userproblemspdf이거풀어줘 답안집

생성일: 2025-08-26
총 문항 수: 29개

문제 1. 소프트웨어 공학에서 모델링(Modeling)과 관련한 설명으로 틀린 것은?
1) 개발팀이 응용 문제를 이해하는 데 도움을 줄 수 있다.
2) 유지보수 단계에서만 모델링 기법을 활용한다.
3) 개발된 시스템에 대하여 여러 분야의 엔지니어들이 공동된 내용을 공유하는 데 도움을 준다.
4) 절차적인 프로그램을 위한 자료 흐름도는 프로세스 위주의 모델링 방법이다.

정답: 2
풀이: 소프트웨어 공학에서 모델링은 시스템 개발의 여러 단계에서 사용됩니다. 모델링 기법은 요구사항 분석, 설계, 구현, 테스트, 유지보수 단계 등 다양한 단계에서 활용되며, 이는 개발팀이 응용 문제를 이해하고, 여러 분야의 엔지니어들이 공동의 내용을 공유하는 데 도움을 줍니다. 따라서 "유지보수 단계에서만 모델링 기법을 활용한다."라는 설명은 틀린 것입니다. 모델링은 시스템의 초기 단계부터 유지보수 단계까지 지속적으로 중요한 역할을 하며, 절차적인 프로그램을 위한 자료 흐름도는 프로세스 중심의 모델링 방법으로 분류됩니다. 이러한 이유로 2번 선택지가 틀린 설명으로 판단됩니다. 과목: 소프트웨어설계

문제 2. UML 모델에서 한 객체가 다른 객체에게 오퍼레이션을 수행하도록 지정하는 의미적 관계로 옳은 것은?
1) Dependency
2) Realization
3) Generalization
4) Association

정답: 1
풀이: UML(Unified Modeling Language)에서 객체 간의 관계는 여러 가지로 나눌 수 있습니다. 주어진 보기 중에서 'Dependency'는 한 객체가 다른 객체에 대해 의존성을 가지고 있으며, 이는 객체가 다른 객체의 오퍼레이션을 수행할 수 있음을 나타냅니다. 즉, A 객체가 B 객체의 기능을 사용한다면, A는 B에 의존하고 있다고 할 수 있습니다. 반면, 'Realization'은 인터페이스와 그 구현 간의 관계를 나타내며, 'Generalization'은 상속 관계를 의미하고, 'Association'은 객체 간의 단순한 연결 관계를 나타냅니다. 따라서 한 객체가 다른 객체에게 오퍼레이션을 수행하도록 지정하는 의미적 관계는 'Dependency'입니다. 과목: 소프트웨어설계

문제 3. 분산 시스템을 위한 마스터-슬레이브(Master-Slave) 아키텍처에 대한 설명으로 틀린 것은?
1) 일반적으로 실시간 시스템에서 사용된다.
2) 마스터 프로세스는 일반적으로 연산, 통신, 조정을 책임진다.
3) 슬레이브 프로세스는 데이터 유지 기능을 수행할 수 없다.
4) 마스터 프로세스는 슬레이브 프로세스들을 제어할 수 있다.

정답: 3
풀이: 분산 시스템에서 마스터-슬레이브 아키텍처는 일반적으로 마스터 프로세스가 슬레이브 프로세스를 제어하고 조정하는 구조입니다. 마스터 프로세스는 주로 연산, 통신, 조정 등의 역할을 수행하며, 슬레이브 프로세스는 데이터를 처리하거나 작업을 수행하는 역할을 맡습니다. 하지만 '슬레이브 프로세스는 데이터 유지 기능을 수행할 수 없다.'라는 설명은 틀립니다. 슬레이브 프로세스는 일반적으로 데이터 유지 및 처리 기능을 수행하며, 마스터 프로세스의 지시에 따라 작업을 수행합니다. 따라서 '슬레이브 프로세스는 데이터 유지 기능을 수행할 수 없다.'라는 설명이 틀린 것입니다. 과목: 소프트웨어설계

문제 4. 객체지향 분석 방법론 중 Coad-Yourdon 방법에 해당하는 것은?
1) E-R 다이어그램을 사용하여 객체의 행위를 모델링한다.

