의미하며 데이터 무결성을 위해 각 트랜잭션은 그 전체가 완전히 행해지든지, 아니면 전혀 행해지지 않든지 둘 중 하나가 되도록 처리되어야 한다. 이 시스템은 일반적으로 빠른 응답 시간을 요구하며, 개개의 레코드를 효율적으로 조회하고 수정할 수 있도록 정규화 한다.

**답 :**

**문제 3** 데이터베이스에서 개체 무결성(Entity Integrity)의 개념을 설명하시오. (5점)

**답 :**

**문제 4** 인덱스(index)와 관련된 다음 설명에서 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오. (5점)

인덱스는 데이터 레코드를 빠르게 접근하기 위해 <키 값, 포인터> 쌍으로 구성되는 데이터 구조로, ( ① ) 인덱스와 ( ② ) 인덱스가 있다.

- ( ① ) 인덱스는 인덱스 키의 순서에 따라 데이터가 정렬되어 저장되는 방식으로, 실제 데이터가 순서대로 저장되어 있어 인덱스를 검색하지 않아도 원하는 데이터를 빠르게 찾을 수 있다.
- ( ② ) 인덱스는 인덱스의 키 값만 정렬되어 있을 뿐 실제 데이터는 정렬되지 않는 방식으로, 데이터를 검색하기 위해서는 먼저 인덱스를 검색하여 실제 데이터의 위치를 확인해야 하므로 ( ① )에 비해 검색 속도가 떨어진다.

**답**

- ① :
- ② :

**문제 5** 다음은 키(Key)의 종류별 특징이다. 괄호(①~③)에 들어갈 알맞은 종류를 쓰시오. (5점)

|       |                                                                                |
|-------|--------------------------------------------------------------------------------|
| 슈퍼키   | 한 릴레이션 내에 있는 속성들의 집합으로 구성된 키로, 릴레이션을 구성하는 모든 튜플에 대해 유일성은 만족하지만, 최소성은 만족하지 못한다. |
| ( ① ) | 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미하며, 릴레이션 간의 관계를 표현할 때 사용한다.                |
| 대체키   | 후보키 중에서 선정된 기본키를 제외한 나머지 후보키를 의미한다.                                            |
| ( ② ) | 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용되는 속성들의 부분집합으로, 유일성과 최소성을 모두 만족한다.       |
| ( ③ ) | 후보키 중에서 특별히 선정된 키로 중복된 값과 NULL 값을 가질 수 없다.                                     |

**답**

- ① :
- ② :
- ③ :

**문제 6** DBMS 접속 기술에 대한 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오. (5점)

( )는 데이터베이스에 접근하기 위한 표준 개방형 API로, 1992년 9월 마이크로소프트사에서 출시했다. 개발 언어에 관계 없이 사용할 수 있으며, 프로그램 내 ( ) 문장을 사용하여 MS-Access, DBase, DB2, Excel, Text 등 다양한 데이터베이스에 접근할 수 있다.

**답 :**

**문제 7** 준비 상태에서 대기하고 있는 프로세스 중 하나가 프로세서를 할당받아 실행 상태로 전이되는 과정을 의미하는 프로세스 상태 전이 관련 용어를 쓰시오. (5점)

**답 :**

**문제 8** 다음은 비용 산정 기법의 종류별 특징이다. 괄호에 들어갈 알맞은 종류를 쓰시오. (5점)

|            |                                                                                                                                                |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| 델파이 기법     | 전문가 감정 기법의 주관적인 편견을 보완하기 위해 많은 전문가의 의견을 종합하여 산정하는 기법으로, 전문가들의 편견이나 분위기에 지배되지 않도록 한 명의 조정자와 여러 전문가로 구성된다.                                       |
| COCOMO 모형  | 보헴(Boehm)이 제안한 것으로, 개발할 소프트웨어의 규모를 예측한 후 이를 소프트웨어 종류에 따라 다르게 책정되는 비용 산정 방정식에 대입하여 비용을 산정한다.                                                    |
| (     ) 기법 | 소프트웨어 각 기능의 원시 코드 라인 수의 비관치, 낙관치, 기대치를 측정하여 예측치를 구하고 이를 이용하여 비용을 산정하는 기법으로, 측정이 용이하고 이해하기 쉬워 가장 많이 사용되며, 예측치를 이용하여 생산성, 노력, 개발 기간 등의 비용을 산정한다. |
| 전문가 감정 기법  | 조직 내에 있는 경험이 많은 두 명 이상의 전문가에게 비용 산정을 의뢰하는 기법으로, 가장 편리하고 신속하게 비용을 산정할 수 있으며, 의뢰자로부터 믿음을 얻을 수 있다.                                                |
| Putnam 모형  | 소프트웨어 생명 주기의 전 과정 동안에 사용될 노력의 분포를 가정해 주는 모형으로, 시간에 따른 함수로 표현되는 Rayleigh-Norden 곡선의 노력 분포도를 기초로 한다.                                             |

