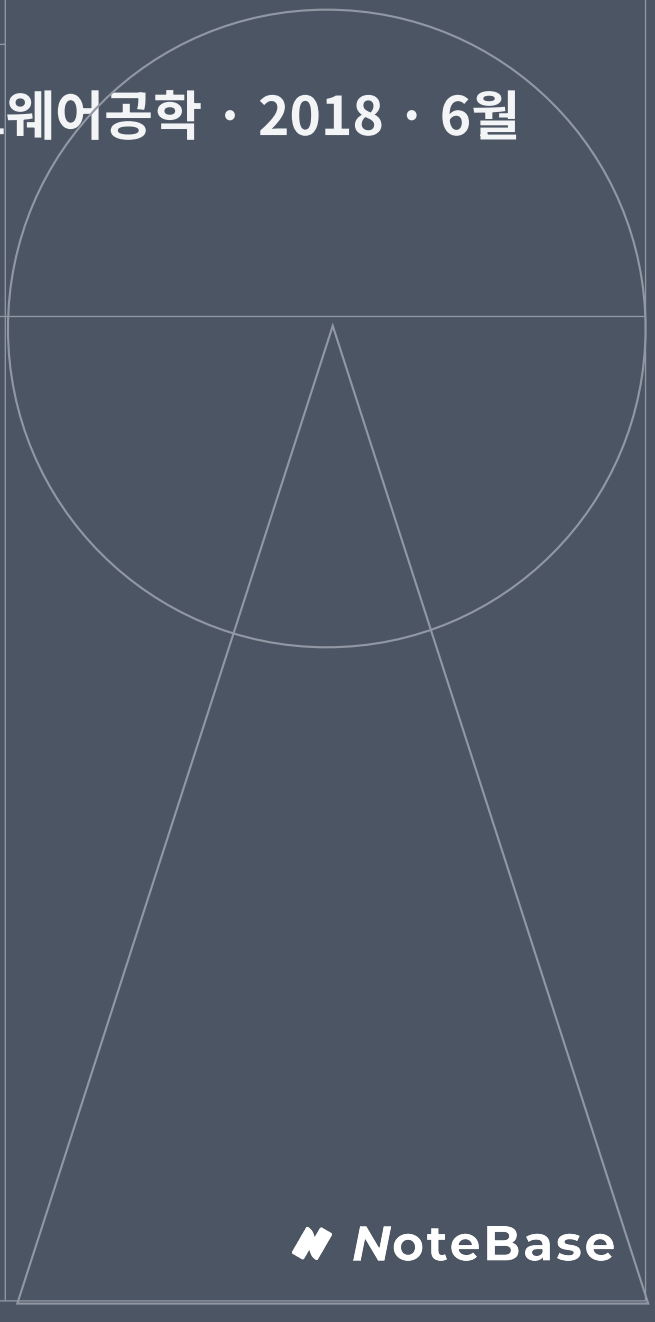
기출OX 해설집

공무원/서울시/7급 · 소프트웨어공학 · 2018 · 6월



NoteBase

기출OX 해설집

공무원/서울시/7급 · 소프트웨어공학 · 2018 · 6월

공무원/서울시/7급 · 소프트웨어공학 · 2018 · 6월

문 1. 소프트웨어 개발 계획 단계에서 도출해야 할 중요한 2가지 요소로 가장 옳은 것은?

① 비용과 일정

② 시간과 일정

③ 자원과 도구

④ 비용과 자원

정답 1번

출제 포인트

이 문제는 소프트웨어 개발 계획 단계에서 대표적으로 도출하는 핵심 관리 요소가 비용과 일정이라는 점을 묻는 문제이다.

핵심 개념

소프트웨어 개발 계획은 프로젝트를 수행하기 전에 필요한 비용, 자원, 기간을 추정하여 실행 가능성과 관리 기준을 정하는 단계이다.
이 단계의 대표 산출 두 축은 얼마의 비용이 드는지와 어떤 일정으로 진행할 것인지이다.

선지별 해설

1번 O: 비용과 일정은 개발 계획 단계의 대표 산출이다. 프로젝트를 시작하기 전에 예산 규모와 수행 일정을 정해야 통제 기준이 세워진다.
2번 X: 틀린 부분은 '시간과 일정'이다. 시간은 일정 산정에 포함되는 개념이어서 서로 독립된 두 핵심 요소로 보기 어렵다. 올바르게는 비용과 일정이다.
3번 X: 틀린 부분은 '도구'이다. 도구는 개발을 지원하는 수단일 뿐 계획 단계에서 반드시 도출해야 하는 대표 두 요소로 보지 않는다. 올바르게는 비용과 일정이다.
4번 X: 틀린 부분은 '자원'이다. 자원 계획도 필요하지만 이 문항이 묻는 대표적인 두 핵심 관리 요소의 짝은 비용과 일정이다. 올바르게는 비용과 일정이다.

NoteBase

오답 포인트

시간, 일정, 자원은 모두 계획 단계와 관련된 말이라 비슷해 보이지만 시험은 대표 산출의 짝을 묻는다. 일정과 겹치는 시간을 고르거나 비용에 포함되어 관리되는 자원을 고르면 틀리기 쉽다.

다시 나오면 이렇게 판단

개발 계획 단계 문제에서는 먼저 '얼마가 드는가'와 '언제 끝나는가'를 찾는다. 비용과 일정이 함께 제시되면 대표적인 정답 조합으로 판단할 수 있다.

암기 포인트

개발 계획의 핵심 두 축은 비용과 일정이다.

NoteBase

문 2. 객체지향 소프트웨어 개발 프로세스 UP(Unified Process)의 특징으로 가장 옳지 않은 것은?

① 반복적(iterative)이고 점진적(incremental)이다.

② 유스케이스기반(use-case driven)으로 한다.

③ 아키텍처 중심(architecture centric)이다.

④ E-R 다이어그램 기반으로 되어 있다.

정답 4번

출제 포인트

이 문제는 객체지향 개발 프로세스인 UP의 핵심 특징과 전통적 데이터 모델링 개념을 구별하는 문제이다.

핵심 개념

UP는 반복적이고 점진적으로 시스템을 완성해 가는 객체지향 개발 프로세스이다. 또한 유스케이스 기반으로 요구사항을 이끌고, 아키텍처 중심으로 초기부터 구조를 안정화하는 것이 핵심이다. 반면 E-R 다이어그램은 데이터베이스의 엔터티와 관계를 표현하는 모델링 도구이지 UP 자체의 기반 개념은 아니다.

선지별 해설

① O: UP는 한 번에 전체를 완성하는 방식이 아니라 반복을 거치며 기능을 확장하는 iterative, incremental 접근을 취한다.
② O: UP는 요구사항을 유스케이스로 조직하고 분석·설계·테스트의 기준으로 삼는 use-case driven 방법론이다.
③ O: UP는 시스템의 골격을 초기에 정립하고 반복마다 다듬는 architecture centric 프로세스이다.
④ X: 틀린 부분은 'E-R 다이어그램 기반'이다. UP는 E-R 다이어그램을 기반으로 하는 방법론이 아니라 UML을 활용하는 객체지향 개발 프로세스이다.

오답 포인트

객체지향 개발 프로세스의 특징과 데이터베이스 모델링 도구를 같은 층위의 개념으로 섞으면 틀리기 쉽다. ④는 개발 프로세스의 성격을 묻는데 E-R 다이어그램이라는 별도 모델링 기법을 끼워 넣어 혼동을 유도한다.

다시 나오면 이렇게 판단

UP 문제에서는 반복적, 점진적, 유스케이스 기반, 아키텍처 중심이라는 세 축을 먼저 확인하면 된다. 선지에 E-R 다이어그램처럼 데이터 모델링 용어가 나오면 UP의 본질적 특징인지부터 구분하면 정답을 고를 수 있다.

암기 포인트

UP는 반복·점진, 유스케이스, 아키텍처 중심이고 E-R은 데이터 모델링 도구이다.

NoteBase

문 3. 역공학(reverse engineering)에 대한 설명으로 가장 옳지 않은 것은?

① 원시코드를 이용하여 원시코드 이상의 추상화된 표현으로 나타내는 것이며 코드를 분석하는 과정이다.

② 수작업으로 하는 것이 어렵기 때문에 자동화 도구를 사용한다.

③ 시스템의 성능, 효율성 개선을 목표로 한다.

④ 원시코드로부터 시스템 명세서를 개발하는 과정이다.

정답 3번

출제 포인트

이 문제는 역공학의 목적과 산출물을 묻고, 성능 개선을 목표로 하는 재공학과 구별하는지를 확인하는 문제이다.