2) 문제 도메인을 기능별로 나누어 수행하는 방법이다.
3) 객체, 구조, 주제, 속성, 서비스 개념을 모두 사용하는 방법이다.
4) Use-Case를 강조하여 사용하는 방법이다.

정답: 3
풀이: Coad-Yourdon 방법론은 객체지향 분석 방법론 중 하나로, 객체, 구조, 주제, 속성, 서비스 개념을 모두 사용하는 방법입니다. 이 방법론은 객체를 정의하고 그 객체들이 어떻게 상호작용하는지를 모델링하는 데 중점을 둡니다. 주어진 보기 중에서 '객체, 구조, 주제, 속성, 서비스 개념을 모두 사용하는 방법이다.'라는 설명이 Coad-Yourdon 방법론의 핵심을 잘 설명하고 있습니다. 다른 보기들은 각각 다른 분석 방법론 또는 개념을 설명하고 있어 Coad-Yourdon 방법론과는 관련이 없습니다. 따라서 정답은 3번입니다. 과목: 소프트웨어설계

문제 5. 소프트웨어 설계에서 객체의 행위를 분석하는 다이어그램은?
1) 클래스 다이어그램
2) 유스케이스 다이어그램
3) 시퀀스 다이어그램
4) 상태 다이어그램

정답: 3
풀이: 소프트웨어 설계에서 객체의 행위를 분석하는 다이어그램은 주로 '시퀀스 다이어그램'이나 '상태 다이어그램'과 같은 다이어그램들이 사용됩니다. 주어진 보기 중에서 '시퀀스 다이어그램'은 객체 간의 상호작용을 시간 순서에 따라 나타내어 객체의 행위와 메시지 흐름을 분석하는 데 적합합니다. 반면, '클래스 다이어그램'은 객체의 정적 구조를 나타내고, '유스케이스 다이어그램'은 시스템의 기능적 요구사항을 표현하며, '상태 다이어그램'은 객체의 상태 변화를 모델링합니다. 그러나 객체의 행위 분석에 가장 직접적으로 관련된 것은 '시퀀스 다이어그램'이므로 정답은 3번입니다. 과목: 소프트웨어설계

문제 6. 다음 중 소프트웨어 설계의 주요 항목이 아닌 것은?
1) 화면 설계
2) 애플리케이션 설계
3) 데이터베이스 설계
4) 네트워크 설계

정답: 4
풀이: 소프트웨어 설계의 주요 항목으로는 화면 설계, 애플리케이션 설계, 데이터베이스 설계가 포함됩니다. 각 항목은 소프트웨어의 기능과 사용자 경험을 직접적으로 개선하는 데 중요한 역할을 합니다. 그러나 네트워크 설계는 소프트웨어 설계의 주요 항목으로 간주되지 않습니다. 네트워크 설계는 시스템 간의 통신 및 데이터 전송 방식에 중점을 두며, 소프트웨어의 내부 구조나 사용자 인터페이스와는 직접적인 관련이 없습니다. 따라서 "네트워크 설계"는 소프트웨어 설계의 주요 항목이 아닌 것으로 판단됩니다. 과목: 소프트웨어설계

문제 7. 소프트웨어 아키텍처의 유형 중 ' 파 이프 필 터 아키텍처'에 대한 설명으로 옳은 것은?
1) 데이터는 파 이프를 통해 단방향으로 흐른다.
2) 데이터는 파 이프를 통해 양 방향으로 흐른다.
3) 필 터는 데이터를 변환 하 고 처리한다.
4) 파 이프는 데이터를 저장 하는 역 할을 한다.

정답: 1
풀이: '파이프 필터 아키텍처'는 소프트웨어 아키텍처의 한 유형으로, 데이터가 파이프를 통해 단방향으로 흐르는 구조입니다. 이 아키텍처의 핵심 개념은 각 필터가 데이터를 입력받아 처리한 후, 그 결과를 다음 필터로 전달하는 것입니다. 따라서 데이터 흐름은 항상 한 방향으로만 이루어지며, 각 필터는 독립적으로 동작하여 데이터를 변환하고 처리합니다. 주어진 보기 중에서 '데이터는 파이프를 통해 단방향으로 흐른다.'라는 설명이 정확하게 이 아키텍처의 특징을 설명하고 있으므로 정답은 1번입니다. 나머지 보기인 '데이터는 파이프를 통해 양 방향으로 흐른다.'는 사실과 다르며, '필터는 데이터를 변환하고 처리한다.'는 설명은 맞지만 주어진 질문의 초점이 아니므로 정답이 아닙니다. '파이프는 데이터를 저장하는 역할을 한다.'는 잘못된 설명입니다. 따라서 1번이 정답입니다. 과목: 소프트웨어설계

문제 8. 디 자인 패턴 의 종류 중 생 성 패턴 에 해당하는 것은?

