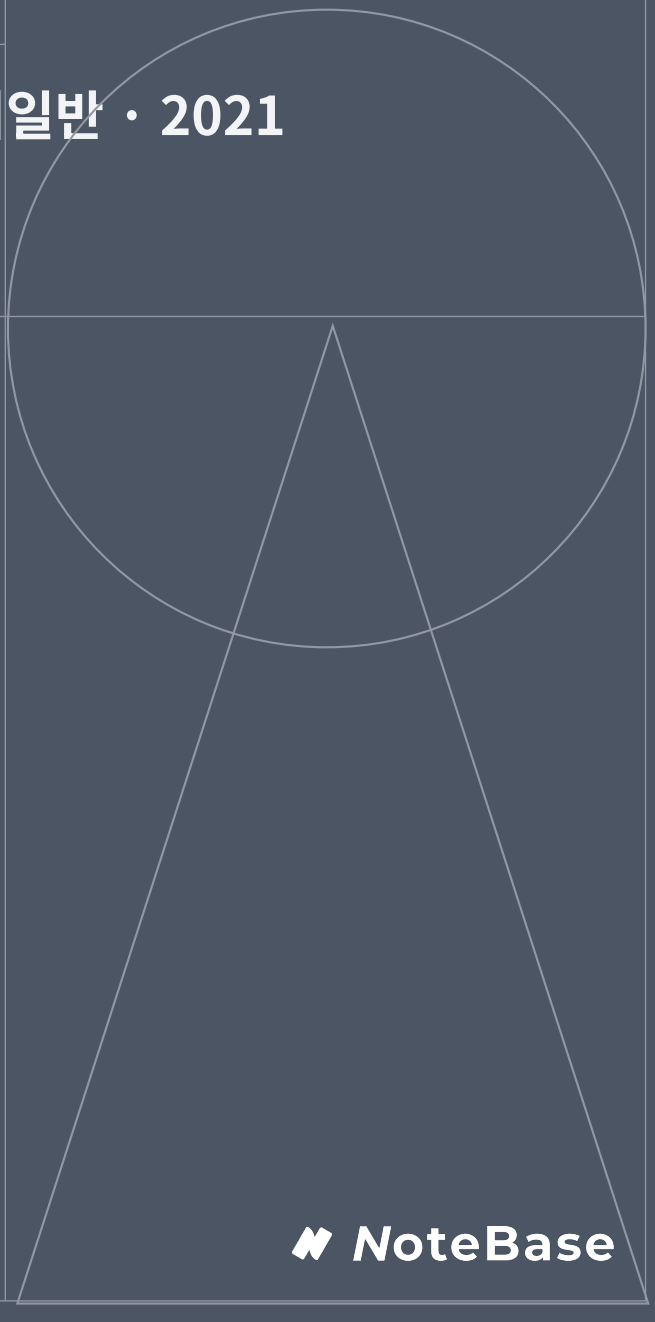
기출OX 해설집

공무원/국회직/9급 · 컴퓨터일반 · 2021



NoteBase

기출OX 해설집

공무원/국회직/9급 · 컴퓨터일반 · 2021

공무원/국회직/9급 · 컴퓨터일반 · 2021

1. 다음 카르노맵으로 표현되는 최적화된 논리식의 결과 Y로 옳은 것은?(X는 don't care를 나타낸다)

C \ AB	00	01	11	10
0	1	0	0	X
1	X	0	1	1

① Y = AB + C
② Y = AC + B'
③ Y = A + B + C
④ Y = A'B + C
⑤ Y = ABC'

정답 2번

출제 포인트

카르노맵에서 인접한 1과 don't care를 2의 거듭제곱 개수로 묶어 최소 논리식을 구하는 문제이다.

조건 정리

주어진 값: 1인 칸은 ABC = 000, 101, 111이고, don't care인 칸은 001, 100이다.
구해야 할 값: 1인 모든 칸을 포함하는 최소 곱의 합 논리식이다.
주의할 조건: don't care는 묶음을 크게 만드는 데 필요할 때 1로 사용할 수 있으며, 카르노맵의 맨 왼쪽 열과 맨 오른쪽 열도 서로 인접한다.

NoteBase

사용할 공식

카르노맵에서 묶음 안에서 값이 변하는 변수는 소거하고, 일정한 변수만 곱항에 남긴다. 각 묶음에서 얻은 곱항을 OR로 더한다.

이 원리를 쓰는 이유: 주어진 1과 don't care를 인접 묶음으로 만들면 변수 수가 가장 적은 논리식을 직접 얻을 수 있기 때문이다.

계산 과정

1. 열 $AB = 00$과 $AB = 10$은 양 끝에서 서로 인접한다. 두 열의 두 행을 don't care까지 포함해 4칸으로 묶으면 A와 C는 변하고 $B = 0$만 일정하므로 B'가 남는다.
2. 아래 행 $C = 1$에서 열 $AB = 11$과 $AB = 10$의 두 1을 묶으면 B는 변하고 $A = 1, C = 1$이 일정하므로 AC가 남는다.
3. 두 묶음의 항을 OR로 결합하면 $Y = B' + AC$가 된다.

선지 판단

① X — $Y = AB + C$는 $ABC = 000$에서 0이지만 카르노맵의 값은 1이다. 틀린 식이며, 올바르게는 $Y = B' + AC$이다.
② O — B'는 양 끝 두 열의 4칸 묶음에서, AC는 아래 행의 오른쪽 두 칸 묶음에서 얻어지므로 모든 1을 포함하는 최소식이다.
③ X — $Y = A + B + C$는 $ABC = 000$에서 0이 되어 카르노맵의 1을 포함하지 못한다. 틀린 식이며, 올바르게는 $Y = B' + AC$이다.
④ X — $Y = A'B + C$는 $ABC = 000$에서 0이 되어 카르노맵의 1과 일치하지 않는다. 틀린 식이며, 올바르게는 $Y = B' + AC$이다.
⑤ X — $Y = ABC'$는 $ABC = 110$에서 1이지만 카르노맵의 해당 칸은 0이다. 틀린 식이며, 올바르게는 $Y = B' + AC$이다.

따라서 계산 결과 $Y = B' + AC$이므로 정답은 2번이다.

오답 포인트

양 끝 열이 인접한다는 점과 don't care를 1처럼 활용할 수 있다는 점을 놓치면 4칸 묶음의 B'를 찾지 못한다. 또한 묶음 안에서 변하는 변수는 남기는 것이 아니라 소거한다.

다시 나오면 이렇게 판단

먼저 1을 모두 포함하도록 don't care와 가장자리 인접 관계를 활용해 가능한 가장 큰 묶음을 만든다. 그다음 각 묶음에서 일정한 변수만 남기고, 얻은 곱항들을 OR로 결합한다.

NoteBase

암기 포인트

카르노맵은 크게 묶을수록 변수가 줄어든다. 양 끝은 인접하고, don’t care는 큰 묶음에 선택적으로 사용한다.

NoteBase

2. 컴퓨터 프로세서의 종류는 RISC(Reduced Instruction Set Computer) 방식과 CISC(Complex Instruction Set Computer) 방식으로 나눌 수 있다. CISC 방식의 구조를 적용한 프로세서는?

① Intel(인텔) Core i3
② Apple(애플) A14
③ Qualcomm(퀄컴) Snapdragon 860
④ Samsung(삼성) Exynos 850
⑤ MIPS(밉스) microAptiv processor

정답 1번

출제 포인트

이 문제는 프로세서가 사용하는 명령어 집합 구조를 기준으로 RISC 계열과 CISC 계열을 구분하는 문제이다.

핵심 개념

CISC는 하나의 명령어로 복합적인 작업을 수행할 수 있도록 다양하고 복잡한 명령어를 제공하는 구조이며, 대표적인 계열은 x86이다.
RISC는 자주 사용하는 단순한 명령어를 중심으로 명령어 집합을 구성하여 빠르고 효율적인 실행을 지향하는 구조이며, 대표적인 계열은 ARM과 MIPS이다.
이 문항에서는 x86 기반의 Intel Core i3만 CISC에 해당하고, ARM 또는 MIPS 기반인 나머지 프로세서는 RISC에 해당한다.

선지별 해설

① O — Intel Core i3는 x86 명령어 집합을 사용하는 프로세서로, x86은 대표적인 CISC 구조이다.
② X — Apple A14를 CISC 프로세서로 본 부분이 틀리다. Apple A14는 ARM 계열 명령어 집합을 사용하는 RISC 기반 프로세서이다.
③ X — Qualcomm Snapdragon 860을 CISC 프로세서로 본 부분이 틀리다. Snapdragon 860의 CPU는 ARM 계열 구조를 바탕으로 하므로 RISC 기반 프로세서이다.
④ X — Samsung Exynos 850을 CISC 프로세서로 본 부분이 틀리다. Exynos 850의 CPU는 ARM Cortex 계열을 사용하므로 RISC 기반 프로세서이다.
⑤ X — MIPS microAptiv processor를 CISC 프로세서로 본 부분이 틀리다. microAptiv는 MIPS 명령어 집합을 사용하는 RISC 기반 프로세서이다.

NoteBase

오답 포인트

제품명이나 제조사만 보고 판단하면 혼동하기 쉽다. ②~④는 서로 다른 회사의 모바일 프로세서이지만 모두 ARM 계열의 RISC 기반이고, ⑤도 이름에 MIPS가 명시된 RISC 기반이다. 따라서 ①의 x86 계열만 CISC라는 점이 핵심이다.

