

1. 소프트웨어의 특징으로 가장 적절하지 않은 것은?

- ① 많아 없어지지 않는다.
- ② 제조하는 것이 아니라 개발한다.
- ③ 개발 완료 후 시험이 어렵고 수정할 수 없다.
- ④ 유형의 매체에 저장되지만 개념적이고 무형적이다.

2. 소프트웨어공학의 정의로 가장 적절하지 않은 것은?

- ① 유일한 제품이나 서비스를 만들기 위해 수행되어야 할 일시적인 활동 과정이다.
- ② 소프트웨어의 개발과 운용, 유지보수에 대하여 체계적이며, 훈련된 계량화할 수 있는 접근 방식을 적용한 것이다.
- ③ 프로그램을 설계하고 개발하며, 운용, 유지보수에 관련된 문서를 작성하는 데 필요한 과학적인 지식을 실용화한 것이다.
- ④ 효율적으로 작동하는 신뢰성 있는 소프트웨어를 경제적으로 획득하기 위해 적절한 공학적 원리를 수립하여 활용하는 것이다.

3. 다음 소프트웨어 개발 프로세스 모델에 대한 설명으로 적절한 것을 있는 대로 모두 고른 것은?

- 가. 소프트웨어 개발 프로세스 모델은 프로젝트에 대한 전체적인 기본 골격을 세워준다.
- 나. 소프트웨어 개발 프로세스는 소프트웨어를 개발할 때 일을 수행하는 작은 단위인 작업의 집합을 이야기한다.
- 다. 소프트웨어 개발 프로세스 모델은 소프트웨어 개발 프로세스에서 소프트웨어를 어떻게 개발할 것인가에 대한 전체 흐름을 체계화한 개념으로 프로그램 설계라고 한다.
- 라. 소프트웨어 개발 프로세스 모델은 넓은 의미에서 절차나 과정뿐만 아니라 작업을 수행하는 데 필요한 방법과 도구를 포함하지만 인적 자원에 대해서는 포함하지 않는다.
- 마. 소프트웨어 개발 프로세스는 수많은 반복과 시행착오를 통해서 얻은 절차나 방법 등에 관한 지식을 같은 작업을 수행하는 다른 사람들에게 전달함으로써 시행착오를 줄이고 빠르게 적용할 수 있는 가이드 역할을 한다.

- ① 가, 나, 다
- ② 가, 나, 마
- ③ 나, 다, 라
- ④ 다, 라, 마

4. 다음 설명에 해당하는 소프트웨어 개발 프로세스 모델의 단계로 가장 적절한 것은?

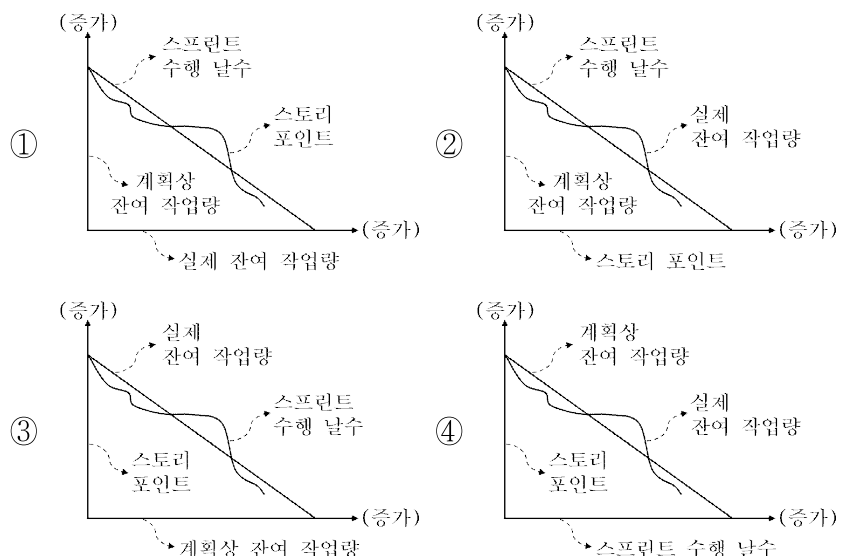
Grady Booch, Ivar Jacobson, James Rumbaugh가 각자 제안한 객체지향 분석/설계 방법의 중요한 특징을 결합하여 제안하였다. 반복/점증적 특징을 갖는 프로세스로, 4개의 단계와 10개의 작업 흐름으로 구성된다. 단계는 시간의 경과에 따른 시스템의 진화를 묘사한다. 시스템 개발은 4개의 단계를 반복하며, 각 반복에서 여러 작업을 수행한다.