1) Adapter 패턴
2) Bridge 패턴
3) Builder 패턴
4) Proxy 패턴

정답: 3
풀이: 생성 패턴은 객체의 생성과 관련된 디자인 패턴으로, 객체를 생성하는 방법을 정의합니다. 주어진 보기 중에서 'Builder 패턴'은 객체의 생성 과정을 단계별로 나누어 복잡한 객체를 쉽게 생성할 수 있게 해주는 생성 패턴입니다. 'Adapter 패턴', 'Bridge 패턴', 'Proxy 패턴'은 각각 구조 패턴이나 행동 패턴으로 분류됩니다. 따라서 생성 패턴에 해당하는 것은 'Builder 패턴'이므로 정답은 3번입니다. 과목: 소프트웨어설계

문제 9. 소프트웨어 설계에서 결합 도에 대한 설명으로 옳은 것은?
1) 결합 도가 높 을수록 소프트웨어의 유지보수가 쉽 다.
2) 결합 도가 낮 을수록 소프트웨어의 재 사용성이 높 다.
3) 결합 도는 모 듈 간의 의 존 성을 나 타낸 다.
4) 결합 도는 모 듈 의 응집 력 을 나 타낸 다.

정답: 2
풀이: 결합도(Coupling)는 모듈 간의 의존성을 나타내는 지표로, 결합도가 높을수록 모듈 간의 의존성이 강해져서 유지보수가 어려워집니다. 반면, 결합도가 낮을수록 모듈 간의 독립성이 높아져서 소프트웨어의 유지보수와 재사용성이 용이해집니다. 따라서 주어진 보기 중에서 '결합도가 낮을수록 소프트웨어의 재사용성이 높다.'는 설명이 옳은 것입니다. 나머지 보기들은 결합도에 대한 잘못된 설명이므로 정답은 2번입니다. 과목: 소프트웨어설계

문제 10. 객체지향 설계에서 다이어그램을 사용하여 객체의 행위를 표현 하는 것은?
1) 클래스 다이어그램
2) 유스케이스 다이어그램
3) 시퀀스 다이어그램
4) 상태 다이어그램

정답: 3
풀이: 객체지향 설계에서 객체의 행위를 표현하는 다이어그램은 주로 '시퀀스 다이어그램'과 '상태 다이어그램'이 사용됩니다. 이 중에서 '시퀀스 다이어그램'은 객체 간의 상호작용을 시간 순서에 따라 나타내어, 객체의 행위와 메시지 흐름을 분석하는 데 적합합니다. '클래스 다이어그램'은 객체의 정적 구조를 나타내고, '유스케이스 다이어그램'은 시스템의 기능적 요구사항을 표현합니다. '상태 다이어그램'은 객체의 상태 변화를 모델링하지만, 객체의 행위를 구체적으로 설명하는 데는 '시퀀스 다이어그램'이 더 직접적입니다. 따라서 주어진 보기 중에서 객체의 행위를 표현하는 데 가장 적합한 것은 '시퀀스 다이어그램'이므로 정답은 3번입니다. 과목: 소프트웨어설계

문제 11. 소프트웨어 설계에서 모듈의 응집도에 대한 설명으로 옳은 것은?
1) 응집도가 높을수록 모듈의 재사용성이 낮다.
2) 응집도가 낮을수록 모듈의 재사용성이 높다.
3) 응집도는 모듈의 내부 응집력을 나타낸다.
4) 응집도는 모듈 간의 결합력을 나타낸다.