**답 :**

**문제 9** 다음은 1부터 70까지의 난수를 구하는 프로그램을 C 언어로 구현한 것이다. 괄호에 적합한 라이브러리 함수를 넣어 코드를 완성하시오. (5점)

```
#include <stdio.h>
#include <stdlib.h>
main() {
    for (int i = 0; i < 10; i++) {
        printf("%d\n", (1 + (     ) % 70));
    }
}
```

**답 :**



**문제 10** 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
public class Test {  
    public static void main(String[] args) {  
        String str = "Operation";  
        String rst = change(str);  
        System.out.println(str.charAt(3));  
        System.out.println(rst.charAt(3));  
    }  
  
    static String change(String x) {  
        String y = new String();  
        int n = x.length() - 1;  
        for(int i = n; i >= 0; i--)  
            y += x.charAt(i);  
        return y;  
    }  
}
```

**답 :**

**문제 11** Spring Source사와 Accenture사가 2007년 공동 개발한 오픈소스 프레임워크로, 데이터베이스나 파일의 데이터를 교환하는데 필요한 컴포넌트들을 제공하며, 로그 관리, 추적, 트랜잭션 관리, 작업 처리 통계, 작업 재시작 등의 다양한 기능을 제공하는 배치 스케줄러(Batch Scheduler)를 쓰시오. (5점)

**답 :**

**문제 12** 다음은 UI 흐름 설계에서 사용되는 컨트롤에 대한 설명이다. 괄호(①, ②)에 들어갈 알맞은 컨트롤을 쓰시오. (5점)

|        |                                                                                                                |
|--------|----------------------------------------------------------------------------------------------------------------|
| ( ① )  | <ul style="list-style-type: none"> <li>• 목록에서 항목을 선택하거나 입력할 수 있다.</li> <li>• 자주 사용하는 값을 초기값으로 설정한다.</li> </ul> |
| 텍스트 박스 | <ul style="list-style-type: none"> <li>• 입력이 가능함을 표시한다.</li> <li>• 필드 길이, 텍스트 정렬 방식 등을 지정한다.</li> </ul>        |
| ( ② )  | <ul style="list-style-type: none"> <li>• 여러 개의 값 중 하나만을 선택할 수 있다.</li> <li>• 자주 사용하는 값을 초기값으로 설정한다.</li> </ul> |
| 체크 박스  | 여러 개의 값 중 하나 이상을 선택할 수 있다.                                                                                     |

**답**

- ① :
- ② :

**문제 13** 다음 설명의 괄호에 들어갈 가장 적합한 데이터베이스 관련 용어를 쓰시오. (5점)

어떤 테이블 R에서 X와 Y를 각각 R의 속성 집합의 부분 집합이라 할 때 속성 X의 값 각각에 대해 시간에 관계없이 항상 속성 Y의 값이 오직 하나만 연관되어 있을 때 Y는 X에 ( )이라고 하고,  $X \rightarrow Y$ 로 표기한다.

**답 :**

**문제 14** 다음 설명에 부합하는 네트워크 공격 유형을 쓰시오. (5점)

여러 곳에 분산된 공격 지점에서 한 곳의 서버에 대해 공격을 수행하는 네트워크 공격 유형으로, 네트워크에서 취약점이 있는 호스트들을 탐색한 후 이들 호스트들에 공격용 툴을 설치하여 에이전트(Agent)로 만든 후 공격에 이용한다. 공격의 범위를 확대하기 위해 일부 호스트에 다수의 에이전트를 관리할 수 있는 핸들러(Handler) 프로그램을 설치하여 마스터(Master)로 지정한 후 공격에 이용하기도 한다.

**답 :**

**문제 15** DNS(Domain Name System)의 개념을 설명하시오. (5점)

**답 :**

**문제 16** 가상 기억장치 관리와 관련된 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오. (5점)

- ( )은 프로세스가 일정 시간 동안 자주 참조하는 페이지들의 집합이다.
- 데닝(Denning)이 제안한 프로그램의 움직임에 대한 모델로, 프로그램의 Locality 특징을 이용한다.
- 자주 참조되는 ( )을 주기억장치에 상주시킴으로써 페이지 부재 및 페이지 교체 현상이 줄어들어 프로세스의 기억장치 사용이 안정된다.
- 시간이 지남에 따라 자주 참조하는 페이지들의 집합이 변화하기 때문에 ( )은 시간에 따라 변경된다.

**답 :**

**문제 17** 다음 C언어로 구현된 프로그램을 분석하여 7을 입력했을 때 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
#include <stdio.h>
main() {
    int find, tmp, result, dif = 99;
    int arr[] = { 5, 3, 9, 14, 1, 12 };
    scanf("%d", &find);
    for (int x = 0; x < 6; x++) {
        tmp = arr[x] > find ? arr[x] - find : find - arr[x];
        if (tmp <= dif) {
            result = arr[x];
            dif = tmp;
        }
    }
    printf("%d %d", result, dif);
}
```

