

1. 소프트웨어 개발 프로세스 모델에 해당하지 않는 것은?

- ① 폭포수 모델(waterfall model)
- ② 요구사항 엔지니어링(requirements engineering)
- ③ 프로토타이핑(prototyping)
- ④ 증분형 개발 모델(incremental model)

2. 소프트웨어 아키텍처 평가 방법 중, 여러 가지 품질 속성에 초점을 맞추어 평가하여 아키텍처의 설계 타협점을 찾아내고, 아키텍처 내부에 숨어 있는 리스크를 발견하는데 사용되는 것은?

- ① SAAM(Software Architecture Analysis Method)
- ② SAEM(Software Architecture Evaluation Method)
- ③ ATAM(Architecture Trade-off Analysis Method)
- ④ AQEM(Architecture Quality Evaluation Method)

3. 정적 분석 도구에 대한 설명으로 가장 옳지 않은 것은?

- ① 원시 코드가 실행되는 동안의 동작이나 반응을 추적하고 보고한다.
- ② 원시 코드의 문법적 적합성을 자동으로 평가하여 잘못된 문장을 표기한다.
- ③ 원시 코드의 그래프를 생성하여 논리 흐름을 보여주고 구조적 결함이 있는지 체크한다.
- ④ 원시 코드에 정의된 데이터 구조 및 선언, 컴포넌트 인터페이스를 검사하여 잘못된 링크나 데이터 정의의 충돌, 잘못된 데이터의 사용 등을 발견한다.

4. 애자일 방법론과 연관성이 가장 적은 것은?

- ① 사용자 스토리 ② 분산 시스템 모델링
- ③ 스크럼 ④ 짝 프로그래밍

5. <보기>는 모듈 간의 결합 형태를 보여주고 있다. 결합도가 약한 것부터 강한 것 순서대로 바르게 나열한 것은?

<보기>

- ㄱ. 공통 결합(common coupling)
- ㄴ. 내용 결합(content coupling)
- ㄷ. 스탬프 결합(stamp coupling)
- ㄹ. 자료 결합(data coupling)
- ㅁ. 제어 결합(control coupling)

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

6. '문서편집기 소프트웨어'의 비기능적 요구사항으로 가장 옳은 것은?

- ① 텍스트 형식(TXT)으로 작성된 문서를 불러들일 수 있어야 한다.
- ② HTML 형식으로 문서를 저장할 수 있어야 한다.
- ③ 문서 저장은 3초 이내에 완수되어야 한다.
- ④ 문서 저장 중 사용자가 단축키로 취소할 수 있어야 한다.

7. RUP(Rational Unified Process)의 4+1 뷰(view)에 대한 <보기>의 설명과 용어를 옳게 짝지은 것은?

<보기>

- (가) 주로 설계자와 개발자를 위한 것으로 클래스, 객체, 관계 등과 같은 정적인 구조와 객체가 다른 객체에 메시지를 전달할 때 발생하는 동적인 협동을 설명한다.
- (나) 자원의 효율적인 사용, 병행 실행 및 비동기 이벤트의 처리뿐 아니라 병행 실행되는 스레드 간의 통신과 동기화 등을 다룬다.

(가)

(나)

- ① 구현(implementation) 뷰 프로세스(process) 뷰
- ② 논리(logical) 뷰 구현(implementation) 뷰
- ③ 논리(logical) 뷰 프로세스(process) 뷰
- ④ 프로세스(process) 뷰 구현(implementation) 뷰

8. 소프트웨어 형상 변경 제어 활동에서, 형상 항목을 변경하는 이유를 <보기>에서 모두 고른 것은?