다시 나오면 이렇게 판단

프로세서의 제조사가 아니라 명령어 집합 계열을 확인한다. x86 계열은 CISC, ARM과 MIPS 계열은 RISC로 분류한다.

암기 포인트

x86은 CISC, ARM·MIPS는 RISC이다.

NoteBase

3. 프로세서에서 I/O 장치의 요청을 처리하는 방법에는 인터럽트(Interrupt) 방식과 폴링(Polling) 방식이 있다. 폴링 방식의 단점으로 옳은 것은?

① I/O 장치가 다른 요청을 처리하는 동안 문맥교환(context switch)을 통해서 다른 작업을 처리할 수 있다.
② 프로세서가 반복적으로 I/O 장치의 상태를 체크해야 한다.
③ 프로세서가 문맥교환(context switch)을 하지 않고 I/O 장치의 요청을 처리할 수 있다.
④ 프로세서가 문맥교환(context switch)을 빈번하게 수행한다.
⑤ I/O 장치에서 요청이 있어야 처리 루틴을 시작한다.

정답 2번

출제 포인트

인터럽트 방식과 폴링 방식의 차이를 이해하고, 폴링 방식에서 프로세서가 I/O 장치의 상태를 반복 확인하느라 처리 시간을 소비한다는 단점을 판단하는 문제이다.

핵심 개념

폴링은 프로세서가 I/O 장치의 준비 여부나 요청 발생 여부를 주기적으로 직접 확인하는 방식이다. 구현이 단순하지만 장치가 준비되지 않은 동안에도 상태를 계속 검사하므로 CPU 시간이 낭비될 수 있다.
인터럽트는 I/O 장치가 처리가 필요할 때 프로세서에 신호를 보내는 방식이다. 프로세서는 신호를 기다리는 동안 다른 작업을 수행할 수 있지만, 인터럽트 처리 과정에서 실행 흐름의 전환과 문맥 저장·복원에 따른 오버헤드가 발생할 수 있다.

NoteBase

선지별 해설

① X — I/O 처리 중 다른 작업으로 전환할 수 있다는 내용은 폴링의 단점이 아니다. 틀린 표현은 '폴링 방식의 단점'으로 보는 부분이며, 올바르게는 인터럽트나 비동기 I/O를 이용하면 대기 중인 작업 대신 다른 작업을 수행하여 CPU를 효율적으로 활용할 수 있다는 설명이다.
② O — 폴링에서는 프로세서가 I/O 장치의 상태를 반복적으로 확인해야 한다. 장치가 아직 준비되지 않았더라도 검사 작업이 계속되어 CPU 시간이 낭비될 수 있으므로 대표적인 단점이다.
③ X — 문맥교환 없이 장치 상태를 직접 검사하여 요청을 처리할 수 있다는 것은 폴링의 단순한 처리 특성이지 단점이 아니다. 틀린 표현은 이를 '폴링 방식의 단점'으로 보는 부분이며, 올바르게는 폴링의 단점은 문맥교환의 부재가 아니라 반복 검사에 따른 CPU 시간 낭비이다.
④ X — 폴링은 상태 확인을 반복하지만 그때마다 문맥교환을 수행하는 것은 아니다. 틀린 표현은 '문맥교환을 빈번하게 수행한다'이며, 올바르게는 폴링이 빈번하게 수행하는 것은 I/O 장치의 상태 검사이다.
⑤ X — 처리 루틴은 프로세서가 폴링으로 장치의 요청 또는 준비 상태를 확인한 뒤 실행되므로, 단순히 I/O 장치에 요청이 있어야 한다는 사실 자체는 폴링의 단점이 아니다. 틀린 표현은 이를 '폴링 방식의 단점'으로 보는 부분이며, 올바르게는 장치의 요청을 발견할 때까지 프로세서가 상태를 반복 확인한다는 점이 단점이다.

오답 포인트

'반복적으로 수행한다'는 표현 때문에 ④를 고르기 쉽지만, 폴링에서 반복되는 것은 문맥교환이 아니라 장치 상태 확인이다. 또한 ③의 문맥교환 없이 처리한다는 특성과 ②의 반복 검사로 CPU 시간을 소비한다는 단점을 구분해야 한다.

다시 나오면 이렇게 판단

프로세서가 장치 상태를 계속 묻는다면 폴링이고, 장치가 프로세서에 처리 필요를 알린다면 인터럽트이다.
폴링의 단점을 물으면 반복 검사에 따른 바쁜 대기와 CPU 시간 낭비를 우선 판단한다.

암기 포인트

폴링은 CPU가 계속 확인하고, 인터럽트는 장치가 CPU에 알린다.

NoteBase

4. 데이터베이스에서 사용되는 NULL에 대한 설명으로 옳은 것만을 모두 고르면?

ㄱ. NULL은 데이터 값이 존재하지 않는다는 것을 나타내는 특별한 표시어(special marker)이다.
ㄴ. NULL인 데이터를 검색할 때 IS NULL을 사용한다.
ㄷ. NULL은 값이 없으므로 "0(Zero)" 값을 가지고 있다.

① ㄱ
② ㄱ, ㄴ
③ ㄱ, ㄷ
④ ㄴ, ㄷ
⑤ ㄱ, ㄴ, ㄷ

정답 2번

출제 포인트

NULL의 의미와 NULL 값을 검색할 때 사용하는 조건식을 구별하는 문제이다.

핵심 개념

NULL은 데이터 값이 존재하지 않거나 알려지지 않은 상태를 나타내는 특별한 표시이다. 숫자 0이나 빈 문자열처럼 실제로 저장된 값과는 다르다.
NULL 여부는 일반 비교 연산자가 아니라 IS NULL 또는 IS NOT NULL로 판단한다.

NoteBase

선지별 해설

① X
ㄱ은 옳지만, NULL 검색에 IS NULL을 사용한다는 ㄴ도 옳으므로 ㄱ만 고른 것이 틀렸다. 올바른 구성은 ㄱ, ㄴ이다.

② O
ㄱ은 NULL의 의미를 올바르게 설명하고, ㄴ은 NULL 검색 조건을 올바르게 제시한다. ㄷ은 틀렸으므로 ㄱ, ㄴ만 묶은 ②가 옳다.

③ X
ㄱ은 옳지만 ㄷ의 '0(Zero) 값을 가지고 있다'는 설명이 틀렸다. NULL은 0이라는 값을 가진 것이 아니라 값이 없거나 알려지지 않은 상태를 나타낸다. 올바른 구성은 ㄱ, ㄴ이다.

④ X
ㄴ은 옳지만 ㄷ이 틀렸고, 옳은 ㄱ이 빠져 있다. ㄷ은 'NULL은 0이 아니라 값의 부재 또는 미확정 상태를 나타낸다'로 고쳐야 하며, 올바른 구성은 ㄱ, ㄴ이다.

⑤ X
ㄱ과 ㄴ은 옳지만 ㄷ이 틀렸으므로 세 문장을 모두 고를 수 없다. ㄷ은 'NULL은 0과 구별되는 값의 부재 또는 미확정 상태를 나타낸다'로 고쳐야 하며, 올바른 구성은 ㄱ, ㄴ이다.

오답 포인트

ㄷ의 '값이 없으므로 0'이라는 표현이 함정이다. 0은 존재하는 숫자 값이지만 NULL은 값이 없거나 알려지지 않았음을 나타내므로 서로 다르다.

다시 나오면 이렇게 판단

NULL 관련 문항에서는 먼저 NULL과 0·빈 문자열을 구별하고, 검색 조건에 IS NULL 또는 IS NOT NULL이 쓰였는지 확인한다.

암기 포인트

NULL은 0이 아니며, NULL 검색은 IS NULL로 한다.

NoteBase

5. 분산 데이터 저장 환경에 관리 대상 데이터를 저장함으로써 누구도 임의로 수정할 수 없고 누구나 변경의 결과를 열람할 수 있도록 하는 분산 컴퓨팅 기술은?

① 트랜잭션(Transaction)
② 해시함수(Hash Function)
③ 블록체인(Blockchain)
④ 데이터 마이닝(Data Mining)
⑤ 인공신경망(Artificial Neural Network)

정답 3번

출제 포인트

관리 대상 데이터를 분산 저장하고, 임의 변경을 어렵게 하며, 참여자들이 변경 결과를 함께 확인할 수 있게 하는 기술이 무엇인지 묻는 문제이다.

핵심 개념

블록체인은 거래 등의 데이터를 블록 단위로 기록하고, 각 블록을 해시로 연결한 원장을 여러 참여 노드에 분산 저장하는 기술이다. 참여 노드들이 원장 사본을 공유하고 합의에 따라 기록을 갱신하므로 특정 참여자가 과거 기록을 임의로 바꾸기 어렵고 변경 내역을 공동으로 확인할 수 있다.
트랜잭션은 처리 작업의 단위, 해시함수는 입력값을 고정 길이의 해시값으로 변환하는 함수, 데이터 마이닝은 데이터에서 유용한 패턴을 찾는 분석 과정, 인공신경망은 데이터의 패턴을 학습하는 계산 모델이다.