핵심 개념

역공학: 기존 소프트웨어를 분석하여 코드보다 상위 수준의 설계, 구조, 명세를 추출하는 과정이다. 키워드: 분석, 추상화, 이해.
재공학: 기존 소프트웨어를 수정·변환하여 성능, 품질, 유지보수성을 높이는 과정이다. 키워드: 개선, 변환, 품질 향상.

선지별 해설

① O
원시코드를 이용해 원시코드 이상의 추상화된 표현을 만들어 코드를 분석한다는 설명이므로 역공학의 정의에 맞는다.

② O
규모가 크고 복잡한 시스템은 수작업만으로 구조를 파악하기 어렵기 때문에 역공학 도구 같은 자동화 도구를 활용하는 경우가 많다.

③ X
틀린 부분은 "시스템의 성능, 효율성 개선을 목표로 한다"는 표현이다.
실제로 성능과 효율성 개선은 재공학의 목표에 가깝고, 역공학의 직접 목적은 기존 시스템을 분석하여 구조와 명세를 이해하고 추출하는 데 있다.

④ O
원시코드로부터 시스템 명세서나 설계 정보를 복원하는 것은 역공학의 대표적 결과이므로 옳은 설명이다.

NoteBase

오답 포인트

역공학과 재공학은 모두 기존 시스템을 대상으로 하므로 쉽게 혼동된다. 이 문항의 함정은 "분석하여 추출하는 과정"과 "개선하여 바꾸는 과정"을 일부러 섞어 놓은 ③의 표현에 있다.

다시 나오면 이렇게 판단

기존 코드를 보고 설계·명세·구조를 알아내면 역공학이고, 기존 시스템을 고쳐 성능·품질·유지보수성을 높이면 재공학으로 판단한다.

암기 포인트

역공학은 이해와 추출, 재공학은 개선과 변환이다.

NoteBase

문 4. ISO/IEC 9126 품질 모델 중 소프트웨어를 쉽게 변경할 수 있는 특성으로 가장 옳은 것은?

① 유지보수성(maintainability)

② 재사용성(reusability)

③ 신뢰성(reliability)

④ 사용 용이성(usability)

정답 1번

출제 포인트

이 문제는 ISO/IEC 9126 품질 특성 중 유지보수성과 재사용성·신뢰성·사용 용이성을 구별하는 문제이다.

핵심 개념

유지보수성: 소프트웨어를 분석, 수정, 시험하여 쉽게 변경할 수 있는 정도이다. 키워드는 변경 용이성이다.
재사용성: 소프트웨어 자산이나 구성요소를 다른 시스템이나 다른 용도에 다시 활용할 수 있는 정도이다. 키워드는 재활용 가능성이다.
신뢰성: 정해진 조건에서 요구된 기능을 오류 없이 안정적으로 수행하는 정도이다. 키워드는 안정성과 정확성이다.
사용 용이성: 사용자가 배우고 이해하고 조작하기 쉬운 정도이다. 키워드는 사용 편의성이다.

선지별 해설

① O: 유지보수성은 소프트웨어를 쉽게 수정하고 개선할 수 있는 성질이므로 문제의 '쉽게 변경할 수 있는 특성'과 정확히 대응한다.
② X: 틀린 부분은 '재사용성'이다. 재사용성은 기존 모듈이나 코드를 다른 곳에 다시 사용할 수 있는 성질이지, 현재 소프트웨어 자체를 쉽게 변경하는 성질이 아니다. 올바르게는 '다른 시스템이나 기능에 다시 활용할 수 있는 특성'이다.
③ X: 틀린 부분은 '신뢰성'이다. 신뢰성은 장애 없이 일관되게 동작하는 성질을 뜻한다. 올바르게는 '주어진 환경에서 요구 기능을 안정적으로 수행하는 특성'이다.
④ X: 틀린 부분은 '사용 용이성'이다. 사용 용이성은 사용자가 배우고 사용하기 쉬운 성질을 뜻한다. 올바르게는 '사용자 입장에서 이해와 조작이 쉬운 특성'이다.

오답 포인트

문제의 핵심 단어는 '변경'이다. 품질 특성 이름이 모두 추상적이어서 비슷해 보이지만, 변경은 유지보수성, 다시 사용은 재사용성, 안정 동작은 신뢰성, 사용 편의는 사용 용이성으로 각각 기준이 다르다.

NoteBase

다시 나오면 이렇게 판단

선지의 동사를 먼저 본다. 수정·개선·변경이면 유지보수성, 재사용이면 재사용성, 안정 운용이면 신뢰성, 배우기 쉽고 쓰기 쉬우면 사용 용이성으로 판단된다.

암기 포인트

유지보수성은 바꾸기 쉬움, 재사용성은 다시 쓰기 쉬움, 신뢰성은 고장 없이 안정적임, 사용 용이성은 쓰기 쉬움이다.

NoteBase

문 5. <보기>와 같이 프로그램을 변경하는 작업으로 가장 옳은 것은?

<보기>

변경 전

```
public class Painter {
  public double calculatePaintRequired(double row, double col, double paintPerUnit) {
                double area = row * col;
                double paintRequired = area/paintPerUnit;
                return paintRequired;
  }
}
```

변경 후

```
public class Painter {
  public double calculatePaintRequired(double row, double col, double paintPerUnit) {
                double area = calculateArea(row, col);
                double paintRequired = area/paintPerUnit;
                return paintRequired;
  }
   private double calculateArea(double row, double col) {
                double area = row * col;
                return area;
  }
}
```

① 재사용(reuse)

② 재공학(reengineering)

③ 역공학(reverse engineering)

④ 리팩토링(refactoring)

정답 4번

출제 포인트

이 문제는 외부 동작은 유지한 채 내부 구조만 개선하는 리팩토링과 재사용, 재공학, 역공학의 차이를 구분하는 문제이다.

핵심 개념

재사용은 기존 소프트웨어 자산을 다른 개발이나 기능 구현에 가져다 쓰는 것이다.
재공학은 기존 시스템을 분석하고 재구성하여 품질과 유지보수성을 높이는 포괄적 개선 활동이다.
역공학은 코드나 실행 결과로부터 설계, 명세, 구조를 거꾸로 추출하는 것이다.
리팩토링은 외부 기능은 바꾸지 않고 내부 코드 구조를 개선하는 것이다.
이 보기의 판단 기준은 기능 결과가 그대로인지, 그리고 변경의 초점이 내부 구조 개선인지이다.

NoteBase

기출OX 해설집 · 공무원/서울시/7급 · 소프트웨어공학 · 2018 · 6월

선지별 해설

① X
재사용은 이미 만들어진 코드, 모듈, 컴포넌트를 다른 곳에서 다시 활용하는 개념이다. 보기처럼 한 메서드 안의 계산식을 별도 메서드로 분리한 변경 자체를 재사용이라고 하지는 않는다.
틀린 표현은 '보기와 같은 변경 작업'을 재사용으로 본 점이다. 올바르게는 기존 자산을 다른 기능이나 시스템에서 활용할 때 재사용이라고 한다.

② X
재공학은 기존 소프트웨어를 분석, 개편, 재구성하는 넓은 범주의 활동이다. 리팩토링이 재공학의 일부로 포함될 수는 있지만, 이 문항은 코드 변경의 직접적 성격을 묻고 있으므로 더 정확한 답은 리팩토링이다.
틀린 표현은 이 구체적 메서드 추출 작업을 바로 재공학이라고 단정한 점이다. 올바르게는 동작 보존형 내부 구조 개선은 리팩토링이다.

③ X
역공학은 구현된 코드에서 설계나 명세를 추론해 내는 활동이다. 보기에서는 설계를 추론한 것이 아니라 calculateArea 메서드를 새로 만들어 실제 코드를 재구성하였다.
틀린 표현은 코드 구조 변경 작업을 역공학으로 본 점이다. 올바르게는 코드에서 설계 정보를 거꾸로 얻어낼 때 역공학이라고 한다.

④ O
변경 전후 모두 면적을 구한 뒤 필요한 페인트 양을 계산한다는 외부 기능은 같다. 달라진 점은 row * col 계산을 calculateArea(row, col)로 분리하여 가독성, 중복 관리, 유지보수성을 높인 점이며, 이는 대표적인 리팩토링인 메서드 추출이다.

오답 포인트

재공학과 리팩토링을 같은 말처럼 섞어 고르기 쉽다는 점이 함정이다. 재공학은 더 넓은 상위 개념이고, 이 보기처럼 동작은 그대로 둔 채 코드를 정리하는 직접적 작업명은 리팩토링이다. 역공학은 '분석', 리팩토링은 '구조 개선'이라는 차이도 자주 뒤바뀐다.

