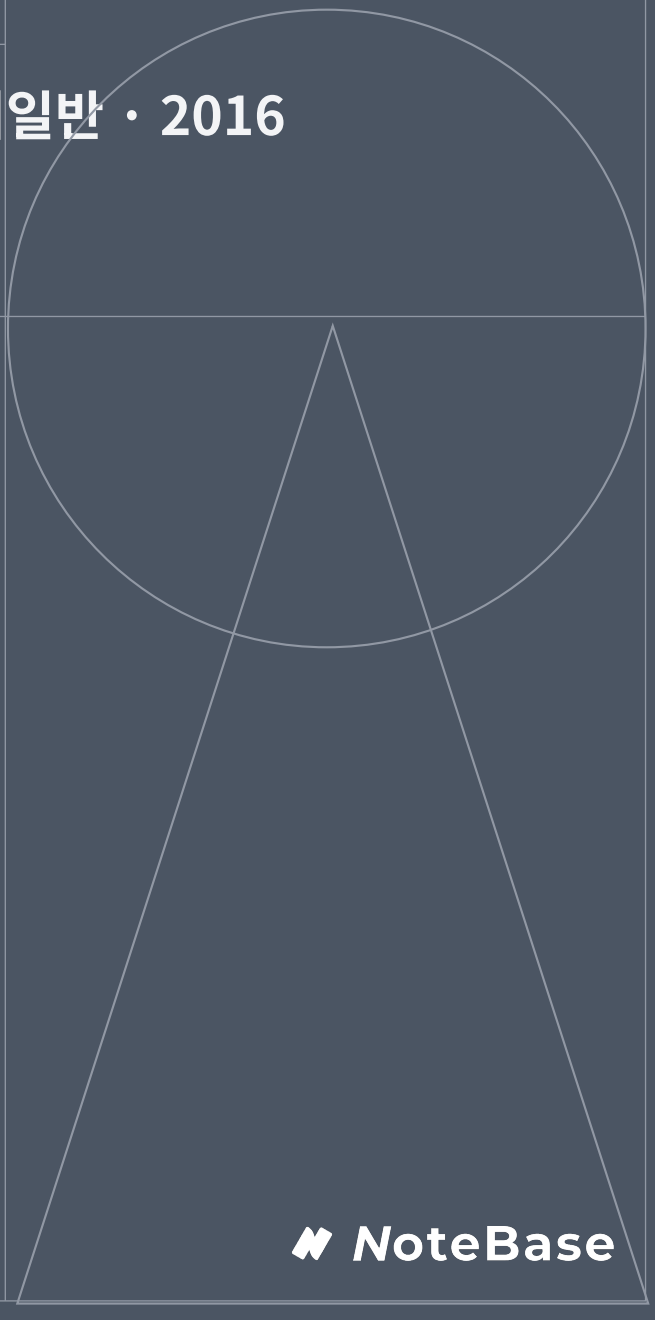
기출OX 해설집

공무원/국회직/9급 · 컴퓨터일반 · 2016



NoteBase

기출OX 해설집

공무원/국회직/9급 · 컴퓨터일반 · 2016

공무원/국회직/9급 · 컴퓨터일반 · 2016

1. 다음 중 DBMS를 이용하여 데이터를 관리하고 활용함으로써 얻을 수 있는 장점으로 옳지 않은 것은?

① 데이터의 중복성을 제어하여 저장공간의 낭비를 방지한다.
② 조직 내의 다양한 응용 사이의 데이터 일관성을 유지한다.
③ 효율적인 질의 처리를 위한 저장 구조와 탐색 기법을 제공한다.
④ 무결성 제약조건을 만족하도록 데이터를 관리한다.
⑤ 사용자가 데이터베이스의 모든 데이터를 자유롭게 액세스할 수 있다.

정답 5번

출제 포인트

DBMS를 이용한 데이터 관리의 장점과 사용자 접근 통제의 필요성을 구분하는 문제이다.

핵심 개념

DBMS는 데이터베이스를 정의·저장·검색·갱신하고 여러 사용자의 접근을 통제하는 소프트웨어이다. 중복성 제어, 일관성 유지, 효율적인 질의 처리, 무결성 보장, 보안과 권한 관리가 주요 기능이다. 특히 데이터 공유는 허가된 범위에서 이루어지며, 모든 사용자가 모든 데이터에 자유롭게 접근하는 것을 뜻하지 않는다.

선지별 해설

① O — DBMS는 동일한 데이터가 여러 곳에 불필요하게 반복 저장되는 것을 제어하여 저장 공간의 낭비와 갱신 이상을 줄인다.
② O — 데이터를 통합하여 관리하므로 여러 응용 사이에서 동일한 데이터 값과 규칙을 유지해 데이터 일관성을 높인다.
③ O — DBMS는 인덱스 등의 저장 구조와 검색·질의 처리 기법을 제공하여 필요한 데이터를 효율적으로 찾고 처리하게 한다.
④ O — 도메인·키·참조 무결성 등의 제약조건을 적용하여 데이터가 정해진 규칙을 만족하도록 관리한다.
⑤ X — 틀린 표현은 '모든 데이터를 자유롭게 액세스할 수 있다'이다. 올바르게는 DBMS가 사용자별 권한을 설정하여 인가된 사용자가 허용된 데이터와 연산에만 접근할 수 있도록 통제한다.

NoteBase

오답 포인트

데이터 공유를 접근의 무제한 허용으로 오해하기 쉽다. DBMS의 공유 기능은 동시 접근을 지원한다는 뜻이며, 보안을 위해 사용자와 역할에 따른 접근 권한은 제한된다.

다시 나오면 이렇게 판단

DBMS의 장점을 묻는 선지에서는 중복성 제어·일관성·효율성·무결성·보안을 확인한다. '모든 사용자', '모든 데이터', '자유롭게 접근'처럼 권한 통제를 부정하는 절대적 표현은 잘못된 설명으로 판단한다.

암기 포인트

DBMS는 데이터를 함께 쓰게 하되, 권한까지 풀어 주지는 않는다.

NoteBase

2. 다음 중 플립플롭(flip-flop)의 용도에 해당하는 것은?

① n비트의 입력에서 1의 개수가 짝수면 1, 홀수면 0을 출력한다.
② 1비트의 0과 1의 두 개의 상태 중 하나를 안정적으로 저장할 수 있다.
③ n비트의 입력에 따라 2n개의 출력 중 하나만 1을 출력한다.
④ 두 비트의 입력에 대하여 합과 자리올림(carry)을 출력한다.
⑤ 여러 개의 입력 회선 중 선택된 한 회선의 입력을 출력 회선으로 출력한다.

정답 2번

출제 포인트

플립플롭의 기본 기능과 패리티 회로, 디코더, 전가산기, 멀티플렉서의 기능을 구별하는 문제이다.

핵심 개념

플립플롭은 두 개의 안정된 상태 중 하나를 유지하여 1비트의 이진 정보 0 또는 1을 저장하는 순서논리회로이다. 클록이나 제어 입력에 따라 상태가 바뀌며, 입력이 사라진 뒤에도 다음 상태 변화가 있을 때까지 저장값을 유지한다.
패리티 회로는 입력에 포함된 1의 개수를 이용해 오류 검출용 패리티를 생성하거나 검사하고, 디코더는 입력 코드에 대응하는 출력선 하나를 활성화한다. 전가산기는 두 비트와 자리올림 입력을 더해 합과 자리올림을 출력하며, 멀티플렉서는 여러 입력 중 선택된 하나를 출력으로 전달한다.

선지별 해설

① X — 입력에 포함된 1의 개수가 짝수인지 홀수인지에 따라 값을 내는 것은 패리티 생성·검사 회로의 기능이다. 틀린 표현은 이를 플립플롭의 용도로 본 부분이며, 올바르게는 '입력의 1의 개수에 따른 패리티를 생성하거나 검사한다'이다.
② O — 플립플롭은 0과 1에 대응하는 두 안정 상태 중 하나를 유지하므로 1비트의 정보를 안정적으로 저장할 수 있다.
③ X — n비트 입력 코드에 따라 여러 출력 중 하나만 활성화하는 것은 디코더의 기능이다. 틀린 표현은 이를 플립플롭의 용도로 본 부분이며, 올바르게는 '일반적인 n-to-2^n 디코더는 입력 코드에 대응하는 출력 하나를 활성화한다'이다.
④ X — 두 비트의 입력에 대한 합과 자리올림을 출력하는 것은 가산기의 기능이다. 특히 두 입력만 더하면 반가산기이고, 이전 자리의 자리올림 입력까지 포함하면 전가산기이다. 틀린 표현은 이를 플립플롭의 용도로 본 부분이며, 올바르게는 '가산기는 합과 자리올림을 출력한다'이다.
⑤ X — 여러 입력 회선 중 선택 신호로 지정된 한 회선의 값을 출력 회선으로 전달하는 것은 멀티플렉서의 기능이다. 틀린 표현은 이를 플립플롭의 용도로 본 부분이며, 올바르게는 '멀티플렉서는 선택된 입력 하나를 출력으로 전달한다'이다.

