Chapter 1. 연습문제

1. 컴퓨터의 발전과정에서 소프트웨어의 개발속도가 하드웨어의 발전 속도를 따라가지 못해 사용자들의 요구사항을 감당할 수 없는 문제가 발생함을 의미하는 것은? (1)

   ① 소프트웨어 위기(Crisis)　　② 소프트웨어 오류(Error)
   ③ 소프트웨어 버그(Bug)　　④ 소프트웨어 유지보수(Maintenance)

2. 소프트웨어에 대한 잘못된 통념에 해당되지 않는 것은? (4)
   ① 소프트웨어는 유연하기 때문에 요구사항의 변경을 소프트웨어에 반영하는 것은 용이하다.
   ② 소프트웨어에 대한 소스 코드를 작성해서 실행을 시키기 전까지는 소프트웨어의 품질을 평가할 수는 없다.
   ③ 소프트웨어 공학은 별로 필요 없는 덩치 큰 문서를 생성하게 하여 개발을 매우 지연되게 만든다.
   ④ 지체된 프로젝트에 일정을 뒤늦게 투입하는 것은 오히려 프로젝트의 일정을 더욱 지연시킬 수가 있다.

3. 소프트웨어 공학에 대한 설명으로 옳지 않은 것은? (3)
   ① 공학적인 지식을 소프트웨어 설계와 제작에 응용하고, 이를 개발, 운영 및 유지보수하는 데 필요한 문서화과정이다.
   ② 비교적 큰 규모의 소프트웨어 시스템을 대상으로 한다.
   ③ 개개의 프로그래머에 의해 소프트웨어 시스템을 공학적인 원리로 개발하는 것을 의미한다.
   ④ 소프트웨어 제품의 체계적 생산과 유지보수에 관련된 기술적 관리적 원리이다.
   ⑤ 시스템 개발에 있어서 기술적인 사항과 비기술적인 사항을 통합한다.

4. 소프트웨어 공학에 대한 설명으로 틀린 것은? (2)
   ① 소프트웨어 위기를 극복하기 위한 방안을 제시한다.
   ② 소프트웨어 공학은 관리기술과 방법론보다는 기법, 도구를 더 중시한다.
   ③ 소프트웨어의 개발, 운용, 유지보수 및 폐기처분을 위한 제도적인 접근방안이다.
   ④ 컴퓨터 하드웨어에서 신뢰성 있게 운용되는 소프트웨어를 경제성 있게 개발하기 위해 공학적 원리를 응용하고 확립시킨 이론이다.

5. 소프트웨어의 특성으로 올바르게 설명한 것은? (4)
   ① 소프트웨어는 수정이 용이하고 사용자가 쉽게 고칠 수 있다.
   ② 하드웨어와 달리 소프트웨어는 마모되며 품질이 저하된다.
   ③ 소프트웨어는 보고 만질 수 없기 때문에 무형이라 말한다.

④ 소프트웨어는 사용자가 원하는 대로 작동하여야 한다.
⑤ 소프트웨어는 개발, 설계, 제조되지만 수학적 방정식이 성립되지는 않는다.

6. 소프트웨어 위기가 의미하는 것이 아닌 것은? (1)
① 정교하고 높은 기술력을 가지는 하드웨어의 개발로 극복할 수 있다.
② 소프트웨어 특성에 관한 이해 부족이 원인이 될 수 있다.
③ 다양한 소프트웨어 요구사항을 충족시키지 못하였다.
④ 체계적인 소프트웨어 공학의 발전이 필요하다.
⑤ 새로운 소프트웨어를 요구하는 시장의 수요가 급증하였다.

7. 소프트웨어 위기 현상을 올바르게 설명한 것은? (1)
① 새로운 소프트웨어를 요구하는 사람의 수요에 비해 공급이 부족하다.
② 소프트웨어 개발 기술의 속도가 하드웨어 개발 기술의 속도보다 빠르다.
③ 일정 지연 사태는 인력을 추가로 투입시키면 해결할 수 있다.
④ 소프트웨어 개발에 대한 강력한 도구와 자동화 기술이 많이 소개되었다.
⑤ 하드웨어 가격에 비해 소프트웨어 가격이 상대적으로 싸기 때문에 소프트웨어의 위기를 맞이하게 되었다.

8. 소프트웨어의 특징에 대한 설명으로 옳지 않은 것은? (3)
① 소프트웨어 생산물의 구조가 코드 안에 숨어 있다.
② 논리적 절차에 다라 개발된다.
③ 사용에 의해 마모되거나 소멸된다.
④ 요구나 환경에 따라 적절히 변형시킬 수 있다.

9. 소프트웨어 위기의 현상으로 보기 어려운 것은? (4)
① 프로젝트 개발 일정과 예산 측정의 어려움
② 소프트웨어 유지보수 비용의 증가
③ 소프트웨어 개발 적체 현상
④ 소프트웨어 개발 인력의 증가

10. 소프트웨어 위기를 가져온 원인에 해당하지 않는 것은? (4)
① 소프트웨어 규모 증대와 복잡도에 따른 개발 비용 증가
② 프로젝트 관리기술의 부재
③ 소프트웨어 개발기술에 대한 훈련 부족
④ 소프트웨어 수요의 감소

11. 소프트웨어 공학의 기본 원칙이라고 볼 수 없는 것은? (4)

① 현대적인 프로그래밍 기술 적용
② 지속적인 검증 시행
③ 결과에 대한 명확한 기록 유지
④ 충분한 인력 투입

12. 공학적으로 잘 작성된 소프트웨어의 특성이 아닌 것은? (3)
① 소프트웨어는 신뢰성이 높아야 하며 효율적이어야 한다.
② 소프트웨어는 사용자가 원하는 대로 동작해야 한다.
③ 소프트웨어는 편리성이나 유지보수성에 점차 비중을 적게 두는 경향이 있다.
④ 소프트웨어는 잠재적인 에러가 가능한 한 적어야 하며 유지보수가 용이해야 한다.

13. 소프트웨어 공학에 대한 가장 적절한 설명은? (3)
① 소프트웨어 위기를 완전히 해결한 공학적 원리이다.
② 신뢰성 있는 소프트웨어를 만들기 위한 도구만을 연구하는 학문이다.
③ 가장 경제적으로 신뢰도 높은 소프트웨어를 만들기 위한 방법, 도구와 절차들의 체계이다.
④ 점차 많은 비용이 소요되는 소프트웨어 개발에서 가장 경제적인 방법을 찾고자 하는 것이다.

14. 소프트웨어 공학이 나타나게 된 배경과 관계가 먼 것은? (2)
① 소프트웨어 비용의 증가
② 유지보수 비용의 감소
③ 소프트웨어 품질과 생산성의 재고
④ 특정 개인의 의존한 시스템 개발

15. 소프트웨어 공학에 대한 설명으로 거리가 먼 것은? (3)
① 소프트웨어 공학이란 소프트웨어의 개발, 운용, 유지보수 및 파기에 대한 체계적인 접근 방법이다.
② 소프트웨어 공학은 소프트웨어 제품의 품질을 향상시키고 소프트웨어 생산성과 작업 만족도를 증대시키는 것이 목적이다.
③ 소프트웨어 공학의 궁극적 목표는 최대의 비용으로 계획된 일정보다 가능한 빠른 시일 내에 소프트웨어를 개발하는 것이다.
④ 소프트웨어 공학은 신뢰성 있는 소프트웨어를 경제적인 비용으로 획득하기 위해 공학적 원리를 정립하고 이를 이용하는 학문이다.

Chapter 2. 연습문제

1. 다음과 같은 소프트웨어 시스템을 개발할 때, 적용할 수 있는 적합한 개발 모델은? (1)

전자교환기 소프트웨어 시스템은 지난 수십 년 간 개발되어 사용해왔으며, 새로운 기능이 추가되기보다는 다양한 하드웨어 플랫폼에 맞도록 최적화시키는 일이 빈번히 일어났다.

① 폭포수(waterfall) 모델
② 프로토타이핑(prototyping) 모델
③ 나선형(spiral) 모델
④ UP(Unified Process) 모델

2. 고객의 요구사항을 명확하게 파악하기 어렵고, 프로젝트의 실현가능성이 의문시되는 경우에 프로젝트 관리자가 적용할 수 있는 가장 적절한 소프트웨어 개발 모델은? (4)

① RAD(Rapid Application Development) 모델
② 나선형 모델
③ 폭포수 모델
④ 시제품화(Prototyping) 모델

3. 소프트웨어 프로토타이핑에 대한 설명으로 옳지 않은 것은? (2)
① 요구사항을 수집한 후 소프트웨어 프로토타입을 만들어서 요구사항을 효과적으로 유도, 수집한다.
② 프로토타이핑은 설계 단계에서 수행한다.
③ 빠른 프로토타이핑(rapid prototyping)을 위해서 4세대 기법이나 재사용 가능한 소프트웨어 구성요소를 이용한다.
④ 프로토타이핑에 의해 만들어진 프로토타입은 폐기될 수 있고, 재사용될 수도 있다.

4. 소프트웨어 개발 프로세스 중 프로토타입 모델에 대한 설명 중 옳지 않은 것은? (3)
① 개발자가 고객의 요구사항을 불완전하게 파악한 채로 프로젝트를 진행할 때 유용하게 사용되는 모델이다.
② 프로토타입 모델은 “요구사항 수집 -> 프로토타입 설계 -> 프로토터입 구현 -> 고객평가 -> 프로토타입 정제”의 순서로 단계가 진행된다.
③ 프로토타입 모델은 고객의 진정한 요구가 무엇인지를 결정하는 데 있으므로 프로토타입의 내부적 구조가 매우 중요하다.
④ 프로토타입 모델의 장점은 정확하고 품질 좋은 요구사항 명세를 작성할 수 있다는 것이다.

5. 다음의 설명에 해당하는 소프트웨어 생명주기 모델은? (1)

- 계획수립 – 위험분석 – 개발 – 고객평가의 4단계를 가짐.
- 프로토타입을 지속적으로 발전시켜 최종 소프트웨어 개발까지 이르는 개발방법으로 위험관리가 중심인 생명주기 모델임.
- 비용이 많이 들고 시간이 걸리는 시스템을 구축해나갈 때 가장 현실적인 접근 방법임.
- 성과를 보면서 조금씩 투자하여 위험부담을 줄일 수 있는 방법임.

① 나선형 모델 ② 폭포수 모델
③ 프로토타이핑 모델 ④ 컴포넌트 기반 모델

6. 나선형 모델이 다른 개발 모델에 비하여 가질 수 있는 가장 큰 장점은? (3)
① 빠른 시스템 개발이 가능하다.
② 시스템 개발 전문가에 대한 의존도를 상당히 낮출 수 있다.
③ 프로젝트 초기 단계에 위험 요소를 발견하고 제거할 수 있다.
④ 시스템의 재사용성을 크게 향상시킬 수 있다.

7. 나선형 모델에서 단계별로 수행하는 작업은? (2)
① 위험분석 – 계획 및 정의 – 개발 – 고객평가
② 계획 및 정의 – 위험분석 – 개발 – 고객평가
③ 계획 및 정의 – 개발 – 위험분석 – 고객평가
④ 위험분석 – 계획 및 정의 – 고객평가 – 개발

8. 소프트웨어 개발 프로세스 모델에 대한 설명 중 맞는 것은? (4)
① 폭포수 모델은 응용분야가 복잡하고 세부적 과정이 필요한 분야에 적합한 모델이다.
② 점증적 모델은 개발 제품을 릴리스할 때마다 기능의 완성도를 높이는 점증적 개발과 새로운 기능을 추가하는 반복적 방법을 병행하여 사용하기도 한다.
③ V 모델은 신뢰성을 높이기 위한 테스트 작업을 강조한 것이므로 문서와 결과물 도출에 중점을 두고 있다.
④ 나선형 모델의 진화단계는 계획수립, 위험분석, 개발, 평가이다.

9. 신속한 소프트웨어 개발(Rapid Software Development)에 관한 설명이 옳지 않은 것은? (2)
① Agile 방법은 개발 오버헤드를 줄임으로써 소프트웨어를 신속히 생산하도록 하는 반복적 개발 방법이다.
② 실행 가능한, 쓰고 버리는 프로토타입의 장점은 프로토타입의 사용 양식이 최종적으로 인도되는 시스템의 사용 방식과 일치한다는 점이다.
③ RAD(Rapid Application Development) 환경은 데이터베이스 프로그래밍 언어, 폼 생성 도구, 사무 응용으로의 연결 기능을 포함한다.
④ XP(eXtreme Programming)는 시험 우선 개발, 리팩토링 및 고객 참여와 같은 사항을 포함

한다.

10. 다음의 특징에 맞는 개발 접근 방식은? (4)

가. 고객이 개발의 우선순위를 결정 나. 최소한의 문서화 다. 숙련된 개발자들이 빠르게 개발 라. 빠르게 개발하고 단위 시험으로 검증하는 것이 핵심

① 정보공학 개발 ② 객체지향 개발
③ 컴포넌트 기반 개발 ④ Agile Programming

11. UP(Unified Process) 개발공정에 대한 설명으로 틀린 것은? (3)
① 개념(Inception) 단계 – 범위 설정 및 타당성 검토
② 전개(Elaboration) 단계 – 상세설계
③ 구축(Construction) 단계 – 운영되는 프로그램
④ 전환(Transition) 단계 – 사용자 환경변화에 따른 적응 실시스템

12. UP(Unified Process)는 대표적인 객체지향 및 컴포넌트 기반 개발 방법론이다. UP에 대한 설명으로 가장 거리가 먼 것은? (3)
① UP는 도입(inception), 정련(elaboration), 구축(construction), 전이(transition)의 4단계로 시스템 개발 과정을 구분한다.
② UP는 반복(iteration)을 함으로써 사용자 피드백을 통하여 요구사항을 만족시키는데 효과적이다.
③ 정련 단계에서는 도입 단계에서 정의된 아키텍처를 바탕으로 시스템의 세부 기능을 구현하는데 초점을 둔다.
④ UP는 점증적(incremental) 접근 방법으로 사용한다. 즉, 개발과정이 반복적으로 수행되면서 시스템의 구현 기능이 점증적으로 추가된다.

