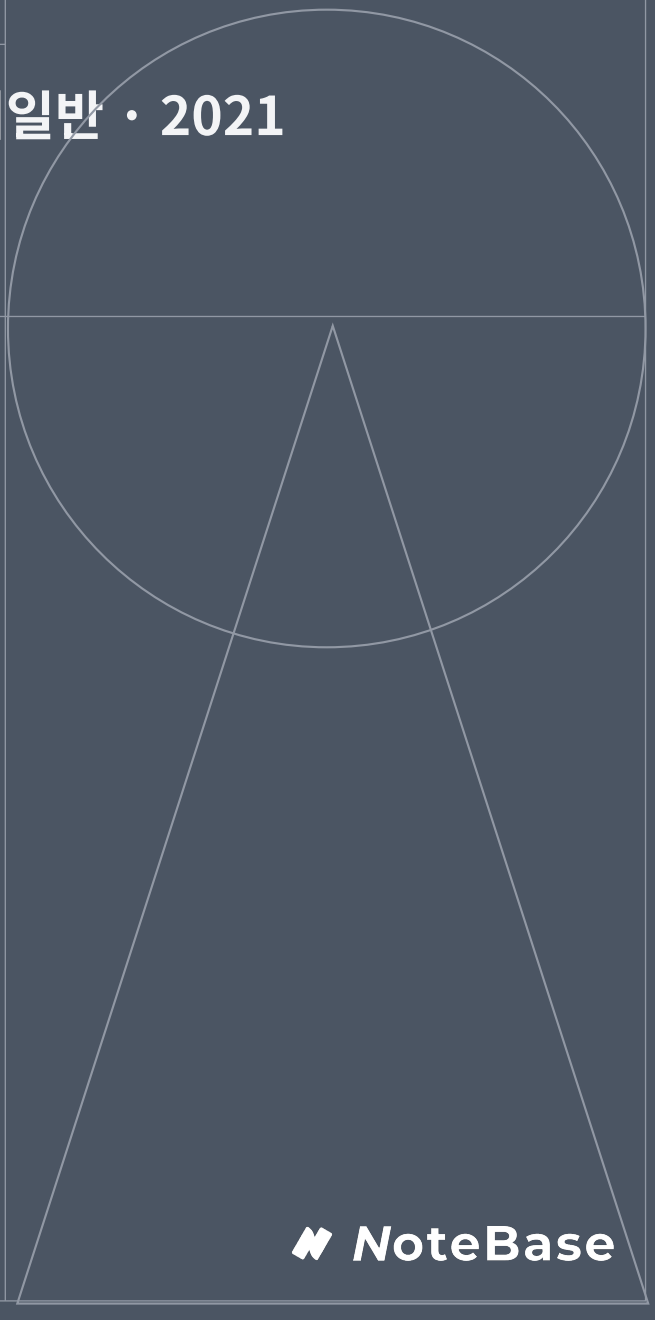
기출OX 해설집

공무원/지방직/9급 · 컴퓨터일반 · 2021



NoteBase

기출OX 해설집

공무원/지방직/9급 · 컴퓨터일반 · 2021

공무원/지방직/9급 · 컴퓨터일반 · 2021

문 1. 가상화폐와 관련이 가장 적은 것은?

① 채굴(mining)

② 소켓(socket)

③ 비트코인(bitcoin)

④ 거래(transaction)

정답 2번

간단한 해설

소켓은 프로그램이 네트워크로 데이터를 주고받는 **통신 접점**이다. 가상화폐 시스템에서도 사용할 수 있지만, 가상화폐의 발행·이전 자체를 나타내는 고유 개념은 아니다.

NoteBase

선지별 해설

① 채굴(mining): 비트코인과 같은 작업증명 방식의 가상화폐에서 거래를 담은 새 블록을 추가하기 위해 계산 작업을 수행하는 과정이다. 채굴에 성공하면 규칙에 따라 새로 발행된 코인과 거래 수수료를 보상으로 받을 수 있으므로 **거래 기록의 확정과 코인 발행**에 직접 관련된다.

② 소켓(socket): 프로그램이 네트워크를 통해 데이터를 보내고 받는 **통신의 끝점**이다. 가상화폐 프로그램뿐 아니라 웹·메신저 등 다양한 프로그램에서 쓰는 범용 네트워크 개념이므로, 가상화폐 자체와의 직접적인 관련성이 낮다.

③ 비트코인(bitcoin): 중앙 관리기관 없이 참여자들이 거래 기록을 공유하고 검증하는 대표적인 **가상화폐 자체**이다. 따라서 가상화폐와 직접 관련된다.

④ 거래(transaction): 가상화폐의 이전 등을 나타내는 기록이다. 예를 들어 다른 사람에게 비트코인을 보내면 그 이전을 나타내는 거래가 생성되고, 검증을 거쳐 블록체인에 기록되므로 **가상화폐의 이전과 기록**에 직접 관련된다.

가상화폐와의 직접적인 관련성을 비교하면, 채굴은 블록 생성 과정, 비트코인은 가상화폐의 한 종류, 거래는 이전 기록에 해당한다. 범용 네트워크 통신 개념인 ② 소켓이 가상화폐와의 관련성이 가장 적다.

NoteBase

문 2. 다음 설명에 해당하는 기술은?

실제 환경에 가상 사물을 합성해 원래 존재하는 사물처럼 보이도록 하는 기술이다.

① MPEG(Moving Picture Experts Group)
② AI(Artificial Intelligence)
③ AR(Augmented Reality)
④ VOD(Video On Demand)

정답 3번

간단한 해설

실제 환경에 가상 사물을 합성하여 함께 보이게 하는 기술은 AR(증강현실)이다. 현실의 장면을 바탕으로 가상의 정보를 덧붙인다는 점이 핵심이다.