NoteBase

오답 포인트

모든 선지가 디지털 논리회로의 실제 기능을 설명하므로 회로 이름을 섞기 쉽다. ②의 핵심어인 '두 개의 안정 상태'와 '1비트 저장'이 플립플롭을 직접 가리키며, ①은 패리티, ③은 디코딩, ④는 덧셈, ⑤는 입력 선택 기능이다.

다시 나오면 이렇게 판단

회로의 동작 목적을 먼저 찾는다. 상태 유지·1비트 저장이면 플립플롭, 코드에 따른 한 출력 활성화이면 디코더, 합과 자리올림이면 가산기, 여러 입력 중 하나의 선택이면 멀티플렉서로 판단한다.

암기 포인트

플립플롭은 1비트 저장, 디코더는 한 출력 선택, 가산기는 합과 자리올림, 멀티플렉서는 한 입력 전달이다.

NoteBase

3. 다음 중 배열에 저장된 n개의 레코드를 키의 오름차순으로 정렬하는 알고리즘에 대한 설명으로 옳은 것은?

① 힙(heap) 정렬은 안정적 정렬 알고리즘이다.
② 최악의 경우 퀵(quick) 정렬의 시간 복잡도는 O(n logn)이다.
③ 평균적인 상황에서 병합(merge) 정렬의 시간 복잡도는 O(n logn)이다.
④ 이미 정렬되어 있는 경우 병합 정렬의 시간 복잡도는 O(n)이다.
⑤ 삽입(insertion) 정렬은 평균적인 상황에서 n이 클수록 퀵 정렬에 비해 빠르다.

정답 3번

출제 포인트

힙 정렬, 퀵 정렬, 병합 정렬, 삽입 정렬의 안정성과 입력 상태별 시간 복잡도를 구분하는 문제이다.

핵심 개념

힙 정렬: 힙 자료구조를 이용해 최댓값 또는 최솟값을 반복적으로 꺼내 정렬하며, 일반적으로 불안정 정렬이다. 시간 복잡도는 최선·평균·최악 모두 $O(n\log n)$이다.
퀵 정렬: 피벗을 기준으로 배열을 분할해 정렬한다. 평균 시간 복잡도는 $O(n\log n)$이지만 분할이 계속 한쪽으로 치우치면 최악의 경우 $O(n^2)$이다.
병합 정렬: 배열을 나눈 뒤 정렬된 부분 배열을 병합한다. 입력의 정렬 상태와 관계없이 시간 복잡도는 $O(n\log n)$이다.
삽입 정렬: 각 원소를 앞쪽의 정렬된 부분에서 알맞은 위치에 삽입한다. 이미 정렬된 경우 $O(n)$이지만 평균·최악 시간 복잡도는 $O(n^2)$이다.

선지별 해설

① X — 틀린 표현은 '힙 정렬은 안정적 정렬 알고리즘이다'이다. 힙을 구성하고 원소를 교환하는 과정에서 같은 키를 가진 레코드의 상대적 순서가 바뀔 수 있으므로, 일반적인 힙 정렬은 불안정 정렬이다.
② X — 틀린 표현은 '최악의 경우 퀵 정렬의 시간 복잡도는 $O(n\log n)$이다'이다. 피벗이 계속 최솟값이나 최댓값으로 선택되어 분할이 한쪽으로 치우치면 최악 시간 복잡도는 $O(n^2)$이다.
③ O — 병합 정렬은 배열을 계속 절반으로 나누는 단계가 약 $\log n$번이고, 각 단계에서 전체 n개 원소를 병합하므로 평균 시간 복잡도는 $O(n\log n)$이다.
④ X — 틀린 표현은 '이미 정렬되어 있으면 병합 정렬의 시간 복잡도는 $O(n)$이다'이다. 표준 병합 정렬은 입력이 이미 정렬되어 있어도 분할과 병합을 수행하므로 시간 복잡도는 $O(n\log n)$이다.
⑤ X — 틀린 표현은 'n이 클수록 삽입 정렬이 퀵 정렬보다 빠르다'이다. 평균적으로 삽입 정렬은 $O(n^2)$, 퀵 정렬은 $O(n\log n)$이므로 n이 커질수록 일반적으로 퀵 정렬이 더 유리하다.

NoteBase

오답 포인트

평균 시간 복잡도와 최악 시간 복잡도를 바꾸어 보거나, 이미 정렬된 입력에서 빠른 삽입 정렬의 특성을 병합 정렬에 잘못 적용한 것이 함정이다. 특히 퀵 정렬은 평균적으로 빠르지만 최악에는 $O(n^2)$이고, 병합 정렬은 입력 상태와 무관하게 $O(n \log n)$이라는 차이를 구분해야 한다.

다시 나오면 이렇게 판단

각 정렬 알고리즘을 '안정성', '평균 시간 복잡도', '최악 시간 복잡도', '이미 정렬된 경우'의 네 기준으로 나누어 판단한다. 힙은 불안정, 퀵은 평균 $O(n \log n)$·최악 $O(n^2)$, 병합은 입력 상태와 관계없이 $O(n \log n)$, 삽입은 이미 정렬되었을 때 $O(n)$으로 정리된다.

암기 포인트

힙은 불안정, 퀵은 평균 빠름·최악 제곱, 병합은 항상 $n \log n$, 삽입은 정렬된 입력에서 선형이다.

NoteBase

4. 다음 중 기계학습(machine learning)에 관련된 설명으로 옳지 않은 것은?

① 학습 시 개별 데이터에 대한 미리 지정된 레이블 또는 목표치가 없는 경우에 적용하는 기법들을 비교사 또는 자율 학습(unsupervised learning)이라 부른다.

② 대표적인 unsupervised learning 기법으로는 clustering이 있다.

③ 인공 신경망(artificial neural network) 기법을 이용하여 unsupervised learning을 시행할 수 있다.

④ Decision tree는 unsupervised learning 기법으로 분류된다.

⑤ SVM(Support Vector Machine)은 교사 또는 지도 학습(supervised learning) 기법이다.

정답 4번

출제 포인트

지도학습과 비지도학습의 차이를 이해하고 대표적인 기계학습 기법을 올바르게 분류할 수 있는지를 묻는 문제이다.

핵심 개념

지도학습은 정답에 해당하는 레이블이나 목표값이 주어진 데이터로 입력과 출력의 관계를 학습하는 방법이다. 대표적으로 의사결정나무와 SVM을 이용한 분류·회귀가 있다.
비지도학습은 레이블이나 목표값이 없는 데이터에서 군집, 구조, 특징을 찾아내는 방법이다. 대표적으로 군집화가 있으며, 신경망도 자기조직화지도나 오토인코더와 같은 방식으로 비지도학습에 활용될 수 있다.

선지별 해설

① O — 개별 데이터에 미리 지정된 레이블이나 목표값이 없을 때 데이터의 구조나 패턴을 찾는 학습을 비지도학습이라고 한다.
② O — 군집화는 정답 레이블 없이 유사한 데이터끼리 묶는 대표적인 비지도학습 기법이다.
③ O — 인공신경망은 지도학습에만 한정되지 않으며, 자기조직화지도나 오토인코더처럼 레이블 없이 구조나 특징을 학습하는 데에도 사용할 수 있다.
④ X — 틀린 표현은 'Decision tree는 unsupervised learning 기법'이다. 의사결정나무는 레이블 또는 목표값을 이용해 분류나 회귀 규칙을 학습하는 대표적인 지도학습 기법이다.
⑤ O — SVM은 레이블이 주어진 학습 데이터로 분류 경계나 회귀 함수를 학습하는 지도학습 기법이다.

NoteBase

오답 포인트

③처럼 하나의 모델 계열이 여러 학습 방식에 활용될 수 있다는 점과 ④의 알고리즘 분류를 혼동하기 쉽다. 인공신경망은 학습 목적과 구성에 따라 비지도학습에도 쓰일 수 있지만, 일반적인 의사결정나무는 레이블을 기준으로 분기 규칙을 만드는 지도학습 기법이다.

다시 나오면 이렇게 판단

먼저 학습 데이터에 레이블이나 목표값이 필요한지를 본다. 정답을 이용해 분류·회귀하면 지도학습이고, 정답 없이 군집이나 잠재 구조를 찾으면 비지도학습이다.

암기 포인트

레이블 있음은 지도학습, 레이블 없음은 비지도학습이다. 의사결정나무·SVM은 지도학습, 군집화는 비지도학습으로 구분한다.

NoteBase

5. 다음 중 자료구조 큐(queue)에 대한 설명으로 옳은 것은?

① 후입선출(last-in first-out) 특성을 갖는 자료구조이다.
② 데이터를 넣는 위치는 뒤(rear)이고, 데이터를 꺼내는 위치는 앞(front)인 선형 리스트이다.
③ 프로그램 실행 시 함수 호출과 복귀를 위한 처리에 유용하다.
④ 인덱스를 이용하여 지정된 임의 위치에서 직접 데이터를 저장하거나 읽기 위한 자료구조이다.
⑤ 하나의 노드가 최대 2개의 자식 노드를 가질 수 있는 구조이다.

정답 2번

출제 포인트

큐의 삽입·삭제 위치와 선입선출 특성을 스택, 배열, 이진 트리의 특징과 구별하는 문제이다.