13. RUP(Rational Unified Process) 과정에서 프로젝트 계획, 시스템을 위한 아키텍처 프레임워크 확립, 문제영역의 이해 등이 완결된 이후 이행해야 하는 단계는? (3)

① 도입 단계(inception phase) ② 정련 단계(elaboration phase)
③ 구축 단계(construction phase) ④ 전이 단계(transition phase)

14. 다음은 UP(Unified Process)의 어떤 단계를 설명한 것인가? (2)

시스템 아키텍처 확립을 위한 설계관련 작업의 비중이 크게 나타나고 프로젝트 계획이 상세히 작성된다.

① 도입 단계(inception phase)
② 정련 단계(elaboration phase)
③ 구축 단계(construction phase)
④ 전이 단계(transition phase)

15. 객체지향 개발 프로세스인 RUP(Rational Unified Process)에 대한 설명으로 옳지 않은 것은? (2)
① 반복적 개발 프로세스이다.
② 비즈니스 관점보다는 기술적인 관점에 초점을 맞춘다.
③ 고객의 요구사항을 명확하게 문서화하고 그 요구사항의 변경을 추적한다.
④ 구축(construction) 단계에서도 요구사항을 기술한다.

16. RUP(Rational Unified Process)의 특징을 잘 설명한 것은? (3)

A. RUP는 도입(Inception), 정련(Elaboration), 구축(Construction), 전이(Transition)의 순서로 진행된다. B. RUP는 개발기간이 짧은 프로젝트에 적용하는 것이 효과적이다. C. RUP는 Rational Software 사에서 개발한 기능 중심 프로세스이다. D. RUP는 소프트웨어 개발공정을 반복적인 사이클로 진행한다.

① A, C
② B, C
③ A, D
④ B, D

17. RUP(Rational Unified Process)에서 프로세스를 나타내는 관점이 아닌 것은? (4)
① 시간에 따라 변하는 모델을 나타내는 동적 관점
② 정해진 대로 행동하는 프로세스 활동을 나타내는 정적 관점
③ 프로세스 동안에 사용되는 실무 관행을 제시하는 실무 관점
④ 프로세스 진행 중 이해당사자(stakeholder)들의 조율을 위한 중재 관점

18. 프로젝트 수행 시 발생하는 위험을 관리하고 최소화하는 것이 주된 목적인 소프트웨어 프로세스 모델은? (3)
① 폭포수 모델
② 4세대 기법(4th Generation Technique)
③ 나선형 모델
④ 구축 및 수정 모델(Build-Fix Model)

19. 다음 설명은 개발 방법론 중에서 나선형 모델을 나타내고 있다. 다음 중 나선형 모델의 각 단계별 작업내용을 순서대로 나열한 것으로 가장 적절한 것은? (1)

나선형 모델은 비선형적이며 반복적으로 개발이 진행되므로, 소프트웨어 품질 중 강인성을 높일 수 있는 방법이다. 특히 이 방법은 개발자나 사용자가 각 확장 단계에서 발생할 수 있는 위험에 대한 이해와 대책이 가능하다. 따라서 프로젝트가 실패할 위험을 사전에 최소화할 수 있다.

① 계획수립 - 위험분석 - 개발 - 평가 ② 위험분석 - 계획수립 - 개발 - 평가
③ 계획수립 - 개발 - 위험분석 - 평가 ④ 계획수립 - 위험분석 - 평가 - 개발

20. 시스템 개발주기(SDLC)에 대한 설명 중 틀린 것은? (4)
① 각종 문서는 언제까지 작성되어야 하고 언제 검토할 것인지를 개발자들에게 알려준다.
② 사용자와 개발자 사이의 의사소통을 원활하게 한다.
③ 용어의 표준화가 가능해진다.
④ 사용자를 위한 문제 해결 시스템에 초점을 맞추기보다는 소프트웨어 개발에만 치중해야 한다.

21. 다음 중 소프트웨어 시스템 개발 생명주기 순서로 올바른 것은? (2)
① 요구사항 분석 ->시스템 정의 -> 설계 -> 코딩
② 시스템 정의 -> 요구사항 분석 -> 설계 -> 코딩
③ 시스템 정의 -> 설계 -> 분석 -> 코딩
④ 요구사항 분석 -> 설계 -> 코딩 -> 시스템 정의

22. 소프트웨어 생명주기 중 요구분석 단계에서 이루어지는 것은? (1)

① 기능 정의 ② 일정 계획
③ 비용 산정 ④ 모듈 정의
⑤ 화이트 박스

23. 소프트웨어 생명주기 모델에 관한 설명 중 가장 적절한 것은? (3)
① 폭포수형 모델은 사용자 또는 고객과의 상호작용을 강조하는 모델이다.
② 점증적 모델에서 필수적인 프로토타입은 가능한 한 신속하게 제작하는 것이다.
③ 나선형 모델은 위험분석을 강조하는 모델이다.
④ 신속한 프로토타이핑 모델은 소프트웨어 개발의 각 단계가 뚜렷하게 구분되는 모델이다.

24. 소프트웨어 개발 모델에 대한 설명으로 옳지 않은 것은? (2)
① 폭포수 모델은 실제 프로젝트가 순차적이라기보다는 반복적인 성향을 가지므로 개발 모델로 적합하지 않은 경우가 많다.
② 폭포수 모델은 프로그램의 모든 요구사항을 초기에 완전히 파악하도록 요구하므로 개발 프

로젝트의 불명확성을 미연에 방지할 수 있다는 장점이 있다.
③ 폭포수 모델에서는 개발 초기에 작동 가능한 프로그램을 제공하지 못하므로 고객에게 많은 인내심을 요구한다.
④ 원형 모델(prototyping model)에서 개발자는 시제품을 빨리 완성하기 위해서라면 효율성과 무관한 알고리즘을 사용해도 좋다.
⑤ 나선형 모델은 폭포수 모델과 원형 모델의 장점을 취합하여 점진적인 성과를 보면서 위험 부담을 줄일 수 있는 방법이다.

25. 소프트웨어 시스템의 프로토타입 모델을 신속히 만드는 가장 중요한 목적은? (2)
① 알고리즘을 결정한다.
② 사용자의 요구사항을 정확히 파악한다.
③ 소프트웨어의 전체적인 구조를 결정한다.
④ 자료구조를 결정한다.

26. 프로토타입 모델에 대한 설명 중 틀린 것은? (2)
① 사용자 요구분석의 어려움을 해결하기 위해 실제 개발될 소프트웨어의 견본품(prototyping)을 만들어 의사소통의 도구로 사용된다.
② 프로토타입 모델은 재사용 가능 모듈을 사용한다.
③ 프로토타입은 사용자와 시스템간의 인터페이스에 초점을 맞춰 개발되는데 고객으로부터 피드백을 얻은 후에는 버리는 경우도 있다.
④ 프로토타입의 진정한 역할은 고객의 요구가 무엇인지 결정하는데 있으므로 프로토타입의 내부적 구조는 크게 상관하지 않아도 된다.
⑤ 원형 모델(prototyping model)에서 개발자는 시제품을 빨리 완성하기 위해서라면 효율성과 무관한 알고리즘을 사용해도 좋다.

27. 나선형 모델에 대한 설명 중 틀린 것은? (5)
① 폭포수와 프로토타이핑 모델의 장점을 결합하였다.
② 계획-위험성 분석-공학화-사용자 평가의 순서로 나선으로 구성된다.
③ 점진적으로 개발에 의해 구성하기 때문에 위험부담을 줄일 수 있다.
④ 비용이 많이 들고 대규모 시스템에 적합하고 현실적으로 많이 사용한다.
⑤ 폭포수와 프로토타입에 비해 간단하고 4단계만 수행하면 되므로 단순한 프로젝트에 적합하다.

28. 소프트웨어 공학 패러다임 중 위험(risk) 부담을 고려한 패러다임은? (3)

① 폭포수 모델
② 프로토타입 패러다임
③ 나선형 패러다임
④ 4GL

29. 소프트웨어 생명주기 모델에서 나선형 모델에 관한 설명으로 옳지 않은 것은? (1)
① 이상적인 모델이기 때문에 실세계에서의 소프트웨어 개발과정과는 크게 다를 수 있다.
② 폭포수 모델과 프로토타이핑 모델의 장점을 결합시킨 모델이다.
③ 계획, 위험 분석, 공학, 사용자 평가의 과정을 반복적으로 수행한다.
④ 소프트웨어 생산과정을 진화적(evolutionary)인 방법으로 접근하는 방식이다.
⑤ 규모가 큰 소프트웨어를 제작하는데 있어 가장 현실적인 방법이다.

30. 소프트웨어 수명 주기 모형 중 폭포수 모형의 개발 단계로 옳은 것은? (2)
① 계획 – 분석 – 설계 – 시험 – 구현 – 유지보수
② 계획 – 분석 – 설계 – 구현 – 시험 – 유지보수
③ 계획 – 설계 – 분석 – 구현 – 시험 – 유지보수
④ 계획 – 분석 – 설계 – 구현 – 시험 – 유지보수

31. 소프트웨어 생명주기 모델 중 Boehm이 제시한 고전적 생명주기 모델로서 선형 순차적 모델이라고도 하며, 타당성 검토, 계획, 요구사항 분석, 설계, 구현, 테스트, 유지보수의 단계를 통해 소프트웨어를 개발하는 모델은? (1)

① 폭포수 모델 ② 프로토타입 모델
③ 나선형 모델 ④ RAD 모델

32. 폭포수 모델에 대한 설명으로 옳지 않은 것은? (2)
① 소프트웨어 개발 과정의 각 단계가 순차적으로 진행된다.
② 앞 단계에서 발견하지 못한 오류를 다음 단계에서 발견했을 때 오류 수정이 용이하다.
③ 두 개 이상의 과정이 병행 수행되거나 이전 단계로 넘어가는 경우가 많다.
④ 개발 과정 중에 발생하는 새로운 요구나 경험을 설계에 반영하기 힘들다.

33. 소프트웨어 수명 주기 모델 중 폭포수 모델에 대한 설명으로 옳지 않은 것은? (4)
① 적용 사례가 많다.
② 단계별 정의가 분명하다.
③ 단계별 산출물이 명확하다.
④ 요구사항의 변경이 용이하다.

34. 폭포수 모델에 대한 설명으로 옳지 않은 것은? (2)
① 앞 단계가 끝나야만 다음 단계로 넘어갈 수 있다.
② 요구 분석 단계에서 프로토타입을 사용하는 것이 특징이다.
③ 제품의 일부가 될 매뉴얼을 작성해야 한다.

④ 각 단계가 끝난 후 결과물이 명확히 나와야 한다.

35. 다음 설명에 해당하는 생명주기 모델은? (1)

가장 오래된 모델로 많은 적용 사례가 있지만 요구사항의 변경이 어려우며, 각 단계의 결과가 확인된 후에야 다음 단계로 넘어간다. 선형 순차적 모델로 고전적 생명주기 모델이라고도 한다.

① 폭포수(waterfall) 모델 ② 프로토타입(prototype) 모델
③ 코코모(COCOMO) 모델 ④ 점진적(spiral) 모델

36. 시스템의 일부 혹은 시스템의 모형을 만드는 과정으로서, 요구된 소프트웨어의 일부를 구현하여, 추후 구현 단계에서 사용될 골격 코드가 되는 모형은? (3)

① 폭포수 모형 ② 점증적 모형
③ 프로토타입 모형 ④ 계획 수립 모형

37. 실제 상황이 나오기 전에 가상으로 시뮬레이션을 통해 최종 결과물에 대한 예측을 할 수 있는 소프트웨어 수명주기 모형은? (2)

① 집중적 모형(Spiral Model) ② 프로토타입 모형(Prototype Model)
③ 코코모 모형(COCOMO Model) ④ 폭포수 모형(Waterfall Model)

38. 프로토타이핑 모델에 대한 설명으로 옳지 않은 것은? (4)
① 프로토타이핑 모델은 발주자나 개발자 모두에게 공통의 참조 모델을 제공한다.
② 사용자의 요구사항을 충실히 번영할 수 있다.
③ 프로토타이핑 모델은 소프트웨어 생명주기에서 유지보수가 없어지고 개발 단계 안에서 유지보수가 이루어지는 것으로 볼 수 있다.
④ 최종 결과물이 만들어지는 소프트웨어 개발 완료시점에서 최초로 오류 발견이 가능하다.

39. 프로토타입 모델의 장점으로 가장 적절한 것은? (3)
① 프로젝트관리가 용이하다.
② 노력과 비용이 절감된다.
③ 요구사항을 충실히 반영한다.
④ 관리와 개발이 명백히 구분된다.

40. 소프트웨어 생명주기 모델 중 다음 설명에 해당하는 것은? (3)

• 시스템 기능을 사용자에게 미리 보여줌으로써 개발자와 사용자 간의 오해요소를 줄인다. • 사용자와 개발자 간의 커뮤니케이션이 원활하지 못할 때 서로의 이해에 도움을 준다. • 실제 개발될 시스템 견본을 미리 만들어 최종 결과물을 예측하는 모델이다.

① 폭포수 모델 ② 나선형 모델
③ 프로토타입 모델 ④ 4GT 모델

41. 소프트웨어 수명 주기 모델 중 나선형 모델의 단계와 그 순서가 올바르게 구성된 것은? (2)
① Planning -> Requirement Analysis -> Development -> Maintenance
② Planning -> Risk Analysis -> Engineering -> Customer Evaluation
③ Requirement Analysis -> Planning -> Design -> Maintenance
④ Requirement Analysis -> Risk Analysis -> Development -> Maintenance

42. Boehm이 제안한 나선형 모델의 태스크(Task)에 해당되지 않는 것은? (3)

① 계획 수립(Planning) ② 위험 분석(Risk Analysis)
③ 객체 구현(Object Implementation) ④ 고객 평가(Customer Evaluation)

33. 나선형 모델에 대한 설명으로 옳지 않은 것은? (4)
① 여러 번의 개발 과정을 거쳐 점진적으로 완벽한 소프트웨어를 개발한다.
② 대규모 시스템의 소프트웨어 개발에 적합하다.
③ 위험성 평가에 크게 의존하기 때문에 이를 발견하지 않으면 문제가 발생할 수 있다.
④ 실제 개발된 소프트웨어에 대한 시제품을 만들어 최종 결과물을 예측하는 모델이다.

