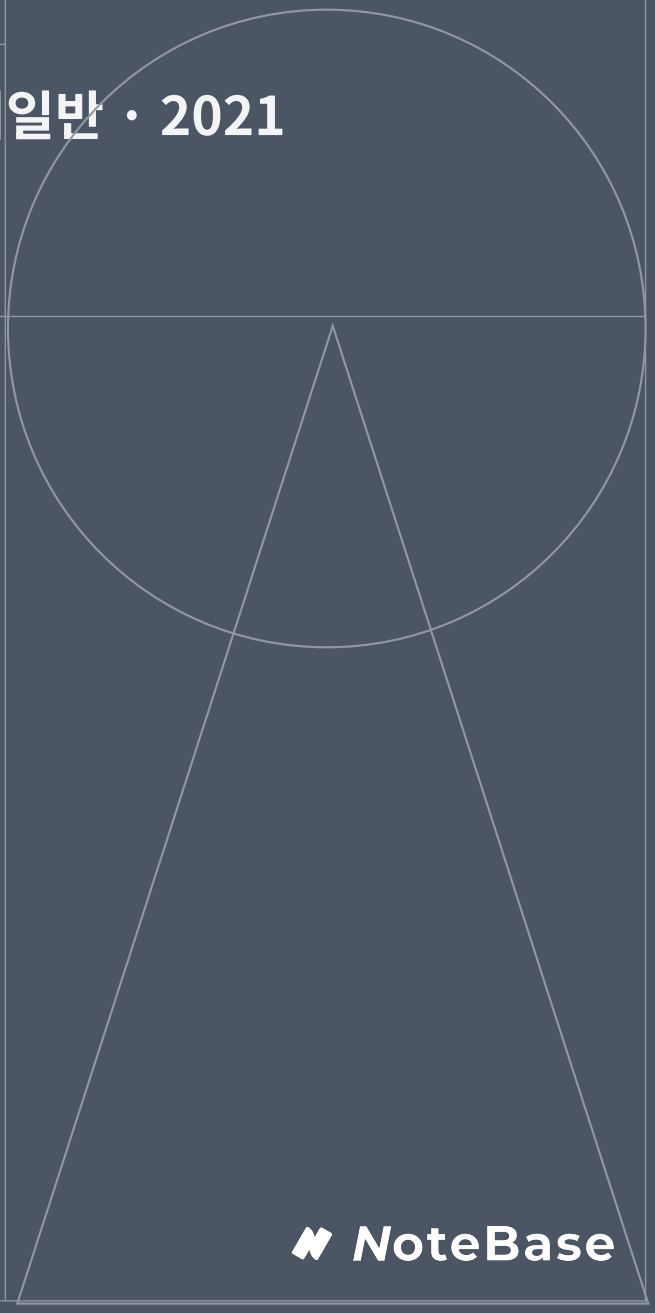
기출OX 해설집

공무원/계리직/9급 · 컴퓨터일반 · 2021



NoteBase

기출OX 해설집

공무원/계리직/9급 · 컴퓨터일반 · 2021

공무원/계리직/9급 · 컴퓨터일반 · 2021

문 1. 관계데이터베이스 관련 다음 설명에서 ㉠~㉣에 들어갈 용어를 바르게 짝지은 것은?

> (㉠) 무결성 제약이란 각 릴레이션(relation)에 속한 각 애트리뷰트(attribute)가 해당 (㉡)을 만족하면서 (㉢)할 수 없는 (㉣) 값을 가져서는 안 된다는 것을 말한다.

	㉠	㉡	㉢	㉣
①	참조	고립성	변경	외래키
②	개체	고립성	참조	기본키
③	참조	도메인	참조	외래키
④	개체	도메인	변경	기본키

정답 3번

출제 포인트

참조 무결성 제약에서 도메인, 참조, 외래키가 어떤 관계를 이루는지 묻는 문제이다.

핵심 개념

참조 무결성: 외래키 값은 참조되는 릴레이션의 기본키 값과 일치하거나 널 값이어야 한다는 제약이다. 따라서 각 속성은 자신의 도메인에 속하는 값을 가져야 하고, 외래키에는 참조할 수 없는 값이 들어가서는 안 된다.
개체 무결성은 기본키를 구성하는 속성이 널 값을 가질 수 없다는 제약이고, 고립성은 트랜잭션이 서로 간섭하지 않고 실행되는 성질이다.

NoteBase

선지별 해설

① X
'고립성'과 '변경'이 틀린 표현이다. 고립성은 트랜잭션의 성질이므로 속성값의 범위를 정하는 말이 아니다. 올바르게는 '참조-도메인-참조-외래키'이다.

② X
'개체-고립성-참조-기본키'가 틀린 조합이다. 개체 무결성은 기본키의 널 값 금지를 다루며, 이 문장의 핵심은 참조할 수 없는 외래키 값을 금지하는 참조 무결성이다. 올바르게는 '참조-도메인-참조-외래키'이다.

③ O
각 속성은 해당 도메인을 만족해야 하며, 참조 무결성에 따라 참조할 수 없는 외래키 값을 가져서는 안 된다. 따라서 '참조-도메인-참조-외래키'가 모두 알맞다.

④ X
'개체', '변경', '기본키'가 틀린 표현이다. 기본키의 널 값 금지가 아니라 외래키의 참조 가능 여부를 설명하는 문장이므로, 올바르게는 '참조-도메인-참조-외래키'이다.

오답 포인트

'기본키'라는 말만 보고 개체 무결성을 고르기 쉽지만, 문장의 결정적 표현은 '참조할 수 없는 값'이다. 참조 대상과의 일치 여부를 검사하는 키는 기본키가 아니라 외래키이다.

다시 나오면 이렇게 판단

기본키의 널 값 금지는 개체 무결성, 외래키가 참조 대상에 존재하는지의 검사는 참조 무결성으로 판단한다. 속성이 가질 수 있는 값의 범위를 묻는 말은 도메인이다.

암기 포인트

개체 무결성은 기본키, 참조 무결성은 외래키, 값의 허용 범위는 도메인이다.

NoteBase

문 2. 다음 워크시트에서 [D1] 셀에 =A1+$B2를 입력한 후 [D1] 셀을 복사하여 [D5] 셀에 붙여넣기 했을 때 [D5] 셀에 표시될 수 있는 결과로 옳은 것은?

	A	B	C	D
1	1	2	3	
2	2	4	6	
3	3	6	9	
4	4	8	12	
5	5	10	15	
6				

① 1 ② 7 ③ 9 ④ 15

정답 1번

출제 포인트

이 문제는 수식을 다른 셀에 복사할 때 절대 참조와 혼합 참조가 어떻게 바뀌는지 판단하는 문제이다.

조건 정리

주어진 수식: D1 셀의 A1 + $B2
복사 위치: D1에서 D5로 4행 아래 이동
구해야 할 값: 복사된 수식이 참조하는 셀의 합
주의할 조건: A1은 열과 행이 모두 고정되고, $B2는 B열만 고정되며 행은 상대적으로 이동한다.

사용할 공식

복사된 수식은 A1 + $B6이다.

이 원리를 쓰는 이유: D1에서 D5로 4행 내려가므로 상대 행 번호 2는 6으로 바뀌지만, 달러 기호가 붙은 열 B와 절대 참조 A1은 변하지 않기 때문이다.

계산 과정

1. A1은 절대 참조이므로 복사 후에도 A1을 가리킨다. A1의 값은 1이다.
2. $B2는 행이 상대 참조이므로 4행 아래로 복사하면 $B6이 된다. B6은 빈 셀이므로 덧셈에서 0으로 계산된다.
3. 따라서 1 + 0 = 1이다.

NoteBase

선지 판단

① O — 복사된 수식은 A1 + $B6이고, A1은 1이며 B6은 빈 셀이므로 결과는 1이다.
② X — 7은 계산 결과와 다르다. 틀린 값 7이 아니라 올바른 결과는 1이다.
③ X — 9는 계산 결과와 다르다. 틀린 값 9가 아니라 올바른 결과는 1이다.
④ X — 15는 계산 결과와 다르다. 틀린 값 15가 아니라 올바른 결과는 1이다.

오답 포인트

$B2에서 달러 기호는 열 B만 고정한다. 행 2까지 고정된 것으로 오해하거나, D5와 같은 행에 있는 B5의 값 10 또는 C5의 값 15를 그대로 선택하면 틀린다. 실제 참조 셀은 B6이며 빈 셀은 이 덧셈에서 0으로 계산된다.

다시 나오면 이렇게 판단

먼저 원래 셀과 붙여넣을 셀 사이의 행·열 이동량을 구한 뒤, 달러 기호가 없는 행이나 열에만 그 이동량을 반영한다. 마지막으로 바뀐 참조 셀의 실제 값을 대입한다.

암기 포인트

달러 기호가 붙은 열·행은 고정되고, 붙지 않은 열·행만 복사 방향과 거리만큼 이동한다.

NoteBase

문 3. 관계데이터베이스의 인덱스(index)에 대한 설명으로 옳은 것의 총 개수는?