정답: 3
풀이: 응집도(Cohesion)는 모듈 내의 요소들이 얼마나 밀접하게 관련되어 있는지를 나타내는 지표입니다. 응집도가 높다는 것은 모듈의 모든 구성 요소가 서로 강하게 연결되어 있으며, 특정한 기능이나 목적을 수행하기 위해 함께 작동한다는 것을 의미합니다. 따라서 응집도가 높을수록 모듈의 재사용성이 높아지는 경향이 있습니다. 반면, 응집도가 낮으면 모듈의 구성 요소들이 서로 관련성이 적어지므로, 모듈이 특정한 기능을 수행하기 어렵고 재사용성이 낮아질 수 있습니다. 주어진 보기 중에서 '응집도는 모듈의 내부 응집력을 나타낸다.'라는 설명이 가장 정확하게 응집도를 설명하고 있으므로 정답은 3번입니다. 나머지 보기들은 응집도와 관련하여 잘못된 설명입니다. 과목: 소프트웨어설계

문제 12. 다음 중 소프트웨어 설계의 검증 방법이 아닌 것은?
1) 테 스트
2) 리 뷰
3) 워크스 루
4) 컴파 일

정답: 4
풀이: 소프트웨어 설계의 검증 방법으로는 일반적으로 '테스트', '리뷰', '워크숍' 등이 포함됩니다. 이들 방법은 소프트웨어의 설계가 요구사항을 충족하는지, 결함이 없는지를 확인하기 위해 활용됩니다. 반면 '컴파일'은 소프트웨어를 실행 가능한 형태로 변환하는 과정으로, 검증 방법이 아닙니다. 따라서 '컴파일'은 소프트웨어 설계의 검증 방법이 아닌 것으로 판단됩니다. 과목: 소프트웨어설계

문제 13. 소프트웨어 설계에서 인터 페 이스 설계의 중요성이 아닌 것은?
1) 시스템 간의 호환 성
2) 사용자의 편 의성
3) 소프트웨어의 유지보수
4) 하 드 웨어의 성능

정답: 4
풀이: 인터페이스 설계는 소프트웨어 설계에서 중요한 요소로, 사용자와 시스템 간의 상호작용을 정의하고 사용자의 편의성을 높이는 역할을 합니다. 또한, 시스템 간의 호환성을 고려해야 하며, 소프트웨어의 유지보수에도 영향을 미칩니다. 그러나 '하드웨어의 성능'은 인터페이스 설계와 직접적인 관련이 없습니다. 하드웨어 성능은 시스템의 전반적인 성능에 영향을 미치지만, 소프트웨어의 인터페이스 설계와는 별개의 문제입니다. 따라서 '하드웨어의 성능'은 인터페이스 설계의 중요성이 아닌 것으로 판단됩니다. 과목: 소프트웨어설계

문제 14. 소프트웨어 아키텍처의 유형 중 '데이터 중심 아키텍처'에 대한 설명으로 옳은 것은?
1) 데이터는 중앙 집중식으로 관리된다.
2) 데이터는 분산되어 관리된다.
3) 데이터는 애플리케이션과 독립적이다.
4) 데이터는 시스템의 일부이다.

정답: 1
풀이: '데이터 중심 아키텍처'는 데이터가 중앙 집중식으로 관리되는 구조를 의미합니다. 이 아키텍처에서는 모든 데이터가 중앙 데이터베이스에 저장되고, 여러 애플리케이션이 이 중앙 데이터베이스에 접근하여 데이터를 읽거나 수정하는 방식으로 동작합니다. 따라서 '데이터는 중앙 집중식으로 관리된다.'라는 설명이 정확하게 이 아키텍처의 특징을 설명하고 있으므로 정답은 1번입니다. 나머지 보기인 '데이터는 분산되어 관리된다.'는 데이터 중심 아키텍처의 정의와 상반되는 설명이며, '데이터는 애플리케이션과 독립적이다.'는 애플리케이션이 데이터에 의존하는 구조에서 맞지 않으며, '데이터는 시스템의 일부이다.'는 일반적인 설명으로 구체적인 아키텍처의 특성을 설명하지 않기 때문에 정답이 아닙니다. 따라서 1번이 정답입니다. 과목: 소프트웨어설계

문제 15. 소프트웨어 패 키 징 의 정의로 가 장 적절한 것은?
1) 모 듈 별로 생 성한 실행 파 일을 묶 어 배포 용 설 치파 일을 만 드 는 것
2) 소프트웨어 개발 과정의 첫 단계
3) 사용자 요구사항을 분석하는 과정
4) 소프트웨어 테 스트를 수행하는 과정

