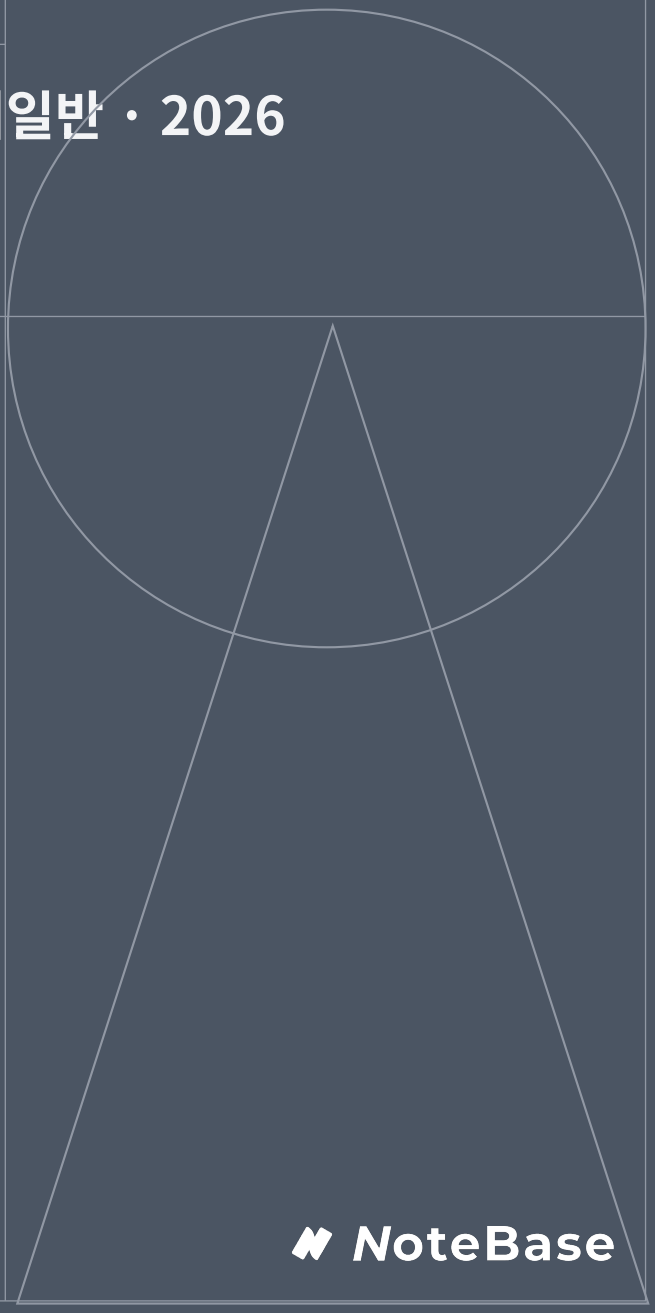
기출OX 해설집

공무원/군무원/9급 · 컴퓨터일반 · 2026



NoteBase

기출OX 해설집

공무원/군무원/9급 · 컴퓨터일반 · 2026

공무원/군무원/9급 · 컴퓨터일반 · 2026

1. 연결리스트(Linked List)에 대한 설명으로 가장 적절하지 않은 것은? (단, 연결리스트의 크기는 n이다.)
   ① 단순 연결리스트(Singly Linked List)의 특정 노드의 탐색은 포인터를 이용하여 탐색하므로 시간복잡도가 O(1)이다.
   ② 원형 연결리스트(Circular Linked List)에서 헤드 포인터가 마지막 노드를 가리킬 때, 첫 번째 노드 삽입과 삭제의 시간복잡도는 O(1)이다.
   ③ 이중 연결리스트(Doubly Linked List)에서 특정 노드의 포인터를 안다면 해당 노드 삭제의 시간복잡도는 O(1)이다.
   ④ 단순 연결리스트에서 위치를 알고 있는 특정 노드 삭제의 시간복잡도는 O(n)이다.

정답 1번

출제 포인트

연결 리스트의 종류와 주어진 포인터 조건에 따라 탐색·삽입·삭제의 시간복잡도가 어떻게 달라지는지를 묻는 문제이다.

NoteBase

핵심 개념

단순 연결 리스트는 각 노드가 다음 노드의 포인터만 가지는 구조이므로 특정 노드를 찾을 때 처음부터 순차적으로 탐색해야 한다.
원형 연결 리스트는 마지막 노드의 다음 포인터가 첫 번째 노드를 가리키는 구조이다. 마지막 노드를 가리키는 포인터가 있으면 첫 번째 노드는 그 다음 포인터로 즉시 접근할 수 있다.
이중 연결 리스트는 각 노드가 이전 노드와 다음 노드의 포인터를 모두 가지는 구조이므로 삭제할 노드의 포인터를 알면 양쪽 연결을 바로 수정할 수 있다.

선지별 해설

① X — 단순 연결 리스트에서 포인터를 따라 특정 노드를 찾으려면 최악의 경우 노드 n개를 차례로 확인해야 하므로 탐색의 시간복잡도는 $O(n)$이다. 틀린 표현은 '$O(1)$'이며, 올바르게는 '특정 노드의 탐색은 $O(n)$이다'이다. 따라서 가장 적절하지 않은 설명이다.
② O — 마지막 노드를 가리키는 포인터가 있으면 첫 번째 노드는 마지막 노드의 다음 포인터로 바로 찾는다. 첫 번째 노드의 삽입과 삭제는 일정한 수의 포인터만 바꾸므로 $O(1)$이다.
③ O — 이중 연결 리스트에서 삭제할 노드의 포인터를 알면 그 노드의 이전 노드와 다음 노드를 서로 연결하여 삭제할 수 있다. 탐색이 필요하지 않고 포인터 변경 횟수가 일정하므로 $O(1)$이다.
④ O — 단순 연결 리스트에서 특정 노드를 삭제하려면 그 노드를 가리키는 이전 노드의 포인터를 수정해야 한다. 대상 노드의 위치를 알아도 이전 노드를 찾기 위해 처음부터 순차 탐색할 수 있으므로 시간복잡도는 $O(n)$이다.

오답 포인트

①의 '포인터를 이용한다'는 말만 보고 상수 시간이라고 판단하기 쉽다. 연결 리스트가 포인터로 연결되어 있다는 사실은 임의의 특정 노드에 즉시 접근할 수 있다는 뜻이 아니며, 해당 노드의 포인터가 이미 주어진 경우와 그 포인터를 탐색해야 하는 경우를 구분해야 한다.

다시 나오면 이렇게 판단

먼저 목표 노드의 포인터가 이미 주어졌는지 확인한다. 포인터를 얻기 위한 순차 탐색이 필요하면 $O(n)$이고, 필요한 노드의 포인터가 주어져 있어 일정한 수의 연결만 바꾸면 $O(1)$이다. 단순 연결 리스트의 삭제에서는 이전 노드에 접근할 수 있는지도 함께 확인한다.

암기 포인트

연결 리스트는 탐색 $O(n)$, 포인터가 확보된 자리의 연결 변경은 $O(1)$이다. 단순 연결 리스트의 삭제는 이전 노드가 핵심이다.

NoteBase

2. 임계구역(Critical Section)에 대한 설명으로 가장 적절하지 않은 것은?
   ① 임계구역을 수행 중인 프로세스가 있을 때 다른 프로세스는 임계구역을 수행할 수 없어야 한다.
   ② 피터슨 알고리즘은 두 개의 프로세스 간 임계구역을 해결하기 위해 고안된 것이다.
   ③ 세마포어는 오직 두 개의 값만 갖는 변수로서 상호배제의 해결을 위해 고안된 것이다.
   ④ 임계구역 문제의 해결 방안인 모니터(Monitor)는 한 순간에 한 프로세스만 모니터에 진입하도록 보장한다.

정답 3번

출제 포인트

임계구역의 상호배제 원리와 피터슨 알고리즘, 세마포어, 모니터의 특징을 구별하는 문제이다.

핵심 개념

임계구역은 여러 프로세스가 공유 자원에 접근하는 코드 영역이며, 한 프로세스가 실행 중일 때 다른 프로세스가 동시에 들어가지 못하도록 상호배제를 보장해야 한다.
피터슨 알고리즘은 두 프로세스의 진입 의사와 실행 순서를 나타내는 변수를 이용하는 소프트웨어적 상호배제 기법이다.
세마포어는 정수 값과 대기·신호 연산을 이용해 상호배제와 프로세스 동기화를 구현하는 도구이다. 일반 세마포어는 여러 정수 값을 가질 수 있고, 0과 1만 사용하는 것은 이진 세마포어이다.
모니터는 공유 데이터와 연산을 하나로 묶고 한 번에 하나의 프로세스 또는 스레드만 내부에서 실행되도록 하는 고수준 동기화 구조이다.

NoteBase

선지별 해설

① O — 한 프로세스가 임계구역을 수행하는 동안 다른 프로세스의 임계구역 진입을 막는 것은 상호배제의 기본 원칙이다.
② O — 피터슨 알고리즘은 두 프로세스 사이의 임계구역 문제를 해결하기 위해 고안된 소프트웨어 알고리즘이다.
③ X — 틀린 표현은 '오직 두 개의 값만 갖는 변수'이다. 세마포어는 일반적으로 정수 값을 사용하는 동기화 도구이며, 0과 1의 두 값만 갖는 것은 세마포어 전체가 아니라 이진 세마포어이다. 올바르게는 '세마포어는 정수 값과 대기·신호 연산을 이용해 상호배제와 동기화를 구현하며, 이진 세마포어는 0과 1만 갖는다'이다.
④ O — 모니터는 내부의 공유 자원 접근을 캡슐화하고 한 순간에 하나의 프로세스 또는 스레드만 모니터 내부에서 실행되도록 상호배제를 보장한다.

오답 포인트

③은 이진 세마포어의 특징인 '두 값'을 세마포어 전체의 특징으로 일반화한 선지이다. 이진 세마포어와 여러 값을 가질 수 있는 계수 세마포어를 구분해야 한다.