- ① 구현 → 완성 → 문제 식별
- ② 시작 → 구체화 → 구축 → 전이
- ③ 요구 분석 → 위험 분석 → 개발 → 평가
- ④ 요구 분석 → 구조 설계 → 모듈 설계 → 구현

5. 애자일(agile) 모델에 대한 설명으로 가장 적절하지 않은 것은?

- ① 변경에 대한 대응보다 계획 준수가 더욱 중요하다.
- ② 문서화보다 동작하는 소프트웨어가 더욱 중요하다.
- ③ 프로세스나 도구보다 개인과 상호작용이 더욱 중요하다.
- ④ 계약 협상보다 고객과의 지속적인 협력이 더욱 중요하다.

6. 스크럼(scrum) 방식에서 사용하는 소멸 차트(burndown chart)로 가장 적절한 것은?



7. DevOps 프로세스 단계와 사용 가능한 자동화 도구를 가장 적절하게 나열하지 않은 것은?

	단계	도구
①	빌드	Puppet, Fortify
②	테스트	Junit, Selenium
③	릴리즈	Microsoft Azure, AWS
④	모니터링 및 피드백	Elastic, Splunk

8. 다음 설명에 해당하는 SOLID 원칙으로 가장 적절한 것은?

- 하위 클래스를 만들 때, 상위 클래스의 기능을 교체하는 것이 아닌 기능을 유지하면서 확장할 수 있어야 한다.
 - 자식 타입은 언제나 부모 타입을 대체할 수 있어야 한다. 이 원칙을 지키지 않으면 Refused Bequest라는 나쁜 냄새(bad smell)가 생긴다.
 - 상위 클래스 객체를 하위 클래스 객체로 교체하여도 프로그램의 정확성, 수행하는 업무 등과 같은 프로그램 속성에 대한 변경이 없어야 한다.

- ① OCP(Open-Closed Principle)
- ② LSP(Liskov Substitution Principle)
- ③ ISP(Interface Segregation Principle)
- ④ DIP(Dependency Inversion Principle)

9. 다음 설명에 해당하는 주요 활동으로 가장 적절하지 않은 것은?

위험(risk)이란 프로젝트에 심각한 영향을 미쳐 프로젝트 성패에 직접적인 영향을 미칠 수 있는 사건을 의미한다. 프로젝트를 성공적으로 완수하기 위해 미래에 발생할 수 있는 불확실한 위험 요소를 미리 파악하고, 발생 가능성을 분석하여, 이를 해결하기 위한 전략을 수립해야 한다.

- ① 위험 식별
- ② 위험 분석
- ③ 위험 제거
- ④ 위험 감시

10. ISO/IEC 25019 사용자 품질 모델에 대한 특성으로 가장 적절하지 않은 것은?

- ① ISO/IEC 25019는 소프트웨어 자체에 대한 품질평가보다 사용자가 소프트웨어를 사용하는 동안의 경험에 대한 품질평가를 위한 것이다.
- ② 효율성(efficiency)은 유용성, 신뢰성, 쾌적성, 안정성 기반으로 설정된 목표와 제약조건에 대한 효과적인 달성 여부를 평가한다.
- ③ 맥락 포괄성(context coverage)은 다른 상황이나 환경에서 얼마나 잘 적응하고 대응하는지, 유연하게 확장이나 변경할 수 있는지에 대하여 평가한다.
- ④ 위험 회피성(freedom from risk)은 소프트웨어가 예측 가능한지, 소프트웨어 사용으로 인한 사용자의 건강과 안전, 경제적, 환경적 위험을 완화하는가에 대해 평가한다.

11. 소프트웨어 품질 측정 척도에 대한 설명으로 가장 적절하지 않은 것은?

- ① MTTR(Mean Time To Repair)가 높을수록 정확성이 좋다.
- ② MTTF(Mean Time To Failure)가 낮을수록 가용성이 좋다.
- ③ MTTC(Mean Time To Change)가 낮을수록 유지보수성이 좋다.
- ④ MTBF(Mean Time Between Failure)가 높을수록 신뢰성이 좋다.

12. 다음 설명의 괄호(㉠, ㉡)에 들어갈 용어를 가장 적절하게 나열한 것은?

(㉠)의 주요 활동 중 수립한 (㉡)은/는 정형화한 절차로만 변경할 수 있는 명세 또는 결과물이다. 이는 소프트웨어 개발 과정에서 변경 요청을 공식적으로 검토하고 승인한 시점 이후 개발의 기초로 이용한다.