정답: 1
풀이: 소프트웨어 패키징의 정의는 소프트웨어를 모듈 별로 생성한 실행 파일을 묶어 배포용 설치 파일을 만드는 과정입니다. 소프트웨어 패키징은 소프트웨어를 사용자에게 제공하기 위해 필요한 모든 구성 요소를 포함하는 설치 파일을 만드는 것을 의미합니다. 이는 소프트웨어의 설치와 배포를 용이하게 하기 위해 필수적인 과정입니다. 나머지 보기인 '소프트웨어 개발 과정의 첫 단계', '사용자 요구사항을 분석하는 과정', '소프트웨어 테스트를 수행하는 과정'은 소프트웨어 개발의 다른 단계나 활동을 설명하는 것이지 패키징의 정의가 아닙니다. 따라서 정답은 1번입니다. 과목: 소프트웨어설계

문제 16. 소프트웨어 품질 목 표 중 하나 이상의 하 드 웨어 환경 에서 운 용 되 기 위해 쉽 게 수정 될 수 있는 시스템 능 력 을 의미하는 것은?
1) Correctness
2) Portability
3) Efficiency
4) Usability

정답: 2

풀이: 질문에서 "하드웨어 환경에서 운용되기 위해 쉽게 수정될 수 있는 시스템 능력"을 묻고 있습니다. 이 설명은 'Portability'에 해당합니다. Portability는 소프트웨어가 다양한 하드웨어 환경에서 쉽게 이동하거나 수정될 수 있는 능력을 의미합니다. 다른 보기인 'Correctness'(정확성)는 소프트웨어가 요구사항을 충족하는 정도를 나타내고, 'Efficiency'(효율성)는 자원 사용의 최적화를 의미하며, 'Usability'(사용성)는 사용자가 소프트웨어를 얼마나 쉽게 사용할 수 있는지를 나타냅니다. 따라서 이 문제에서 요구하는 능력은 소프트웨어가 다양한 환경에서 쉽게 수정될 수 있는 'Portability'가 정답입니다. 과목: 소프트웨어설계

문제 17. 단위 테 스트(Unit Test)의 주목적인 것은?
1) 통 합 테 스트
2) 시스템 테 스트
3) 인수 테 스트
4) 개별 모 듈 이 정 확 하게 구 현되었 는지 확 인하는 것

정답: 4
풀이: 단위 테스트(Unit Test)는 소프트웨어 개발 과정에서 개별 모듈이나 컴포넌트를 독립적으로 검증하는 테스트입니다. 그 주목적인 것은 각 모듈이 정확하게 구현되었는지를 확인하는 것입니다. 이는 소프트웨어의 품질을 높이고, 버그를 조기에 발견하여 수정할 수 있는 기회를 제공합니다. 따라서 주어진 보기 중에서 '개별 모듈이 정확하게 구현되었는지 확인하는 것'이 단위 테스트의 주목적인 것이므로 정답은 4번입니다. 나머지 보기는 통합 테스트, 시스템 테스트, 인수 테스트와 관련된 내용으로, 단위 테스트의 목적과는 다릅니다. 과목: 소프트웨어설계

문제 18. DRM(Digital Rights Management)의 구성 요소에 포함되지 않는 것은?
1) 클리어링 하우스
2) 콘텐츠 제공자
3) 분배자(유통)
4) 소비자

정답: 1
풀이: DRM(Digital Rights Management)의 구성 요소는 콘텐츠 보호 및 관리와 관련된 여러 요소로 이루어져 있습니다. 일반적으로 DRM의 구성 요소에는 콘텐츠 제공자, 분배자(유통), 소비자가 포함됩니다. 이들 각각은 콘텐츠의 생성, 배포 및 소비 과정에서 중요한 역할을 합니다. 그러나 '클리어링 하우스'는 DRM의 구성 요소로 일반적으로 간주되지 않습니다. 클리어링 하우스는 주로 거래의 중개 및 조정 역할을 하는 기관으로, DRM의 핵심 구성 요소라고 보기는 어렵습니다. 따라서 '클리어링 하우스'는 DRM의 구성 요소에 포함되지 않는 것으로 판단됩니다. 과목: 정보시스템구축관리