다시 나오면 이렇게 판단

'0과 1만 사용'이라는 표현은 이진 세마포어에 한정하고, 세마포어 일반은 정수 값을 이용한 동기화 도구로 판단한다. 피터슨 알고리즘은 두 프로세스, 모니터는 내부에 한 번에 하나만 진입한다는 기준으로 구별한다.

암기 포인트

임계구역은 상호배제, 피터슨은 두 프로세스, 두 값만 쓰는 세마포어는 이진 세마포어, 모니터는 한 번에 하나이다.

NoteBase

3. 관계 데이터베이스에서 제1정규형(1NF)을 만족하기 위한 조건으로 가장 적절한 것은?
① 기본키가 반드시 2개 이상이어야 한다.
② 하나의 속성에는 하나의 값만 저장되어야 한다.
③ 이행 함수 종속을 제거해야 한다.
④ 부분 함수 종속을 제거해야 한다.

정답 2번

출제 포인트

관계 데이터베이스의 제1정규형(1NF)이 요구하는 원자값 조건과 제2·제3정규형의 종속성 제거 조건을 구분하는 문제이다.

핵심 개념

제1정규형(1NF)은 릴레이션의 각 속성에 더 이상 나눌 수 없는 하나의 원자값만 저장되도록 한 상태이다. 제2정규형(2NF)은 부분 함수 종속을 제거한 상태이고, 제3정규형(3NF)은 이행 함수 종속을 제거한 상태이다.

선지별 해설

① X — '기본키가 반드시 2개 이상'이라는 표현이 틀렸다. 기본키는 후보키 중 하나를 선택한 것이며, 하나의 속성으로 구성될 수도 있고 둘 이상의 속성을 결합한 복합키일 수도 있다. 올바르게는 '기본키의 속성 수와 관계없이 각 속성값이 원자값이어야 한다'이다.
② O — 1NF에서는 한 속성의 한 셀에 반복 집합이나 여러 값을 넣지 않고 하나의 원자값만 저장해야 한다.
③ X — '이행 함수 종속을 제거'하는 것은 1NF가 아니라 3NF의 조건이다. 올바르게는 '2NF를 만족하면서 기본키가 아닌 속성 사이의 이행 함수 종속을 제거해야 3NF를 만족한다'이다.
④ X — '부분 함수 종속을 제거'하는 것은 1NF가 아니라 2NF의 조건이다. 올바르게는 '1NF를 만족하고 기본키가 아닌 속성이 기본키 전체에 완전 함수 종속되도록 부분 함수 종속을 제거해야 2NF를 만족한다'이다.

오답 포인트

③의 이행 함수 종속과 ④의 부분 함수 종속은 정규화 단계가 뒤바뀌기 쉬운 표현이다. 1NF는 값의 원자성, 2NF는 부분 함수 종속 제거, 3NF는 이행 함수 종속 제거에 대응한다.

다시 나오면 이렇게 판단

선지의 핵심어를 정규형과 연결한다. '한 칸에 하나의 값·원자값'이면 1NF, '부분 함수 종속 제거'이면 2NF, '이행 함수 종속 제거'이면 3NF로 판단한다.

NoteBase

암기 포인트

1NF는 원자값, 2NF는 부분 종속 제거, 3NF는 이행 종속 제거이다.

NoteBase

4. SQL 명령 중에서 데이터를 조회할 때 사용하는 명령으로 가장 적절한 것은?
① INSERT
② UPDATE
③ DELETE
④ SELECT

정답 4번

출제 포인트

SQL 명령이 데이터를 추가·수정·삭제·조회하는 기능 중 어디에 해당하는지를 구분하는 문제이다.

핵심 개념

SELECT는 데이터베이스에서 조건에 맞는 데이터를 조회하는 명령이다. INSERT는 새 데이터를 추가하고, UPDATE는 기존 데이터를 수정하며, DELETE는 기존 데이터를 삭제한다.

선지별 해설

① X
INSERT는 테이블에 새 데이터를 추가하는 명령이다. '데이터를 조회한다'는 표현이 틀렸으며, 올바르게는 '새 데이터를 삽입한다'이다.

② X
UPDATE는 테이블에 이미 저장된 데이터를 수정하는 명령이다. '데이터를 조회한다'는 표현이 틀렸으며, 올바르게는 '기존 데이터를 갱신한다'이다.

③ X
DELETE는 테이블에서 데이터를 삭제하는 명령이다. '데이터를 조회한다'는 표현이 틀렸으며, 올바르게는 '기존 데이터를 삭제한다'이다.

④ O
SELECT는 테이블에서 원하는 열과 행의 데이터를 검색하여 조회하는 명령이므로 문제의 조건에 맞는다.

오답 포인트

INSERT, UPDATE, DELETE, SELECT가 모두 데이터에 사용하는 SQL 명령이라는 점만 보고 기능을 혼동하기 쉽다. 이 문항의 핵심 표현인 '조회'는 SELECT와 연결된다.

NoteBase

다시 나오면 이렇게 판단

SQL 명령은 동작을 나타내는 핵심어로 판단한다. 추가는 INSERT, 수정은 UPDATE, 삭제는 DELETE, 조회는 SELECT이다.

암기 포인트

조회 SELECT, 추가 INSERT, 수정 UPDATE, 삭제 DELETE이다.

NoteBase

5. 다음과 같은 데이터가 있을 때 이진 탐색을 수행하면 처음 비교되는 값은 무엇인가?

10, 20, 30, 40, 50, 60, 70

① 10
② 20
③ 40
④ 70

정답 3번

출제 포인트

이 문제는 오름차순으로 정렬된 데이터에서 이진 탐색을 시작할 때 처음 비교하는 중앙값을 찾는 문제이다.

조건 정리

주어진 값: 10, 20, 30, 40, 50, 60, 70
구해야 할 값: 이진 탐색에서 처음 비교되는 값
주의할 조건: 데이터가 정렬되어 있고 원소가 7개이므로 가운데인 네 번째 원소를 먼저 비교한다.

사용할 공식

중앙 위치는 $mid = \lfloor (low + high)/2 \rfloor$로 정한다.
이 공식을 쓰는 이유: 이진 탐색은 현재 탐색 범위의 중앙 원소와 찾을 값을 비교한 뒤 탐색 범위를 절반으로 줄이는 방식이기 때문이다.

계산 과정

1. 원소는 모두 7개이다.
2. 1부터 시작하는 위치를 기준으로 중앙 위치는 $(1 + 7)/2 = 4$이다.
3. 네 번째 원소는 40이므로 처음 비교되는 값은 40이다.

선지 판단

① X — 10은 첫 번째 원소이므로 처음 비교하는 중앙값이 아니다. 틀린 값 10을 올바르게 고치면 40이다.
② X — 20은 두 번째 원소이므로 처음 비교하는 중앙값이 아니다. 틀린 값 20을 올바르게 고치면 40이다.
③ O — 40은 7개 원소 중 네 번째에 있는 중앙 원소이므로 처음 비교되는 값이다.
④ X — 70은 마지막 원소이므로 처음 비교하는 중앙값이 아니다. 틀린 값 70을 올바르게 고치면 40이다.
계산 결과 40이므로 정답은 3번이다.

NoteBase

오답 포인트

이진 탐색을 순차 탐색처럼 생각하여 첫 원소 10부터 비교한다고 판단하거나, 범위의 끝값 70을 먼저 비교한다고 판단하기 쉽다. 이진 탐색은 정렬된 데이터의 중앙 원소부터 비교한다.

다시 나오면 이렇게 판단

정렬 여부를 확인한 뒤 현재 탐색 범위의 중앙 위치를 구하고, 그 위치의 원소를 첫 비교값으로 판단한다.

암기 포인트

이진 탐색은 정렬된 데이터의 가운데 값부터 비교한다.

NoteBase

6. 해싱(Hashing)에서 발생하는 충돌(Collision)에 대한 설명으로 가장 적절한 것은?
① 데이터가 자동으로 삭제되는 현상이다.
② 해시 함수가 실행되지 않는 오류이다.
③ 저장된 데이터가 정렬되는 과정이다.
④ 서로 다른 키가 동일한 해시 주소를 가지는 현상이다.

정답 4번

출제 포인트

해싱에서 충돌이 무엇을 뜻하는지 정확한 정의를 묻는 문제이다.

핵심 개념

해싱은 해시 함수를 이용해 키를 저장 위치에 대응시키는 방법이다.
해시 충돌은 서로 다른 키가 해시 함수에 의해 동일한 해시 주소로 대응되는 현상이다. 충돌은 데이터 삭제, 함수 실행 실패, 데이터 정렬을 의미하지 않는다.

선지별 해설

① X
틀린 표현은 '데이터가 자동으로 삭제되는 현상'이다. 충돌이 발생했다고 해서 데이터가 자동으로 삭제되는 것은 아니며, 올바르게는 서로 다른 키가 동일한 해시 주소에 대응되는 현상이다.

② X
틀린 표현은 '해시 함수가 실행되지 않는 오류'이다. 충돌은 해시 함수가 정상적으로 실행된 결과에서도 발생할 수 있으며, 올바르게는 서로 다른 키의 해시 결과가 같은 주소가 되는 현상이다.

③ X
틀린 표현은 '저장된 데이터가 정렬되는 과정'이다. 해싱은 키를 저장 위치에 대응시키는 방식이며 충돌은 정렬 과정이 아니다. 올바르게는 서로 다른 키가 동일한 해시 주소를 갖는 현상이다.

④ O
서로 다른 키가 동일한 해시 주소를 가지는 현상은 해시 충돌의 정확한 정의이다.

NoteBase

오답 포인트

①의 삭제, ②의 함수 실행 오류, ③의 정렬은 해시 충돌과 관계없는 현상이다. 충돌은 해시 함수의 작동 여부가 아니라 서로 다른 키의 결과 주소가 같은지로 판단한다.

다시 나오면 이렇게 판단

선지에서 '서로 다른 키'와 '동일한 해시 주소'가 함께 제시되는지 확인한다. 두 조건이 모두 있으면 해시 충돌을 설명한 것이다.