**답 :**

**문제 18** 웹 서비스에 사용되는 프로토콜에 대한 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오. (5점)

( )는 컴퓨터 네트워크 상에서 HTTP/HTTPS, SMTP 등을 이용하여 XML을 교환하기 위한 통신 규약으로, 웹 서비스에서 사용되는 메시지의 형식과 처리 방법을 지정한다. ( )는 기본적으로 HTTP 기반에서 동작하기 때문에 프록시와 방화벽의 영향 없이 통신이 가능하다. 다음은 ( )의 주요 요소이다.

|          |                                                |
|----------|------------------------------------------------|
| Envelope | XML 문서를 메시지로 정의하는 것으로, 메시지에 대한 요소와 접근방법을 정의한다. |
| Header   | 메시지에 포함되는 웹 서비스를 정의하는 것으로, 생략이 가능하다.           |
| Body     | 실제 메시지가 포함된다.                                  |

**답 :**

**문제 19** 데이터베이스와 관련한 다음 <처리 조건>에 부합하도록 SQL문의 괄호(①, ②)에 알맞은 명령어를 쓰시오. (5점)

<처리 조건>

- 테이블명은 사원으로 정의한다.
- 사원 테이블의 구조는 다음과 같다.

<사원>

| 속성명 | 데이터 타입      | 비고          |
|-----|-------------|-------------|
| 성명  | VARCHAR(10) | PRIMARY KEY |
| 부서  | VARCHAR(20) |             |
| 경력  | INT         |             |
| 주소  | VARCHAR(20) |             |
| 기본급 | INT         |             |

- 성명이 '홍길동', '경력'이 10인 사원의 인사 정보를 입력하는 SQL문을 완성하십시오.

INSERT ( ① ) 사원(성명, 경력) ( ② )('홍길동', 10);

**답**

- ① :
- ② :

**문제 20** <표1>의 평가 점수에 따라 성적을 부여하는 소프트웨어를 테스트하기 위해 <표2>와 같이 입력 조건의 경계 부분을 입력 자료로 준비했다. 이 테스트를 가리키는 용어를 쓰시오. (5점)

<표 1>

| 평가 점수  | 성적 |
|--------|----|
| 80~100 | A  |
| 60~79  | B  |
| 40~59  | C  |
| 0~39   | D  |

<표 2>

| 입력 자료        | 예상 결과      |
|--------------|------------|
| 80, 100, 101 | A, 오류(101) |
| 60, 79       | B          |
| 40, 59       | C          |
| -1, 0, 39    | D, 오류(-1)  |

**답 :**



## 5회 실전 모의고사 정답 및 해설

### [답안 작성 방법 안내]

‘운영체제(OS; Operation System)’처럼 한글과 영문으로 제시되어 있는 경우 ‘운영체제’, ‘OS’, ‘Operation System’ 중 1가지만 쓰면 됩니다.

### [문제 1]

정형 분석(Formal Analysis)

### [문제 2]

OLTP(Online Transaction Processing)

### [문제 3]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

개체 무결성은 기본 테이블의 기본키를 구성하는 어떤 속성도 Null 값이나 중복값을 가질 수 없다는 규정이다.