NoteBase

선지별 해설

① X — 트랜잭션은 하나의 논리적 작업 단위이며, 그 자체가 데이터를 여러 노드에 분산 저장하는 기술은 아니다. 틀린 표현은 '트랜잭션이 해당 분산 컴퓨팅 기술이다'라는 대응이다. 올바르게는 트랜잭션을 데이터 처리에서 하나의 작업 단위라고 한다.
② X — 해시함수는 임의 길이의 입력을 고정 길이의 해시값으로 변환한다. 블록 연결과 위변조 탐지에 활용되지만, 해시함수만으로 원장을 분산 저장하고 참여자들이 변경 결과를 공유하지는 않는다. 틀린 표현은 '해시함수 자체가 분산 원장 기술이다'라는 대응이다. 올바르게는 해시함수를 블록체인을 구성하는 암호학적 도구 중 하나라고 한다.
③ O — 블록체인은 블록을 해시로 연결한 원장을 여러 노드에 분산 저장하고 참여자 간 합의를 통해 갱신한다. 따라서 한 참여자가 기록을 임의로 바꾸기 어렵고 참여자들이 변경된 원장 상태를 함께 확인할 수 있으므로 문제의 설명에 해당한다.
④ X — 데이터 마이닝은 대량의 데이터에서 규칙, 관계, 패턴 등 유용한 정보를 찾아내는 분석 기술이다. 틀린 표현은 '데이터 마이닝이 데이터의 임의 수정을 막는 분산 저장 기술이다'라는 대응이다. 올바르게는 데이터 마이닝을 데이터 분석·지식 발견 기술이라고 한다.
⑤ X — 인공신경망은 뉴런의 연결 구조를 모방하여 데이터의 패턴을 학습하고 분류·예측 등에 사용하는 모델이다. 틀린 표현은 '인공신경망이 변경 불가능한 분산 원장을 제공한다'라는 대응이다. 올바르게는 인공신경망을 기계학습 모델의 하나라고 한다.

오답 포인트

블록체인에서 해시함수를 사용한다는 사실 때문에 ②를 고르기 쉽다. 그러나 해시는 데이터의 변화를 탐지하고 블록을 연결하는 구성 요소일 뿐이며, 원장의 분산 저장과 참여자 간 합의까지 포함하는 전체 기술은 블록체인이다.

다시 나오면 이렇게 판단

문제에 '분산 원장', '여러 참여자의 원장 공유', '합의에 따른 기록 갱신', '위변조가 어려움'이 함께 제시되면 블록체인으로 판단한다. 단순히 입력을 일정한 길이의 값으로 바꾸거나 데이터 변화를 탐지하는 기능만 묻는다면 해시함수로 판단한다.

암기 포인트

해시는 블록을 잇는 도구이고, 블록체인은 원장을 분산해 함께 검증하는 구조이다.

NoteBase

6. 다음 데이터를 순서대로 입력하여 이진 탐색 트리를 만들었다. 이 트리의 단말 노드의 수로 옳은 것은?

데이터 입력: 12 16 7 28 60 31 23 5 35 25 2 10 15 6

① 4개
② 5개
③ 6개
④ 7개
⑤ 8개

정답 3번

출제 포인트

주어진 값을 입력 순서대로 이진 탐색 트리에 삽입한 뒤, 자식이 하나도 없는 단말 노드의 수를 구하는 문제이다.

올바른 순서

1. 12를 루트로 둔다.
2. 이후 값은 현재 노드보다 작으면 왼쪽, 크면 오른쪽으로 이동하여 빈 자리에 삽입한다.
3. 모든 값을 삽입하면 주요 연결은 12의 왼쪽에 7, 오른쪽에 16이 놓이고, 7의 왼쪽에는 5, 오른쪽에는 10이 놓인다. 5의 왼쪽에는 2, 오른쪽에는 6이 놓인다.
4. 16의 왼쪽에는 15, 오른쪽에는 28이 놓이고, 28의 왼쪽에는 23, 오른쪽에는 60이 놓인다. 23의 오른쪽에는 25가, 60의 왼쪽에는 31이, 31의 오른쪽에는 35가 놓인다.
5. 자식이 없는 노드는 2, 6, 10, 15, 25, 35이므로 단말 노드는 모두 6개이다.

왜 이 순서: 이진 탐색 트리의 각 삽입 위치는 앞서 삽입된 노드들과의 대소 비교로 결정되므로 반드시 제시된 입력 순서를 유지해야 한다.

선지별 해설

① X — 단말 노드는 4개가 아니라 2, 6, 10, 15, 25, 35의 6개이다.
② X — 단말 노드는 5개가 아니라 2, 6, 10, 15, 25, 35의 6개이다.
③ O — 자식이 없는 노드가 2, 6, 10, 15, 25, 35로 모두 6개이므로 옳다.
④ X — 단말 노드는 7개가 아니라 2, 6, 10, 15, 25, 35의 6개이다.
⑤ X — 단말 노드는 8개가 아니라 2, 6, 10, 15, 25, 35의 6개이다.

NoteBase

오답 포인트

값이 작은 노드나 트리의 가장 아래쪽에 있는 노드만 단말 노드라고 판단하면 틀린다. 단말 노드는 위치나 값과 관계없이 왼쪽 자식과 오른쪽 자식이 모두 없는 노드이며, 23과 31처럼 자식이 하나라도 있는 노드는 제외한다.

다시 나오면 이렇게 판단

입력 순서를 유지하며 각 값을 대소 비교로 삽입한 다음, 완성된 트리에서 왼쪽·오른쪽 자식이 모두 없는 노드만 표시하여 센다.

암기 포인트

이진 탐색 트리는 작은 값은 왼쪽, 큰 값은 오른쪽이며, 단말 노드는 자식이 0개인 노드이다.

NoteBase

7. **소프트웨어 테스트 방법 중 한 모듈의 수정이 다른 부분에 영향을 끼칠 수도 있다고 생각하여 수정 전 모듈 뿐 아니라 관련된 모듈까지 문제가 없는지 검사하는 테스트 방법은?**

① 회귀 테스트(Regression Test)
② 인수 테스트(Acceptance Test)
③ 통합 테스트(Integration Test)
④ 단위 테스트(Unit Test)
⑤ 시스템 테스트(System Test)

정답 1번

출제 포인트

소프트웨어를 수정한 뒤 그 변경이 기존 기능이나 연관 기능에 예상하지 못한 결함을 일으키지 않았는지 확인하는 테스트를 구별하는 문제이다.

핵심 개념

회귀 테스트는 소프트웨어 변경 후 이전에 정상 작동하던 기능과 관련 기능이 여전히 올바르게 작동하는지 다시 검사하는 테스트이다.
인수 테스트는 사용자·고객의 요구사항과 인수 기준을 충족하는지 확인하고, 통합 테스트는 결합된 모듈 사이의 인터페이스와 상호작용을 확인한다. 단위 테스트는 개별 모듈을 검사하며, 시스템 테스트는 통합된 전체 시스템을 대상으로 시스템 요구사항의 충족 여부를 검사한다.

선지별 해설

① O — 회귀 테스트는 수정으로 인한 부작용이 수정된 부분뿐 아니라 기존 기능과 관련 모듈에 발생하지 않았는지 재검사한다. 문제의 설명과 일치한다.
② X — '수정의 영향으로 기존·연관 기능에 문제가 없는지 검사한다'는 설명을 인수 테스트라고 한 부분이 틀렸다. 올바르게는 인수 테스트는 완성된 소프트웨어가 사용자·고객의 요구사항과 인수 기준을 충족하여 인수 가능한지 확인하는 테스트이다.
③ X — '변경 후 기존 기능과 관련 모듈의 이상 여부를 검사한다'는 설명을 통합 테스트라고 한 부분이 틀렸다. 올바르게는 통합 테스트는 여러 모듈을 결합한 뒤 모듈 간 인터페이스, 데이터 전달, 상호작용의 오류를 검사하는 테스트이다.
④ X — '수정이 다른 부분에 미친 영향을 검사한다'는 설명을 단위 테스트라고 한 부분이 틀렸다. 올바르게는 단위 테스트는 함수, 클래스, 모듈 등 개별 소프트웨어 단위를 독립적으로 검사하는 테스트이다.
⑤ X — '수정 후 기존 기능의 정상 작동 여부를 재검사한다'는 설명을 시스템 테스트라고 한 부분이 틀렸다. 올바르게는 시스템 테스트는 통합이 완료된 전체 시스템이 명세된 시스템 요구사항을 충족하는지 검사하는 테스트이다.

NoteBase

오답 포인트

문제에 '관련된 모듈까지'라는 표현이 있어 ③ 통합 테스트로 착각하기 쉽다. 그러나 구별의 핵심은 모듈을 결합해 인터페이스를 검사하는지가 아니라, 수정 후 기존 기능에 부작용이 생겼는지를 다시 검사하는지이다.

다시 나오면 이렇게 판단

문장에서 '수정·변경 후', '기존 기능', '다른 부분에 영향', '다시 검사'가 함께 나타나면 회귀 테스트로 판단한다. '모듈 간 연결·인터페이스'는 통합 테스트, '사용자 요구와 인수 기준'은 인수 테스트로 구별한다.

암기 포인트

변경 후 부작용 재검사는 회귀, 모듈 간 연결 검사는 통합이다.

NoteBase

8. 다음 IP 주소 중 클래스 B에 해당하는 것은?

① 100.200.150.25
② 126.255.150.25
③ 180.100.150.25
④ 192.168.150.25
⑤ 203.252.150.25

정답 3번

출제 포인트

클래스 기반 IPv4 주소 체계에서 첫 번째 옥텟의 범위로 클래스 A·B·C를 구분할 수 있는지를 묻는 문제이다.

핵심 개념

클래스 기반 주소 체계는 IP 주소의 첫 번째 옥텟으로 클래스를 판별한다.
클래스 A: 첫 번째 옥텟이 1~126이다.
클래스 B: 첫 번째 옥텟이 128~191이다.
클래스 C: 첫 번째 옥텟이 192~223이다.
따라서 첫 번째 옥텟이 180인 180.100.150.25가 클래스 B에 해당한다.