암기 포인트

서로 다른 키, 같은 해시 주소이면 충돌이다.

NoteBase

7. 기계학습 방법 중에서 지도학습(Supervised Learning)에 대한 설명으로 가장 적절한 것은?
① 정답 데이터 없이 데이터의 숨겨진 패턴을 스스로 찾는 학습 방식이다.
② 보상과 벌을 기반으로 최적의 행동을 학습하는 방식이다.
③ 입력 데이터와 정답 데이터를 함께 학습하여 결과를 예측하는 방식이다.
④ 학습 데이터를 사용하지 않고 규칙을 이용하여 학습하는 방식이다.

정답 3번

출제 포인트

지도학습의 핵심인 입력 데이터와 정답 데이터의 동시 사용을 비지도학습·강화학습·규칙 기반 방식과 구별하는 문제이다.

핵심 개념

지도학습: 입력 데이터와 그에 대응하는 정답(레이블)을 함께 학습하여 새로운 입력의 결과를 예측하는 기계학습 방식이다.
비지도학습: 정답 레이블 없이 데이터 자체의 구조나 숨겨진 패턴을 찾는 방식이다.
강화학습: 에이전트가 환경과 상호작용하며 받은 보상과 벌을 바탕으로 더 나은 행동 전략을 학습하는 방식이다.
규칙 기반 방식: 사람이 미리 정한 규칙에 따라 결과를 도출하는 방식으로, 데이터에서 모델을 학습하는 지도학습과 다르다.

선지별 해설

① X — '정답 데이터 없이 데이터의 숨겨진 패턴을 스스로 찾는' 것은 비지도학습의 설명이다. 올바르게는 지도학습은 입력 데이터와 정답 데이터를 함께 사용하여 입력과 정답 사이의 관계를 학습하는 방식이다.
② X — '보상과 벌을 기반으로 최적의 행동을 학습하는' 것은 강화학습의 설명이다. 올바르게는 지도학습은 보상 신호가 아니라 입력별 정답 레이블을 이용해 예측 규칙을 학습하는 방식이다.
③ O — 입력 데이터와 정답 데이터를 함께 학습하여 새로운 입력의 결과를 예측한다는 설명은 지도학습의 정의에 해당한다.
④ X — '학습 데이터를 사용하지 않고 규칙을 이용하여 학습하는' 것은 지도학습의 설명이 아니다. 올바르게는 지도학습은 정답이 표시된 학습 데이터를 사용하며, 미리 정한 규칙을 적용하는 것은 규칙 기반 처리 방식이다.

NoteBase

오답 포인트

①의 '정답 데이터 없이'는 비지도학습, ②의 '보상과 벌'은 강화학습을 가리킨다. 학습이라는 공통 표현만 보고 고르지 않고, 정답 레이블·패턴 탐색·보상 중 어떤 정보가 학습 기준인지 구별해야 한다.

다시 나오면 이렇게 판단

정답이 붙은 입력-정답 쌍을 사용하면 지도학습, 정답 없이 데이터 구조를 찾으면 비지도학습, 행동 결과의 보상을 기준으로 전략을 개선하면 강화학습으로 판단한다.

암기 포인트

지도학습은 정답으로 배우고, 비지도학습은 패턴을 찾으며, 강화학습은 보상으로 배운다.

NoteBase

8. 제어 구성요소에서 장치와 역할의 조합으로 가장 적절한 것은?
① 프로그램 카운터 – 다음 실행할 명령의 주소 번지를 저장
② 명령어 레지스터 – 명령 실행에서 접근하는 주메모리 데이터값을 저장
③ 인덱스 레지스터 – 실행 명령을 저장
④ 주소 레지스터 – 명령에 따라 연산한 결과값을 저장

정답 1번

출제 포인트

CPU의 제어 관련 레지스터가 명령어 주소, 현재 명령어, 기억장치 데이터·주소, 연산 결과 중 무엇을 저장하는지 구별하는 문제이다.

핵심 개념

프로그램 카운터(PC)는 다음에 실행할 명령어의 주소를 저장한다. 명령어 레지스터(IR)는 현재 해독·실행 중인 명령어를 저장한다. 기억장치 데이터 레지스터(MDR)는 주기억장치에서 읽거나 주기억장치에 쓸 데이터를 저장하고, 기억장치 주소 레지스터(MAR)는 접근할 주기억장치의 주소를 저장한다. 인덱스 레지스터는 유효주소 계산에 사용하는 인덱스값을 저장하며, 누산기는 산술·논리 연산의 중간값이나 결과값을 저장한다.

선지별 해설

① O — 프로그램 카운터는 명령어 실행 순서를 제어하기 위해 다음에 실행할 명령어의 주소를 저장하므로 장치와 역할의 연결이 적절하다.
② X — 틀린 표현은 '명령어 레지스터가 명령 실행에서 접근하는 주메모리 데이터값을 저장한다'이다. 명령어 레지스터는 현재 해독·실행 중인 명령어를 저장하며, 주기억장치에서 읽거나 쓸 데이터는 기억장치 데이터 레지스터(MDR)에 저장한다.
③ X — 틀린 표현은 '인덱스 레지스터가 실행 명령을 저장한다'이다. 실행할 현재 명령어는 명령어 레지스터(IR)에 저장하며, 인덱스 레지스터는 유효주소를 계산할 때 더하는 인덱스값을 저장한다.
④ X — 틀린 표현은 '주소 레지스터가 연산 결과값을 저장한다'이다. 기억장치 주소 레지스터(MAR)는 접근할 주기억장치의 주소를 저장하며, 연산의 중간값이나 결과값은 주로 누산기에 저장한다.

NoteBase

오답 포인트

②는 명령어 레지스터와 기억장치 데이터 레지스터를, ③은 인덱스 레지스터와 명령어 레지스터를, ④는 주소 레지스터와 누산기를 서로 바꾸어 놓은 선지이다. 특히 '명령어', '데이터', '주소', '연산 결과'라는 저장 대상에 주목해야 한다.

다시 나오면 이렇게 판단

저장 대상부터 확인한다. 다음 명령어의 주소이면 PC, 현재 명령어이면 IR, 주기억장치와 주고받는 데이터이면 MDR, 접근할 기억장치 주소이면 MAR, 주소 계산용 인덱스값이면 인덱스 레지스터, 연산 결과이면 누산기로 판단한다.

암기 포인트

PC는 다음 주소, IR은 현재 명령어, MDR은 데이터, MAR은 기억장치 주소, 인덱스 레지스터는 주소 계산, 누산기는 연산 결과이다.

NoteBase

9. 소프트웨어 형상관리에 대한 설명으로 가장 적절하지 않은 것은?

① 소프트웨어 개발과정에서 변경 사항을 체계적으로 관리하기 위한 활동이다.

② 형상 항목에 소스 코드, 설계 문서, 테스트 문서 등이 포함될 수 있다.

③ 버전 관리는 형상관리의 중요한 기능 중 하나로 변경 이력을 추적할 수 있다.

④ 형상관리는 프로젝트 일정 관리를 목적으로 하며 소프트웨어 품질과는 관련이 없다.

정답 4번

출제 포인트

소프트웨어 형상관리의 목적, 관리 대상인 형상 항목, 버전 관리의 기능을 구분하는 문제이다. 문제는 '가장 적절하지 않은 것'을 묻는다.

핵심 개념

소프트웨어 형상관리는 소프트웨어 생명주기 동안 소스 코드와 각종 문서 등의 형상 항목을 식별하고, 변경을 통제하며, 변경 상태와 이력을 추적하여 산출물의 일관성과 무결성을 유지하는 활동이다.
형상 항목은 변경을 식별·통제해야 하는 관리 단위이며 소스 코드, 설계 문서, 테스트 문서 등이 포함될 수 있다.
버전 관리는 형상관리의 핵심 기능으로, 형상 항목의 버전과 변경 이력을 기록하고 필요한 버전을 식별·복원할 수 있게 한다.

선지별 해설

① O — 형상관리는 소프트웨어 개발 과정에서 발생하는 변경 사항을 식별하고 체계적으로 통제·추적하기 위한 활동이므로 적절하다.
② O — 소스 코드뿐 아니라 설계 문서와 테스트 문서처럼 변경 관리가 필요한 산출물도 형상 항목에 포함될 수 있으므로 적절하다.
③ O — 버전 관리는 형상 항목의 버전과 변경 이력을 기록·추적하는 형상관리의 중요한 기능이므로 적절하다.
④ X — 틀린 표현은 '프로젝트 일정 관리를 목적으로 하며 소프트웨어 품질과는 관련이 없다'이다. 형상관리의 주된 목적은 형상 항목의 변경을 통제하고 산출물의 일관성·무결성을 유지하는 것이며, 이는 오류와 불일치를 줄여 소프트웨어 품질 보증과 밀접하게 관련된다. 올바르게는 '형상관리는 변경을 체계적으로 통제하여 소프트웨어 산출물의 일관성과 무결성을 유지하고 품질 보증을 지원한다'이다.

NoteBase

오답 포인트

④의 '일정 관리'와 '품질과는 관련이 없다'가 함정이다. 일정 관리는 프로젝트 관리의 주요 영역이고, 형상관리는 변경 통제와 산출물의 무결성 유지가 중심이므로 품질과 직접 관련된다.

다시 나오면 이렇게 판단

선지의 핵심어를 확인한다. '형상 항목 식별·변경 통제·상태 기록·변경 이력·버전 관리·무결성 유지'는 형상관리와 연결하고, '일정 관리만을 목적으로 한다'거나 '품질과 무관하다'는 표현은 틀린 것으로 판단한다.

암기 포인트

형상관리는 변경을 통제하고 이력을 추적하여 산출물의 일관성·무결성과 품질을 지킨다.

NoteBase

10. OSI 7계층 중에서 네트워크 계층의 역할로 가장 적절한 것은?
① 사용자와 응용 프로그램에 네트워크 서비스 제공
② 최적의 경로를 선택하여 패킷을 전달
③ 데이터 암호화와 압축 수행
④ 물리적인 케이블 연결 및 전기 신호 처리

정답 2번

출제 포인트

OSI 7계층 가운데 네트워크 계층의 역할을 다른 계층의 역할과 구별하는 문제이다.