다시 나오면 이렇게 판단

코드가 바뀌어도 외부 기능과 계산 결과가 그대로이면 먼저 리팩토링 가능성을 본다. 특히 긴 메서드를 나누거나 중복 코드를 정리하거나 이름을 개선한 경우처럼 내부 구조 개선이 중심이면 리팩토링이다. 반대로 코드에서 설계나 명세를 추출하는 것이 중심이면 역공학이고, 기존 자산을 다른 개발에 가져다 쓰는 것이 중심이면 재사용이다.

NoteBase

암기 포인트

동작 그대로, 구조만 개선은 리팩토링이다. 코드에서 설계를 거꾸로 읽어내면 역공학이다.

NoteBase

문 6. 테스트의 목표를 수립하고, 테스트의 범위를 선정하는 활동으로 가장 옳은 것은?

① 테스트 계획

② 테스트 설계

③ 테스트 실행

④ 테스트 평가

정답 1번

출제 포인트

이 문제는 테스트 활동을 계획, 설계, 실행, 평가 단계로 구분할 수 있는지를 묻는다.

핵심 개념

테스트 계획은 테스트의 목표와 범위, 일정, 자원을 정하는 활동이다.
테스트 설계는 어떤 테스트 케이스와 입력 데이터, 예상 결과로 검증할지를 구체화하는 활동이다.
테스트 실행은 준비된 테스트 케이스를 실제로 수행하고 결과를 기록하는 활동이다.
테스트 평가는 실행 결과를 분석하여 품질 수준과 종료 여부를 판단하는 활동이다.

선지별 해설

1번 O: 테스트의 목표를 수립하고 테스트 범위를 선정하는 일은 테스트 계획의 핵심 업무이므로 옳다.
2번 X: 틀린 부분은 목표 수립과 범위 선정을 테스트 설계로 본 점이다. 테스트 설계는 실제로는 테스트 케이스, 입력값, 예상 결과, 절차를 구체화하는 단계이다.
3번 X: 틀린 부분은 목표 수립과 범위 선정을 테스트 실행으로 본 점이다. 테스트 실행은 설계된 테스트를 실제로 수행하여 결과를 확인하는 단계이다.
4번 X: 틀린 부분은 목표 수립과 범위 선정을 테스트 평가로 본 점이다. 테스트 평가는 실행 결과를 분석하고 품질 달성 여부를 판단하는 단계이다.

오답 포인트

테스트 계획과 테스트 설계가 모두 사전 활동이어서 혼동하기 쉽다. 범위와 목표를 정하면 계획이고, 구체적인 케이스와 절차를 만들면 설계이다.

다시 나오면 이렇게 판단

문장에서 무엇을 시험할지, 어디까지 시험할지를 묻으면 계획으로 본다. 어떤 케이스와 데이터로 시험할지를 묻으면 설계, 실제 수행이면 실행, 결과 분석과 품질 판정이면 평가로 본다.

NoteBase

암기 포인트

목표·범위는 계획, 케이스·절차는 설계, 수행은 실행, 결과 판정은 평가이다.

NoteBase

문 7. 소프트웨어 아키텍처에 대한 설명으로 옳은 것을 <보기>에서 모두 고른 것은?

<보기>
ㄱ. 품질 속성을 포함한 비기능적인 요구사항이 주요한 아키텍처 설계 목표가 된다.
ㄴ. 서브시스템으로 분할할 때 서브시스템 간의 결합도는 낮추고 서브시스템 내의 응집도는 높이는 방향으로 설계가 되어야 한다.
ㄷ. RUP(Rational Unified Process)는 주요 위험 요인들을 파악하고 시스템의 초기 골격 아키텍처를 먼저 구현하여 이를 바탕으로 반복적인 개발을 통해 시스템을 완성해 간다.
ㄹ. ATAM은 시나리오를 기반으로 하여 아키텍처를 평가하는 방법 중 하나이다.

① ㄱ, ㄴ
② ㄱ, ㄴ, ㄷ
③ ㄱ, ㄴ, ㄷ, ㄹ
④ ㄴ, ㄹ

정답 3번

출제 포인트

소프트웨어 아키텍처의 핵심 원리로서 비기능 요구사항, 결합도·응집도, RUP의 아키텍처 중심 반복 개발, ATAM의 시나리오 기반 평가를 함께 묻는 문제이다.

핵심 개념

비기능 요구사항은 성능·가용성·보안·변경 용이성 같은 품질 속성을 뜻하며 아키텍처 설계의 주요 목표가 된다.
서브시스템 분할 원리는 서브시스템 간 결합도는 낮추고 서브시스템 내부 응집도는 높이는 것이다.
RUP는 위험 중심의 반복적 개발 프로세스로, 초기에 실행 가능한 골격 아키텍처를 마련한 뒤 반복적으로 시스템을 확장한다.
ATAM은 품질 속성 시나리오를 기준으로 아키텍처의 장단점과 위험을 평가하는 방법이다.

NoteBase

선지별 해설

1번 X: ㄱ, ㄴ은 모두 옳다. 다만 ㄷ의 RUP 설명도 옳고 ㄹ의 ATAM 설명도 옳으므로 'ㄱ, ㄴ'만 고른 조합은 불완전하다. 올바르게는 'ㄱ, ㄴ, ㄷ, ㄹ'이다.
2번 X: ㄱ, ㄴ, ㄷ은 모두 옳다. 그러나 ㄹ도 옳으므로 'ㄱ, ㄴ, ㄷ'만 고른 조합이 틀리다. 올바르게는 'ㄱ, ㄴ, ㄷ, ㄹ'이다.
3번 O: ㄱ은 품질 속성을 포함한 비기능 요구사항이 아키텍처 목표가 된다는 설명으로 옳다. ㄴ은 낮은 결합도와 높은 응집도를 말하므로 옳다. ㄷ은 RUP의 위험 중심·초기 아키텍처·반복 개발 특징을 설명하므로 옳다. ㄹ은 ATAM이 시나리오 기반 아키텍처 평가 방법이라는 설명으로 옳다.
4번 X: ㄴ과 ㄹ은 옳다. 그러나 ㄱ도 옳고 ㄷ도 옳으므로 'ㄴ, ㄹ'만 고른 조합은 틀리다. 올바르게는 'ㄱ, ㄴ, ㄷ, ㄹ'이다.

오답 포인트

RUP를 단순한 개발 절차 정도로만 기억하거나 ATAM을 설계 기법으로 혼동하면 ㄷ 또는 ㄹ을 제외한 조합을 고르기 쉽다. 또한 조합형 문제에서는 개별 진술 O/X를 끝까지 확인하지 않고 중간에 멈추면 2번에서 멈추는 실수가 나온다.

다시 나오면 이렇게 판단

조합형 아키텍처 문제는 먼저 ㄱ~ㄹ 각각의 진위를 독립적으로 판정한 뒤, 마지막에 그 결과와 정확히 일치하는 조합을 찾는 방식으로 판단한다. 개념 키워드는 '품질 속성=비기능 요구사항', '낮은 결합도·높은 응집도', 'RUP=위험 중심 반복 개발', 'ATAM=시나리오 기반 평가'로 묶으면 된다.

암기 포인트

아키텍처는 품질 속성으로 설계하고, 모듈은 낮은 결합도·높은 응집도로 나누며, RUP는 초기에 골격 아키텍처를 세우고, ATAM은 시나리오로 평가한다.

NoteBase

문 8. <보기>에서 기술한 활동과 관련 있는 국제 표준으로 가장 옳은 것은?

<보기>
소프트웨어를 개발할 때 변경 관리가 필요한 항목을 선정하고 그 항목은 아무나 변경할 수 없도록 적절한 통제를 한다. 변경할 내용을 기록하고 그 기록과 실제 변경된 내용이 일치하는지 확인하는 활동이다.

① ISO/IEC/ISO 29119
② IEEE 829
③ IEEE 1042
④ ISO/IEC 9899

정답 3번

출제 포인트

이 문제는 소프트웨어 형상관리에서 형상항목 선정, 변경 통제, 변경 기록, 형상감사에 해당하는 활동을 어떤 표준과 연결해야 하는지 묻는다.

핵심 정리

IEEE 1042는 소프트웨어 형상관리 지침이다. 형상식별, 형상통제, 형상상태기록, 형상감사 같은 활동을 다룬다.
IEEE 829는 테스트 계획서, 테스트 케이스, 테스트 결과서 등 소프트웨어 테스트 문서를 다루는 표준이다.
ISO/IEC/IEEE 29119 계열은 소프트웨어 테스트 전반을 다루는 국제 표준이다.
ISO/IEC 9899는 C 프로그래밍 언어의 문법과 의미를 규정하는 표준이다.