선지별 해설

① X — 100.200.150.25의 첫 번째 옥텟은 100이므로 클래스 A이다. '클래스 B'가 아니라 '클래스 A'로 고쳐야 한다.
② X — 126.255.150.25의 첫 번째 옥텟은 126이므로 클래스 A이다. '클래스 B'가 아니라 '클래스 A'로 고쳐야 한다.
③ O — 180.100.150.25의 첫 번째 옥텟은 180이고, 180은 128~191에 포함되므로 클래스 B이다.
④ X — 192.168.150.25의 첫 번째 옥텟은 192이므로 클래스 C이다. '클래스 B'가 아니라 '클래스 C'로 고쳐야 한다.
⑤ X — 203.252.150.25의 첫 번째 옥텟은 203이므로 클래스 C이다. '클래스 B'가 아니라 '클래스 C'로 고쳐야 한다.

오답 포인트

두 번째 이후의 옥텟까지 보고 판단하거나 경계값 126, 192를 잘못 분류하기 쉽다. 클래스는 첫 번째 옥텟만 확인하며, 126은 클래스 A이고 192부터는 클래스 C이다.

NoteBase

다시 나오면 이렇게 판단

IP 주소의 첫 번째 옥텟만 떼어 낸 뒤 1~126이면 A, 128~191이면 B, 192~223이면 C로 판정한다.

암기 포인트

첫 옥텟 기준으로 A는 1~126, B는 128~191, C는 192~223이다.

NoteBase

9. GPU(Graphics Processing Unit)는 기계학습, 인공신경망, 블록체인과 같은 응용프로그램을 실행시킬 때 자주 이용된다. 그 이유로 옳은 것은?

① GPU는 CPU보다 적은 수의 ALU(Arithmetic Logic Unit)를 포함하고 있기 때문이다.
② GPU는 CPU에 비해서 분기문을 효과적으로 수행할 수 있는 구조이기 때문이다.
③ GPU는 CPU에 비해서 전력을 적게 소모하기 때문이다.
④ GPU는 CPU에 비해서 대량의 간단한 연산을 병렬로 빠르게 처리하는 데 적합한 구조이기 때문이다.
⑤ GPU는 CPU에 비해서 하나의 칩에 들어가는 트랜지스터의 수가 적기 때문이다.

정답 4번

출제 포인트

GPU가 기계학습·인공신경망·블록체인처럼 같은 형태의 연산을 대량으로 수행하는 작업에 적합한 구조적 이유를 묻는 문제이다.

핵심 개념

GPU: 다수의 비교적 단순한 연산 유닛으로 같은 종류의 연산을 여러 데이터에 병렬 수행하여 처리량을 높이는 프로세서이다. 키워드는 대규모 병렬 처리이다.
CPU: 복잡한 제어, 다양한 명령, 순차 처리와 분기 처리를 빠르게 수행하도록 설계된 범용 프로세서이다. 키워드는 복잡한 제어와 낮은 지연 시간이다.
GPU는 단순 연산의 대량 병렬 처리에 강하고, CPU는 복잡한 흐름 제어와 분기가 많은 작업에 강하다.

NoteBase

선지별 해설

① X — 틀린 표현은 ‘CPU보다 적은 수의 ALU를 포함’한다는 부분이다. GPU는 대량 병렬 연산을 위해 일반적으로 CPU보다 훨씬 많은 연산 유닛을 배치한다. 올바르게는 ‘GPU는 다수의 ALU 또는 연산 코어를 포함하여 병렬 처리에 유리하다’이다.
② X — 틀린 표현은 ‘분기문을 효과적으로 수행’한다는 부분이다. GPU에서는 함께 실행되는 작업들이 서로 다른 분기 경로를 택하면 분기 발산이 발생하여 병렬 처리 효율이 낮아질 수 있다. 올바르게는 ‘복잡한 분기와 흐름 제어는 일반적으로 CPU가 더 효율적으로 처리한다’이다.
③ X — 틀린 표현은 ‘CPU에 비해서 전력을 적게 소모하기 때문’이라는 부분이다. GPU의 주된 활용 이유는 낮은 총전력 소비가 아니라 대량 병렬 연산의 높은 처리량이며, 장치에 따라 GPU의 총소비전력이 더 클 수도 있다. 올바르게는 ‘GPU는 병렬화 가능한 연산에서 높은 처리량과 우수한 연산당 효율을 낼 수 있기 때문에 활용된다’이다.
④ O — GPU는 다수의 비교적 단순한 연산 유닛으로 동일하거나 유사한 연산을 많은 데이터에 동시에 적용하는 데 적합하다. 행렬·벡터 연산과 해시 계산처럼 병렬화할 수 있는 대량 연산을 빠르게 처리할 수 있으므로 옳다.
⑤ X — 틀린 표현은 ‘하나의 칩에 들어가는 트랜지스터의 수가 적기 때문’이라는 부분이다. GPU의 강점은 트랜지스터 총수가 적다는 데 있지 않고 많은 연산 유닛에 자원을 배분하여 병렬 처리량을 높이는 구조에 있다. 올바르게는 ‘GPU는 많은 연산 유닛을 집적하고 연산 처리에 큰 비중을 두는 구조이기 때문에 대량 병렬 처리에 유리하다’이다.

오답 포인트

③의 ‘전력을 적게 소모한다’와 ⑤의 ‘트랜지스터 수가 적다’는 부수적인 하드웨어 수치를 GPU 활용의 본질적 이유처럼 제시한 함정이다. GPU의 핵심은 총전력이나 트랜지스터 총수가 아니라 다수의 연산 유닛을 이용한 높은 병렬 처리량이다. 또한 ②의 분기 처리는 GPU보다 CPU가 상대적으로 강한 영역이다.

다시 나오면 이렇게 판단

같은 계산을 많은 데이터에 반복하는 작업이면 GPU의 대규모 병렬 처리를 연결하고, 복잡한 분기·제어와 순차 실행이 핵심이면 CPU를 연결한다. GPU의 장점을 묻는 선지에서는 ALU·연산 코어의 수가 많고 단순 연산을 병렬로 처리한다는 표현을 우선 확인한다.

암기 포인트

GPU는 다수의 단순 연산을 병렬로, CPU는 복잡한 제어와 분기를 빠르게 처리한다.

NoteBase

10. 다음 프로그램의 A3 클래스에서 사용할 수 있는 객체 변수들로 옳은 것만을 모두 고르면?

```
public class A1 {
  public int x;
  private int y;
  protected int z;
    …
 }
public class A2 extends A1 {
  protected int a;
  private int b;
    …
 }
public class A3 extends A2 {
  private int q;
    …
 }
```

① x, q
② x, y, b, q
③ x, y, z, q
④ x, z, a, q
⑤ x, y, z, a, b, q

정답 4번

출제 포인트

상속 관계에서 public, protected, private 접근 지정자에 따라 A3 클래스가 직접 사용할 수 있는 인스턴스 변수를 구분하는 문제이다.

핵심 개념

public 멤버는 다른 클래스에서도 접근할 수 있으므로 A1의 x를 A3에서 사용할 수 있다.
protected 멤버는 같은 패키지의 클래스와 하위 클래스에서 접근할 수 있으므로 A1의 z와 A2의 a를 하위 클래스 A3에서 사용할 수 있다.
private 멤버는 선언된 클래스 내부에서만 접근할 수 있다. 따라서 A1의 y와 A2의 b는 A3에서 직접 사용할 수 없지만, A3 자신이 선언한 q는 A3 내부에서 사용할 수 있다.
결국 A3에서 사용할 수 있는 객체 변수는 x, z, a, q이다.

NoteBase

선지별 해설

① X — x와 q는 사용할 수 있지만, 상속받은 protected 변수 z와 a도 사용할 수 있으므로 목록이 불완전하다. 틀린 부분은 z와 a를 제외한 것이며, 올바른 목록은 x, z, a, q이다.
② X — A1의 private 변수 y와 A2의 private 변수 b는 각각 선언 클래스 밖인 A3에서 직접 사용할 수 없다. 또한 protected 변수 z와 a가 빠져 있다. 올바른 목록은 x, z, a, q이다.
③ X — A1의 private 변수 y는 A3에서 직접 사용할 수 없고, A2의 protected 변수 a가 빠져 있다. y를 제외하고 a를 포함하면 x, z, a, q이다.
④ O — A1의 public 변수 x, A1과 A2의 protected 변수 z와 a, A3이 직접 선언한 private 변수 q는 모두 A3 내부에서 사용할 수 있다.
⑤ X — y와 b는 각각 A1과 A2가 선언한 private 변수이므로 A3에서 직접 사용할 수 없다. y와 b를 제외한 x, z, a, q가 올바르다.

오답 포인트

private 멤버도 하위 클래스가 사용할 수 있다고 착각하거나, protected 멤버를 같은 클래스에서만 사용할 수 있다고 오해하기 쉽다. 이 문제에서는 y와 b는 제외하고, 상속 계층의 protected 변수 z와 a는 포함해야 한다.

다시 나오면 이렇게 판단

현재 클래스가 직접 선언한 멤버를 먼저 포함한 뒤, 상위 클래스 멤버는 public과 protected를 포함하고 private를 제외한다. 이 기준을 A3에 적용하면 q를 포함하고 x, z, a를 상속 계층에서 선택한다.