핵심 개념

네트워크 계층은 서로 다른 네트워크 사이에서 패킷이 목적지까지 가도록 논리 주소를 바탕으로 경로를 선택하고 전달하는 계층이다.
응용 계층은 사용자와 응용 프로그램에 네트워크 서비스를 제공하고, 표현 계층은 데이터의 형식 변환·암호화·압축을 담당하며, 물리 계층은 케이블과 전기·광 신호를 통해 비트를 전송한다.

선지별 해설

① X — '사용자와 응용 프로그램에 네트워크 서비스 제공'은 응용 계층의 역할이다. 올바르게는 응용 계층이 사용자와 응용 프로그램에 네트워크 서비스를 제공한다.
② O — 네트워크 계층은 라우팅을 통해 최적의 경로를 선택하고 패킷을 목적지까지 전달한다.
③ X — '데이터 암호화와 압축 수행'은 표현 계층의 역할이다. 올바르게는 표현 계층이 데이터의 암호화와 압축을 수행한다.
④ X — '물리적인 케이블 연결 및 전기 신호 처리'는 물리 계층의 역할이다. 올바르게는 물리 계층이 전송 매체의 연결과 전기·광 신호를 이용한 비트 전송을 담당한다.

오답 포인트

OSI 계층의 기능을 모두 통신 기능으로 뭉뚱그리면 ①·③·④도 네트워크 계층의 역할처럼 보일 수 있다. '경로 선택', '라우팅', '패킷 전달'이 네트워크 계층을 가리키는 결정적 표현이다.

다시 나오면 이렇게 판단

선지의 핵심 동사를 확인한다. 사용자 서비스는 응용 계층, 암호화·압축은 표현 계층, 경로 선택·패킷 전달은 네트워크 계층, 케이블·신호·비트 전송은 물리 계층으로 판단한다.

NoteBase

암기 포인트

응용은 서비스, 표현은 변환·암호화·압축, 네트워크는 경로·패킷, 물리는 신호·비트이다.

NoteBase

11. TCP 프로토콜의 헤더에 포함되는 정보에 대한 설명으로 가장 적절하지 않은 것은?

① 순서 번호(Sequence Number)는 데이터의 전송 순서를 확인하기 위해 사용된다.

② 확인 응답 번호(Acknowledgement Number)는 수신을 확인한 다음에 받을 데이터의 번호를 나타낸다.

③ 목적지 MAC 주소(Destination MAC Address)는 TCP 헤더에 포함되어 수신 장치를 식별한다.

④ 플래그(Flag) 비트에는 연결 설정 및 종료를 위한 SYN, ACK, FIN 등의 정보가 포함된다.

정답 3번

출제 포인트

TCP 헤더에 포함되는 전송 제어 정보와 데이터 링크 계층의 주소 정보를 구분하는 문제이다.

핵심 개념

TCP 헤더는 순서 번호, 확인 응답 번호, 플래그 등 신뢰성 있는 전송과 연결 제어에 필요한 정보를 담는다. 반면 MAC 주소는 같은 네트워크 구간에서 프레임을 전달하기 위한 데이터 링크 계층의 주소이며 이더넷 프레임 헤더에 들어간다.

선지별 해설

① O — 순서 번호는 TCP 세그먼트에 실린 데이터의 시작 바이트 번호를 나타내므로 수신 측이 데이터의 순서를 맞추고 누락 여부를 판단하는 데 사용된다.

② O — 확인 응답 번호는 수신 측이 다음에 받기를 기대하는 바이트의 순서 번호를 나타낸다. 따라서 그 번호 직전까지의 데이터가 정상적으로 수신되었음을 알린다.

③ X — 틀린 부분은 '목적지 MAC 주소는 TCP 헤더에 포함된다'이다. 올바르게는 목적지 MAC 주소가 데이터 링크 계층의 이더넷 프레임 헤더에 포함되어 다음 전달 구간의 수신 장치를 식별한다. TCP 헤더에는 출발지·목적지 포트 번호가 포함된다.

④ O — TCP 헤더의 플래그 비트에는 연결 설정의 SYN, 응답 확인의 ACK, 연결 종료의 FIN 등이 포함되어 연결 상태를 제어한다.

NoteBase

오답 포인트

③은 '목적지 주소'라는 표현 때문에 TCP의 목적지 포트 번호와 MAC 주소를 혼동하게 만든다. MAC 주소는 프레임 헤더, 포트 번호는 TCP 헤더에 포함된다는 계층 구분이 핵심이다.

다시 나오면 이렇게 판단

TCP 헤더 항목은 포트 번호·순서 번호·확인 응답 번호·플래그로 판단하고, MAC 주소가 나오면 데이터 링크 계층의 프레임 헤더 항목으로 판단한다.

암기 포인트

TCP는 포트·순서·확인·플래그, 이더넷 프레임은 MAC 주소이다.

NoteBase

12. 다음 C 프로그램의 출력 결과는?

```
#include <stdio.h>
int main()
{
    int data[5] = {10, 20, 30, 40, 50};
    int *ptr = (data + 1);

    *(ptr + 1) = 100;
    printf("%d, %d\n", data[1], data[2]);
}
```

① 20, 20
② 20, 100
③ 100, 20
④ 100, 100

정답 2번

출제 포인트

배열 이름과 포인터의 관계를 이용하여 포인터 연산이 실제로 변경하는 배열 원소를 추적하는 문제이다.

조건 정리

초기 배열은 data[0]부터 차례로 10, 20, 30, 40, 50이다.
ptr = data + 1이므로 ptr은 data[1]을 가리킨다.
printf는 data[1]과 data[2]를 이 순서대로 출력한다.

사용할 원리

C에서 data는 식에서 첫 원소의 주소처럼 사용되므로 data + 1은 &data[1]이다. 포인터에 1을 더하면 다음 int 원소로 이동하므로 ptr + 1은 &data[2]이고, *(ptr + 1)은 data[2]와 같은 원소이다.

실행 과정

1. ptr에 data + 1을 대입하므로 ptr은 data[1]을 가리킨다.
2. ptr + 1은 data[2]의 주소가 된다.
3. *(ptr + 1) = 100은 data[2]의 값을 30에서 100으로 변경한다.
4. data[1]은 변경되지 않아 20이고 data[2]는 100이므로 출력은 20, 100이다.

NoteBase

선지 판단

① X — ‘20, 20’에서 두 번째 20이 틀리다. data[2]는 100으로 변경되므로 올바른 출력은 20, 100이다.
② O — data[1]은 20으로 유지되고 data[2]는 100으로 변경되므로 20, 100이 출력된다.
③ X — ‘100, 20’은 두 값이 모두 틀리다. 대입 대상은 data[1]이 아니라 data[2]이고 출력 순서도 data[1], data[2]이므로 올바른 출력은 20, 100이다.
④ X — ‘100, 100’에서 첫 번째 100이 틀리다. data[1]은 변경되지 않아 20이므로 올바른 출력은 20, 100이다.

오답 포인트

ptr이 data[1]을 가리킨다는 사실만 보고 *(ptr + 1)도 data[1]이라고 착각하기 쉽다. 역참조하기 전에 ptr + 1 연산이 한 원소 앞으로 이동하여 data[2]를 가리킨다는 점이 핵심 함정이다.

다시 나오면 이렇게 판단

포인터 대입식에서 기준 원소를 먼저 찾고, 포인터 덧셈만큼 배열 첨자를 이동한 뒤, 역참조 대입으로 바뀐 원소와 printf의 출력 순서를 차례로 추적한다.

암기 포인트

ptr = data + k이면 ptr은 data[k]를 가리키고, *(ptr + n)은 data[k + n]과 같다.

NoteBase

13. 정렬 알고리즘은 시간적 복잡도와 공간적 복잡도에서 각각 장단점이 있다. 알고리즘의 성능을 비교한 설명으로 가장 적절하지 않은 것은?
① 버블 정렬에 비해 병합 정렬의 평균 성능이 좋다.
② 삽입 정렬에 비해 힙 정렬의 평균 성능이 좋다.
③ 병합 정렬에 비해 퀵 정렬의 최악 성능이 좋다.
④ 선택 정렬에 비해 힙 정렬의 최악 성능이 좋다.

정답 3번

출제 포인트

정렬 알고리즘의 평균 시간 복잡도와 최악 시간 복잡도를 비교하여 잘못된 설명을 찾는 문제이다.

핵심 개념

버블·삽입·선택 정렬은 평균 시간 복잡도가 모두 $O(n^2)$이고, 병합·힙 정렬은 평균과 최악 모두 $O(n\log n)$이다.
퀵 정렬은 평균적으로 $O(n\log n)$이지만 피벗 분할이 계속 한쪽으로 치우치면 최악의 경우 $O(n^2)$이 된다.
공간 측면에서는 배열 기준 병합 정렬이 보통 $O(n)$의 보조 공간을 사용하고, 힙·삽입·선택·버블 정렬은 제자리 정렬로 구현할 수 있다. 퀵 정렬은 재귀 호출 스택이 필요하다.

선지별 해설

① O — 버블 정렬의 평균 시간 복잡도는 $O(n^2)$, 병합 정렬의 평균 시간 복잡도는 $O(n\log n)$이므로 병합 정렬의 평균 성능이 더 좋다.
② O — 삽입 정렬의 평균 시간 복잡도는 $O(n^2)$, 힙 정렬의 평균 시간 복잡도는 $O(n\log n)$이므로 힙 정렬의 평균 성능이 더 좋다.
③ X — 틀린 표현은 '퀵 정렬의 최악 성능이 좋다'이다. 퀵 정렬의 최악 시간 복잡도는 $O(n^2)$인 반면 병합 정렬의 최악 시간 복잡도는 $O(n\log n)$이다. 올바르게는 '퀵 정렬에 비해 병합 정렬의 최악 성능이 좋다'이다.
④ O — 선택 정렬의 최악 시간 복잡도는 $O(n^2)$, 힙 정렬의 최악 시간 복잡도는 $O(n\log n)$이므로 힙 정렬의 최악 성능이 더 좋다.