핵심 풀이

AR(Augmented Reality, 증강현실)은 **실제 환경을 유지하면서 가상 사물이나 정보를 겹쳐 보여 주는** 기술이다. 지문의 '실제 환경'은 바탕이 되는 현실 장면을, '가상 사물을 합성'은 그 장면에 컴퓨터가 만든 대상을 추가하는 것을 뜻한다. 예를 들어 스마트폰 카메라로 비춘 실제 방에 가상의 가구를 배치하여, 그 가구가 방 안에 놓인 것처럼 보여 주는 방식이다.

가상현실(VR)이 컴퓨터로 만든 가상 환경을 체험하게 하는 데 비해, 증강현실(AR)은 **현실 장면과 가상 대상을 함께** 보여 준다. 따라서 지문에서 실제 환경이 바탕으로 남아 있다는 점이 두 개념을 구별하는 단서이다.

NoteBase

문 3. 일반적인 컴퓨터 시스템에서 정확한 값으로 표현하기 가장 어려운 것은?

① $\sqrt{2}$
② $1\frac{3}{4}$
③ 2.5
④ -0.25×2^{-5}

정답 1번

간단한 해설

$\sqrt{2}$는 **무리수**이므로 이진수로도 소수 부분이 끝나거나 일정하게 반복되지 않는다. 따라서 **유한한 비트의 수치 표현**으로는 정확한 값을 저장할 수 없어 근삿값을 사용한다.

선지별 해설

일반적인 컴퓨터는 유한한 비트로 수를 표현하므로, 주어진 수가 유한한 이진수로 표현되는지를 비교한다.
기약분수의 분모가 2의 거듭제곱인 수는 이진 소수 부분이 유한하게 끝난다.

① $\sqrt{2}$: 무리수이므로 이진수의 소수 부분이 **끝나지 않고 순환하지도 않는다**. 저장할 비트를 늘리면 더 가까운 근삿값을 얻을 수 있지만, 유한한 비트로 정확한 수치 전체를 표현할 수는 없다.

② $1\frac{3}{4}$: $1 + \frac{1}{2} + \frac{1}{4}$이므로 이진수로 $(1.11)_2$이다. 소수점 아래 두 비트로 끝나므로 **정확한 표현이 가능하다**.

③ 2.5: $2 + \frac{1}{2}$이므로 이진수로 $(10.1)_2$이다. 십진 소수라고 해서 반드시 오차가 생기는 것은 아니며, 이 수는 **유한한 이진 소수**이므로 정확히 표현할 수 있다.

④ -0.25×2^{-5}: $-0.25 = -2^{-2}$이므로 주어진 값은 $-2^{-7} = -\frac{1}{128}$이다. 절댓값이 이진수로 $(0.0000001)_2$인 유한소수이므로 정확히 표현할 수 있다. **음수라는 부호는 정확한 표현을 방해하지 않는다**.

따라서 나머지 세 수와 달리 유한한 이진수로 정확히 표현할 수 없는 것은 ①의 $\sqrt{2}$이다.

NoteBase

문 4. 노드(node)가 11개 있는 트리의 간선(edge) 개수는?

① 10

② 11

③ 12

④ 13

정답 1번

간단한 해설

트리의 간선 수는 **노드 수보다 1개 적다**. 노드가 11개이면 간선은 10개이다.

계산 과정

트리는 모든 노드가 연결되어 있고 사이클이 없는 구조이다. 임의의 노드 하나를 루트로 정하면 **루트를 제외한 각 노드는 부모와 간선 하나로 연결**된다. 따라서 노드 수를 n, 간선 수를 e라고 할 때 $e = n - 1$이다.

노드 수가 11개이므로 간선 수는 11 − 1 = 10개이다. 이는 ①의 10과 일치한다.

NoteBase

문 5. CPU에서 명령어를 처리하는 단계 중 가장 첫 번째에 위치하는 것은?

① 실행(execution)

② 메모리 접근(memory access)

③ 명령어 인출(instruction fetch)

④ 명령어 해독(instruction decode)

정답 3번

간단한 해설

CPU는 수행할 명령어를 먼저 메모리에서 가져와야 그 내용을 해독하고 실행할 수 있다. 이 첫 과정이 **명령어 인출**이다.

핵심 풀이

명령어 인출(instruction fetch): 프로그램 카운터(PC)가 가리키는 주소의 명령어를 메모리에서 읽어 명령어 레지스터(IR)에 저장한다. 처리할 명령어를 확보하는 과정이므로 **명령어 처리의 첫 단계**이다.

명령어 해독과 실행: 가져온 명령어를 해독하여 어떤 연산을 해야 하는지 파악한 뒤, 그에 따라 연산을 수행한다. 따라서 해독과 실행은 모두 **명령어 인출 이후**에 이루어진다.

메모리 접근: 선지에서 말하는 메모리 접근 단계는 명령어 수행에 필요한 데이터를 읽거나 결과를 메모리에 쓰는 단계이다. 인출에서도 메모리를 읽지만, **명령어를 가져오는 인출**과 **데이터를 읽고 쓰는 메모리 접근**은 처리 단계의 명칭으로 구분한다.

NoteBase

문 6. 캐시(cache)에 대한 설명으로 옳지 않은 것은?

① CPU와 인접한 곳에 위치하거나 CPU 내부에 포함되기도 한다.

② CPU와 상대적으로 느린 메인(main) 메모리 사이의 속도 차이를 줄이기 위해 사용된다.

③ 다중프로세서 시스템에서는 write–through 정책을 사용하더라도 데이터 불일치 문제가 발생할 수 있다.

④ 캐시에 쓰기 동작을 수행할 때 메인 메모리에도 동시에 쓰기 동작이 이루어지는 방식을 write–back 정책이라고 한다.

정답 4번

간단한 해설

캐시에 쓸 때 메인 메모리에도 함께 쓰는 방식은 **write-through**이다. write-back은 변경 내용을 캐시에 먼저 반영하고, 변경된 블록이 교체될 때 등에 메인 메모리로 반영한다.