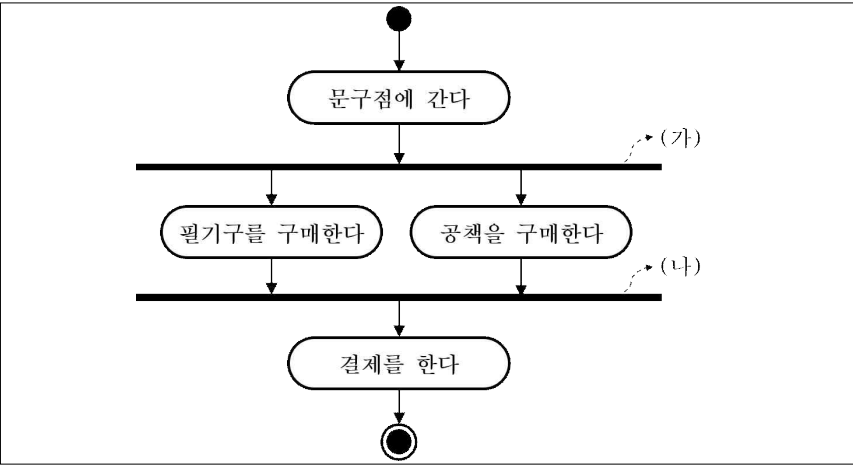
	㉠	㉡
①	유지보수	기준선
②	유지보수	변경 제어
③	형상관리	기준선
④	형상관리	변경 제어

13. 다음 의존 관계와 소요 기간을 기반으로 작업별 여유 시간(slack time) 계산 요소에 대한 설명으로 가장 적절하지 않은 것은?

작업명	시작	가	나	다	라	마	종료
선행작업	-	시작	시작	가, 나	다, 마	나	라
소요시간	2주	4주	1주	2주	1주	7주	1주

- ① ‘가’ 작업의 EST는 ‘나’ 작업의 LST와 같다.
- ② ‘다’ 작업의 EFT는 ‘가’ 작업의 LFT와 같다.
- ③ ‘라’ 작업의 EST는 ‘마’ 작업의 LST와 같다.
- ④ ‘마’ 작업의 EFT는 ‘다’ 작업의 LFT와 같다.

14. 다음 활동(activity) 다이어그램에서 (가)와 (나)에 해당하는 노드를 가장 적절하게 나열한 것은?



- | | | |
|---|-----|-----|
| | (가) | (나) |
| ① | 조인 | 병합 |
| ② | 조인 | 포크 |
| ③ | 포크 | 병합 |
| ④ | 포크 | 조인 |

15. 소프트웨어 개발 비용 계산을 위해 사용하는 델파이(Delphi) 기법에 대한 설명으로 가장 적절하지 않은 것은?

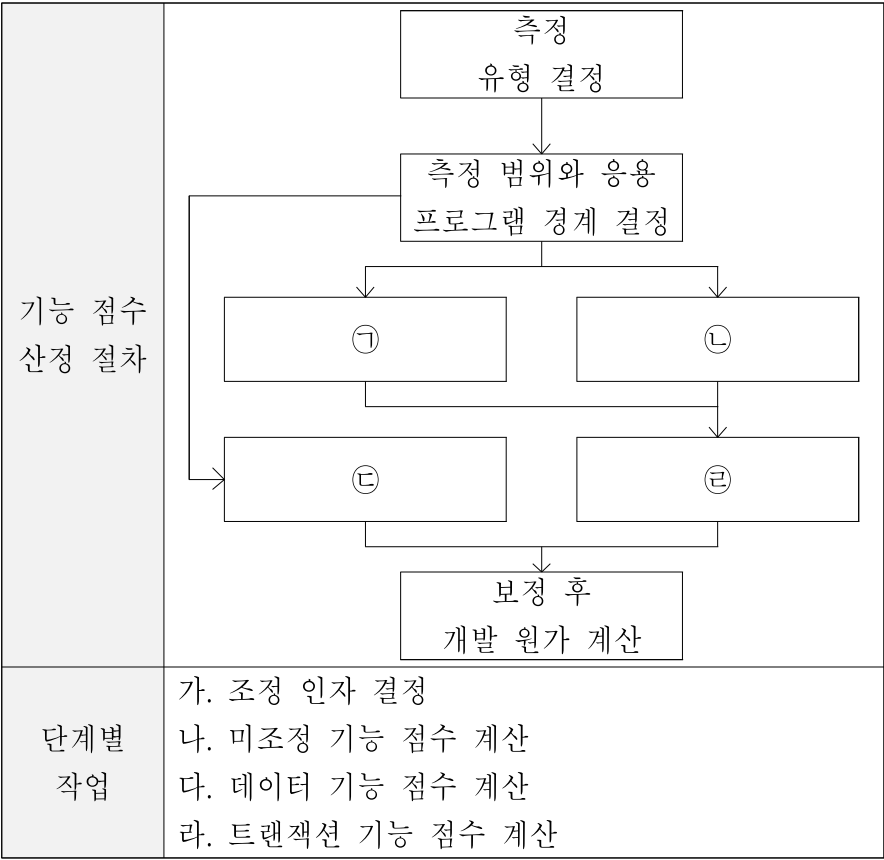
- ① 과거 유사 프로젝트 수행 경험을 바탕으로 개발 비용을 추정한다.
- ② 전문가의 주관적인 판단으로 실제 개발 비용과 편차가 클 수 있다.
- ③ 전문가 의견을 조정자가 수집하고 이를 조정하는 상향식 방법이다.
- ④ 전문가들의 편견이나 분위기에 끌려가지 않도록 조정자가 필요하다.