ㄱ. 기본키의 경우, 자동으로 인덱스가 생성되며 인덱스 구축 시 두 개 이상의 칼럼(column)을 결합하여 인덱스를 생성할 수 있다.
ㄴ. SQL 명령문의 검색 결과는 인덱스 사용 여부와 관계없이 동일하며 인덱스는 검색 속도에 영향을 미친다.
ㄷ. 데이터베이스의 전체적인 성능을 향상시키기 위해서는 테이블의 모든 칼럼(column)에 대하여 인덱스를 생성해야 한다.
ㄹ. 인덱스는 칼럼(column)에 대하여 생성되며 테이블 내의 데이터를 순차적으로 접근하여 검색 결과를 제공한다.

① 1개 ② 2개 ③ 3개 ④ 4개

정답 2번

출제 포인트

관계데이터베이스 인덱스의 생성 방식, 검색 결과와 속도에 미치는 영향, 적절한 설계 원칙을 구분하여 옳은 보기의 개수를 판단하는 문제이다.

핵심 개념

인덱스는 테이블의 특정 열 값을 기준으로 원하는 행을 빠르게 찾도록 돕는 별도의 탐색 구조이다. 기본키에는 일반적으로 인덱스가 자동 생성되며, 둘 이상의 열을 묶은 복합 인덱스도 만들 수 있다. 인덱스는 SQL문의 논리적인 검색 결과를 바꾸지 않고 검색 성능을 높일 수 있지만, 저장 공간과 데이터 변경 비용이 추가되므로 필요한 열에 선택적으로 생성한다.

NoteBase

보기 판단

ㄱ O — 기본키를 지정하면 일반적으로 이를 지원하는 인덱스가 자동 생성되며, 둘 이상의 열을 결합한 복합 인덱스도 생성할 수 있다.
ㄴ O — 같은 조건의 SQL문은 인덱스 사용 여부와 관계없이 같은 검색 결과를 반환하며, 인덱스는 주로 검색 속도와 실행 효율에 영향을 준다.
ㄷ X — 틀린 표현은 '모든 칼럼에 대하여 인덱스를 생성해야 한다'이다. 올바르게는 검색 조건, 정렬, 조인 등에 자주 사용되고 선택도가 적절한 열을 중심으로 인덱스를 생성해야 한다. 모든 열에 인덱스를 만들면 저장 공간을 더 쓰고 삽입·수정·삭제 시 인덱스 갱신 비용이 커질 수 있다.
ㄹ X — 틀린 표현은 '테이블 내의 데이터를 순차적으로 접근하여'이다. 올바르게는 인덱스의 탐색 구조를 이용해 조건에 맞는 데이터 위치를 찾아 불필요한 테이블 전체 순차 검색을 줄이는 방식이다.

선지별 해설

① X — 옳은 보기는 ㄱ 하나가 아니라 ㄱ과 ㄴ의 2개이다. 올바르게는 '2개'이다.
② O — ㄱ과 ㄴ이 옳고 ㄷ과 ㄹ이 틀리므로 옳은 보기의 총 개수는 2개이다.
③ X — 옳은 보기는 3개가 아니라 2개이다. 올바르게는 '2개'이다.
④ X — 옳은 보기는 4개가 아니라 2개이다. 올바르게는 '2개'이다.

오답 포인트

인덱스가 검색을 빠르게 한다는 사실을 '모든 열에 만들수록 전체 성능이 좋아진다'로 확대하면 ㄷ을 옳다고 판단하기 쉽다. 또한 인덱스를 사용한 접근과 테이블 전체를 처음부터 훑는 순차 접근을 같은 것으로 보면 ㄹ도 놓치기 쉽다.

다시 나오면 이렇게 판단

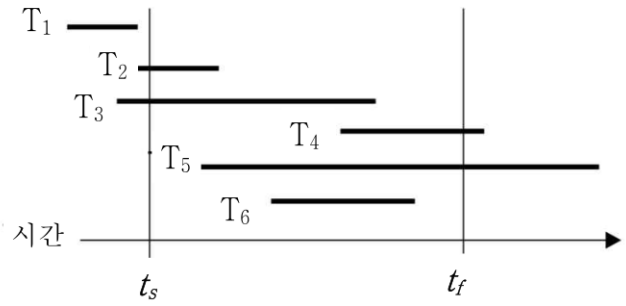
각 보기를 먼저 독립적으로 O/X 판단한 뒤 O의 개수를 센다. 인덱스 문항에서는 '검색 결과는 동일, 검색 속도는 향상 가능, 변경 비용은 증가, 필요한 열에 선택적으로 생성'이라는 기준으로 판단한다.

암기 포인트

인덱스는 결과를 바꾸는 장치가 아니라 탐색을 빠르게 하는 장치이며, 많을수록 무조건 좋은 것은 아니다.

NoteBase

문 4. 트랜잭션(transaction)의 복구(recovery) 진행 시 복구대상을 제외, 재실행(Redo), 실행취소(Undo) 할 것으로 구분하였을 때 옳은 것은?



T_1, T_2, T_3, T_4, T_5, T_6 선분은 각각 해당 트랜잭션의 시작과 끝 시점을, t_s는 검사점(checkpoint)이 이루어진 시점을, t_f는 장애(failure)가 발생한 시점을 의미한다.

	제외	재실행	실행취소
①	T_1	T_2, T_3	T_4, T_5, T_6
②	T_1	T_2, T_3, T_6	T_4, T_5
③	T_2, T_3	T_1, T_6	T_4, T_5
④	T_4, T_5	T_6	T_1, T_2, T_3

정답 2번

출제 포인트

이 문제는 체크포인트 시점 t_s와 장애 시점 t_f을 기준으로 각 트랜잭션을 복구 제외, 재실행(Redo), 실행취소(Undo)로 구분하는 문제이다. 체크포인트 전에 완료된 트랜잭션은 제외하고, 체크포인트 이후부터 장애 전까지 완료된 트랜잭션은 재실행하며, 장애 시점까지 완료되지 않은 트랜잭션은 실행취소한다.

NoteBase

선지별 해설

① X — T_6은 체크포인트 후 시작하여 장애 전에 완료되었으므로 재실행 대상인데 실행취소 대상으로 잘못 분류하였다. 올바르게는 제외 T_1, 재실행 T_2, T_3, T_6, 실행취소 T_4, T_5이다.
② O — T_1은 체크포인트 전에 완료되어 복구 대상에서 제외된다. T_2, T_3, T_6은 체크포인트 이후의 갱신을 반영할 수 있도록 장애 전에 완료된 트랜잭션이므로 재실행하고, T_4, T_5는 장애 시점에 진행 중이므로 실행취소한다.
③ X — T_2, T_3을 제외하고 T_1을 재실행한 분류가 잘못되었다. T_1은 체크포인트 전에 완료되어 제외하고, T_2, T_3, T_6을 재실행하며, T_4, T_5를 실행취소해야 한다.
④ X — T_4, T_5는 장애 시점에 완료되지 않았으므로 제외가 아니라 실행취소 대상이다. 또한 T_1은 제외 대상이고 T_2, T_3, T_6은 재실행 대상이다. 올바른 분류는 제외 T_1, 재실행 T_2, T_3, T_6, 실행취소 T_4, T_5이다.

오답 포인트

트랜잭션의 시작 시점만 보고 판단하면 틀리기 쉽다. T_2, T_3처럼 체크포인트 전에 시작했더라도 체크포인트 후에 완료되면 재실행 대상이며, T_6처럼 체크포인트 후에 시작했더라도 장애 전에 완료되면 역시 재실행 대상이다. 장애선을 통과하는 T_4, T_5만 미완료 트랜잭션이므로 실행취소한다.

다시 나오면 이렇게 판단

먼저 각 트랜잭션의 종료점이 체크포인트보다 앞인지, 체크포인트와 장애 사이인지, 장애 뒤까지 이어지는지를 확인한다. 종료점이 체크포인트 전이면 제외, 체크포인트와 장애 사이이면 재실행, 장애 시점에도 진행 중이면 실행취소로 분류한다.

암기 포인트

체크포인트 전 완료는 제외, 장애 전 완료는 Redo, 장애 시 미완료는 Undo이다.

NoteBase

문 5. 다음 워크시트에서 수식 =VLOOKUP(LARGE(C4:C11,3), C4:F11, 4, 0)에 의해 표시될 수 있는 결과로 옳은 것은?

	A	B	C	D	E	F
1	2021년 1월 판매현황 분석					
2						
3	상품명	판매단가	초과/부족수량	목표수량	판매수량	판매금액
4	공기청정기	150	10	100	110	16,500
5	김치냉장고	85	13	15	28	2,380
6	드럼세탁기	90	-5	35	30	2,700
7	스마트TV	150	13	45	58	8,700
8	의류건조기	230	5	20	25	5,750
9	인덕션오븐	120	20	30	50	6,000
10	무선청소기	70	8	30	38	2,660
11	식기세척기	150	-10	40	30	4,500

① 58 ② 2,380 ③ 8,700 ④ 16,500

정답 2번

출제 포인트

이 문제는 LARGE 함수로 범위에서 세 번째로 큰 값을 구한 뒤, 그 값을 VLOOKUP 함수의 찾을 값으로 사용하여 정확히 일치하는 첫 번째 행의 판매금액을 구하는 문제이다.