44. 다음 설명의 개발 방법론에 가장 부합하는 모델은? (1)

반복되는 각 주기마다 먼저 목표를 설정하며 목표를 성취하기 위한 방안과 존재하는 제약사항을 파악한다. 다음은 여러 대안들에 대하여 프로젝트의 위험분석을 실시한다. 그리고 문제와 위험을 해결하는 전략을 개발하는데, 벤치마킹, 시뮬레이션 등과 같은 방법을 이용한다. 그 후에 소프트웨어의 개발 및 검증을 수행하고 다음 단계를 계획한다.

① 나선형 모델 ② 점증적 모델(incremental model)
③ 프로토타이핑 모델 ④ V 모델

45. 통합 프로세스 모델(UP: Unified Process)의 특징을 설명한 것으로 옳지 않은 것은? (4)
① 반복적(iterative)이고, 점진적(incremental)인 개발 방법이다.
② 유스케이스를 기반으로 한다.
③ 아키텍처 중심의 개발을 지향한다.

④ 위험 관리는 포함되어있지 않다.

46. 애자일(agile) 선언문에 대한 설명으로 옳지 않은 것은? (3)
① 프로세스와 도구보다 개인과 그들의 협업에 더 가치를 둔다.
② 포괄적인 문서화보다 제대로 작동하는 소프트웨어에 더 가치를 둔다.
③ 고객과의 협력보다는 계약 협상에 더 가치를 둔다.
④ 계획에 따르기보다는 변화에 대응하는 것에 더 가치를 둔다.

47. 애자일(agile) 방법은 설계와 문서화보다 소프트웨어 그 자체에 초점을 두는 개발방법론이다. 애자일 방법 중에 제일 많이 알려진 것이 익스트림 프로그래밍(eXtreme Programming: XP)인데, 다음 중 XP와 가장 연관성이 적은 것은? (4)
① 페어 프로그래밍(pair programming)
② 지속적인 통합(continuous integration)
③ 리팩토링(refactoring)
④ 제네릭 프로그래밍(generic programming)

48. 익스트림 프로그래밍(eXtreme Programming: XP)에 대한 설명으로 옳지 않은 것은? (3)
① 요구사항은 스토리 카드에 기록되고, 릴리스(release)마다 스토리의 상대적 우선순위가 결정된다.
② 기능이 구현되기 이전에 그 기능을 시험하기 위해 자동화된 단위시험 프레임워크를 이용한다.
③ 현재의 요구사항과 미래의 요구사항을 충족할 수 있도록 충분한 설계를 한다.
④ 모든 개발자들이 코드에 대한 공동 책임을 가지며 개발자 누구든지 어떤 코드라도 변경할 수 있다.

49. Agile 소프트웨어 개발에서 많이 사용되는 XP(eXtreme Programming)의 프로세스와 그에 대한 설명으로 가장 거리가 먼 것은? (4)
① 계획 수립: 요구사항은 릴리스에 포함될 스토리 카드와 스토리에 기록되고, 가용시간에 상대적인 우선순위를 결정한다.
② 설계: 복잡한 표현보다는 간단한 설계를 선호한다.
③ 코딩: 핵심 개념 중 하나로 두 사람이 작업을 같이하는 pair programming이 있다.
④ 테스팅: 코딩 후에 단위 테스트 케이스를 생성하고 이후 인수 테스트가 이루어진다.

50. 애자일(Agile) 소프트웨어 개발과 가장 관련이 적은 내용은? (4)
① 적응적 소프트웨어 개발(adaptive software development)
② 익스트림 프로그래밍(extreme programming)
③ 테스트 주도 개발(test-driven development)

④ 철저한 계획 및 문서화

51. TDD(Test Driven Development)에 대한 설명으로 가장 적절하지 않은 것은? (3)
① 코딩하기 전에 테스트를 먼저 생각함으로써 결함이 없는 소프트웨어를 구현하는 것을 목표로 한다.
② 테스트 우선 접근 방식(Test First Approach)이라고도 부른다.
③ 프로그래머가 코드를 작성하기 전에 수준의 기능 테스트를 먼저 작성하지는 않는다.
④ 프로그래머는 먼저 테스트 코드를 작성한 다음 그 테스트를 통과하는 실제코드를 단계적으로 만들어간다.

52. 다음 개발환경에 가장 적합한 개발기법은? (3)

가능한 한 짧은 시간 내에 소프트웨어를 개발하고자 하며, 개발자들은 상호간 개발하는 과정에서 다른 작업자의 작업을 확인하는 pair programming, continuous integration, simple design, small release, incremental planning 등에 익숙해 있으며 개발과정에서 리팩토링을 수행할 수 있다고 한다. 모든 요구사항들을 위한 시나리오 카드가 준비되어 있으며 또한 고객의 요구사항을 직접적으로 그리고 신속히 반영하기 위해 고객을 참여시키고자 한다.

① 프로토타입 모델
② 폭포수 모델
③ XP(eXtreme Programming)
④ CBD(컴포넌트 기반 개발)

53. 소프트웨어 개발 프로세스인 XP(eXtreme Programming)의 실무 관행(practice)에 해당하지 않는 것은? (3)
① 짝 프로그래밍(pair programming)
② 소규모 릴리스(small release)
③ 구현우선개발(implementation-first development)
④ 점증적인 계획 수립(incremental planning)

54. 코드 리뷰(Code Review)의 기능을 직접적으로 수행할 수 있는 XP(eXtreme Programming)의 실무 관행(practice)은? (2)
① 단순 설계(simple design)
② 짝 프로그래밍(pair programming)
③ 소규모 릴리스(small release)
④ 메타포(metaphor)

55. 다음 중 Agile 방법론 중의 하나인 XP(eXtreme Programming)의 12가지 실천 사항에 속하지 않는 것은? (4)

① 지속적인 통합
② 페어(pair) 프로그래밍
③ 코드 공동 소유
④ 통계적 품질 관리

56. 익스트림 프로그래밍(XP)에 대한 설명으로 옳은 것은? (2)
① 중소규모 프로젝트보다 대규모 프로젝트에 적용이 적합하다.
② 개발되는 코드에 대한 집단적 소유권(collective ownership)을 갖는다.
③ 요구사항 분석 및 설계에 대한 비중을 높일 수 있다.
④ 문서화 작업으로 발생하는 부하가 증가된다.

Chapter 3. 연습문제

1. 소프트웨어 측정을 직접 측정(direct measure)과 간접 측정(indirect measure)으로 나누어 볼 수 있는데, 이들에 대한 설명으로 틀린 것은? (1)
   ① 직접 측정들에는 기능성, 비용, 노력이 해당한다.
   ② 직접 측정들에는 코드 라인수(LOC), 메모리의 크기, 오류의 수 등이 있다.
   ③ 간접 측정들에는 효율성, 품질, 복잡도, 유지보수성 등이 해당한다.
   ④ 소프트웨어 측정은 품질 척도에 쓰인다.

2. 소프트웨어 프로젝트의 효과적인 관리는 4P에 초점을 둔다. 4P에 해당하지 않는 것은? (4)

   ① 사람(people)　　② 제품(product)
   ③ 프로젝트(project)　　④ 프로그램(program)

3. 소프트웨어 프로젝트의 계획 수립 시 문제에 대한 정의를 기술해야 하는데, 다음 중 문제의 정의 시에 기술할 항목이 나닌 것은? (5)
   ① 전반적인 업무의 조사 및 분석
   ② 전산화 추진 전략의 제시
   ③ 하드웨어, 소프트웨어, 인력 서브시스템에서 제공되는 기능 및 제약사항
   ④ 개발과정 및 생산제품에 대한 시스템 차원의 목표와 요구사항
   ⑤ 각 모듈이 해결해야 할 문제의 정의

4. 소프트웨어 프로젝트 관리에 대한 설명으로 가장 옳은 것은? (1)
   ① 주어진 기간 내에 최소의 비용으로 사용자를 만족시키는 시스템 개발
   ② 주어진 기간은 연장하되 최소의 비용으로 시스템을 개발
   ③ 소요 인력은 최소로 하되 정책 결정은 신속하게 처리
   ④ 개발에 따른 산출물 관리

5. 프로젝트 관리의 대상으로 거리가 먼 것은? (3)

   ① 비용 관리　　② 일정 관리
   ③ 고객 관리　　④ 품질 관리

6. 효과적인 소프트웨어 프로젝트 관리를 위한 3P에 해당되지 않는 것은? (2)

① People(사람): 인적 자원
② Product(생산물): 생산 일정
③ Problem(문제): 문제 인식
④ Process(프로세스): 작업 계획

7. 소프트웨어 프로젝트 관리에 영향을 주는 3대 요소는? (1)

① 사람, 문제, 프로세스
② 문제, 프로젝트, 작업
③ 사람, 문제, 도구
④ 작업, 문제, 도구

8. 소프트웨어 프로젝트 관리를 효율적으로 수행하기 위한 3P 중 소프트웨어 프로젝트를 수행하기 위한 Framework의 고려와 가장 연관된 것은? (4)

① People
② Problem
③ Product
④ Process

9. 프로젝트 계획 수립을 시작할 때 제일 먼저 해야 하는 작업은? (4)

① 개발 완료 날짜 파악
② 과거의 데이터를 분석하는 일
③ 개발 비용 산정
④ 프로젝트의 규모 파악

10. 소프트웨어 프로젝트 계획 수립 시 소프트웨어 영역(범위) 결정의 주요요소로 거리가 먼 것은? (2)

① 기능
② 인적 자원
③ 인터페이스
④ 성능

11. 소프트웨어 프로젝트를 신뢰성 있게 예측하는 방법 중 현실성이 부족한 것은? (1)

① 예측을 가능한 한 뒤로 미룬다.
② 이미 수행된 유사 프로젝트를 참고한다.
③ 프로젝트를 상대적으로 잘게 분리하여 예측한다.
④ 경험적 예측 모델을 활용한다.

12. 프로젝트 계획 단계에 대한 설명으로 옳지 않은 것은? (3)

① 제한된 자원과 일정에 대한 최적의 방법을 찾고자 노력해야 한다.
② 계획에 따라 소프트웨어의 품질이 결정되기도 한다.
③ 비용 추정에 관한 문제는 계획 단계에 포함되지 않는다.

④ 계획 단계에서 프로젝트 관리자의 임무는 매우 중요하다.

13. 일정 계획과 관계가 먼 것은? (3)

① 작업 분해
② CPM 네트워크
③ 프로그램 명세서
④ 간트 차트(Gantt Chart)

14. CPM 네트워크에 대한 설명으로 옳지 않은 것은? (4)

① 노드에서 작업을 표시하고 간선은 작업 사이의 전후 의존 관계를 나타낸다.
② 프로젝트 완성에 필요한 작업을 나열하고 작업에 필요한 소요 기간을 예측하는데 사용한다.
③ 박스 노드는 프로젝트의 중간 점검을 뜻하는 이정표로 이 노드 위에는 예상 완료 시간을 표시한다.
④ 한 이정표에서 다른 이정표에 도달하기 전의 작업은 모두 완료되지 않아도 다음 작업을 진행할 수 있다.

15. Gantt Chart에 포함되지 않는 사항은? (4)

① 이정표
② 작업 일정
③ 작업기간
④ 주요 작업 경로

16. CPM에 대한 설명으로 옳지 않은 것은? (2)

① 프로젝트 내에서 각 작업이 수행되는 시간과 각 작업 사이의 관계를 파악할 수 있다.
② 작업 일정을 한눈에 볼 수 있도록 해주며 막대 그래프의 형태로 표현한다.
③ 경영층의 과학적인 의사 결정을 지원한다.
④ 효과적인 프로젝트의 통제를 가능하게 해 준다.

17. 프로젝트 일정 관리 시 사용하는 간트 차트에 대한 설명으로 옳지 않은 것은? (4)

① 막대로 표시하며, 수평 막대의 길이는 각 태스크의 기간을 나타낸다.
② 이정표, 기간, 작업, 프로젝트 일정을 나타낸다.
③ 시간선(Time-Line) 차트라고도 한다.
④ 작업들간의 상호 관련성, 결정 경로를 표시한다.