핵심 개념

큐(queue)는 먼저 들어온 데이터를 먼저 꺼내는 선입선출(FIFO) 자료구조이다. 일반적인 큐에서는 뒤(rear)에서 데이터를 삽입하고 앞(front)에서 데이터를 삭제한다.
스택(stack)은 가장 나중에 들어온 데이터를 먼저 꺼내는 후입선출(LIFO) 자료구조이다.
배열(array)은 인덱스로 각 원소의 위치를 지정하여 직접 접근하는 자료구조이고, 이진 트리(binary tree)는 각 노드가 최대 2개의 자식 노드를 갖는 계층형 자료구조이다.

선지별 해설

① X — 틀린 표현은 '후입선출(last-in first-out)'이다. 큐는 먼저 들어온 데이터가 먼저 나오는 선입선출(first-in first-out) 자료구조이다. 후입선출은 스택의 특성이다.
② O — 큐는 뒤(rear)에서 데이터를 넣고 앞(front)에서 데이터를 꺼내므로 올바른 설명이다.
③ X — 함수 호출과 복귀에는 가장 최근에 호출된 함수부터 먼저 복귀해야 하므로 후입선출 방식의 스택이 유용하다. 올바르게는 '스택은 프로그램 실행 시 함수 호출과 복귀를 위한 처리에 유용하다'이다.
④ X — 인덱스를 이용하여 지정된 위치의 데이터를 직접 저장하거나 읽는 것은 배열의 특성이다. 올바르게는 '배열은 인덱스를 이용해 지정된 위치의 데이터에 직접 접근하는 자료구조이다'이다.
⑤ X — 하나의 노드가 최대 2개의 자식 노드를 가질 수 있는 구조는 이진 트리이다. 올바르게는 '이진 트리는 각 노드가 최대 2개의 자식 노드를 갖는 자료구조이다'이다.

오답 포인트

①의 LIFO와 ③의 함수 호출·복귀는 큐가 아니라 스택의 특징이다. 또한 ④는 배열, ⑤는 이진 트리의 정의이므로 서로 다른 자료구조의 대표 키워드를 큐의 특징으로 혼동하지 않아야 한다.

NoteBase

다시 나오면 이렇게 판단

삽입과 삭제의 순서를 먼저 확인한다. 먼저 들어온 것이 먼저 나오면 큐(FIFO), 나중에 들어온 것이 먼저 나오면 스택(LIFO)이다. 인덱스 직접 접근은 배열, 자식이 최대 2개인 계층 구조는 이진 트리로 판단한다.

암기 포인트

큐는 뒤에서 넣고 앞에서 뺀다: rear 삽입, front 삭제, FIFO이다.

NoteBase

6. 개발할 프로그램의 LOC(Line of Code)가 50,000이고 개발에 참여할 프로그래머가 5명, 각 프로그래머들의 평균 생산성이 1,000LOC/MM(Man Month)일 때 개발에 소요되는 기간은?

① 10개월 ② 15개월 ③ 20개월
④ 30개월 ⑤ 40개월

정답 1번

출제 포인트

프로그램 규모를 생산성으로 나누어 총 개발 노력량을 구한 뒤, 이를 참여 프로그래머 수로 나누어 개발 기간을 계산하는 문제이다.

조건 정리

주어진 값: 프로그램 규모는 50, 000*LOC*, 프로그래머는 5명, 1인당 평균 생산성은 1, 000*LOC*/*MM*이다.
구해야 할 값: 5명이 함께 개발할 때의 달력상 개발 기간이다.
주의할 조건: *MM*은 한 사람이 한 달 동안 투입하는 작업량인 맨먼스(Man-Month)를 뜻한다.

사용할 공식

총 개발 노력량=프로그램 규모1인당 생산성
개발 기간=총 개발 노력량프로그래머 수
생산성이 1인당 맨먼스 기준으로 주어졌으므로 먼저 전체 작업에 필요한 맨먼스를 구하고, 5명이 동시에 작업한다는 조건을 반영해 인원수로 나눈다.

계산 과정

1. 총 개발 노력량은 다음과 같다.

$$\frac{50,000LOC}{1,000LOC/MM} = 50MM$$

2. 5명이 투입되므로 개발 기간은 다음과 같다.

(50 MM)/(5)=10개월

따라서 개발에 소요되는 기간은 10개월이다.

NoteBase

선지 판단

① O — 계산 결과가 10개월이므로 옳다.
② X — 15개월은 계산 결과와 다르다. 50 MM을 5명으로 나누면 10개월이므로 올바른 기간은 10개월이다.
③ X — 20개월은 계산 결과와 다르다. 1인당 생산성만 적용하면 총 50 MM이 필요하고, 이를 5명으로 나누면 올바른 기간은 10개월이다.
④ X — 30개월은 계산 결과와 다르다. 팀의 월간 생산성은 $5 \times 1,000 = 5,000 LOC$이므로 $50,000 LOC$ 개발에는 올바르게는 10개월이 걸린다.
⑤ X — 40개월은 계산 결과와 다르다. 총 개발 노력량 50 MM을 참여 인원 5명으로 나눈 올바른 기간은 10개월이다.

오답 포인트

$50,000 LOC \div 1,000 LOC/MM = 50 MM$은 한 사람이 일할 때 필요한 총 노력량이지 5명이 함께 일할 때의 개발 기간이 아니다. 총 노력량을 구한 뒤 반드시 참여 인원 5명으로 한 번 더 나누어야 한다.

다시 나오면 이렇게 판단

프로그램 규모를 1인당 생산성으로 나누어 총 맨먼스를 구하고, 총 맨먼스를 투입 인원수로 나누어 달력상 개발 기간을 계산한다. 마지막에는 팀 전체의 월간 생산성으로 다시 계산해 단위를 검산한다.

암기 포인트

개발 기간은 규모÷(인원×1인당 생산성)이다.

NoteBase

7. 다음 자바 프로그램에서 ㉠에 넣을 수 있는 문장과 그 결과 ㉡의 문장에 의해 출력되는 결과가 모두 옳은 것은?

```
class Person {
    String name;
    public Person(String n) {  name = n;  }
    public void whoRU() {
        System.out.println(name+"입니다.");
    }
}
class Student extends Person {
    String school;
    public Student(String n, String s) {
        super(n);
        school = s;
    }
    public void whoRU() {
        System.out.println(school+"학교에  다니는  "+name+"입니
다.");
    }
}
public class People {
    public static void main(String args[]) {
        ㉠
        obj.whoRU();
    }       ㉡
}
```

① ㉠의 문장 : Person obj = new Student("김철수", "으뜸중");
㉡의 출력 : 김철수입니다.

② ㉠의 문장 : Student obj = new Person("김철수");
㉡의 출력 : 김철수입니다.

③ ㉠의 문장 : Person obj = new Student("김철수", "으뜸중");
㉡의 출력 : 으뜸중학교에 다니는 김철수입니다.

④ ㉠의 문장 : Student obj = new Student("김철수", "으뜸중");
㉡의 출력 : 김철수입니다.

⑤ ㉠의 문장 : Student obj = new Person("김철수");
㉡의 출력 : 출력이 되지 않음

정답 3번

출제 포인트

이 문제는 상속 관계에서 참조 변수에 객체를 대입할 수 있는 조건과, 오버라이딩된 메서드가 실제 객체의 타입에 따라 호출되는 동적 바인딩을 묻는 문제이다.

NoteBase

핵심 개념

업캐스팅: 자식 클래스의 객체를 부모 클래스 타입의 참조 변수로 가리키는 것으로, 묵시적으로 허용된다.
대입 가능 관계: Person 타입 변수는 Person 객체와 Student 객체를 가리킬 수 있지만, Student 타입 변수는 명시적 형 변환 없이 Person 객체를 가리킬 수 없다.
동적 바인딩: 참조 변수의 선언 타입이 Person이어도 실제 객체가 Student이면 오버라이딩된 Student의 whoRU()가 실행된다.
따라서 Person obj = new Student("김철수", "으뜸중");에서 obj.whoRU()를 호출하면 Student의 메서드가 실행되어 "으뜸중학교에 다니는 김철수입니다."가 출력된다.

선지별 해설

① X — Person obj = new Student("김철수", "으뜸중");는 올바른 업캐스팅이지만, 실제 객체가 Student이므로 Student의 whoRU()가 호출된다. 틀린 표현은 출력 결과인 "김철수입니다."이며, 올바르게는 "으뜸중학교에 다니는 김철수입니다."가 출력된다.
② X — Student 타입 변수에 부모 클래스인 Person 객체를 직접 대입할 수 없으므로 컴파일 오류가 발생한다. 틀린 부분은 Student obj = new Person("김철수");이며, 올바르게는 Person obj = new Person("김철수");로 선언하면 "김철수입니다."가 출력된다.
③ O — Person 타입 변수에 Student 객체를 대입하는 업캐스팅이 허용되고, 동적 바인딩에 따라 Student가 재정의한 whoRU()가 호출된다. 따라서 "으뜸중학교에 다니는 김철수입니다."가 출력된다.
④ X — Student obj = new Student("김철수", "으뜸중");는 올바른 문장이지만 Student의 whoRU()가 실행된다. 틀린 표현은 출력 결과인 "김철수입니다."이며, 올바르게는 "으뜸중학교에 다니는 김철수입니다."가 출력된다.
⑤ X — Student obj = new Person("김철수");는 부모 객체를 자식 타입 변수에 직접 대입한 문장이므로 ㉠에 넣을 수 있는 유효한 문장이 아니다. 컴파일 실패를 실행 결과인 "출력이 되지 않음"으로 볼 수 없다. 올바르게는 Person obj = new Person("김철수");로 선언하면 "김철수입니다."가 출력된다.