NoteBase

오답 포인트

퀵 정렬이 평균적으로 빠르다는 사실을 최악의 경우에도 그대로 적용하면 ③을 놓치기 쉽다. 퀵 정렬은 불균형 분할이 반복되면 $O(n^2)$까지 느려지지만, 병합 정렬은 입력 상태와 관계없이 최악에도 $O(n \log n)$을 유지한다.

다시 나오면 이렇게 판단

먼저 선지가 평균 성능과 최악 성능 중 무엇을 비교하는지 확인한다. 평균은 버블·삽입·선택의 $O(n^2)$과 병합·힙·퀵의 $O(n \log n)$을 비교하고, 최악은 퀵 정렬이 $O(n^2)$으로 내려간다는 예외를 반영한다.

암기 포인트

평균은 병합·힙·퀵이 $O(n \log n)$, 최악까지 $O(n \log n)$을 지키는 것은 병합·힙이다. 퀵의 최악은 $O(n^2)$이다.

NoteBase

14. 운영체제의 메모리 관리에 대한 설명으로 가장 적절하지 않은 것은?

① 페이징에서의 사용자 프로세스가 실행되는 주소는 가상 주소 공간이다.

② 가상 주소 공간의 페이지는 물리 주소 공간의 프레임으로 대응된다.

③ 페이지 테이블은 가상 주소 공간과 물리 주소 공간을 변환하는 역할을 한다.

④ 하나의 물리 주소 프레임은 여러 가상 주소 페이지에 대응될 수 없다.

정답 4번

출제 포인트

페이징 방식에서 가상 주소의 페이지가 물리 주소의 프레임으로 변환되는 관계와 페이지 테이블의 역할을 묻는 문제이다.

핵심 개념

가상 주소: 사용자 프로세스가 실행되면서 생성하는 논리적 주소이다.
페이지와 프레임: 가상 주소 공간은 페이지로, 물리 메모리는 같은 크기의 프레임으로 나뉜다.
페이지 테이블: 가상 페이지 번호에 대응하는 물리 프레임 번호를 저장하여 주소 변환을 지원한다. 하나의 가상 페이지는 특정 시점에 하나의 물리 프레임에 매핑되지만, 공유 메모리나 별칭 매핑에서는 여러 가상 페이지가 하나의 물리 프레임을 함께 가리킬 수 있다.

선지별 해설

① O — 사용자 프로세스가 명령어를 실행하면서 사용하는 주소는 가상 주소이며, 실제 물리 주소로 변환된 뒤 메모리에 접근한다.
② O — 가상 주소 공간의 고정 크기 블록인 페이지는 물리 주소 공간의 같은 크기 블록인 프레임에 대응된다.
③ O — 페이지 테이블은 가상 페이지 번호와 물리 프레임 번호의 대응 정보를 보관하여 가상 주소를 물리 주소로 변환하는 데 사용된다.
④ X — 틀린 표현은 '여러 가상 주소 페이지에 대응될 수 없다'이다. 올바르게는 공유 메모리나 별칭 매핑을 위해 하나의 물리 프레임이 여러 가상 페이지에 대응될 수 있다.

NoteBase

오답 포인트

‘페이지 하나가 한 프레임에 매핑된다’는 기본 관계를 역으로도 일대일 관계라고 생각하기 쉽다. 그러나 여러 페이지 테이블 엔트리가 동일한 프레임 번호를 가리킬 수 있으므로 ④의 ‘대응될 수 없다’가 틀린 표현이다.

다시 나오면 이렇게 판단

주소 변환 문제에서는 가상 주소를 페이지 번호와 변위로, 물리 주소를 프레임 번호와 변위로 구분한다. 페이지 테이블은 페이지 번호를 프레임 번호로 바꾸며, 여러 엔트리가 같은 프레임을 가리키는 공유 매핑도 가능하다는 점으로 선지를 판단한다.

암기 포인트

가상 공간은 페이지, 물리 메모리는 프레임, 변환 정보는 페이지 테이블이며 한 프레임은 여러 페이지가 공유할 수 있다.

NoteBase

15. SRAM과 DRAM에 대한 설명으로 가장 적절한 것은?

① 일반적으로 DRAM은 SRAM에 비하여 고속으로 접근 가능하다.

② 일반적으로 DRAM은 SRAM에 비하여 전력 소모가 크다.

③ DRAM은 콘덴서, SRAM은 플립플롭을 비트 저장 방법으로 사용한다.

④ 일반적으로 SRAM은 DRAM과 달리 Refresh를 필요로 한다.

정답 3번

출제 포인트

이 문제는 SRAM과 DRAM의 저장 소자, 접근 속도, 소비 전력, Refresh 필요 여부를 비교하는 문제이다.

핵심 개념

SRAM은 플립플롭에 데이터를 저장하며 전원이 공급되는 동안 별도의 Refresh 없이 값을 유지하는 휘발성 메모리이다. 속도가 빠르지만 회로가 복잡하고 집적도가 낮아 주로 캐시 메모리에 사용된다.
DRAM은 콘덴서의 전하 유무로 데이터를 저장하는 휘발성 메모리이다. 전하가 자연스럽게 소실되므로 주기적인 Refresh가 필요하며, SRAM보다 느리지만 회로가 단순하고 집적도가 높아 주로 주기억장치에 사용된다.
한 줄 대비: SRAM은 플립플롭·고속·Refresh 불필요, DRAM은 콘덴서·상대적 저속·Refresh 필요이다.

NoteBase

선지별 해설

① X
틀린 표현은 ‘DRAM은 SRAM에 비하여 고속으로 접근 가능하다’이다. 일반적으로 SRAM이 DRAM보다 접근 속도가 빠르며, 올바르게는 ‘DRAM은 SRAM보다 접근 속도가 느리다’이다.

② X
틀린 표현은 ‘DRAM은 SRAM에 비하여 전력 소모가 크다’이다. 일반적으로 DRAM은 셀 구조가 단순하여 SRAM보다 소비 전력이 작다. 올바르게는 ‘일반적으로 SRAM의 소비 전력이 DRAM보다 크다’이다.

③ O
DRAM은 콘덴서의 전하로 비트를 저장하고, SRAM은 플립플롭으로 비트를 저장하므로 옳다.

④ X
틀린 표현은 ‘SRAM은 Refresh를 필요로 한다’이다. Refresh가 필요한 것은 전하가 누설되는 콘덴서를 사용하는 DRAM이다. 올바르게는 ‘SRAM은 Refresh가 필요 없고, DRAM은 주기적인 Refresh가 필요하다’이다.

오답 포인트

①·②·④는 SRAM과 DRAM의 특성을 서로 바꿔 놓은 선지이다. 특히 DRAM의 D를 동적(dynamic)으로 기억하면, 전하가 사라지기 전에 계속 Refresh해야 한다는 점을 구별할 수 있다.

다시 나오면 이렇게 판단

먼저 저장 소자를 확인한다. 플립플롭이면 SRAM, 콘덴서이면 DRAM이다. 이어서 플립플롭은 빠르고 Refresh가 필요 없으며, 콘덴서는 상대적으로 느리고 Refresh가 필요하다는 연결 관계로 판단한다.

암기 포인트

SRAM은 플립플롭·고속·무 Refresh, DRAM은 콘덴서·저속·Refresh이다.

NoteBase

16. 디자인 패턴 중 생성 패턴(Creational Pattern)에 해당하는 것은?
① Singleton 패턴
② Observer 패턴
③ Adapter 패턴
④ Decorator 패턴

정답 1번

출제 포인트

GoF 디자인 패턴을 생성 패턴, 구조 패턴, 행위 패턴으로 구분할 수 있는지를 묻는 문제이다.

핵심 개념

생성 패턴: 객체의 생성 과정과 인스턴스화 방식을 다루는 패턴이다. 키워드는 '객체 생성'이다.
구조 패턴: 클래스와 객체를 조합하여 더 큰 구조를 만드는 방식을 다루는 패턴이다. 키워드는 '구조와 결합'이다.
행위 패턴: 객체 사이의 책임 분배와 상호작용 방식을 다루는 패턴이다. 키워드는 '행동과 통신'이다.

선지별 해설

① O — Singleton 패턴은 클래스의 인스턴스를 하나만 생성하고 그 인스턴스에 대한 전역 접근점을 제공하므로 생성 패턴에 해당한다.
② X — Observer 패턴을 생성 패턴으로 분류한 부분이 틀렸다. Observer 패턴은 한 객체의 상태가 바뀌면 의존하는 여러 객체에 이를 통지하는 행위 패턴이다.
③ X — Adapter 패턴을 생성 패턴으로 분류한 부분이 틀렸다. Adapter 패턴은 호환되지 않는 인터페이스를 함께 사용할 수 있도록 변환하는 구조 패턴이다.
④ X — Decorator 패턴을 생성 패턴으로 분류한 부분이 틀렸다. Decorator 패턴은 객체를 감싸 기능을 동적으로 추가하는 구조 패턴이다.

오답 포인트

모든 선지가 객체를 다루는 패턴이므로 이름만 보고 생성 패턴으로 착각하기 쉽다. '인스턴스를 하나만 생성한다'는 ①의 설명은 생성 방식 자체를 통제하지만, ②는 객체 간 통지, ③과 ④는 객체의 연결 구조를 다룬다.

다시 나오면 이렇게 판단

패턴이 객체를 어떻게 만들지 결정하면 생성 패턴, 객체를 어떻게 조합할지 결정하면 구조 패턴, 객체들이 어떻게 역할을 나누고 통신할지 결정하면 행위 패턴으로 판단한다.

NoteBase

암기 포인트

Singleton은 생성, Adapter·Decorator는 구조, Observer는 행위이다.