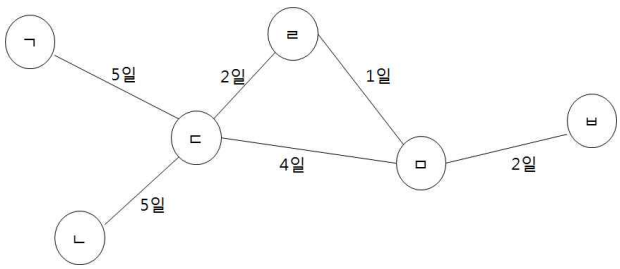
18. 개발 프로젝트의 작업을 파악하고 순서와 일정을 계획하여 임계경로를 구할 수 있는 것은? (1)

① PERT
② Gantt chart
③ WBS
④ Program Development Book
⑤ SDLC

19. 프로젝트 관리를 위한 도구 및 기법이 아닌 것은? (1)

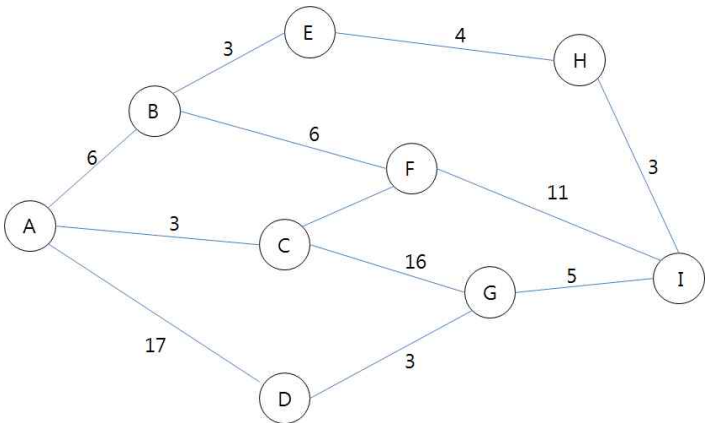
① Use-case diagram
② PERT Chart
③ GANTT Chart
④ Work Breakdown Structure

20. 아래의 CPM network에서 Critical Path를 구하면? (1)



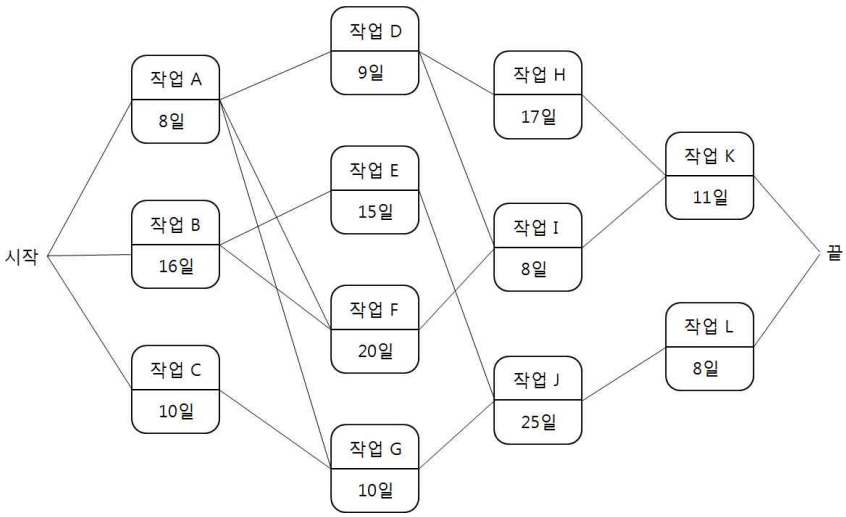
① ㄱ ㄷ ㅁ ㅂ
② ㄴ ㄷ ㅁ ㅂ
③ ㄱ ㄷ ㄹ ㅁ ㅂ
④ ㄴ ㄷ ㄹ ㅁ ㅂ

21. 다음은 프로젝트의 일정관리 도구인 CPM network이다. 임계 경로로서 알맞은 것은? (5)



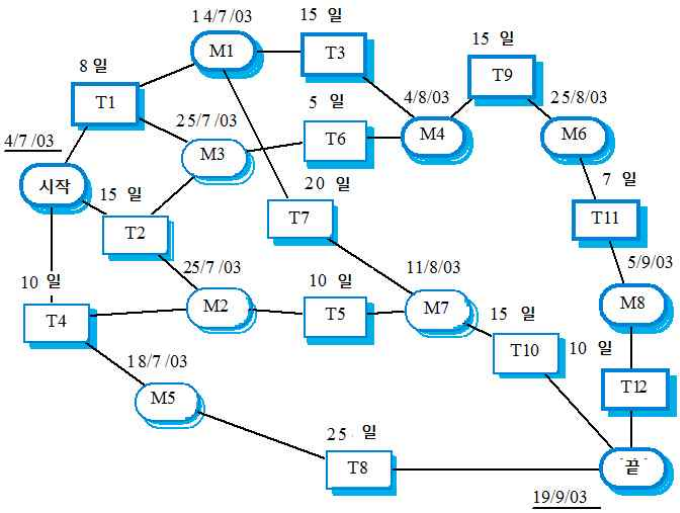
① ABFI
② ACFI
③ ABEHI
④ ACGI
⑤ ADGI

22. 다음 그림은 어떤 프로젝트의 일정 계획이다. 간선은 작업들 간의 의존 관계를 나타내며, 노드는 작업과 그 작업의 소요기간을 나타낸다. 프로젝트 일정 계획의 최소 예측기간과 위험관리 중점 노드가 올바르게 짝지어진 것은? (4)



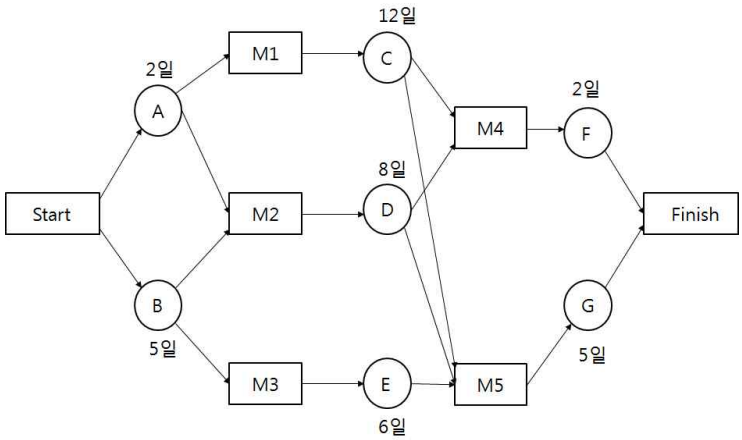
① 36, A
② 36, H
③ 64, A
④ 64, J
⑤ 55, B

23. 프로젝트 일정계획을 위해 액티비티 네트워크가 사용된다. 다음의 액티비티 네트워크를 갖는 프로젝트의 경우 프로젝트가 종료되는데 소요되는 최소의 시간은 얼마인가? (4)



① 35일 ② 40일
③ 52일 ④ 55일

24. 다음 그림은 프로젝트 일정 관리에 사용되는 작업 네트워크(Activity Network)를 보여준다. 임계 경로(Critical Path)는 다음 중 어느 것인가? (원은 작업, 원위의 숫자는 작업소요가간, 네모는 이정표) (2)



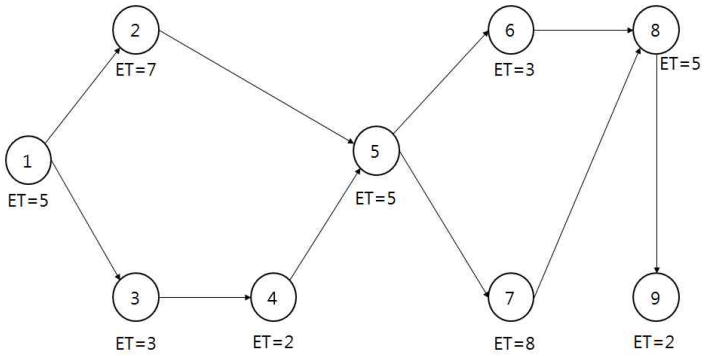
① Start – A – M2 – D – M4 – F –Finish
② Start – A – M1 – C – M5 – G –Finish
③ Start – A – M1 – C – M4 – F –Finish
④ Start – B – M3 – E – M5 – G –Finish
⑤ Start – B – M2 – D – M5 – G –Finish

25. 다음은 어떤 프로젝트를 구성하는 작업들의 선행 작업과 소요기간을 나타낸 것이다. 이러한 작업 의존 관계를 바탕으로 작업 D를 최대한 빠르게 시작할 수 있는 착수일과 최대한 늦추어 시작할 수 있는 착수일 간의 차이는? (2)

작업	선행 작업	소요기간(일)
A	start	2
B	start	4
C	A	3
D	B, C	2
E	C	4
end	B, D, E	

① 1일 ② 2일 ③ 3일 ④ 4일

26. 다음 PERT Chart에 대한 설명으로 옳지 않은 것은? (3)



① 프로젝트가 종료되는 데에 소요되는 최소 사간은 32일이다.
② 프로젝트 일정 중 임계 경로는 1-2-5-7-8-9이다.
③ 여유시간(slack time)은 4번 노드가 가장 많으며 2일의 여유시간이 있다.
④ 최대한 빠르게 끝날 수 있는 시간(earliest finish time)과 최대로 늦추어 끝날 수 있는 시간(latest finish time)이 같은 노드는 임계 경로에 있다.

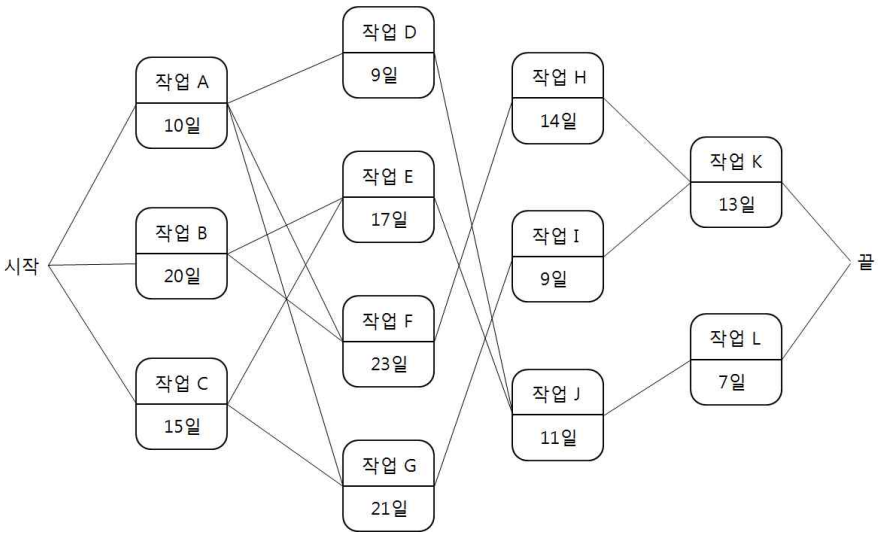
27. 프로젝트 일정계획수립에 사용되는 표기법별 용도에 대한 설명으로 옳지 못한 것은? (4)
① 바 차트(bar chart) - 작업(activity)에 대한 책임자 및 작업의 시작과 종료 시점을 표현
② 액티비티 네트워크(activity network) - 작업들 간의 종속성(dependency)을 표현
③ 작업 분해 구조(work breakdown structure) - 프로젝트 완성에 필요한 소작업들의 계층적 목록
④ 플로우차트(flowchart) - 단위 업무(task) 데이터의 흐름을 나타냄

28. 다음 표는 어떤 프로젝트를 수행하는데 필요한 작업, 작업 수행기간, 작업들 간의 종속 관계(선후 관계)를 나타낸 것이다. 프로젝트의 전체 일정은 지연되지 않고 반드시 원래의 일정대로 종료되어야 한다면 T8 작업을 수행하는데 허락되는 최대 지연시간은? (2)

작업	작업 수행 기간	종속 관계
T1	15	-
T2	10	T1
T3	15	T1, T2
T4	15	-
T5	15	T4
T6	10	T3
T7	10	T6
T8	15	T3, T5
T9	10	T7

① 10일 ② 15일 ③ 20일 ④ 25일

29. 다음 그림은 12개의 작업으로 구성된 프로젝트의 일정 계획이다. 간선은 작업들 간의 의존 관계를 나타내며, 노드는 작업과 그 작업의 소요기간을 나타낸다. 프로젝트 일정 계획의 최소 예측기간과 위험관리 중점 노드가 바르게 짝지어진 것은? (3)



① 60일, 작업 F
② 70일, 작업 I
③ 70일, 작업 F
④ 37일, 작업 J

30. 다음 표는 어떤 프로젝트를 수행하는데 필요한 작업, 작업 수행기간, 선후행 관계를 나타낸 것이다. 이 프로젝트의 일정에서 여유 기간(slack time)이 0인 작업은? (2)

작업	작업 수행 기간(일)	선행 작업
A	10	-
B	15	-
C	20	A
D	15	B
E	15	A, B
F	10	C
G	20	D
H	30	E
I	20	H

① A
② B
③ C
④ D

31. 다음 표는 어떤 프로젝트를 수행하는데 필요한 작업, 작업 수행기간, 작업들 간의 종속 관계(선후 관계)를 나타낸 것이다. T5 작업을 수행하는데 5일이 지연되어 15일이 소요되었다고 하자. 그렇지만 프로젝트의 전체 일정은 지연되지 않고 반드시 원래의 일정대로 종료되어야 한다면 T8 작업을 수행하는데 허락되는 최대 지연시간은 얼마인가? (2)

작업	수행 기간	직전 선행 작업
T1	15	-
T2	10	T1
T3	10	T1, T2
T4	15	-
T5	10	T4
T6	10	T3
T7	15	T6
T8	15	T3, T5
T9	5	T7

① 10일 ② 15일 ③ 20일 ④ 25일

32. 다음은 어떤 소프트웨어 프로젝트를 구성하는 작업들의 선행 작업과 소요기간을 나타낸 표이다. 이 프로젝트를 위한 액티비티 네트워크의 임계경로는? (2)

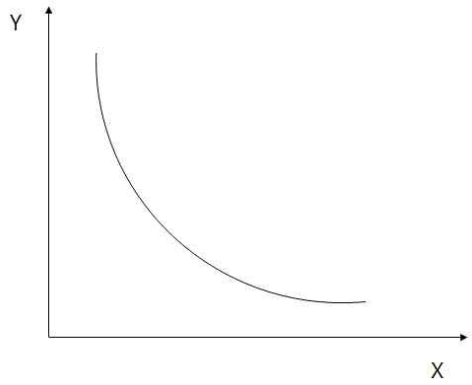
작업	T1	T2	T3	T4	T5	T6	T7	종료
소요 기간 (일)	6	4	2	3	5	3	2	
선행 작업	-	-	T1	T2	T2, T3, T4	T3, T5	T5, T6	T7

① T1 -> T3 -> T6 -> T7
② T1 -> T3 -> T5 -> T6 -> T7
③ T2 -> T5 -> T6 -> T7
④ T2 -> T4 -> T5 -> T6 -> T7

33. 소프트웨어 프로젝트의 비용 결정 요소와 관련이 적은 것은? (3)

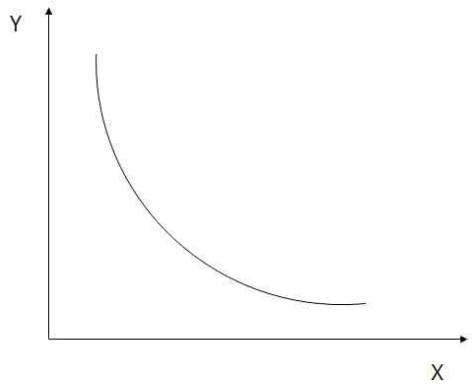
① 개발자의 능력 ② 요구되는 신뢰도
③ 하드웨어의 성능 ④ 개발제품의 복잡도