선지별 해설

① O: 캐시는 CPU가 빠르게 접근할 수 있도록 **CPU 내부 또는 CPU와 가까운 곳**에 배치한다. CPU 내부에 통합된 캐시도 있고, CPU 외부에 배치된 캐시도 있다.

② O: 캐시는 CPU와 메인 메모리 사이에서 자주 사용하거나 최근 사용한 명령어와 데이터를 보관하는 고속 메모리이다. 필요한 내용이 캐시에 있으면 느린 메인 메모리에 접근하지 않아도 되므로 **CPU의 메모리 접근 대기 시간**을 줄인다.

③ O: write-through는 쓰기 내용을 캐시와 메인 메모리에 함께 반영하지만, **다른 프로세서의 캐시까지 자동으로 갱신하는 것은 아니다**. 예를 들어 두 프로세서의 캐시에 같은 데이터의 값 1이 저장되어 있을 때, 한 프로세서가 이를 2로 바꾸어 메인 메모리에도 반영해도 다른 캐시에는 1이 남을 수 있다. 따라서 다중프로세서 시스템에서는 다른 캐시의 사본을 무효화하거나 갱신하는 등 캐시 일관성을 유지하는 별도 처리가 필요하다.

④ X: 캐시와 메인 메모리에 함께 쓰는 방식은 **write-through**이다. write-back은 쓰기 시 캐시의 내용만 먼저 바꾸고 해당 블록이 수정되었음을 표시한 뒤, 변경된 블록이 교체될 때 등에 메인 메모리에 반영한다. 따라서 선지의 'write-back 정책'을 'write-through 정책'으로 고쳐야 한다.

NoteBase

문 7. 가상 기계(virtual machine)에 대한 설명으로 옳지 않은 것은?

① 가상 기계 모니터 또는 하이퍼바이저(hypervisor)는 가상 기계를 지원하는 소프트웨어이다.

② 가상 기계 모니터는 호스트 운영체제 위에서만 실행된다.

③ 데스크톱 환경에서 Windows나 Linux와 같은 운영체제를 여러 개 실행하기 위해 사용되기도 한다.

④ 가상 기계가 호스트 운영체제 위에서 동작할 때, 이 기계 위에서 동작하는 응용 프로그램은 처리 속도가 느려질 수 있다.

정답 2번

간단한 해설

②의 '호스트 운영체제 위에서만'이라는 한정이 잘못되었다. 하이퍼바이저에는 호스트 운영체제 위에서 실행되는 방식뿐 아니라 **하드웨어 위에서 직접 실행되는 방식**도 있다.

선지별 해설

① O: 가상 기계 모니터(VMM)는 하이퍼바이저라고도 하며, 가상 기계를 생성하고 실행하는 소프트웨어이다. 물리적인 CPU·메모리·입출력 자원을 가상 기계에 배분하고 관리하여 **각 가상 기계에 독립된 실행 환경**을 제공한다.

② X: 하이퍼바이저가 실행되는 위치는 방식에 따라 다르다. **1형은 하드웨어 위에서 직접 실행**되고, 2형은 호스트 운영체제 위에서 실행된다. 따라서 '호스트 운영체제 위에서만 실행된다'는 설명은 2형의 특징을 모든 하이퍼바이저에 적용한 것이다.

③ O: 가상 기계마다 별도의 운영체제를 설치하면 **한 물리적 컴퓨터에서 여러 운영체제를 함께 실행**할 수 있다. 예를 들어 Windows 데스크톱에서 가상 기계를 만들어 Linux를 실행하면, 기존 Windows 환경을 사용하면서 Linux 환경도 이용할 수 있다.

④ O: 호스트 운영체제 위에서 가상 기계를 실행하면, 그 안의 응용 프로그램은 가상화 계층을 거쳐 물리적 자원을 이용한다. 이 과정에서 자원 관리와 입출력 처리 등에 **추가 작업 부담**이 생기고, 호스트와 가상 기계가 CPU·메모리 등을 나누어 사용하므로 처리 속도가 느려질 수 있다. 성능 저하의 정도는 가상화 방식과 작업의 특성, 자원 여유에 따라 달라진다.

NoteBase

문 8. 프로세스(process)에 대한 설명으로 옳지 않은 것은?

① 실행 중인 프로그램이다.

② 프로그램 코드 외에도 현재의 활동 상태를 갖는다.

③ 준비(ready) 상태는 입출력 완료 또는 신호의 수신 같은 사건(event)이 일어나기를 기다리는 상태이다.

④ 호출한 함수의 반환 주소, 매개변수 등을 저장하기 위해 스택을 사용한다.

정답 3번

간단한 해설

입출력 완료나 신호 수신 같은 사건을 기다리는 것은 **대기(waiting) 상태**이다. 준비(ready) 상태는 실행에 필요한 조건을 갖추고 **CPU 할당을 기다리는 상태**이다.

선지별 해설

① O: 프로세스는 **실행 중인 프로그램**이다. 프로그램이 저장된 명령어들의 집합이라면, 프로세스는 그 프로그램을 실행하는 활동과 실행에 필요한 상태를 포함한다. 여기서 실행 중이라는 말은 현재 CPU를 사용하는 순간만을 뜻하지 않으며, 준비 상태나 대기 상태의 프로세스도 포함한다.

② O: 프로세스는 프로그램 코드뿐 아니라 **현재 어디까지 실행했는지를 나타내는 상태 정보**도 갖는다. 다음에 실행할 명령어의 주소를 나타내는 프로그램 카운터, CPU 레지스터 값 등이 이에 해당한다. 운영체제는 이러한 정보를 보존하여 중단된 실행을 나중에 이어 갈 수 있게 한다.