암기 포인트

하위 클래스에서는 상위 클래스의 public·protected는 접근 가능하고 private는 직접 접근 불가능하다. 자기 클래스의 private는 접근 가능하다.

NoteBase

11. 다음은 컴퓨터에서 데이터를 저장할 수 있는 기억장치 종류를 나열한 것이다. 이를 접근(access) 속도가 빠른 순서대로 바르게 나열한 것은?

A. 마그네틱(Magnetic) 디스크
B. SRAM(Static RAM)
C. DRAM(Dynamic RAM)
D. 플래시 메모리(Flash Memory)

① A-B-C-D
② B-C-D-A
③ B-D-C-A
④ C-B-A-D
⑤ D-C-A-B

정답 2번

출제 포인트

SRAM, DRAM, 플래시 메모리, 마그네틱 디스크의 일반적인 접근 속도 순서를 구분하는 문제이다.

핵심 개념

SRAM은 플립플롭 방식의 휘발성 반도체 메모리로, 재충전이 필요 없어 네 장치 중 접근 속도가 가장 빠르다.
DRAM은 커패시터에 데이터를 저장하고 주기적으로 재충전하는 휘발성 반도체 메모리로, SRAM보다 느리지만 플래시 메모리보다 빠르다.
플래시 메모리는 전원이 꺼져도 데이터가 유지되는 비휘발성 반도체 메모리로, 일반적으로 DRAM보다 느리고 마그네틱 디스크보다 빠르다.
마그네틱 디스크는 회전하는 원판과 기계식 헤드를 사용하는 보조기억장치이므로 탐색 및 회전 지연이 발생해 네 장치 중 가장 느리다.
따라서 빠른 순서는 B(SRAM) → C(DRAM) → D(플래시 메모리) → A(마그네틱 디스크)이다.

NoteBase

선지별 해설

① X — A-B-C-D는 기계식 마그네틱 디스크를 가장 빠르게 둔 점이 틀렸다. 올바른 순서는 B-C-D-A이다.
② O — B-C-D-A는 SRAM → DRAM → 플래시 메모리 → 마그네틱 디스크의 접근 속도 순서와 일치한다.
③ X — B-D-C-A는 플래시 메모리(D)를 DRAM(C)보다 빠르게 둔 점이 틀렸다. 올바른 순서는 B-C-D-A이다.
④ X — C-B-A-D는 DRAM(C)을 SRAM(B)보다 빠르게 두고, 마그네틱 디스크(A)를 플래시 메모리(D)보다 빠르게 둔 점이 틀렸다. 올바른 순서는 B-C-D-A이다.
⑤ X — D-C-A-B는 플래시 메모리(D)를 SRAM(B)과 DRAM(C)보다 빠르게 두고 SRAM(B)을 가장 느리게 둔 점이 틀렸다. 올바른 순서는 B-C-D-A이다.

오답 포인트

플래시 메모리가 비휘발성 반도체 메모리라는 이유만으로 DRAM보다 빠르다고 판단하기 쉽다. 그러나 이 문항의 핵심은 휘발성 여부가 아니라 접근 속도이며, 일반적으로 SRAM과 DRAM이 플래시 메모리보다 빠르다.

다시 나오면 이렇게 판단

먼저 기계식 동작이 있는 마그네틱 디스크를 가장 뒤에 둔다. 반도체 메모리 중에서는 SRAM을 가장 앞에 두고, 그다음 DRAM, 플래시 메모리 순으로 배열한다.

암기 포인트

빠른 순서: SRAM → DRAM → Flash → Disk, 즉 B-C-D-A이다.

NoteBase

12. 선점형(Preemptive) 스케줄링에 해당하지 않는 것은?

① MFQ(Multilevel Feedback Queue) 스케줄링
② RR(Round Robin) 스케줄링
③ MLQ(Multilevel Queue) 스케줄링
④ SRT(Shortest Remaining Time) 스케줄링
⑤ HRN(Highest Response ratio Next) 스케줄링

정답 5번

출제 포인트

이 문제는 CPU 스케줄링 기법을 선점형과 비선점형으로 구분할 수 있는지를 묻는다.

핵심 개념

선점형 스케줄링은 운영체제가 실행 중인 프로세스의 CPU를 강제로 회수하여 다른 프로세스에 배정할 수 있는 방식이다. 비선점형 스케줄링은 실행 중인 프로세스가 종료되거나 대기 상태로 전환되어 스스로 CPU를 내놓을 때까지 CPU를 빼앗지 않는 방식이다. MFQ·RR·MLQ·SRT는 선점이 가능한 기법이고, HRN은 응답률이 가장 높은 프로세스를 골라 실행하는 비선점형 기법이다.

선지별 해설

① O — MFQ(Multilevel Feedback Queue)는 프로세스가 여러 우선순위 큐 사이를 이동하며, 높은 우선순위 작업의 도착이나 시간 할당량 만료에 따라 실행 중인 작업을 교체할 수 있는 선점형 스케줄링이다.
② O — RR(Round Robin)은 정해진 시간 할당량이 끝나면 실행 중인 프로세스에서 CPU를 회수하여 다음 프로세스에 넘기는 대표적인 선점형 스케줄링이다.
③ O — MLQ(Multilevel Queue)는 프로세스를 여러 고정 큐로 나누고, 선점 우선순위 방식에서는 높은 우선순위 큐의 프로세스가 낮은 우선순위 큐의 실행을 선점할 수 있으므로 이 문항의 분류에서 선점형에 해당한다.
④ O — SRT(Shortest Remaining Time)는 남은 실행 시간이 더 짧은 프로세스가 도착하면 현재 프로세스를 중단하고 새 프로세스에 CPU를 배정하는 선점형 스케줄링이다.
⑤ X — HRN(Highest Response Ratio Next)을 '선점형 스케줄링'이라고 한 부분이 틀렸다. HRN은 대기 시간과 서비스 시간을 반영한 응답률이 가장 높은 프로세스를 선택한 뒤 그 실행을 선점하지 않는 비선점형 스케줄링이다. 올바르게는 'HRN은 비선점형 스케줄링이다.'이다.

오답 포인트

HRN은 대기 시간을 계속 반영하여 우선순위가 변하므로 선점형처럼 보일 수 있다. 그러나 우선순위 변화는 다음에 실행할 프로세스를 고를 때 사용될 뿐, 이미 실행 중인 프로세스의 CPU를 강제로 회수하지 않는다.

NoteBase

다시 나오면 이렇게 판단

실행 중인 프로세스에서 CPU를 강제로 회수하는 장치가 있는지를 확인한다. 시간 할당량을 쓰는 RR, 더 짧은 잔여 시간이 오면 교체하는 SRT, 우선순위 큐 간 선점이 가능한 MFQ·MLQ는 선점형으로 판단하고, 선택 후 완료까지 실행하는 HRN은 비선점형으로 판단한다.

암기 포인트

RR은 시간 만료, SRT는 더 짧은 잔여 시간, MFQ·MLQ는 우선순위 큐로 선점하고, HRN은 응답률로 선택하되 선점하지 않는다.

NoteBase

13. 페이지 프레임(page frame)의 수가 4이고 가상 페이지(virtual page)의 수가 8인 가상 메모리에서 선입선출(FIFO) 페이지 교체 정책이 사용된다. 처음에 4개의 페이지 프레임들이 비어 있다고 가정했을 때 페이지 참조 열이 0, 1, 7, 2, 3, 2, 7, 1, 0, 3이라면 페이지 부재(page fault) 횟수와 페이지 교체(page replacement) 횟수로 옳은 것은?

① 페이지 부재 횟수 = 6, 페이지 교체 횟수 = 2
② 페이지 부재 횟수 = 6, 페이지 교체 횟수 = 3
③ 페이지 부재 횟수 = 7, 페이지 교체 횟수 = 2
④ 페이지 부재 횟수 = 7, 페이지 교체 횟수 = 3
⑤ 페이지 부재 횟수 = 8, 페이지 교체 횟수 = 3

정답 1번

출제 포인트

이 문제는 FIFO 페이지 교체 정책에서 페이지 부재와 실제 페이지 교체를 구분하고, 참조열에 따라 메모리 상태를 추적하는 문제이다.

조건 정리

주어진 값: 페이지 프레임은 4개이고, 참조열은 0, 1, 7, 2, 3, 2, 7, 1, 0, 3이다.
구해야 할 값: 페이지 부재 횟수와 페이지 교체 횟수이다.
주의할 조건: 빈 프레임에 페이지를 처음 적재하는 것도 페이지 부재이지만, 내보낼 페이지가 없으므로 페이지 교체는 아니다. FIFO에서는 메모리에 가장 먼저 들어온 페이지를 가장 먼저 내보낸다.

사용할 원리

참조한 페이지가 프레임에 없으면 페이지 부재를 1회 센다. 빈 프레임이 있으면 그곳에 적재하고, 프레임이 모두 찬 뒤 부재가 발생하면 가장 오래전에 들어온 페이지를 교체한다. 이 원리를 쓰는 이유는 처음 4회의 적재와 이후의 실제 교체를 따로 세어야 하기 때문이다.