34. 소프트웨어 개발 비용은 여러 가지의 다른 요소들과 일정한 상관관계가 있다. 아래 그래프의 Y축을 개발 비용이라고 했을 때, X축은 어떤 요소라고 보는 것이 가장 타당한가? (2)



① 시스템 크기
② 개발 기간
③ 신뢰도
④ 투입 인력

35. 소프트웨어 개발 비용은 여러 가지의 다른 요소들과 일정한 상관관계가 있다. 아래 그래프의 Y축을 개발 비용이라고 했을 때, X축 요소로 볼 수 있는 것 중 가장 적절한 것은? (1)



① 관련 업무지식 및 경험
② 투입 인력
③ 신뢰도
④ 시스템의 규모

36. LOC 기법에 의하여 예측된 총 라인수가 35,000라인일 경우 개발에 투입될 프로그래머의 수가 5명이고, 프로그래머들의 평균 생산성이 월 평균 500라인이라고 할 때 개발에 소요되는 시간은? (3)

① 10개월 ② 12개월
③ 14개월 ④ 16개월

37. 두 개의 모듈로 구성된 프로젝트가 있다. LOC(Line of Code) 기반으로 규모를 추정하려고 한다. 각 모듈의 규모 추정이 아래와 같을 때, 프로젝트의 총 규모 LOC는? (2)

모듈 번호	낙관적 LOC	보통 LOC	비관적 LOC
1	100	200	300
2	60	100	200

① 300 ② 310
③ 320 ④ 330

38. A회사에서 ERP 시스템 개발을 위해 추정된 규모는 33,600 LOC이다. 이 회사의 과거 평균 생산성이 600LOC/PM이고, 직원의 평균 인건비가 480만원이라고 할 때, 이 프로젝트의 개발에 필요한 인원 및 개발비용으로 옳은 것은? (3)

① 55명 268,800,000원 ② 55명 269,000,000원
③ 56명 268,800,000원 ④ 56명 269,000,000원

39. LOC/MD는 하루(D) 동안 프로그래머(M)가 작성한 코드 라인수(LOC)로서 생산성을 나타내는 단위이다. 어떤 프로젝트에 20명의 프로그래머가 투입되어 1년 동안 100,000LOC를 작성했다고 하자. 1년간 실제 일한 날이 250일이라고 할 때, 이 프로젝트의 생산성은? (2)

① 15LOC/MD ② 20LOC/MD
③ 25LOC/MD ④ 40LOC/MD

40. 개발 예정인 소프트웨어의 소스코드 라인 수는 50,000 SLOC(Source Line of Code)로 추정되었다. 이 소프트웨어를 다음 표의 조건에서 8명의 개발자가 개발하려 할 때, 소프트웨어 개발이 완료되기까지 최소 몇 개월이 소요될 것으로 추정되는가? (4)

개발 언어	C++
C++언어 라인수/FP	50 SLOC/FP
개발자 1인당 월평균 생산성	50 FP/month

① 13개월 ② 20개월
③ 23개월 ④ 25개월

41. 어떤 소프트웨어의 총 개발 기간은 12개월이며, 5명의 인력이 처음에 투입되었다. 5개월 시작 시점에 추가로 3명이 투입되어 개발이 완료되었다면, 이 소프트웨어의 개발에 투입된 총 인-월(Man-Month)은? (3)

① 75 ② 81 ③ 84 ④ 96

42. 두 명의 개발자가 5개월에 걸쳐 10,000 라인의 코드를 개발하였을 때, 월별(Person Month) 생산성 측정을 위한 계산 방식으로 가장 적합한 것은? (3)

① 10,000/2 ② 10,000/5
③ 10,000/(5×2) ④ (2×10,000)/5

43. 어떤 소프트웨어 개발을 위해 10명의 개발자가 10개월 동안 참여되었다. 그런데 그 중 7명은 10개월 동안 계속 참여했지만 3명은 3개월 동안만 부분적으로 참여했다. 이 소프트웨어 개발을 위한 인월(Man Month)은 얼마인가? (3)

① 100 ② 70 ③ 79 ④ 60

44. 비용 산정 기법 중 소프트웨어 각 기능의 원시 코드 라인 수의 비관치, 낙관치, 기대치를 측정하여 예측치를 구하고 이를 이용하여 비용을 산정하는 기법은? (3)

① Effort Per Task 기법 ② 전문가 감정 기법
③ LOC 기법 ④ 델파이 기법

45. LOC 기법에 의하여 예측된 총 라인 수가 25,000라인일 경우 개발에 투입될 프로그래머의 수가 5명이고, 프로그래머들의 평균 생산성이 월당 500라인일 때, 개발에 소요되는 시간은? (3)

① 8개월 ② 9개월 ③ 10개월 ④ 11개월

46. 다음 소프트웨어 기능점수 모델에 대한 설명으로 적절하지 않은 것은? (3)

① Albrecht가 최초로 제안했다.
② 입출력의 수, 조회의 수, 파일의 수, 외부 인터페이스의 수들을 집합시켜 소프트웨어가 제공해야 하는 하나의 기능점수로 환산한다.
③ 데이터의 복잡도, 프로그램간의 구조적 연관성, 사용 프로그램언어를 복합적으로 고려한다.
④ 데이터 복잡도가 낮고, 처리 알고리즘이 중요한 소프트웨어에는 적합하지 못하다.

47. 기능점수 산정방식 중에서 소프트웨어 규모를 산정하기 위한 항목으로 틀린 것은? (2)

① 외부 입력(External Input)
② 내부 출력(Internal Output)
③ 논리적 내부파일(Internal Logical File)
④ 외부 조회(External Inquiry)

48. 다음 중 IFPUG(International Function Point Users Group)에서 정의한 기능점수 측정 유형이 아닌 것은? (3)

① 개발 프로젝트 기능점수 측정
② 개선 프로젝트 기능점수 측정
③ 서비스 프로젝트 기능점수 측정
④ 애플리케이션 기능점수 측정

49. 기능점수는 사용자가 요구한 소프트웨어 기능을 기반으로 프로젝트의 크기를 산정한다. 일반적으로 기능점수를 산출하는 방법으로 IFPUG(International Function Point Users Group)에서 제공하는 7단계 산출 프로세스를 따른다. 아래의 표를 참조하여 기능점수 산출 프로세스를 가장 바르게 표시한 것은? (3)

가) 데이터 기능 측정
나) 미조정 기능점수 결정
다) 트랜잭션 기능 측정
라) 조정 기능점수 결정
마) 조정 인자 결정
바) 측정 유형의 결정
사) 측정 범위와 애플리케이션 경계 식별

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

50. 기능점수방법에서 사용되는 기능점수 항목이 아닌 것은? (3)

① 조회 개수
② 파일 개수
③ 코드 라인수
④ 인터페이스 개수

51. 기능점수를 계산하기 위해서는 소프트웨어를 사용자 관점에서 다섯 가지의 기능 요소로 구분한 후, 각 요소의 복잡도를 계산해야 한다. 다섯 가지 기능 요소에 해당하지 않는 것은? (5)

① 외부 입력(External Input)
② 외부 출력(External Output)
③ 외부 조회(External Inquiry)
④ 내부 논리 파일(Internal Logical File)
⑤ 외부 논리 파일(External Logical File)

52. 기능점수에 대한 설명으로 옳지 않은 것은? (3)

① 기능점수는 소프트웨어 시스템이 가지는 기능을 정량화한 것이다.
② 기능점수의 산출 시 적용되는 가중치는 시스템의 특성에 따라 달라질 수 있다.
③ 기능점수는 구현언어와 밀접한 관련이 있는 메트릭(metric)이다.
④ 기능점수는 원시코드가 작성되기 전이라도 계산할 수 있다.

53. 기능점수에서 트랜잭션 기능의 복잡도를 결정하는 요소로만 나열한 것은? (2)

① DET, RET
② DET, FTR
③ FTR, RET
④ FTR, VAF

54. 다음 시스템의 요구사항에 대한 기능점수 산정 시 트랜잭션 유형, FTR, DET가 가장 바르게 짝지어진 것은? (1)

사원정보 데이터베이스는 종업원 아이디, 종업원 이름, 직위, 소재지, 전화번호 필드로 구성되어 있으며, 사용자가 '등록' 버튼을 클릭하면 사원정보 데이터베이스에 사원정보를 추가한다. 입력이 잘못되면 에러 메시지를 출력한다.

① EI, 1, 7
② EO, 2, 6
③ ILF, 1, 6
④ ILF, 2, 5

55. 기능점수에서 EI 중의 하나인 회원 삭제 기능의 복잡도를 산정하기 위해 DET 개수를 산출하려고 한다. ILF인 회원 정보는 회원 번호, 회원 이름, 연락처, 주소, 이메일 필드로 구성된다. 회원 삭제는 회원 번호와 기능 키(Ctrl+R)를 선택하여 이루어진다. 만약 삭제하려는 회원 번호가 회원 정보에 존재하지 않는다면, 오류 메시지가 출력된다. 산출되는 DET 개수는? (1)

① 3개
② 4개
③ 5개
④ 6개

56. 총 기능점수가 7,500FP 규모의 프로그램을 5명의 프로그래머가 참여하여 개발할 때 30개월이 소요될 것으로 예측되었다. 프로그래머 1인당 월평균 생산성은 얼마인가? (2)

① 25FP/월 ② 50FP/월
③ 150FP/월 ④ 250FP/월

57. 기능점수 항목과 복합 가중값, 개발 예정인 소프트웨어의 기능 구성, 기능점수와 코드라인수와의 관계가 다음과 같이 주어졌을 때, 개발 예정인 소프트웨어의 예상되는 소스코드 라인수를 가장 잘 추정한 것은? (3)

기능점수 항목 및 복합 가중값				SW 기능 구성(개수)		
기능항목	단순	보통	복잡	단순	보통	복잡
외부입력	3	4	6	4	1	0
외부출력	4	5	7	2	4	0
외부조회	3	4	6	2	2	2
내부논리파일	7	10	15	0	4	4
외부연계파일	5	7	10	2	0	2

언어	평균SLOC/FP
C	130
Pascal	90
C++	50
VB	30

① C 언어로 구현하면 24,000SLOC(Source Line of Code)로 추정된다.
② Pascal 언어로 구현하면 19,000SLOC로 추정된다.
③ C++ 언어로 구현하면 10,000SLOC로 추정된다.
④ VB 언어로 구현하면 5,000SLOC로 추정된다.

58. 소프트웨어 사업 대가산정 가이드라인 중, 기능점수 방식에 의한 소프트웨어 개발비 산정에 적용되는 보정계수에 해당하지 않는 것은? (2)

① 품질 및 특성 보정계수 ② 단계별 보정계수
③ 어플리케이션 유형 보정계수 ④ 언어 보정계수

59. 소프트웨어 사업 대가산정 가이드라인 중, 기능점수 방식에 의한 소프트웨어 개발비 산정에 사용되지 않는 요소는? (3)

① 개발원가 ② 직접경비
③ 간접경비 ④ 이윤

60. 소프트웨어 비용예측 방법이 아닌 것은? (3)

① 알고리즘 모델 ② 전문가 판단
③ 브룩스 법칙(Brooks law) ④ 상향식 산정

61. 다음 예는 비용산정 기법의 한 예이다. 내용에 가장 적절한 기법은? (2)

작업은 가능한 시간을 다 채울 때까지 확장된다. 비용은 객관적인 평가에 의한 것이 아니라 이용 가능한 자원에 의해 결정된다. 만약 소프트웨어가 12개월 이내에 인도되어야 하고, 5명의 인원이 이용 가능하다면 필요한 노력은 60person-month로 추정된다.

① 알고리즘 비용 모델 ② 파킨슨의 법칙
③ 유추에 의한 산정 ④ 전문가의 판단

62. 소프트웨어 개발비용 산정기법 중에서 미리 준비된 식과 표를 이용하여 비용을 산정할 수 있는 알고리즘 방식(algorithmic) 기법에 속하는 것은? (2)

① Expert Judgement ② COCOMO
③ Estimation by Analogy ④ Parkinsonian Estimation
⑤ Price-to-Win Estimation

63. 소프트웨어 프로젝트의 소요기간과 비용추정에 대한 설명으로 옳지 않은 것은? (4)

① COCOMO는 먼저 완성될 시스템의 규모를 추정하고 이를 준비된 식에 대입하여 소요 인원•월(Man-Month)을 예측한다.
② 기능점수방법은 원시코드의 구현에 이용되는 프로그래밍언어에 독립적이다.
③ 기능점수방법은 경험 중심적이기 때문에 비용을 산정하려면 단위 시간당 프로그래머의 생산성을 기능점수로 표현한 생산성 메트릭이 있어야 한다.
④ LOC 기반 추정과 기능점수 기반 추정은 사용 정보만 다를 뿐 동일한 결과를 내므로 선택적으로 사용이 가능하다.

64. 소프트웨어 개발비용을 산정하는데 다양한 기법들이 사용된다. 이와 거리가 가장 먼 것은? (3)

① 알고리즘 모델 ② 전문가 판단
③ 브룩스 법칙(Brooks Law) ④ 상향식 산정

65. 그룹회의의 부작용이 일어나지 않으면서도 전문가의 의견일치에 도달시키는 소프트웨어 개발비용 산정 기법은? (2)

① 원시 코드 라인수 산정방법
② 델파이(Delphi)식 산정기법
③ COCOMO 모형
④ 생명주기 예측 모형
⑤ 기능점수 모형

66. 다음 중 COCOMO 모델 유형에 속하지 않는 것은? (3)

① 단순형(Organic)
② 중간형(Semi-detached)
③ 분리형(Detached)
④ 임베디드형(Embedded)

67. 비용예측방법에서 COCOMO에 의한 방법 중 트랜잭션 처리시스템이나 운영체제, 데이터베이스 관리 시스템 등의 30만 라인 이하의 소프트웨어를 개발하는 유형은? (2)

① Organic 프로젝트
② Semidetached 프로젝트
③ Organic, Embedded 프로젝트
④ Embedded 프로젝트

68. COCOMO 모델에 대한 설명 중 맞는 것은? (2)

가) 적용하는 프로젝트의 규모에 따라 3단계로 구성된다. 나) 적용하는 프로젝트의 규모는 관계없다. 다) 프로젝트 기간 뿐 아니라 노력(effort)에 대한 추정도 가능하다. 라) 계산에 LOC(Line of Code)는 사용하지 않는다.