문제 19. 소프트웨어 생 명주기 모형 중 폭포 수 모형의 장점 이 아닌 것은?
1) 모형의 적용 경험 과 성공 사 례 가 많 음
2) 단계별 정의가 분명하 고 , 전 체 공조의 이해가 용이
3) 단계별 산 출물 이 정 확 하여 개발 공정의 기준 점 을 잘 제시
4) 개발 과정 중에 발 생 하는 새 로 운 요구나 경험 을 설계에 반 영 하기 쉬 움

정답: 4
풀이: 폭포수 모형은 소프트웨어 개발 생명주기에서 각 단계를 순차적으로 진행하는 전통적인 방법론입니다. 이 모형의 장점으로는 모형의 적용 경험과 성공 사례가 많고, 각 단계별 정의가 분명하여 전체 공정의 이해가 용이하며, 단계별 산출물이 정확하여 개발 공정의 기준점을 잘 제시하는 것 등이 있습니다. 그러나 '개발 과정 중에 발생하는 새로운 요구나 경험을 설계에 반영하기 쉬움'은 폭포수 모형의 장점이 아닙니다. 폭포수 모형은 각 단계가 완료되어야 다음 단계로 넘어가는 구조이기 때문에, 개발 과정 중에 새로운 요구사항이 발생하는 경우 이를 반영하기가 어렵습니다. 따라서 이 항목은 폭포수 모형의 장점이 아닌 것으로 판단됩니다. 과목: 소프트웨어설계

문제 20. 소프트웨어 재 공학의 이 점 으로 틀린 것은?
1) 위 험 부담 감 소
2) 개발 시간 단 축
3) 개발 비용 절 감
4) 시스템 명세의 오 류 증 가

정답: 4
풀이: 소프트웨어 재공학의 주요 목표 중 하나는 시스템의 품질을 높이고, 개발 과정에서 발생할 수 있는 위험을 줄이는 것입니다. "위험 부담 감소", "개발 시간 단축", "개발 비용 절감"은 소프트웨어 재공학의 긍정적인 효과로 볼 수 있습니다. 그러나 "시스템 명세의 오류 증가"는 소프트웨어 재공학의 목표와 정면으로 반대되는 결과입니다. 일반적으로 소프트웨어 재공학은 시스템 명세의 품질을 높이고 오류를 줄이기 위한 활동을 포함하므로, 이 선택지는 틀린 설명으로

판단됩니다. 과목: 소프트웨어설계

문제 21. 소프트웨어 재 사용 방법 중 합 성 중 심 방법에 해당하는 것은?
1) 전 자 칩 과 같 은 소프트웨어 부품 , 즉 블 록(모 듈)을 만들어서 끼 워 맞추 어 소프트웨어를 완 성시키는 방법
2) 추 상화 형태로 쓰 여진 명세를 구체화하여 프로그램을 만 드 는 방법
3) 소프트웨어 개발 과정을 테 스트하는 방법
4) 소프트웨어 결 과를 테 스트하는 방법

정답: 1
풀이: 질문에서 묻고 있는 "합성 중심 방법"은 소프트웨어 재사용 방법 중 하나로, 전자 칩과 같은 소프트웨어 부품, 즉 블록(모듈)을 만들어서 끼워 맞추어 소프트웨어를 완성시키는 방법을 의미합니다. 이 방법은 소프트웨어의 모듈화를 통해 재사용성을 높이고, 각 모듈을 조합하여 새로운 소프트웨어를 만드는 방식입니다. 나머지 보기들은 각각 다른 소프트웨어 개발 방법이나 과정에 해당하며, 합성 중심 방법의 정의와는 일치하지 않습니다. 따라서 정답은 1번입니다. 과목: 소프트웨어설계

문제 22. COCOMO 모형이 사용 되는 목적은?
1) 소프트웨어 개발 비용 산정
2) 소프트웨어 개발 시간 산정
3) 소프트웨어 개발 자 원 산정
4) 소프트웨어 개발 위 험 산정