조건 정리

주어진 값: C4:C11은 10, 13, -5, 13, 5, 20, 8, -10이다.
구해야 할 값: LARGE(C4:C11,3)의 결과를 C4:C11에서 찾아 C4:F11 범위의 네 번째 열인 F열 값을 반환한 결과이다.
주의할 조건: 마지막 인수 0은 정확히 일치하는 값을 찾는다는 뜻이며, 같은 값이 여러 개이면 위에서부터 처음 만나는 행을 반환한다.

사용할 공식

LARGE(범위, *k*)는 범위에서 *k*번째로 큰 값을 반환한다.
VLOOKUP(찾을 값, 범위, 열 번호, 0)은 범위의 첫 번째 열에서 찾을 값과 정확히 일치하는 첫 번째 행을 찾아 지정한 열의 값을 반환한다.
이 문제에서는 LARGE의 결과가 VLOOKUP의 찾을 값으로 바로 사용되므로 두 함수의 계산 순서와 중복값 처리 방식을 함께 적용한다.

NoteBase

계산 과정

1. C4:C11을 큰 순서로 보면 20, 13, 13, 10, 8, 5, -5, -10이므로 세 번째로 큰 값은 13이다.
2. VLOOKUP은 C4:C11에서 13과 정확히 일치하는 행을 위에서부터 찾는다. 첫 번째 13은 C5에 있다.
3. C4:F11에서 네 번째 열은 F열이므로 F5의 판매금액 2,380을 반환한다. 따라서 수식의 결과는 2,380이다.

선지 판단

① X — 58은 스마트TV 행의 판매수량 E7 값이다. 이 수식은 첫 번째 열에서 13을 찾고 네 번째 열인 F열 값을 반환하므로, 올바른 결과는 2,380이다.
② O — 첫 번째로 발견되는 13은 C5이고, 해당 행의 네 번째 열 F5 값이 2,380이므로 수식 결과와 일치한다.
③ X — 8,700은 두 번째 13이 있는 스마트TV 행의 F7 값이다. VLOOKUP은 중복된 정확한 일치값 중 첫 번째 행을 반환하므로, 올바른 결과는 2,380이다.
④ X — 16,500은 C열 값이 10인 공기청정기 행의 F4 값이다. 찾을 값은 10이 아니라 13이므로, 올바른 결과는 2,380이다.

오답 포인트

LARGE의 세 번째 큰 값을 서로 다른 값 중 세 번째 값인 10으로 세거나, C열에 있는 두 개의 13 중 뒤쪽 행을 선택하면 틀린다. LARGE는 중복값도 각각 순위에 포함하고, VLOOKUP의 정확히 일치 검색은 위에서부터 처음 만난 일치 행을 반환한다.

다시 나오면 이렇게 판단

중첩 함수는 안쪽 함수부터 계산한다. 먼저 LARGE의 결과를 중복값까지 포함해 구하고, 이어서 VLOOKUP 범위의 첫 번째 열에서 그 값을 위에서부터 찾은 뒤 지정된 열 번호를 센다. 마지막 인수가 0이면 정확히 일치하는 첫 번째 행을 선택한다.

암기 포인트

LARGE는 중복값도 순위에 포함하고, VLOOKUP의 정확히 일치는 첫 번째 일치 행을 반환한다.

NoteBase

문 6. UML의 클래스 다이어그램에서 클래스 사이의 관계에 대한 설명으로 옳지 않은 것은?

① 일반화(generalization) 관계는 일반화한 부모 클래스와 실체화한 자식 클래스 간의 상속 관계를 나타낸다.

② 연관(association) 관계에서 다중성(multiplicity)은 관계 사이에 개입하는 클래스의 인스턴스 개수를 의미한다.

③ 의존(dependency) 관계는 한 클래스가 다른 클래스를 참조하는 것으로 지역변수, 매개변수 등을 일시적으로 사용하는 관계이다.

④ 집합(aggregation) 관계는 강한 전체와 부분의 클래스 관계이므로 전체 객체가 소멸되면 부분 객체도 소멸된다.

정답 4번

출제 포인트

UML 클래스 다이어그램의 일반화, 연관, 의존, 집합 관계를 구별하고, 집합과 합성의 생명주기 차이를 판단하는 문제이다.

핵심 개념

일반화는 일반적인 부모 클래스와 구체적인 자식 클래스 사이의 상속 관계이다.
연관은 클래스의 객체들이 지속적으로 서로 연결되는 구조적 관계이며, 다중성은 한 객체와 연결될 수 있는 상대 클래스 인스턴스의 수를 나타낸다.
의존은 한 클래스가 다른 클래스를 매개변수나 지역변수 등으로 일시적으로 사용하는 관계이다.
집합은 부분 객체가 전체 객체와 독립적으로 존재할 수 있는 약한 전체-부분 관계이다. 합성은 부분 객체의 생명주기가 전체 객체에 종속되는 강한 전체-부분 관계이다.
집합은 부분의 독립 생존, 합성은 전체와 부분의 생명주기 종속으로 구별한다.

NoteBase

선지별 해설

① O — 일반화는 일반화된 부모 클래스와 실체화된 자식 클래스 사이의 상속 관계를 나타내므로 옳다.
② O — 연관의 다중성은 연관 끝에 표시되며, 한 클래스의 인스턴스 하나와 연결될 수 있는 상대 클래스 인스턴스의 개수를 나타내므로 옳다.
③ O — 의존은 한 클래스가 다른 클래스를 매개변수·지역변수 등으로 참조하여 일시적으로 사용하는 관계이므로 옳다.
④ X — 틀린 표현은 '집합 관계는 강한 전체와 부분의 관계이고, 전체 객체가 소멸되면 부분 객체도 소멸된다'이다. 올바르게는 집합은 부분 객체가 전체 객체와 독립적으로 존재할 수 있는 약한 전체–부분 관계이다. 전체가 소멸될 때 부분도 함께 소멸하는 강한 관계는 합성이다.

오답 포인트

④는 집합과 합성의 특징을 바꾸어 제시한 선지이다. '전체가 소멸하면 부분도 소멸한다'는 생명주기 종속 표현은 집합이 아니라 합성을 가리킨다.

다시 나오면 이렇게 판단

전체가 없어져도 부분이 독립적으로 남을 수 있으면 집합, 전체가 없어질 때 부분도 함께 없어지면 합성으로 판단한다. 일반화는 상속, 연관은 지속적 연결과 다중성, 의존은 일시적 사용을 기준으로 구별한다.

암기 포인트

집합은 약한 소유·독립 생존, 합성은 강한 소유·생명주기 종속이다.

NoteBase

문 7. 다음에서 설명하는 소프트웨어 아키텍처의 유형으로 옳은 것은?

> ○ 사용자 인터페이스를 시스템의 비즈니스 로직 부분과 분리하는 구조
> ○ 결합도(coupling)를 낮추기 위한 소프트웨어 아키텍처 패턴 구조
> ○ 디자인 패턴 중 옵서버(observer) 패턴에 해당하는 구조

① 클라이언트-서버(client-server) 아키텍처
② 브로커(broker) 아키텍처
③ MVC(Model-View-Controller) 아키텍처
④ 계층형(layered) 아키텍처

정답 3번

출제 포인트

사용자 인터페이스와 비즈니스 로직의 분리, 낮은 결합도, 옵서버 패턴이라는 단서를 소프트웨어 아키텍처 유형과 연결하는 문제이다.

핵심 개념

MVC 아키텍처는 데이터를 담당하는 Model, 화면 표시를 담당하는 View, 입력과 흐름을 제어하는 Controller로 역할을 분리하는 구조이다. Model과 View의 분리로 사용자 인터페이스와 비즈니스 로직의 결합도를 낮출 수 있으며, Model의 상태 변화를 View에 알리는 관계는 옵서버 패턴으로 구현할 수 있다.
클라이언트-서버는 서비스 요청자와 제공자의 역할 분리가 핵심이고, 브로커는 중개자를 통한 분산 컴포넌트 간 통신이 핵심이며, 계층형은 기능을 여러 계층으로 나누어 상하 계층 사이의 인터페이스로 연결하는 구조이다.

NoteBase

선지별 해설

① X
클라이언트-서버 아키텍처는 클라이언트가 서비스를 요청하고 서버가 이를 제공하는 분산 구조이다. 제시된 세 단서, 특히 옵서버 패턴을 구조적 특징으로 하는 아키텍처라는 연결이 틀리다. 올바르게는 '서비스 요청자와 서비스 제공자를 분리하는 구조'이다.

② X
브로커 아키텍처는 브로커가 분산된 컴포넌트 사이의 요청과 응답을 중개하여 직접적인 의존성을 줄이는 구조이다. 결합도를 낮춘다는 단서 일부에는 부합하지만 사용자 인터페이스와 비즈니스 로직의 분리 및 옵서버 패턴이라는 단서까지 충족하지 못한다. 올바르게는 '브로커가 컴포넌트 간 통신을 중개하는 구조'이다.

③ O
MVC 아키텍처는 Model, View, Controller의 책임을 분리하므로 사용자 인터페이스와 비즈니스 로직의 결합도를 낮춘다. 또한 Model의 변경을 View에 통지하는 관계를 옵서버 패턴으로 구현할 수 있으므로 세 설명에 모두 부합한다.

