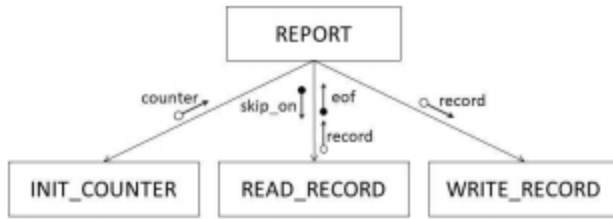


1. 다음 구조도(structure chart)에 표시된 모듈 REPORT와 하위 모듈 READ\_RECORD 사이에 나타난 결합(coupling)의 단계는 무엇인지 설명하시오. (5점)



REPORT 모듈과 READ\_RECORD 모듈 사이에 자료흐름과 제어흐름이 나타나고 있음.

- 변수나 자료구조를 전달하는 자료흐름은 자료결합이나 스탬프 결합으로 볼 수 있음.

- 제어흐름은 모듈의 논리적 흐름을 제어하는 제어용 신호를 주고 받는 제어결합으로 볼 수 있음.

2. ( **테스트 오라클** )은 소프트웨어 명세서로부터 추출한 옳다고 믿을 수 있는 값으로, 예상되는 결과나 그것을 구할 수 있는 수단을 말한다. 또한, 테스트를 위해 생성된 테스트 지원 프로그램을 ( **테스트 하니스** )라고 한다. 테스트 드라이버나 테스트 스텝이 이에 해당한다.

3. 모듈 안의 구성요소들이 공통의 목적을 달성하기 위해 관련되어 있는 정도를 응집력이라고 하며, 모듈 안의 구성요소가 동일한 입력과 출력을 사용하는 작업들로 구성된 ( **교환적 응집** )은 한 요소의 출력이 다른 요소의 입력으로 사용되도록 모듈이 구성된 ( **순차적 응집** ) 보다 바람직하지 않다.

4. 모듈화 설계에 대한 설명으로 옳지 않은 것을 모두 고르시오. ( )

- ① 모듈화가 잘 된 소프트웨어일수록 재사용 가능성이 높다.
- ② 모듈 사이의 불필요한 상호 교류를 최소화함으로써 결합력을 낮출 수 있다.
- ③ 모듈 사이의 기능적 관련성이 강할수록 응집력이 높다.
- ④ 기능응집력이 높은 모듈에서는 모든 요소가 하나의 기능 구현을 위해 구성된다.
- ⑤ 전역변수를 사용할 경우, 프로그램 코드가 간단해지므로 유지보수가 편리하다.

5. 아키텍처 스타일에 관한 설명 중 옳지 않은 것을 모두 고르시오. ( )

- ① Peer-to-Peer 구조에서 각 서브시스템은 서비스를 제공하고 요청하는 능력이 있다.
- ② 계층구조에서는 하위층의 서브시스템이 제공하는 서비스를 상위층의 서브시스템이 사용하도록 구성된다.
- ③ 중앙저장소 구조는 서브시스템들이 단일 중앙저장소 자료를 접근하고 변경하므로, 서브시스템들 사이의 결합이 매우 강하다.
- ④ MVC 구조는 사용자 인터페이스를 시스템의 다른 부분과 분리하여 변경에 대한 영향을 줄인다.
- ⑤ 파이프필터 구조는 서브시스템이 입력 데이터를 받아 처리하고 결과를 다른 시스템에 보내는 작업이 반복되는 형태로 인터랙티브 시스템에 적합하다.

6. 통합테스트 단계에 적용할 방법에 대한 선택이 옳지 못한 것을 고르시오. ( )

- ① 시스템 구조도의 최하위 층에 있는 모듈을 먼저 테스트하기 위하여 상향식을 선택하였다.
- ② 특수하고 중요한 기능을 수행하는 모듈을 먼저 테스트하기 위해 연쇄식을 채택하였다.
- ③ 일정 계획의 융통성을 확보하기 위해 빅뱅통합 테스트 방식을 채택하였다.
- ④ 명령어 처리 모듈을 먼저 구현하고 테스트하기 위해 하향식을 선택하였다.
- ⑤ 오류의 위치와 원인을 쉽게 찾아내기 위해 빅뱅통합을 선택하였다.

7. ( **형상관리** )는 개발주기 동안 생성된 산출물을 관리하고, 소프트웨어 시스템과 컴포넌트의 상태를 추적하는 작업이다. 특히 ( **베이스라인** )은 소프트웨어 개발 과정 중 산출물의 집합이 만들어지는 특정 시점으로 프로젝트의 중요한 상태를 정의하기 위해 사용하며, 프로젝트 성공을 위해 반드시 거쳐야 하는 중요한 지점이다.

