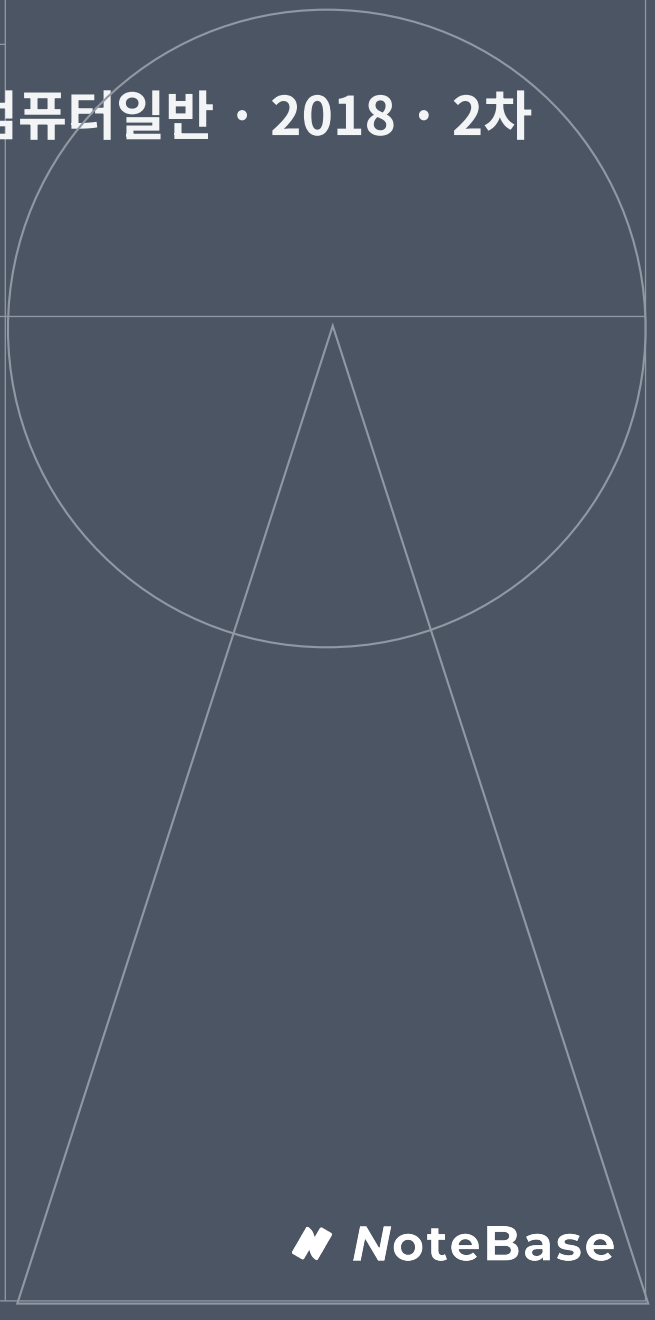
기출OX 해설집

공무원/해경/일반직 9급 · 컴퓨터일반 · 2018 · 2차



NoteBase

기출OX 해설집

공무원/해경/일반직 9급 · 컴퓨터일반 · 2018 · 2차

공무원/해경/일반직 9급 · 컴퓨터일반 · 2018 · 2차

문 1. 다음은 CPU스케줄링 기법에 대한 설명이다. 기아(starvation)현상이 발생할 수 있는 스케줄링 기법은 몇 개인가?

ㄱ. 프로세스마다 우선순위를 부여하여 높은 순서대로 작업을 처리하는 것이다.
ㄴ. 작업들 가운데 가장 짧은 작업을 먼저 처리하는 것으로 평균 반환 시간이 가장 짧다.
ㄷ. 먼저 들어온 작업을 먼저 처리해 주는 것으로, 특별한 경우 효율이 떨어질 수 있다.
ㄹ. 작업에 시간의 할당량이 설정되는 방법으로, 시분할 시스템에서 특별한 형태로 설계된 것이다.
ㅁ. SJF기법의 단점인 긴 작업과 짧은 작업의 불평등을 보완하는 기법이다.

① 2개
② 3개
③ 4개
④ 5개

정답 1번

출제 포인트

CPU 스케줄링 기법별로 특정 프로세스가 계속 선택되지 못하는 기아 현상의 발생 가능성을 구분하는 문제이다.

NoteBase

핵심 개념

기아는 실행 가능한 프로세스가 다른 프로세스에 계속 밀려 무기한 대기할 수 있는 현상이다.
ㄱ은 우선순위 스케줄링이다. 높은 우선순위 프로세스가 계속 도착하면 낮은 우선순위 프로세스에 기아가 발생할 수 있다.
ㄴ은 최단 작업 우선(SJF) 스케줄링이다. 짧은 작업이 계속 도착하면 긴 작업이 계속 미뤄져 기아가 발생할 수 있다.
ㄷ은 선입선처리(FCFS) 스케줄링이다. 도착 순서대로 처리하므로 대기 시간이 길어질 수는 있어도 먼저 도착한 작업이 무기한 추월당하지는 않는다.
ㄹ은 라운드 로빈(RR) 스케줄링이다. 준비 큐의 각 프로세스에 시간 할당량을 차례로 부여하므로 기아가 발생하지 않는다.
ㅁ은 최고 응답률 우선(HRN) 스케줄링이다. 응답률을 (대기 시간+서비스 시간)/서비스 시간으로 정하고, 대기할수록 응답률이 높아지게 하여 SJF의 기아 문제를 보완한다.
따라서 기아가 발생할 수 있는 것은 ㄱ과 ㄴ의 2개이다.

선지별 해설

① O — 기아가 발생할 수 있는 기법은 우선순위 스케줄링과 SJF의 2개이므로 옳다.
② X — ‘3개’가 틀린 표현이다. FCFS, RR, HRN은 해당하지 않으므로 올바른 개수는 2개이다.
③ X — ‘4개’가 틀린 표현이다. 기아 가능성이 있는 것은 ㄱ과 ㄴ뿐이므로 올바른 개수는 2개이다.
④ X — ‘5개’가 틀린 표현이다. 다섯 설명이 모두 기아를 일으키는 것이 아니며 올바른 개수는 2개이다.

오답 포인트

FCFS에서 긴 작업 때문에 뒤의 짧은 작업들이 오래 기다리는 호위 효과와 기아를 혼동하기 쉽다. 오래 기다리는 것만으로는 기아가 아니며, 다른 작업에 계속 추월당해 무기한 연기될 가능성이 있어야 한다. 또한 HRN은 SJF 계열이지만 대기 시간을 응답률에 반영하므로 SJF의 기아를 보완한다.

다시 나오면 이렇게 판단

새로 도착한 작업이 기존 작업을 계속 추월할 수 있는지를 확인한다. 우선순위와 SJF는 기아 가능, FCFS와 RR은 기아 없음, HRN은 대기 시간 반영으로 기아 보완으로 구분한다.

암기 포인트

기아 가능은 우선순위·SJF, 기아 보완은 에이징·HRN이다.

NoteBase

문 2. 인터럽트(Interrupt)에 대한 설명으로 옳지 않은 것은?

① 단일회선 인터럽트 요청신호 체계의 경우는 고유 회선 인터럽트 요청신호 체계와 달리 장치 식별 루틴이 필요하다.

② 인터럽트를 처리하기 위해 수행하는 프로그램 루틴을 인터럽트 서비스 루틴(ISR)이라고 하며, 오버플로에 의한 인터럽트가 발생하면 오버플로 발생 경고 메시지와 주 프로그램을 종료하는 인터럽트 서비스 루틴이 동작한다.

③ 폴링은 소프트웨어에 의한 인터럽트를 요청한 장치 식별 방식이다.

④ 벡터 인터럽트 방식은 장치를 식별하기 위한 별도의 프로그램 루틴이 있다.

정답 4번

출제 포인트

인터럽트 요청 장치를 식별하는 방식과 인터럽트 서비스 루틴의 역할을 구분하는 문제이다.

핵심 개념

인터럽트 서비스 루틴(ISR)은 인터럽트가 발생했을 때 해당 원인을 처리하도록 실행되는 프로그램이다.
단일회선 방식은 여러 장치가 하나의 인터럽트 요청 회선을 공유하므로 어느 장치가 요청했는지 별도로 식별해야 한다. 폴링은 프로그램이 각 장치의 상태를 차례로 조사하는 소프트웨어 식별 방식이다. 벡터 인터럽트는 장치 또는 인터럽트 제어기가 제공한 인터럽트 벡터로 처리할 ISR의 위치를 직접 결정하는 방식이다.

