1. 형상 관리(configuration management)의 활동 중 형상 식별(configuration identification) 단계에서 수행되는 활동을 <보기>에서 모두 고른 것은?

<보기>
ㄱ. 형상 관리 대상을 선정
ㄴ. 변경사항 요청 및 심사
ㄷ. 형상 관리를 위한 식별자 규칙 선정
ㄹ. 변경을 완료하고 새로운 버전 번호 부여
ㅁ. 형상 관리가 계획서대로 관리되고 있는지 검사

① ㄱ, ㄷ ② ㄴ, ㄷ
③ ㄷ, ㄹ ④ ㄹ, ㅁ

2. 컴포넌트 기반 소프트웨어 개발(CBSD)에 대한 설명으로 가장 옳지 않은 것은?
① 고객이 원하는 요구들을 지속적으로 피드백한다.
② 이미 개발된 컴포넌트를 조립하여 시스템을 개발하는 과정이다.
③ 도메인 영역에서 재사용이 가능한 기능적 요구사항을 명확하게 정의하는 것이 필요하다.
④ 기능의 추가나 변경이 필요한 경우 요구에 따라 커스터마이징이 가능하다.

3. 시퀀스(sequence) 다이어그램에 대한 설명으로 가장 옳지 않은 것은?
① 각각의 유스케이스(use case)에 대해서 작성하는 것이 일반적이다.
② 유스케이스 시나리오를 확장하여 시스템 내부의 객체들이 상호작용하는 과정까지는 나타낼 수 없다.
③ 데이터의 송수신은 매개변수와 반환값으로 표현된다.
④ 상호작용 과정에서 어떠한 데이터를 주고받았는지 밝힐 수 있다.

4. 설계 단계에서 고려할 사항으로 가장 옳지 않은 것은?
① 설계 과정에서 추상화 메커니즘을 적절히 이용한다면 구조적이고 단계적인 설계를 할 수 있다.
② 정보은닉은 필요하지 않은 정보는 접근할 수 없도록 하여, 한 모듈 또는 하부 시스템이 다른 모듈의 구현에 영향을 받지 않게 설계되는 것을 의미한다.
③ 단계적 정제는 기본 설계 단계에서 나타나는 프로그램의 구조에서 점차 모듈에 대한 세부 사항으로 내려가면서 구체화된다.
④ 모듈화는 시스템을 지능적으로 관리할 수 있지만 복잡도의 문제를 해결할 수는 없다.

5. 상향식(bottom-up) 통합테스트에 대한 설명으로 가장 옳지 않은 것은?
① 프로그램 구조에서 가장 하위레벨에 있는 컴포넌트부터 구축하고 테스트한다.
② 하위레벨의 컴포넌트를 결합하여 특정 소프트웨어 하위기능을 수행하는 클러스터(cluster)를 만든다.
③ 테스트 케이스 입력과 출력을 조정하기 위한 드라이버(driver)를 작성한다.
④ 테스트된 컴포넌트는 아직 테스트되지 않은 컴포넌트의 호출을 위하여 스텁(stub)을 작성한다.

6. 시스템을 릴리스하고 사용자에게 배포, 설치한 후 사용하다 보면 변경이 필요하게 된다. 변경의 이유로 가장 옳지 않은 것은?
① 운영 환경의 변화 – 하드웨어, 플랫폼, 시스템 형상의 변화로 소프트웨어 수정이 필요하다.
② 비즈니스 절차의 변화 – 비즈니스 운영 절차가 바뀌면 소프트웨어 시스템도 그에 따라서 변경하여야 한다.
③ 미래의 문제를 배제하기 위하여 변경 – 미래에 일어날 수 있는 문제를 피하기 위하여 소프트웨어 시스템에 대한 변경이 필요하다.
④ 버그 제거 – 신뢰성을 개선하기 위하여 복잡한 컴포넌트를 다시 설계하고 구현한다.

7. 요구사항 타당성 검증 시의 내용 중 가장 옳지 않은 것은?
① 명확성 – 요구분석의 내용이 모호함 없이 모든 참여자들에 의해 명확하게 이해될 수 있는가?
② 일관성 – 사용자의 요구를 에러 없이 완전하게 반영하고 있는가?
③ 검증 가능성 – 요구사항 명세서에 기술된 내용이 사용자의 요구를 만족하는가?
④ 추적 가능성 – 시스템 요구사항과 시스템 설계 문서를 추적할 수 있는가?

8. 설계품질의 결합도 중 가장 바람직한 결합도에 해당하는 것은?
① 내용결합도(content coupling)
② 제어결합도(control coupling)
③ 데이터결합도(data coupling)
④ 외부결합도(external coupling)

9. Lehman의 소프트웨어 진화 법칙에 대한 설명으로 가장 옳지 않은 것은?
① 시스템이 릴리스된 후 그 시스템이 다른 시스템으로 대체될 때까지 변경된다.
② 진화하는 시스템의 유지보수 프로세스의 평균 작업량은 시스템이 소멸될 때까지 일정하다.
③ 소프트웨어 시스템의 구조는 변경될수록 복잡도가 낮아지는 경향이 있다.
④ 시스템의 평균 성장률은 소멸될 때까지 일정하다. 즉 시스템 개발의 전 단계에 걸쳐 각 버전의 변화는 거의 일정하다.