④ X
계층형 아키텍처는 시스템의 책임을 프레젠테이션 계층, 비즈니스 계층, 데이터 계층 등으로 나누고 정해진 계층 인터페이스를 통해 상호 작용하게 하는 구조이다. 관심사 분리와 낮은 결합도에는 도움이 되지만 옵서버 패턴이 이 구조를 식별하는 핵심 단서는 아니다. 올바르게는 '시스템을 책임별 계층으로 분리하는 구조'이다.

오답 포인트

'결합도를 낮춘다'는 표현은 브로커와 계층형에도 적용될 수 있어 이 단서 하나만 보면 혼동하기 쉽다. 이 문항에서는 '사용자 인터페이스와 비즈니스 로직의 분리'와 '옵서버 패턴'까지 세 단서를 함께 만족하는지를 판단해야 한다.

다시 나오면 이렇게 판단

화면과 업무 데이터·로직의 분리, Model·View·Controller의 역할 구분, Model 변경에 따른 View 갱신이 함께 제시되면 MVC로 판단한다. 요청자와 제공자는 클라이언트-서버, 통신 중개자는 브로커, 상하 책임 분할은 계층형으로 구별한다.

암기 포인트

MVC는 Model·View·Controller의 책임 분리이며, Model의 변화가 View에 전달되는 관계는 옵서버 패턴과 연결된다.

NoteBase

문 8. 다음 C 프로그램의 실행 결과로 옳은 것은?

```
#include <stdio.h>

void main(void) {
        int a = 1, b = 2, c = 3;
        {
          int b = 4, c = 5;
          a = b;
          {
            int c;
            c = b;
          }
          printf("%d %d %d\n", a, b, c);
        }
}
```

① 1 2 3
② 1 4 5
③ 4 2 3
④ 4 4 5

정답 4번

출제 포인트

이 문제는 C 언어에서 중첩 블록의 변수 유효 범위와 같은 이름의 안쪽 변수가 바깥쪽 변수를 가리는 원리를 묻는다.

핵심 개념

블록 유효 범위: 중괄호 안에서 선언한 지역 변수는 그 선언 지점부터 해당 블록이 끝날 때까지만 사용할 수 있다.
변수 가림: 안쪽 블록에서 바깥쪽 변수와 같은 이름을 선언하면, 안쪽 블록에서는 가장 가까운 선언의 변수가 사용된다.
실행 흐름: 바깥쪽 변수는 처음에 a=1, b=2, c=3이다. 첫 번째 안쪽 블록의 b=4와 c=5가 바깥쪽 b와 c를 가린다. 따라서 a=b에서 바깥쪽 a가 4가 된다. 더 안쪽 블록의 c=b에서는 새로 선언한 c에 첫 번째 안쪽 블록의 b 값 4가 저장되지만, 이 c는 그 블록이 끝나면 유효하지 않다. printf가 실행되는 위치에서는 a=4, b=4, c=5이다.

NoteBase

선지별 해설

① X

1 2 3은 변수 가림과 a=b의 대입을 반영하지 않은 결과이다. 틀린 출력 1 2 3 → 올바른 출력은 4 4 5이다.

② X

1 4 5는 첫 번째 안쪽 블록의 b와 c는 반영했지만, a=b로 바깥쪽 a가 1에서 4로 바뀌는 것을 빠뜨렸다. 틀린 출력 1 4 5 → 올바른 출력은 4 4 5이다.

③ X

4 2 3은 a의 변경만 반영하고 printf가 첫 번째 안쪽 블록 안에서 실행되어 그 블록의 b=4와 c=5를 참조한다는 점을 놓친 결과이다. 틀린 출력 4 2 3 → 올바른 출력은 4 4 5이다.

④ O

a=b에서 바깥쪽 a는 4가 되고, printf에서는 첫 번째 안쪽 블록의 b=4와 c=5가 유효하므로 4 4 5가 출력된다.

오답 포인트

가장 안쪽 블록의 c=b 때문에 출력되는 c도 4라고 착각하기 쉽다. 그러나 그 c는 가장 안쪽 블록이 끝나면서 유효 범위를 벗어나며, printf에서는 바로 바깥 블록의 c=5를 참조한다.

다시 나오면 이렇게 판단

각 문장의 위치에서 같은 이름의 선언 중 가장 가까운 변수를 찾고, 대입으로 실제 값이 바뀌는 변수를 표시한다. 블록을 벗어날 때는 그 블록에서 선언된 변수를 제외한 뒤 출력문 위치에서 다시 값을 추적한다.

암기 포인트

안쪽의 같은 이름은 바깥 변수를 가리고, 블록이 끝나면 안쪽 변수도 사라진다.

NoteBase

문 9. 클라우드 서버에 저장된 데이터 용량이 1024PB(Peta Byte)일 때 이 데이터와 동일한 크기의 저장 용량으로 옳지 않은 것은? (단, 1KB는 1024Byte)

① 1024^{-1}ZB(Zetta Byte)

② 1024^{2}TB(Tera Byte)

③ 1024^{-3}YB(Yotta Byte)

④ 1024^{4}MB(Mega Byte)

정답 3번

출제 포인트

인접한 데이터 저장 단위가 1024배씩 차이 난다는 관계를 이용하여 서로 다른 단위 표현이 같은 용량인지 판단하는 문제이다.

조건 정리

주어진 용량: 1024PB
구해야 할 것: 주어진 용량과 같지 않은 표현
주의할 조건: 문제에서 1KB = 1024Byte를 제시했으므로 인접 단위 사이를 1024배 관계로 환산한다.

사용할 공식

저장 단위의 순서는 MB → GB → TB → PB → EB → ZB → YB이고, 오른쪽의 한 단계 큰 단위는 왼쪽 단위의 1024배 용량이다. 따라서 1EB = 1024PB이다. 각 선지가 1EB와 같은지 비교하기 위해 이 관계를 사용한다.

계산 과정

주어진 용량을 기준 단위로 바꾸면 다음과 같다.
1024PB = 1EB
그리고 단위 간 관계에 따라 다음 등식이 성립한다.
$1EB = 1024^{-1}ZB = 1024^{2}TB = 1024^{4}MB$
반면 YB에서 세 단계 작은 단위는 PB이므로 다음과 같다.
$1024^{-3}YB = 1PB$
따라서 ③만 1EB와 같지 않다.

NoteBase

선지 판단

① O — 1ZB = 1024EB이므로 1024^{-1}ZB = 1EB = 1024PB이다.
② O — 1PB = 1024TB이므로 1024^{2}TB = 1024PB = 1EB이다.
③ X — 1024^{-3}YB는 1PB이므로 주어진 1024PB = 1EB보다 작다. 틀린 표현 1024^{-3}YB는 올바르게 1024^{-2}YB로 고치면 주어진 용량과 같다.
④ O — MB에서 EB까지는 네 단계이므로 1024^{4}MB = 1EB = 1024PB이다.

오답 포인트

③에서 YB → ZB → EB → PB의 세 단계를 세면 1024^{-3}YB는 1PB이다. 주어진 값은 1024PB이므로, 단계를 하나 더 내려가 같은 값이라고 착각하는 것이 함정이다. 또한 문제의 부정형 표현인 '옳지 않은 것'을 놓치지 않아야 한다.

다시 나오면 이렇게 판단

먼저 주어진 용량을 하나의 기준 단위로 환산하고, 단위가 한 단계 커질 때마다 수치의 지수를 1만큼 줄여 각 선지를 같은 기준 단위로 비교한다. 마지막에는 문제의 극성인 '옳은 것'과 '옳지 않은 것'을 다시 확인한다.

암기 포인트

MB → GB → TB → PB → EB → ZB → YB: 한 단계마다 1024배이며, 1024PB = 1EB이다.

NoteBase

문 10. 유비쿼터스 컴퓨팅 기술에 대한 설명으로 옳지 않은 것은?

① 노매딕 컴퓨팅(nomadic computing)은 사용자가 모든 장소에서 사용자 인증 없이 다양한 정보기기로 동일한 데이터에 접근하는 기술이다.

② 엑조틱 컴퓨팅(exotic computing)은 스스로 생각하여 현실세계와 가상세계를 연계하는 컴퓨팅을 실현해 주는 기술이다.

③ 감지 컴퓨팅(sentient computing)은 센서가 사용자의 상황을 인식하여 사용자가 필요한 정보를 제공해 주는 기술이다.

④ 임베디드 컴퓨팅(embedded computing)은 사물에 마이크로칩을 장착하여 서비스 기능을 내장하는 컴퓨팅 기술이다.

정답 1번

출제 포인트

유비쿼터스 컴퓨팅을 구현하는 노매딕 컴퓨팅, 엑조틱 컴퓨팅, 감지 컴퓨팅, 임베디드 컴퓨팅의 의미를 구별하는 문제이다.

핵심 개념

노매딕 컴퓨팅: 사용자가 장소나 사용하는 정보기기에 크게 구애받지 않고 필요한 정보와 서비스에 접근할 수 있게 하는 기술이다. 접근의 편재성이 핵심이며, 사용자 인증을 생략한다는 뜻은 아니다.
엑조틱 컴퓨팅: 컴퓨터가 스스로 상황을 판단하고 현실세계와 가상세계를 연계하도록 하는 기술이다.
감지 컴퓨팅: 센서로 사용자와 주변 상황을 인식하여 상황에 맞는 정보나 서비스를 제공하는 기술이다.
임베디드 컴퓨팅: 사물이나 기기에 마이크로프로세서 등을 내장하여 정해진 기능이나 서비스를 수행하게 하는 기술이다.