선지별 해설

① O. 단일회선 방식에서는 여러 장치가 같은 요청선을 사용하므로 요청 장치를 찾는 식별 루틴이 필요하다. 장치마다 고유 요청 회선을 쓰는 방식은 활성화된 회선 자체로 장치를 구별할 수 있다.
② O. 인터럽트 발생 시 원인에 맞는 처리를 수행하는 프로그램이 ISR이다. 오버플로 인터럽트의 처리 루틴은 오버플로 사실을 알리고, 정상 실행을 계속할 수 없는 경우 주 프로그램을 종료하도록 구성될 수 있다.
③ O. 폴링은 CPU가 각 장치의 상태 레지스터를 프로그램으로 차례로 검사하여 인터럽트를 요청한 장치를 찾는 소프트웨어 식별 방식이다.
④ X. 틀린 표현은 '장치를 식별하기 위한 별도의 프로그램 루틴이 있다'이다. 벡터 인터럽트 방식에서는 장치 또는 인터럽트 제어기가 제공하는 인터럽트 벡터로 해당 ISR의 시작 위치를 직접 찾으므로 장치 식별을 위한 별도의 소프트웨어 탐색 루틴이 필요하지 않다.

오답 포인트

폴링도 벡터 인터럽트도 요청 장치를 구별한다는 결과만 보면 ③과 ④를 혼동하기 쉽다. 차이는 식별 수단이다. 폴링은 프로그램이 장치를 하나씩 검사하지만, 벡터 방식은 전달된 벡터가 처리 루틴을 직접 가리킨다.

NoteBase

다시 나오면 이렇게 판단

공유 회선이면 별도 장치 식별이 필요한지 확인하고, 식별 방식이 소프트웨어의 순차 검사이면 폴링, 인터럽트 벡터에 의한 직접 분기이면 벡터 인터럽트로 판단한다.

암기 포인트

폴링은 하나씩 묻고, 벡터는 주소로 바로 간다.

NoteBase

문 3. 2진수 1001에 대한 해밍코드로 옳은 것은?(단, 짝수 패리티 체크를 사용한다)

① 0110010

② 1000011

③ 0100101

④ 0011001

정답 4번

출제 포인트

이 문제는 7비트 해밍코드에서 패리티 비트의 위치를 정하고 짝수 패리티를 계산하는 문제이다.

조건 정리

주어진 데이터는 1001이다.
7비트 해밍코드에서는 위치 1, 2, 4에 패리티 비트 p_1, p_2, p_4를 놓고, 위치 3, 5, 6, 7에 데이터 비트 1, 0, 0, 1을 차례로 놓는다.
각 패리티 검사 집합에서 1의 개수가 짝수가 되도록 패리티 비트를 정한다.

사용할 공식

짝수 패리티 조건은 다음과 같다.
$p_1 \oplus b_3 \oplus b_5 \oplus b_7 = 0$
$p_2 \oplus b_3 \oplus b_6 \oplus b_7 = 0$
$p_4 \oplus b_5 \oplus b_6 \oplus b_7 = 0$
각 식은 해당 위치를 이진수로 나타냈을 때 패리티 위치의 비트가 1인 자리들을 검사하기 때문에 사용한다.

계산 과정

데이터를 배치하면 위치 1부터 7까지 $p_1, p_2, 1, p_4, 0, 0, 1$이 된다.
p_1의 검사 대상에는 데이터의 1이 두 개 있으므로 $p_1 = 0$이다.
p_2의 검사 대상에도 데이터의 1이 두 개 있으므로 $p_2 = 0$이다.
p_4의 검사 대상에는 데이터의 1이 한 개 있으므로 짝수로 만들기 위해 $p_4 = 1$이다.
따라서 완성된 해밍코드는 0011001이다.

NoteBase

선지 판단

① X — 0110010에서 데이터 위치 3, 5, 6, 7을 읽으면 1010이므로 주어진 데이터 1001을 보존하지 않는다. 올바른 코드는 0011001이다.
② X — 1000011에서 데이터 위치 3, 5, 6, 7을 읽으면 0011이므로 주어진 데이터와 다르다. 올바른 코드는 0011001이다.
③ X — 0100101에서 데이터 위치 3, 5, 6, 7을 읽으면 0101이므로 주어진 데이터와 다르다. 올바른 코드는 0011001이다.
④ O — 데이터 위치에는 1001이 들어 있고, 위치 1·3·5·7, 2·3·6·7, 4·5·6·7의 1의 개수가 각각 2개, 2개, 2개로 모두 짝수이다.

오답 포인트

데이터 비트와 패리티 비트를 연속해서 적거나 패리티 위치를 0부터 세면 다른 선지가 나온다. 이 문항에서는 위치를 1부터 세어 2의 거듭제곱 위치인 1, 2, 4를 패리티 비트 자리로 잡는 것이 핵심이다.

다시 나오면 이렇게 판단

먼저 위치 1, 2, 4를 비워 두고 데이터 비트를 3, 5, 6, 7에 배치한 다음, 각 검사 집합의 1의 개수가 짝수가 되도록 패리티 비트를 채운다. 마지막에는 세 검사 집합의 XOR가 모두 0인지 검산한다.

암기 포인트

해밍코드는 1·2·4가 패리티 자리이고, 짝수 패리티의 정상 검사 결과는 모두 0이다.

NoteBase

문 4. 구글이 클라우드 시대를 겨냥해서 만든 차세대 태블릿 PC용 OS는?

① tinyOS

② 크롬 OS

③ 비스타

④ 안드로이드

정답 2번

출제 포인트

클라우드 중심의 구글 운영체제인 크롬 OS를 다른 운영체제와 구별하는 문제이다.

핵심 개념

크롬 OS는 구글이 웹과 클라우드 서비스의 활용을 중심으로 설계한 운영체제이다. 안드로이드는 스마트폰과 태블릿 등 모바일 기기를 주된 대상으로 하는 구글의 모바일 운영체제이고, TinyOS는 무선 센서 네트워크용 경량 운영체제이며, 윈도우 비스타는 마이크로소프트의 개인용 컴퓨터 운영체제이다.

선지별 해설

① X — TinyOS는 무선 센서 네트워크의 저전력 센서 노드용 경량 운영체제이다. 틀린 표현은 구글이 클라우드 시대를 겨냥해 만든 태블릿 PC용 OS라는 취지이며, 올바르게는 센서 네트워크용 운영체제이다.
② O — 크롬 OS는 구글이 웹 애플리케이션과 클라우드 서비스 이용을 중심으로 개발한 운영체제이므로 문항의 설명에 해당한다.
③ X — 비스타는 구글이 아니라 마이크로소프트가 개발한 윈도우 계열 개인용 컴퓨터 운영체제이다. 틀린 표현은 구글이 만든 OS라는 부분이며, 올바르게는 마이크로소프트의 윈도우 비스타이다.
④ X — 안드로이드는 구글의 운영체제이지만 스마트폰과 태블릿 등 모바일 기기를 중심으로 설계되었다. 틀린 표현은 이 문항에서 말하는 클라우드 중심의 차세대 PC용 OS라는 부분이며, 올바르게는 크롬 OS이다.

오답 포인트

안드로이드도 구글이 만들었고 태블릿에서 사용되므로 ④를 고르기 쉽다. 그러나 문항의 결정적 단서는 '클라우드 시대를 겨냥한' 운영체제이며, 이는 웹과 클라우드 중심의 크롬 OS를 가리킨다.

다시 나오면 이렇게 판단

구글 운영체제를 구별할 때 웹·클라우드·크롬북 중심이면 크롬 OS, 스마트폰·모바일 앱 중심이면 안드로이드로 판단한다.

NoteBase

암기 포인트

크롬 OS는 웹·클라우드, 안드로이드는 모바일, TinyOS는 센서 네트워크이다.

NoteBase

문 5. RAID(Redundant Array of Inexpensive Disks)에 관한 설명으로 옳지 않은 것은?

① RAID 0은 디스크 스트라이핑 방식으로 빠른 입·출력이 가능하도록 여러 개의 하드디스크에 데이터를 분산 저장한다.

② RAID 3은 RAID 0처럼 데이터를 byte단위로 분산 저장하는데 에러검출·수정을 위해 패리티 드라이브를 사용한다.

③ RAID 4는 데이터를 bit 단위로 여러 디스크에 분할하여 저장하며 별도의 패리티 디스크를 사용한다.

④ RAID 5는 별도의 패리티 디스크를 사용하지 않고 데이터를 저장하는 디스크에 패리티를 라운드 로빈 방식으로 분산하여 저장한다.

정답 3번

출제 포인트

RAID 0·3·4·5를 데이터 분할 단위와 패리티 저장 방식으로 구별하는 문제이다.