③ X: 입출력 완료나 신호 수신 같은 사건을 기다리는 것은 **대기(waiting) 상태**이다. 준비(ready) 상태는 사건을 기다리는 상태가 아니라 **CPU를 할당받으면 실행할 수 있는 상태**이다. 예를 들어 입출력을 요청하고 완료를 기다리는 동안에는 대기 상태에 있고, 입출력이 끝나 실행할 수 있게 되면 준비 상태로 옮겨 CPU 할당을 기다린다.

④ O: 프로세스는 함수 호출에 필요한 반환 주소와 매개변수, 지역 변수 등을 저장하는 데 스택을 사용한다. 함수 호출과 복귀는 **나중에 호출된 함수가 먼저 복귀하는 후입선출 구조**와 맞는다. 예를 들어 A 함수가 B 함수를 호출하면 B의 실행을 마친 뒤 스택에 보관된 반환 주소를 이용해 A의 호출 다음 지점으로 돌아간다.

NoteBase

문 9. 자바 프로그래밍 언어에 대한 설명으로 옳은 것은?

① 클래스에서 상속을 금지하는 키워드는 this이다.

② 인터페이스(interface)는 추상 메소드를 포함할 수 없다.

③ 메소드 오버라이딩(overriding)은 상위 클래스에 정의된 메소드와 하위 클래스에서 재정의되는 메소드의 매개변수 개수와 자료형 등이 서로 다른 것을 의미한다.

④ 메소드 오버로딩(overloading)은 한 클래스 내에 동일한 이름의 메소드가 여러 개 있고 그 메소드들의 매개변수 개수 또는 자료형 등이 서로 다른 것을 의미한다.

정답 4번

간단한 해설

메소드 오버로딩은 **같은 이름의 메소드를 매개변수 개수나 자료형이 다르게** 정의하는 것이다. 호출할 때 전달한 인수에 맞는 메소드가 선택된다.

선지별 해설

① X: 클래스의 상속을 금지하는 키워드는 **final**이다. final로 선언한 클래스는 하위 클래스를 만들 수 없다. this는 인스턴스 메소드나 생성자에서 현재 객체를 가리킬 때 사용하는 키워드이다.

② X: 인터페이스는 **추상 메소드를 포함할 수 있다**. 추상 메소드는 구현 본문 없이 선언되며, 인터페이스를 구현하는 구체 클래스는 해당 메소드를 구현해야 한다. 따라서 '포함할 수 없다'를 '포함할 수 있다'로 고쳐야 한다.

③ X: 오버라이딩은 하위 클래스가 상위 클래스에서 상속받은 메소드를 **같은 이름과 매개변수 목록으로 재정의**하는 것이다. 예를 들어 상위 클래스의 show()를 하위 클래스에서 같은 show()로 재정의하여 동작을 바꿀 수 있다. 매개변수 개수나 자료형을 달리하는 것은 오버로딩을 구분하는 기준이다.

④ O: 오버로딩은 같은 클래스 안에서 이름이 같은 메소드를 **매개변수 개수나 자료형이 다르게** 정의하는 것이다. 예를 들어 add(int a, int b)와 add(double a, double b)는 이름은 같지만 매개변수 자료형이 달라 함께 정의할 수 있다. 반환형만 다르고 매개변수 목록이 같으면 오버로딩이 성립하지 않는다.

NoteBase

문 10. 다음 C++ 프로그램의 실행 결과로 옳은 것은?

```
#include <iostream>
using namespace std;

class Student {
public:
     Student():Student(0) {};
     Student(int id):_id(id) {
           if (_id > 0) _cnt++;
     };
     static void print() { cout << _cnt;};
     void printID() { cout << ++_id;};

private:
     int _id;
     static int _cnt;
};

int Student::_cnt = 0;

int main() {
     Student A(2);
     Student B;
     Student C(4);
     Student D(-5);
     Student E;
     Student::print();
     E.printID();
     return 0;
}
```

① 21

② 22

③ 30

④ 31

정답 1번

간단한 해설

생성자에서 _id가 양수인 A와 C만 **공유 변수 _cnt를 증가**시키므로 첫 출력은 2이다. E의 _id는 0에서 **전위 증가**로 1이 된 뒤 출력되며, 두 값이 구분 없이 이어져 21이 된다.

NoteBase

계산 과정

변수와 생성자: _id는 객체마다 따로 가지는 멤버 변수이고, static으로 선언된 _cnt는 **모든 객체가 공유하는 하나의 변수**이다. _cnt의 초기값은 0이며, 생성자에서 _id > 0일 때만 1 증가한다. 따라서 _cnt는 생성된 전체 객체 수를 나타내지는 않는다. 기본 생성자 Student()는 Student(0)에 생성을 위임하므로, 인수 없이 만든 객체의 _id는 0이 된다.

A부터 C까지의 생성: Student A(2)는 A의 _id를 2로 설정하고 _cnt를 0에서 1로 증가시킨다. Student B는 기본 생성자를 거쳐 _id가 0이 되므로 _cnt는 1로 유지된다. Student C(4)는 _id가 4이므로 _cnt를 1에서 2로 증가시킨다.

D와 E의 생성: Student D(-5)는 _id가 -5이므로 양수 조건을 만족하지 않아 _cnt를 증가시키지 않는다. Student E도 기본 생성자를 거쳐 _id가 0이 되므로 _cnt를 증가시키지 않는다. 다섯 객체의 생성이 끝났을 때 공유 변수 _cnt는 2이고, E의 _id는 0이다.

Student::print() 실행: 정적 멤버 함수 print()가 공유 변수 _cnt의 현재 값인 2를 출력한다.

E.printID() 실행: ++_id는 **값을 먼저 1 증가시킨 뒤 그 값을 사용하는** 전위 증가 연산이다. 따라서 E의 _id가 0에서 1이 되고, 증가한 값 1이 출력된다. 이 호출은 E의 _id만 바꾸며 생성자를 다시 실행하지 않으므로 _cnt는 2로 유지된다.

출력 결합: 두 출력문 사이에 공백이나 줄바꿈이 없으므로 먼저 출력한 2와 뒤이어 출력한 1이 붙어 21이 된다. 이는 ①의 값과 일치한다.