선지별 해설

① X: 29119 계열은 소프트웨어 테스트 표준이다. 보기의 활동은 테스트 절차가 아니라 형상항목 선정과 변경 통제, 기록 일치 확인이므로 형상관리 영역이다. 틀린 표현은 이를 변경 관리 표준으로 본 점이다. 올바르게는 ISO/IEC/IEEE 29119 계열은 소프트웨어 테스트 관련 표준이다.
② X: IEEE 829는 소프트웨어 테스트 문서 표준이다. 테스트 계획서나 테스트 결과서 같은 문서 체계를 설명하는 표준이므로 형상통제나 형상감사를 직접 설명하는 표준이 아니다. 틀린 표현은 형상관리 활동과 연결한 점이다. 올바르게는 테스트 문서 표준이다.
③ O: IEEE 1042는 소프트웨어 형상관리 지침이므로 보기의 설명과 가장 잘 맞는다. 형상항목을 정하고 임의 변경을 통제하며 변경 내용을 기록하고, 그 기록과 실제 산출물이 일치하는지 확인하는 활동은 형상상태기록과 형상감사의 성격을 가진다.
④ X: ISO/IEC 9899는 C 언어 표준이다. 프로그래밍 언어의 구문과 의미를 정의하는 표준이지 소프트웨어 변경 관리 절차를 규정하는 표준이 아니다. 틀린 표현은 변경 관리 활동과 관련 있는 표준으로 본 점이다. 올바르게는 C 언어 표준이다.

NoteBase

오답 포인트

보기에 '변경'이라는 단어가 있어 테스트 변경이나 문서 갱신으로 오해하기 쉽다. 그러나 핵심 키워드는 '형상항목 선정', '임의 변경 방지', '기록과 실제 변경의 일치 확인'이며, 이는 소프트웨어 형상관리의 전형적 표현이다.

다시 나오면 이렇게 판단

표준 번호 문제는 번호보다 대표 키워드로 판단한다. 테스트와 테스트 문서가 보이면 29119나 829를 떠올리고, 형상항목 선정, 변경 통제, 상태기록, 감사가 보이면 IEEE 1042 같은 형상관리 표준으로 연결한다.

암기 포인트

1042는 형상관리, 829는 테스트 문서, 29119는 소프트웨어 테스트, 9899는 C 언어 표준이다.

NoteBase

문 9. 국제표준 ISO/IEC 9126과 비교하여 국제표준 ISO/IEC 25010에 추가된 소프트웨어 품질특성 항목으로 가장 옳은 것은?

① 사용성 (usability)
② 호환성 (compatibility)
③ 유지보수성 (maintainability)
④ 이식성 (portability)

정답 2번

출제 포인트

이 문제는 ISO/IEC 9126의 품질특성과 ISO/IEC 25010에서 새로 추가된 품질특성을 구별하는 문제이다.

핵심 개념

ISO/IEC 9126은 기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성의 6개 품질특성으로 구성된다.
ISO/IEC 25010은 이를 확장하여 기능 적합성, 성능 효율성, 호환성, 사용성, 신뢰성, 보안성, 유지보수성, 이식성의 8개 품질특성으로 정리한다.
핵심 차이는 25010에서 호환성과 보안성이 추가되었다는 점이다.

선지별 해설

① X: 사용성은 사용자가 소프트웨어를 쉽게 이해하고 학습하며 사용하는 정도를 뜻하는 품질특성이다. 이는 ISO/IEC 9126에도 이미 포함되어 있었다. 틀린 표현은 '추가된'이다. 올바르게는 '사용성은 9126과 25010에 모두 포함된 품질특성이다.'이다.
② O: 호환성은 다른 시스템이나 제품과 함께 정보를 교환하고 필요한 기능을 수행할 수 있는 정도를 뜻한다. 호환성은 ISO/IEC 25010에서 새로 명시된 품질특성이다.
③ X: 유지보수성은 소프트웨어를 수정, 개선, 시험하기 쉬운 정도를 뜻한다. 이는 ISO/IEC 9126에도 이미 존재했다. 틀린 표현은 '추가된'이다. 올바르게는 '유지보수성은 기존 9126에도 있던 품질특성이다.'이다.
④ X: 이식성은 한 환경에서 다른 환경으로 소프트웨어를 이전하기 쉬운 정도를 뜻한다. 이 역시 ISO/IEC 9126에 포함되어 있었다. 틀린 표현은 '추가된'이다. 올바르게는 '이식성은 9126과 25010에 공통으로 포함된 품질특성이다.'이다.

오답 포인트

9126과 25010의 공통 항목과 추가 항목을 섞어 묻는 함정이다. 사용성, 유지보수성, 이식성은 이름이 익숙해서 새 표준에만 있는 것처럼 착각하기 쉽지만, 실제 추가 항목은 호환성과 보안성이다.

NoteBase

다시 나오면 이렇게 판단

품질모델 비교 문제는 먼저 9126의 기존 6개 특성을 떠올린 뒤, 25010에서 늘어난 항목이 무엇인지 확인하면 된다. 기존에 없던 항목만 고르면 정답을 빠르게 판단할 수 있다.

암기 포인트

9126은 6개, 25010은 8개이며 추가 키워드는 호환성·보안성이다.

NoteBase

문 10. 유사한 클래스들 사이의 공유되는 속성과 동작을 묶어주는 추상화 기법으로 가장 옳은 것은?

① 집단화(aggregation)
② 추상화(abstraction)
③ 캡슐화(encapsulation)
④ 일반화(generalization)

정답 4번

출제 포인트

이 문제는 객체지향의 추상화 관련 개념 중 일반화와 집단화, 추상화, 캡슐화의 차이를 구별할 수 있는지를 묻는다.

핵심 개념

일반화는 여러 유사한 클래스의 공통 속성과 동작을 상위 클래스로 묶는 기법이다. 키워드는 공통점 추출, 상위 클래스, 상속이다.
추상화는 복잡한 대상에서 본질적인 특징만 뽑아내어 표현하는 더 넓은 개념이다. 키워드는 본질만 표현, 세부사항 생략이다.
집단화는 객체나 클래스 사이의 전체-부분 관계를 표현하는 연관이다. 키워드는 has-a 관계이다.
캡슐화는 데이터와 메서드를 하나로 묶고 내부 구현을 외부에서 직접 보지 못하게 하는 기법이다. 키워드는 정보 은닉이다.

선지별 해설

1번 X: 집단화는 전체와 부분을 묶는 관계 표현이지, 유사한 클래스들의 공통 속성과 동작을 상위 개념으로 정리하는 기법이 아니다. 틀린 표현은 "공유되는 속성과 동작을 묶어주는"이다. 올바르게는 "집단화는 객체 또는 클래스 사이의 전체-부분 관계를 표현한다"이다.
2번 X: 추상화는 본질적 특성만 남기고 세부를 생략하는 상위 개념이므로, 문제처럼 유사한 클래스들의 공통 속성과 동작을 하나의 상위 클래스로 묶는 구체적 기법을 바로 가리키지는 않는다. 틀린 표현은 "가장 옳은 것"에서 추상화를 일반화와 동일하게 본 점이다. 올바르게는 "이 상황을 직접 설명하는 기법은 일반화이고, 일반화는 추상화의 한 방식이다"이다.
3번 X: 캡슐화는 속성과 동작을 클래스 내부에 함께 두고 접근을 통제하는 개념이다. 공통점을 여러 클래스 사이에서 묶는 것이 아니라 한 클래스 내부를 보호하는 데 초점이 있다. 틀린 표현은 "유사한 클래스들 사이의 공유되는"이다. 올바르게는 "캡슐화는 한 클래스 내부의 데이터와 메서드를 묶고 외부 접근을 제한한다"이다.
4번 O: 일반화는 여러 하위 클래스에 공통으로 나타나는 속성과 연산을 추출하여 상위 클래스로 정의하는 기법이므로, 문제의 설명과 정확히 일치한다.

NoteBase

오답 포인트

2번의 추상화가 가장 큰 함정이다. 일반화도 넓게 보면 추상화의 한 형태이므로 단어만 보면 맞아 보이지만, 문제는 "유사한 클래스들 사이의 공통 속성과 동작을 묶는다"라는 구체적 상황을 제시하므로 정답은 일반화로 좁혀야 한다.

다시 나오면 이렇게 판단

문장에서 여러 클래스의 공통점을 뽑아 상위 클래스로 올리는지, 한 클래스 내부를 숨기는지, 전체-부분 관계를 나타내는지를 먼저 구분하면 된다. 공통점 추출이면 일반화, 내부 보호면 캡슐화, 전체-부분이면 집단화이다.

암기 포인트

공통점 추출은 일반화, 본질만 남기면 추상화, 내부를 감추면 캡슐화, 전체-부분은 집단화이다.