핵심 개념

RAID는 여러 디스크를 하나의 저장 장치처럼 구성하여 성능이나 장애 복구 능력을 높이는 방식이다.
RAID 0은 블록 단위 스트라이핑만 사용하고 패리티가 없다. RAID 3은 바이트 단위 스트라이핑과 전용 패리티 디스크를 사용한다. RAID 4는 블록 단위 스트라이핑과 전용 패리티 디스크를 사용한다. RAID 5는 블록 단위 스트라이핑을 사용하며 패리티를 모든 디스크에 분산한다.

선지별 해설

① O — RAID 0은 데이터를 여러 디스크에 분산하는 스트라이핑 방식이므로 병렬 입·출력 성능을 높일 수 있다. 다만 패리티가 없어 장애 복구 기능은 제공하지 않는다.
② O — RAID 3은 데이터를 바이트 단위로 스트라이핑하고 별도의 패리티 드라이브에 패리티를 저장하여 한 디스크의 고장에 대응한다.
③ X — 틀린 표현은 ‘bit 단위’이다. RAID 4는 데이터를 블록 단위로 여러 디스크에 분산하고 별도의 전용 패리티 디스크를 사용한다.
④ O — RAID 5는 전용 패리티 디스크를 두지 않고 데이터와 패리티 블록을 여러 디스크에 순환 분산하므로 특정 패리티 디스크에만 부하가 집중되는 문제를 줄인다.

오답 포인트

③의 ‘별도의 패리티 디스크’는 RAID 4의 올바른 특징이므로 맞는 말처럼 보이지만, 데이터 분할 단위를 ‘bit 단위’라고 바꾼 부분이 함정이다. RAID 3은 바이트 단위, RAID 4는 블록 단위이다.

NoteBase

다시 나오면 이렇게 판단

각 RAID를 '스트라이핑 단위'와 '패리티 위치'의 두 축으로 구별한다. RAID 3은 바이트·전용 패리티, RAID 4는 블록·전용 패리티, RAID 5는 블록·분산 패리티이다.

암기 포인트

RAID 3은 바이트, RAID 4는 블록, RAID 5는 분산 패리티이다.

NoteBase

문 6. 싱글사인온(SSO, Single Sign On)에 대한 설명으로 옳지 않은 것은?

① SSO는 관리의 투명성을 높이고, 신뢰성 및 효율성을 높일 수 있다.

② SSO의 가장 큰 약점은 일단 최초 인증과정을 통과하면 모든 서버나 사이트에 접속할 수 있다는 것이다.

③ SSO는 레거시(Legacy)시스템이나 클라이언트/서버 환경에서 적용할 수 있다.

④ SSO는 각 자원별로 한 번의 사인을 필요로 하는 개별 인증 방식이다.

정답 4번

출제 포인트

싱글사인온(SSO)의 핵심인 통합 인증의 의미와 장점, 보안상 약점, 적용 범위를 구별하는 문제이다.

핵심 개념

SSO는 사용자가 한 번 인증하면 별도의 반복 로그인 없이 서로 연계된 여러 시스템의 자원을 부여된 권한 범위에서 이용하게 하는 통합 인증 방식이다. 인증을 중앙에서 일관되게 관리하여 편의성과 관리 효율을 높이지만, 인증 정보가 탈취되거나 통합 인증 지점에 문제가 생기면 피해가 여러 연계 시스템으로 확대될 수 있다.

선지별 해설

① O — 인증과 접근 관리를 통합하면 계정과 접근 이력을 일관되게 관리할 수 있으므로 관리의 투명성, 신뢰성 및 효율성을 높일 수 있다.
② O — 최초 인증을 통과하면 SSO에 연계되고 접근 권한이 부여된 여러 서버나 사이트를 다시 인증하지 않고 이용할 수 있다. 따라서 인증 정보가 노출되면 여러 자원에 대한 접근이 한꺼번에 허용될 수 있다는 점이 큰 보안상 약점이다. 이 선지의 '모든 서버나 사이트'는 SSO 연계 및 권한 부여 범위의 자원을 뜻한다.
③ O — SSO는 웹 환경에만 한정되지 않으며, 연동 수단을 두어 레거시 시스템이나 클라이언트/서버 환경에도 적용할 수 있다.
④ X — 틀린 부분은 '각 자원별로 한 번의 사인을 필요로 하는 개별 인증 방식'이다. 각 자원마다 따로 로그인하는 것은 개별 인증이며, SSO는 한 번의 인증으로 연계된 여러 자원에 접근하는 통합 인증 방식이다.

오답 포인트

②의 '모든 서버나 사이트'만 보고 틀렸다고 판단하기 쉽다. 그러나 이 표현은 아무 시스템에나 무제한으로 접속한다는 뜻이 아니라 SSO에 연계되고 사용자에게 권한이 부여된 범위를 전제로 하며, 하나의 인증 정보가 침해되면 피해 범위가 넓어진다는 약점을 설명한다. 결정적인 오류는 ④가 SSO를 자원별 개별 인증 방식이라고 뒤바꾼 데 있다.

NoteBase

다시 나오면 이렇게 판단

‘한 번 인증하여 여러 연계 자원 이용’이면 SSO이고, ‘자원마다 별도 로그인’이면 개별 인증이다. 장점은 인증의 일원화와 반복 로그인 감소이고, 약점은 인증 실패나 인증 정보 침해의 영향이 여러 연계 자원으로 확대될 수 있다는 점이다.

암기 포인트

SSO는 한 번의 인증, 여러 연계 자원이다. 자원마다 한 번씩 인증하는 방식이 아니다.

NoteBase

문 7. 다음 부울 함수식 F를 간략화한 결과로 옳은 것은?

$F = ABC + ABC' + AB'C + AB'C' + A'B'C + A'B'C'$

① $F = A' + B$
② $F = A + B'$
③ $F = AB'$
④ $F = A'B$

정답 2번

출제 포인트

이 문제는 부울 대수의 결합법칙, 보수법칙, 흡수법칙을 이용하여 논리식을 간략화하는 문제이다.

조건 정리

주어진 식은 $F = ABC + ABC' + AB'C + AB'C' + A'B'C + A'B'C'$이다. C와 C'가 한 쌍으로 나타나는 항들을 묶어 $C + C' = 1$을 적용한다.

사용할 공식

보수법칙: $X + X' = 1$
항등법칙: $X \cdot 1 = X$
흡수 형태: $X + X'Y = X + Y$
각 항에서 공통 인수를 묶을 수 있고, 마지막에 $A + A'B' = A + B'$를 적용할 수 있으므로 이 공식들을 사용한다.

계산 과정

1. 앞의 네 항을 묶으면 $ABC + ABC' + AB'C + AB'C' = AB(C + C') + AB'(C + C') = AB + AB' = A$이다.
2. 뒤의 두 항을 묶으면 $A'B'C + A'B'C' = A'B'(C + C') = A'B'$이다.
3. 따라서 $F = A + A'B' = A + B'$이다.

선지 판단

① X — $F = A' + B$가 아니라 $F = A + B'$이다. 틀린 표현 $A' + B$는 A와 B의 보수를 서로 바꾸어 적용한 것이며, 올바른 식은 $A + B'$이다.
② O — 계산 결과가 정확히 $F = A + B'$이므로 옳다.
③ X — $F = AB'$는 $A = 1, B = 1$일 때 실제 함수값이 1인 경우를 제외하므로 지나치게 제한된 식이다. 틀린 표현 AB'는 올바르게는 $A + B'$이다.
④ X — $F = A'B$는 변수와 보수 및 연산이 모두 잘못된 식이다. 틀린 표현 $A'B$는 올바르게는 $A + B'$이다.

NoteBase

오답 포인트

①처럼 보수 기호를 다른 변수에 붙이거나, ③·④처럼 최종 합 $A + B'$를 곱으로 바꾸는 것이 함정이다. 특히 $A + A'B' = A + B'$에서 A'를 그대로 남기지 않는 점이 핵심이다.

다시 나오면 이렇게 판단

서로 C와 C'만 다른 항을 먼저 묶어 $C + C' = 1$로 제거한 뒤, 남은 식에 $X + X'Y = X + Y$를 적용한다.

암기 포인트

보수쌍은 1로 소거되고, $X + X'Y = X + Y$이다.

NoteBase

문 8. 다음 중 빅데이터의 내용 중 옳지 않은 것은?

① 빅데이터의 처리 과정으로는 수집-저장-분석-처리-시각화(표현)-이용-폐기 순이다.

② 빅데이터의 특성으로 Volume(규모), Velocity(속도), Variety(다양성)이다.

③ 자연어 처리는 빅데이터 분석기술 중의 하나이다.

④ 빅데이터는 그림, 영상 등 비정형 데이터를 포함한다.

정답 1번

출제 포인트

빅데이터의 처리 과정, 대표 특성인 3V, 분석 기술, 데이터 형태를 구분하는 문제이다.

