

## 문제집 - 문제(1~29)

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.

문제 2. UML 모델에서 한 객체가 다른 객체에게 오퍼레이션을 수행하도록 지정하는 의미적 관계로 옳은 것은?

1. Dependency
2. Realization
3. Generalization
4. Association

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.

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.

문제 5. 소프트웨어 설계에서 객체의 행위를 분석하는 다이어그램은?

1. 클래스 다이어그램
2. 유스케이스 다이어그램
3. 시퀀스 다이어그램
4. 상태 다이어그램

문제 6. 다음 중 소프트웨어 설계의 주요 항목이 아닌 것은?

1. 화면 설계
2. 애플리케이션 설계
3. 데이터베이스 설계
4. 네트워크 설계

문제 7. 소프트웨어 아키텍처의 유형 중 '파이프 필터 아키텍처'에 대한 설명으로 옳은 것은?

1. 데이터는 파이프를 통해 단방향으로 흐른다.
2. 데이터는 파이프를 통해 양방향으로 흐른다.
3. 필터는 데이터를 변환하고 처리한다.

4. 파이프는 데이터를 저장하는 역할을 한다.

문제 8. 디자인 패턴의 종류 중 생성 패턴에 해당하는 것은?

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

문제 9. 소프트웨어 설계에서 결합도에 대한 설명으로 옳은 것은?

1. 결합도가 높을수록 소프트웨어의 유지보수가 쉽다.
2. 결합도가 낮을수록 소프트웨어의 재사용성이 높다.
3. 결합도는 모듈 간의 의존성을 나타낸다.
4. 결합도는 모듈의 응집력을 나타낸다.

문제 10. 객체지향 설계에서 다이어그램을 사용하여 객체의 행위를 표현하는 것은?

1. 클래스 다이어그램
2. 유스케이스 다이어그램
3. 시퀀스 다이어그램
4. 상태 다이어그램

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.

문제 12. 다음 중 소프트웨어 설계의 검증 방법이 아닌 것은?

1. 테스트
2. 리뷰
3. 워크스루
4. 컴파일

문제 13. 소프트웨어 설계에서 인터페이스 설계의 중요성이 아닌 것은?

1. 시스템 간의 호환성
2. 사용자의 편의성
3. 소프트웨어의 유지보수
4. 하드웨어의 성능

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.

문제 15. 소프트웨어 패키징의 정의로 가장 적절한 것은?

1. 모듈별로 생성한 실행 파일을 묶어 배포용 설치파일을 만드는 것
2. 소프트웨어 개발 과정의 첫 단계
3. 사용자 요구사항을 분석하는 과정
4. 소프트웨어 테스트를 수행하는 과정

문제 16. 소프트웨어 품질 목표 중 하나 이상의 하드웨어 환경에서 운용되기 위해 쉽게 수정될 수 있는 시스템 능력을 의미하는 것은?

1. Correctness
2. Portability
3. Efficiency
4. Usability

문제 17. 단위 테스트(Unit Test)의 주목적인 것은?

1. 통합 테스트
2. 시스템 테스트
3. 인수 테스트
4. 개별 모듈이 정확하게 구현되었는지 확인하는 것

문제 18. DRM(Digital Rights Management)의 구성 요소에 포함되지 않는 것은?

1. 클리어링 하우스

2. 콘텐츠 제공자
3. 분배자(유통)
4. 소비자

문제 19. 소프트웨어 생명주기 모형 중 폭포수 모형의 장점이 아닌 것은?

1. 모형의 적용 경험과 성공 사례가 많음
2. 단계별 정의가 분명하고, 전체 공조의 이해가 용이
3. 단계별 산출물이 정확하여 개발 공정의 기준점을 잘 제시
4. 개발 과정 중에 발생하는 새로운 요구나 경험을 설계에 반영하기 쉬움

문제 20. 소프트웨어 재공학의 이점으로 틀린 것은?

1. 위험 부담 감소
2. 개발 시간 단축
3. 개발 비용 절감
4. 시스템 명세의 오류 증가

문제 21. 소프트웨어 재사용 방법 중 합성 중심 방법에 해당하는 것은?

1. 전자 칩과 같은 소프트웨어 부품, 즉 블록(모듈)을 만들어서 끼워 맞추어 소프트웨어를 완성시키는 방법
2. 추상화 형태로 쓰여진 명세를 구체화하여 프로그램을 만드는 방법
3. 소프트웨어 개발 과정을 테스트하는 방법
4. 소프트웨어 결과를 테스트하는 방법

문제 22. COCOMO 모형이 사용되는 목적은?

1. 소프트웨어 개발 비용 산정
2. 소프트웨어 개발 시간 산정
3. 소프트웨어 개발 자원 산정
4. 소프트웨어 개발 위험 산정

문제 23. 기능 점수(Function Point) 모형의 설명으로 맞는 것은?

1. 소프트웨어 개발 비용을 산정하는 방법
2. 소프트웨어 개발 시간을 산정하는 방법
3. 소프트웨어의 기능에 따라 점수를 부여하여 개발 규모를 산정하는 방법
4. 소프트웨어의 성능에 따라 점수를 부여하여 개발 규모를 산정하는 방법

문제 24. 내장형(Embedded Mode) 소프트웨어 개발 유형에 해당하는 것은?

1. 5만 라인 이하의 소프트웨어를 개발하는 유형
2. 30만 라인 이하의 소프트웨어를 개발하는 유형
3. 30만 라인 이상의 초대형 규모의 트랜잭션 처리 시스템이나 운영체제 등을 개발하는 유형
4. 범용 환경에서 사용이 가능하도록 일반적인 배포 형태로 패키징이 진행되는 유형

문제 25. DDL에 속하는 명령어는?

1. SELECT
2. INSERT
3. CREATE
4. UPDATE

문제 26. DML에 속하는 명령어들은?

1. SELECT, INSERT, UPDATE, DELETE
2. CREATE, ALTER, DROP
3. COMMIT, ROLLBACK
4. GRANT, REVOKE

문제 27. SQL에서 데이터를 입력하는 명령어는?

1. INSERT INTO
2. UPDATE
3. DELETE FROM
4. SELECT

문제 28. SELECT 문의 기본 구조로 옳은 것은?

1. SELECT \* FROM 테이블명

2. INSERT INTO 테이블명 VALUES (값)
3. UPDATE 테이블명 SET 열=값
4. DELETE FROM 테이블명

문제 29. 데이터베이스의 논리적 설계 단계에서 수행하는 작업은?

1. 데이터 모델링
2. 데이터베이스 구조 설계
3. 데이터 사전 작성
4. 데이터 변환