NoteBase

선지별 해설

① X — 노매딕 컴퓨팅은 장소와 기기의 제약을 줄여 어디서나 데이터에 접근하게 하는 기술이지만, '사용자 인증 없이' 접근하는 기술은 아니다. 틀린 표현인 '사용자 인증 없이'는 '필요한 사용자 인증과 보안 절차를 거쳐'로 고쳐야 한다.
② O — 엑조틱 컴퓨팅은 컴퓨터가 스스로 판단하여 현실세계와 가상세계를 연계하는 컴퓨팅을 구현하는 기술이라는 설명에 해당한다.
③ O — 감지 컴퓨팅은 센서가 사용자의 상태와 주변 상황을 인식하고, 그 상황에 필요한 정보를 제공하는 기술이다.
④ O — 임베디드 컴퓨팅은 사물이나 기기에 마이크로프로세서와 컴퓨팅 기능을 내장하여 특정 서비스 기능을 수행하게 하는 기술이다.

오답 포인트

①은 '모든 장소에서 다양한 정보기기로 동일한 데이터에 접근한다'는 앞부분이 노매딕 컴퓨팅의 특징과 비슷해 보이게 만든 뒤, 보안과 관련된 '사용자 인증 없이'를 끼워 넣었다. 어디서나 접근할 수 있다는 것과 인증 없이 접근한다는 것은 서로 다른 의미이다.

다시 나오면 이렇게 판단

장소·기기의 제약 없는 접근은 노매딕, 현실세계와 가상세계의 연계는 엑조틱, 센서를 통한 상황 인식은 감지, 사물 내부에 컴퓨팅 기능을 넣는 것은 임베디드로 판단한다. 접근 편의성을 인증 생략이나 보안 절차 부재로 바꾸어 설명한 선지는 틀린 것으로 판단한다.

암기 포인트

노매딕은 어디서나 접근, 엑조틱은 현실·가상 연계, 감지는 센서·상황 인식, 임베디드는 사물 속 컴퓨터이다.

NoteBase

문 11. 하나의 컴퓨터 시스템에서 여러 개의 어플리케이션(application)들이 함께 주기억장치에 적재되어 하나의 CPU 자원을 번갈아 사용하는 형태로 수행되게 하는 기법으로 옳은 것은?

① 다중프로그래밍(multi-programming)
② 다중프로세싱(multi-processing)
③ 병렬처리(parallel processing)
④ 분산처리(distributed processing)

정답 1번

출제 포인트

여러 응용 프로그램을 주기억장치에 함께 적재한 뒤 하나의 CPU가 번갈아 실행하는 다중프로그래밍의 개념을 구별하는 문제이다.

핵심 개념

다중프로그래밍은 여러 프로그램을 주기억장치에 동시에 적재하고, 하나의 CPU가 실행할 프로그램을 번갈아 선택하여 CPU 이용률을 높이는 방식이다.
다중프로세싱은 한 컴퓨터 시스템에서 둘 이상의 CPU 또는 처리 장치가 여러 작업을 동시에 수행하는 방식이다.
병렬처리는 여러 처리 장치가 하나의 작업을 나누어 동시에 처리하는 방식이다.
분산처리는 네트워크로 연결된 여러 독립 컴퓨터가 작업과 자원을 나누어 처리하는 방식이다.

선지별 해설

① O — 여러 응용 프로그램이 주기억장치에 함께 적재되고 하나의 CPU를 번갈아 사용한다는 설명은 다중프로그래밍에 해당한다.
② X — 틀린 부분은 '다중프로세싱'이다. 다중프로세싱은 둘 이상의 CPU 또는 처리 장치가 여러 작업을 동시에 수행하는 방식이며, 제시된 방식은 다중프로그래밍이다.
③ X — 틀린 부분은 '병렬처리'이다. 병렬처리는 여러 처리 장치가 하나의 작업을 나누어 동시에 처리하는 방식이며, 하나의 CPU가 여러 프로그램을 번갈아 실행하는 방식은 다중프로그래밍이다.
④ X — 틀린 부분은 '분산처리'이다. 분산처리는 네트워크로 연결된 여러 독립 컴퓨터가 작업을 나누어 처리하는 방식이며, 한 시스템의 하나의 CPU를 번갈아 사용하는 방식은 다중프로그래밍이다.

오답 포인트

'여러 프로그램'이라는 표현만 보고 다중프로세싱을 고르기 쉽다. 이 문항의 결정적 단서는 '함께 주기억장치에 적재'와 '하나의 CPU를 번갈아 사용'이며, 이는 여러 CPU를 사용하는 다중프로세싱이 아니라 다중프로그래밍을 뜻한다.

NoteBase

다시 나오면 이렇게 판단

CPU 수와 실행 관계를 먼저 확인한다. 하나의 CPU가 주기억장치의 여러 프로그램을 번갈아 실행하면 다중프로그래밍, 여러 CPU가 동시에 처리하면 다중프로세싱, 여러 독립 컴퓨터가 네트워크로 협력하면 분산처리로 판단한다.

암기 포인트

여러 프로그램·하나의 CPU·번갈아 실행은 다중프로그래밍이다.

NoteBase

문 12. 주기억장치와 CPU 캐시 기억장치만으로 구성된 시스템에서 다음과 같이 기억장치 접근 시간이 주어질 때 이 시스템의 캐시 적중률(hit ratio)로 옳은 것은?

- ㅇ 주기억장치 접근 시간: T_m=80ns
- ㅇ CPU 캐시 기억장치 접근 시간: T_c=10ns
- ㅇ 기억장치 평균 접근 시간(expected memory access time): T_a=17ns

① 80% ② 85% ③ 90% ④ 95%

정답 3번

출제 포인트

이 문제는 캐시 적중 시와 실패 시의 접근 시간을 적중률에 따라 가중평균하여 캐시 적중률을 구하는 문제이다.

조건 정리

주기억장치 접근 시간: $T_m = 80\text{ns}$
CPU 캐시 기억장치 접근 시간: $T_c = 10\text{ns}$
평균 접근 시간: $T_a = 17\text{ns}$
구해야 할 값: 캐시 적중률 h
이 문항에서는 적중 시 10ns, 실패 시 80ns가 걸리는 두 경우의 가중평균을 적용한다.

사용할 공식

$T_a = hT_c + (1 - h)T_m$
적중할 확률은 h, 실패할 확률은 $1 - h$이므로 각 경우의 접근 시간에 그 확률을 곱해 평균 접근 시간을 구한다.

계산 과정

주어진 값을 공식에 대입한다.
$17 = 10h + 80(1 - h)$
식을 정리한다.
$17 = 80 - 70h$
$70h = 63$
$h = 0.9 = 90\%$
따라서 캐시 적중률은 90%이다.

NoteBase

선지 판단

① X — $h = 0.80$이면 $T_a = 0.80 \times 10 + 0.20 \times 80 = 24$ns로, 주어진 17ns와 다르다. 80%가 아니라 90%가 올바르다.
② X — $h = 0.85$이면 $T_a = 0.85 \times 10 + 0.15 \times 80 = 20.5$ns로, 주어진 값보다 크다. 85%가 아니라 90%가 올바르다.
③ O — $h = 0.90$이면 $T_a = 0.90 \times 10 + 0.10 \times 80 = 17$ns가 되어 조건과 일치한다.
④ X — $h = 0.95$이면 $T_a = 0.95 \times 10 + 0.05 \times 80 = 13.5$ns로, 주어진 값보다 작다. 95%가 아니라 90%가 올바르다.

오답 포인트

실패 확률을 h로 두거나 적중률과 실패율을 모두 h로 두면 식이 잘못된다. 적중률이 h이면 실패율은 반드시 $1 - h$이며, 이 문항에서는 각 접근 시간에 해당 확률을 곱한 가중평균을 사용한다.

다시 나오면 이렇게 판단

평균 접근 시간 문제에서는 먼저 적중률을 h, 실패율을 $1 - h$로 놓고, 각 경우의 시간에 확률을 곱한 뒤 합하여 방정식을 세운다. 마지막에는 구한 적중률을 원래 식에 대입해 평균 시간이 일치하는지 검산한다.

암기 포인트

평균 접근 시간은 '적중 확률 × 캐시 시간 + 실패 확률 × 주기억장치 시간'이다: $T_a = hT_c + (1 - h)T_m$.

NoteBase

문 13. 컴퓨터 시스템의 주기억장치 및 보조기억장치에 대한 설명으로 옳지 않은 것은?

① RAM은 휘발성(volatile) 기억장치이며 HDD 및 SSD는 비휘발성(non-volatile) 기억장치이다.

② RAM의 경우, HDD나 SSD 등의 보조기억장치에 비해 상대적으로 접근 속도가 빠르다.

③ SSD에서는 일반적으로 특정 위치의 데이터를 읽는 데 소요되는 시간이 같은 위치에 데이터를 쓰는 데 소요되는 시간보다 더 오래 걸린다.

④ SSD의 경우, 일반적으로 HDD보다 가볍고 접근 속도가 빠르며 전력 소모가 적다.