16. 다음에 대한 MM(Man Month)으로 가장 적절한 것은?

- 프로젝트 복잡도 계수는 1.3이다.
- 유지보수 인력은 전체의 20%를 추가한다.
- 개발자 1인당 평균 생산성은 800 LOC/MM이다.
- 소프트웨어 예상 LOC(Line Of Code)는 120,000이다.

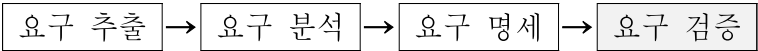
- ① 39 ② 162 ③ 195 ④ 234

17. 다음 기능 점수 산정 절차에 들어갈 단계를 가장 적절하게 나열한 것은?



- | | | | | |
|---|---|---|---|---|
| | ㉠ | ㉢ | ㉡ | ㉣ |
| ① | 나 | 가 | 라 | 다 |
| ② | 나 | 라 | 가 | 다 |
| ③ | 다 | 라 | 가 | 나 |
| ④ | 다 | 가 | 라 | 나 |

18. 다음 요구 공학 프로세스의 요구 검증 단계에서 검증하는 항목에 대한 설명으로 가장 적절하지 않은 것은?



- ① 검증 가능성(verifiability)은 모든 요구사항이 구현 가능한지 검증해야 한다.
- ② 일관성(consistency)은 모든 요구사항은 모순되지 않고 하나의 해석만 가능해야 한다.
- ③ 완전성(completeness)은 사용자의 모든 요구사항을 포함하고 있는지를 확인해야 한다.
- ④ 추적 가능성(traceability)은 설계문서 및 기능과 요구사항의 관계가 추적 가능해야 한다.

19. 다음 구조적 분석기법의 자료 흐름도에서 사용하는 기호에 대한 의미를 가장 적절하게 나열한 것은?

(가)	(나)	(다)
		

- | | (가) | (나) | (다) |
|---|--------|--------|--------|
| ① | 프로세스 | 외부엔티티 | 데이터저장소 |
| ② | 프로세스 | 데이터저장소 | 외부엔티티 |
| ③ | 데이터저장소 | 외부엔티티 | 프로세스 |
| ④ | 외부엔티티 | 데이터저장소 | 프로세스 |

20. 구조적 분석기법 요소에 대한 설명으로 가장 적절하지 않은 것은?

- ① 자료사전은 자료 흐름도에 존재하는 데이터에 대한 자세한 설명을 기록한 것이다.
- ② 배경도는 하나의 프로세스로 구성되며, 개발하게 될 시스템의 명칭을 부여한다.
- ③ 소단위 명세서는 자료 흐름도에서 프로세스의 실제 알고리즘 형태를 작성한다.
- ④ E-R 다이어그램은 사용사례에 대하여 도식화하여 액터에 대한 설명을 작성한다.

21. 다음 중 비기능적 요구사항만 있는 대로 모두 고른 것은?

- 가. 보안 요구사항을 만족해야 한다.

나. 가용성을 최소 99% 이상 보장한다.

다. 파일 열기는 3초 이내에 완수해야 한다.

라. 사용자 입력값은 1부터 99까지 만으로 제한한다.

마. 텍스트(txt) 형식으로 문서를 저장할 수 있어야 한다.

바. 시스템 장애로 인한 서비스 불가 시간이 연 10시간을 넘지 않아야 한다.

- ① 가, 나, 다, 라
- ② 가, 나, 다, 바
- ③ 나, 다, 마, 바
- ④ 다, 라, 마, 바

22. 다음 밑줄 친 자바 소스 코드를 다형성(polymorphism)을 위한 메서드 오버로딩(method overloading)과 메서드 오버라이딩(overriding)에 따라 가장 적절하게 짝 지은 것은?

```
class Shape{
...(중략)...
    ㉠public int area(int len) { ...(중략)... }
    ㉡public int area(int width, int height) { ...(중략)... }
...(중략)...
}

class Circle extends Shape{
...(중략)...
    ㉢public int area(int radius) { ...(중략)... }
...(중략)...
}
```

- | | <u>오버로딩</u> | <u>오버라이딩</u> |
|---|-------------|--------------|
| ① | ㉠, ㉡ | ㉠, ㉢ |
| ② | ㉠, ㉡ | ㉡, ㉢ |
| ③ | ㉠, ㉢ | ㉡, ㉢ |
| ④ | ㉠, ㉢ | ㉠, ㉡ |

23. 다음에서 설명하는 아키텍처 스타일로 가장 적절한 것은?