① A, B
② A, C
③ B, C
④ B, D

69. 다음 중 COCOMO 모형과 거리가 먼 것은? (1)

① 상위 및 하위모형으로 구분할 수 있다.
② 적용모드는 organic 모드, semi-detached 모드, embedded 모드로 구분할 수 있다.
③ 개발된 소프트웨어의 코드 줄 수에 기반한 비용추정모형이다.
④ 여러 프로젝트의 실적을 통계 분석하여 유도된 공식을 이용한다.

70. 소프트웨어 개발 노력을 추정하기 위한 다음 COCOMO 모델의 설명 중 틀린 것은? (4)

① Intermediate 모델은 영향을 미치는 15개 요인들을 포함한다.
② Basic 모델은 프로젝트 크기와 유형을 제외하고는 프로젝트에 영향을 미치는 요인들을 고려하지 않는다.
③ Intermediate 모델은 개발할 제품, 컴퓨터, 개인, 프로젝트의 특성을 고려한다.
④ Basic 모델은 개발의 노력(effort)에 대한 함수가 아니다.

71. 다음 중에서 COCOMO 모델에서 사용되는 노력승수값을 구하기 위해서 사용되는 요소가 아닌 것은? (4)

① 제품의 특성 ② 컴퓨터의 특성
③ 개발 요원의 특성 ④ 사용자의 특성

72. COCOMO 모델에서 사용되는 노력승수값 중 제품의 특성과 가장 거리가 먼 것은? (3)

① 요구되는 신뢰도 ② 제품의 복잡도
③ 실행시간의 제약 ④ 데이터베이스 크기

73. COCOMO II 모델은 1981년에 발표된 COCOMO 81 모델을 개선한 것이다. 다음 중 COCOMO II 모델의 서브모델이 아닌 것은? (4)
① 응용결합 모델(Application-Composition Model)
② 초기설계 모델(Early Design Model)
③ 재사용 모델(Reuse Model)
④ 컴포넌트 모델(Component Model)

74. COCOMO II의 서브모델이 아닌 것은? (2)
① 초기 설계 모델(early design model)
② 임베디드 모델(embedded model)
③ 응용 결합 모델(application composition model)
④ 재사용 모델(reuse model)

75. 새로운 프로젝트를 수행하기 위하여 COCOMO 모델을 적용하여 측정한 결과, 노력(Man Month) 값이 500이었고, 프로젝트 전체 기간 중에 개발자 10명이 고정 투입되었다. 프로젝트 전체 기간과 구현 단계에서 비용이 바르게 짝지어진 것은? 9단, 개발자 연봉은 3,000만원, 구현 단계는 총 노력의 40%를 차지하며, 기타 비용은 제외한다.) (3)

① 40개월, 3억원 ② 40개월, 4억원
③ 50개월, 5억원 ④ 50개월, 6억원

76. 10만 라인에 해당하는 인터넷 쇼핑몰 개발 프로젝트(A), 새로운 언어의 컴파일러 개발 프로젝트(B), 미사일 유도시스템 개발 프로젝트(C), 데이터베이스 관리 시스템(DBMS) 개발 프로젝트(D)가 있다고 가정할 때, Boehm의 연산방식을 이용한 비용 추정방식(COCOMO)에 의거 각각의 개발노력(PM)의 계산식으로 가장 적절한 것은? (1)

① PM(A) = 2.4 × $100^{1.05}$ ② PM(B) = 2.4 × $100^{1.12}$
③ PM(C) = 3.0 × $100^{1.12}$ ④ PM(B) = 3.0 × $100^{1.20}$

77. COCOMO 모델의 노력 조정 수치의 요소가 아인 것은? (5)

① 제품의 속성 ② 개발요원의 속성
③ 컴퓨터의 속성 ④ 프로젝트의 속성
⑤ 제품의 크기

78. COCOMO 비용승수에 대한 설명으로 옳지 않은 것은? (5)
① 제품 속성에는 신뢰성, 데이터베이스 크기, 제품의 복잡도가 포함된다.
② 컴퓨터 속성들은 하드웨어 플랫폼에 의해서 소프트웨어에 부관된 제한조건들이다.
③ 개인적 속성들은 프로젝트에 참여하는 사람들의 경험을 고려한 승수들이다.
④ 프로젝트 속성들은 소프트웨어 도구들의 사용, 프로젝트 개발일정, 최신 프로그래밍 기법의 사용과 관계된다.
⑤ 일정에 대한 속성은 요구된 개발 일정이 개인적 속성들이 얼마나 부합하는지에 대한 척도이다.

79. 프로젝트 비용 산정 기법에 대한 설명으로 틀린 것은? (2)
① COCOMO는 프로그램 예상 크기를 바탕으로 프로젝트 수행에 필요한 경비를 예측이 가능하다.
② COCOMO에서 Person-Month란 프로젝트 수행기간을 나타낸다.
③ Function Point 모형에서는 사용자 입력수, 사용자 출력수, 사용자 조회수, 파일수, 외부인터페이스수 등의 자료가 필요하다.
④ 계산된 FP에서는 프로젝트 수행에 필요한 비용 예측이 가능하다.
⑤ FP애서는 소프트웨어 예상 크기가 프로그램에 따라 예측이 가능하다.

80. COCOMO 모델의 노력보정계수(effort adjustment factor)에 의하면 자질이 뛰어난 프로그래머는 0.7, 자질이 다소 떨어지는 프로그래머인 경우에는 1.17, 특정 프로그래밍 언어에 대한 경험이 풍부하면 0.95, 특정 언어에 대한 경험이 거의 없으면 1.14로 각각 되어 있다. 만일 어떤 고용주가 소프트웨어 프로젝트를 수행하기 위해 앞의 정보만 이용하여 C 프로그래머를 고용하고자 한다면 어느 것이 가장 옳은 판단일까? (3)
① C 언어에 대한 경험이 있고 없고의 차이는 자질이 있고 없고의 차이보다 훨씬 많은 영향을 미치므로 자질보다는 C 언어의 구사능력이 무조건 우선돼야 한다.
② 자질이 있고 없고의 차이는 C 언어에 대한 경험이 있고 없고의 차이보다 훨씬 많은 영향을 미치므로 C 언어의 구사능력보다는 자질이 무조건 우선돼야 한다.

③ C 언어에 대한 경험이 거의 없다 하더라도 자질이 뛰어난 프로그래머를 고용하는 것이 C 언어에 대한 경험은 풍부하지만 자질 면에서 다소 떨어지는 프로그래머를 선택하는 것보다 빨리 프로젝트를 진행시킬 수 있다.
④ 자질 면에서 다소 떨어지나 C 언어에 대한 경험이 풍부한 프로그래머를 고용하는 것이 자질 면에서 뛰어나다 하더라도 C 언어에 대한 경험이 거의 없는 프로그래머를 선택하는 것보다 빨리 프로젝트를 진행시킬 수 있다.
⑤ C 프로그래머를 반드시 고용해야 다음 때문에 자질은 아무 문제가 될 수 없고 C 언어 구사 능력만을 우선적으로 고려해야 한다.

81. 1995년에 발표된 COCOM IS에 대한 설명으로 옳지 않은 것은? (3)
① 개발 초기에 LOC를 정확히 예측한다는 것은 어려운 일이므로 다양한 규모척도를 이용한다.
② 1단계는 어플리케이션 결합단계로 객체점수 혹은 어플리케이션점수라 부르는 규모척도를 이용한다.
③ 2단계는 초기 설계단계로 아키텍처를 구성하는 컴포넌트의 개수와 복잡 도를 규모척도로 이용한다.
④ 3단계는 기능점수(Function Point)와 LOC를 규모척도로 이용한다.
⑤ 재사용, 요구 분석, 요구변경을 반영할 수 있는 모델이다.

82. COCOMO의 비용 산정에 의해 개발에 소요되는 노력이 40PM(Programmer-Month)으로 계산되었다. 개발에 소요되는 기간이 5개월이고, 1인당 인건비가 100만원이라면 이 프로젝트에 소요되는 총 인건비는 얼마인가? (3)

① 2억원 ② 1억원 ③ 4천만원 ④ 2천만원

83. 소프트웨어 추정 모형(Estimation Model)이 아닌 것은? (4)

① COCOMO ② Putnam
③ Function Point ④ PERT

84. COCOMO의 프로젝트 모드가 아닌 것은? (3)

① Organic Mode ② Semi-detached Mode
③ Medium Mode ④ Embedded Mode

85. COCOMO 기법에 의한 소프트웨어 모형에 속하지 않는 것은? (2)

① Basic COCOMO
② Putnam COCOMO
③ Intermediate COCOMO
④ Detailed COCOMO

86. COCOMO 모델에 대한 설명으로 옳지 않은 것은? (2)

① Boehm이 제시한 비용 추정 모델이다.
② 비용 추정 단계 및 적용 변수의 구체화 정도에 따라 기본(Basic0, 중간(Intermediate), 진보(Advanced) 모델로 구분할 수 있다.
③ 비용 견적의 강도 분석 및 비용 견적의 유연성이 높아 소프트웨어 개발비 견적에 널리 통용되고 있다.
④ 기본(Basic) 모형은 단순히 소프트웨어의 크기와 개발 모드에 의해서 구해진다.

87. COCOMO 비용 예측 모델에 대한 설명으로 옳지 않은 것은? (3)

① B. Boehm이 제안한 원시 프로그램의 규모에 의한 비용 예측 모형이다.
② 소프트웨어의 종류에 따라 다르게 책정되는 비용 산정 방정식을 이용한다.
③ COCOMO 방법은 가정과 제약 조건 없이 모든 시스템에 적용할 수 있다.
④ 같은 규모의 프로그램이라도 그 성격에 따라 비용이 다르게 생성된다.

88. COCOMO 모델 중 기관 내부에서 개발된 중소규모의 소프트웨어로 일괄 자료 처리나 과학 기술 계산용, 비즈니스 자료 처리용으로 5만 라인 이하의 소프트웨어를 개발하는 유형은? (1)

① Organic Mode
② Semi-Detached Model
③ Semi-Embedded Model
④ Embedded Model

89. 비용 예측 방법에서 원시 프로그램의 규모에 의한 방법(COCOMO Model) 중 최대형 규모의 트랜잭션 처리 시스템이나 운영체제 등의 소프트웨어를 개발하는 유형은? (4)

① Organic 프로젝트
② Semi-Detached 프로젝트
③ Sequential 프로젝트
④ Embedded 프로젝트

90. COCOMO 모형에 대한 설명으로 옳지 않은 것은? (4)

① 산정 결과는 프로젝트를 완성하는데 필요한 Man-Month로 나타난다.
② Boehm이 고안한 개발비 산정 모델로 프로젝트의 예상되는 크기와 유형에 관한 정보가 주로 사용된다.
③ 프로젝트 특성을 15개로 나누고 각각에 대한 승수값을 제시하였다.
④ 각 모델별로 개발되어지는 프로젝트 개발 유형에 따라 Object Mode, Dynamic Mode,

Function Mode의 세 가지 모드로 구분한다.

91. 소프트웨어 프로젝트 일정이 지연된다고 해서 프로젝트 말기에 새로운 인원을 추가 투입하면 프로젝트는 더욱 지연되게 된다고 주장하는 법칙은? (3)

① Putnam의 법칙 ② Mayer의 법칙
③ Brooks의 법칙 ④ Boehm의 법칙

92. 소프트웨어 개발 시 인력관리에 대한 서술 중 옳은 것은? (5)

① 소프트웨어 개발은 주로 개발자 중심으로 진행되기 때문에 고급인력을 많이 투여할수록 품질이 높은 소프트웨어를 낮은 비용으로 개발할 수 있다.
② 소프트웨어 개발에서 인력을 많이 투여할수록 생산성을 향상시킬 수 있다.
③ 소프트웨어 개발진도가 늦어지면 관리자는 즉시 개발자들을 더 투여하여 지정된 시간 내에 제품을 개발하여야 한다.
④ 프로그램 설계단계보다 코딩단계에 더 많은 고급인력을 투여하여야 한다.
⑤ 개발진도가 늦어진 과제에 인력을 추가하면 일반적으로 과제진도가 더욱 늦어진다.

93. 브룩스(Brooks) 법칙의 의미로 가장 적절한 것은? (2)

① 프로젝트 개발에는 많은 개발자가 필요하지 않다.
② 새로운 개발 인력이 진행 중인 프로젝트에 투입될 경우 작업 적응 기간과 부작용으로 인해 빠른 시간 내에 프로젝트는 완료될 수 없다.
③ 프로젝트에는 많은 비용이 투입되어야 한다.
④ 프로젝트에 개발자가 많이 참여할수록 프로젝트의 준공기간은 지연된다.

94. 브룩스(Brooks) 법칙에 해당하는 항목은? (4)

① 소프트웨어 개발 인력은 초기에 많이 투입하고 후기에 점차 감소시켜야 한다.
② 소프트웨어 개발 노력은 40-20-40으로 해야 한다.
③ 소프트웨어 개발은 소수의 정예 요원으로 시작한 후 점차 증원해야 한다.
④ 소프트웨어 개발 일정이 지연된다고 해서 말기에 새로운 인원을 투입하면 일정은 더욱 지연된다.

95. 지연되고 있는 프로젝트에 더 많은 프로그래머를 투입할 경우 “그 프로젝트는 더 지연될 수 있다”는 브룩스의 법칙을 가장 잘 설명한 것은? (1)

① 새로 업무에 참여한 프로그래머는 시스템에 대한 이해 및 개발진행 중인 프로그래머와의 의사소통에 많은 시간이 소요되어 프로젝트의 진행을 지연시킬 수 있다.
② 효율적인 개발진행을 위해서는 초기에 가능한 한 많은 인원을 투입하여야 한다.
③ 프로젝트 계획 수립에 있어서 많은 인원으로 단기에 처리하기보다는 적은 인원으로 장기간에

수행하는 것이 유리하다.