<보기>

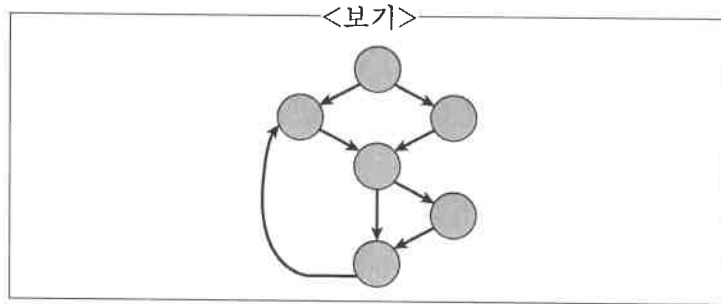
- ㄱ. 소프트웨어의 결함
- ㄴ. 하드웨어의 변경
- ㄷ. 운영 요구의 변경
- ㄹ. 고객이나 사용자로부터의 개선 요구
- ㅁ. 예산, 프로젝트 일정, 기간의 변경

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

9. Z명세 기법에 대한 설명으로 가장 옳지 않은 것은?

- ① 대수학적 방법에 해당한다.
- ② 정형적 명세 언어를 사용한다.
- ③ 명세를 작성하는 단위는 스키마이다.
- ④ 집합과 함수, 술어 논리 등의 수식을 사용한다.

10. <보기>의 제어 흐름 그래프의 순환 복잡도(cyclomatic complexity)는?



- ① 4 ② 5
③ 6 ④ 7

11. 소프트웨어 아키텍처 패턴에 해당하는 것은?

- ① 객체 지향 설계(Object Oriented Design)
② 리포지토리(repository)
③ 테스트 주도 개발(Test-Driven Development)
④ 유스케이스 다이어그램(usecase diagram)

12. <보기>의 조건을 따라 원시코드 라인 수(LOC: Line of Code) 기법으로 계산한 소프트웨어 개발의 노력(Man-Month)으로 가장 옳은 값[MM]은?

- <보기>
- 소프트웨어 프로젝트의 예상 원시코드 라인 수 : 80,000 LOC
 - 생산성 계수: 600 LOC/MM
 - 프로젝트 복잡도 계수: 1.2
 - 추가적인 유지보수 인력: 전체의 10% 추가 투입

- ① 133.3 ② 146.7
③ 160.0 ④ 176.0

13. 프로젝트관리 지식체계(PMBOK)의 관리 항목과 세부 프로세스의 연결이 옳은 것을 <보기>에서 모두 고른 것은?

- <보기>
- ㄱ. 프로젝트 의사소통 관리: 프로젝트 조직 기획, 팀 확보, 팀 개발
 - ㄴ. 프로젝트 위험 관리: 자원 기획, 비용 산정, 비용 예산 수립, 비용 통제
 - ㄷ. 프로젝트 범위 관리: 착수, 범위 기획, 범위 정의, 범위 검증, 범위 변경 통제
 - ㄹ. 프로젝트 일정 관리: 작업 정의, 작업 순서, 작업 기간 산정, 일정 개발, 일정 통제

- ① ㄱ, ㄷ ② ㄴ, ㄷ
③ ㄴ, ㄹ ④ ㄷ, ㄹ

14. SPICE(ISO/IEC 15504) 모델에서 정의한 프로세스 수행 능력 단계(level)에 해당하지 않는 것은?

- ① 수행 단계(performed process)
② 관리 단계(managed process)
③ 검증 단계(verified process)
④ 최적화 단계(optimizing process)

15. 맥콜(McCall)의 품질 요소에 대한 설명으로 옳은 것을 <보기>에서 모두 고른 것은?

- <보기>
- ㄱ. 맥콜의 품질 요소는 크게 제품 운영, 제품 개선, 제품 변환으로 분류할 수 있다.
 - ㄴ. 효율성은 사용자가 요구하는 기능을 수행하는 데 CPU와 메모리 같은 자원을 얼마나 사용하는지와 관련된 특성이다.
 - ㄷ. 이식성은 시스템의 일부나 전체를 다른 애플리케이션에서도 얼마나 쉽게 사용할 수 있는지를 나타낸다.
 - ㄹ. 무결성은 개발된 소프트웨어가 사용자의 기능적 요구 사항을 담은 요구분석명세서와 얼마나 일치하는지를 나타낸다.