NoteBase

17. 파이프라인에서의 해저드와 관련된 설명으로 가장 적절하지 않은 것은?
① 구조적 해저드 – 파이프라인 단계 간 장치 사용에 대한 경쟁, 충돌로 발생
② 데이터 해저드 – 이전 명령의 결과값이 산출되기 전 다음 명령의 읽기에서 발생
③ 제어 해저드 – 조건부 분기 등 실행 순서의 변경으로 인하여 발생
④ 전방 전달(포워딩) – 명령어 실행 순서를 바꿔서 성능 저하를 최소화

정답 4번

출제 포인트

파이프라인 해저드의 세 유형인 구조적 해저드·데이터 해저드·제어 해저드의 발생 원인과 데이터 해저드 완화 기법인 전방 전달의 원리를 구별하는 문제이다.

핵심 개념

구조적 해저드: 여러 명령이 같은 하드웨어 자원을 동시에 사용하려 할 때 생기는 자원 충돌이다.
데이터 해저드: 명령 사이의 데이터 의존성 때문에 뒤 명령이 필요한 값을 제때 얻지 못하는 현상이다.
제어 해저드: 분기 명령 등으로 다음에 실행할 명령이 확정되지 않아 생기는 현상이다.
전방 전달(포워딩): 앞 명령의 결과를 레지스터에 기록할 때까지 기다리지 않고, 그 결과가 필요한 뒤 명령의 입력으로 직접 전달하여 데이터 해저드에 따른 지연을 줄이는 기법이다.

선지별 해설

① O — 구조적 해저드는 파이프라인의 여러 단계가 같은 장치를 동시에 사용하려 할 때 발생하는 하드웨어 자원 경쟁 또는 충돌이다.
② O — 앞 명령의 결과가 아직 산출되거나 기록되지 않았는데 뒤 명령이 그 값을 읽으려 하면 쓰기 후 읽기(RAW) 의존에 해당하는 데이터 해저드가 발생한다.
③ O — 조건부 분기처럼 실행 흐름을 바꾸는 명령이 있으면 다음에 처리할 명령이 달라질 수 있으므로 제어 해저드가 발생한다.
④ X — 틀린 표현은 '명령어 실행 순서를 바꿔서'이다. 전방 전달은 명령의 실행 순서를 바꾸는 기법이 아니라, 앞 명령의 연산 결과를 필요한 뒤 명령에 직접 전달하여 데이터 해저드로 인한 대기 시간을 줄이는 기법이다.

NoteBase

오답 포인트

④의 '성능 저하를 최소화'라는 목적만 보면 전방 전달처럼 보이지만, 핵심 동작은 실행 순서 변경이 아니라 결과값의 우회 전달이다. 명령 실행 순서를 바꾸는 방식과 포워딩을 혼동하면 안 된다.

다시 나오면 이렇게 판단

발생 원인을 기준으로 구분한다. 자원 충돌이면 구조적 해저드, 값의 선후 의존이면 데이터 해저드, 분기로 다음 명령이 불확실하면 제어 해저드이다. 포워딩은 '결과값을 직접 우회 전달'한다는 표현으로 판단한다.

암기 포인트

구조는 자원 충돌, 데이터는 값 의존, 제어는 분기, 포워딩은 결과 우회 전달이다.

NoteBase

18. 다음 표는 컴파일러와 인터프리터에 대하여 비교한 것이다. 이 중 빈칸 (가)~(라)를 순서대로 나열한 것으로 가장 적절한 것은?

구분	컴파일러	인터프리터
실행 이진 파일 저장	예	아니오
상대적 실행 속도	(가)	(나)
해당 언어 예	(다)	(라)

① 고속, 저속, C, 자바스크립트
② 저속, 고속, Swift, 파이썬
③ 고속, 저속, PHP, Rust
④ 저속, 고속, Go, Ruby

정답 1번

출제 포인트

이 문제는 컴파일러와 인터프리터를 실행 파일 저장 여부, 상대적 실행 속도, 대표 언어로 구분하는 문제이다.

핵심 개념

컴파일러는 원시 프로그램 전체를 한꺼번에 번역하여 실행 파일을 만들고 저장하는 번역기이다. 번역이 끝난 실행 파일을 바로 실행하므로 상대적 실행 속도는 고속이며, 대표 언어로 C가 있다.
인터프리터는 원시 프로그램을 명령문 단위로 번역하면서 즉시 실행하는 번역기이다. 별도의 실행 파일을 저장하지 않고 실행할 때마다 번역 과정이 필요하므로 상대적 실행 속도는 저속이며, 대표 언어로 자바스크립트가 있다.
따라서 (가)~(라)는 고속, 저속, C, 자바스크립트 순서이다.

NoteBase

선지별 해설

① O — 컴파일러는 고속, 인터프리터는 저속이고, C는 컴파일러 방식, 자바스크립트는 인터프리터 방식의 대표 예이므로 네 빈칸이 모두 알맞다.
② X — 틀린 부분은 컴파일러를 '저속', 인터프리터를 '고속'이라고 한 표현이다. Swift와 파이썬의 예시는 각 칸에 맞지만, 속도는 '고속, 저속'으로 고쳐야 하므로 전체 순서는 '고속, 저속, Swift, 파이썬'이 된다.
③ X — 틀린 부분은 PHP를 컴파일러 언어, Rust를 인터프리터 언어의 예로 둔 것이다. 두 언어의 위치를 바꾸어 '고속, 저속, Rust, PHP'로 고쳐야 한다.
④ X — 속도와 언어 예시가 모두 뒤바뀌었다. Go는 컴파일러 방식, Ruby는 인터프리터 방식의 대표 예이므로 '고속, 저속, Go, Ruby'로 고쳐야 한다.

오답 포인트

번역에 걸리는 시간과 번역 후 프로그램의 실행 속도를 혼동하기 쉽다. 컴파일러는 실행 전에 전체 번역이 필요하지만, 일단 생성된 실행 파일은 번역 없이 실행되므로 상대적 실행 속도가 빠르다. 또한 ②는 언어 예시가 맞더라도 속도 배열이 반대이므로 정답이 될 수 없다.

다시 나오면 이렇게 판단

먼저 실행 파일을 저장하면 컴파일러, 저장하지 않고 명령문 단위로 번역·실행하면 인터프리터로 구분한다. 이어서 컴파일러는 고속, 인터프리터는 저속이라는 상대적 실행 속도를 연결하고, 각 언어 예시가 해당 방식에 맞는지 확인한다.

암기 포인트

컴파일러는 '전체 번역·실행 파일·고속·C', 인터프리터는 '한 줄씩 번역·실행 파일 없음·저속·자바스크립트'로 연결한다.

NoteBase

19. 데이터베이스의 설계 단계 순서로 가장 적절한 것은?

① 요구분석 → 개념설계 → 논리설계 → 물리설계 → 구현

② 요구분석 → 물리설계 → 개념설계 → 논리설계 → 구현

③ 요구분석 → 물리설계 → 논리설계 → 개념설계 → 구현

④ 요구분석 → 논리설계 → 개념설계 → 물리설계 → 구현

정답 1번

출제 포인트

이 문제는 데이터베이스를 실제로 구축하기까지 요구사항을 구체화해 가는 설계 단계의 표준 순서를 묻는 문제이다.

올바른 순서

1. 요구분석 — 사용자의 요구사항과 처리할 데이터의 범위 및 조건을 파악한다.
2. 개념설계 — 요구사항을 바탕으로 개체, 속성, 관계 등 현실 세계의 구조를 개념적 모델로 표현한다.
3. 논리설계 — 개념적 모델을 관계형 스키마 등 목표 데이터 모델에 맞는 논리 구조로 변환한다.
4. 물리설계 — 저장 구조, 접근 경로, 인덱스 등 실제 저장과 성능에 관한 사항을 결정한다.
5. 구현 — 설계 결과를 데이터베이스 관리 시스템에서 스키마와 객체로 생성한다.

왜 이 순서: 앞 단계의 산출물이 다음 단계의 입력이 되므로, 무엇이 필요한지 분석한 뒤 개념 구조를 만들고 이를 논리 구조와 물리 구조로 차례로 구체화한 후 구현한다.

NoteBase

선지별 해설

① O — 요구분석 → 개념설계 → 논리설계 → 물리설계 → 구현의 순서가 표준적인 설계 흐름과 일치한다.

② X — 틀린 부분은 물리설계를 개념설계와 논리설계보다 앞에 둔 것이다. 물리설계는 논리 구조가 확정된 뒤 저장 구조와 접근 경로를 정하는 단계이다. 올바르게는 요구분석 → 개념설계 → 논리설계 → 물리설계 → 구현이다.

③ X — 틀린 부분은 물리설계를 먼저 하고 개념설계를 논리설계 뒤에 둔 것이다. 개념설계가 먼저 현실의 요구를 개념적 구조로 만들고, 논리설계가 이를 목표 데이터 모델로 변환한다. 올바르게는 요구분석 → 개념설계 → 논리설계 → 물리설계 → 구현이다.

④ X — 틀린 부분은 논리설계를 개념설계보다 앞에 둔 것이다. 논리 스키마는 개념설계에서 만든 개념적 모델을 바탕으로 작성되므로 개념설계가 선행한다. 올바르게는 요구분석 → 개념설계 → 논리설계 → 물리설계 → 구현이다.

오답 포인트

개념설계·논리설계·물리설계의 이름이 비슷해 순서를 바꾸기 쉽다. 특히 ④처럼 논리설계를 개념설계보다 먼저 두는 함정에 주의한다. 설계는 추상적인 개념 구조에서 논리 구조로, 다시 실제 저장을 위한 물리 구조로 구체화된다.

다시 나오면 이렇게 판단

절차형 문항에서는 각 단계의 산출물이 다음 단계의 입력인지 확인한다. 데이터베이스 설계는 요구사항 파악 → 현실 세계의 개념화 → 데이터 모델에 맞춘 논리화 → 저장 방식의 물리화 → 실제 구축의 흐름으로 판단한다.

암기 포인트

요구 → 개념 → 논리 → 물리 → 구현, 즉 추상적인 요구를 점차 구체적인 저장 구조로 바꾼다.