NoteBase

계산 과정

1. 0, 1, 7, 2는 모두 처음 참조되므로 부재가 4회 발생하며, 빈 프레임에 차례로 적재된다. 이때 프레임은 [0, 1, 7, 2]이고 교체는 0회이다.
2. 3은 프레임에 없고 프레임이 가득 찼으므로 부재가 발생한다. FIFO에 따라 가장 먼저 들어온 0을 3으로 교체한다. 누적 부재는 5회, 누적 교체는 1회이다.
3. 이어지는 2, 7, 1은 모두 프레임 안에 있으므로 적중이며, 부재와 교체 횟수는 늘지 않는다.
4. 0은 프레임에 없으므로 부재가 발생한다. 현재 가장 오래된 1을 0으로 교체한다. 누적 부재는 6회, 누적 교체는 2회이다.
5. 마지막 3은 프레임 안에 있으므로 적중이다. 따라서 페이지 부재 횟수는 4 + 2 = 6회이고 페이지 교체 횟수는 2회이다.

선지 판단

① O — 계산 결과가 페이지 부재 6회, 페이지 교체 2회이므로 옳다.
② X — 페이지 부재 6회는 맞지만 페이지 교체 3회가 틀리다. 올바르게는 페이지 교체 2회이다. 최초의 빈 프레임 적재를 교체로 세지 않는다.
③ X — 페이지 교체 2회는 맞지만 페이지 부재 7회가 틀리다. 올바르게는 페이지 부재 6회이다. 이미 적재된 2, 7, 1, 마지막 3의 참조는 적중이다.
④ X — 페이지 부재 7회와 페이지 교체 3회가 모두 틀리다. 올바르게는 페이지 부재 6회, 페이지 교체 2회이다.
⑤ X — 페이지 부재 8회와 페이지 교체 3회가 모두 틀리다. 올바르게는 페이지 부재 6회, 페이지 교체 2회이다. 적중한 페이지를 부재로 중복 계산하지 않는다.

오답 포인트

이 문제의 함정은 페이지 부재와 페이지 교체를 같은 횟수로 세는 데 있다. 처음 4개 페이지는 빈 프레임에 들어가므로 부재에는 포함되지만 교체에는 포함되지 않는다. 또한 FIFO의 순서는 최근 참조 시점이 아니라 프레임에 들어온 시점을 기준으로 한다.

다시 나오면 이렇게 판단

참조마다 프레임 존재 여부를 먼저 확인한다. 없으면 부재를 세고, 빈 프레임이 없을 때만 FIFO 큐의 맨 앞 페이지를 내보내며 교체를 추가로 센다. 적중한 페이지는 FIFO 순서를 바꾸지 않는다.

암기 포인트

빈 프레임 적재는 부재 O·교체 X이고, 가득 찬 뒤의 미적중은 부재 O·교체 O이다.

NoteBase

14. 키(Key)란 데이터베이스에서 조건에 만족하는 튜플을 찾거나 순서대로 정렬할 때 다른 튜플들과 구별할 수 있는 유일한 기준이 되는 속성(Attribute)이다. 그 중 릴레이션을 구성하는 모든 튜플에 대해 유일성은 만족하지만 최소성은 만족하지 못하는 키로 옳은 것은?

① 기본키(Primary Key)
② 대체키(Alternate Key)
③ 복합키(Composite Key)
④ 후보키(Candidate Key)
⑤ 슈퍼키(Super Key)

정답 5번

출제 포인트

이 문제는 키의 두 성질인 유일성과 최소성을 기준으로 슈퍼키와 후보키를 구별하는 문제이다.

핵심 개념

유일성은 키의 값으로 각 튜플을 서로 구별할 수 있는 성질이고, 최소성은 튜플을 유일하게 식별하는 데 불필요한 속성이 하나도 없는 성질이다.
슈퍼키는 유일성을 만족하는 하나 이상의 속성 집합이며 최소성은 요구하지 않는다. 후보키는 슈퍼키 가운데 최소성까지 만족하는 키이다. 기본키는 후보키 중 대표로 선택된 키이고, 대체키는 선택되지 않은 나머지 후보키이다. 복합키는 둘 이상의 속성으로 구성된 키를 뜻하므로 그 명칭만으로 최소성 충족 여부가 결정되지는 않는다.

선지별 해설

① X — 기본키는 후보키 중 하나를 대표 키로 선택한 것이므로 유일성과 최소성을 모두 만족한다. 틀린 표현은 '기본키'이며, 유일성만 만족하고 최소성은 만족하지 않을 수 있는 키는 '슈퍼키'이다.
② X — 대체키는 기본키로 선택되지 않은 후보키이므로 유일성과 최소성을 모두 만족한다. 틀린 표현은 '대체키'이며, 올바른 개념은 '슈퍼키'이다.
③ X — 복합키는 둘 이상의 속성을 결합해 만든 키라는 구성 형태를 나타낸다. 불필요한 속성이 없으면 후보키가 될 수 있고, 불필요한 속성을 포함하면 최소성을 만족하지 않는 슈퍼키가 될 수 있으므로 문제의 성질을 항상 나타내는 명칭은 아니다. 틀린 표현은 '복합키'이며, 유일성만을 필수로 하는 키라는 뜻으로는 '슈퍼키'가 맞다.
④ X — 후보키는 튜플을 유일하게 식별하는 슈퍼키 중 최소성을 만족하는 키이다. 틀린 표현은 '후보키'이며, 최소성이 요구되지 않는 키는 '슈퍼키'이다.
⑤ O — 슈퍼키는 릴레이션의 모든 튜플을 유일하게 식별할 수 있는 속성 집합이다. 식별에 불필요한 속성을 더 포함해도 슈퍼키이므로 최소성을 만족하지 않을 수 있다.

NoteBase

오답 포인트

‘유일하게 식별한다’는 말만 보고 후보키를 고르는 것이 함정이다. 후보키도 유일성을 만족하지만 최소성까지 반드시 만족한다. 반면 슈퍼키는 유일성만 필수이고 불필요한 속성을 포함할 수 있다. 또한 복합키는 속성 개수에 따른 명칭이므로 최소성 여부와 직접 대응하지 않는다.

다시 나오면 이렇게 판단

유일성만 제시되면 슈퍼키를, 유일성과 최소성이 함께 제시되면 후보키를 판단한다. 후보키 중 실제 대표로 선택된 것은 기본키이고, 선택되지 않은 것은 대체키이다.

암기 포인트

슈퍼키는 유일성, 후보키는 유일성+최소성이다.

NoteBase

15. 리스트의 양쪽 끝에서 노드의 삽입과 삭제가 모두 가능한 선형 리스트로서, 리스트의 양쪽 끝 노드를 각각 가리키는 두 개의 포인터를 갖는 자료구조로 옳은 것은?

① 스택(Stack)
② 큐(Queue)
③ 히프(Heap)
④ 데크(Deque)
⑤ 이진 트리(Binary Tree)

정답 4번

출제 포인트

리스트의 양쪽 끝에서 노드를 삽입하고 삭제할 수 있는 선형 자료구조를 구별하는 문제이다.

핵심 개념

데크(Deque)는 앞쪽과 뒤쪽 양 끝에서 삽입과 삭제를 모두 허용하는 선형 자료구조이다. 스택은 한쪽 끝에서만 삽입과 삭제가 이루어지고, 큐는 한쪽 끝에서 삽입하고 반대쪽 끝에서 삭제한다.

선지별 해설

① X — 스택(Stack)은 삽입과 삭제가 모두 한쪽 끝인 top에서 이루어지는 후입선출(LIFO) 구조이다. '양쪽 끝에서 삽입과 삭제가 모두 가능하다'는 표현이 틀렸으며, 올바르게는 '한쪽 끝에서만 삽입과 삭제가 가능하다'이다.
② X — 큐(Queue)는 rear에서 삽입하고 front에서 삭제하는 선입선출(FIFO) 구조이다. '양쪽 끝에서 삽입과 삭제가 모두 가능하다'는 표현이 틀렸으며, 올바르게는 '한쪽 끝에서는 삽입하고 반대쪽 끝에서는 삭제한다'이다.
③ X — 히프(Heap)는 부모와 자식 사이의 대소 관계를 유지하는 완전 이진 트리 기반의 자료구조로, 우선순위가 가장 높은 원소를 빠르게 찾는 데 사용된다. '양쪽 끝을 두 포인터로 가리키는 선형 리스트'라는 표현이 틀렸으며, 올바르게는 '우선순위 관계를 유지하는 완전 이진 트리 기반 구조'이다.
④ O — 데크(Deque)는 double-ended queue의 약자로, front와 rear 양쪽 끝에서 삽입과 삭제를 모두 수행할 수 있는 선형 자료구조이다.
⑤ X — 이진 트리(Binary Tree)는 각 노드가 최대 두 개의 자식 노드를 갖는 비선형 계층 자료구조이다. '리스트의 양쪽 끝에서 삽입과 삭제가 모두 가능한 선형 구조'라는 표현이 틀렸으며, 올바르게는 '각 노드가 최대 두 자식을 갖는 비선형 트리 구조'이다.

NoteBase

오답 포인트

스택, 큐, 데크는 모두 삽입과 삭제를 수행하지만 연산을 허용하는 끝이 다르다. 특히 큐도 front와 rear를 사용하므로 양쪽 끝에서 모든 연산이 가능한 것으로 착각하기 쉽지만, 일반 큐는 rear 삽입과 front 삭제로 역할이 정해져 있다.

다시 나오면 이렇게 판단

삽입과 삭제가 허용되는 위치를 기준으로 판단한다. 한쪽 끝에서 두 연산을 모두 하면 스택, 서로 다른 끝에서 삽입과 삭제를 나누어 하면 큐, 양쪽 끝에서 두 연산을 모두 하면 데크이다.

암기 포인트

스택은 한쪽 끝, 큐는 양 끝의 역할 분담, 데크는 양쪽 끝에서 삽입·삭제가 모두 가능하다.