NoteBase

문 11. 다음 C 프로그램의 실행 결과로 옳은 것은?

```
#include <stdio.h>
int main()
{
    int count, sum = 0;

    for ( count = 1; count <= 10; count++) {
        if ((count % 2) == 0)
            continue;
        else
            sum += count;
    }
    printf("%d\n", sum);
}
```

① 10
② 25
③ 30
④ 55

정답 2번

간단한 해설

count가 짝수이면 **continue로 덧셈을 건너뛰므로** 1부터 10까지의 홀수만 sum에 누적된다. 출력값은 1 + 3 + 5 + 7 + 9 = 25이다.

NoteBase

계산 과정

초기화와 반복 범위: sum은 누적 합을 저장하며 처음 값은 0이다. for문의 count는 1에서 시작하여 count <= 10인 동안 1씩 증가하므로, 반복 본문에서 처리하는 값은 1부터 10까지이다.

짝수 처리: %는 나눗셈의 나머지를 구하는 연산자이다. count % 2가 0이면 짝수이므로 continue가 실행된다. 이때 **현재 반복의 남은 처리를 건너뛰고** for문의 count++로 이동한 뒤 조건을 다시 검사한다. 따라서 짝수는 sum에 더해지지 않는다.

홀수 누적: count가 홀수이면 나머지가 1이므로 else의 sum += count가 실행된다. 이 문장은 기존 sum에 count를 더해 다시 sum에 저장한다. count가 1, 3, 5, 7, 9일 때 sum은 차례로 1, 4, 9, 16, 25가 된다.

출력: count가 10일 때도 짝수이므로 덧셈을 건너뛰고, count가 11이 되면 반복 조건이 거짓이 되어 종료한다. printf의 %d는 최종 sum을 정수로 출력하므로 실행 결과는 25이며, ②의 값과 일치한다.

NoteBase

문 12. 클래스기반 주소지정에서 IPv4 주소 131.23.120.5가 속하는 클래스는?

① Class A

② Class B

③ Class C

④ Class D

정답 2번

간단한 해설

클래스기반 주소지정에서 첫 번째 옥텟이 **128~191**이면 Class B이다. 131.23.120.5의 첫 번째 옥텟은 131이므로 이 범위에 속한다.

공통 해설

클래스기반 IPv4 주소는 **첫 번째 옥텟의 선두 비트**로 클래스를 구분한다. 옥텟은 점으로 구분한 각 8비트 부분을 뜻한다. Class A는 0, Class B는 10, Class C는 110, Class D는 1110으로 시작한다. 따라서 클래스는 주소의 나머지 세 옥텟과 관계없이 첫 번째 옥텟으로 판별한다.

131.23.120.5의 첫 번째 옥텟은 131이며, 이를 8비트 이진수로 나타내면 10000011이다. 선두 두 비트가 10이므로 Class B에 해당한다. 이는 Class B의 첫 번째 옥텟 범위인 128~191에 131이 포함된다는 뜻이기도 하다.

NoteBase

문 13. IPv4 CIDR 표기법에서 네트워크 접두사(prefix)의 길이가 25일 때, 이에 해당하는 서브넷 마스크(subnet mask)는?

① 255.255.255.0

② 255.255.255.128

③ 255.255.255.192

④ 255.255.255.224

정답 2번

간단한 해설

접두사 길이 25는 서브넷 마스크에서 **앞의 25비트가 1**이라는 뜻이다. 앞의 24비트는 255.255.255이고, 마지막 8비트는 10000000이므로 마스크는 255.255.255.128이다.

공통 해설

IPv4 주소는 32비트이며, 8비트씩 나눈 네 묶음을 각각 10진수로 표기한다. CIDR의 접두사 길이는 네트워크 부분의 비트 수를 나타낸다. 이에 대응하는 서브넷 마스크는 **네트워크 부분을 1, 나머지 호스트 부분을 0**으로 표시한다. 따라서 접두사 길이가 25이면 왼쪽부터 1이 25개, 그 뒤에 0이 7개 이어진다.

앞의 세 묶음은 각각 11111111이므로 10진수로 255이다. 이 세 묶음이 24비트를 차지하므로, 마지막 묶음에서는 맨 왼쪽 1비트만 1이고 나머지는 0인 10000000이 된다. 8비트 이진수의 맨 왼쪽 자리 값은 128이므로, /25에 대응하는 서브넷 마스크는 255.255.255.128이다.

NoteBase

문 14. 다음 설명에 해당하는 기술은?

◯ 클라이언트의 요구에 대한 응답 시간을 줄일 수 있다.
◯ 외부 인터넷과 연결된 트래픽을 줄일 수 있다.
◯ 최근 호출된 객체의 사본을 저장한다.

① DNS
② NAT
③ Router
④ Proxy server

정답 4번

간단한 해설

프록시 서버는 **최근 요청된 객체의 사본을 캐시에 저장**하고 재사용할 수 있다. 재사용 가능한 사본으로 요청에 응답하면 외부 서버에서 객체를 다시 받아오는 일을 줄여 응답 시간과 외부 인터넷 트래픽을 줄일 수 있다.

핵심 풀이

프록시 서버(Proxy server)는 클라이언트와 외부 서버 사이에서 요청과 응답을 중계한다. 캐시 기능을 갖춘 프록시 서버는 웹 문서나 이미지처럼 요청된 객체의 사본을 저장한다. 지문의 '최근 호출된 객체의 사본을 저장한다'는 설명은 이러한 **프록시 캐싱**을 가리킨다.