정답 3번

출제 포인트

RAM과 HDD·SSD의 휘발성, 접근 속도, 읽기·쓰기 속도 및 물리적 특성을 비교하는 문제이다.

핵심 개념

RAM: CPU가 실행 중인 프로그램과 데이터를 임시로 저장하는 주기억장치이다. 전원이 꺼지면 내용이 사라지는 휘발성 메모리이며, 보조기억장치보다 접근 속도가 빠르다.
HDD: 회전하는 자기 디스크와 기계식 헤드를 이용하는 비휘발성 보조기억장치이다.
SSD: 플래시 메모리를 이용하는 비휘발성 보조기억장치이다. 기계식 구동부가 없어 일반적으로 HDD보다 가볍고 빠르며 전력 소모가 적고, 읽기 속도가 쓰기 속도보다 빠르다.

NoteBase

선지별 해설

① O
RAM은 전원이 끊기면 저장 내용이 사라지는 휘발성 기억장치이고, HDD와 SSD는 전원이 꺼져도 데이터가 유지되는 비휘발성 기억장치이다.

② O
RAM은 CPU가 처리할 데이터를 임시로 보관하는 주기억장치이므로 HDD나 SSD 같은 보조기억장치보다 일반적으로 접근 속도가 빠르다.

③ X
틀린 표현은 '읽는 데 소요되는 시간이 쓰는 데 소요되는 시간보다 더 오래 걸린다'이다. 일반적으로 SSD는 읽기 속도가 쓰기 속도보다 빠르므로, 올바르게는 '특정 위치의 데이터를 읽는 데 걸리는 시간이 같은 위치에 데이터를 쓰는 데 걸리는 시간보다 짧다'이다.

④ O
SSD는 기계식 회전판과 헤드가 없어 일반적으로 HDD보다 가볍고 접근 속도가 빠르며 전력 소모가 적다.

오답 포인트

③은 '읽기 시간'과 '쓰기 시간'의 대소 관계를 뒤집은 선지이다. SSD는 읽기보다 쓰기 과정에 더 많은 시간이 드는 것이 일반적이므로, 읽기 시간이 더 오래 걸린다는 설명이 틀리다.

다시 나오면 이렇게 판단

전원 차단 후 데이터 유지 여부로 휘발성을 구분하고, 장치 구조로 성능을 판단한다. RAM은 휘발성·고속이고, SSD는 기계식 구동부가 없어 HDD보다 빠르고 가벼우며 저전력이다. SSD의 읽기와 쓰기를 비교할 때는 일반적으로 읽기가 더 빠르다고 판단한다.

암기 포인트

RAM은 휘발성·고속, HDD와 SSD는 비휘발성이다. SSD는 HDD보다 가볍고 빠르며 저전력이고, 일반적으로 읽기가 쓰기보다 빠르다.

NoteBase

문 14. 다음 표에서 보인 4개의 프로세스들을 시간 할당량(time quantum)이 5인 라운드로빈(round-robin) 스케줄링 기법으로 실행시켰을 때 평균 반환 시간으로 옳은 것은?

프로세스	도착 시간	실행 시간
P1	0	10
P2	1	15
P3	3	6
P4	6	9

(단, 반환 시간이란 프로세스가 도착하는 시점부터 실행을 종료할 때까지 소요된 시간을 의미한다. 또한, 이들 4개의 프로세스들은 I/O 없이 CPU만을 사용한다고 가정하며, 문맥교환(context switching)에 소요되는 시간은 무시한다.)

① 24.0 ② 29.0 ③ 29.75 ④ 30.25

정답 2번

출제 포인트

이 문제는 시간 할당량이 5인 라운드로빈 스케줄링의 실행 순서를 구한 뒤 각 프로세스의 반환 시간을 평균하는 문제이다.

조건 정리

시간 할당량은 5이고 문맥교환 시간은 무시한다.
도착 시간과 실행 시간은 각각 P1: 0, 10 / P2: 1, 15 / P3: 3, 6 / P4: 6, 9이다.
반환 시간은 프로세스의 완료 시각에서 도착 시각을 뺀 값이다.

사용할 공식

프로세스 i의 반환 시간은 $T_i = C_i - A_i$이다. 여기서 C_i는 완료 시각, A_i는 도착 시각이다.
평균 반환 시간은 $\frac{T_1 + T_2 + T_3 + T_4}{4}$이다. 라운드로빈에서는 준비 큐의 선두 프로세스가 최대 5만큼 실행된 뒤, 남은 실행 시간이 있으면 큐의 뒤로 이동하므로 이 공식을 쓰기 전에 각 완료 시각을 실행 순서대로 구해야 한다.

NoteBase

계산 과정

실행 순서는 0~5 P1, 5~10 P2, 10~15 P3, 15~20 P1, 20~25 P4, 25~30 P2, 30~31 P3, 31~35 P4, 35~40 P2이다.
따라서 완료 시각은 P1=20, P2=40, P3=31, P4=35이다.
각 반환 시간은 P1: $20 - 0 = 20$, P2: $40 - 1 = 39$, P3: $31 - 3 = 28$, P4: $35 - 6 = 29$이다.
평균 반환 시간은 $\frac{20 + 39 + 28 + 29}{4} = \frac{116}{4} = 29.0$이다.

선지 판단

① X — 24.0은 네 프로세스의 반환 시간 20, 39, 28, 29를 평균한 값이 아니다. 틀린 값 24.0은 올바르게는 29.0이다.
② O — 계산한 평균 반환 시간이 29.0과 일치한다.
③ X — 29.75는 계산 결과 29.0과 일치하지 않는다. 틀린 값 29.75는 올바르게는 29.0이다.
④ X — 30.25는 계산 결과 29.0과 일치하지 않는다. 틀린 값 30.25는 올바르게는 29.0이다.

오답 포인트

반환 시간을 실행 시간이나 대기 시간과 혼동하거나, 반환 시간에서 도착 시각을 빼지 않으면 오답이 나온다. 특히 P4는 시각 6에 도착하므로 P2가 5~10 동안 실행되는 사이 준비 큐에 들어가며, P1보다 먼저 실행되는 것으로 처리하면 실행 순서와 완료 시각이 달라진다.

다시 나오면 이렇게 판단

시간축에 도착한 프로세스를 준비 큐에 순서대로 넣고, 각 프로세스를 시간 할당량만큼 실행한 뒤 남은 작업은 큐 뒤로 보낸다. 모든 완료 시각을 구한 다음 완료 시각에서 도착 시각을 빼고 그 값을 평균한다.

암기 포인트

라운드로빈은 준비 큐 순서, 반환 시간은 완료 시각-도착 시각이다.

NoteBase

문 15. LRU(Least Recently Used) 교체 기법을 사용하는 요구 페이징(demand paging) 시스템에서 3개의 페이지 프레임(page frame)을 할당받은 프로세스가 다음과 같은 순서로 페이지에 접근했을 때 발생하는 페이지 부재(page fault) 횟수로 옳은 것은? (단, 할당된 페이지 프레임들은 초기에 모두 비어 있다고 가정한다.)

페이지 참조 순서(page reference string):
1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 4, 5, 6, 7, 4, 5, 6, 7, 4, 5, 6, 7

① 7번 ② 10번 ③ 14번 ④ 15번

정답 4번

출제 포인트

이 문제는 3개의 페이지 프레임에서 LRU 교체를 적용하여 페이지 부재 횟수를 구하는 문제이다. LRU는 페이지 부재가 발생하고 빈 프레임이 없을 때 가장 오랫동안 참조되지 않은 페이지를 교체한다.

조건 정리

주어진 값: 페이지 프레임은 3개이고, 처음에는 모두 비어 있다.
구해야 할 값: 전체 페이지 참조열에서 발생하는 페이지 부재 횟수이다.
주의할 조건: 프레임에 이미 있는 페이지를 다시 참조하면 페이지 적중이므로 부재 횟수에 더하지 않는다.

사용할 원리

각 참조마다 해당 페이지가 프레임에 있는지 확인한다. 없으면 페이지 부재를 1회 더하고, 빈 프레임이 없으면 현재 프레임의 페이지 중 가장 오래전에 사용된 페이지를 내보낸다. 이 원리를 쓰는 이유는 문제에서 교체 기법을 LRU로 지정했기 때문이다.

계산 과정

1. 처음 1, 2, 3을 참조할 때 프레임이 비어 있으므로 각각 1회씩, 총 3회의 페이지 부재가 발생하고 프레임은 [1, 2, 3]이 된다.
2. 이어지는 1, 2, 3의 세 번 반복은 모두 프레임 안의 페이지를 참조하므로 9회 모두 적중이다. 따라서 앞의 12개 참조까지 부재는 3회이다.
3. 다음 4, 5, 6, 7을 참조하면 4가 1을, 5가 2를, 6이 3을, 7이 4를 차례로 교체하므로 4회 모두 부재이다.
4. 이후에도 4, 5, 6, 7이 두 번 더 반복된다. 페이지는 4종류인데 프레임은 3개뿐이어서 다시 참조할 페이지가 직전에 이미 교체된 상태가 되므로 8회 모두 부재이다.
5. 전체 페이지 부재 횟수는 3 + 12 = 15회이다.