10. <보기>는 A, B회사의 월 평균 생산성과 1인당 월 평균 임금을 보여준다. 두 회사가 각각 49,300 LOC(Line of Code) 시스템 개발에 참여할 경우에 대한 설명으로 가장 옳지 않은 것은? (단, LOC 외의 다른 비용은 고려하지 않는다.)

<보기>

회사명 / 회사 정보	A	B
월 평균 생산성 LOC/MM(man month)	725	580
1인당 월 평균 임금	410만 원	320만 원

① A 회사의 노력(man month)은 68MM이다.
② B 회사의 노력(man month)은 85MM이다.
③ A 회사가 B 회사보다 프로젝트 총 비용이 적게 든다.
④ B 회사의 프로젝트 총 비용은 27,200만 원이다.

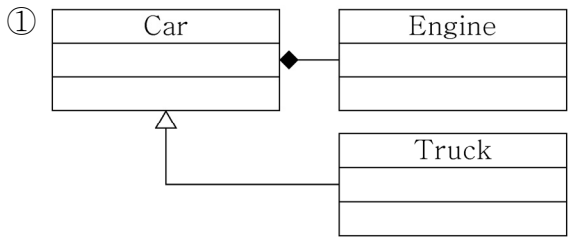
11. 나선형(spiral) 모델의 특징으로 가장 옳지 않은 것은?
① 반복 주기가 시작될 때 소프트웨어의 목표와 제약 조건을 결정한다.
② 한 번의 개발 주기를 거치면서 시스템이 완성된다.
③ 프로토타입을 만들면서 위험을 분석한다.
④ 개발을 위한 계획 및 요구분석 후에 위험 요소에 대하여 검토한다.

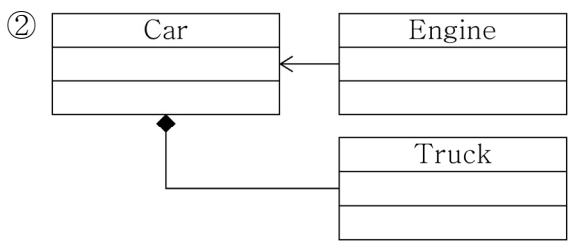
12. 스크럼(scrum) 방법론에서 SRS(Software Requirement Specification)나 TRS(Technical Requirement Specification)에 해당하는 목록은?
① 제품 백로그(product backlog)
② 스프린트(Sprint)
③ 스크럼 마스터(Scrum master)
④ 스프린트 트래킹(Sprint tracking)

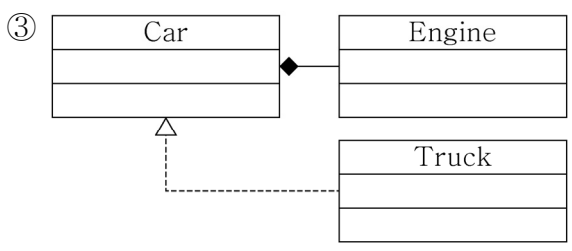
13. <보기>의 자바 소스코드를 UML로 표현했을 때 가장 옳은 것은?

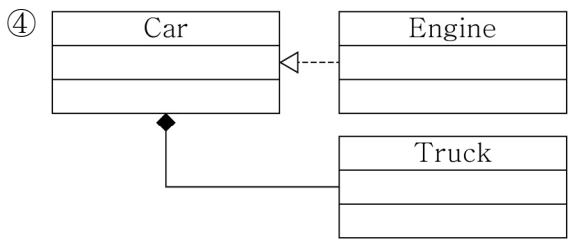
<보기>

```
Class Engine {
    …
}
Class Car {
    Engine carEngine = new Engine();
    …
}
Class Truck extends Car{
    …
}
```

①


②


③


④


14. ISO/IEC 9126의 6가지 품질 특성 중 신뢰성(reliability)을 구성하는 하위 특성에 대한 설명으로 가장 옳은 것은?
① 적응성(adaptability)은 소프트웨어가 설치되어 있는 운영체제나 미들웨어 또는 하드웨어 환경에서 정상적으로 잘 작동할 수 있는 능력을 말한다.
② 안정성(stability)은 소프트웨어 변경으로 인한 예상치 못한 결과를 최소화하는 능력을 말한다.
③ 정확성(accuracy)은 사용자가 요구하는 정밀도를 유지하거나 허용 범위 내의 결괏값을 제공할 수 있는 능력을 말한다.
④ 결함 수용성(fault tolerance)은 소프트웨어 일부에서 고장이 발생해도 요구되는 기능을 유지할 수 있는지를 나타낸다.