NoteBase

16. 소프트웨어 공학의 개발 과정에 대한 설명으로 옳지 않은 것은?

① 계획-목표를 세우고 달성하기 위하여 체계적인 진행 상황 관리를 할 수 있도록 행동 방안을 마련한다.

② 요구사항 문서화-사용자의 요구사항을 명세서로 작성하는 과정으로 프로젝트에 관계된 모든 사람이 이해하기 쉽게 작성해야 한다.

③ 설계-요구사항을 반영하여 설계서를 작성하는 과정으로 변화에 쉽게 적응할 수 있고 유지보수가 용이하도록 작성해야 한다.

④ 구현-프로그램을 제작 및 구현하는 단계로 보통 본 과정에서 완벽히 제작하여 시험 단계를 생략한다.

⑤ 시험과 유지 및 보수 - 프로그램 구현 이후 품질 보증을 위하여 제품의 오류를 발견하고 수정, 배포 이후에는 사용 시에 생기는 문제에 관하여 관리하여야 한다.

정답 4번

출제 포인트

소프트웨어 개발 과정의 계획·요구사항 분석·설계·구현·시험·유지보수 단계가 각각 수행하는 역할을 구별하는 문제이다.

핵심 개념

소프트웨어 개발 과정은 필요한 소프트웨어를 체계적으로 만들고 품질을 확인한 뒤 운영 중의 변경과 문제에 대응하는 일련의 활동이다.
계획은 목표와 진행 방안을 정하고, 요구사항 분석은 사용자의 요구를 명확히 문서화하며, 설계는 구현할 구조와 방법을 정한다. 구현은 설계를 프로그램으로 옮기는 단계이고, 시험은 구현 결과가 요구사항에 맞게 동작하는지와 오류가 있는지를 확인하는 단계이다. 유지보수는 배포 후 발생하는 오류 수정, 환경 변화 대응, 기능 개선 등을 수행한다.

NoteBase

선지별 해설

① O — 계획 단계에서는 개발 목표를 세우고 이를 달성하기 위한 일정, 자원, 업무 및 진행 관리 방안을 마련한다.
② O — 요구사항 문서화 단계에서는 사용자의 요구를 요구사항 명세서로 명확하게 작성하며, 개발자와 사용자 등 프로젝트 관계자가 공통으로 이해할 수 있어야 한다.
③ O — 설계 단계에서는 요구사항을 반영해 시스템의 구조와 구현 방식을 설계서로 표현한다. 변경에 유연하게 대응하고 유지보수가 쉽도록 모듈화와 구조화를 고려하는 것이 바람직하다.
④ X — 틀린 표현은 '완벽히 제작하여 시험 단계를 생략한다'이다. 구현 단계는 설계 내용을 실제 프로그램으로 작성하는 단계이며, 구현만으로 오류가 없다고 보장할 수 없으므로 시험 단계를 생략해서는 안 된다. 올바르게는 구현 후 별도의 시험을 통해 요구사항 충족 여부와 결함을 확인한다.
⑤ O — 시험에서는 구현된 제품의 오류를 찾아 수정하고 요구사항 충족 여부를 확인해 품질을 보증한다. 유지보수에서는 배포 후 사용 중 발견되는 결함, 환경 변화, 기능 개선 요구 등을 관리한다.

오답 포인트

④는 구현을 충실히 하면 시험이 필요 없다는 표현으로 두 단계를 혼동하게 만든다. 프로그램은 구현 과정에서 결함이 생길 수 있으므로 구현의 완성도를 높이는 것과 시험을 생략하는 것은 전혀 다른 문제이다.

다시 나오면 이렇게 판단

각 문장의 핵심 동사를 단계와 연결한다. 목표·일정·관리는 계획, 요구의 명확화는 요구사항 분석, 구조와 방법의 결정은 설계, 코딩은 구현, 결함과 요구 충족 확인은 시험, 배포 후 수정과 개선은 유지보수에 해당한다. 시험을 생략하거나 구현만으로 무결성을 보장한다는 문장은 틀린 설명으로 판단한다.

암기 포인트

계획은 방향, 요구사항은 무엇을, 설계는 어떻게, 구현은 코딩, 시험은 검증, 유지보수는 배포 후 대응이다.

NoteBase

17. 5G 네트워크의 주요 기술 중 하나로, 데이터 속도를 높이고 대용량 MIMO(Multiple Input Multiple Output) 안테나를 사용해 대역폭을 향상시켜 실제 필요한 장소에 집중적으로 무선 신호를 전송하는 기술은?

① 광대역 부호 다중 분할 접속(Wideband Code Division Multiple Access)
② 펨토셀(Femto Cell)
③ 주파수 집성기술(Carrier Aggregation)
④ 고속 하향 패킷 접속(High Speed Downlink Packet Access)
⑤ 빔 포밍(Beam Forming)

정답 5번

출제 포인트

대용량 MIMO 안테나를 이용하여 전파의 방향을 조절하고, 필요한 장소나 사용자에게 무선 신호를 집중하는 5G 기술을 구별하는 문제이다.

핵심 개념

빔 포밍(Beam Forming)은 여러 안테나에서 내보내는 신호의 위상과 세기를 조절하여 특정 방향에서 신호가 강해지도록 빔을 형성하는 기술이다. 필요한 사용자나 장소에 신호를 집중하므로 수신 품질과 데이터 전송 효율을 높일 수 있다.
주파수 집성은 여러 주파수 대역을 묶어 전송 대역폭을 넓히는 기술이고, 빔 포밍은 안테나 신호의 방향을 특정 공간으로 집중하는 기술이라는 점에서 구별된다.

NoteBase

선지별 해설

① X
광대역 부호 다중 분할 접속(WCDMA)은 여러 사용자를 서로 다른 확산 부호로 구분하여 접속하게 하는 이동통신 다중 접속 방식이다. 틀린 표현은 필요한 장소에 무선 신호를 집중하는 기술이라는 연결이며, 올바르게는 WCDMA는 광대역 확산 부호를 이용하는 다중 접속 방식이다.

② X
펨토셀(Femto Cell)은 가정이나 소규모 실내에 설치하는 저전력·소형 기지국으로, 좁은 지역의 통신 음영을 줄이고 용량을 보완한다. 틀린 표현은 대용량 MIMO 안테나로 신호 방향을 집중하는 기술이라는 연결이며, 올바르게는 펨토셀은 소형 기지국을 설치하여 국소 지역의 통신 환경을 개선하는 기술이다.

③ X
주파수 집성기술(Carrier Aggregation)은 여러 개의 반송파를 하나처럼 결합하여 더 넓은 전송 대역폭을 확보하는 기술이다. 틀린 표현은 실제 필요한 장소에 무선 신호를 집중한다는 부분이며, 올바르게는 주파수 집성은 복수의 주파수 대역을 묶어 데이터 전송 속도를 높이는 기술이다.

④ X
고속 하향 패킷 접속(HSDPA)은 WCDMA 계열에서 기지국에서 단말기로 향하는 하향 패킷 데이터 전송 속도를 높이는 기술이다. 틀린 표현은 MIMO 안테나로 특정 장소에 신호를 집중하는 기술이라는 연결이며, 올바르게는 HSDPA는 하향 링크의 패킷 전송 성능을 개선하는 기술이다.

⑤ O
빔 포밍(Beam Forming)은 다수의 안테나 신호를 제어하여 특정 사용자나 장소 방향으로 전파 에너지를 집중한다. 문제의 대용량 MIMO 안테나 사용과 필요한 장소로의 집중 전송이라는 특징에 정확히 해당한다.

오답 포인트

③ 주파수 집성도 대역폭과 데이터 속도를 높이므로 정답처럼 보일 수 있다. 그러나 주파수 집성은 여러 반송파를 결합하는 주파수 영역의 기술이고, 문제의 핵심 표현인 필요한 장소에 신호를 집중하는 것은 공간 방향을 제어하는 빔 포밍이다.

다시 나오면 이렇게 판단

여러 주파수 대역을 묶는다는 표현이면 주파수 집성, 소형 기지국을 설치한다는 표현이면 펨토셀, 안테나 신호를 특정 사용자나 방향에 집중한다는 표현이면 빔 포밍으로 판단한다.

암기 포인트

반송파를 묶으면 주파수 집성, 전파의 방향을 모으면 빔 포밍이다.

NoteBase

18. 파일 전송 프로토콜(FTP)에 대한 설명으로 옳은 것만을 모두 고르면?

ㄱ. 응용 계층 프로토콜이다.
ㄴ. FTP 프로토콜에서 제어 연결은 20번 포트, 데이터 연결은 21번 포트를 사용한다.
ㄷ. FTP 연결은 능동(Active) 전송 모드와 수동(Passive) 전송 모드로 구분한다.

① ㄱ
② ㄱ, ㄴ
③ ㄱ, ㄷ
④ ㄴ, ㄷ
⑤ ㄱ, ㄴ, ㄷ

정답 3번

출제 포인트

FTP의 계층, 제어·데이터 연결에 사용하는 포트 번호, 능동·수동 전송 모드의 구분을 묻는 문제이다.

핵심 개념

FTP는 TCP/IP 응용 계층에서 파일을 전송하는 프로토콜이다. 기본 포트는 제어 연결이 21번이고, 능동 모드의 데이터 연결이 20번이다.
능동(Active) 모드는 서버가 클라이언트에 데이터 연결을 설정하고, 수동(Passive) 모드는 클라이언트가 서버가 알려 준 포트로 데이터 연결을 설정한다.
따라서 ㄱ은 옳고, ㄴ은 제어 포트와 데이터 포트를 서로 바꾸어 서술했으므로 틀리며, ㄷ은 옳다.