### [문제 4]

① 클러스터드(Clustered) ② 언클러스터드(Non-Clustered)

### [문제 5]

① 외래키(Foreign Key) ② 후보키(Candidate Key) ③ 기본키(Primary Key)

### [문제 6]

ODBC(Open DataBase Connectivity)

### [문제 7]

디스패치(Dispatch)

### [문제 8]

LOC(원시 코드 라인 수)

### [문제 9]

rand( )

※ 답안 작성 시 주의 사항

프로그래밍 언어에서 사용하는 변수, 함수의 이름은 대소문자를 구분하기 때문에 변수, 함수 이름을 작성할 때는 대소문자를 구분해서 정확히 작성해야 합니다.

### 해설

```
#include <stdio.h>
#include <stdlib.h>
main( ) {
    for (int i = 0; i < 10; i++) {
        printf("%d\n", (1 + rand( ) % 70));
    }
}
```

rand( ) 함수가 정의되어 있는 헤더 파일이다.

반복 변수 i가 0에서 시작하여 1씩 증가하면서 10보다 작은 동안 다음 문장을 반복 수행한다. 즉 다음 문장을 10회 반복 수행한다.  
rand( ) 함수로 발생한 난수를 70으로 나눈 나머지에 1을 더하여 출력한다. 즉 1~70 사이의 난수를 출력한다.

[문제 10]

r  
t

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값을 한 줄로 r, t 혹은 r t로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

해설

```
public class Test {
    public static void main(String[] args) {
        ① String str = "Operation";
        ②⑨ String rst = change(str);
        ⑩ System.out.println(str.charAt(3));
        ⑪ System.out.println(rst.charAt(3));
    }

    ③ static String change(String x) {
        ④ String y = new String();
        ⑤ int n = x.length() - 1;
        ⑥ for(int i = n; i >= 0; i--)
        ⑦     y += x.charAt(i);
        ⑧ return y;
    }
}
```

- ① 문자열 변수 str을 선언하고 "Operation"으로 초기화한다.
- ② 문자열 변수 rst를 선언하고, str의 값 "Operation"을 인수로 하여 change() 메소드를 호출한 다음 돌려받은 값을 rst에 저장한다.
- ③ 메소드의 리턴값이 문자열인 change() 메소드의 시작점이다. ②번에서 전달받은 "Operation"을 문자열 변수 x로 받는다.
- ④ 문자열 변수 y를 선언한다. y는 배열과 동일한 객체 변수이므로 초기값이 없을 때는 new 예약어를 사용한다.
- ⑤ 정수형 변수 n을 선언하고, 문자열 변수 x의 길이에서 1을 뺀 8(9-1)로 초기화한다.
- ⑥ 반복 변수 i가 n에서 시작하여 1씩 감소하면서 0보다 크거나 작은 동안 ⑦번을 반복하여 수행한다.
- ⑦ 'y = y + x.charAt(i)'와 동일하다. y의 값에 x에서 i번째에 있는 문자를 더한다. 문자 간의 더하기(+) 연산은 앞의 문자에 뒤의 문자를 붙여서 반환한다.  
(예) 'abc' + 'y' = 'abcy'

- ⑧ y의 값을 호출한 곳(main() 메소드)으로 반환한다.
- ⑨ change() 메소드로부터 반환받은 값을 rst에 저장한다.
- ⑩ str의 3번째에 있는 문자를 출력한 후 다음 줄로 이동한다.
- ⑪ rst의 3번째에 있는 문자를 출력한 후 다음 줄로 이동한다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

| x         | n | i  | x.charAt(i) | y         | 출력 |
|-----------|---|----|-------------|-----------|----|
| Operation | 8 | 8  | n           | n         | r  |
|           |   | 7  | o           | no        | t  |
|           |   | 6  | i           | noi       |    |
|           |   | 5  | t           | noit      |    |
|           |   | 4  | a           | noita     |    |
|           |   | 3  | r           | noitar    |    |
|           |   | 2  | e           | noitare   |    |
|           |   | 1  | p           | noitarep  |    |
|           |   | 0  | O           | noitarepO |    |
|           |   | -1 |             |           |    |

**[답안 작성 방법 안내]**

‘운영체제(OS; Operation System)’처럼 한글과 영문으로 제시되어 있는 경우 ‘운영체제’, ‘OS’, ‘Operation System’ 중 1가지만 쓰면 됩니다.

**[문제 11]**

스프링 배치(Spring Batch)

**[문제 12]**

① 콤보 박스(Combo Box) ② 라디오 박스(Radio Box)

**[문제 13]**

함수적 종속(Functional Dependency)

**[문제 14]**

DDoS(Distributed Denial of Service, 분산 서비스 거부)

**[문제 15]**

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

DNS는 문자로 된 도메인 네임을 컴퓨터가 이해할 수 있는 IP 주소로 변환하는 역할을 하는 시스템이다.