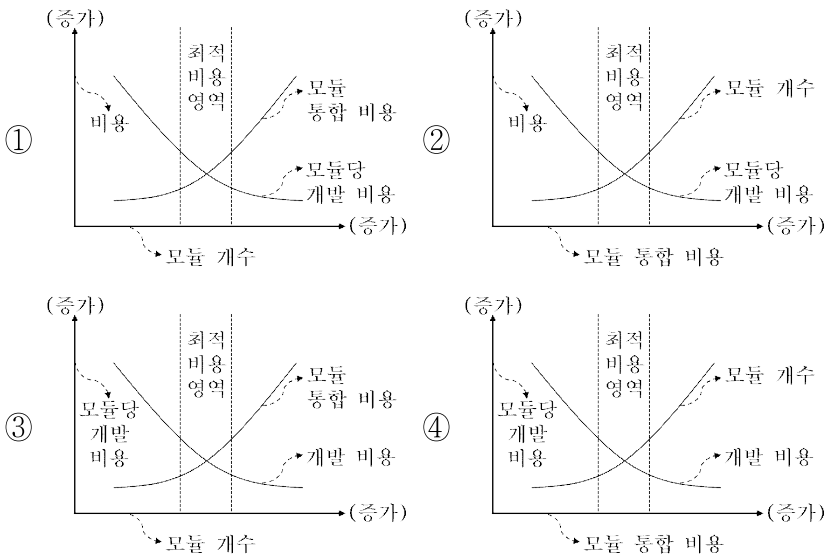
사용자 인터페이스로부터 비즈니스 로직을 분리하여 애플리케이션의 시각적 요소나 그 이면에서 실행하는 비즈니스 로직을 서로 영향을 주지 않고 쉽게 수정 가능한 스타일의 아키텍처이다. 모델, 뷰, 제어 세 부분으로 이루어진 구조로 되어 있다.

- ① 계층 스타일
- ② MVC 스타일
- ③ 데이터 흐름 스타일
- ④ 클라이언트 - 서버 스타일

24. 파이프 필터 소프트웨어 아키텍처에 대한 설명으로 가장 적절한 것은?

- ① 모든 필터가 동시에 작동하는 병렬처리 형식으로 처리 속도가 향상된다.
- ② 서비스 크기를 작게 나눈 아키텍처로 서비스를 독립적으로 개발하고 확장할 수 있다.
- ③ 서브 시스템이 입력 데이터를 받아 처리하고 결과를 다른 시스템에 보내는 작업이 반복된다.
- ④ 소프트웨어 기능을 수직으로 상호작용하는 여러 층으로 분리한 구조로 각 층 사이에 상호작용이 필요한 시스템에 가장 적합하다.

25. 최적 소프트웨어 개발 비용에 관한 그래프로 가장 적절한 것은?



26. UML 4+1뷰를 이용한 시스템 표현 시, 시스템의 병렬 처리나 동기화 처리 등 시스템의 런타임(runtime) 동작에 초점을 맞춘 관점으로 가장 적절한 것은?

- ① 논리 관점(Logical view)
- ② 프로세스 관점(Process view)
- ③ 유스케이스 관점(Usecase view)
- ④ 구현 관점(Implementation view)

27. 응집도와 결합도 정렬 방향에 따라 괄호(㉠~㉤)에 들어갈 소프트웨어 품질을 가장 적절하게 나열한 것은? (단, 결합도와 응집도는 모두 정렬되어 있다)

(㉠) ←	결합도	→ (㉡)
데이터 결합, ...(생략)... ,외부 결합, ...(생략)... , 내용 결합		
(㉢) ←	응집도	→ (㉣)
기능적 응집, ...(생략)... , 절차적 응집, ...(생략)... , 우연적 응집		

- | | | | | |
|---|----|----|----|----|
| | ㉠ | ㉡ | ㉢ | ㉣ |
| ① | 나쁨 | 좋음 | 나쁨 | 좋음 |
| ② | 나쁨 | 좋음 | 좋음 | 나쁨 |
| ③ | 좋음 | 나쁨 | 나쁨 | 좋음 |
| ④ | 좋음 | 나쁨 | 좋음 | 나쁨 |

28. 다음 C 소스 코드의 두 함수 사이의 결합도와 funcB()의 응집도를 가장 적절하게 나열한 것은?