NoteBase

선지 판단

① X — ‘7번’은 틀린 값이다. 처음 1, 2, 3에서 발생하는 3회와 뒤의 4, 5, 6, 7 구간에서 발생하는 12회를 모두 합쳐야 하므로 올바르게는 15번이다.
② X — ‘10번’은 틀린 값이다. 4, 5, 6, 7의 12개 참조는 프레임이 3개뿐인 순환 참조라서 일부가 적중하는 것이 아니라 모두 부재가 된다. 올바르게는 15번이다.
③ X — ‘14번’은 틀린 값이다. 마지막 참조 7도 그 직전의 4, 5, 6 참조 과정에서 프레임 밖으로 밀려나 있으므로 페이지 부재가 발생한다. 올바르게는 15번이다.
④ O — ‘15번’은 처음 구간의 3회와 뒤 구간의 12회를 합한 정확한 페이지 부재 횟수이다.

오답 포인트

핵심 함정은 4, 5, 6, 7의 반복에서 적중이 생긴다고 생각하는 것이다. 서로 다른 페이지 4개를 프레임 3개로 순환 참조하면 다음 차례의 페이지가 항상 가장 오래전에 사용된 페이지로서 이미 교체되어 있으므로 12개 참조가 전부 부재가 된다. 특히 마지막 7을 적중으로 잘못 처리하면 14회를 선택하게 된다.

다시 나오면 이렇게 판단

LRU 문제는 참조마다 프레임 존재 여부를 먼저 확인하고, 부재이면서 프레임이 가득 찬 경우에만 가장 오래전에 사용된 페이지를 교체한다. 프레임 수보다 하나 많은 페이지를 같은 순서로 순환 참조하는 구간에서는 모든 참조가 부재가 될 수 있는지도 함께 확인한다.

암기 포인트

LRU는 ‘가장 오래전에 사용한 페이지’를 교체한다. 3프레임에서 4페이지를 순환 참조하면 전부 부재가 된다.

NoteBase

문 16. 인터넷에서 사용하는 IPv6에 대한 설명으로 옳지 않은 것은?

① 패킷 헤더의 체크섬(checksum)을 통해 데이터 무결성 검증 기능을 지원한다.

② QoS(Quality of Service) 보장을 위해 흐름 레이블링(flow labeling) 기능을 지원한다.

③ IPv6의 주소 체계는 16비트씩 8개 부분, 총 128비트로 구성되어 있다.

④ IPv6 주소 표현에서 연속된 0에 대한 생략을 위한 :: 표기는 1번만 가능하다.

정답 1번

출제 포인트

IPv6의 기본 헤더 구조, 흐름 레이블, 주소 길이와 0 압축 표기 규칙을 구별하는 문제이다.

핵심 개념

IPv6는 128비트 주소를 사용하며, 기본 헤더를 단순화하여 라우터의 처리 부담을 줄인 인터넷 프로토콜이다. IPv6 기본 헤더에는 헤더 체크섬 필드가 없고, QoS 처리를 위한 흐름 레이블 필드가 있다. 주소는 16비트 단위 8개로 나누어 16진수로 표현하며, 연속된 0 묶음은 ::로 한 번만 압축할 수 있다.

선지별 해설

① X

틀린 표현은 '패킷 헤더의 체크섬을 통해 데이터 무결성 검증 기능을 지원한다'이다. IPv6 기본 헤더에는 헤더 체크섬 필드가 없다. 올바르게는 IPv6가 헤더 체크섬을 생략하며, 필요한 오류 검출과 무결성 확인은 상위 계층이나 링크 계층의 검증 기능이 담당한다고 표현해야 한다.

② O

IPv6 헤더의 흐름 레이블은 같은 흐름에 속하는 패킷을 식별하여 QoS 등 특별한 처리가 가능하도록 한다.

③ O

IPv6 주소는 총 128비트이며, 16비트씩 8개 부분으로 나누어 각 부분을 16진수로 표현한다.

④ O

연속된 0 부분을 ::로 줄이는 0 압축 표기는 한 IPv6 주소에서 한 번만 사용할 수 있다. 두 번 이상 사용하면 각 ::가 몇 개의 0 부분을 뜻하는지 결정할 수 없기 때문이다.

NoteBase

오답 포인트

IPv4 헤더에 체크섬 필드가 있다는 사실을 IPv6에도 그대로 적용하기 쉽다. IPv6는 기본 헤더를 단순화하면서 헤더 체크섬을 제거했다는 점이 핵심 함정이다.

다시 나오면 이렇게 판단

IPv6 관련 선지는 '128비트 주소', '8개의 16비트 부분', '흐름 레이블 지원', '기본 헤더 체크섬 없음', '::는 주소당 한 번'의 다섯 기준으로 판단한다.

암기 포인트

IPv6는 128비트 · 흐름 레이블 O · 기본 헤더 체크섬 X · :: 한 번이다.

NoteBase

문 17. 다음 정수를 왼쪽부터 순서대로 삽입하여 이진 탐색 트리(binary search tree)를 구성했을 때 단말 노드(leaf node)를 모두 나열한 것은?

44, 36, 62, 3, 16, 51, 75, 68, 49, 85, 57

① 16, 49, 51, 57, 85
② 16, 49, 57, 68, 85
③ 49, 51, 57, 68, 85
④ 49, 57, 68, 75, 85

정답 2번

출제 포인트

주어진 정수를 순서대로 이진 탐색 트리에 삽입한 뒤, 왼쪽 자식과 오른쪽 자식이 모두 없는 단말 노드를 찾는 문제이다.

올바른 순서

1. 44를 루트로 둔다.
2. 36은 44의 왼쪽에, 62는 44의 오른쪽에 삽입한다.
3. 왼쪽 부분 트리에는 3을 36의 왼쪽에, 16을 3의 오른쪽에 삽입한다.
4. 오른쪽 부분 트리에는 51을 62의 왼쪽에, 75를 62의 오른쪽에 삽입한다.
5. 68은 75의 왼쪽에, 49는 51의 왼쪽에, 85는 75의 오른쪽에, 57은 51의 오른쪽에 삽입한다.
6. 최종 트리에서 자식이 하나도 없는 노드를 고르면 16, 49, 57, 68, 85이다.

왜 이 순서: 새 값은 루트부터 각 노드와 비교하여 작은 경우 왼쪽, 큰 경우 오른쪽으로 이동한 뒤 처음 만나는 빈 자리에 삽입되므로, 앞서 삽입된 노드가 뒤에 삽입되는 노드의 경로를 결정한다.

NoteBase

선지별 해설

① X — 51은 왼쪽 자식 49와 오른쪽 자식 57을 가지므로 단말 노드가 아니며, 단말 노드 68이 빠져 있다. 틀린 표현인 '16, 49, 51, 57, 85'는 '16, 49, 57, 68, 85'로 고쳐야 한다.

② O — 16, 49, 57, 68, 85는 모두 자식이 없으며, 그 밖의 노드는 적어도 하나의 자식을 가진다.

③ X — 단말 노드 16이 빠져 있고, 51은 자식 49와 57이 있으므로 단말 노드가 아니다. 틀린 표현인 '49, 51, 57, 68, 85'는 '16, 49, 57, 68, 85'로 고쳐야 한다.

④ X — 단말 노드 16이 빠져 있고, 75는 왼쪽 자식 68과 오른쪽 자식 85를 가지므로 단말 노드가 아니다. 틀린 표현인 '49, 57, 68, 75, 85'는 '16, 49, 57, 68, 85'로 고쳐야 한다.

오답 포인트

51과 75는 삽입 직후에는 단말 노드이지만, 뒤에서 각각 49·57과 68·85가 연결되면서 내부 노드가 된다. 중간 상태가 아니라 모든 값의 삽입이 끝난 최종 상태에서 단말 노드를 판단해야 한다.

다시 나오면 이렇게 판단

각 값을 제시된 순서대로 삽입하고 부모에서 자식으로 이어지는 연결을 모두 표시한 다음, 최종적으로 나가는 연결이 하나도 없는 노드만 추린다.

암기 포인트

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

NoteBase

문 18. 다음 과정을 통해 수행되는 정렬 알고리즘의 특징으로 옳지 않은 것은?

초기값	15	9	8	1	4
1단계	9	15	8	1	4
2단계	8	9	15	1	4
3단계	1	8	9	15	4
4단계	1	4	8	9	15

① 최악의 경우에 시간 복잡도는 $O(n^2)$이다.
② 원소 수가 적거나 거의 정렬된 경우에 효과적이다.
③ 선택정렬(selection sort)에 비해 비교연산 횟수가 같거나 적다.
④ 정렬 대상의 크기만큼 추가 공간이 필요하다.

정답 4번

출제 포인트

주어진 단계가 삽입 정렬임을 파악하고, 삽입 정렬의 시간 복잡도·비교 횟수·추가 공간 특성을 구별하는 문제이다.

핵심 개념

삽입 정렬은 배열의 앞부분을 정렬된 영역으로 유지하면서 다음 원소를 알맞은 위치에 삽입하는 정렬 알고리즘이다. 표에서는 15 다음의 9를 앞에 삽입하고, 이어서 8, 1, 4를 차례로 정렬된 앞부분에 삽입한다. 최악의 시간 복잡도는 $O(n^2)$이고, 제자리 정렬이므로 추가 공간 복잡도는 $O(1)$이다.