**[문제 16]**

워킹 셋(Working Set)

**[문제 17]**

9 2

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 콤마를 넣어 9, 2로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

**해설**

문제의 코드는 배열에서 입력받은 수 7개 가장 가까운 수를 찾아 출력하는 프로그램입니다. 7과 가장 가까운 수를 구하려면 7과 다른 수들의 차를 계산한 후 차이를 비교하여 차이가 가장 작은 수를 찾으면 됩니다. 주의할 점은 차이를 계산할 때 음수가 나오면 안 되므로 7과 비교할 값의 대·소를 비교한 후 큰 수에서 작은 수를 빼야 합니다.

```
#include <stdio.h>
main() {
    ① int find, tmp, result, dif = 99;
    ② int arr[ ] = { 5, 3, 9, 14, 1, 12 };
    ③ scanf("%d", &find);
    ④ for (int x = 0; x < 6; x++) {
        ⑤ tmp = arr[x] > find ? arr[x] - find : find - arr[x];
        ⑥ if (tmp <= dif) {
            ⑦ result = arr[x];
            ⑧ dif = tmp;
        }
    }
    ⑨ printf("%d %d", result, dif);
}
```



- ① 정수형 변수 find, tmp, result, dif를 선언하고, dif를 99로 초기화한다.
  - ② 6개의 요소를 갖는 정수형 배열 arr을 선언하고 초기화한다.
  - ③ 정수를 입력받아 find에 저장한다.
  - ④ 반복 변수 x가 0에서 시작하여 1씩 증가하면서 6보다 작은 동안 ⑤~⑧번을 반복하여 수행한다.
  - ⑤ arr[x]의 값이 find의 값보다 크면 tmp에 arr[x]의 값에서 find의 값을 뺀 값을 저장하고, 아니면 tmp에 find의 값에서 arr[x]의 값을 뺀 값을 저장한다.
  - ⑥ tmp의 값이 dif의 값보다 작거나 같으면 ⑦, ⑧번을 수행하고, 아니면 반복문의 처음인 ④번으로 돌아간다.
  - ⑦ result에 arr[x]의 값을 저장한다.
  - ⑧ dif에 tmp의 값을 저장한다.
  - ⑨ result를 출력하고 공백 한 칸을 띄운다. 이어서 dif의 값을 출력한다.
- 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

|     | arr[0] | arr[1] | arr[2] | arr[3] | arr[4] | arr[5] |
|-----|--------|--------|--------|--------|--------|--------|
| arr | 5      | 3      | 9      | 14     | 1      | 12     |

| find | x | arr[x] | tmp | result | dif | 출력  |
|------|---|--------|-----|--------|-----|-----|
| 7    | 0 | 5      | 2   | 5      | 99  | 9 2 |
|      | 1 | 3      | 4   | 9      | 2   |     |
|      | 2 | 9      | 2   |        | 2   |     |
|      | 3 | 14     | 7   |        |     |     |
|      | 4 | 1      | 6   |        |     |     |
|      | 5 | 12     | 5   |        |     |     |
|      | 6 |        |     |        |     |     |

#### [문제 18]

SOAP(Simple Object Access Protocol)

#### [문제 19]

① INTO ② VALUES

※ 답안 작성 시 주의 사항

대 · 소문자를 구분하지 않습니다. 단 스펠링이 하나라도 틀렸을 경우 부분 점수 1도 없는 오답으로 처리된다는 것을 잊지 마세요.

#### 해설

INSERT INTO 사원(성명, 경력)  
VALUES('홍길동', 10);

〈사원〉 테이블의 '성명', '경력'에 삽입한다.  
'성명'에 '홍길동'을, '경력'에 10을 삽입한다.

#### [문제 20]

경계값 분석(Boundary Value Analysis)

m · e · m · o



# 나는 시험에 나오는 것만 공부한다!

## 이제 시나공으로 한 번에 정복하세요!

기초 이론부터  
완벽하게 공부해서  
안전하게 합격하고  
싶어요!

### 기본서 (필기/실기)



#### 특 정

자세하고 친절한 이론으로  
기초를 쌓은 후 바로 문제풀  
이를 통해 정리한다.