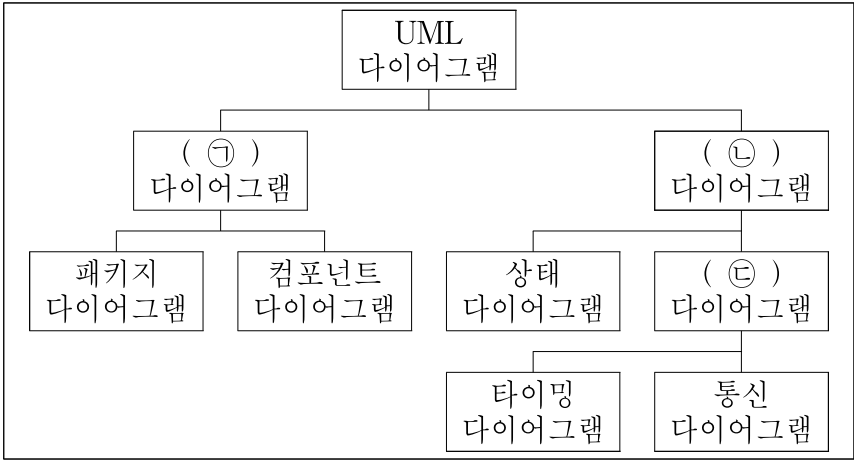
```

void funcA() {
    int x, y, res, flag;
    scanf("%d %d %d\n", &x, &y, &flag);
    res = funcB(x, y, flag);
    printf("%d\n", res);
}

int funcB(int a, int b, int f) {
    if(f == 1) return (a * b);
    else return (a / b);
}
    
```

- ① 제어 결합, 논리적 응집
- ② 제어 결합, 기능적 응집
- ③ 공통 결합, 논리적 응집
- ④ 공통 결합, 기능적 응집

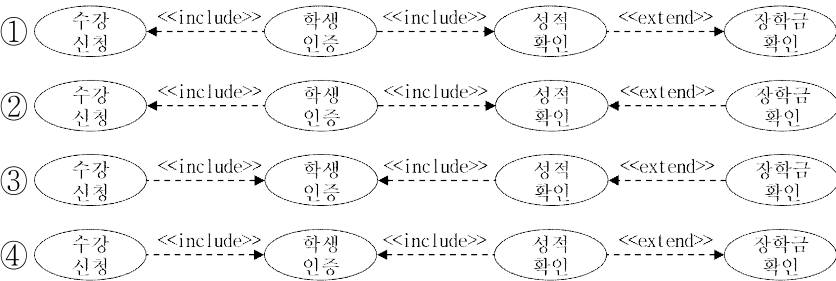
29. 다음 분류의 괄호(㉠~㉤)에 들어갈 용어를 가장 적절하게 나열한 것은?



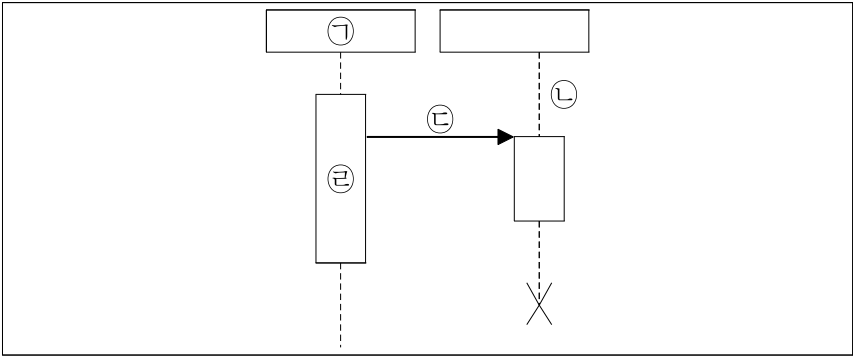
- | | | | |
|---|----|------|------|
| | ㉠ | ㉡ | ㉢ |
| ① | 동작 | 구조 | 상호작용 |
| ② | 동작 | 상호작용 | 구조 |
| ③ | 구조 | 상호작용 | 동작 |
| ④ | 구조 | 동작 | 상호작용 |

30. 다음 설명에 관한 사용사례 다이어그램(usecase diagram)으로 가장 적절한 것은?

학생은 수강 신청과 성적 확인을 위해 학생 인증 과정이 필요하다. 또한, 성적이 높은 경우 장학금 확인이 가능하다.