오답 포인트

①과 ④는 참조 변수의 선언 타입만 보고 Person의 whoRU()가 호출된다고 착각하게 만든다. 오버라이딩된 인스턴스 메서드는 실제 생성된 객체의 타입을 기준으로 호출된다. ⑤는 컴파일 실패와 정상 실행 후 출력이 없는 경우를 혼동하게 만든다.

다시 나오면 이렇게 판단

먼저 대입문의 왼쪽 참조 타입이 오른쪽 객체를 가리킬 수 있는지 확인한다. 컴파일 가능한 문장이면 new 뒤의 실제 객체 타입을 확인하고, 그 클래스가 오버라이딩한 메서드의 실행 결과를 판단한다.

NoteBase

암기 포인트

부모 변수로 자식 객체를 가리킬 수 있고, 오버라이딩 메서드는 실제 객체를 따라간다.

NoteBase

8. 다음 중 JSP(Java Server Page)에 대한 설명으로 옳은 것은?

① 클라이언트에서 동작하도록 만들어진 스크립트이다.
② 마이크로소프트 윈도 환경에서만 동작한다.
③ JSP는 서블릿 기술과는 배타적으로 사용한다.
④ 웹 서버에서 정적으로 웹 페이지를 생성할 수 있게 한다.
⑤ JSP는 HTML 코드 안에 자바 코드를 삽입하는 형태로 작성한다.

정답 5번

출제 포인트

JSP가 실행되는 위치, 운영체제 의존성, 서블릿과의 관계, 동적 웹 페이지 생성 방식 및 작성 형태를 구분하는 문제이다.

핵심 개념

JSP(JavaServer Pages)는 HTML 문서 안에 자바 기반의 서버 측 코드를 넣어 동적인 웹 콘텐츠를 만드는 기술이다. JSP는 웹 서버 측에서 처리되며, 실행 과정에서 서블릿으로 변환되어 동작한다. 따라서 JSP와 서블릿은 배타적인 기술이 아니라 서로 연계되는 기술이다.

선지별 해설

① X — 틀린 표현은 '클라이언트에서 동작'이다. JSP 코드는 웹 서버 측에서 실행되고, 처리 결과인 HTML 등이 클라이언트에 전달된다.
② X — 틀린 표현은 '마이크로소프트 윈도 환경에서만'이다. JSP는 자바 기반 기술이므로 자바 실행 환경과 JSP를 지원하는 서버가 있으면 특정 운영체제에 한정되지 않는다.
③ X — 틀린 표현은 '서블릿 기술과는 배타적으로 사용'이다. JSP는 실행 과정에서 서블릿으로 변환되므로 서블릿 기술을 기반으로 하며 상호 보완적으로 사용된다.
④ X — 틀린 표현은 '정적으로 웹 페이지를 생성'이다. JSP는 요청에 따라 서버에서 내용을 처리하여 동적인 웹 페이지를 생성하는 데 사용된다.
⑤ O — JSP는 HTML 코드 안에 자바 코드를 삽입하는 형태로 작성할 수 있는 서버 측 웹 기술이다.

오답 포인트

①의 '클라이언트', ②의 '윈도 환경에서만', ③의 '배타적', ④의 '정적'은 모두 JSP의 성격을 반대로 바꾼 표현이다. 특히 JSP 파일에 HTML이 많이 보인다는 이유로 클라이언트 측 기술이나 정적 페이지 기술로 판단하기 쉽지만, JSP 코드는 서버에서 처리된다.

NoteBase

다시 나오면 이렇게 판단

JSP 관련 선지는 ‘서버 측 실행’, ‘운영체제 비종속’, ‘서블릿으로 변환’, ‘동적 콘텐츠 생성’, ‘HTML 안에 자바 코드 삽입’이라는 기준으로 판단한다.

암기 포인트

JSP는 서버에서 실행되고 서블릿으로 변환되며, HTML 안에 자바 코드를 넣어 동적 웹 페이지를 만든다.

NoteBase

9. 다음 중 IEEE 802.11 무선랜 표준에 대한 설명으로 옳지 않은 것은?

① 종류에 따라 최대 전송속도가 달라지고 최대 1 Gbps이상 전송할 수 있는 표준도 존재한다.

② CSMA/CA 방식의 MAC 프로토콜을 사용한다.

③ 대부분의 경우 다수의 사용 가능 채널(channel)이 존재하여 근접한 거리에 있는 서로 다른 두 쌍의 컴퓨터가 서로 간섭을 받지 않고 동시에 통신할 수 있다.

④ 모든 AP(Access Point)는 유선랜 기술로 인터넷에 연결되어 있어야 어떤 특정 사용자가 해당 AP를 통해서 인터넷에 접속할 수 있다.

⑤ AP 없이도 서로 다른 두 컴퓨터가 통신할 수 있다.

정답 4번

출제 포인트

IEEE 802.11 무선랜의 전송 속도, 매체 접근 방식, 채널, AP의 역할, 애드혹 통신을 구분하는 문제이다.

핵심 개념

IEEE 802.11은 무선랜의 물리 계층과 MAC 계층을 규정하는 표준군이다. 무선 매체에서는 충돌을 미리 감지하기 어려우므로 CSMA/CA로 충돌을 회피하며, 통신 형태에는 AP를 거치는 인프라스트럭처 모드와 단말끼리 직접 통신하는 애드혹 모드가 있다. AP는 무선 단말을 다른 네트워크나 분배 시스템에 연결하는 접속점이지만, 반드시 유선 인터넷에 연결되어야 하거나 모든 사용자에게 접속을 허용해야 하는 장치는 아니다.

선지별 해설

① O — IEEE 802.11은 여러 세대의 표준으로 이루어지며 표준별 최대 전송 속도가 다르다. 일부 표준은 이론상 최대 전송 속도가 1*Gbps* 이상이다.

② O — IEEE 802.11의 MAC은 무선 매체의 충돌을 피하기 위해 CSMA/CA 방식을 사용한다.

③ O — 무선랜에는 복수의 채널이 있으며, 가까운 통신들이 서로 겹치지 않는 채널을 사용하면 간섭을 줄이고 동시에 통신할 수 있다.

④ X — 틀린 부분은 '모든 AP가 유선랜 기술로 인터넷에 연결되어 있어야 하고 어떤 특정 사용자도 접속할 수 있다'는 표현이다. AP는 인터넷 연결 없이 로컬 무선망만 구성할 수도 있고 다른 방식으로 상위망에 연결될 수도 있으며, 인증과 접근 제어를 통해 허용된 사용자만 접속하게 할 수도 있다. 올바르게는 'AP는 무선 단말에 네트워크 접속을 제공하지만 유선 인터넷 연결과 불특정 사용자의 자유로운 접속이 필수 조건은 아니다'이다.

⑤ O — 애드혹 모드에서는 AP 없이도 서로 다른 두 컴퓨터가 무선으로 직접 통신할 수 있다.

NoteBase

오답 포인트

④는 가정이나 공공장소에서 흔히 보는 '유선 인터넷 공유용 AP'를 모든 AP의 필수 형태로 일반화한 선지이다. '모든', '유선랜', '어떤 특정 사용자도'와 같은 절대적 표현을 함께 묶은 것이 함정이다.

다시 나오면 이렇게 판단

802.11 문제에서는 MAC 방식은 CSMA/CA, AP 경유는 인프라스트럭처 모드, AP 없는 직접 통신은 애드혹 모드로 구분한다. AP의 인터넷 연결 방식이나 사용자 공개 여부는 필수가 아니라 구성과 보안 정책에 따라 달라진다고 판단한다.

암기 포인트

802.11은 CSMA/CA, AP 없이 직접 통신하면 애드혹, AP는 인터넷·유선·공개 접속이 필수가 아니다.

NoteBase

10. 프로토콜은 컴퓨터 간 데이터 전송의 효율성과 신뢰성을 보장하기 위해 여러 가지 기능을 수행한다. 다음 중 프로토콜의 일반적인 기능에 해당하지 않는 것은?

① 주소 지정
② 오류 제어
③ 데이터 분할 및 조합
④ 비동기화
⑤ 흐름 제어 및 캡슐화

정답 4번

출제 포인트

프로토콜이 데이터 통신 과정에서 수행하는 일반적인 기능과 그에 해당하지 않는 개념을 구별하는 문제이다.

핵심 개념

프로토콜은 통신 주체 사이에서 데이터의 형식, 전송 순서, 오류 처리 방법 등을 정한 규칙의 집합이다. 대표 기능에는 주소 지정, 오류 제어, 데이터 분할 및 재조합, 동기화, 흐름 제어, 캡슐화 등이 있다. 비동기화는 이러한 일반 기능으로 분류되지 않으며, 통신 시점과 순서를 맞추는 동기화가 일반 기능이다.