NoteBase

문 11. 객체지향 분석에서 기능 모델링에 대한 설명으로 가장 옳지 않은 것은?

① 객체 모델링, 동적 모델링에 이어 시스템을 기술하는 세 번째 단계이다.

② 시스템에 요구되는 객체들의 구조와 관계가 정립된다.

③ 동적 모델링에서 나타난 동작들이 어떠한 기능으로 이루어져 있는지 나타낸다.

④ UML 다이어그램 중 액티비티(activity) 다이어그램을 사용한다.

정답 2번

출제 포인트

이 문제는 객체지향 분석의 세 가지 모델인 객체 모델링, 동적 모델링, 기능 모델링의 역할 차이를 구분하는 문제이다.

핵심 개념

객체 모델링: 시스템을 구성하는 객체가 무엇인지와 객체들 사이의 구조·관계를 정리하는 모델이다. 결정 차이축: 정적인 구성과 관계 / 키워드: 객체, 속성, 관계
동적 모델링: 시간의 흐름에 따라 객체가 어떻게 반응하고 상태가 어떻게 변하는지를 나타내는 모델이다. 결정 차이축: 사건과 상태 변화 / 키워드: 이벤트, 상태, 동작
기능 모델링: 시스템이 어떤 기능을 수행하고 데이터가 어떻게 흐르며 변환되는지를 나타내는 모델이다. 결정 차이축: 처리 기능과 흐름 / 키워드: 기능, 처리, 데이터 흐름

NoteBase

선지별 해설

① O
기능 모델링은 객체 모델링과 동적 모델링을 보완하여 시스템이 수행하는 기능을 기술하는 단계로 본다. 객체의 구조를 본 뒤, 동작을 보고, 그 동작이 어떤 기능 처리로 이루어지는지 정리하는 설명이므로 옳다.

② X
'객체들의 구조와 관계'가 틀린 표현이다.
기능 모델링은 구조와 관계를 정리하는 모델이 아니라 시스템의 기능과 데이터 흐름을 나타내는 모델이다.
올바르게는 '시스템이 수행하는 기능과 처리 흐름이 정리된다'이다.

③ O
동적 모델링이 시간적 동작과 상태 변화를 보여 준다면, 기능 모델링은 그 동작이 어떤 처리 기능으로 실현되는지 표현한다. 기능 관점에서 시스템의 일을 설명한 것이므로 옳다.

④ O
UML에서는 기능 흐름이나 업무 처리 절차를 표현할 때 액티비티 다이어그램을 활용한다. 기능 모델링을 UML로 나타내는 설명으로 볼 수 있으므로 옳다.

오답 포인트

객체 모델링과 기능 모델링을 모두 '시스템 설명'으로 뭉뚱그리면 ②처럼 구조·관계를 기능 모델링에 잘못 배치하기 쉽다. 이 문항은 '무엇이 존재하는가'와 '무엇을 처리하는가'를 바꿔 넣는 함정을 사용한다.

다시 나오면 이렇게 판단

선지에 객체의 구조, 속성, 관계가 나오면 객체 모델링인지 먼저 본다. 선지에 처리 기능, 데이터 흐름, 업무 흐름이 나오면 기능 모델링으로 판단한다. 상태 변화, 이벤트, 시간 흐름이 나오면 동적 모델링으로 구분한다.

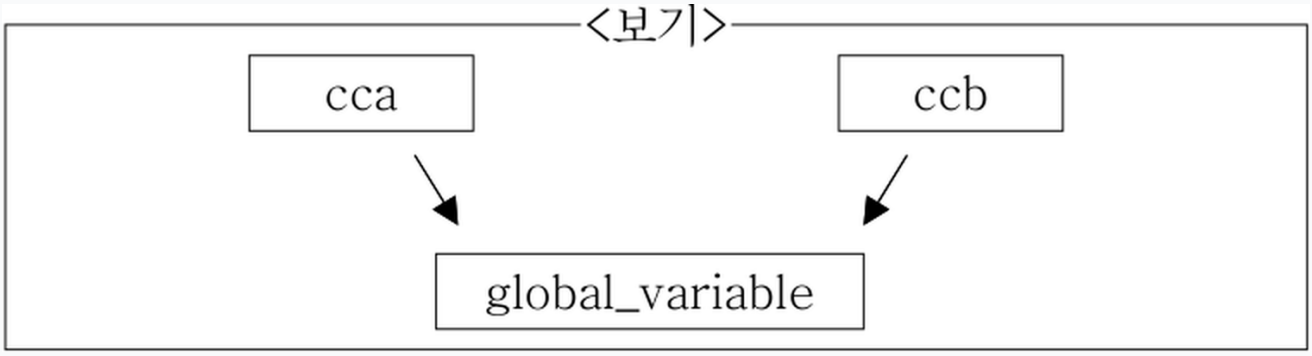
암기 포인트

객체 모델링은 무엇이 있는가, 동적 모델링은 어떻게 변하는가, 기능 모델링은 무엇을 처리하는가이다.

NoteBase

문 12. 두 모듈 간의 상호작용 정도를 나타내는 모듈 결합도에서 <보기>와 같이 두 모듈들이 같은 전역 데이터를 접근하는 결합도로 가장 옳은 것은?

<보기>



① 데이터 결합도(data coupling)

② 내용 결합도(content coupling)

③ 공통 결합도(common coupling)

④ 제어 결합도(control coupling)

정답 3번

출제 포인트

이 문제는 모듈들이 같은 전역 데이터를 공유할 때의 결합도를 다른 결합도 개념과 구별하는 문제이다.

핵심 개념

공통 결합도는 둘 이상의 모듈이 동일한 전역 변수나 공용 데이터를 함께 사용하는 결합이다. 키워드는 전역 데이터 공유이다.
데이터 결합도는 필요한 데이터만 매개변수로 주고받는 결합이다. 키워드는 인수 전달이다.
내용 결합도는 한 모듈이 다른 모듈의 내부 데이터나 내부 처리에 직접 접근하는 결합이다. 키워드는 내부 침범이다.
제어 결합도는 처리 순서를 바꾸는 제어 정보나 플래그를 전달하는 결합이다. 키워드는 제어값 전달이다.

NoteBase

선지별 해설

① X: 데이터 결합도는 모듈 사이에 필요한 데이터만 매개변수로 전달할 때이다. 틀린 표현은 '같은 전역 데이터를 접근'이다. 올바르게는 '매개변수로 필요한 데이터만 주고받는 경우 데이터 결합도'이다.
② X: 내용 결합도는 다른 모듈의 내부 로직이나 내부 데이터를 직접 참조하거나 수정할 때이다. 틀린 표현은 '전역 데이터 공유'이다. 올바르게는 '다른 모듈의 내부 구현에 직접 의존하는 경우 내용 결합도'이다.
③ O: 두 모듈 cca와 ccb가 모두 global_variable이라는 같은 전역 데이터를 사용하므로 공통 결합도이다. 공용 영역을 여러 모듈이 함께 쓰는 형태가 핵심 근거이다.
④ X: 제어 결합도는 분기나 수행 여부를 결정하는 제어값을 다른 모듈에 넘겨주는 경우이다. 틀린 표현은 '전역 데이터 공유'이다. 올바르게는 '플래그나 상태코드처럼 제어 목적의 값을 전달하는 경우 제어 결합도'이다.

오답 포인트

전역 변수와 단순 데이터 전달을 같은 것으로 보면 ①과 ③을 헷갈리기 쉽다. 데이터 결합도는 매개변수 전달이고, 공통 결합도는 여러 모듈이 같은 전역 데이터를 공유한다는 점에서 구별된다.

다시 나오면 이렇게 판단

문제 그림이나 설명에서 전역 변수, 공용 메모리, shared data가 보이면 먼저 공통 결합도를 떠올린다. 매개변수 전달이면 데이터 결합도이고, 제어용 플래그 전달이면 제어 결합도이며, 다른 모듈 내부를 직접 건드리면 내용 결합도이다.

암기 포인트

전역 변수 공유는 공통 결합도, 인수 전달은 데이터 결합도, 제어값 전달은 제어 결합도, 내부 침범은 내용 결합도이다.

NoteBase

문 13. 소프트웨어 시험 기법에서 블랙 박스 시험 기법의 특징으로 가장 옳지 않은 것은?

① 동등 분할(equivalence partitioning)

② 조건 검사(condition coverage)

③ 경곗값 분석(boundary value analysis)

④ 원인-결과 그래프(cause-effect graphing)

정답 2번

출제 포인트

이 문제는 블랙 박스 시험 기법과 화이트 박스 시험 기법의 구분을 묻는 문제이다.