같은 객체를 다시 요청할 때 캐시에 재사용 가능한 사본이 있으면 외부 서버에서 객체 전체를 다시 전송받지 않고 응답할 수 있다. 예를 들어 여러 클라이언트가 같은 이미지를 요청하면 저장된 이미지 사본을 재사용할 수 있다. 이처럼 **외부 서버와 주고받는 데이터가 줄어들기 때문에** 외부 인터넷 트래픽과 응답 시간을 줄일 수 있다. 다만 캐시에 사본이 없거나 저장된 사본을 그대로 재사용할 수 없는 경우에는 외부 서버에 요청하거나 유효성을 확인해야 하므로, 모든 요청에서 같은 효과가 보장되는 것은 아니다.

DNS의 주된 역할은 도메인 이름에 대응하는 IP 주소 등의 정보를 조회하는 것이고, NAT는 패킷의 IP 주소 등을 변환하는 기술이다. 라우터(Router)는 목적지에 따라 패킷을 전달할 경로를 선택한다. 이들과 구별되는 지문의 핵심은 **요청된 객체 자체의 사본을 저장하여 재사용**한다는 점이다.

NoteBase

문 15. 노드 7, 13, 61, 38, 45, 26, 14를 차례대로 삽입하여 최대 히프(heap)를 구성한 뒤 이 트리를 중위 순회할 때, 첫 번째로 방문하는 노드는?

① 7

② 14

③ 45

④ 61

정답 1번

간단한 해설

주어진 순서로 최대 히프를 만들면 루트 61의 왼쪽 자식은 45이고, 45의 왼쪽 자식은 7이다. 중위 순회는 **왼쪽 서브트리 → 현재 노드 → 오른쪽 서브트리** 순서이므로 가장 왼쪽 끝의 7을 먼저 방문한다.

NoteBase

계산 과정

최대 히프는 완전 이진 트리이며 **부모의 값이 자식의 값보다 크거나 같다**. 새 노드는 마지막 레벨의 왼쪽부터 빈자리를 채우도록 삽입하고, 부모보다 크면 부모와 교환하며 위로 올라간다. 아래 배열은 트리를 위에서 아래로, 같은 레벨에서는 왼쪽부터 읽은 값이다.

7 삽입: 첫 노드이므로 루트에 놓는다. 배열은 [7]이다.

13 삽입: 7의 왼쪽 자식에 놓는다. 부모 7보다 크므로 서로 교환하여 [13, 7]이 된다.

61 삽입: 루트 13의 오른쪽 자식에 놓는다. 부모 13보다 크므로 서로 교환하여 [61, 7, 13]이 된다.

38 삽입: 7의 왼쪽 자식에 놓고 부모 7과 교환한다. 새 부모 61보다는 작으므로 멈추며, 배열은 [61, 38, 13, 7]이다.

45 삽입: 38의 오른쪽 자식에 놓고 부모 38과 교환한다. 새 부모 61보다는 작으므로 멈추며, 배열은 [61, 45, 13, 7, 38]이다.

26 삽입: 13의 왼쪽 자식에 놓고 부모 13과 교환한다. 새 부모 61보다는 작으므로 멈추며, 배열은 [61, 45, 26, 7, 38, 13]이다.

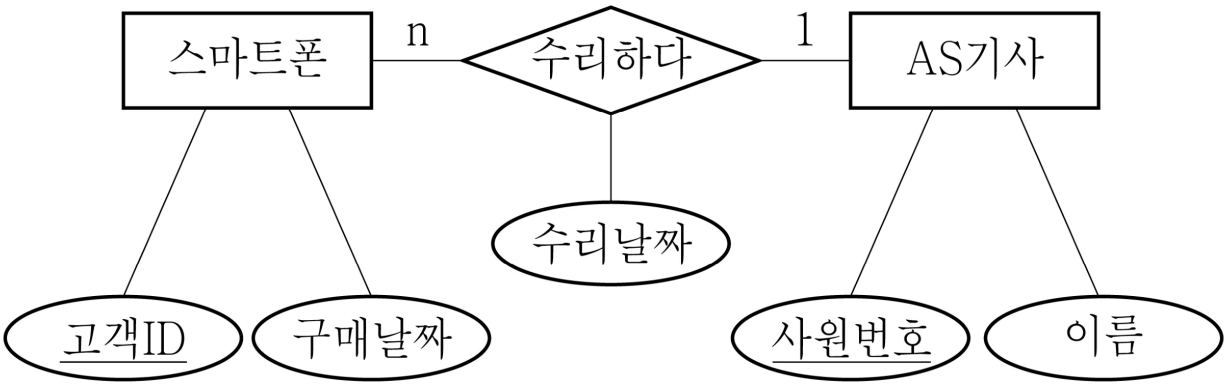
14 삽입: 26의 오른쪽 자식에 놓는다. 부모 26보다 작으므로 교환하지 않으며, 최종 배열은 [61, 45, 26, 7, 38, 13, 14]이다.

중위 순회: 각 노드에서 **왼쪽 서브트리를 먼저 순회한 뒤 현재 노드를 방문**하고, 이어 오른쪽 서브트리를 순회한다. 따라서 루트 61에서 왼쪽 자식 45로, 다시 왼쪽 자식 7로 내려간다. 7에는 왼쪽 자식이 없으므로 7을 처음 방문한다.

첫 번째 방문 노드는 7이며, 이는 ①의 값과 일치한다.

NoteBase

문 16. 다음 그림은 스마트폰 수리와 관련된 E-R 다이어그램의 일부이다. 이에 대한 설명으로 옳지 않은 것은?



① '수리하다' 관계는 속성을 가지고 있다.

② 'AS기사'와 '스마트폰'은 일대다 관계이다.

③ '스마트폰'은 다중값 속성을 가지고 있다.

④ '사원번호'는 키 속성이다.

정답 3번

간단한 해설

스마트폰에 연결된 '고객ID'와 '구매날짜'는 모두 단일 타원으로 표시되어 있다. 한 개체가 여러 값을 가질 수 있는 **다중값 속성은 이중 타원**으로 나타내므로, 이 그림에는 다중값 속성이 표시되어 있지 않다.