④ 프로젝트의 효율적인 관리를 위해서는 최소한의 인원으로 운영하여야 한다.

96. “지연되고 있는 프로젝트에 보다 많은 프로그래머를 투입할수록 더 지연된다.”라는 F. Brooks가 관찰한 사실이며, 이를 Brooks의 법칙이라고 부른다. 이 법칙이 적용되는 이유에 대한 설명으로 옳지 않은 것은? (3)

① 소프트웨어 모듈 사이의 상호작용에 대한 복잡도가 증가하기 때문이다.

② 프로그래머, 프로젝트 관리자 및 고객 사이의 의사소통 필요가 증가하기 때문이다.

③ 소프트웨어 개발 프로세스들이 병렬적으로 발생하기 때문이다.

④ 프로젝트에 신규 투입된 프로그래머에게 교육이 필요하기 때문이다.

⑤ 소프트웨어 개발 프로세스들 간의 상호의존성이 존재하기 때문이다.

비용산정 관련 문제 해설

36. (3) MM = 개발기간 ×투입인원 = 총 LOC/1인당 월평균 생산 코드라인수 = 총 LOC/생산성 = 35,000/500=70
개발기간 = MM/인원 = 70/5=14

37. (2) 추정LOC = (낙관적 + 4보통LOC + 비관적LOC)/6 = (100+4×100+200)/6=310

38. 생산성 = 총 LOC/PM 이므로 PM = 33,600/600=56, 개발비용 = PM×1인당 월 평균 인건비 = 56×480만원=268,800,000원 (3)

40. (4) MM = 총 LOC/1인당 월평균 생산코드 라인수 = 500,000/(50×50) = 200
개발기간 = MM.개발자수 = 200/8 = 25개월

56. 1인당 월평균 생산성 = FP/MM = 7,500/150 = 50FP/월 (2)

57. (3) EI = (단순 3×4) + (보통 4×1) + (복잡 6×0) = 16
∴ FP = 16+28+26+100+30 = 200
∴ C++ 200×50=10,000 SLOC
C 200×130=26,000 Pascal 200×90=18,000 VB 200×30=6,000

75. (3) MM = 개발기간 × 인원, 500 = 개발기간 × 10명 ∴ 개발기간 = 50개월
개발비용 = MM × 1인당 인건비(단위비용)
연봉이 3000만원이므로, 1인당 월급여는 250만원
총 인건비 = 500(50개월×10명) × 250만원 = 개발비용 = 500(50개월×10명) × 250만원 = 1,250,000,000
구현단계는 총 노력의 40%임로 1,250,000,000 × 0.4 = 500,000,000

Chapter 4. 연습문제

1. 요구분석에 대한 설명으로 옳지 않은 것은? (3)
   ① 각 요구사항은 명확하고, 구체적이고, 정확하고, 검증이 가능하도록 정의되고 기술되어야 한다.
   ② 요구사항은 고품질의 소프트웨어를 개발하고 검증할 수 있는 기초를 제공한다.
   ③ 고객과 개발자가 서로 당연한 것으로 인정하는 요구사항은 생략하여도 무방하다.
   ④ 요구사항은 크게 기능적인 요구사항과 성능, 신뢰성, 가용성, 보안성, 안전성 등의 비기능적 요구사항으로 분류된다.

2. 요구사항 명세화 원리에 가장 적합하지 않은 것은? (4)
   ① 소프트웨어가 동작하는 주변 환경을 정립하라.
   ② 명세서의 내용과 구조를 변경사항에 쉽게 대처할 수 있도록 작성하라.
   ③ 자료와 외부자극에 대해 반응하는 시스템 행위 모델을 개발하라.
   ④ 구현 작업을 생각하여 알고리즘을 상세히 기술하라.

3. 요구공학(requirement engineering)에 대한 설명으로 옳지 않은 것은? (2)
   ① use case는 요구사항을 수집하고 분석하는 과정에서 사용된다.
   ② 요구사항을 정의하는 과정에서 그것을 어떻게 구현해야할지 명기해야 한다.
   ③ 분석패턴은 특종 응용도메인에서 요구단위로 재사용될 수 있다.
   ④ 타당성 조사는 요구공학 초기의 주요활동 중 하나이다.

4. 요구공학(requirement engineering) 프로세스의 첫 단계인 타당성 조사(Feasibility Study)에 대한 설명으로 틀린 것은? (3)
   ① 시스템이 조직의 전체 목표에 부합하는지에 대한 평가가 이루어진다.
   ② 타당성 조사 보고서가 작성된다.
   ③ 프로토타이핑과 구조적 분석 방법이 사용된다.
   ④ 현재의 기술과 주어진 예산과 일정 내에서 개발될 수 있는가에 대한 평각 이루어진다.

5. 요구분석과 정의단계에서 프로토타이핑을 적용하여 얻는 이점과 거리가 먼 것은? (1)
   ① 시스템 개발비용을 효율적으로 감소시킬 수 있다.
   ② 누락된 사용자 서비스를 찾아낼 수 있다.
   ③ 응용프로그램의 타당성이나 유용성을 시연하는데 이용할 수 있다.
   ④ 시스템 기능 시연을 통해 개발자와 사용자간의 오해를 줄일 수 있다.

6. 한 학급의 비디오 감시 시스템을 개발하는데, 다음의 요구사항 중에서 비기능적 요구사항에 해당하는 것으로 묶은 것은? (4)

가) 개발할 비디오 시스템은 95%의 가용성을 가지도록 한다. 나) 비디오 감시 카메라가 일정한 시간 간격마다 비디오 상에 담은 화면을 출력 저장하여 보관하게 만든다. 다) 비디오 감시 카메라에 담은 화면을 선명하게 출력 화면으로 출력한다.

① 가, 나 ② 가, 나, 다
③ 나, 다 ④ 가, 다

7. 다음은 도서관리 시스템에서 필요한 요구사항을 설명하고 있다. 그 요구사항에 대한 예가 잘못 연결된 것은? (3)
① 성능 요구사항 – '최대 오류 발생시간은 4주이다.'
② 운영 요구사항 – '도서관리 기능은 3번 키를 입력해야 한다.'
③ 검증 요구사항 – '도서관리 시스템은 도서 대출, 반납 기능을 제공해야 한다.'
④ 인터페이스 요구사항 – '도서 대출 기능과 도서 반납 기능은 도서 정보를 교환한다.'

8. 비기능적 요구사항에 대한 명세로 적절하게 표현되지 않은 것은? (2)
① 시스템은 초당 10개 이상의 트랜잭션을 처리할 수 있어야 한다.
② 시스템은 20세 이상의 사업장의 작업자에게 사용편리성을 제공해야 한다.
③ 시스템에 설치되는 모듈 A의 크기는 500 Kbyte 이하이어야 한다.
④ 시스템의 평균 고장시간은 시간당 0.1초 이하를 유지해야 한다.

9. 요구분석 명세서의 평가기준에 대한 설명으로 옳은 것은? (2)
① 무결성 – 요구분석은 모호한 점이 업도록 간결하고 명쾌하게 작성해야 한다.
② 일관성 – 요구분석서 안에 서로 모순되는 부분이 없어야 한다.
③ 명확성 – 사용자의 요구를 오류 없이 완벽하게 반영하고 있어야 한다.
④ 추적가능성 – 개발된 시스템이 요구분석에 기술된 내용과 일치해야 한다.

10. 다음은 요구사항 명세서가 지녀야 할 기본조건을 설명한 것이다. 명세서가 지녀야 할 기본조건의 설명으로 가장 적절하지 않은 것은? (3)
① 설계 과정을 이한 문제의 정의에 도움을 줄 수 있을 것
② 소프트웨어가 올바르다고 판단하기 위한 수단으로 사용할 수 있을 것
③ 개발자 중심으로 만들어서 구현 시 즉시 사용할 수 있을 것
④ 소프트웨어 제품 및 개발과정을 공학화하는 핵심이 될 수 있을 것

11. 요구사항 문서의 점검 활동과 그 내용이 바르게 연결된 것은? (1)
① 일관성 점검 - 문서에 있는 요구사항이 상충되지 않는지 검사
② 추적가능성 점검 - 요구사항 문서의 내용이 여러 의미로 해석되는 모호한 점이 있는지 검사
③ 증명가능성 점검 - 요구사항 문서가 모든 기능을 정이하고 시스템 사용자가 의도한 제약조건을 모두 포함하는지 검사
④ 실현성 점검 - 미래의 기술을 사용하여 요구사항이 실제로 구현될 수 있는지 검사

12. 다음 중 요구명세서에 포함될 사항과 가장 거리가 먼 것은? (4)

① 기능적 요구
② 설계 제약
③ 외부 인터페이스
④ 소프트웨어 구조

13. 요구사항 모델링 기법과 가장 관련이 적은 것은? (3)

① UML(Unified Modeling Language)
② SDL(Specification and Design Language)
③ CPM(Critical Path Method)
④ SCR(Software Cost Reduction)

14. 요구공학의 공정 순서를 바르게 나열한 것은? (4)

(ㄱ) 요구사항 분석 (ㄴ) 요구사항 검증 (ㄷ) 요구사항 명세 (ㄹ) 요구사항 추출

① ㄱㄴㄷㄹ
② ㄷㄱㄹㄴ
③ ㄹㄱㄴㄷ
④ ㄹㄱㄷㄴ

15. 요구사항 명세기법에 대한 설명으로 옳지 않은 것은? (4)
① 시스템의 요구사항 특성을 명세하기 위해 자연어를 기반으로 하는 서술형태 또는 그림 중심으로 명세하는 것을 비정형 명세기법이라 한다.
② 수학적 원리와 표기법을 적용하여 시스템의 요구특성을 보다 정확하고 간결한 상태로 명세하는 것을 정형 명세라 한다.
③ 비정형 명세는 요구사항에 대한 불충분한 명세, 일관성의 결여, 내용의 모호성 여부를 검증하기 어렵다.
④ 안전성, 신뢰성, 보안성과 같은 특성이 매우 중요한 시스템 개발 분야에서 비정형 명세기법의 사용이 증가하고 있다.

16. 시스템의 기능적 요구를 파악하기 위한 질문만을 모두 고른 것은? (2)

(ㄱ) 입력, 출력이 무엇이며 어떤 형태를 갖는가? (ㄴ) 시스템이 무엇을 하는가? (ㄷ) 어떤 하드웨어가 사용될 것인가? (ㄹ) 어떤 데이터가 얼마나 오랜 기간 유지되어야 하는가? (ㅁ) 얼마나 쉽게 위치나 플랫폼을 변동할 수 있는가?

① ㄱㄴㄷ
② ㄱㄴㄹ
③ ㄴㄷㅁ
④ ㄴㄹㅁ

17. 비기능적 요구사항을 제품 요구사항, 조직 요구사항, 외부 요구사항으로 구분할 때 제품 요구사항에 해당하지 않는 것은? (4)

① 경험이 많은 사용자는 1시간의 교육을 받으면 시스템을 사용할 수 있어야 하고, 사용자의 일별 평균 에러 발생 횟수가 1.0회를 초과하면 안 된다.
② 시스템의 이용가능성은 99.0% 이상이어야 한다.
③ 특정 하드웨어에 종속적인 기능의 비율은 1.0% 미만이어야 한다.
④ 시스템은 고객의 개인정보보호를 위해, 사용자 이름과 아이디를 제외한 어떤 정보도 시스템 관리자에게 제공하지 않아야 한다.

18. 다음 설명에 해당하는 사용자 요구사항 추출(elicitation) 방법은? (4)

이 방법의 목적은 소프트웨어 엔지니어의 아이디어에 대한 피드백을 조기에 받아서 요구사항을 취합하는 것이다. 이 방법의 가장 단순한 형태는 시스템이 수행될 때 무엇이 일어날 지를 설명하기 위하여 종이에 화면 순서를 기술하여 고객과 사용자에게 보여주는 것이다.

① 관찰
② 인터뷰
③ 브레인스토밍
④ 프로토타이핑

19. 다음은 소프트웨어 개발 시 요구사항 명세서에 대한 설명이다. 요구사항 명세서에 대한 설명으로 가장 부적절한 것은? (4)

① 요구사항 명세서는 그것을 사용하는 사람과 만드는 사람 양쪽 모두에게 명백해야 한다.
② 요구사항 명세서는 완전하여야 한다. 따라서 기능성, 성능, 제약사항과 같이 모든 중요한 것은 문서화되어야 한다.
③ 요구사항 명세서는 명세의 다른 부분들이 서로 충돌되어서는 안 되며, 일관되어야 한다.
④ 요구사항 명세서는 요구사항이 충돌하는지 아닌지를 결정하기 위한 제한된 프로세스가 반드시 있어야 하지만 그것을 증명할 필요는 없다.

20. 요구분석 명세서의 평가기준에 대한 설명으로 옳은 것은? (2)

① 무결성 – 요구분석은 모호한 점이 없도록 간결하고 명쾌하게 작성해야 한다.
② 일관성 – 요구분석서 안에 서로 모순되는 부분이 없어야 한다.
③ 명확성 – 사용자의 요구를 오류 없이 완벽하게 반영하고 있어야 한다.
④ 추적가능성 – 개발된 시스템이 요구분석에 기술된 내용과 일치해야 한다.

21. 요구사항 분석 과정에 대한 설명 중 옳지 않은 것은? (5)

① 의뢰자가 무엇을 원하는지 결정하기 위해서 다양한 방법을 사용해야 한다.
② 시스템의 설계 이전에 문제에 대한 분석이 이루어져야 한다.
③ 시스템이 어떻게 구현될 지에 대한 것은 언급하지 않는다.
④ 요구사항 추출, 분석 단계가 끝나면 요구사항의 정의, 명세 단계가 이루어진다.
⑤ 유스 케이스는 시스템의 기능적 및 비기능적 요구사항을 결정해 준다.