8. 다음 그림처럼 설계 (가)를 설계 (나)와 같이 변경했다면, 가장 관련이 깊은 설계 원칙은 ? (                    )  
 (가) 서비스 구현 클래스의 a(), b() 연산을 사용하는 클라이언트 클래스와 서비스 구현 클래스와 연관된 관계 구조  
 (나) 서비스 구현 클래스는 서비스인터페이스에 의존하고 서비스 구현 클래스는 서비스인터페이스를 구현하는 구조



- ① 단일 책임의 원칙(Single Responsibility Principle)
- ② 리스코프 교체 원칙(Liskov Substitution Principle)
- ③ 의존관계 역전의 원칙(Dependency Inversion Principle)
- ④ 인터페이스 분리의 원칙(Interface Segregation Principle)

9. 컴퓨터과학과 대학원의 입학전형 기준이 평점 3.0 이상이라고 한다. 하지만 학부 전공이 컴퓨터과학이 아니면, 보충과목을 수강하도록 조건부로 합격이 된다. 대학원 합격여부에 대한 테스트케이스 설계를 위해 동치클래스로 분할하시오. (5점)

평점 →

전공 →

입학전형

→ 합격

→ 조건부합격

→ 불합격

분류	입력	출력
동치클래스1	평점 ≥ 3.0 AND 전공 == '컴퓨터과학'	합격
동치클래스2	평점 ≥ 3.0 AND 전공 != '컴퓨터과학'	조건부합격
동치클래스3	평점 < 3.0 AND 전공 == '컴퓨터과학'	불합격
동치클래스4	평점 < 3.0 AND 전공 != '컴퓨터과학'	불합격

10. 다음과 같은 명세에 따라 작성된 코드를 테스트하고자 한다. 프로그램의 분기 커버리지를 만족하기 위한 테스트 데이터의 집합은 ? (                    )

```

integer x, y, z
if (x > 10 AND y = 10)
then  z = 5
if(x = 10 OR z > 3)
then  z = z + 10
print(z);
          
```

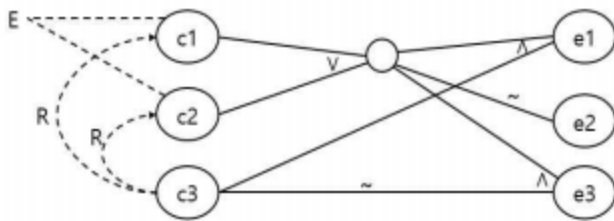
- ① (x = 20, y = 10, z = 0), (x = 20, y = 20, z = 0)
- ② (x = 20, y = 10, z = 10), (x = 20, y = 10, z = 0)
- ③ (x = 10, y = 10, z = 10), (x = 20, y = 20, z = 0)
- ④ (x = 20, y = 10, z = 10), (x = 20, y = 20, z = 0)
- ⑤ (x = 20, y = 10, z = 10), (x = 10, y = 10, z = 10)

11. 소프트웨어 아키텍처 관점은 시스템을 특정 구성요소와 그들 사이의 관계로 표현하는 것으로, 코드의 구조를 표현하는 (   모듈   ) 뷰, 런타임 구조인 통신 및 호출관계를 표현하는 (   컴포넌트와 커넥터   )뷰, 소프트웨어와 환경의 배치 구조를 표현하는 (   배치   ) 뷰가 있다.

12. 다음 설계 패턴에 대한 설명 중 옳지 않은 것을 모두 고르시오. (                    )
- ① 설계패턴은 GoF의 23개 패턴을 기반으로 생성패턴, 구조패턴, 행위패턴, 기능패턴으로 분류된다.
  - ② 싱글톤 패턴은 한 클래스의 인스턴스가 유일하게 하나만 생성되는 것을 보장하기 위해 사용한다.
  - ③ 어댑터 패턴은 객체의 생성이나 표현되는 방식에 관계없이 시스템을 독립적으로 생성할 수 있다.
  - ④ 설계문제에서 자주 발생하는 문제에 대한 일반적인 해결책인 설계패턴의 기본적인 원리는 위임과 상속이다.
  - ⑤ 옵저버 패턴은 발행구독 패턴으로도 알려져 있으며, 일대다 의존성 관계에 있는 객체의 변경문제를 다룬다.
  - ⑥ 기존 객체에 대한 변경 없이 새로운 행위를 추가하거나 다양한 작업을 수행할 때 사용하는 기능패턴은 비지터 패턴이다.