선지별 해설

① X — ㄱ만 제시하여 옳은 ㄷ을 누락했다. 올바른 조합은 ㄱ, ㄷ이다.
② X — ㄴ은 제어 연결 20번, 데이터 연결 21번이라고 포트를 뒤바꾸어 서술했다. 올바르게는 제어 연결이 21번이고 능동 모드의 데이터 연결이 20번이며, 조합은 ㄱ, ㄷ이다.
③ O — ㄱ의 응용 계층 설명과 ㄷ의 능동·수동 모드 구분이 모두 옳고, 틀린 ㄴ은 제외한 조합이다.
④ X — ㄴ을 포함하고 옳은 ㄱ을 누락했다. 올바른 조합은 ㄱ, ㄷ이다.
⑤ X — 포트 번호를 뒤바꾼 ㄴ까지 포함했다. 올바른 조합은 ㄱ, ㄷ이다.

NoteBase

오답 포인트

ㄴ의 20번과 21번을 서로 바꾼 표현이 핵심 함정이다. FTP에서는 제어 연결이 21번이고, 능동 모드의 데이터 연결이 20번이다.

다시 나오면 이렇게 판단

FTP 문항에서는 먼저 응용 계층 프로토콜인지 확인한 뒤, 제어 21번·능동 데이터 20번을 대조하고, 능동과 수동은 데이터 연결을 시작하는 쪽으로 구분한다.

암기 포인트

FTP는 응용 계층, 제어는 21번, 능동 데이터는 20번이며 전송 모드는 Active와 Passive로 나뉜다.

NoteBase

19. 블록체인 기술로서, 장애가 있더라도 전체의 3분의 1을 넘지 않는다면 시스템이 정상 작동하도록 하는 기법은?

① 지분증명(Proof of Stake)
② 위임된 지분증명(Delegated Proof of Stake)
③ 작업증명(Proof of Work)
④ 비잔틴 장애 허용(Byzantine Fault Tolerance)
⑤ 샤딩(Sharding)

정답 4번

출제 포인트

이 문제는 전체 노드 중 악의적이거나 오작동하는 노드가 3분의 1 미만일 때도 합의를 유지하는 장애 허용 기법을 구별하는 문제이다.

핵심 개념

비잔틴 장애 허용(BFT)은 일부 노드가 임의의 값이나 거짓 정보를 보내더라도 정상 노드들이 동일한 결과에 합의하도록 하는 기법이다. 고전적인 BFT 합의에서는 장애 노드 수를 f, 전체 노드 수를 n이라 할 때 $n \geq 3f + 1$이 필요하므로 장애 노드가 전체의 3분의 1 미만이면 합의를 유지할 수 있다.
지분증명·위임된 지분증명·작업증명은 블록 생성자와 합의 참여 방식을 정하는 합의 방식이고, 샤딩은 처리 대상을 나누어 확장성을 높이는 기술이다.

NoteBase

선지별 해설

① X — 지분증명(Proof of Stake)은 보유하거나 예치한 지분을 기준으로 블록 검증자를 선정하는 합의 방식이다. 틀린 부분은 '장애 노드가 3분의 1 미만이면 정상 작동하게 하는 기법'으로 본 점이며, 올바르게는 '지분을 기준으로 검증 참여 권한을 정하는 방식'이다.
② X — 위임된 지분증명(Delegated Proof of Stake)은 지분 보유자의 투표로 대표 검증자를 선출하여 합의하는 방식이다. 틀린 부분은 이를 3분의 1 장애 허용 조건 자체로 본 점이며, 올바르게는 '선출된 대표 노드가 블록 생성과 검증을 담당하는 방식'이다.
③ X — 작업증명(Proof of Work)은 계산 문제를 먼저 해결한 참여자가 블록을 제안하도록 하는 합의 방식이다. 틀린 부분은 이를 3분의 1 장애 허용 기법으로 본 점이며, 올바르게는 '연산 능력과 작업량을 바탕으로 블록 생성 권한을 경쟁하는 방식'이다.
④ O — 비잔틴 장애 허용(Byzantine Fault Tolerance)은 일부 노드가 고장 나거나 악의적으로 행동해도 정상 노드들이 합의하도록 한다. 고전적인 BFT 조건인 $n \geq 3f + 1$에 따라 장애 노드가 전체의 3분의 1 미만이면 정상 합의가 가능하다.
⑤ X — 샤딩(Sharding)은 데이터와 처리 작업을 여러 샤드로 분할하여 병렬 처리함으로써 확장성을 높이는 기술이다. 틀린 부분은 이를 3분의 1 장애 허용 기법으로 본 점이며, 올바르게는 '네트워크의 처리 부담을 여러 그룹으로 나누는 확장성 기술'이다.

오답 포인트

①~③도 블록체인의 합의와 관련된 기술이어서 혼동하기 쉽다. 그러나 '장애가 있더라도'와 '3분의 1 미만'이라는 표현은 검증자 선정 방식이 아니라 비잔틴 장애 허용의 정족수 조건을 가리킨다. ⑤는 합의 방식이 아니라 확장성 개선 기술이다.

다시 나오면 이렇게 판단

문제에 '악의적 노드', '임의의 장애', '3분의 1 미만', '합의 유지'가 함께 제시되면 BFT를 우선 판단한다. '지분'은 PoS, '대표자 투표'는 DPoS, '연산 경쟁'은 PoW, '데이터 분할'은 샤딩으로 구별한다.

암기 포인트

BFT는 3분의 1 미만의 비잔틴 장애를 견디며, 핵심 조건은 $n \geq 3f + 1$이다.

NoteBase

20. 시큐어 코딩(Secure Coding) 기법으로 옳지 않은 것은?

① 화이트리스트 방식으로 허용된 확장자만 업로드를 허용한다.
② 입력 화면 폼(FORM) 작성시 POST 방식보다 GET 방식을 사용한다.
③ 사용자로부터 입력받은 스크립트 관련 문자열을 필터링하여 변환한다.
④ 인자화된 질의문(parameterized query)을 사용한다.
⑤ 외부의 부적절한 입력이 명령어로 사용될 가능성을 배제해야 한다.

정답 2번

출제 포인트

이 문제는 파일 업로드, 웹 입력, 데이터베이스 질의, 운영체제 명령 실행에서 입력값을 안전하게 처리하는 시큐어 코딩 기법을 묻는다.

핵심 개념

시큐어 코딩은 외부 입력을 신뢰하지 않고 허용 목록 검증, 안전한 변환, 인자화된 질의 등으로 보안 취약점이 생기지 않게 구현하는 방법이다.
파일 업로드는 허용된 형식만 통과시키고, XSS는 입력 검증과 출력 위치에 맞는 인코딩으로 대응하며, SQL 삽입은 인자화된 질의로 방지한다. 운영체제 명령에 외부 입력을 직접 결합하지 않는 것도 핵심 원칙이다.

선지별 해설

① O — 화이트리스트 방식으로 업무상 필요한 확장자만 허용하면 위험한 실행 파일이나 스크립트 파일의 업로드 가능성을 줄일 수 있다.
② X — 틀린 표현은 ‘POST 방식보다 GET 방식을 사용한다’이다. GET은 입력값이 URL의 쿼리 문자열에 노출되어 브라우저 기록, 로그, 참조 URL 등에 남을 수 있으므로 입력 폼에는 GET보다 POST 방식을 사용하는 것이 안전하다. 다만 POST 자체가 암호화를 뜻하지 않으므로 중요한 정보는 HTTPS로 전송해야 한다.
③ O — 사용자 입력의 스크립트 관련 문자열을 검증·필터링하고 안전한 형태로 변환하면 악성 스크립트 실행을 막는 데 도움이 된다. 출력 위치에 맞는 인코딩을 함께 적용하는 것이 핵심이다.
④ O — 인자화된 질의는 SQL 명령문과 사용자 입력값을 분리하여 입력값이 SQL 구문으로 해석되는 것을 막으므로 SQL 삽입 공격 방지에 효과적이다.
⑤ O — 외부 입력이 운영체제 명령어나 명령 인자로 직접 사용되면 명령어 삽입 취약점이 발생할 수 있으므로, 외부 입력의 명령어 사용 가능성을 배제하고 필요한 경우 허용 목록으로 엄격히 검증해야 한다.

NoteBase

오답 포인트

GET과 POST를 단순한 전송 방식의 차이로만 보는 것이 함정이다. GET은 입력값을 URL에 표시하고 남길 수 있으므로, 입력 폼의 민감한 값을 다루는 방식으로는 POST보다 부적절하다. 또한 POST를 사용해도 HTTPS가 없으면 전송 내용이 자동으로 보호되는 것은 아니다.

다시 나오면 이렇게 판단

외부 입력이 파일, HTML, SQL, 운영체제 명령의 일부로 해석될 수 있는지를 먼저 확인한다. 해석될 가능성이 있으면 허용 목록 검증, 문맥별 인코딩, 인자화된 질의, 명령어와 데이터의 분리가 안전한 대응이다. 입력값을 URL에 노출하는 GET을 POST보다 우선하라는 설명은 부적절하다.

암기 포인트

업로드는 화이트리스트, XSS는 검증·인코딩, SQL은 인자화, 명령어는 외부 입력 분리, 입력 폼은 GET보다 POST이다.

NoteBase