선지별 해설

① O: 마름모는 관계를, 타원은 속성을 나타낸다. '수리하다' 마름모에 '수리날짜' 타원이 연결되어 있으므로, **'수리날짜'는 '수리하다' 관계의 속성**이다. 개체뿐 아니라 관계도 속성을 가질 수 있다.

② O: 'AS기사' 쪽에 1, '스마트폰' 쪽에 n이 표시되어 있다. 이는 **AS기사 한 명이 여러 스마트폰을 수리할 수 있고**, 각 스마트폰은 한 명의 AS기사와 연결되는 일대다 관계를 뜻한다.

③ X: '스마트폰'의 '고객ID'와 '구매날짜'는 모두 단일 타원으로 그려져 있다. 다중값 속성은 한 개체에 대해 여러 값을 가질 수 있는 속성으로, **이중 타원**으로 표시한다. 따라서 그림의 스마트폰에는 다중값 속성이 아니라 단일값 속성이 표시되어 있다. 속성이 두 개 있다는 것과 하나의 속성이 여러 값을 가진다는 것은 다르다.

④ O: 속성 이름 아래의 **밑줄은 키 속성**을 나타낸다. '사원번호'에 밑줄이 있으므로, 사원번호는 각 AS기사를 유일하게 식별하는 키 속성이다.

NoteBase

문 17. UML(Unified Modeling Language) 버전 2.0에 대한 설명으로 옳지 않은 것은?

① 액터(actor)는 사람이 아닌 경우도 있다.

② 클래스(class) 다이어그램은 시스템의 클래스들과 그들 간의 연관을 보여준다.

③ 유스케이스(usecase) 다이어그램은 사용자와 시스템 간의 상호 작용을 보여준다.

④ 시퀀스(sequence) 다이어그램은 시스템이 내부 또는 외부 이벤트에 대해 어떻게 반응하는지 보여준다.

정답 4번

간단한 해설

④의 '내부 또는 외부 이벤트에 대해 어떻게 반응하는지'를 상태 변화로 나타내는 것은 **상태 머신 다이어그램**의 역할이다. 시퀀스 다이어그램은 상호 작용에 참여하는 객체들이 주고받는 **메시지의 시간적 순서**를 보여준다.

선지별 해설

① O: 액터는 모델링 대상 시스템의 외부에서 시스템과 상호 작용하는 역할이다. **사람뿐 아니라 외부 시스템이나 장치**도 액터가 될 수 있다. 예를 들어 쇼핑몰 시스템과 결제 정보를 주고받는 외부 결제 시스템도 액터에 해당한다.

② O: 클래스 다이어그램은 클래스의 속성과 연산, 클래스 사이의 연관·일반화 등의 관계를 표현한다. 따라서 시스템을 구성하는 클래스들과 그 관계를 보여 주는 **정적 구조** 다이어그램이다.

③ O: 유스케이스 다이어그램은 시스템이 사용자에게 제공하는 기능을 유스케이스로 나타내고, 그 기능에 참여하는 액터와의 관계를 표현한다. 이를 통해 **사용자 관점에서 시스템과의 상호 작용**을 파악할 수 있다.

④ X: 시퀀스 다이어그램은 객체 등의 참여자를 생명선으로 나타내고, 이들 사이에 전달되는 **메시지의 시간적 순서**를 표현한다. 이벤트에 반응하여 시스템이나 객체의 상태가 어떻게 변하는지 나타내는 것은 **상태 머신 다이어그램**이다. 예를 들어 요청 이벤트를 받아 대기 상태에서 처리 상태로 전이하는 모습은 상태 머신 다이어그램으로 표현한다.

NoteBase

문 18. 같은 값을 옳게 나열한 것은?

① $(264)_8$, $(181)_{10}$
② $(263)_8$, $(AC)_{16}$
③ $(10100100)_2$, $(265)_8$
④ $(10101101)_2$, $(AD)_{16}$

정답 4번

간단한 해설

이진수는 **오른쪽부터 네 자리씩 묶어** 16진수 한 자리로 바꿀 수 있다. $(10101101)_2$의 두 묶음 1010과 1101은 각각 A와 D이므로 $(AD)_{16}$과 같은 값이다.

선지별 해설

서로 다른 진법의 수는 10진수로 바꾸어 값을 비교할 수 있다. 각 자리 숫자에 오른쪽부터 1, 밑, 밑의 제곱에 해당하는 자릿값을 곱해 더한다. 16진수에서 A, B, C, D는 각각 10, 11, 12, 13을 나타낸다.

① X: $(264)_8$의 10진수 값은 $2 \cdot 8^2 + 6 \cdot 8 + 4 = 180$이다. 따라서 181과 같지 않으며, 같은 값으로 나타내려면 **10진수 값을 180으로** 고쳐야 한다.

② X: $(263)_8$은 $2 \cdot 8^2 + 6 \cdot 8 + 3 = 179$이고, $(AC)_{16}$은 $10 \times 16 + 12 = 172$이다. 두 값이 다르며, 179는 $11 \times 16 + 3$이므로 $(263)_8$과 같은 16진수는 $(B3)_{16}$이다.

③ X: $(10100100)_2$는 1인 자리의 자릿값을 더하면 $128 + 32 + 4 = 164$이다. 반면 $(265)_8$은 $2 \cdot 8^2 + 6 \cdot 8 + 5 = 181$이다. 이진수를 오른쪽부터 세 자리씩 묶으면 010 100 100이므로, 같은 값의 8진수는 $(244)_8$이다. 맨 왼쪽에 보충한 0은 값을 바꾸지 않는다.

④ O: $(10101101)_2$는 $128 + 32 + 8 + 4 + 1 = 173$이고, $(AD)_{16}$은 $10 \times 16 + 13 = 173$이므로 두 수의 값이 같다. 16은 2의 네 제곱이므로 **이진수 네 자리가 16진수 한 자리에 대응**한다. 1010 1101에서 1010은 10인 A, 1101은 13인 D에 대응하여 AD가 된다.