15. <보기>는 CMMI의 성숙도(maturity) 5단계의 내용을 요약하고 있다. 이들을 초기 단계부터 최적화 단계까지 순서대로 바르게 나열한 것은?

<보기>

㉮ 기본적인 프로젝트 관리 체계 수립
㉯ 프로세스 최적화
㉰ 조직 차원의 표준 프로세스를 통한 프로젝트 지원
㉱ 정량적으로 프로세스가 측정/통제됨
㉲ 예측/통제 불가능

① ㉱ – ㉰ – ㉮ – ㉲ – ㉯
② ㉱ – ㉮ – ㉰ – ㉲ – ㉯
③ ㉲ – ㉰ – ㉮ – ㉱ – ㉯
④ ㉲ – ㉮ – ㉰ – ㉱ – ㉯

16. <보기>와 같은 특징을 갖는 테스트 기법은?

<보기>

- 프로그램의 소스코드를 분석해 결함을 찾아내는 테스트 기법
- 시스템의 실행을 동반하지 않음
- 무한루프, 초기화되지 않은 변수, 실행 불가능한 코드 등을 발견할 수 있음
- 소스코드의 잠재적 오류를 찾아냄으로써 디버깅에 유용한 정보 생성

① 정적 분석(static analysis)
② 단위 테스트(unit test)
③ 스모크 테스트(smoke test)
④ 코드 스멜(code smell)

17. GoF(Gang of Four)가 제시한 디자인 패턴에 대한 설명으로 가장 옳지 않은 것은?

① Factory Method – 객체를 생성하는 처리를 파생 클래스로 분리하여 처리하도록 캡슐화하는 패턴
② Adapter – 한 클래스의 인터페이스를 클라이언트에서 필요로 하는 인터페이스로 변환해주는 패턴
③ State – 객체의 상태에 따라 객체의 행위 내용을 변경해주는 패턴
④ Strategy – 이미 고정된 자료구조에 행위를 쉽게 추가할 수 있도록 해주는 패턴

18. ISO 품질 평가 표준에 대한 설명으로 가장 옳지 않은 것은?

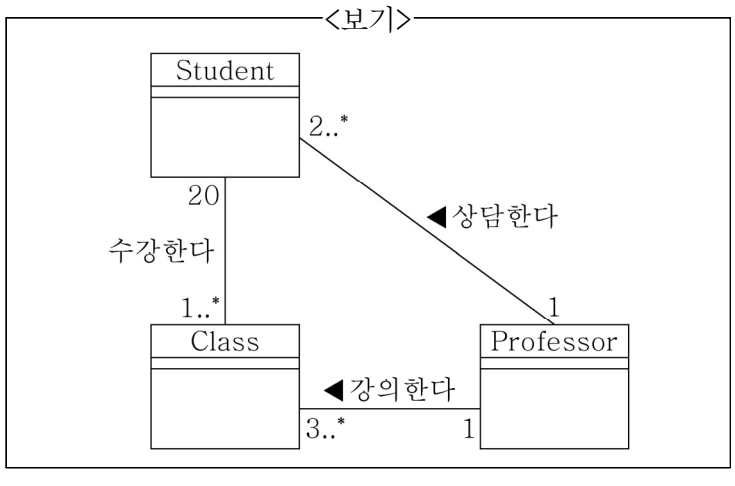
① ISO/IEC 9126 : 소프트웨어 품질의 특성을 정의하고 품질평가 메트릭을 정의
② ISO/IEC 12207 : 설계, 생산, 설치 및 서비스 과정에 대한 품질보증 모델
③ ISO/IEC 14598 : 소프트웨어 제품의 품질을 측정하거나 평가하기 위한 방법과 절차를 규정
④ ISO/IEC 15504 : 소프트웨어 프로세스를 평가하고 개선함으로써 품질 및 생산성을 높이기 위한 표준

19. 소프트웨어 아키텍처 스타일에 대한 설명으로 가장 옳지 않은 것은?

① 계층형(layering) : 시스템을 기능에 따라 수직적인 계층으로 구분하고, 각 계층은 서비스의 집합을 제공
② MVC(Model-View-Controller) : 사용자 인터페이스와 데이터 처리 로직을 독립적으로 분리하여 변경에 대한 영향을 줄임
③ 파이프 필터(pipe and filter) : 프로그램에서 감지되고 처리될 수 있는 사건(event) 중심의 시스템에 적합한 아키텍처
④ 데이터 중심형(저장소 구조)(repository) : 주요 데이터가 중앙에서 관리되고 서브시스템이 공유데이터에 접근해 정보를 사용

20. <보기>의 클래스(class) 다이어그램에 대한 설명으로 가장 옳은 것은?

<보기>



① 'Professor'는 1개 이상의 'Class'를 강의한다.
② 'Class'를 20명 이상의 'Student'가 수강한다.
③ 'Professor'는 2명 이상의 'Student'를 상담한다.
④ 'Student'는 20개의 'Class'를 수강한다.

이 면은 여백입니다.