31. 다음 순서(sequence) 다이어그램의 각 요소(㉠~㉤)에 대한 용어를 가장 적절하게 나열한 것은?



- | | | | | |
|---|--------|-------|--------|-------|
| | ㉠ | ㉡ | ㉢ | ㉣ |
| ① | 객체 | 생명선 | 메서드 호출 | 활동 막대 |
| ② | 객체 | 활동 막대 | 메서드 호출 | 생명선 |
| ③ | 메서드 호출 | 활동 막대 | 객체 | 생명선 |
| ④ | 메서드 호출 | 생명선 | 객체 | 활동 막대 |

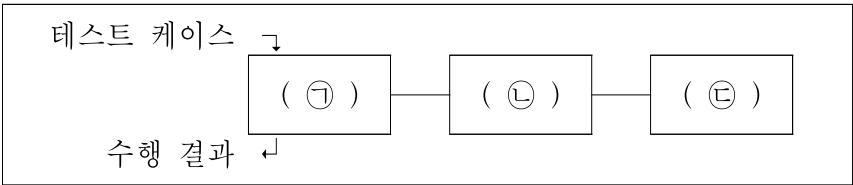
32. 다음 자바 소스 코드에 적용한 디자인 패턴으로 가장 적절한 것은?

```

abstract class Shape{ ...(중략)... }
class Circle extends Shape {
...(중략)...
public:
    Circle(Circle src) { super(src);
        this.radius = src.getRadius(); ...(중략)... }
    Shape Clone() { return new Circle(this); }
    int getRadius() { return radius; }
private:
    int radius;
...(중략)...
}
    
```

- ① 어댑터(adapter)
 ② 싱글톤(singleton)
 ③ 프로토타입(prototype)
 ④ 플라이웨이트(flyweight)

33. 다음 단위 테스트 환경의 괄호(㉠~㉣)에 들어갈 용어로 가장 적절한 것은?



- | | | | |
|---|------|--------|--------|
| | ㉠ | ㉡ | ㉢ |
| ① | 스터브 | 테스트 모듈 | 드라이버 |
| ② | 스터브 | 드라이버 | 테스트 모듈 |
| ③ | 드라이버 | 테스트 모듈 | 스터브 |
| ④ | 드라이버 | 스터브 | 테스트 모듈 |

34. 다음 소스 코드에서 조건 커버리지(coverage)를 만족하지만, 분기 커버리지는 만족하지 않는 테스트데이터가 인자인 함수 호출을 가장 적절하게 나열한 것은? (단, 단축 연산은 적용하지 않는다)

소스 코드	<pre> void func(int x, int y) { if((x > 10) && (y != 10)) { ...(중략)... } } </pre>
함수 호출	가. func(9, 9); 나. func(10, 10); 다. func(11, 10); 라. func(11, 11);

- ① 가, 나
 ② 나, 다
 ③ 다, 라
 ④ 가, 다

35. 다음에서 정적 테스트 기법의 비정형 방법을 있는 대로 모두 고른 것은?

가. 개별 검토	나. 동료 검토
다. 검토 회의	라. 기호 계산
마. 모형 검증	바. 감사(auditing)

- ① 가, 나, 다, 라
 ② 가, 나, 다, 바
 ③ 가, 라, 마, 바
 ④ 나, 다, 라, 마

36. 다음 테스트 방법에 대한 명칭을 가장 적절하게 나열한 것은?

가. 개발 소프트웨어 시스템을 사용자에게 전달하기 전에 수행하는 테스트로 구축된 시스템에 각 요구사항이 올바르게 구현되었는지를 점검하는 데모 형식으로 테스트를 진행한다.
 나. 변경한 부분을 중점적으로 테스트하기 위하여 변경에 영향을 받는 범위에 대한 테스트 케이스를 마련하여 테스트를 수행한다. 테스트 수행 범위를 최소화하기 때문에 비용 절감에 효과적이거나 완전한 테스트가 이루어지지 않는다는 단점이 있다.

	가	나
①	인수 테스트	선택적 회귀 테스트
②	인수 테스트	우선순위 회귀 테스트
③	통합 테스트	선택적 회귀 테스트
④	통합 테스트	우선순위 회귀 테스트

37. 다음의 보기에서 설명하는 테스트로 가장 적절한 것은?

평소보다 많은 비정상적인 양, 값, 빈도, 부피 등으로 부하를 발생시켜 부하가 최고치인 상황에서 시스템의 반응을 살피고 이때 발생하는 오류를 찾는 방법이다.