핵심 개념

빅데이터는 기존 방식으로 처리하기 어려운 대규모·고속·다양한 데이터를 수집하고 처리·분석하여 가치를 얻는 데이터 및 관련 기술을 말한다.
처리 과정에서는 분석에 적합한 형태로 데이터를 정제·변환하는 처리가 분석보다 먼저 이루어진다.
대표적인 3V는 Volume(규모), Velocity(속도), Variety(다양성)이며, 정형 데이터뿐 아니라 그림·영상·텍스트 같은 비정형 데이터도 포함한다.

선지별 해설

① X
틀린 부분은 '분석-처리'의 순서이다. 데이터 처리는 원자료를 정제·변환하여 분석 가능한 형태로 만드는 단계이므로 분석보다 앞선다. 올바른 순서는 수집-저장-처리-분석-시각화(표현)-이용-폐기이다.

② O
Volume(규모), Velocity(속도), Variety(다양성)는 빅데이터의 대표적인 특성인 3V이다.

③ O
자연어 처리는 사람의 언어인 텍스트나 음성에서 의미와 정보를 추출하는 기술로, 비정형 데이터를 다루는 빅데이터 분석 기술에 해당한다.

④ O
빅데이터는 표처럼 구조가 정해진 정형 데이터뿐 아니라 그림과 영상 같은 비정형 데이터도 포함한다.

NoteBase

오답 포인트

①은 모든 처리 단계의 이름을 자연스럽게 나열하지만 '분석'과 '처리'의 순서를 바꾸었다. 분석하려면 먼저 원자료를 정제·변환하는 처리가 필요하다는 선후 관계가 핵심 함정이다.

다시 나오면 이렇게 판단

빅데이터 처리 순서는 원자료를 모으고 저장한 뒤, 분석 가능한 형태로 처리하고, 그 결과를 분석·시각화하여 이용하는 흐름으로 판단한다. 특성 문제에서는 3V인 규모·속도·다양성을 확인한다.

암기 포인트

수집-저장-처리-분석-시각화-이용-폐기, 3V는 규모·속도·다양성이다.

NoteBase

문 9. 자바 언어에 대한 설명 중 옳지 않은 것은?

① 인터페이스는 다중상속을 지원하지만 추상클래스는 단일상속만을 지원한다.

② protected로 선언된 필드는 하위클래스에게만 접근을 허용한다.

③ 하위클래스 객체를 상위클래스 타입의 변수가 참조하는 것은 허용하지만, 상위클래스 객체를 하위클래스 타입의 변수가 참조하는 것은 허용하지 않는다.

④ private로 선언된 개체변수는 그 개체변수를 포함하는 클래스의 자식클래스라 하더라도 직접 접근할 수 없다.

정답 2번

출제 포인트

Java의 상속 구조, 참조 타입 변환, protected와 private의 접근 범위를 구별하는 문제이다.

핵심 개념

인터페이스: 클래스가 구현해야 할 동작의 규약이며 여러 인터페이스를 동시에 구현할 수 있고, 인터페이스도 여러 인터페이스를 상속할 수 있다.
클래스 상속: 일반 클래스와 추상클래스 모두 extends로 직접 상속할 수 있는 상위클래스는 하나뿐이다.
protected: 같은 패키지의 클래스와 다른 패키지의 하위클래스에서 접근할 수 있다.
private: 선언한 클래스 내부에서만 직접 접근할 수 있다.

NoteBase

선지별 해설

① O
인터페이스는 여러 인터페이스를 상속할 수 있고 클래스도 여러 인터페이스를 구현할 수 있다. 반면 추상클래스도 클래스이므로 하나의 클래스만 직접 상속하는 단일상속 규칙을 따른다.

② X
틀린 부분은 '하위클래스에게만 접근을 허용한다'이다. protected 멤버는 하위클래스뿐 아니라 같은 패키지에 속한 다른 클래스에서도 접근할 수 있다.

③ O
하위클래스 객체를 상위클래스 타입 변수로 참조하는 업캐스팅은 허용된다. 실제 객체가 상위클래스 객체라면 이를 하위클래스 타입 변수에 그대로 대입할 수 없으며, 명시적으로 형변환해도 실행 시 하위클래스 객체가 아니므로 정상적으로 참조할 수 없다.

④ O
private 개체변수는 선언한 클래스 내부에서만 직접 접근할 수 있으므로 자식클래스에서도 직접 접근할 수 없다. 필요한 경우 선언 클래스가 제공하는 접근 메서드를 통해 간접적으로 다룬다.

오답 포인트

②의 'protected는 하위클래스 전용'이라는 표현이 함정이다. protected는 상속 관계만이 아니라 같은 패키지에도 접근을 허용하므로 private보다 범위가 넓다.

다시 나오면 이렇게 판단

접근 제한자는 먼저 같은 클래스, 같은 패키지, 하위클래스, 그 밖의 클래스 순으로 접근 범위를 확인한다. protected가 나오면 '같은 패키지 또는 하위클래스', private가 나오면 '선언 클래스 내부에서만 직접 접근'으로 판단한다.

암기 포인트

protected는 같은 패키지와 하위클래스, private는 선언 클래스 내부이다. 클래스는 단일상속, 인터페이스는 다중 구현·상속이 가능하다.

NoteBase

문 10. 다음 프로그램의 실행 결과는?

```
#include <stdio.h>

int main()
{
    int arr[]={8, 5, 3, 1, 2, 7, 9};
    int *p=arr+2, a=0, b=0;

    a=*++p;
    b=(*p)++;
    printif("%d, %d\n", a, b);
    return 0;
}
```

① 3, 3
② 1, 1
③ 3, 1
④ 1, 2

정답 2번

출제 포인트

포인터에 대한 전위 증가와 포인터가 가리키는 값에 대한 후위 증가를 구분하여 출력값을 추적하는 문제이다.

조건 정리

배열 arr의 원소는 차례로 8, 5, 3, 1, 2, 7, 9이다.
p는 처음에 arr+2이므로 값 3이 저장된 arr[2]를 가리킨다.
구해야 할 값은 a=*++p와 b=(*p)++를 차례로 실행한 뒤의 a와 b이다.

사용할 원리

전위 증가 ++p는 포인터 p를 먼저 다음 원소로 이동시킨 뒤 역참조한다. 후위 증가 (*p)++는 p가 가리키는 현재 값을 식의 결과로 사용한 뒤 그 배열 원소의 값을 1 증가시킨다. 연산자의 결합에 따라 *++p는 *(++p), (*p)++는 역참조한 값 자체의 후위 증가로 해석된다.

NoteBase

계산 과정

1. 처음 p는 arr[2]를 가리키며 *p는 3이다.
2. a=*++p에서 p를 먼저 증가시키면 p는 arr[3]을 가리킨다. arr[3]의 값은 1이므로 a=1이다.
3. b=(*p)++에서 현재 *p의 값 1을 b에 먼저 대입하므로 b=1이다. 그 후 arr[3]의 값만 2로 증가하며 b는 바뀌지 않는다.
4. 따라서 printf가 출력하는 값은 1, 1이다.

선지 판단

① X — 3, 3은 틀린 출력이다. ++p가 실행되면 p는 값 3이 있는 arr[2]에서 값 1이 있는 arr[3]으로 이동한다. 올바른 출력은 1, 1이다.
② O — a=*++p의 결과가 1이고, b=(*p)++는 증가 전 값 1을 b에 대입하므로 1, 1이 출력된다.
③ X — 3, 1에서 첫 번째 값 3이 틀렸다. *++p는 현재 원소 3을 사용한 뒤 이동하는 식이 아니라 p를 먼저 이동시켜 다음 원소 1을 읽는 식이므로 올바른 출력은 1, 1이다.
④ X — 1, 2에서 두 번째 값 2가 틀렸다. (*p)++는 증가 전 값 1을 b에 대입한 다음 arr[3]만 2로 바꾸므로 올바른 출력은 1, 1이다.

오답 포인트

*++p를 (*p)++처럼 해석하면 a를 3으로 잘못 판단할 수 있다. 또한 (*p)++에서 증가된 값 2가 b에 저장된다고 생각하기 쉽지만, 후위 증가는 증가 전 값 1을 식의 결과로 사용한다.

다시 나오면 이렇게 판단

먼저 괄호를 보충하여 연산 대상을 확정한 뒤, 포인터의 위치와 배열 원소의 값을 별도로 추적한다. 전위 증가는 증가 후 값을 사용하고 후위 증가는 증가 전 값을 사용한다.

암기 포인트

*++p는 포인터를 먼저 이동한 뒤 읽고, (*p)++는 현재 원소를 먼저 사용한 뒤 그 원소를 증가시킨다.

NoteBase

문 11. 다음에 설명하는 컬러 모델은?