NoteBase

20. 화이트박스 및 블랙박스에 대한 설명으로 가장 적절한 것은?

① 화이트박스 시험은 프로그램 내부와 무관하게 가능한 모든 데이터를 입력 시험

② 블랙박스 시험은 프로그램의 모든 명령이 실행되는 조합이 존재하는지 시험

③ 화이트박스 시험은 프로그램의 논리 흐름과 복잡도를 산출하여 시험

④ 블랙박스 시험은 프로그램 조건식을 분석, 가능한 오류값을 점검하여 시험

정답 3번

출제 포인트

화이트박스 시험과 블랙박스 시험을 프로그램 내부 구조의 참조 여부로 구분하는 문제이다.

핵심 개념

화이트박스 시험: 프로그램의 내부 논리 구조와 제어 흐름을 바탕으로 문장, 분기, 조건, 경로 등을 검사하는 구조 기반 시험이다. 키워드는 내부 구조·논리 흐름·경로·복잡도이다.
블랙박스 시험: 프로그램 내부 구현을 보지 않고 요구 명세에 따른 입력과 출력 및 기능을 검사하는 명세 기반 시험이다. 키워드는 입력·출력·기능·동등 분할·경계값 분석이다.
핵심 구분은 내부 구조를 기준으로 시험하면 화이트박스, 외부 기능과 명세를 기준으로 시험하면 블랙박스라는 점이다.

NoteBase

선지별 해설

① X
화이트박스 시험은 프로그램 내부와 무관하게 입력만 검사하는 시험이 아니다. 틀린 표현은 '프로그램 내부와 무관하게 가능한 모든 데이터를 입력'이다. 올바르게는 화이트박스 시험은 프로그램의 내부 논리와 제어 구조를 분석하여 문장, 분기, 조건, 경로가 적절히 수행되는지를 검사한다고 해야 한다. 내부 구조와 무관하게 입력과 출력을 검사하는 것은 블랙박스 시험의 관점이다.

② X
프로그램의 명령과 실행 경로가 수행되는지를 확인하는 것은 내부 구조를 바탕으로 하는 화이트박스 시험이다. 틀린 표현은 '블랙박스 시험'이다. 올바르게는 프로그램의 문장이나 경로의 실행 조합을 검사하는 것은 화이트박스 시험이라고 해야 한다.

③ O
프로그램의 논리 흐름을 분석하고 순환 복잡도와 같은 지표를 산출하여 독립 경로를 정하는 기초 경로 시험은 대표적인 화이트박스 시험 방법이다. 따라서 화이트박스 시험에 대한 설명으로 적절하다.

④ X
프로그램 내부의 조건식을 분석하는 조건 시험은 화이트박스 시험에 해당한다. 틀린 표현은 '블랙박스 시험은 프로그램 조건식을 분석'한다는 부분이다. 올바르게는 화이트박스 시험이 내부 조건식의 참·거짓과 조합을 검사하며, 블랙박스 시험은 내부 조건식을 보지 않고 명세를 바탕으로 경계값이나 오류가 예상되는 입력값 등을 선정하여 검사한다고 해야 한다.

오답 포인트

①과 ②는 화이트박스와 블랙박스의 특징을 서로 뒤바꾸었고, ④는 화이트박스의 조건식 분석과 블랙박스에서 사용할 수 있는 오류 추정 관점을 한 문장에 섞었다. 특히 '모든 데이터', '모든 명령', '조건식' 같은 단어보다 내부 구조를 실제로 참조하는지를 먼저 판단해야 한다.

다시 나오면 이렇게 판단

선지에 내부 논리, 문장, 분기, 조건, 경로, 순환 복잡도가 나오면 화이트박스 시험으로 판단한다. 요구 명세, 입력과 출력, 기능, 동등 분할, 경계값이 나오면 블랙박스 시험으로 판단한다.

암기 포인트

화이트박스는 내부의 길을 보고 검사하고, 블랙박스는 겉으로 드나드는 입력과 출력을 보고 검사한다.

NoteBase

21. 다음과 같은 소스의 C 프로그램을 실행하여 5를 입력하였다. 출력값으로 가장 적절한 것은?

```
#include<stdio.h>
int CalNum(int n);
int main( ) {
        int n;
        printf("양수를 넣으시오: ");
        scanf("%d",&n);
        printf("결과 = %d\n\r", CalNum(n));
        return 0;
}

int CalNum(int n) {
        if (n == 1)
                return 1;
        else if (n%2 == 0)
                return n*CalNum(n-1);
        else
                return n+CalNum(n-1);
}
```

① 결과 = 33
② 결과 = 25
③ 결과 = 20
④ 결과 = 16

정답 2번

출제 포인트

재귀 함수의 종료 조건을 확인하고, 홀수와 짝수에 서로 다른 연산을 적용하여 반환값을 구하는 문제이다.

NoteBase

조건 정리

주어진 값: $n = 5$
구해야 할 값: $CalNum(5)$
주의할 조건: $n = 1$이면 1을 반환한다. n이 짝수이면 $n \times CalNum(n-1)$, 홀수이면 $n + CalNum(n-1)$을 반환한다.

사용할 공식

$CalNum(n) = 1$: $n = 1$인 경우
$CalNum(n) = n \times CalNum(n-1)$: n이 짝수인 경우
$CalNum(n) = n + CalNum(n-1)$: n이 홀수인 경우
이 식을 쓰는 이유는 함수가 n의 값에 따라 세 분기로 나뉘어 자기 자신을 $n-1$로 다시 호출하기 때문이다.

계산 과정

1. $CalNum(1) = 1$이다.
2. $CalNum(2) = 2 \times CalNum(1) = 2$이다.
3. $CalNum(3) = 3 + CalNum(2) = 5$이다.
4. $CalNum(4) = 4 \times CalNum(3) = 20$이다.
5. $CalNum(5) = 5 + CalNum(4) = 25$이다.

따라서 출력되는 결과값은 25이다.

선지 판단

① X — '결과 = 33'은 계산값 25와 다르므로 틀리다. 올바르게는 '결과 = 25'이다.
② O — $CalNum(5) = 5 + 4 \times (3 + 2 \times 1) = 25$이므로 '결과 = 25'가 맞다.
③ X — '결과 = 20'은 $CalNum(4)$의 값에서 계산을 멈춘 것이므로 틀리다. 입력값 5의 마지막 홀수 분기에서 5를 더해야 하므로 올바르게는 '결과 = 25'이다.
④ X — '결과 = 16'은 재귀 호출에 따른 덧셈과 곱셈을 순서대로 적용한 값이 아니므로 틀리다. 올바르게는 '결과 = 25'이다.

오답 포인트

가장 큰 함정은 $CalNum(4) = 20$을 구한 뒤 재귀 계산을 끝내는 것이다. 처음 입력한 5는 홀수이므로 마지막에 $5 + CalNum(4)$를 계산해야 한다. 또한 조건식 $n\%2 == 0$은 짝수 판별이므로 짝수일 때만 곱셈을 적용한다.

다시 나오면 이렇게 판단

재귀 함수는 먼저 종료 조건을 찾고, 가장 작은 입력값부터 각 호출의 반환값을 차례로 계산한다. 각 단계에서는 현재 n의 홀짝을 다시 확인하여 덧셈과 곱셈 중 해당 연산만 적용한다.

NoteBase

암기 포인트

기저 조건은 1, 짝수는 곱하기, 홀수는 더하기이다.

NoteBase

22. 네트워크상 장치에 대한 설명으로 가장 적절한 것은?

① 게이트웨이 - 동일 프로토콜을 사용하는 네트워크 간 연결 기능을 제공

② 라우터 - 동일 프로토콜을 사용하는 패킷의 전송 경로를 설정, 패킷 전달

③ 브릿지 - 동일 프로토콜을 사용하는 패킷을 모두 전달

④ 스위치(L2) - 포트로 수신된 패킷 페이로드 데이터를 분석, 필터링

정답 2번

출제 포인트

게이트웨이, 라우터, 브리지, L2 스위치가 연결하는 대상과 전달·필터링에 사용하는 정보를 구분하는 문제이다.

핵심 개념

게이트웨이는 서로 다른 프로토콜이나 구조를 사용하는 네트워크 사이에서 변환과 연결을 담당한다.
라우터는 네트워크 계층에서 목적지 IP 주소를 바탕으로 패킷의 경로를 선택하고 다음 네트워크로 전달한다.
브리지는 데이터 링크 계층에서 MAC 주소를 확인하여 필요한 프레임만 다른 세그먼트로 전달하거나 차단한다.
L2 스위치는 다중 포트 브리지처럼 동작하며 프레임의 MAC 주소 정보를 이용해 전달 포트를 결정한다.

선지별 해설

① O — 게이트웨이는 서로 다른 프로토콜을 사용하는 네트워크 사이에서 프로토콜 변환을 수행하여 네트워크 간 연결 기능을 제공하므로 선지의 설명은 일반적인 정의에 부합한다. 따라서 원문의 문구만으로는 ①을 오답으로 배제하기 어렵다.
② O — 라우터는 동일한 네트워크 계층 프로토콜을 사용하는 네트워크 사이에서 라우팅 테이블 등을 이용해 패킷의 전송 경로를 정하고 패킷을 전달한다. 공식 채점 답은 ②이지만, ①도 타당하므로 문항에는 복수 정답 가능성이 있다.
③ X — 틀린 표현은 '패킷을 모두 전달'이다. 브리지는 모든 트래픽을 무조건 전달하는 장치가 아니라 프레임의 목적지 MAC 주소를 확인하여 필요한 프레임만 다른 세그먼트로 전달하고 나머지는 필터링한다.
④ X — 틀린 표현은 '패킷 페이로드 데이터를 분석'이다. L2 스위치는 일반적으로 페이로드의 내용을 분석하는 것이 아니라 이더넷 프레임 헤더의 출발지·목적지 MAC 주소를 학습하고 확인하여 전달 또는 필터링한다.