13. 다음과 같은 문제에 대한 원인결과 그래프를 작성하시오. (5점)

첫 번째 열의 문자가 'U' 또는 'R'이고 두 번째 열이 숫자(0~9)이면 파일이 업데이트가 수행된다. 첫 번째 문자가 잘못된 경우 메시지 'Access Denied'가 출력된다. 두 번째 열이 숫자가 아니면 메시지 'Invalid Input'가 출력된다.



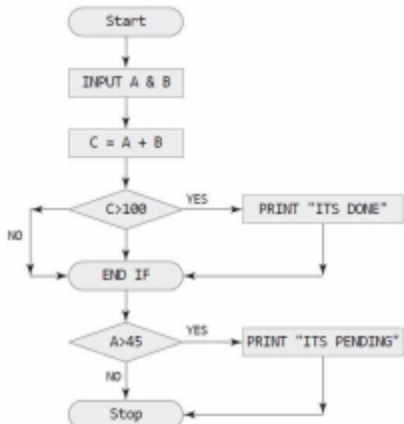
- c1 : 첫번째 열의 문자가 'U'
- c2 : 첫번째 열의 문자가 'R'
- c3 : 두번째 열이 숫자(0~9)
- e1 : 파일 업데이트 수행
- e2 : 'access denied' 출력
- e3 : 'invalid input' 출력

14. 오른쪽 <그림 1> 같은 코드에 대해 문장 커버리지를 만족하는 테스트 데이터 A, B를 작성하시오. (5점)

A = 50, B = 60

15. 다음 중 객체지향 방법론의 동적 모델링에 대한 설명 중 옳은 것을 모두 고르시오. ( )

- ① 동적 모델은 시스템 내부 설계를 위해 메시지의 교환이나 상태 변화를 표현한다.
- ② 동적 모델은 서브 시스템 사이의 구성을 모델링한다.
- ③ 동적 모델은 외부 자극에 대한 반응을 외부 관점에서 표현한 것이다.
- ④ 동적 모델은 시스템에 존재하는 객체들 사이의 인터랙션을 모델링한다.
- ⑤ 동적 모델은 시간의 흐름에 따른 시스템의 여러 요소의 변화를 나타낸다.
- ⑥ 동적 모델은 구조적 모델과 같은 관점을 갖는다.



<그림 1>

16. 유지보수의 유형 중 다음 설명에 해당하는 것은 무엇인가 ? ( )

운영체제나 컴파일러와 같은 프로그래밍 환경의 변화와 주변장치 또는 다른 시스템 요소가 항상 되거나 변경될 때 대처하기 위한 것이다.

- ① 수정형 유지보수
- ② 적응형 유지보수
- ③ 완성형 유지보수
- ④ 예방형 유지보수

17. 소프트웨어의 내적품질요소가 아닌 것을 모두 고르시오. ( )

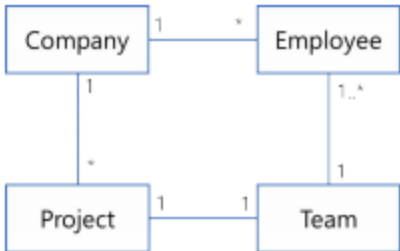
- ① 사용자 친숙성
- ② 검증가능성
- ③ 유지보수성
- ④ 이식성
- ⑤ 재사용성
- ⑥ 견고성
- ⑦ 상호운용성
- ⑧ 보안성

18. 액티비티 다이어그램의 요소에 대한 설명으로 틀린 것을 모두 고르시오. ( )

- |                                      |                           |
|--------------------------------------|---------------------------|
| ① 액션 - 분할 할 수 없는 동작                  | ② 객체 노드 - 객체 흐름에 연결된 객체   |
| ③ 상태 - 특정 제어 흐름을 나타냄                 | ④ 포크 - 액티비티 흐름이 병렬적으로 수행됨 |
| ⑤ 병합 - 액티비티의 병렬 흐름이 다시 합해지는 시점       |                           |
| ⑥ 스왑 레인 - 액티비티의 수행을 담당하는 역할에 따른 분할 선 |                           |

19. 다음은 4개 클래스(Company, Project, Employee, Team)의 관계를 설명하고 있다. 이 관계를 표현하는 클래스 다이어그램을 작성하시오. 다중도가 반드시 포함되도록 작성하시오. (5점)

- 회사(Company)는 프로젝트(Project)를 진행시킨다.
  - 각 프로젝트는 사원(Employee)으로 구성된 팀(Team)에 의해 수행된다.