선지별 해설

① O — 주소 지정은 송신자와 수신자를 식별하여 데이터가 올바른 목적지로 전달되게 하는 프로토콜의 일반 기능이다.
② O — 오류 제어는 전송 중 발생한 오류를 검출하고 필요하면 재전송이나 복구를 수행하여 신뢰성을 높이는 기능이다.
③ O — 데이터 분할 및 조합은 큰 데이터를 전송 가능한 단위로 나누고 수신 측에서 원래 데이터로 재조립하는 기능이다.
④ X — 틀린 표현은 '비동기화'이다. 프로토콜은 송수신 측의 전송 시점과 순서를 맞추는 '동기화'를 수행하며, 비동기화는 일반적인 프로토콜 기능이 아니다.
⑤ O — 흐름 제어는 송수신 속도 차이로 인한 데이터 손실을 막고, 캡슐화는 데이터에 주소나 제어 정보가 담긴 헤더 등을 덧붙이는 기능이다.

오답 포인트

④의 '비동기화'를 '비동기 통신 방식'과 혼동하기 쉽다. 프로토콜의 일반 기능을 묻는 문맥에서는 통신 주체 간 타이밍과 순서를 맞추는 '동기화'가 맞는 표현이다.

NoteBase

다시 나오면 이렇게 판단

각 용어가 데이터의 목적지 식별, 오류 복구, 단위 분할·재조립, 송수신 시점 조정, 전송량 조절, 제어 정보 부가 중 하나에 해당하는지 확인한다. ‘비동기화’처럼 통신의 일치를 깨뜨리는 표현은 일반 기능에서 제외한다.

암기 포인트

프로토콜은 주소·오류·분할·동기·흐름·캡슐화를 관리한다. 비동기화가 아니라 동기화이다.

NoteBase

11. 다음 중 프로세스 교착 상태의 해결 방안에 해당하지 않는 것은?

① Prevention
② Avoidance
③ Detection
④ Elimination
⑤ Recovery

정답 4번

출제 포인트

프로세스 교착 상태의 처리 방안을 예방, 회피, 탐지, 회복으로 구분할 수 있는지를 묻는 문제이다.

핵심 개념

예방(Prevention)은 교착 상태의 필요조건 중 하나가 성립하지 않도록 시스템을 설계하는 방법이다.
회피(Avoidance)는 자원 할당 상태를 검사하여 시스템이 안전 상태를 유지할 때만 자원을 할당하는 방법이다.
탐지(Detection)는 교착 상태의 발생을 허용한 뒤 발생 여부를 검사하는 방법이고, 회복(Recovery)은 탐지된 교착 상태를 프로세스 종료나 자원 선점·복귀 등으로 해소하는 방법이다.
제거(Elimination)는 이 문항에서 묻는 교착 상태 처리 방안의 표준 분류에 포함되지 않는다.

선지별 해설

① O — Prevention은 교착 상태의 성립 조건을 미리 차단하는 예방 기법이므로 해결 방안에 해당한다.
② O — Avoidance는 자원 할당 전에 안전 상태 여부를 판단하여 교착 상태를 피하는 회피 기법이므로 해결 방안에 해당한다.
③ O — Detection은 교착 상태 발생을 허용한 뒤 이를 검사해 찾아내는 탐지 기법이므로 처리 방안에 해당한다.
④ X — Elimination은 교착 상태 처리 방안의 표준 분류가 아니다. 틀린 표현은 'Elimination'이며, 올바른 표준 분류는 Prevention, Avoidance, Detection, Recovery이다.
⑤ O — Recovery는 탐지된 교착 상태에서 프로세스를 종료하거나 자원을 선점하는 등의 방식으로 정상 상태를 복구하는 기법이므로 해결 방안에 해당한다.

오답 포인트

일상적인 의미에서 elimination도 '제거'이므로 교착 상태 해결책처럼 보이지만, 운영체제의 표준 분류 용어는 예방, 회피, 탐지, 회복이다. 특히 탐지는 단독으로 끝나는 것이 아니라 탐지 후 회복과 연결된다는 점도 함께 구분한다.

NoteBase

다시 나오면 이렇게 판단

교착 상태 문제에서는 발생 조건을 사전에 막으면 예방, 안전 상태만 유지하면 회피, 발생 후 찾아내면 탐지, 찾아낸 뒤 해소하면 회복으로 판단한다. 이 네 범주에 없는 일반적 표현은 표준 분류인지 확인한다.

암기 포인트

교착 상태 처리의 네 갈래는 예방·회피·탐지·회복이다.

NoteBase

12. 다음 SQL 명령문들의 실행 후 상황에 대한 설명으로 옳은 것은?

```
CREATE TABLE UWORDS (ID INTEGER PRIMARY KEY, UWORD CHAR(5), FREQ INTEGER);
INSERT INTO UWORDS VALUES (500, 'THIS', 500);
INSERT INTO UWORDS VALUES (510, 'IS', 600);
INSERT INTO UWORDS VALUES (520, 'TEST', 700);
SELECT UWORD FROM UWORDS WHERE ID > 500;
DELETE FROM UWORDS WHERE FREQ < 600;
COMMIT;
```

① UWORDS 테이블의 레코드(record)의 개수는 3개이다.
② 3개의 레코드가 출력된다.
③ 출력 결과에서 700이란 숫자는 보이지 않는다.
④ UWORDS 테이블의 컬럼(column)의 개수는 2개이다.
⑤ UWORDS 테이블에 TEST라는 단어는 저장되어 있지 않다.

정답 3번

출제 포인트

이 문제는 SQL 명령문의 실행 순서에 따라 SELECT의 출력 결과와 DELETE 후 테이블에 남는 레코드 수를 구분할 수 있는지를 묻는다.

핵심 개념

INSERT는 테이블에 레코드를 추가하고, SELECT는 조건에 맞는 레코드에서 지정한 컬럼만 조회하며, DELETE는 조건에 맞는 레코드를 삭제한다.
세 번의 INSERT 직후에는 (500, 'THIS', 500), (510, 'IS', 600), (520, 'TEST', 700)의 3개 레코드가 있다.
SELECT UWORD FROM UWORDS WHERE ID > 500;은 ID가 510과 520인 레코드의 UWORD만 조회하므로 IS와 TEST, 총 2개 값이 출력된다.
그 뒤 DELETE FROM UWORDS WHERE FREQ < 600;은 FREQ가 500인 THIS 레코드만 삭제하므로 COMMIT 후에는 IS와 TEST의 2개 레코드가 남는다.

NoteBase

선지별 해설

① X — 모든 명령 실행 후에는 FREQ가 600과 700인 레코드만 남으므로 레코드 수는 3개가 아니라 2개이다. 틀린 표현 '3개이다'는 '2개이다'로 고쳐야 한다.
② X — SELECT의 WHERE 조건 ID > 500을 만족하는 레코드는 ID 510과 520의 2개이므로 3개가 아니라 2개의 UWORD 값이 출력된다. 틀린 표현 '3개의 레코드가 출력된다'는 '2개의 값인 IS와 TEST가 출력된다'로 고쳐야 한다.
③ O — SELECT 절에는 UWORD만 지정되어 있으므로 출력 결과에는 IS와 TEST만 나타난다. 700은 TEST 레코드의 FREQ 값이지만 FREQ는 조회 대상 컬럼이 아니므로 출력되지 않는다.
④ X — UWORDS 테이블은 ID, UWORD, FREQ로 정의되어 있으므로 컬럼 수는 2개가 아니라 3개이다. 틀린 표현 '2개이다'는 '3개이다'로 고쳐야 한다.
⑤ X — TEST 레코드의 FREQ는 700이어서 FREQ < 600이라는 삭제 조건을 만족하지 않으므로 삭제되지 않고 저장되어 있다. 틀린 표현 '저장되어 있지 않다'는 '저장되어 있다'로 고쳐야 한다.

오답 포인트

SELECT의 출력 내용과 DELETE 이후 테이블의 최종 상태를 섞어 판단하기 쉽다. 특히 ③의 700은 WHERE 조건 판정에는 사용되는 값이 아니라 조회된 레코드의 FREQ 값이며, SELECT 목록이 UWORD 하나뿐이므로 화면에 나타나지 않는다.

다시 나오면 이렇게 판단

SQL 문을 위에서 아래로 실행하면서 먼저 각 행의 변화를 추적한 뒤, SELECT에서는 WHERE가 고르는 행과 SELECT 절이 보여 주는 열을 따로 확인한다. 마지막으로 DELETE 조건을 적용하여 테이블에 실제로 남는 행을 계산한다.

암기 포인트

WHERE는 행을 고르고, SELECT 목록은 출력할 열을 고르며, DELETE는 조건에 맞는 행 자체를 없앤다.

NoteBase

13. 다음 중 가상기억장치에 대한 설명으로 옳은 것은?

① 프로세스의 주소공간 전체가 주기억장치에 적재된다.
② 실행 중인 프로세스들에 필요한 전체 주소공간 크기의 합은 주기억장치 용량보다 클 수 없다.
③ 프로세스에서 사용되는 가상주소의 순서와 주기억장치의 물리주소 순서는 일치한다.
④ 페이징 기법에서 페이지의 크기는 프로그램의 모듈 단위로 정해지므로 모두 다르다.
⑤ 프로세스에서 사용되는 가상주소를 주기억장치의 물리주소로 변환하는 것은 프로세스의 실행 중에 이루어진다.