핵심 개념

블랙 박스 시험은 내부 코드 구조를 보지 않고 명세와 입출력 기준으로 검증하는 기법이다.
동등 분할은 입력값을 같은 성질의 집합으로 나누어 대표값을 시험하는 기법이다.
경계값 분석은 오류가 자주 발생하는 경계 지점을 시험하는 기법이다.
원인-결과 그래프는 입력 조건과 출력 결과의 논리 관계로 시험 케이스를 도출하는 기법이다.
조건 검사는 프로그램 내부 조건식의 참·거짓 수행 여부를 보는 구조 기반 기법이므로 화이트 박스 시험에 속한다.

선지별 해설

① O: 동등 분할은 입력 도메인을 유효 클래스와 무효 클래스로 나누어 대표값을 시험하는 블랙 박스 기법이다.
② X: 틀린 부분은 "블랙 박스 시험 기법"이라는 분류이다. 조건 검사는 조건식의 참·거짓이 모두 수행되는지를 코드 구조 기준으로 확인하므로 화이트 박스 시험 기법이다. 올바르게는 "조건 검사는 화이트 박스 시험 기법이다"라고 고쳐야 한다.
③ O: 경계값 분석은 입력 범위의 최소값, 최대값, 그 인접값처럼 오류가 집중되기 쉬운 경계를 시험하는 대표적 블랙 박스 기법이다.
④ O: 원인-결과 그래프는 입력 조건의 조합과 그 결과를 논리적으로 연결해 시험 케이스를 설계하는 블랙 박스 기법이다.

오답 포인트

조건 검사처럼 coverage라는 표현이 들어가면 시험 범위를 넓게 본다는 인상 때문에 블랙 박스로 오해하기 쉽다. 그러나 핵심은 입력값이 아니라 코드 내부 조건식의 수행 여부를 본다는 점이다.

NoteBase

다시 나오면 이렇게 판단

선지가 입력값 집합, 경계, 명세, 결과 조합을 기준으로 시험 케이스를 만드는 내용이면 블랙 박스일 가능성이 높다. 반대로 조건문, 분기, 경로, 문장 실행 여부처럼 코드 내부 구조를 기준으로 하면 화이트 박스이다.

암기 포인트

동등 분할·경계값 분석·원인-결과 그래프는 블랙 박스이고, 조건·분기·경로 커버리지는 화이트 박스이다.

NoteBase

문 14. 2017년 발표된 웹 애플리케이션 보안 취약점 OWASP TOP 10에 포함되지 않은 것은?

① 크로스 사이트 스크립팅(XSS)

② 인젝션

③ 보안 설정 미흡

④ 안전한 암호화 저장

정답 4번

출제 포인트

이 문제는 OWASP Top 10 2017의 공식 취약점 항목명과 과거 표현을 구분하는 문제이다.

핵심 개념

OWASP Top 10은 웹 애플리케이션의 대표 보안 취약점을 분류한 목록이다.
2017 목록에는 인젝션, 보안 설정 미흡, 크로스 사이트 스크립팅(XSS) 등이 포함된다.
반면 '안전한 암호화 저장'은 취약점 이름이 아니며, 2013 표현을 끌어오더라도 '안전하지 않은 암호화 저장'이 관련 표현이다. 2017에서는 이 성격의 문제가 '민감 데이터 노출'로 다뤄진다.

선지별 해설

1번 O: 크로스 사이트 스크립팅(XSS)은 OWASP Top 10 2017에 포함된 항목이다. 공격자가 신뢰된 웹 페이지에 스크립트를 주입하는 취약점이므로 포함 대상이 맞다.
2번 O: 인젝션은 OWASP Top 10 2017의 대표 항목이다. SQL, NoSQL, LDAP 등 해석기에 악성 명령을 주입하는 취약점이므로 포함된다.
3번 O: 보안 설정 미흡은 OWASP Top 10 2017에 포함된다. 기본 계정 방치, 불필요한 기능 노출, 잘못된 서버 설정 같은 문제가 여기에 해당한다.
4번 X: 틀린 부분은 '안전한 암호화 저장'이라는 표현이다. 이는 취약점명이 아니라 오히려 안전한 보안 실천처럼 들리는 표현이다. 2017 포함 항목으로 고치면 '민감 데이터 노출'이 맞고, 과거 용어를 쓰려면 '안전하지 않은 암호화 저장'이어야 한다.

오답 포인트

4번은 과거 OWASP 분류를 어설프게 기억하면 속기 쉽다. '암호화 저장'이라는 단어만 보고 익숙하다고 판단하면 틀리며, '안전한'인지 '안전하지 않은'인지와 2017 공식 항목명이 무엇인지까지 함께 봐야 한다.

다시 나오면 이렇게 판단

OWASP Top 10 문제는 항목의 공식 명칭을 먼저 확인한 뒤, 현재 묻는 연도의 목록과 정확히 일치하는지 판단한다. 특히 보안 취약점 명칭인지, 보안 대책처럼 들리는 표현인지를 구분하면 함정을 피할 수 있다.

NoteBase

암기 포인트

OWASP 2017은 XSS·인젝션·보안 설정 미흡은 포함되고, '안전한 암호화 저장'은 취약점명이 아니다.

NoteBase

문 15. 발주자가 요청한 최종 결과물의 일부 기능을 사전에 점검해 볼 수 있는 소프트웨어 생명주기 모델로 가장 옳은 것은?

① 폭포수 모델(waterfall model)

② 나선형 모델(spiral model)

③ 프로토타이핑 모델(prototyping model)

④ V 모델(V model)

정답 3번

출제 포인트

이 문제는 소프트웨어 생명주기 모델 중 발주자가 요구 기능을 시제품으로 먼저 확인할 수 있는 프로토타이핑 모델의 특징을 묻는 문제이다.

핵심 개념

폭포수 모델은 요구분석-설계-구현-시험을 순차적으로 진행하는 선형 모델이다. 키워드는 순차 진행이다.
나선형 모델은 위험 분석을 중심으로 계획-위험 분석-개발-평가를 반복하는 모델이다. 키워드는 위험 관리와 반복이다.
프로토타이핑 모델은 최종 시스템을 만들기 전에 핵심 기능의 시제품을 먼저 만들어 사용자 요구를 확인하고 보완하는 모델이다. 키워드는 시제품과 요구사항 구체화이다.
V 모델은 개발 단계와 이에 대응하는 테스트 단계를 짝지어 검증과 확인을 강화한 모델이다. 키워드는 검증과 테스트 대응이다.

NoteBase

선지별 해설

① X
폭포수 모델은 각 단계를 순차적으로 완료한 뒤 다음 단계로 넘어가는 방식이므로 발주자가 일부 기능을 시제품처럼 미리 점검하는 데 적합하지 않다.
틀린 부분은 이 모델이 사전 기능 점검에 적합하다고 볼 수 있다는 점이다. 올바르게는 요구가 안정적이고 단계별 산출물을 문서 중심으로 관리하는 모델이다.

② X
나선형 모델은 반복 개발을 하기는 하지만 핵심은 위험 분석과 위험 완화이다. 발주자가 요청 기능의 일부를 시제품으로 먼저 확인하는 대표 목적의 모델로는 프로토타이핑 모델이 더 정확하다.
틀린 부분은 가장 직접적인 답으로 나선형을 고른 점이다. 올바르게는 사용자 요구 기능의 사전 확인에는 프로토타이핑 모델이 적합하다.

③ O
프로토타이핑 모델은 최종 결과물 일부 기능을 시제품 형태로 먼저 구현하여 발주자나 사용자가 사전에 점검하고 요구사항을 보완할 수 있게 한다. 문제의 설명과 가장 정확히 일치한다.

④ X
V 모델은 각 개발 단계에 대응하는 테스트를 미리 연결해 검증과 확인을 체계화한 모델이다. 사용자에게 일부 기능을 먼저 보여 주는 시제품 중심 모델은 아니다.
틀린 부분은 V 모델을 사전 기능 점검용 모델로 본 점이다. 올바르게는 V 모델은 테스트 계획과 검증 활동을 강화하는 모델이다.

오답 포인트

② 나선형 모델은 반복 개발을 한다는 점 때문에 ③과 혼동하기 쉽다. 그러나 나선형의 중심 키워드는 위험 분석이고, 프로토타이핑의 중심 키워드는 시제품을 통한 요구사항 확인이라는 점이 다르다. ④ V 모델도 초기에 테스트를 고려한다는 이유로 사전 점검과 비슷해 보이지만, 그것은 사용자 시연이 아니라 검증 절차의 대응 관계이다.