NoteBase

오답 포인트

브리지와 L2 스위치가 데이터를 전달한다는 사실만 보고 ③·④를 맞다고 판단하기 쉽다. 그러나 ③의 '모두'와 ④의 '페이로드'가 함정이다. 두 장치는 MAC 주소를 기준으로 선택적으로 프레임을 전달한다.

다시 나오면 이렇게 판단

장치마다 판단 기준을 확인한다. 서로 다른 프로토콜의 변환·연결은 게이트웨이, 목적지 IP와 경로 선택은 라우터, MAC 주소에 따른 프레임 전달·필터링은 브리지와 L2 스위치에 해당한다.

암기 포인트

게이트웨이는 프로토콜 변환, 라우터는 IP 경로, 브리지·L2 스위치는 MAC 주소이다.

NoteBase

23. 다음 보기에서 설명된 프로토콜로 가장 적절한 것은?

> 보기 : 해당 네트워크상의 모든 호스트와 라우터에게 브로드캐스트하며, 요청과 응답을 통해 획득한 주소값을 캐시테이블에 저장, 통신 효율을 높임.

① ICMP ② IGMP
③ NAT ④ ARP

정답 4번

출제 포인트

이 문제는 요청을 브로드캐스트하고 응답으로 얻은 주소 대응 정보를 캐시에 저장하는 ARP의 동작을 다른 네트워크 프로토콜과 구별하는 문제이다.

핵심 개념

ARP(Address Resolution Protocol)는 같은 로컬 네트워크에서 IPv4 주소에 대응하는 MAC 주소를 알아내는 프로토콜이다. ARP 요청은 브로드캐스트로 전달되고, 해당 IPv4 주소를 가진 장치가 자신의 MAC 주소를 응답하면 그 대응 관계가 ARP 캐시에 저장된다.
ICMP는 IP 통신의 오류 보고와 진단, IGMP는 IPv4 멀티캐스트 그룹 구성원 관리, NAT는 사설 주소와 공인 주소 등의 변환을 담당한다.

선지별 해설

① X — ICMP는 IP 패킷 전달 중 발생한 오류를 알리거나 네트워크 상태를 진단하는 제어용 프로토콜이다. 틀린 표현은 '브로드캐스트 요청으로 주소값을 얻어 캐시한다'는 설명이며, 올바르게는 '오류 보고와 진단 정보를 전달한다'이다.
② X — IGMP는 호스트와 라우터 사이에서 IPv4 멀티캐스트 그룹의 가입 상태를 관리한다. 틀린 표현은 '주소값을 캐시테이블에 저장한다'는 설명이며, 올바르게는 '멀티캐스트 그룹 구성원 정보를 관리한다'이다.
③ X — NAT는 패킷의 IP 주소와 필요하면 포트 번호를 변환하여 내부 네트워크와 외부 네트워크의 통신을 중계한다. 틀린 표현은 '브로드캐스트 요청과 응답으로 주소 대응 관계를 알아낸다'는 설명이며, 올바르게는 '사설 주소와 공인 주소 등의 주소를 변환한다'이다.
④ O — ARP는 로컬 네트워크에 ARP 요청을 브로드캐스트하고, 응답으로 확인한 IPv4 주소와 MAC 주소의 대응 관계를 ARP 캐시에 저장하여 이후 통신의 주소 해석을 효율화한다.

NoteBase

오답 포인트

보기의 '주소값'만 보고 IP 주소를 변환하는 NAT를 고르기 쉽다. 그러나 '로컬 네트워크 브로드캐스트 → 요청·응답 → 캐시테이블 저장'이라는 연속 동작은 ARP를 가리킨다.

다시 나오면 이렇게 판단

IPv4 주소에 대응하는 MAC 주소, 브로드캐스트 요청, ARP 캐시가 함께 제시되면 ARP로 판단한다. 오류·진단은 ICMP, 멀티캐스트 그룹 관리는 IGMP, 주소 변환은 NAT로 구분한다.

암기 포인트

ARP는 IP 주소로 MAC 주소를 찾고 그 대응 관계를 캐시한다.

NoteBase

24. 클라우드 컴퓨팅에 대한 설명으로 가장 적절하지 않은 것은?

① SaaS – 소프트웨어 실행파일을 배포하기 위한 도구를 제공받아 프로그램을 전달함

② PaaS – 개발을 위한 플랫폼 도구를 제공받아 사용자 서비스를 개발 공급함

③ IaaS – 물리적 전산 자원을 대여, 서버, 스토리지, 네트워크 등을 제공받음

④ AIaaS – AI 서비스 도구를 제공받아 지능형 사용자 서비스를 개발 공급함

정답 1번

출제 포인트

클라우드 서비스 유형을 사용자가 제공받는 자원과 책임 범위에 따라 구분하는 문제이다. 특히 완성된 소프트웨어를 이용하는 SaaS와 응용 프로그램 개발·배포 환경을 이용하는 PaaS를 구별해야 한다.

핵심 개념

SaaS: 서비스 제공자가 운영하는 완성된 소프트웨어를 인터넷으로 이용하는 방식이다. 키워드는 '소프트웨어 사용'이다.

PaaS: 응용 프로그램을 개발·실행·배포할 수 있는 플랫폼과 도구를 제공받는 방식이다. 키워드는 '개발·배포 환경'이다.

IaaS: 서버, 스토리지, 네트워크 같은 기초 전산 인프라를 가상화된 자원으로 제공받는 방식이다. 키워드는 '인프라 임대'이다.

AIaaS: 인공지능 모델, API, 개발 도구 등 AI 기능과 환경을 서비스로 제공받아 지능형 서비스를 구현하는 방식이다. 키워드는 'AI 기능·도구 활용'이다.

SaaS는 완성된 소프트웨어의 이용, PaaS는 프로그램 개발·배포 플랫폼의 이용, IaaS는 기초 인프라의 이용, AIaaS는 AI 기능·도구의 이용으로 구분한다.

NoteBase

선지별 해설

① X — '소프트웨어 실행파일을 배포하기 위한 도구를 제공받아 프로그램을 전달한다'는 설명은 SaaS가 아니라 PaaS에 해당한다. 올바르게는 'SaaS는 서비스 제공자가 운영하는 완성된 소프트웨어를 사용자가 인터넷을 통해 이용하는 방식이다'라고 해야 한다.
② O — PaaS는 응용 서비스를 개발하고 실행·배포하는 데 필요한 플랫폼과 개발 도구를 제공한다.
③ O — IaaS는 서버, 스토리지, 네트워크 등의 기초 전산 자원을 서비스 형태로 제공한다.
④ O — AIaaS는 AI 모델·API·개발 도구 등 AI 기능을 제공하여 사용자가 지능형 서비스를 개발할 수 있게 한다.

오답 포인트

①의 '소프트웨어'라는 말만 보고 SaaS로 판단하기 쉽다. 그러나 핵심 표현은 '실행파일을 배포하기 위한 도구'이며, 개발·실행·배포 도구를 제공하는 것은 PaaS이다. SaaS는 개발 도구가 아니라 완성된 소프트웨어 자체를 이용하게 한다.

다시 나오면 이렇게 판단

선지에서 사용자가 무엇을 제공받는지 확인한다. 완성된 프로그램이면 SaaS, 개발·배포 환경이면 PaaS, 서버·스토리지·네트워크이면 IaaS, AI 모델·API·도구이면 AIaaS로 판단한다.

암기 포인트

SaaS는 소프트웨어 사용, PaaS는 개발 플랫폼, IaaS는 인프라, AIaaS는 AI 기능이다.

NoteBase

25. KVM(Kernel-based Virtual Machine)에 대한 설명으로 가장 적절하지 않은 것은?

① 가상머신을 동작시키기 위한 기반 소프트웨어이다.

② 윈도우즈에서 이를 이용하여 직접 가상화를 제공한다.

③ 하이퍼바이저(Hypervisor)의 일종이다.

④ 게스트 운영체제로 윈도우즈와 리눅스 등 제한 없이 사용 가능하다.

정답 2번

출제 포인트

KVM이 리눅스 커널을 기반으로 동작하는 가상화 기술이라는 점과 호스트 운영체제·게스트 운영체제의 역할을 구분하는 문제이다.

핵심 개념

KVM(Kernel-based Virtual Machine)은 리눅스 커널에 포함된 가상화 기능으로, 하드웨어 가상화 지원을 이용하여 리눅스 시스템을 가상머신을 실행하는 하이퍼바이저 환경으로 만든다.
호스트는 KVM을 제공하고 가상머신을 관리하는 리눅스 시스템이며, 게스트는 그 위의 가상머신에서 실행되는 윈도우·리눅스 등의 운영체제이다.

선지별 해설

① O — KVM은 가상머신을 생성·실행하기 위한 리눅스 커널 기반의 핵심 가상화 소프트웨어이다.
② X — 틀린 표현은 '윈도우즈에서 이를 이용하여 직접 가상화를 제공한다'이다. KVM은 리눅스 커널의 기능이므로 KVM을 제공하는 호스트 운영체제는 리눅스이다. 윈도우는 KVM 위에서 게스트 운영체제로 실행할 수 있다.
③ O — KVM은 리눅스 커널을 하이퍼바이저로 동작하게 하는 가상화 기술이므로 하이퍼바이저의 일종으로 본다.
④ O — KVM 가상머신에는 리눅스뿐 아니라 윈도우도 게스트 운영체제로 설치하여 사용할 수 있으며, 일반적으로 게스트 운영체제를 KVM용으로 별도 수정할 필요가 없다.

오답 포인트

②는 '윈도우를 게스트로 실행할 수 있다'는 사실을 '윈도우가 KVM 호스트로서 직접 가상화를 제공한다'는 말로 바꾼 함정이다. 호스트와 게스트를 혼동하면 틀리기 쉽다.

NoteBase

다시 나오면 이렇게 판단

KVM 문항에서는 먼저 호스트와 게스트를 구분한다. KVM의 호스트 기반은 리눅스 커널이고, 윈도우와 리눅스는 게스트 운영체제로 실행될 수 있다.

암기 포인트

KVM은 리눅스가 호스트, 윈도우·리눅스가 게스트이다.

NoteBase