RGB 컬러의 양상을 나타내는 요소들이다. 이러한 용어들은 CRT 내의 각 픽셀들의 색상과 관련하여 가장 자주 사용된다. 색상들이 R, G, B 요소들의 형태로 표현될 수 있는 것과 마찬가지로 색조, 채도 및 명도에 따라 가능한 모든 색상들을 표현할 수 있다.

① HSB

② CMYK

③ YUV

④ YIQ

정답 1번

출제 포인트

색상을 색조(Hue), 채도(Saturation), 명도(Brightness)로 표현하는 컬러 모델을 구별하는 문제이다.

핵심 개념

HSB는 색상을 색조·채도·명도의 세 요소로 나타내는 컬러 모델이다. 색조는 색의 종류, 채도는 색의 선명한 정도, 명도는 밝고 어두운 정도를 뜻한다.
HSB는 H(Hue)·S(Saturation)·B(Brightness), CMYK는 인쇄용 감산 혼합, YUV와 YIQ는 밝기 성분과 색차 성분을 분리하는 방식으로 구별한다.

선지별 해설

① O — HSB는 색조(Hue), 채도(Saturation), 명도(Brightness)로 색을 표현하므로 지문의 설명과 일치한다.
② X — CMYK는 청록(Cyan), 자홍(Magenta), 노랑(Yellow), 검정(Key/Black)을 사용하는 감산 혼합 컬러 모델이다. 틀린 부분은 색조·채도·명도로 표현한다는 연결이며, 올바르게는 HSB가 이에 해당한다.
③ X — YUV는 Y의 밝기 성분과 U·V의 색차 성분으로 색 정보를 표현하는 컬러 모델이다. 틀린 부분은 색조·채도·명도로 표현한다는 연결이며, 올바르게는 HSB가 이에 해당한다.
④ X — YIQ는 Y의 밝기 성분과 I·Q의 색상 성분으로 색 정보를 표현하는 컬러 모델이다. 틀린 부분은 색조·채도·명도로 표현한다는 연결이며, 올바르게는 HSB가 이에 해당한다.

오답 포인트

YUV와 YIQ에도 밝기를 나타내는 Y 성분이 있어 명도라는 말과 혼동하기 쉽다. 그러나 지문은 밝기만이 아니라 색조·채도·명도의 세 요소를 함께 제시하므로 HSB를 가리킨다.

NoteBase

다시 나오면 이렇게 판단

컬러 모델의 영문 머리글자를 설명의 핵심 용어와 대응시킨다. Hue·Saturation·Brightness가 함께 나오면 HSB, Cyan·Magenta·Yellow·Black이 나오면 CMYK, 밝기와 색차 성분의 분리가 나오면 YUV 또는 YIQ로 판단한다.

암기 포인트

HSB는 색조·채도·명도, CMYK는 인쇄, YUV·YIQ는 밝기와 색상 정보의 분리이다.

NoteBase

문 12. 다음 중 용어 설명이 옳지 않은 것은?

① PPP서비스는 사이트에 접속하면 로그인 창에 아이디가 표시 되도록 하거나 쇼핑몰에서 검색해본 상품목록이 표시 되도록 하는 기능을 말한다.

② 라우터(Router)는 네트워크 계층에서 망을 연결하고 라우팅 알고리즘을 이용하여 최적의 경로를 선택하여 패킷을 전송한다.

③ 게이트웨이(Gateway)는 두 개의 서로 다른 형태의 네트워크를 상호 연결시켜 주는 관문 역할을 하는 장치이다.

④ TELNET은 인터넷 상에 있는 원격지의 컴퓨터에 접속하여 자신의 컴퓨터처럼 사용할 수 있도록 해주는 인터넷 서비스이다.

정답 1번

출제 포인트

인터넷·네트워크 관련 용어인 쿠키, PPP, 라우터, 게이트웨이, TELNET의 기능을 구별하는 문제이다.

핵심 개념

쿠키(Cookie): 웹 사이트가 이용자의 컴퓨터에 저장하는 작은 정보로, 로그인 상태나 방문·검색 기록 등을 기억하는 데 쓰인다.
PPP(Point-to-Point Protocol): 두 지점 사이의 직접 연결에서 데이터를 전송하기 위한 데이터 링크 계층 프로토콜이다.
라우터(Router): 서로 다른 네트워크 사이에서 패킷의 전달 경로를 선택하는 장치이다.
게이트웨이(Gateway): 구조나 프로토콜이 서로 다른 네트워크를 연결하고 필요하면 변환을 수행하는 관문이다.
TELNET: 네트워크를 통해 원격 컴퓨터에 접속하여 명령을 실행하는 원격 접속 서비스이다.

선지별 해설

① X — 틀린 표현은 ‘PPP서비스’이다. 사이트가 로그인 정보나 검색 상품 목록 등을 기억하여 다시 표시하는 기능은 쿠키(Cookie)를 이용한 것이다. PPP는 두 지점 간 통신을 위한 데이터 링크 계층 프로토콜이다.
② O — 라우터는 네트워크 계층에서 네트워크들을 연결하고, 라우팅 정보를 바탕으로 패킷을 목적지까지 전달할 경로를 선택한다.
③ O — 게이트웨이는 서로 다른 구조나 통신 방식을 사용하는 네트워크를 연결하는 관문 역할을 하며, 필요한 경우 프로토콜 변환도 수행한다.
④ O — TELNET은 인터넷상의 원격 컴퓨터에 접속하여 그 컴퓨터의 명령을 실행할 수 있게 하는 원격 접속 서비스이다.

NoteBase

오답 포인트

①은 웹 사이트가 이용자 정보를 기억한다는 설명 자체는 자연스럽지만, 그 기능의 명칭을 'PPP'로 바꾸어 놓았다. 이 기능은 쿠키이며, PPP는 웹 이용 기록을 저장하는 서비스가 아니라 지점 간 데이터 전송 프로토콜이다.

다시 나오면 이렇게 판단

로그인 상태·방문 기록·검색 기록을 기억하면 쿠키, 네트워크 사이의 경로 선택이면 라우터, 서로 다른 네트워크의 연결·변환이면 게이트웨이, 원격 명령 접속이면 TELNET으로 판단한다. PPP는 두 지점 간 연결 프로토콜로 구분한다.

암기 포인트

쿠키는 웹 정보 기억, PPP는 점대점 연결, 라우터는 경로 선택, 게이트웨이는 이종망 연결, TELNET은 원격 접속이다.

NoteBase

문 13. CPU 내부에 있는 특수 목적용 레지스터 중 인터럽트 수행과정에서 원래의 프로세스가 수행될 수 있도록 프로그램 카운터의 주소를 임시로 저장하는 레지스터를 무엇이라 하는가?

① 명령 레지스터

② 상태 레지스터

③ 기억장치 버퍼 레지스터

④ 스택 포인터

정답 4번

출제 포인트

인터럽트가 발생했을 때 복귀 주소를 보존하는 스택과 스택 포인터의 역할을 다른 특수 목적 레지스터와 구별하는 문제이다.

핵심 개념

인터럽트 처리 전에는 중단된 프로그램으로 돌아갈 수 있도록 다음에 실행할 명령의 주소인 복귀 주소를 스택에 저장한다. 스택 포인터는 스택의 최상단 위치를 가리키므로, 복귀 주소를 저장하고 다시 꺼내는 과정에 사용된다. 명령 레지스터는 현재 해독·실행 중인 명령을, 상태 레지스터는 조건 코드와 CPU 상태를, 기억장치 버퍼 레지스터는 기억장치와 주고받는 데이터를 임시로 보관한다.

선지별 해설

① X — 명령 레지스터는 현재 해독하거나 실행하는 명령어를 보관한다. '프로그램 카운터의 복귀 주소를 임시로 저장한다'는 설명이 틀렸으며, 올바르게는 '현재 실행 중인 명령어를 보관한다'이다.
② X — 상태 레지스터는 연산 결과의 조건 코드와 인터럽트 허용 여부 등 CPU의 상태 정보를 보관한다. '프로그램 카운터의 복귀 주소를 임시로 저장한다'는 설명이 틀렸으며, 올바르게는 'CPU의 상태와 조건 코드를 보관한다'이다.
③ X — 기억장치 버퍼 레지스터는 CPU와 기억장치 사이에서 전송되는 명령어나 데이터를 임시로 보관한다. '인터럽트 복귀 주소를 보존한다'는 설명이 틀렸으며, 올바르게는 '기억장치에서 읽거나 기억장치에 쓸 데이터를 임시로 보관한다'이다.
④ O — 스택 포인터는 스택의 최상단 주소를 가리킨다. 인터럽트가 발생하면 중단 지점으로 복귀하는 데 필요한 프로그램 카운터 값이 스택에 저장되고, 스택 포인터가 그 저장 위치를 관리한다.