정답 5번

출제 포인트

가상기억장치에서 프로세스의 가상주소 공간이 주기억장치의 물리주소 공간에 대응되는 방식을 묻는 문제이다.

핵심 개념

가상기억장치는 보조기억장치를 이용해 프로세스에 실제 주기억장치보다 큰 논리적 주소 공간을 제공하는 기법이다. 실행에 필요한 페이지 일부만 주기억장치에 적재할 수 있으며, 가상주소는 페이지 테이블 등을 통해 물리주소로 변환된다.
페이징은 가상주소 공간을 일정한 크기의 페이지로, 물리 메모리를 같은 크기의 프레임으로 나누어 대응시키는 방식이다. 페이지는 연속된 물리 프레임에 놓일 필요가 없다.

선지별 해설

① X — '주소공간 전체가 주기억장치에 적재된다'는 부분이 틀리다. 올바르게는 실행에 필요한 페이지들만 주기억장치에 적재되고, 나머지는 필요할 때 적재될 수 있다.
② X — '전체 주소공간 크기의 합은 주기억장치 용량보다 클 수 없다'는 부분이 틀리다. 올바르게는 가상기억장치를 사용하므로 실행 중인 프로세스들의 가상주소 공간 합이 주기억장치 용량보다 클 수 있다.
③ X — '가상주소의 순서와 물리주소의 순서가 일치한다'는 부분이 틀리다. 올바르게는 각 가상 페이지가 임의의 물리 프레임에 대응될 수 있으므로 두 주소의 배치 순서는 일치할 필요가 없다.
④ X — '페이지의 크기는 프로그램의 모듈 단위로 정하므로 모두 다르다'는 부분이 틀리다. 올바르게는 페이징의 페이지 크기는 시스템에서 정한 일정한 크기이며, 프로그램의 논리적 모듈에 따라 가변 크기로 나누는 것은 세그먼테이션의 특징이다.
⑤ O — 프로세스가 사용하는 가상주소는 명령 실행 중 주소 변환 하드웨어와 페이지 테이블을 통해 주기억장치의 물리주소로 변환된다.

NoteBase

오답 포인트

①의 '전체 적재'와 ②의 '주기억장치 용량을 초과할 수 없음'은 가상기억장치의 목적과 반대되는 표현이다. 또한 ④는 고정 크기인 페이징과 논리적 모듈별 가변 크기인 세그먼테이션을 바꾸어 놓은 선지이다.

다시 나오면 이렇게 판단

가상기억장치 문제에서는 '필요한 부분만 적재', '가상주소 공간은 물리 메모리보다 클 수 있음', '페이지와 프레임은 고정 크기', '가상주소와 물리주소의 배치는 불연속 가능', '실행 중 주소 변환'의 다섯 기준으로 판단한다.

암기 포인트

가상은 크게, 적재는 일부만, 페이지는 고정 크기, 배치는 불연속, 주소 변환은 실행 중에 이루어진다.

NoteBase

14. 다음 중 모바일 데이터베이스의 특징 또는 종류에 해당하지 않는 것은?

① 클라이언트 측 데이터베이스의 복제 및 비동기화 기능
② 저성능 CPU와 제한된 주기억장치를 가진 모바일 기기에 탑재 가능
③ 내장형 데이터베이스
④ SQLite
⑤ DB2 Everyplace

정답 1번

출제 포인트

모바일 데이터베이스가 모바일 환경에서 갖추어야 할 기능과 대표적인 구현 종류·제품을 구별하는 문제이다.

핵심 개념

모바일 데이터베이스는 처리 능력과 메모리가 제한된 모바일 기기에서 데이터를 저장·관리하는 데이터베이스이다. 네트워크 연결이 불안정하거나 일시적으로 끊겨도 로컬 데이터를 이용할 수 있도록 복제 기능을 활용하며, 다시 연결되면 서버와 데이터의 일관성을 맞추는 동기화 기능이 필요하다. 내장형 데이터베이스는 응용 프로그램이나 기기에 포함되어 별도의 대형 데이터베이스 서버 없이 동작하는 형태이다.

선지별 해설

① X — 데이터베이스 복제는 모바일 데이터베이스의 관련 기능이지만, '비동기화 기능'이라는 표현이 틀리다. 복제된 클라이언트 데이터와 서버 데이터의 일관성을 맞추는 동기화 기능이라고 해야 한다. 통신이 비동기 방식으로 처리될 수 있다는 것과 데이터를 비동기화한다는 것은 다른 뜻이다.
② O — 모바일 데이터베이스는 저성능 CPU와 제한된 주기억장치를 가진 모바일 기기의 자원 제약을 고려해 탑재할 수 있도록 설계된다.
③ O — 내장형 데이터베이스는 응용 프로그램이나 모바일 기기에 포함되어 동작하는 데이터베이스 형태로, 모바일 데이터베이스의 종류에 해당한다.
④ O — SQLite는 서버 프로세스 없이 응용 프로그램에 내장되어 동작하는 경량 데이터베이스이므로 모바일 환경에서 널리 사용하는 모바일 데이터베이스에 해당한다.
⑤ O — DB2 Everyplace는 모바일·내장 환경을 대상으로 제공된 경량 데이터베이스 제품이므로 모바일 데이터베이스의 예에 해당한다.

오답 포인트

①의 '비동기화'를 '비동기 방식의 동기화'로 잘못 읽기 쉽다. 이 선지는 동기화 앞에 방식이 붙은 것이 아니라 기능 자체를 '비동기화'라고 표현했으므로 틀린 선지이다.

NoteBase

다시 나오면 이렇게 판단

모바일 데이터베이스 선지는 자원 제약에 적합한지, 로컬 저장·복제를 지원하는지, 서버와 데이터 일관성을 맞추는 동기화를 지원하는지의 순서로 판단한다. 제품명은 SQLite와 DB2 Everyplace처럼 경량·내장·모바일 환경용인지 확인한다.

암기 포인트

모바일 DB의 핵심은 경량·내장·복제·동기화이다. '비동기화'가 아니라 '동기화'이다.

NoteBase

15. 다음 중 UDP와 TCP 모두 해당하는 것은?

① 전송 중 손실된 데이터는 재전송을 통하여 복구한다.
② 수신단에서 수신된 데이터를 송신자가 보낸 순서에 일치하도록 재정렬한다.
③ 헤더(header) 부분에 체크섬(checksum)필드가 있다.
④ 네트워크의 혼잡 시 전송 속도를 줄인다.
⑤ 전송된 데이터의 수신 여부를 확인할 수 있다.

정답 3번

출제 포인트

이 문제는 UDP와 TCP의 전송 신뢰성 기능을 구분하고, 두 프로토콜의 헤더에 공통으로 존재하는 필드를 찾는 문제이다.

핵심 개념

TCP: 연결형 전송 계층 프로토콜로, 순서 제어·확인 응답·재전송·흐름 제어·혼잡 제어를 통해 신뢰성 있는 전송을 제공한다.
UDP: 비연결형 전송 계층 프로토콜로, 기본적으로 순서 보장·확인 응답·재전송·혼잡 제어를 제공하지 않고 오버헤드가 작다.
공통점: TCP 헤더와 UDP 헤더에는 모두 출발지 포트, 목적지 포트, 체크섬 필드가 있다.

선지별 해설

① X — '손실된 데이터를 재전송하여 복구한다'는 TCP의 기능이다. UDP는 손실을 감지한 뒤 자동으로 재전송하는 기능을 제공하지 않는다. 올바르게는 'TCP는 전송 중 손실된 데이터를 재전송하여 복구한다'이다.
② X — '송신자가 보낸 순서에 일치하도록 재정렬한다'는 TCP의 순서 제어 기능이다. UDP는 데이터그램의 도착 순서를 보장하거나 자동으로 재정렬하지 않는다. 올바르게는 'TCP는 순서 번호를 이용하여 데이터를 송신 순서에 맞게 재정렬한다'이다.
③ O — TCP 헤더와 UDP 헤더에는 모두 전송 오류 검사를 위한 체크섬 필드가 있다. 따라서 두 프로토콜 모두에 해당한다.
④ X — 네트워크 혼잡을 감지하여 전송 속도를 조절하는 혼잡 제어는 TCP의 기능이다. UDP 자체에는 이러한 혼잡 제어 기능이 없다. 올바르게는 'TCP는 네트워크 혼잡 시 전송 속도를 줄인다'이다.
⑤ X — 확인 응답을 통해 데이터 수신 여부를 확인하는 기능은 TCP가 제공한다. UDP는 확인 응답 기능이 없어 수신 여부를 자체적으로 확인하지 않는다. 올바르게는 'TCP는 전송한 데이터의 수신 여부를 확인할 수 있다'이다.

NoteBase

오답 포인트

①·②·④·⑤는 모두 TCP의 신뢰성 또는 제어 기능을 UDP에도 적용되는 것처럼 표현한 선지이다. ③은 기능의 동작 방식이 아니라 헤더 필드의 존재 여부를 묻는다는 점이 함정이다.

다시 나오면 이렇게 판단

재전송·순서 보장·확인 응답·혼잡 제어가 나오면 TCP의 기능으로 판단한다. 두 프로토콜의 공통점을 물으면 포트 번호와 체크섬 같은 헤더 필드를 확인한다.