- ① ㄱ, ㄴ ② ㄱ, ㄷ
③ ㄴ, ㄷ ④ ㄴ, ㄹ

16. 화이트박스 테스트의 케이스 선정 기준에 대한 설명으로 가장 옳지 않은 것은?

- ① 경로 검증 기준: 프로그램에 존재하는 모든 실행 가능한 경로를 적어도 1번은 실행해야 한다.
② 기능 검증 기준: 프로그램에 구현된 모든 기능을 적어도 1번은 실행해야 한다.
③ 문장 검증 기준: 프로그램의 모든 문장을 적어도 1번은 실행해야 한다.
④ 분기 검증 기준: 프로그램에 존재하는 모든 분기점에서 참과 거짓에 해당하는 경로를 각각 적어도 1번씩 실행해야 한다.

17. 델파이 기법의 비용 산정 과정에 대한 설명으로 옳은 것을 <보기>에서 모두 고른 것은?

<보기>

- ㄱ. 조정자는 전문가가 모여 비용 산정을 하는 회의에서 간사 역할을 한다.
 ㄴ. 전문가 각자가 비용을 산정하고 이때 계산된 결과를 조정자에게 익명으로 제출한다.
 ㄷ. 전문가는 비용을 산정할 수 있는 자료를 충분히 검토하고, 필요하다면 익명으로 의견을 나눌 수 있다.
 ㄹ. 조정자는 각 전문가가 제출한 자료에서 산정 내용에 차이가 크면 이 문제를 해결하기 위해 회의를 소집한다.

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

18. <보기>는 프로젝트 P의 완성에 필요한 작업 T1~T5에 대해, 작업별 최소한의 소요 기간과 선행 작업을 보여 준다. 프로젝트 P 완성에 소요되는 최소한의 기간[일]은?

<보기>

작업	소요 기간(일)	선행 작업
T1	5	-
T2	8	-
T3	1	T1
T4	2	T1, T2
T5	3	T3, T4

- ① 13 ② 14 ③ 15 ④ 16

19. <보기 1>의 UML 다이어그램 용어와 <보기 2>의 설명을 옳게 짝지은 것은?

<보기 1>

- (가) 통신 다이어그램 (나) 상태 다이어그램
 (다) 컴포넌트 다이어그램 (라) 배치 다이어그램

<보기 2>

- A: 구현 관점에서 정적 모델링을 할 때 사용하는 것으로 어떤 실행 모듈이 존재하고 이들이 서로 어떤 연관성이 있는지의 종속 관계를 나타낸다.
 B: 객체 간에 메시지를 주고받는 관계를 링크라고 하며 링크를 사용해 객체 간의 관계를 표현한다.
 C: 하드웨어 자원을 명시적으로 정의해 시스템의 물리적인 요소를 모델링하고 노드 간의 관계를 나타낸다.
 D: 다이어그램을 그리는 데 필요한 요소는 상태, 전이, 이벤트이다.

- (가) (나) (다) (라)
 ① A B C D
 ② B D A C
 ③ C B D A
 ④ D C A B

20. <보기 1>은 옵서버 패턴(observer pattern)을 이용한 파이썬 코드를 보여준다. 이 코드를 실행한 결과가 <보기 2>가 되도록 (가), (나)에 들어갈 코드를 옳게 짝지은 것은?

<보기 1>

```
class Subject:
    def __init__(self):
        self._observers = []
        self._state = None

    def attach(self, o):
        self._observers.append(o)

    def notify(self):
        for o in self._observers:
            (가)

    def change_state(self, state):
        self._state = state
        (나)

class Observer:
    def __init__(self, id):
        self._id = id

    def update(self, state):
        print(f"{self._id}:{state}")

s = Subject()
s.attach(Observer(1))
s.attach(Observer(2))
s.change_state("Hello")
```

<보기 2>

1:Hello
 2:Hello

- (가) (나)
 ① self.change_state(self._state) self.attach(state)
 ② self.change_state(self._state) self.notify()
 ③ o.update(self._state) self.attach(state)
 ④ o.update(self._state) self.notify()