#### 구 성

본권  
기출문제(5회)  
토막강의

#### 실기

채점 프로그램(워드프로세서,  
컴퓨터활용능력, ITQ)

#### 출 간 종 목

컴퓨터활용능력1급 필기/실기  
컴퓨터활용능력2급 필기/실기  
워드프로세서 필기/실기  
정보처리기사 필기/실기  
정보처리산업기사 필기/실기  
정보처리기능사 필기/실기  
사무자동화산업기사 실기  
ITQ 엑셀/한글/파워포인트  
GTQ 1급/2급

핵심이론만  
체계적으로 정리한 후  
문제풀이를 통해  
정리하고 싶어요!

### SUMMARY (필기)



#### 특 정

시험에 꼭 나오는 핵심이론으  
로 개념을 체계적으로 정리한  
후 기출문제로 마무리한다.

#### 구 성

핵심요약  
기출문제(15회)  
토막강의

#### 출 간 종 목

컴퓨터활용능력1급 필기  
컴퓨터활용능력2급 필기  
워드프로세서 필기  
정보처리기사 필기  
정보처리산업기사 필기  
정보처리기능사 필기  
사무자동화산업기사 필기

이론은 공부했지만  
어떻게 적용되는지  
문제풀이를 통해  
감각을 익히고 싶어요!

### 총정리 (필기)



#### 특 정

간단하게 이론을 정리한 후  
충분한 문제풀이를 통해 실  
전 감각을 향상시킨다.

#### 구 성

핵심요약  
기출문제(10회)  
모의고사(10회)  
토막강의

#### 출 간 종 목

컴퓨터활용능력1급 필기  
컴퓨터활용능력2급 필기  
워드프로세서 필기  
사무자동화산업기사 필기

이론은 완벽해요!  
기출문제로  
마무리하고 싶어요!

### 기출문제집 (필기/실기)



#### 특 정

최신 기출문제를 반복 학습  
하며 최종 마무리한다.

#### 구 성

핵심요약(PDF)  
기출문제(15회)  
토막강의

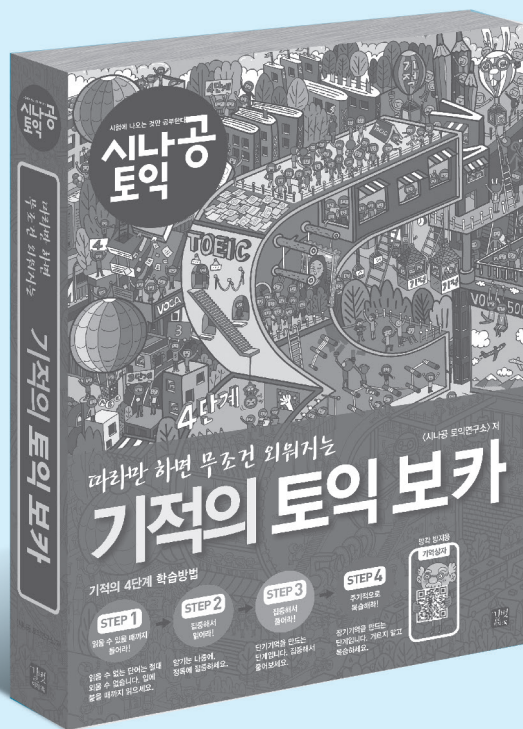
#### 실기

기출문제(10회)

#### 출 간 종 목

컴퓨터활용능력1급 필기/실기  
컴퓨터활용능력2급 필기/실기  
워드프로세서 필기  
사무자동화산업기사 필기

# 따라만 하면 무조건 외워지는 기적의 토익 보카



## 시나공 토익 기적의 토익 보카

〈시나공 토익연구소〉 지음 | 688쪽 | 19,500원

이제껏 이런 기적은 없었다! 가장 효과적인 암기방식을 도입한 <기적의 토익 보카>!

**① 힘들게 외울 필요가 없는 국내 최초 3단계 암기 방식 도입!**

1단계: 무조건 듣는다! 2단계: 무조건 읽는다! 3단계: 무조건 문제를 푼다!

**② 정기토익 출제 단어 완벽 분석!**

신토익 이후 토익 시험에 출제된 단어를 빈출 순으로 정리!

**③ 직독직해 훈련 병행!**

단어뿐만 아니라 독해까지도 해결할 수 있는 문장 해석의 노하우 제시!

**④ <시나공 토익>만의 장기 기억 프로그램 제공!**

단어를 외운 후 절대 잊어버리지 않게 해주는 장기 기억용 특수 반복 암기 프로그램 제공!