암기 포인트

TCP는 신뢰성·순서·제어, UDP는 단순·신속, 체크섬 필드는 둘 다 가진다.

NoteBase

16. 다음 중 입출력 위주의 프로세스와 연산 위주 프로세스의 특성에 따라 CPU 사용 시간(할당량)을 다르게 부여하는 선점 방식의 CPU 스케줄링 기법으로 옳은 것은?

① Round-Robin
② Multi-level Feedback Queue
③ Shortest Job First
④ Highest Response ratio Next
⑤ Deadline

정답 2번

출제 포인트

입출력 위주 프로세스와 연산 위주 프로세스의 실행 특성에 따라 우선순위와 CPU 시간 할당량을 다르게 조정하는 선점형 스케줄링 기법을 구별하는 문제이다.

핵심 개념

다단계 피드백 큐(MLFQ)는 우선순위가 다른 여러 준비 큐를 두고, 프로세스의 CPU 사용 행태에 따라 큐 사이를 이동시키는 선점형 스케줄링 방식이다. 일반적으로 상위 큐에는 짧은 시간 할당량을 주어 대화형·입출력 위주 프로세스의 응답성을 높이고, CPU를 오래 사용하는 프로세스는 하위 큐로 내려 더 긴 시간 할당량을 부여한다. Round-Robin은 모든 준비 프로세스에 동일한 시간 할당량을 순환 적용하고, SJF·HRRN·Deadline은 각각 CPU 버스트 길이·응답률·마감 시각을 선택 기준으로 삼는다.

NoteBase

선지별 해설

① X — Round-Robin은 준비 큐의 프로세스에 원칙적으로 같은 시간 할당량을 차례로 부여한다. 틀린 부분은 '프로세스의 특성에 따라 CPU 시간 할당량을 다르게 부여한다'는 점이며, 올바르게는 동일한 시간 할당량을 순환 적용하는 방식이다.
② O — Multi-level Feedback Queue는 프로세스의 CPU 사용 행태를 반영해 우선순위 큐를 이동시키고, 큐의 단계에 따라 서로 다른 시간 할당량을 적용할 수 있는 선점형 방식이다. 따라서 입출력 위주 프로세스의 빠른 응답과 연산 위주 프로세스의 긴 실행을 함께 조정하기에 알맞다.
③ X — Shortest Job First는 예상 CPU 버스트가 가장 짧은 작업을 먼저 선택하는 방식이다. 틀린 부분은 '특성에 따라 시간 할당량을 다르게 부여한다'는 점이며, 올바르게는 예상 실행 시간이 가장 짧은 작업을 우선 실행하는 방식이다.
④ X — Highest Response Ratio Next는 대기 시간과 서비스 시간을 반영한 응답률이 가장 높은 작업을 선택하는 비선점형 방식이다. 틀린 부분은 '서로 다른 시간 할당량을 주는 선점 방식'이라는 점이며, 올바르게는 응답률을 기준으로 다음 작업을 선택하는 방식이다.
⑤ X — Deadline 스케줄링은 실시간 작업의 마감 시각을 기준으로 실행 순서를 정하는 방식이다. 틀린 부분은 '입출력·연산 특성별 시간 할당량 조정'이라는 점이며, 올바르게는 마감 기한 충족을 우선하여 작업을 선택하는 방식이다.

오답 포인트

① Round-Robin도 시간 할당량을 사용하는 선점형 방식이므로 ②와 혼동하기 쉽다. 그러나 ①은 동일한 시간 할당량의 순환이 핵심이고, ②는 여러 우선순위 큐와 큐별로 다른 시간 할당량 및 피드백에 따른 큐 이동이 핵심이다.

다시 나오면 이렇게 판단

문제에 '입출력 위주와 연산 위주를 구분', '여러 우선순위 큐', '서로 다른 시간 할당량', '실행 결과에 따른 큐 이동'이 함께 나타나면 다단계 피드백 큐로 판단한다. 동일한 시간 할당량을 차례로 주는 방식은 Round-Robin으로 구별한다.

암기 포인트

RR은 같은 시간으로 순환하고, MLFQ는 피드백으로 큐를 이동하며 큐마다 시간을 달리한다.

NoteBase

17. 다음 중 시스템버스에 대한 설명으로 옳지 않은 것은?

① 하드웨어 구성요소를 물리적으로 연결하며 구성요소 사이의 데 이터 통로를 제공한다.
② 주소버스는 중앙처리장치가 주기억장치나 입출력장치에 데이터를 읽거나 쓰기 위해 필요한 주소를 전달하는 통로이다.
③ 제어버스는 주소버스와 데이터버스의 동작을 제어하기 위한 신호의 전달 통로이다.
④ 데이터버스는 중앙처리장치와 기타 모듈(기억장치, 입출력장치 등) 사이의 데이터를 전달하는 통로로 양방향 버스이다.
⑤ 시스템버스는 용도에 따라 주소버스, 입출력버스, 데이터버스, 제어버스로 구성된다.

정답 5번

출제 포인트

시스템버스를 구성하는 주소버스·데이터버스·제어버스의 역할과 전송 방향을 구별하는 문제이다.

핵심 개념

시스템버스는 중앙처리장치, 기억장치, 입출력장치 사이에서 주소·데이터·제어 신호를 전달하는 공통 통로이다.
주소버스: 접근할 기억장치나 입출력장치의 주소를 전달하며, 일반적으로 중앙처리장치에서 외부 장치로 향한다.
데이터버스: 실제 데이터를 전달하며, 읽기와 쓰기를 모두 수행하므로 양방향이다.
제어버스: 읽기·쓰기, 인터럽트 등 각 장치의 동작을 조정하는 제어 신호를 전달한다.
따라서 시스템버스의 구성은 주소버스·데이터버스·제어버스이며, 입출력버스는 이 세 구성요소 중 하나가 아니다.

선지별 해설

① O — 시스템버스는 하드웨어 구성요소를 물리적으로 연결하고 구성요소 사이에 데이터와 신호가 이동하는 통로를 제공한다.
② O — 주소버스는 중앙처리장치가 기억장치나 입출력장치에 접근할 때 대상의 주소를 전달하는 통로이다.
③ O — 제어버스는 주소버스와 데이터버스를 통한 전송이 올바르게 이루어지도록 읽기·쓰기 등의 제어 신호를 전달한다.
④ O — 데이터버스는 중앙처리장치와 기억장치·입출력장치 사이에서 실제 데이터를 전달하며, 읽기와 쓰기가 모두 필요하므로 양방향 버스이다.
⑤ X — 틀린 표현은 '주소버스, 입출력버스, 데이터버스, 제어버스'로 구성된다는 부분이다. 올바르게는 시스템버스는 용도에 따라 주소버스, 데이터버스, 제어버스로 구성된다.

NoteBase

오답 포인트

⑤는 입출력장치가 시스템버스에 연결된다는 사실을 이용해 '입출력버스'를 시스템버스의 구성요소처럼 끼워 넣었다. 연결 대상과 버스의 기능별 구성요소를 혼동하면 안 된다.

다시 나오면 이렇게 판단

각 버스가 무엇을 전달하는지 기준으로 판단한다. 위치를 지정하면 주소버스, 실제 값을 옮기면 데이터버스, 동작을 지시하거나 조정하면 제어버스이다. 구성요소를 묻는 경우에는 주소·데이터·제어의 세 가지인지 확인한다.

암기 포인트

시스템버스는 '주소로 위치 지정, 데이터로 값 전달, 제어로 동작 조정'의 세 가지이다.

NoteBase

18. 다음 C 언어로 작성된 함수는 정수 배열의 원소들의 총 합을 구하여 리턴하는 함수이다. ㉠의 위치에 들어가야 할 코드는?
(단, 함수의 첫 번째 인자는 배열의 시작 위치를 나타내는 포인터, 두 번째 인자는 배열의 크기임)

```
int sum(int* data, const int dsize)
{
        if (dsize > 0)
                return ㉠;
        return 0;
}
```

① *data + sum(data - 1, dsize + 1)
② data + sum(data - 1, dsize + 1)
③ *data + sum(data - 1, dsize - 1)
④ *data + sum(data + 1, dsize - 1)
⑤ data + sum(data + 1, dsize - 1)

정답 4번

출제 포인트

재귀 호출로 배열의 합을 구할 때 현재 원소를 더하고, 다음 원소로 이동하며, 남은 원소 수를 하나씩 줄이는 구조를 판단하는 문제이다.

조건 정리

data는 현재 배열 원소를 가리키는 포인터이고, *data는 그 원소의 정숫값이다. dsize는 아직 합산할 원소의 수이다. dsize가 0이면 0을 반환하므로, 재귀 호출마다 dsize가 1씩 감소해야 한다.

사용할 공식

배열 합의 재귀식은 sum(data, dsize) = *data + sum(data + 1, dsize - 1)이다. 현재 원소 *data를 합에 포함한 뒤 data + 1로 다음 원소를 가리키고, 처리할 원소 수 dsize를 1 줄여야 하기 때문이다.