다시 나오면 이렇게 판단

문장에서 발주자, 사용자, 일부 기능, 미리 확인, 시제품 같은 표현이 보이면 먼저 프로토타이핑 모델을 떠올린다. 반대로 위험 분석이면 나선형, 단계적 순차 진행이면 폭포수, 테스트 대응 관계이면 V 모델로 판단할 수 있다.

암기 포인트

프로토타이핑은 시제품으로 요구를 확인하고, 나선형은 위험을 관리하며, V 모델은 테스트를 대응시킨다.

NoteBase

문 16. 소프트웨어 개발 생명주기 모델 중 나선형 모델(spiral model)의 단계를 순서대로 바르게 나열한 것은?

① 위험 분석 - 고객 평가 - 계획 및 정의 - 개발

② 위험 분석 - 계획 및 정의 - 개발 - 고객 평가

③ 계획 및 분석 - 위험 분석 - 고객 평가 - 개발

④ 계획 및 정의 - 위험 분석 - 개발 - 고객 평가

정답 4번

출제 포인트

이 문제는 나선형 모델의 반복 순서를 아는지를 묻는 절차·순서형 문항이다.

올바른 순서

1. 계획 및 정의 - 목표, 대안, 제약 조건을 정리하는 단계이다.
2. 위험 분석 - 대안별 위험을 분석하고 줄이는 방법을 검토하는 단계이다.
3. 개발 - 설계, 구현, 시험 등 실제 공학적 개발을 수행하는 단계이다.
4. 고객 평가 - 산출물을 고객이 검토하고 다음 반복에 반영할 의견을 주는 단계이다.

왜 이 순서인가: 나선형 모델은 먼저 계획을 세우고 그 계획의 위험을 분석한 뒤 개발을 진행하며, 그 결과를 고객이 평가해 다음 순환의 입력으로 사용한다.

선지별 해설

① X: 틀린 부분은 "위험 분석 - 고객 평가 - 계획 및 정의 - 개발"의 배열이다. 고객 평가는 개발 결과를 본 뒤 이루어지는 단계이므로 초반에 올 수 없다. 올바르게는 "계획 및 정의 - 위험 분석 - 개발 - 고객 평가"이다.

② X: 틀린 부분은 "위험 분석"을 "계획 및 정의"보다 앞에 둔 점이다. 위험 분석은 목표와 대안이 정리된 뒤에 가능하므로 계획 수립이 먼저이다. 올바르게는 "계획 및 정의 - 위험 분석 - 개발 - 고객 평가"이다.

③ X: 틀린 부분은 "고객 평가 - 개발"의 순서이다. 고객 평가는 개발 산출물을 검토하는 단계이므로 개발보다 앞설 수 없다. 올바르게는 "계획 및 정의 - 위험 분석 - 개발 - 고객 평가"이다.

④ O: "계획 및 정의 - 위험 분석 - 개발 - 고객 평가"는 보엠의 나선형 모델 한 주기의 전형적인 순서와 일치한다.

오답 포인트

폭포수 모델의 선형 절차와 헷갈리면 계획과 위험 분석의 선후가 흔들리고, 고객 평가를 요구사항 단계의 확인처럼 착각하면 개발 이전에 배치하는 오류가 발생한다. 이 문항의 핵심 함정은 고객 평가가 매 반복의 마지막 피드백 단계라는 점이다.

NoteBase

다시 나오면 이렇게 판단

나선형 모델 문제는 "계획 수립 -> 위험 분석 -> 공학적 개발 -> 고객 평가"의 반복 고리를 먼저 떠올리면 된다.
특히 고객 평가는 개발 결과를 본 뒤 다음 회전에 반영하는 단계인지로 판단하면 순서를 쉽게 고를 수 있다.

암기 포인트

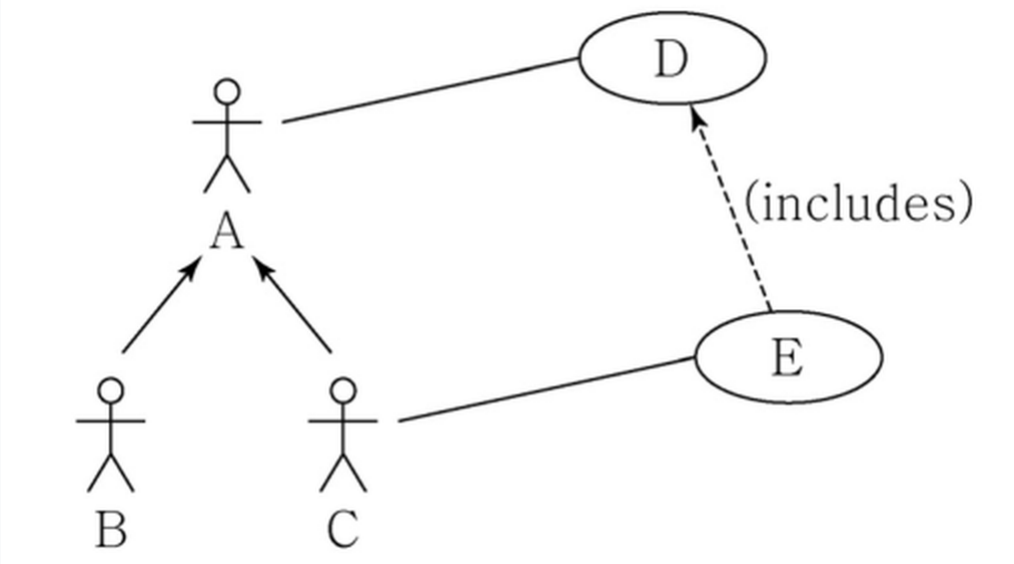
나선형 모델은 계획 후 위험을 보고 개발한 뒤 고객이 평가하는 순환 구조이다.

NoteBase

문 17. <보기>의 설명을 읽고 UML 유스케이스 다이어그램을 작성 할 때, A~E에 각각 들어갈 용어로 가장 옳은 것은?

<보기>
설명: 회원이 아닌 고객(비회원)은 물품 조회만을 할 수 있다. 반면에 회원인 고객은 물품 구매도 가능하다.
그러나 물품 구매를 할 때는 반드시 물품 조회를 실행하여야 한다.

유스케이스 다이어그램:



① A: 고객, B: 비회원, C: 회원, D: 물품 조회, E: 물품 구매

② A: 고객, B: 비회원, C: 회원, D: 물품 구매, E: 물품 조회

③ A: 비회원, B: 회원, C: 고객, D: 물품 조회, E: 물품 구매

④ A: 비회원, B: 회원, C: 고객, D: 물품 구매, E: 물품 조회

정답 1번

출제 포인트

이 문제는 UML 유스케이스 다이어그램에서 액터 일반화 방향과 include 관계 방향을 읽어 A~E의 역할을 정확히 대응시키는 문제이다.

NoteBase

선지별 해설

1번: O. B와 C에서 A로 향하는 일반화 화살표는 A가 상위 액터임을 뜻하므로 A는 고객, B는 비회원, C는 회원이다. 또 E에서 D로 향하는 include 관계는 물품 구매가 물품 조회를 반드시 포함함을 뜻하므로 D는 물품 조회, E는 물품 구매가 맞다.
2번: X. D와 E가 뒤바뀌었다. include 화살표는 포함하는 유스케이스에서 포함되는 유스케이스로 향하므로 올바르게는 D는 물품 조회, E는 물품 구매이다.
3번: X. 액터 일반화 계층이 뒤바뀌었다. 화살표가 B와 C에서 A로 향하므로 A가 고객이고 B와 C가 그 하위인 비회원과 회원이다. 올바르게는 A는 고객, B는 비회원, C는 회원이다.
4번: X. 액터 계층과 유스케이스 대응이 모두 틀렸다. 올바르게는 A는 고객, B는 비회원, C는 회원이며, D는 물품 조회, E는 물품 구매이다.

오답 포인트

일반화 화살표와 include 화살표의 방향을 반대로 읽으면 ③, ④처럼 액터 상하관계가 뒤집히고 ②, ④처럼 조회와 구매가 바뀐다. 특히 include는 화살표가 가리키는 쪽이 포함되는 필수 유스케이스라는 점을 자주 혼동한다.

다시 나오면 이렇게 판단

먼저 일반화 화살표가 어느 액터를 향하는지 보고 상위 액터를 찾는다. 다음으로 include 화살표가 가리키는 유스케이스를 확인해 반드시 선행되거나 공통으로 포함되는 기능을 배정하면 된다.

암기 포인트

일반화 화살표는 상위 액터로 향하고, include 화살표는 포함되는 필수 유스케이스로 향한다.

NoteBase

문 18. PMBOK 프로젝트 관리 지침에서 제시하는 프로젝트 관리 관점 중에서 프로젝트의 요구사항을 만족하여 수행 목표를 달성하기 위한 활동을 다루는 것은?