20. 다음 설명을 UML 클래스 다이어그램으로 표현할 때 가장 적절한 관계는 ? ( )

- 컴퓨터는 여러 개의 부품으로 구성된다.
  - 컴퓨터를 더 이상 사용할 수 없게 되면 그 부품들도 다른 곳에 재사용 할 수 없게 된다.
- ① 상속 관계(inheritance)

② 연관 관계(association)

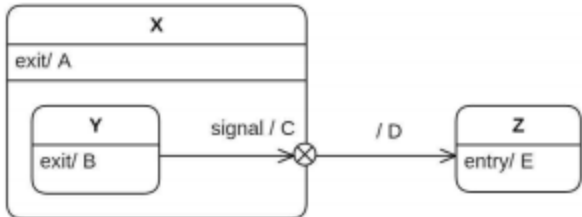
③ **합성 관계(composition)**

④ 의존 관계(dependency)

⑤ 집합 관계(aggregation)

⑥ 링크 관계(link)

21. 오른쪽 상태 다이어그램의 Y 상태에서 signal 이벤트를 받았을 때, Z 상태로 전이하면서 실행되는 액션의 순서로 가장 적절한 것은? ( )



- ① A, B, C, E, D

② **B, C, A, D, E**
- ③ C, B, A, E, D

④ D, C, A, B, E

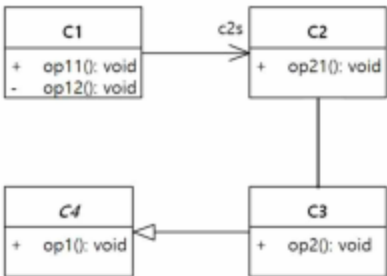
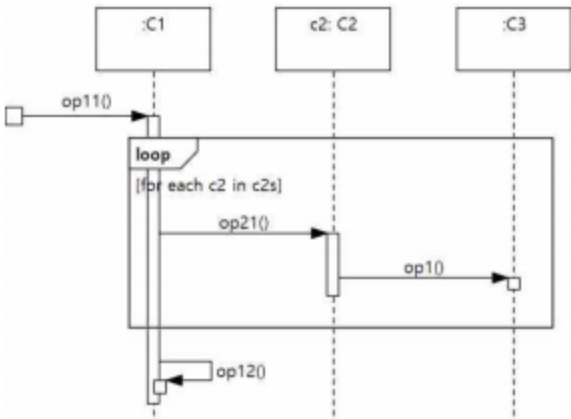
22. 다음은 시퀀스 다이어그램과 이와 일관성이 있는 클래스 다이어그램을 보여 준다. 시퀀스 다이어그램을 기준으로 할 때, 클래스 다이어그램에 존재하는 문제를 설명한 것으로 가장 적절한 것은? ( )

- ① 클래스 C3에 op1() 연산이 반드시 정의되어야 한다.

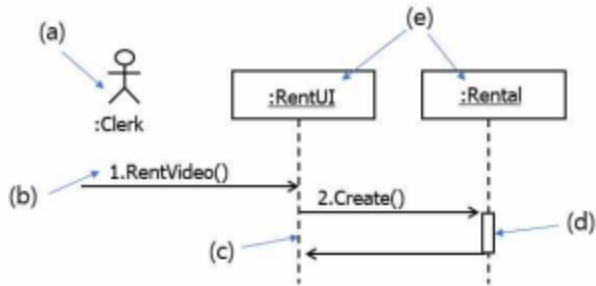
② 클래스 C2에서 클래스 C3로의 관계는 연관관계가 아니라 상속관계가 되어야 한다.

③ **클래스 C1에서 클래스 C2로의 연관관계는 C2 클래스 쪽에 '\*' 등의 다중성이 명시되어야 한다.**

④ 클래스 C1의 op12() 연산은 클래스 C1의 하위클래스에 정의되어야 한다.

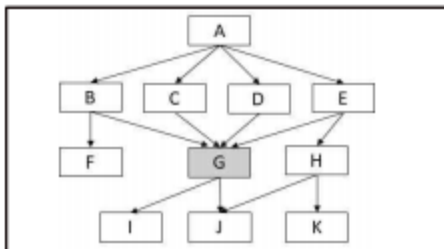


24. 다음 시퀀스 다이어그램에 표시한 내용에 대한 설명이 옳지 못한 것을 모두 고르시오. (                      )



- ① (a) - 시스템과 상호작용하는 액터
- ② (b) - 메시지의 호출
- ③ (c) - 객체의 생명주기를 나타내는 라이프라인
- ④ (d) - 객체의 소멸을 나타내는 제어박스
- ⑤ (e) - 시스템의 기능 단위를 나타내는 유스케이스

25. 모듈의 구조를 나타내는 다음과 같은 프로그램 구조도에서 모듈 G의 fan-in과 fan-out은 ?



fan-in = (    4    )

fan-out = (    2    )