정답: 1
풀이: COCOMO(Constructive Cost Model) 모형은 소프트웨어 개발 프로젝트의 비용, 시간, 자원 등을 예측하기 위해 사용되는 모델입니다. 이 모형은 프로젝트의 규모(주로 코드의 라인 수)에 기반하여 소프트웨어 개발에 필요한 비용을 산정하는 데 중점을 두고 있습니다. COCOMO 모형은 소프트웨어 개발 비용 산정뿐만 아니라, 개발 시간과 자원 산정에도 활용되지만, 질문에서 묻는 주된 목적은 소프트웨어 개발 비용 산정입니다. 따라서 주어진 보기 중에서 '소프트웨어 개발 비용 산정'이 COCOMO 모형의 주요 목적이므로 정답은 1번입니다. 과목: 소프트웨어개발

문제 23. 기능 점 수(Function Point) 모형의 설명으로 맞 는 것은?
1) 소프트웨어 개발 비용을 산정하는 방법
2) 소프트웨어 개발 시간을 산정하는 방법
3) 소프트웨어의 기능에 따라 점 수를 부 여하여 개발 규 모를 산정하는 방법
4) 소프트웨어의 성능에 따라 점 수를 부 여하여 개발 규 모를 산정하는 방법

정답: 3
풀이: 기능 점 수(Function Point) 모형은 소프트웨어의 기능에 따라 점수를 부여하여 개발 규모를 산정하는 방법입니다. 이 모형은 소프트웨어의 기능적 요구사항을 기반으로 하여, 사용자가 소프트웨어를 통해 수행할 수 있는 기능의 수를 측정합니다. 각 기능은 특정한 점수로 평가되며, 이를 통해 전체 소프트웨어의 규모를 파악할 수 있습니다. 따라서 주어진 보기 중에서 '소프트웨어의 기능에 따라 점 수를 부여하여 개발 규모를 산정하는 방법'이라는 설명이 가장 정확하게 기능 점 수 모형을 설명하고 있으므로 정답은 3번입니다. 나머지 보기들은 기능 점 수 모형의 정의와 맞지 않거나 부정확한 설명입니다. 과목: 소프트웨어설계

문제 24. 내 장 형(Embedded Mode) 소프트웨어 개발 유형에 해당하는 것은?
1) 5만 라 인 이하의 소프트웨어를 개발하는 유형
2) 30만 라 인 이하의 소프트웨어를 개발하는 유형
3) 30만 라 인 이상의 초 대형 규 모의 트 랜잭 션 처리 시스템이나 운영 체제 등 을 개발하는 유형
4) 범 용 환경 에서 사용이 가능하도록 일반적인 배포 형태로 패 키 징 이 진행 되 는 유형

정답: 1
풀이: 사용자가 제시한 문제는 "내 장 형(Embedded Mode) 소프트웨어 개발 유형에 해당하는 것은?"입니다. 제시된 보기 중에서 '5만 라인 이하의 소프트웨어를 개발하는 유형'은 임베디드 소프트웨어 개발의 일반적인 범위에

해당합니다. 임베디드 시스템은 보통 하드웨어에 밀접하게 연관되어 작동하며, 비교적 작은 코드베이스를 가지는 경우가 많습니다. 나머지 보기는 30만 라인 이하 또는 30만 라인 이상의 소프트웨어 개발 유형을 언급하고 있어 임베디드 소프트웨어의 특성과는 거리가 있습니다. 따라서 정답은 '5만 라인 이하의 소프트웨어를 개발하는 유형'인 1번입니다. 과목: 소프트웨어설계

문제 25. DDL에 속하는 명 령 어는?
1) SELECT
2) INSERT
3) CREATE
4) UPDATE

정답: 3
풀이: DDL(Data Definition Language)은 데이터베이스의 구조를 정의하고 수정하는 데 사용되는 SQL 명령어의 집합입니다. 주어진 보기 중에서 'CREATE' 명령어는 데이터베이스 객체(예: 테이블, 뷰 등)를 생성하는 데 사용되므로 DDL에 해당합니다. 반면, 'SELECT', 'INSERT', 'UPDATE'는 데이터 조작 언어(DML)에 속하며, 데이터의 조회 및 수정과 관련된 명령어입니다. 따라서 DDL에 속하는 명령어는 'CREATE'이며, 정답은 3번입니다. 과목: 데이터베이스구축