오답 포인트

인터럽트가 발생하면 프로그램 카운터 값뿐 아니라 상태 정보도 보존될 수 있으므로 상태 레지스터를 고르기 쉽다. 그러나 상태 레지스터는 CPU 상태 자체를 담고, 복귀 주소가 저장되는 스택의 위치를 관리하는 것은 스택 포인터이다.

NoteBase

다시 나오면 이렇게 판단

문제에서 '인터럽트 후 원래 프로그램으로 복귀', '복귀 주소 저장', '스택의 최상단'을 묻는다면 스택 포인터를 선택한다. '현재 명령'은 명령 레지스터, '조건 코드'는 상태 레지스터, '기억장치와 전송 중인 데이터'는 기억장치 버퍼 레지스터로 구별한다.

암기 포인트

복귀 주소는 스택에 저장되고, 스택의 꼭대기는 스택 포인터가 가리킨다.

NoteBase

문 14. 객체지향 언어의 설명으로 옳은 것은 모두 몇 개 인가?

ㄱ. 객체와 클래스는 재사용이 가능하다.
ㄴ. 다형성을 지원한다.
ㄷ. 추상 메소드는 선언만 하고 그 내용은 기술하지 않는 메소드이다.
ㄹ. 상속이란 새로운 클래스를 정의할 때 기존클래스의 속성을 물려받고 더 필요한 부분을 추가하는 방법이다.

① 1개
② 2개
③ 3개
④ 4개

정답 4번

출제 포인트

객체지향 언어의 핵심 특징인 재사용성, 다형성, 추상 메소드, 상속의 의미를 정확히 판단하는 문제이다.

핵심 개념

객체지향 프로그래밍은 데이터와 그 데이터를 처리하는 기능을 객체로 묶어 프로그램을 구성하는 방식이다.
재사용성: 이미 정의한 클래스나 생성한 객체를 다시 활용하여 중복을 줄이는 성질이다.
다형성: 같은 이름의 연산이나 메시지가 객체의 실제 유형에 따라 서로 다르게 동작할 수 있는 성질이다.
추상 메소드: 선언부만 두고 구체적인 구현 내용은 하위 클래스 등에서 작성하도록 한 메소드이다.
상속: 기존 클래스의 속성과 기능을 물려받아 새로운 클래스를 정의하고 필요한 부분을 추가하거나 재정의하는 방식이다.

제시문 판단

ㄱ O — 클래스는 한 번 정의한 속성과 메소드를 여러 객체 생성이나 다른 클래스 정의에 재사용할 수 있고, 객체도 다른 프로그램 구성 요소에서 활용할 수 있으므로 재사용이 가능하다.
ㄴ O — 객체지향 언어는 동일한 메시지나 메소드 호출이 객체의 실제 유형에 따라 다르게 수행되는 다형성을 지원한다.
ㄷ O — 추상 메소드는 메소드의 이름과 매개변수 등 선언만 제시하고 구체적인 실행 내용은 기술하지 않는 메소드이다.
ㄹ O — 상속은 기존 클래스의 속성과 기능을 물려받은 뒤 새로운 클래스에 필요한 내용을 추가하거나 재정의하는 방법이다.
따라서 옳은 설명은 ㄱ, ㄴ, ㄷ, ㄹ의 4개이다.

NoteBase

선지별 해설

① X — 옳은 설명이 1개라는 부분이 틀렸다. 올바르게는 ㄱ, ㄴ, ㄷ, ㄹ이 모두 옳으므로 4개이다.
② X — 옳은 설명이 2개라는 부분이 틀렸다. 올바르게는 네 설명이 모두 객체지향 언어의 특성에 부합하므로 4개이다.
③ X — 옳은 설명이 3개라는 부분이 틀렸다. 빠지는 설명 없이 ㄱ부터 ㄹ까지 모두 옳으므로 올바른 개수는 4개이다.
④ O — ㄱ, ㄴ, ㄷ, ㄹ이 모두 옳으므로 옳은 설명은 4개이다.

오답 포인트

ㄷ의 '내용은 기술하지 않는다'는 표현을 메소드가 불완전하다는 뜻으로 오해하기 쉽다. 이는 추상 메소드의 정확한 특징이며, 구체적인 구현은 이를 구현하는 하위 클래스에서 제공한다. 또한 상속은 기존 내용을 그대로 받는 데서 끝나지 않고 필요한 내용을 추가하거나 재정의할 수 있으므로 ㄹ도 옳다.

다시 나오면 이렇게 판단

개수형 문항은 각 문장을 독립적으로 판정한 뒤 O의 수를 센다. 객체지향의 대표 특징은 객체·클래스의 재사용, 다형성, 추상화, 상속이며, 추상 메소드는 선언만 있고 구현은 없다는 점을 기준으로 판단한다.

암기 포인트

객체지향의 핵심은 재사용·다형성·추상화·상속이며, 추상 메소드는 선언만 하고 구현은 하위 클래스에 맡긴다.

NoteBase

문 15. 다음 중 N-S(Nassi-Schneiderman) 차트의 특징으로 옳지 않은 것은?

① 구조적 프로그램을 표현하기 위한 방법으로 도형식 그래픽 설계이다.

② 제어 이동이 가능하며, 그림으로 표현하기 쉽다.

③ 원시코드의 변환이 용이하며 중첩, 회귀 구조에 사용한다.

④ 조건이 복합되어 있는 곳의 처리를 시각적으로 식별하는데 적합하다.

정답 2번

출제 포인트

N-S 차트가 구조적 프로그래밍의 제어 구조를 어떻게 표현하는지 묻는 문제이다.

핵심 개념

N-S 차트는 순차·선택·반복 구조를 중첩된 사각형 블록으로 나타내는 구조적 프로그램 설계 도구이다.
흐름선이나 GOTO와 같은 임의의 제어 이동보다 구조화된 제어 논리의 표현에 초점을 둔다.

선지별 해설

① O — N-S 차트는 구조적 프로그램의 논리를 도형 블록으로 표현하는 그래픽 설계 방법이다.
② X — 틀린 표현은 '제어 이동이 가능하다'이다. N-S 차트는 GOTO와 같은 임의의 제어 이동을 허용하는 방식이 아니라 순차·선택·반복의 구조화된 제어만을 블록으로 표현한다. 따라서 '임의의 제어 이동이 제한되며 구조화된 제어 논리를 그림으로 표현한다'라고 고쳐야 한다.
③ O — 블록 구조가 프로그램의 순차·선택·반복 구조와 대응하므로 원시 코드로 옮기기 쉽고, 블록의 중첩이나 재귀적 구조를 나타내는 데 사용할 수 있다.
④ O — 복합 조건을 선택 블록 안에 구조적으로 배치하므로 조건에 따른 처리 관계를 시각적으로 식별하는 데 적합하다.

오답 포인트

②의 '그림으로 표현'이라는 말만 보고 맞는 선지로 판단하기 쉽다. 핵심 오류는 그림 사용 여부가 아니라 '제어 이동이 가능하다'는 부분이며, N-S 차트는 임의의 분기를 제한하는 구조적 설계 도구이다.

다시 나오면 이렇게 판단

N-S 차트의 특징은 순차·선택·반복, 블록의 중첩, 원시 코드로의 쉬운 변환이다. GOTO나 임의의 제어 이동을 자유롭게 허용한다는 표현이 나오면 틀린 설명으로 판단한다.

암기 포인트

N-S 차트는 구조는 보이고, 임의 이동은 막는다.

NoteBase

문 16. 다음 중 해싱(Hashing)에 대한 설명으로 옳지 않은 것을 모두 고르시오?

ㄱ. 해싱을 이용한 경우 언더플로 처리, 데이터 변환 속도, 슬롯 크기 등을 고려한다.
ㄴ. 자료를 파일에서 검색할 때 키의 비교 방식 대신 키의 값을 계산하여 검색하는 방식이다.
ㄷ. 해싱은 충돌(Collision)이 발생하면 항상 오버플로가 발생한다.
ㄹ. 주소 값이 같은 경우 오버플로의 처리가 쉽지 않다.
ㅁ. 파일의 정렬과 병합이 용이하다.

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

정답 3번

출제 포인트

해싱의 주소 계산 방식과 충돌·오버플로의 관계, 정렬·병합에 대한 특성을 구분하는 문제이다.

핵심 개념

해싱은 키값에 해시 함수를 적용하여 레코드가 저장된 주소를 계산하고, 키를 차례로 비교하지 않고 해당 주소에 접근하는 방식이다.
충돌은 서로 다른 키가 같은 홈 주소로 변환되는 현상이고, 오버플로는 해당 버킷의 슬롯이 모두 차서 레코드를 더 저장할 수 없는 상태이다. 충돌이 생겨도 버킷에 빈 슬롯이 있으면 곧바로 오버플로가 되는 것은 아니다.
해싱은 키의 순서를 보존하지 않으므로 순차 처리나 정렬·병합에는 적합하지 않다.