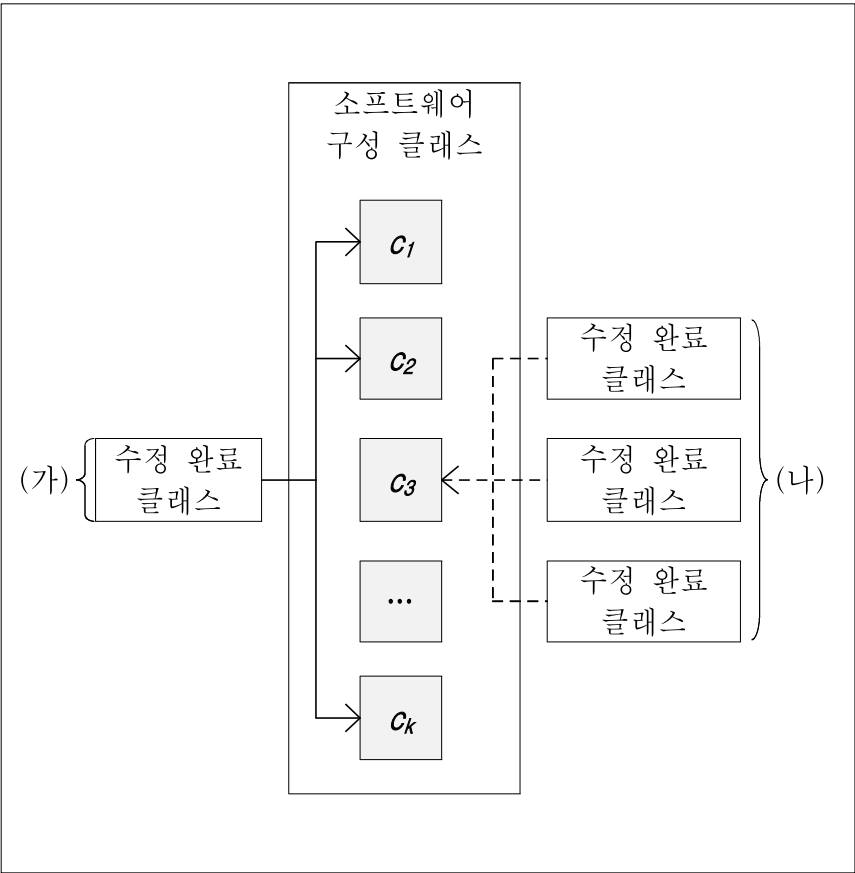
- ① 성능 테스트
- ② 보안 테스트
- ③ 안정성 테스트
- ④ 스트레스 테스트

38. 다음 유지보수에 해당하는 유형으로 가장 적절하지 않은 것은?

- 현재 운영 중인 중고차 관리시스템에 중고차를 판매할 수 있는 기능을 추가한다.
- 현재 운영 중인 중고차 관리시스템에서 세글자를 초과하는 판매자 이름에 대해 화면 출력이 불가하고 프린터 출력 시 동작이 멈추는 문제가 발생하여 이를 보수한다.
- 현재 운영 중인 중고차 관리시스템은 윈도우즈 시스템에서 동작하고 있으나, 리눅스 시스템에서도 동작하도록 보수한다.

- ① 적응 유지보수(adaptive maintenance)
- ② 완전 유지보수(perfective maintenance)
- ③ 수정 유지보수(corrective maintenance)
- ④ 예방 유지보수(preventive maintenance)

39. 다음 그림은 변경요청에 따른 수정이 필요한 클래스의 관계를 보여준다. 이런 형태의 코드 스멜(code smell) 용어를 올바르게 연결한 것으로 가장 적절한 것은? (단, 화살표는 클래스의 수정 완료에 따라 추가변경이 발생하는 클래스를 지칭한다)



- | | (가) | (나) |
|---|-----------------|------------------|
| ① | Middle Man | Message Chains |
| ② | Middle Man | Divergent Change |
| ③ | Shotgun Surgery | Message Chains |
| ④ | Shotgun Surgery | Divergent Change |

40. 프로세스 성숙도를 평가하기 위한 CMMI와 SPICE에 대한 설명 중 가장 적절하지 않은 것은?

- ① CMMI-DEV의 주요 프로세스 영역은 프로세스 관리, 프로젝트 관리, 엔지니어링, 지원으로 구성된다.
- ② SPICE 모델에서 표준으로 정의된 프로세스를 사용하여 계획하고 관리하는 단계는 관리(managed) 단계이다.
- ③ CMMI와 SPICE의 중요한 차이점은 성숙도 레벨과 심사영역의 구분에 있으며, 두 모델의 성숙도의 최고 수준은 모두 레벨 5이다.
- ④ CMMI 모델에서 표준화 프로젝트 프로세스는 존재하나 프로젝트 목표와 활동에 대한 정량적 측정이 불가능한 단계는 정의(defined) 단계이다.