문제 26. DML에 속하는 명 령 어들은?
1) SELECT, INSERT, UPDATE, DELETE
2) CREATE, ALTER, DROP
3) COMMIT, ROLLBACK
4) GRANT, REVOKE

정답: 1
풀이: DML(Data Manipulation Language)은 데이터베이스에서 데이터를 조작하는 데 사용되는 SQL 명령어의 집합입니다. 주어진 보기 중에서 'SELECT', 'INSERT', 'UPDATE', 'DELETE'는 모두 DML에 속하는 명령어입니다. 각각의 명령어는 다음과 같은 기능을 합니다: - 'SELECT': 데이터베이스에서 데이터를 조회하는 명령어 - 'INSERT': 데이터베이스에 새로운 데이터를 추가하는 명령어 - 'UPDATE': 이미 존재하는 데이터를 수정하는 명령어 - 'DELETE': 데이터베이스에서 데이터를 삭제하는 명령어 나머지 보기인 'CREATE, ALTER, DROP'은 DDL(Data Definition Language)에 속하며, 'COMMIT, ROLLBACK'은 트랜잭션 제어 언어(TCL)에 속합니다. 'GRANT, REVOKE'는 권한 제어에 관련된 명령어로 DCL(Data Control Language)에 속합니다. 따라서 DML에 속하는 명령어는 'SELECT, INSERT, UPDATE, DELETE'이며, 정답은 1번입니다. 과목: 데이터베이스구축

문제 27. SQL에서 데이터를 입력 하는 명 령 어는?
1) INSERT INTO
2) UPDATE
3) DELETE FROM
4) SELECT

정답: 1
풀이: SQL에서 데이터를 입력하는 명령어는 'INSERT INTO'입니다. 이 명령어는 데이터베이스의 특정 테이블에 새로운 레코드를 추가할 때 사용됩니다. 다른 보기인 'UPDATE'는 기존의 데이터를 수정하는 명령어이며, 'DELETE FROM'은 특정 데이터를 삭제하는 명령어입니다. 'SELECT'는 데이터를 조회하는 명령어로, 데이터베이스에서 정보를 검색할 때 사용됩니다. 따라서 데이터 입력을 위한 명령어는 'INSERT INTO'가 맞습니다. 과목: 데이터베이스구축

문제 28. SELECT 문의 기 본 구조로 옳은 것은?
1) SELECT * FROM 테 이 블 명
2) INSERT INTO 테 이 블 명 VALUES (값)
3) UPDATE 테 이 블 명 SET 열 = 값
4) DELETE FROM 테 이 블 명

정답: 1
풀이: SQL의 SELECT 문은 데이터베이스에서 데이터를 조회할 때 사용하는 기본적인 명령어입니다. SELECT 문의 기본 구조는 `SELECT * FROM 테이블명`으로, 이는 특정 테이블에서 모든 열의 데이터를 선택하여 조회하겠다는 의미입니다. 다른 보기들은 SQL의 다른 명령어들로, INSERT는 데이터를 추가하고, UPDATE는 기존 데이터를 수정하며, DELETE는 데이터를 삭제하는 명령어입니다. 따라서 주어진 보기 중에서 SELECT 문의 기본 구조에 해당하는

것은 'SELECT * FROM 테이블명'이므로 정답은 1번입니다. 과목: 데이터베이스구축

문제 29. 데이터베이스의 논 리적 설계 단계에서 수행하는 작업 은?

1) 데이터 모델링
2) 데이터베이스 구조 설계
3) 데이터 사 전 작 성
4) 데이터 변환

정답: 1
풀이: 데이터베이스의 논리적 설계 단계에서 수행하는 작업은 '데이터 모델링'입니다. 데이터 모델링은 데이터베이스의 구조를 정의하고, 데이터 간의 관계를 명확히 하여 데이터베이스 설계를 위한 기초를 마련합니다. 이 단계에서는 개체(Entity), 속성(Attribute), 관계(Relationship) 등을 정의하여 데이터의 흐름과 저장 방식을 시각적으로 표현합니다. 나머지 보기인 '데이터베이스 구조 설계', '데이터 사전 작성', '데이터 변환'은 데이터베이스 설계의 다른 단계나 작업에 해당하며, 논리적 설계 단계에서 주로 수행되는 작업은 데이터 모델링입니다. 따라서 정답은 1번입니다.
과목: 데이터베이스구축