선지별 해설

① X — ㄱ과 ㄷ은 옳지 않지만 ㄴ은 옳으므로, ‘ㄱ, ㄴ, ㄷ’은 옳지 않은 설명만을 모은 조합이 아니다. 올바르게는 ㄴ을 제외하고 ㅁ을 포함한 ‘ㄱ, ㄷ, ㅁ’이다.
② X — ㄹ은 같은 주소가 산출될 때 충돌과 그에 따른 오버플로 처리 문제가 생길 수 있다는 옳은 설명이므로, ‘ㄴ, ㄹ, ㅁ’은 옳지 않은 설명만을 모은 조합이 아니다. 또한 ㄴ도 옳다. 올바른 조합은 ‘ㄱ, ㄷ, ㅁ’이다.
③ O — ㄱ의 ‘언더플로 처리’는 ‘오버플로 처리’로 고쳐야 한다. ㄷ의 ‘항상 오버플로가 발생한다’는 표현은 틀리며, 충돌이 발생해도 해당 버킷에 빈 슬롯이 있으면 오버플로는 발생하지 않는다. ㅁ도 틀리며, 해싱은 키의 정렬 순서를 유지하지 않아 파일의 정렬과 병합이 용이하지 않다. 따라서 옳지 않은 것은 ㄱ, ㄷ, ㅁ이다.
④ X — ㄷ과 ㅁ은 옳지 않지만 ㄹ은 옳으므로, ‘ㄷ, ㄹ, ㅁ’은 옳지 않은 설명만을 모은 조합이 아니다. 올바르게는 ㄹ을 제외하고 ㄱ을 포함한 ‘ㄱ, ㄷ, ㅁ’이다.

NoteBase

오답 포인트

ㄱ의 ‘언더플로’를 자연스러운 용어로 넘기거나, 충돌이 일어나면 무조건 오버플로라고 생각하기 쉽다. 충돌은 같은 주소가 계산된 현상이고, 오버플로는 저장 공간까지 부족해진 상태이므로 둘은 동일하지 않다. 또한 해싱의 빠른 직접 검색이라는 장점이 정렬·병합에도 이어지는 것은 아니다.

다시 나오면 이렇게 판단

해싱 선지는 ‘키로 주소를 계산하는가’, ‘충돌과 오버플로를 구별했는가’, ‘키 순서를 보존하는가’의 세 기준으로 판단한다. 충돌은 같은 주소, 오버플로는 버킷의 빈 슬롯 부족, 정렬·병합은 순서 미보존 때문에 불리하다고 구분한다.

암기 포인트

해싱은 키 비교보다 주소 계산, 충돌은 주소 중복, 오버플로는 공간 부족, 정렬·병합은 불리이다.

NoteBase

문 17. UNIX 명령어에 대한 설명으로 옳지 않은 것은 모두 몇 개인가?

ㄱ. ls : 현재 디렉터리의 파일 목록을 표시한다.
ㄴ. fork : 현재의 프로세스를 복제하여 새로운 자식 프로세스를 생성한다.
ㄷ. chmod : 파일에 대한 개인, 그룹, 타인에 대한 접근 권한을 변경한다.
ㄹ. finger : 파일의 차이를 비교하여 출력한다.
ㅁ. lp : 현재 운영체제의 버전 정보를 출력한다.
ㅂ. mount : 기존 파일 시스템에 새로운 파일 시스템을 서브 디렉터리에 연결한다.

① 1개
② 2개
③ 3개
④ 4개

정답 2번

출제 포인트

UNIX의 주요 명령과 시스템 호출이 수행하는 기능을 구별하고, 설명이 틀린 항목의 개수를 판단하는 문제이다.

핵심 개념

ls는 디렉터리의 파일 목록을 표시하고, fork는 현재 프로세스를 복제해 자식 프로세스를 생성한다. chmod는 파일이나 디렉터리의 접근 권한을 변경하며, mount는 파일 시스템을 디렉터리에 연결한다.
finger는 사용자 계정과 로그인 관련 정보를 조회하는 명령이고, lp는 파일을 프린터의 인쇄 대기열로 보내는 명령이다. 파일 차이 비교에는 diff를 사용하고, 운영체제·커널 정보 확인에는 uname을 사용한다.

선지별 해설

① X — 옳지 않은 설명은 ㄹ과 ㅁ의 2개이므로 '1개'가 아니다. 틀린 표현은 '1개'이며, 올바르게는 '2개'이다.
② O — ㄱ의 ls, ㄴ의 fork, ㄷ의 chmod, ㅂ의 mount 설명은 옳고, ㄹ의 finger와 ㅁ의 lp 설명은 옳지 않다. 따라서 옳지 않은 것은 2개이다.
③ X — 옳지 않은 설명은 2개이므로 '3개'가 아니다. 틀린 표현은 '3개'이며, 올바르게는 '2개'이다.
④ X — 옳지 않은 설명은 2개이므로 '4개'가 아니다. 틀린 표현은 '4개'이며, 올바르게는 '2개'이다.
세부 판단: ㄹ은 X이다. finger는 사용자 정보를 조회하며, 파일 차이는 diff가 비교한다. ㅁ은 X이다. lp는 인쇄 작업을 요청하며, 운영체제·커널 정보는 uname으로 확인한다.

NoteBase

오답 포인트

ㄹ에서는 finger와 diff의 기능을, ㅁ에서는 lp와 uname의 기능을 바꾸어 놓았다. 명령 이름만 익숙하다고 판단하면 틀린 항목의 개수를 잘못 셀 수 있다.

다시 나오면 이렇게 판단

각 설명을 하나씩 O/X로 판정한 뒤 X의 개수와 선택지의 개수를 대응한다. 특히 사용자 정보는 finger, 파일 비교는 diff, 인쇄는 lp, 시스템 정보는 uname으로 구별한다.

암기 포인트

finger는 사용자, diff는 차이, lp는 인쇄, uname은 시스템 정보이다.

NoteBase

문 18. 블랙박스(Black box) 시험 기법이 아닌 것은 몇 개인가?

ㄱ. 프로그램의 입력 도메인을 시험사례가 산출될 수 있는 데이터의 클래스로 분류하는 방법이다.
ㄴ. 오류를 감각과 경험으로 찾아본다.
ㄷ. 프로그램 내의 변수 정의의 위치와 변수들의 사용에 따라 프로그램 검사 경로를 선택하는 조건 구조 검사방법이다.
ㄹ. 모듈 내의 논리적인 조건을 수행하는 시험 사례 설계방법이다.
ㅁ. 똑같은 기능의 소프트웨어를 두 개 이상 개발하여 발생 가능한 오류 확률을 최소화하는 것이다.

① 1개
② 2개
③ 3개
④ 4개

정답 2번

출제 포인트

블랙박스 시험과 화이트박스 시험을 구분하고, 제시된 기법 중 블랙박스 시험이 아닌 것의 개수를 판단하는 문제이다.

핵심 개념

블랙박스 시험은 프로그램의 내부 구조를 보지 않고 입력과 출력, 외부 기능을 기준으로 오류를 찾는 시험이다. 동치 분할, 오류 예측, 비교 시험 등이 이에 해당한다.
화이트박스 시험은 프로그램 내부의 논리 구조, 실행 경로, 조건, 변수의 정의와 사용 관계를 기준으로 시험하는 방법이다. 데이터 흐름 시험과 조건 시험 등이 이에 해당한다.
ㄱ은 입력 도메인을 데이터 클래스로 나누는 동치 분할이므로 블랙박스 시험이다. ㄴ은 경험과 직관으로 오류 가능성이 큰 부분을 찾는 오류 예측이므로 블랙박스 시험이다. ㄷ은 변수의 정의와 사용을 따라 경로를 고르는 데이터 흐름 시험이므로 화이트박스 시험이다. ㄹ은 모듈 내부의 논리 조건을 검사하는 조건 시험이므로 화이트박스 시험이다. ㅁ은 같은 기능의 여러 소프트웨어를 독립적으로 개발하여 결과를 비교하는 비교 시험에 해당하므로 블랙박스 시험으로 분류한다. 따라서 블랙박스 시험이 아닌 것은 ㄷ과 ㄹ의 2개이다.

NoteBase

선지별 해설

① X — 블랙박스 시험이 아닌 것은 1개가 아니라 ㄷ과 ㄹ의 2개이다.
② O — ㄷ의 데이터 흐름 시험과 ㄹ의 조건 시험은 내부 구조를 검사하는 화이트박스 시험이므로 2개가 맞다.
③ X — 블랙박스 시험이 아닌 것은 3개가 아니라 2개이다. ㅁ의 비교 시험은 여러 구현의 실행 결과를 비교하는 블랙박스 시험에 해당한다.
④ X — 블랙박스 시험이 아닌 것은 4개가 아니라 2개이다. ㄱ의 동치 분할, ㄴ의 오류 예측, ㅁ의 비교 시험은 블랙박스 시험이다.