① 일정 관리

② 품질 관리

③ 비용 관리

④ 위험 관리

정답 2번

출제 포인트

이 문제는 PMBOK의 프로젝트 관리 지식영역 중 품질 관리의 정의를 정확히 구별할 수 있는지를 묻는다.

핵심 개념

일정 관리는 정해진 기간 안에 프로젝트를 완료하도록 활동의 순서, 기간, 일정을 계획하고 통제하는 영역이다.
품질 관리는 프로젝트의 요구사항을 만족하여 수행 목적을 달성하도록 품질 정책, 품질보증, 품질통제 활동을 다루는 영역이다.
비용 관리는 승인된 예산 범위 안에서 프로젝트를 완료하도록 비용을 산정, 예산편성, 통제하는 영역이다.
위험 관리는 프로젝트 위험을 식별, 분석, 대응, 감시하여 불확실성의 영향을 관리하는 영역이다.

선지별 해설

1번 X: 일정 관리는 완료 시점과 작업 순서를 다루는 영역이지, 요구사항 만족을 위한 품질 활동을 직접 뜻하지 않는다. 틀린 표현은 '요구사항을 만족하여 수행 목표를 달성하기 위한 활동'에 일정 관리를 대응시킨 점이다. 올바르게는 일정 관리는 프로젝트를 제때 끝내기 위한 일정 계획과 통제이다.
2번 O: 품질 관리는 프로젝트가 착수된 목적과 요구사항을 만족하도록 수행되는 활동을 다루므로 발문의 설명과 일치한다.
3번 X: 비용 관리는 예산 수립과 비용 통제가 핵심이다. 틀린 표현은 품질 달성 활동을 비용 관리로 본 점이다. 올바르게는 비용 관리는 승인된 예산 내에서 프로젝트를 수행하도록 비용을 관리하는 영역이다.
4번 X: 위험 관리는 불확실한 사건의 발생 가능성과 영향을 관리하는 영역이다. 틀린 표현은 요구사항 만족을 위한 품질 활동을 위험 관리로 바꾼 점이다. 올바르게는 위험 관리는 위험 식별, 분석, 대응을 다룬다.

오답 포인트

PMBOK 지식영역은 모두 '프로젝트 목표 달성'과 연결되므로 공통 표현만 보고 고르면 헷갈리기 쉽다. 이 문항은 '요구사항을 만족'이라는 품질 관리의 핵심 문구를 잡아내는지가 함정이다.

NoteBase

다시 나오면 이렇게 판단

발문에서 기간, 예산, 위험, 요구사항 만족 중 무엇을 중심으로 설명하는지 먼저 본다. '요구사항 만족'이나 '품질 정책·보증·통제'가 나오면 품질 관리로 판단한다.

암기 포인트

일정은 언제 끝내는가, 비용은 얼마를 쓰는가, 위험은 무엇이 흔드는가, 품질은 요구사항을 만족하는가이다.

NoteBase

문 19. 다음은 '항공권 예약 시스템'의 요구 사항을 나열한 것이다. 기능적 요구 사항에 대한 설명으로 가장 옳지 않은 것은?

① 항공편, 탑승객, 예약을 입력하는 방법을 결정해야 한다.

② 티켓과 리포트에 어떤 정보를 표시할지 결정해야 한다.

③ 여행사와 고객이 데이터베이스에 접근할 때 어떤 정보를 얻을 수 있는지 결정해야 한다.

④ 자주 탑승하는 고객을 서비스하기 위해 시스템을 확장할 수 있도록 설계해야 한다.

정답 4번

출제 포인트

이 문제는 기능 요구 사항과 비기능 요구 사항의 차이를 묻는 문제이다.

핵심 개념

기능 요구 사항: 시스템이 무엇을 입력받고, 어떻게 처리하며, 어떤 결과를 제공해야 하는지를 정한 요구 사항이다. 결정 차이축: 시스템이 수행해야 할 서비스 자체인지 여부이다. 키워드: 기능, 입력, 출력, 처리, 조회
비기능 요구 사항: 시스템이 어느 수준의 품질과 제약을 만족해야 하는지를 정한 요구 사항이다. 결정 차이축: 기능 자체가 아니라 성능·보안·확장성·신뢰성 같은 품질 조건인지 여부이다. 키워드: 성능, 보안, 확장성, 유지보수성

NoteBase

선지별 해설

① O
항공권, 탑승객, 예약을 어떤 방식으로 입력할지는 시스템의 입력 기능을 정하는 내용이므로 기능 요구 사항에 해당한다.

② O
티켓과 리포트에 어떤 정보를 표시할지는 시스템의 출력 기능을 정하는 내용이므로 기능 요구 사항에 해당한다.

③ O
여행사와 고객이 데이터베이스 접근 시 어떤 정보를 얻을 수 있는지는 사용자별 조회 기능과 제공 정보 범위를 정하는 내용이므로 기능 요구 사항에 해당한다.

④ X
틀린 부분은 "시스템을 확장할 수 있도록 설계해야 한다"이다.
이 표현은 자주 탑승하는 고객을 위한 구체적 기능이 아니라 확장성이라는 품질 속성을 말하므로 비기능 요구 사항이다.
올바르게는 "자주 탑승하는 고객을 위한 우대 서비스나 마일리지 관련 기능을 제공해야 한다"처럼 시스템이 수행할 기능을 서술해야 기능 요구 사항이 된다.

오답 포인트

④는 "고객 서비스"라는 표현 때문에 기능 요구처럼 보이지만, 실제로 판단 기준은 서비스 대상이 아니라 문장의 핵심이 무엇인지이다. 핵심이 "확장할 수 있도록 설계"에 있으므로 기능이 아니라 확장성 요구이다.

다시 나오면 이렇게 판단

문장이 시스템이 무엇을 해 주는지를 말하면 기능 요구 사항이고, 얼마나 잘 동작해야 하는지나 얼마나 쉽게 확장·유지보수되어야 하는지를 말하면 비기능 요구 사항이다.

암기 포인트

기능 요구는 "무엇을 한다", 비기능 요구는 "어떻게 좋아야 한다"로 구분된다.

NoteBase

문 20. 클래스 다이어그램의 특징으로 가장 옳지 않은 것은?

① 클래스들의 인터페이스를 나타낸다.

② 클래스는 객체의 집합이 가지는 속성과 동작을 추상적으로 나타낸다.

③ 시스템 동적 구조를 모델링한다.

④ 속성값의 타입을 정하는 것은 시스템 정의에서 중요한 일이 아니므로 객체 설계 단계까지 미뤄도 된다.

정답 3번

출제 포인트

이 문제는 UML 클래스 다이어그램이 표현하는 대상이 정적 구조인지 동적 구조인지를 구별하는 문제이다.

핵심 개념

클래스 다이어그램은 클래스, 속성, 연산, 클래스 사이의 관계를 통해 시스템의 정적 구조를 표현한다.
동적 구조는 시간에 따른 행위와 상호작용을 뜻하며 주로 시퀀스 다이어그램, 상태 다이어그램, 활동 다이어그램 등이 담당한다.
클래스는 객체들의 공통 속성과 동작을 추상화한 개념이다.

선지별 해설

1번 O: 클래스 다이어그램은 클래스의 속성, 연산과 함께 외부에 드러나는 인터페이스를 표현할 수 있으므로 맞는 설명이다.
2번 O: 클래스는 동일한 성질을 가진 객체들의 공통 속성과 동작을 추상화한 것이므로 맞는 설명이다.
3번 X: 틀린 부분은 "동적 구조"이다. 클래스 다이어그램이 모델링하는 것은 시스템의 동적 구조가 아니라 정적 구조이다.
4번 O: 속성의 구체적인 타입 결정은 상세 설계에서 더 정밀하게 다룰 수 있으므로 초기 시스템 정의 단계에서 반드시 모두 확정해야 하는 것은 아니다.

오답 포인트

클래스 다이어그램에 메서드가 보인다는 이유만으로 동작 중심 다이어그램이라고 오해하기 쉽다. 이 문항의 함정은 "동작"이라는 단어와 "동적 구조"를 같은 뜻처럼 섞어 놓은 점이다.

다시 나오면 이렇게 판단

다이어그램 문제가 나오면 먼저 시간 흐름과 메시지 교환을 표현하는지, 아니면 클래스와 관계를 고정된 구조로 보여 주는지를 구분한다. 클래스, 속성, 연관, 일반화가 핵심이면 정적 구조로 판단한다.

NoteBase

암기 포인트

클래스 다이어그램은 정적 구조, 시퀀스·상태·활동 다이어그램은 동적 행위 표현이다.

NoteBase