계산 과정

1. 현재 원소의 값은 *data로 읽는다.
2. 다음 재귀 호출은 data + 1을 전달하여 다음 배열 원소로 이동한다.
3. 이미 한 원소를 처리했으므로 dsize - 1을 전달한다.
4. dsize가 0이 되면 0을 반환하여 재귀 호출이 종료된다.

NoteBase

선지 판단

① X — *data로 현재 값을 더하는 부분은 맞지만, data - 1은 배열 시작 위치의 앞쪽으로 이동하고 dsize + 1은 크기를 늘려 종료 조건에 도달하지 못하게 한다. 올바르게는 *data + sum(data + 1, dsize - 1)이다.
② X — data는 값이 아니라 int*형 포인터이므로 현재 원소의 합에는 *data를 사용해야 한다. 또한 data - 1과 dsize + 1은 이동 방향과 크기 변화가 모두 반대이다. 올바르게는 *data + sum(data + 1, dsize - 1)이다.
③ X — dsize - 1로 종료 조건에는 접근하지만, data - 1은 다음 원소가 아니라 배열 시작 위치의 앞쪽을 가리키게 한다. 올바르게는 *data + sum(data + 1, dsize - 1)이다.
④ O — *data로 현재 원소를 더하고, data + 1로 다음 원소로 이동하며, dsize - 1로 남은 원소 수를 줄이므로 모든 배열 원소의 합을 구한다.
⑤ X — data는 int*형 포인터이므로 data 자체를 정수 합에 더하면 현재 원소의 값이 합산되지 않고 식의 결과도 포인터형이 된다. 올바르게는 data가 아니라 *data를 사용한 *data + sum(data + 1, dsize - 1)이다.

오답 포인트

핵심 함정은 data와 *data의 구분, 그리고 재귀 호출의 진행 방향이다. data는 주소이고 *data는 그 주소에 저장된 값이다. 배열을 앞에서 뒤로 처리하려면 포인터는 data + 1로 이동하고, 종료를 위해 크기는 dsize - 1로 줄어야 한다.

다시 나오면 이렇게 판단

재귀 배열 처리식은 '현재 값 처리 → 포인터를 다음 원소로 이동 → 남은 개수 감소 → 0에서 종료'의 네 조건을 차례로 확인한다.

암기 포인트

배열의 재귀 합은 *data로 값 읽기, data + 1로 전진, dsize - 1로 종료이다.

NoteBase

19. 다음 중 자바의 예외처리에 대한 설명으로 옳은 것은?

① throw는 처리가 정상적으로 이루어졌음을 알리는 명령이다.
② 예외가 발생할 가능성이 있는 문장은 try 블록에 넣는다.
③ catch 블록에는 예외 발생 여부에 관계없이 실행할 문장을 넣는다.
④ 발생한 예외에 대한 처리는 finally 블록에서 할 수 없다.
⑤ 예외가 발생하였으나 예외 처리가 이루어지지 않은 경우 그 예외는 무시되고, 프로그램은 예외 발생 이후 지점부터 계속된다.

정답 2번

출제 포인트

이 문제는 자바 예외 처리에서 try, catch, finally, throw의 역할과 처리되지 않은 예외의 흐름을 구분하는 문제이다.

핵심 개념

try 블록은 예외가 발생할 수 있는 문장을 실행하는 영역이다.
catch 블록은 try 블록에서 발생하여 전달된 예외 중 매개변수의 예외 형식과 일치하는 예외를 처리하는 영역이다.
finally 블록은 예외 발생 여부와 관계없이 실행되는 마무리 영역이며, 자원 해제와 같은 정리 작업에 주로 사용된다.
throw 문은 예외 객체를 명시적으로 발생시키며, 정상 처리가 완료되었음을 알리는 명령이 아니다.

선지별 해설

① X — throw는 예외 객체를 명시적으로 발생시키는 문장이다. 틀린 표현은 '처리가 정상적으로 이루어졌음을 알리는 명령'이며, 올바르게는 '예외를 명시적으로 발생시키는 명령'이다.
② O — 예외가 발생할 가능성이 있는 문장은 try 블록에 두고 실행하며, 발생한 예외는 이어지는 catch 블록에서 형식에 맞게 처리할 수 있다.
③ X — catch 블록은 예외 발생 여부와 관계없이 항상 실행되는 영역이 아니다. 틀린 표현은 '예외 발생 여부에 관계없이'이며, 올바르게는 '처리할 수 있는 예외가 try 블록에서 발생하여 전달된 경우'에 실행할 문장을 넣는다. 예외 발생 여부와 관계없이 실행할 문장은 finally 블록에 둔다.
④ X — finally 블록에도 예외 발생 뒤 수행할 정리·후속 처리 문장을 둘 수 있으므로 '할 수 없다'라는 단정이 틀렸다. 올바르게는 특정 예외를 받아 처리하는 주된 역할은 catch 블록이 담당하고, finally 블록은 예외 발생 여부와 관계없이 정리 작업을 수행한다.
⑤ X — 처리되지 않은 예외는 무시되지 않으며 현재 메서드의 호출자 쪽으로 전파된다. 끝까지 처리되지 않으면 해당 스레드가 비정상 종료되므로, '예외 발생 이후 지점부터 계속된다'가 아니라 정상 실행 흐름이 중단된다고 고쳐야 한다.

NoteBase

오답 포인트

③의 '예외 발생 여부에 관계없이'는 catch가 아니라 finally의 성격이다. 또한 ④는 finally가 특정 예외를 잡는 catch 블록은 아니라는 사실을 '예외와 관련된 어떠한 처리도 할 수 없다'는 뜻으로 지나치게 단정한 표현이다.

다시 나오면 이렇게 판단

예외 가능 문장은 try, 발생한 특정 예외의 처리는 catch, 발생 여부와 무관한 마무리는 finally, 예외의 명시적 발생은 throw로 구분한다. 처리되지 않은 예외는 무시되는 것이 아니라 호출자 쪽으로 전파된다는 기준을 적용한다.

암기 포인트

try는 감시, catch는 포착·처리, finally는 항상 마무리, throw는 예외 발생이다.

NoteBase

20. 다음 중 소프트웨어 개발 생명주기의 대표적인 모델에 대한 설명으로 옳지 않은 것은?

① 프로토타입 모델, 폭포수 모델, 익스트림 프로그래밍 모델, 나선형 모델 등이 있다.
② 가장 전통적인 방법은 폭포수 모델이다.
③ 프로토타입 모델은 사용자의 의견을 중요하게 여긴다.
④ 최근에 등장한 소규모 소프트웨어 개발에 유리한 것은 익스트림 프로그래밍 모델이다.
⑤ 시간과 비용이 적게 들며, 위험 요인을 사전에 분석하여 제거하거나 낮출 수 있는 것은 나선형 모델이다.

정답 5번

출제 포인트

소프트웨어 개발 생명주기 모델별 핵심 특징과 적용 상황을 구분하는 문제이다.

핵심 개념

폭포수 모델은 요구분석부터 유지보수까지 단계를 순차적으로 진행하는 전통적 모델이다. 프로토타입 모델은 시제품을 빠르게 만들어 사용자의 요구와 의견을 확인·반영한다. 익스트림 프로그래밍 모델은 짧은 반복 개발과 잦은 피드백을 중시하여 요구 변화가 잦은 소규모 개발에 적합하다. 나선형 모델은 반복 개발 과정마다 위험을 분석하고 대응하는 위험 중심 모델이지만, 절차가 복잡하여 시간과 비용이 많이 들 수 있다.

선지별 해설

① O — 프로토타입 모델, 폭포수 모델, 익스트림 프로그래밍 모델, 나선형 모델은 모두 대표적인 소프트웨어 개발 생명주기 모델에 해당한다.
② O — 폭포수 모델은 개발 단계를 순서대로 진행하는 가장 전통적인 생명주기 모델이다.
③ O — 프로토타입 모델은 사용자가 시제품을 확인하고 의견을 제시하도록 하여 요구사항을 구체화하므로 사용자의 의견을 중요하게 여긴다.
④ O — 익스트림 프로그래밍 모델은 짧은 주기의 반복 개발과 고객 피드백을 중시하므로 소규모 소프트웨어 개발과 변화가 잦은 환경에 유리하다.
⑤ X — 틀린 표현은 '시간과 비용이 적게 든다'이다. 나선형 모델은 반복적인 위험 분석과 개발을 수행하므로 위험을 사전에 줄일 수 있지만, 절차가 복잡하여 시간과 비용이 많이 들 수 있다.

오답 포인트

⑤는 '위험 요인을 사전에 분석하여 제거하거나 낮출 수 있다'는 나선형 모델의 장점에 '시간과 비용이 적게 든다'는 잘못된 특징을 결합하였다. 문장의 일부가 맞더라도 모든 내용이 맞아야 옳은 선지이다.

NoteBase

다시 나오면 이렇게 판단

각 모델을 대표 키워드와 연결한다. 순차 진행은 폭포수, 시제품과 사용자 의견은 프로토타입, 짧은 반복과 고객 피드백은 익스트림 프로그래밍, 반복적 위험 분석은 나선형으로 판단한다. 나선형에서 위험 감소와 시간·비용 증가는 함께 기억한다.

암기 포인트

폭포수는 순차, 프로토타입은 시제품, XP는 짧은 반복, 나선형은 위험 분석과 높은 비용이다.

NoteBase