오답 포인트

ㅁ을 소프트웨어 신뢰성 설계 기법으로만 보고 블랙박스 시험에서 제외하기 쉽다. 이 문항에서는 동일 기능의 여러 프로그램 결과를 대조하는 비교 시험을 설명한 것으로 보아 블랙박스 시험으로 분류한다. 또한 ㄷ의 '변수 정의와 사용', ㄹ의 '모듈 내 논리적인 조건'은 모두 내부 구조를 본다는 표시이다.

다시 나오면 이렇게 판단

입력·출력과 외부 기능만 보면 블랙박스, 경로·조건·변수의 정의와 사용처럼 프로그램 내부 구조를 보면 화이트박스로 판단한다. 각 항목을 먼저 두 범주로 분류한 뒤 문제에서 요구한 범주의 개수를 센다.

암기 포인트

동치 분할·오류 예측·비교 시험은 블랙박스, 데이터 흐름·조건 시험은 화이트박스이다.

NoteBase

문 19. 다음 엑셀표는 3년간 연이율 4%로 매월 적립하는 월 복리 정기적금의 만기지급금을 계산한 결과이다. 셀 C2에 들어갈 수식으로 옳은 것은?(단, 만기지급금의 10원 단위 미만은 절삭한다)

	A	B	C
1	성명	월적립액	만기지급금
2	김**	₩30,000	₩1,145,440
3	이**	₩50,000	₩1,909,070

① ROUNDDOWN(FV(4%, 3*12, -B2), -1)

② ROUNDDOWN(FV(4%, 3*12, -B2), -2)

③ ROUNDDOWN(FV(4%/12, 3*12, -B2), -1)

④ ROUNDDOWN(FV(4%/12, 3*12, -B2), -2)

정답 3번

출제 포인트

월 복리 정기적금의 미래가치를 구할 때 FV 함수의 이율과 기간을 월 단위로 맞추고, ROUNDDOWN 함수로 10원 단위 미만을 절삭할 수 있는지를 묻는 문제이다.

조건 정리

연이율은 4%이므로 월 이율은 4%/12이다. 적립 기간은 3년이므로 전체 납입 횟수는 3 × 12 = 36회이다. 월 적립액은 B2이며, 만기지급금을 양수로 표시하기 위해 FV 함수의 정기 지급액 인수에는 -B2를 넣는다. 10원 단위 미만 절삭은 ROUNDDOWN의 자릿수 인수에 -1을 사용한다.

사용할 공식

수식은 ROUNDDOWN(FV(4%/12, 3*12, -B2), -1)이다. FV(rate, nper, pmt)는 한 기간의 이율, 전체 기간 수, 매 기간 지급액을 이용해 미래가치를 구한다. 월마다 적립하고 이자가 월 복리로 붙으므로 rate와 nper를 모두 월 단위로 환산해야 한다.

NoteBase

계산 과정

1. 연이율 4%를 12로 나누어 월 이율 4%/12를 구한다.
2. 3년에 12를 곱하여 납입 횟수 36회를 구한다.
3. FV(4%/12, 3*12, -B2)로 만기 미래가치를 양수로 구한다.
4. ROUNDDOWN(미래가치, -1)을 적용해 일의 자리, 즉 10원 단위 미만을 버린다. B2가 30,000원이면 1,145,440원, 50,000원이면 1,909,070원이 되어 표와 일치한다.

선지 판단

① X — FV의 rate에 연이율 4%를 그대로 넣어 매월 4%가 적용된다. 월 복리의 올바른 이율은 4%/12이므로 ROUNDDOWN(FV(4%/12, 3*12, -B2), -1)로 고쳐야 한다.
② X — rate에 연이율 4%를 그대로 넣은 점과 ROUNDDOWN의 자릿수를 -2로 지정한 점이 모두 틀리다. -2는 100원 단위 미만을 절삭하므로, 올바르게는 월 이율 4%/12와 10원 단위 미만 절삭을 뜻하는 -1을 사용해야 한다.
③ O — 월 이율 4%/12, 전체 36회, 지급액 -B2, 10원 단위 미만 절삭을 위한 -1이 모두 조건에 맞는다. 계산 결과도 표의 만기지급금과 일치한다.
④ X — FV 부분은 맞지만 ROUNDDOWN의 -2가 100원 단위 미만 절삭을 뜻하므로 조건과 다르다. 10원 단위 미만을 절삭하려면 마지막 인수를 -1로 고쳐야 한다.

오답 포인트

①·②는 연이율 4%를 월 이율처럼 그대로 넣게 하는 함정이다. ②·④의 -2는 10원 단위가 아니라 100원 단위 미만을 버린다. ROUNDDOWN에서 -1은 십의 자리까지 남기고 그보다 작은 일의 자리를 버린다.

다시 나오면 이렇게 판단

복리 주기와 납입 주기가 월 단위이면 연이율을 12로 나누고 연수를 12배하여 FV의 rate와 nper를 같은 단위로 맞춘다. 절삭 자릿수는 -1이면 10원 단위 미만, -2이면 100원 단위 미만이 제거된다고 판단한다.

암기 포인트

월 복리는 연이율÷12·연수×12, 10원 단위 미만 절삭은 ROUNDDOWN의 -1이다.

NoteBase

문 20. 다음 중 안드로이드의 특징으로 옳지 않은 것은?

① 개방된 환경인만큼 개발 툴과 관련 문서를 모두 무료로 제공한다.

② 운영체제의 핵심이라고 할 수 있는 커널은 공개 운영 체제인 리눅스에 기반한다.

③ 플랫폼에 내장된 빌트인 프로그램과 사용자가 만든 프로그램이 동일한 API를 사용하지만 모든 프로그램이 평등하지는 않다.

④ 검증된 많은 라이브러리를 대거 포함하고 있어 대부분의 기능은 별도의 외부 라이브러리를 사용할 필요가 없다.

정답 3번

출제 포인트

안드로이드의 개방성, 리눅스 커널 기반 구조, 응용 프로그램 간 동등성, 라이브러리 제공이라는 기본 특징을 구별하는 문제이다.

핵심 개념

안드로이드는 리눅스 커널 위에 라이브러리와 애플리케이션 프레임워크를 두고, 개발자가 공개된 도구와 API를 이용해 응용 프로그램을 만들 수 있도록 한 개방형 모바일 플랫폼이다. 플랫폼에 기본 탑재된 프로그램과 사용자가 설치한 프로그램은 같은 애플리케이션 프레임워크의 API를 이용하며, 기본적으로 응용 프로그램이라는 동등한 지위에서 실행된다.

선지별 해설

① O — 안드로이드는 개방형 개발 환경을 지향하며, 애플리케이션 개발에 필요한 SDK 등의 개발 도구와 관련 문서를 무료로 제공한다.
② O — 안드로이드의 커널은 공개 운영체제인 리눅스 커널을 기반으로 하며, 장치 관리·메모리 관리·프로세스 관리 등 운영체제의 핵심 기능을 담당한다.
③ X — 틀린 부분은 '모든 프로그램이 평등하지는 않다'이다. 안드로이드에서는 플랫폼에 내장된 응용 프로그램과 사용자가 만든 응용 프로그램이 같은 API를 사용하며, 기본적으로 모든 응용 프로그램을 동등하게 취급한다. 따라서 '모든 프로그램이 평등하다'로 고쳐야 한다.
④ O — 안드로이드는 그래픽, 데이터베이스, 멀티미디어 등 다양한 기능을 지원하는 라이브러리를 플랫폼에 포함하므로 일반적인 기능은 이를 활용해 구현할 수 있다.

오답 포인트

③은 '동일한 API를 사용한다'는 맞는 설명 뒤에 '평등하지는 않다'라는 부정을 붙인 함정이다. 안드로이드의 대표적 특징은 기본 탑재 응용 프로그램과 개발자가 만든 응용 프로그램을 원칙적으로 동등하게 취급한다는 점이다.

NoteBase

다시 나오면 이렇게 판단

안드로이드의 기본 특징은 '개방형 개발 환경—리눅스 커널—풍부한 라이브러리—응용 프로그램의 동등성'으로 연결해 판단한다. 이 가운데 응용 프로그램에 차등적 지위가 있다고 단정하는 표현은 옳지 않은 설명으로 본다.

암기 포인트

안드로이드: 개방형, 리눅스 커널, 풍부한 라이브러리, 모든 응용 프로그램은 기본적으로 동등하다.

NoteBase