선지별 해설

① O — 입력이 역순에 가까우면 각 원소를 앞의 원소들과 차례로 비교하고 이동해야 하므로 최악의 시간 복잡도는 $O(n^2)$이다.
② O — 원소 수가 적거나 거의 정렬된 배열에서는 삽입 위치를 찾기 위한 비교와 이동이 적어 효율적이다. 특히 이미 정렬된 경우 한 번의 순회 수준으로 처리된다.
③ O — 선택 정렬은 입력 상태와 관계없이 대체로 $n(n-1)/2$회의 비교를 하지만, 삽입 정렬은 앞부분이 정렬되어 있을수록 비교를 일찍 끝낼 수 있다. 따라서 비교 횟수는 선택 정렬과 같거나 더 적을 수 있다.
④ X — 틀린 표현은 '정렬 대상의 크기만큼 추가 공간이 필요하다'이다. 삽입 정렬은 배열 내부에서 원소를 이동시키고 삽입할 값 등을 저장할 소수의 변수만 사용하므로, 올바르게는 '정렬 대상의 크기와 무관하게 $O(1)$의 추가 공간이 필요하다'이다.

NoteBase

오답 포인트

표에서 원소들이 여러 칸 이동하므로 별도의 배열이 필요하다고 착각하기 쉽다. 그러나 원소의 이동은 기존 배열 안에서 이루어지며, 이동 횟수와 추가 공간의 크기는 서로 다른 개념이다.

다시 나오면 이렇게 판단

앞부분을 정렬된 상태로 두고 다음 원소를 그 안의 알맞은 위치에 끼워 넣는 과정이면 삽입 정렬로 판단한다. 그다음 '최악 $O(n^2)$, 거의 정렬된 자료에 유리, 제자리 정렬로 추가 공간 $O(1)$'을 기준으로 선지를 판별한다.

암기 포인트

삽입 정렬은 앞부분에 끼워 넣는 제자리 정렬이다: 최악 $O(n^2)$, 추가 공간 $O(1)$.

NoteBase

문 19. SET(Secure Electronic Transaction)에 대한 설명으로 옳지 않은 것은?

① 프라이버시 보호를 위해 이중서명 프로토콜을 사용한다.
② 카드 소지자는 전자 지갑 소프트웨어가 필요하다.
③ 인증기관(Certification Authority)이 필요하다.
④ SSL(Secure Socket Layer)에 비해 고속으로 동작한다.

정답 4번

출제 포인트

SET의 보안 구조와 동작 특성을 SSL과 비교하여 옳지 않은 설명을 찾는 문제이다.

핵심 개념

SET: 인터넷에서 신용카드 결제를 안전하게 처리하기 위해 카드 소지자·판매자·금융기관을 인증하고 결제 정보를 보호하는 전자상거래 보안 프로토콜이다. 이중서명, 전자 지갑, 인증기관을 활용하지만 처리 구조가 복잡하다.
SSL: 통신 당사자 사이의 전송 구간을 암호화하는 보안 프로토콜이다. SET보다 구조와 처리 절차가 단순하다.
핵심 대비: SET은 결제 참여자 인증과 결제 정보 보호에 특화되지만, SSL보다 복잡하여 처리 속도가 느리다.

선지별 해설

① O — SET은 이중서명을 사용하여 주문 정보와 결제 정보를 연결하면서도 각 정보가 불필요한 상대방에게 노출되지 않도록 보호한다.
② O — 카드 소지자는 결제에 필요한 인증서와 결제 정보를 관리하고 SET 거래를 수행하기 위해 전자 지갑 소프트웨어가 필요하다.
③ O — SET에서는 카드 소지자와 판매자 등 거래 참여자의 신원을 확인할 디지털 인증서를 발급·관리하는 인증기관이 필요하다.
④ X — 틀린 표현은 'SSL에 비해 고속으로 동작한다'이다. 올바르게는 SET은 인증, 이중서명, 암호화 등 처리 절차가 복잡하므로 SSL에 비해 일반적으로 느리게 동작한다.

오답 포인트

SET이 보안 기능을 더 세밀하게 제공한다는 사실을 처리 속도도 더 빠르다는 뜻으로 오해하기 쉽다. ④의 '고속'은 보안성과 성능을 뒤바꾼 표현이다.

NoteBase

다시 나오면 이렇게 판단

SET 관련 선지에서는 이중서명·전자 지갑·인증기관은 핵심 구성 요소로 판단한다. SSL과 속도를 비교하면, 추가 인증과 암호화 절차가 많은 SET이 더 복잡하고 느리다고 판단한다.

암기 포인트

SET은 이중서명·전자 지갑·인증기관을 사용하며, SSL보다 복잡하고 느리다.

NoteBase

문 20. 「개인정보 보호법」의 개인정보 보호 원칙으로 옳은 것의 총 개수는?

ㄱ. 개인정보처리자는 개인정보의 처리 목적에 필요한 범위에서 개인정보의 정확성, 완전성 및 최신성이 보장되도록 하여야 한다.

ㄴ. 개인정보처리자는 개인정보의 처리 목적에 필요한 범위에서 적합하게 개인정보를 처리하여야 하며, 그 목적 외의 용도로 활용 하고자 하는 경우 개인정보 보호책임자의 동의를 받아야 한다.

ㄷ. 개인정보처리자는 개인정보 처리방법 등 개인정보의 처리에 관한 사항은 비공개하여야 하며, 열람청구권 등 정보주체의 권리를 보장하여야 한다.

ㄹ. 개인정보처리자는 개인정보를 가명 또는 익명으로 처리하여도 개인정보 수집목적을 달성할 수 있는 경우 가명처리가 가능한 경우에는 가명에 의하여, 가명처리로 목적을 달성할 수 없는 경우에는 익명에 의하여 처리될 수 있도록 하여야 한다.

① 1개 ② 2개 ③ 3개 ④ 4개

정답 1번

쟁점

이 문제는 「개인정보 보호법」상 개인정보 보호 원칙에 관한 문제이다. 핵심은 목적 범위 내 처리, 처리 사항의 공개, 익명처리와 가명처리의 우선순위를 구별하여 옳은 문장의 개수를 세는 것이다.

용어 풀이

개인정보처리자 = 업무를 목적으로 개인정보파일을 운용하기 위하여 스스로 또는 다른 사람을 통하여 개인정보를 처리하는 자이다.
익명처리 = 더 이상 특정 개인을 알아볼 수 없도록 처리하는 것이고, 가명처리 = 추가 정보 없이는 특정 개인을 알아볼 수 없도록 처리하는 것이다.

NoteBase

기출OX 해설집 · 공무원/계리직/9급 · 컴퓨터일반 · 2021

핵심 구분

현행 「개인정보 보호법」 제3조에 따르면 ㄱ만 옳고 ㄴ, ㄷ, ㄹ은 틀리므로 옳은 것은 총 1개이다.
ㄱ O: 처리 목적에 필요한 범위에서 개인정보의 정확성, 완전성 및 최신성이 보장되도록 하여야 한다.
ㄴ X: 목적 외 이용은 원칙적으로 금지된다. 법률이 정한 예외 사유가 있는 경우에만 가능하며, 개인정보 보호책임자의 동의만으로 허용되는 것이 아니다.
ㄷ X: 개인정보 처리방침 등 개인정보의 처리에 관한 사항은 공개하여야 하고, 열람청구권 등 정보주체의 권리를 보장하여야 한다.
ㄹ X: 익명처리가 가능한 경우에는 익명으로 처리하고, 익명처리로 목적을 달성할 수 없는 경우에 가명으로 처리될 수 있도록 하여야 한다. 제시문은 익명과 가명의 순서를 반대로 서술하였다.

선지별 해설

① O
- ㄱ만 옳으므로 옳은 것의 총 개수는 1개이다.

② X
- 옳은 것은 ㄱ 하나뿐이므로 2개가 아니다.
- 틀린 표현: 2개
- 올바른 표현: 1개

③ X
- ㄴ은 목적 외 이용 원칙에 어긋나고, ㄷ은 공개 원칙을 반대로 서술하며, ㄹ은 익명처리와 가명처리의 우선순위를 뒤집었으므로 3개가 아니다.
- 틀린 표현: 3개
- 올바른 표현: 1개

④ X
- ㄴ, ㄷ, ㄹ이 틀리므로 네 문장이 모두 옳지 않다.
- 틀린 표현: 4개
- 올바른 표현: 1개

오답 포인트

ㄹ의 ‘가명이 먼저, 익명이 나중’이라는 표현이 가장 큰 함정이다. 법은 재식별 위험이 더 낮은 익명처리를 먼저 검토하고, 익명처리로 목적을 달성할 수 없을 때 가명처리를 택하도록 한다. 또한 ㄷ의 ‘비공개’는 법의 공개 원칙을 정반대로 바꾼 표현이다.

NoteBase

다시 나오면 이렇게 판단

각 문장을 먼저 O/X로 판정한 뒤 O의 개수를 센다. 개인정보 보호 원칙은 '목적 내 적합 처리,
정확성·완전성·최신성 보장, 처리 사항 공개와 정보주체 권리 보장, 익명 우선·가명 차선'의 순서로 확인한다.

암기 포인트

목적 안에서 정확·완전·최신, 처리 사항은 공개, 익명 우선·가명 차선이다.

NoteBase