NoteBase

문 19. 관계형 데이터베이스에 대한 설명으로 옳은 것만을 모두 고르면?

ㄱ. 관계형 데이터베이스 스키마(schema)는 릴레이션 스키마의 집합과 무결성 제약조건(integrity constraint)으로 구성된다.
ㄴ. 개체(entity) 무결성 제약조건은 기본 키(primary key)를 구성하는 모든 속성은 널(null) 값을 가지면 안된다는 규칙이다.
ㄷ. 참조(referential) 무결성 제약조건이란 외래 키(foreign key)는 참조할 수 없는 값을 가질 수 없다는 규칙이다.
ㄹ. 후보 키(candidate key)가 되기 위해서는 유일성(uniqueness)과 효율성(efficiency)을 항상 만족해야 한다.

① ㄱ, ㄴ, ㄷ
② ㄱ, ㄴ, ㄹ
③ ㄱ, ㄷ, ㄹ
④ ㄴ, ㄷ, ㄹ

정답 1번

간단한 해설

후보 키의 필수 조건은 **유일성과 최소성**이다. 유일성은 각 튜플을 구별할 수 있다는 뜻이고, 최소성은 그 구별 능력을 유지하면서 속성을 더 뺄 수 없다는 뜻이다.

NoteBase

보기별 해설

ㄱ. O: 관계형 데이터베이스 스키마는 각 테이블의 구조를 나타내는 **릴레이션 스키마의 집합과 무결성 제약조건**으로 구성된다. 릴레이션 스키마는 릴레이션 이름, 속성 이름과 각 속성이 가질 수 있는 값의 범위 등을 정하고, 무결성 제약조건은 저장되는 데이터가 지켜야 할 규칙을 정한다.

ㄴ. O: 개체 무결성 제약조건은 **기본 키를 구성하는 모든 속성에 널을 허용하지 않는다**는 규칙이다. 기본 키는 각 튜플을 식별하는 기준이므로 그 값이 빠져 있으면 안 된다. 여러 속성을 묶어 기본 키로 정한 경우에도 그중 어느 속성에도 널이 들어갈 수 없다.

ㄷ. O: 참조 무결성 제약조건에 따라 **널이 아닌 외래 키 값은 참조 대상의 키 값과 일치해야 한다**. 예를 들어 학생 테이블의 학과번호가 학과 테이블을 참조한다면, 학과번호에 값을 넣을 때는 학과 테이블에 존재하는 번호를 사용해야 한다. 다만 널을 금지하는 별도 제약이 없다면 외래 키에 널을 허용할 수 있으므로, 이 보기의 '참조할 수 없는 값'은 참조 대상에 존재하지 않는 값을 뜻한다.

ㄹ. X: 후보 키가 되려면 유일성과 **최소성**을 만족해야 하며, 효율성은 후보 키의 정의 요건이 아니다. 유일성은 키 값으로 각 튜플을 유일하게 식별할 수 있다는 뜻이다. 최소성은 그 속성 집합에서 하나라도 속성을 빼면 유일성이 유지되지 않는다는 뜻이다. 예를 들어 학번만으로 학생을 구별할 수 있다면, 학번과 이름을 묶은 집합은 유일성을 만족해도 이름을 뺄 수 있으므로 후보 키가 아니다.

옳은 보기는 ㄱ, ㄴ, ㄷ이므로 이 조합을 제시한 ①에 해당한다.

NoteBase

문 20. IT 기술에 대한 설명으로 옳지 않은 것은?

① IoT는 각종 물체에 센서와 통신 기능을 내장해 인터넷에 연결하는 기술이다.

② ITS는 기존 교통체계의 구성 요소에 첨단 기술들을 적용시켜 보다 안전하고 편리한 통행과 전체 교통체계의 효율성을 높이는 시스템이다.

③ IPTV는 인터넷을 이용하여 방송 및 기타 콘텐츠를 TV로 제공하는 서비스 방식이다.

④ GIS는 라디오 주파수를 이용한 비접촉 인식 장치로 태그와 리더기로 구성된 자동 인식 데이터 수집용 무선 통신 시스템이다.

정답 4번

간단한 해설

④의 '라디오 주파수로 태그와 리더기 사이에서 정보를 주고받는 자동 인식 시스템'은 **RFID**에 대한 설명이다. **GIS는 지리정보시스템**으로, 위치와 관련된 정보를 저장·관리·분석하는 데 쓰인다.

선지별 해설

① O: IoT는 사물인터넷으로, 센서와 통신 기능을 갖춘 사물을 **인터넷에 연결해 데이터를 수집·교환**하는 기술이다. 예를 들어 온도 센서가 측정값을 인터넷으로 보내면 사용자가 원격으로 온도를 확인할 수 있다.

② O: ITS는 지능형 교통체계로, 도로·차량·교통시설 등에 정보·통신·제어 기술을 적용한다. 교통 상황을 실시간으로 수집해 운전자에게 제공하거나 신호를 조절하는 등 **교통의 안전성과 운영 효율을 높이는** 시스템이다.

③ O: IPTV는 인터넷 프로토콜을 이용하는 통신망으로 방송과 각종 콘텐츠를 전달하여 **TV에서 시청할 수 있게 하는** 서비스 방식이다. 인터넷을 통한 데이터 전달 기술을 방송 서비스에 이용한다는 점이 핵심이다.

④ X: 선지의 태그와 리더기를 이용한 비접촉 자동 인식은 **RFID(무선 주파수 식별)**이다. 리더기가 전파를 이용해 태그에 저장된 식별 정보를 읽으므로 직접 접촉하지 않고 대상을 인식할 수 있다. **GIS(지리정보시스템)**는 위치와 연결된 공간정보와 속성정보를 저장·관리·분석하고 지도 등으로 표현하는 시스템이다. 따라서 선지의 'GIS'를 'RFID'로 고쳐야 한다.

NoteBase