22. 요구 분석가가 갖추어야 할 능력과 거리가 먼 것은? (1)

① 요구사항에 대한 경제적 가치의 계산 능력
② 추상적 개념을 포착하여 논리적 영역으로 재조직할 수 있는 능력
③ 대립되거나 혼동되는 의뢰자로부터 적절한 사실을 흡수할 수 있는 능력
④ 문서 및 구두 형식의 대화 능력
⑤ 세세한 것뿐만 아니라 전체를 볼 수 있는 능력

23. 사용자의 요구사항 분석 작업이 본질적으로 어려운 이유가 아닌 것은? (4)

① 개발자와 사용자간의 의사소통은 본질적으로 어렵다.
② 개발하고자 하는 시스템 자체가 복잡하다.
③ 거듭되는 수정 요구와 상반된 요구들을 수용하는 통제 기술이 필요하다.
④ 요구사항 분석을 위한 자동화 도구 및 방법들이 없다.
⑤ 요구사항은 완전하고 일관성 있게 작성되어야 한다.

24. 다음의 요구사항 중 기능적 요구사항이 아닌 것만을 모두 나열한 것은? (5)

가) 사용자가 '지불' 버튼을 누른 후 5초 이내에 응답이 있어야 한다. 나) 제품 구입이 수행된 후 재고 데이터베이스가 즉각 수정되어야 하며 해당 품목의 수량이 미리 정한 값 이하로 떨어지면 경고 메시지를 보여주어야 한다. 다) 사용자는 자신의 개인 정보가 유출되지 않도록 구매를 완료한 후에는 로그아웃 하여야 한다.

① 가 ② 나
③ 다 ④ 가, 나
⑤ 가, 다

25. 다음 보기의 내용들이 포함되는 요구분석 명세서의 항목은? (3)

• 시간적 요구: 반응시간, 처리 소요 시간, 처리율 등 • 효율적 요구: 기억장치 규모, 통신 대역폭 등 • 기타: 보안성, 신뢰성 등

① 개발 운용 및 유지보수 환경
② 기능 요구사항
③ 성능 요구사항
④ 변경 및 개선 예정 사항
⑤ 시스템 환경

26. 요구분석 명세서가 갖추어야 할 특징이 아닌 것은? (5)

① 완전성
② 정확성
③ 일치성
④ 명확성
⑤ 구현과의 연계성

27. 요구분석 작업을 위하여 조사하여야 할 사항만으로 이루어진 것은? (2)

① 테스트 일정, 조직 구조, 시스템 성능
② 시스템 환경, 사용자의 작업, 사용자의 타입
③ 언어, 플랫폼, 경쟁력
④ 검토, 정적 분석, 정확도
⑤ 비기능적 요소, 기능적 요소, 단위 테스트

28. 시스템 개발을 위한 첫 단계는 사용자의 요구나 시스템에 대한 분석이라고 할 수 있다. 이 중 사용자의 요구분석을 위해 주로 사용하는 기법이 아닌 것은? (4)

① 사용자 면접
② 현재 사용 중인 각종 문서 검토
③ 설문조사를 통한 의견 수렴
④ 통제 및 보안 분석

29. 분석가가 갖추어야 할 능력 중 가장 중요한 것은? (4)

① 추상적인 개념을 파악하여 논리적인 구성 요소로 분해할 수 있는 능력
② 서로 상반되고 모호한 정보로부터 필요한 사항을 수렴할 수 있는 능력
③ 관련된 하드웨어와 소프트웨어에 관한 최신 기술
④ 거시적 관점에서 세부적인 요소를 관찰할 수 있는 능력

30. 소프트웨어의 전통적 개발 단계 중 요구분석 단계에 대한 설명으로 옳지 않은 것은? (4)
① 프로젝트를 이해할 수 있는 개발의 실질적인 첫 단계이다.
② 현재의 상태를 파악하고 문제를 정의한 후, 문제 해결과 목표를 명확히 도출하는 단계이다.
③ 소프트웨어가 가져야 될 기능을 기술하는 간계이다.
④ 고품질의 소프트웨어를 개발하기 위해 소프트웨어의 내부 구조를 기술하는 단계이다.

31. 사용자의 요구사항 분석 작업이 어려운 이유와 거리가 먼 것은? (2)
① 개발자와 사용자 간의 지식이나 표현의 차이가 커서 상호 이해가 쉽지 않다.
② 사용자의 요구는 예외가 거의 없이 열거와 구조화하기 어렵지 않다.
③ 사용자의 요구사항이 모호하고 부정확하며, 불완전하다.
④ 개발하고자 하는 시스템 자체가 복잡하다.

32. 비기능 요구사항에 대한 설명으로 옳지 않은 것은? (4)
① 예산의 제약, 조직의 정책, 다른 소프트웨어와 하드웨어 시스템과의 상호 운영성, 안전성 규칙과 프라이버시 보호법과 같은 사용자의 필요에 의해 발생한다.
② 요구사항과 목표를 혼합한 문서를 사용하여 비기능적 요구사항을 표현한다.
③ 시스템에서 제공되는 서비스나 기능에 대한 제약이다.
④ 시스템이 제공해야 하는 서비스와 시스템이 특정 입력에 대해 어떻게 반응하는지, 시스템이 특정 상황에서 어떻게 동작해야 하는지에 관한 사항이다.

33. 요구분석명세서를 작성할 때 고려해야 할 사항으로 옳지 않은 것은? (1)
① 시스템의 기능을 구현하는 방법에 대하여 중점적으로 기술한다.
② 제안된 시스템에 영향을 주는 제약조건을 기술한다.
③ 시스템의 인수를 위한 테스트 기준을 제공한다.
④ 소프트웨어의 품질 기준에 대한 우선순위를 정한다.

34. 요구사항 품질 속성에 대한 설명으로 옳지 않은 것은? (3)
① 기능성(functionality) - 요구분석명세서가 '어떻게' 보다 '무엇을'에 관점을 두고 기술되어야 함을 의미한다.
② 명확성(unambiguity) - 요구사항이 모든 이해관계자들에 의해 한 가지 의미로 해석되어야 한다.
③ 완전성(completeness) - 요구사항을 바로 구현할 수 있도록 자료구조나 알고리즘이 명시되어야 한다.
④ 일관성(consistency) - 요구사항들이 서로 모순되지 않아야 한다.

35. 요구공학에서 요구사항을 명확하게 정의하기 위해 사용되는 정형 명세(Formal Specification)에 대한 설명 중 가장 적합하지 않은 것은? (3)

① 비정형 명세에 비해서 누락(Omission), 불일치(Inconsistency) 감사가 쉽다.
② 비정형 명세에 비해서 표현이 간결하다.
③ 사용자 인터페이스와 상호작용을 명세하는 것에 가장 적합하다.
④ 정형 명세는 수학적 개념에 기초한다.

Chapter 5. 연습문제

1. 객체지향 개발 방법의 장점이라고 보기 힘든 것은? (3)

① 재사용성 ② 신속한 설계
③ 실행의 효율성 ④ 확장성

2. 객체지향 소프트웨어공학의 특징이 아닌 것은? (1)
① 구조적 소프트웨어공학에 비해 메모리양이 줄고 수행시간이 빨라져서 개발 결과물의 성능이 향상된다.
② UML은 OMG에서 표준으로 제정한 OOSE의 분석 및 설계 표기법이다.
③ 하나의 클래스로부터 많은 객체를 인스턴스화 할 때 유용한 기술이다.
④ 객체 클래스 상속 기능을 이용한 개발 기술이다.

3. 다음 중 객체지향 방법론의 특징으로 가장 옳지 않은 것은? (4)
① 실세계의 개념을 객체라는 개념으로 개발한다.
② 데이터 사이의 존재하는 복잡한 관계성은 일반화, 집합성, 실체화 등을 통해 개발한다.
③ 시스템 설계 및 구현 시 보다 빠른 프로토타이핑을 할 수 있다.
④ 크고 복잡한 프로그램을 여러 개의 작은 서브프로그램으로 나누고 각 서브프로그램을 다시 더 작은 서브프로그램으로 나누어 개발한다.

4. 다음 객체지향 개발 방법론에 관한 설명 중 틀린 것은? (3)
① 객체지향 개발 방법은 객체, 객체의 속성, 동작, 클래스, 객체 사이의 관계 등을 기본 개념으로 하고 있다.
② 객체지향 분석 기법은 기존의 분석 기법에 비해 실세계의 현상을 보다 정확히 모델링 할 수 있다.
③ 객체지향 방법론은 하향식 문제 접근 방법으로 기능 중심의 정보 모델링이다.
④ 객체지향 개발 방법을 잘 활용하기 위해서는 소프트웨어 개발과정에 대한 이해와 정보모델, 동적모델, 기능모델에 대한 지식이 있어야 하며 모델링 사이의 연관성을 바탕으로 모델링의 결과를 통할 할 수 있어야 한다.

5. 객체지향 개발 기술 및 개발 방법론에 관한 설명 중 틀린 것은? (2)
① 객체지향의 기본 원리로는 추상화, 캡슐화, 상속성 등이 있다.
② 객체지향 프로그래밍 언어는 소프트웨어의 재사용 및 구조화가 지원이 잘 되어 컴퓨터 하드웨어 시스템을 가장 효율적으로 사용할 수 있다.
③ 객체지향 방법론은 소프트웨어 위기를 극복하기 위한 대안으로 발전했으며, 활성화가 예상되

는 CBD 방법론으로 더욱 발전하고 있다.

④ 객체지향 분석은 기존의 방법에 비해 실세계를 보다 정확히 모델링할 수 있고, 분석과 설계의 표현에 차이점이 없어 시스템 개발을 용이하게 해준다.

6. 절차식 패러다임과 객체지향 패러다임에 대한 설명으로 옳지 않은 것은? (2)

① GUI 기반의 프로그램을 개발하는데 객체지향 패러다임이 보다 더 적합하다.

② 절차식 패러다임은 하향식 설계로 변경이 쉽다.

③ 절차식 패러다임의 분석, 설계 단계의 모델이 서로 다르지만 객체지향 패러다임은 객체라는 관점의 통합된 모델을 이용한다.

④ 객체는 잘 구별되는 단위들로서 잘 정의된 인터페이스를 이용하여 상호작용한다.

7. 다음 객체지향과 관련된 설명 중 옳지 않은 것을 모두 나열한 것은? (1)

가) 인스턴스란 클래스와 동일한 개념이라고 할 수 있다. 나) 상속이란 하위클래스가 가지고 있는 속성을 물려받는 것을 의미한다. 다) 다형성이란 하나의 객체가 메시지에 따라 다르게 응답하는 것을 의미한다.

① 가, 나, 다　　② 가, 다

③ 나, 다　　④ 가, 나

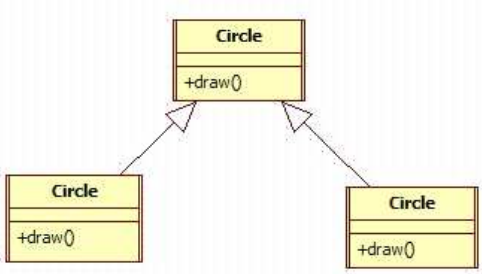
8. 다음 설명에 해당하는 용어는? (3)

클래스나 컴포넌트가 제공하는 서비스들을 나타내는 오퍼레이션들의 집합을 나타낸다. 함수들의 시그니처만 명세할 뿐 함수 구현은 전혀 존재하지 않으며, 속성 값도 존재할 수 없다. 구현은 상속한 클래스에서 하며, 자체는 객체를 생성할 수 없다. 클래스와의 관계는 실체화 관계로 표현한다.

① 추상 클래스　　② 액티브 클래스

③ 인터페이스　　④ 상태머신

9. “Polygon” 클래스가 갖고 있는 “draw” 함수를 하위클래스인 “Triangle”과 “Circle”에서 재정의하여 서로 다른 기능을 수행하도록 한다. 이러한 객체지향의 특징을 무엇이라 하는가? (3)



① 추상화(Abstraction)　② 캡슐화(Encapsulation)
③ 다형성(Polymorphism)　④ 상속성(Inheritance)

10. 다음 설명과 가장 관련이 있는 객체지향 기법의 원리는? (3)

> 라인 그래프, 파이 차트, 히스토그램, 키비아트 다이어그램의 4가지 그래프를 그려야 하는 응용 프로그램이 있다고 하자. 이 프로그램을 설계하기 위해 일반적인 클래스로 graph라는 클래스를 정의하고 각각의 그래프를 나타내는 클래스를 나타내는 클래스들을 graph 클래스의 서브클래스로 정의한다. 그리고 graph 클래스와 각각의 서브클래스에 그래프를 그리는 draw라는 메소드를 정의한다. 그러면 객체는 이들 서브클래스들의 인스턴스인 객체들 어떤 것에도 draw라는 메시지를 보낼 수 있고 메시지를 받은 객체는 적절한 그래프를 생성하기 위해 자기 자신의 draw 메소드를 호출한다.

① 추상화(Abstraction)　② 캡슐화(Encapsulation)
③ 다형성(Polymorphism)　④ 정보은닉(Information hiding)

11. 객체지향 기법에서 다형성에 대한 설명으로 옳은 것은? (4)
① 하나의 클래스로부터 여러 개의 객체를 생성할 수 있다.
② 하나의 베이스 클래스로부터 여러 개의 파생 클래스를 가질 수 있다.
③ 하나의 클래스에는 여러 개의 멤버함수를 가질 수 있다.
④ 하나의 이름을 가진 함수를 다른 여러 가지 방법으로 구현할 수 있다.

12. 상위 클래스는 하위 클래스로 상속이 가능한데, 이런 관계성을 무엇이라 하는가? (2)

① 집단화　② 일반화
③ 정규화　④ 연관화
⑤ 패키지화

13. 객체지향 패러다임의 특성 중 유지보수의 어려움으로 작용하지 않는 것은? (1)

① 캡슐화(encapsulation)　② 다형성(polymorphism)
③ 상속(inheritance)　④ 오버로딩(overloading)
⑤ 다중 상속(multiple inheritance)

14. 객체지향 기법에서 다형성에 대한 설명으로 옳은 것은? (4)
① 하나의 클래스로부터 여러 개의 객체를 생성할 수 있다.
② 하나의 베이스 클래스로부터 여러 개의 파생 클래스를 가질 수 있다.
③ 하나의 클래스에는 여러 개의 멤버함수를 가질 수 있다.

④ 하나의 이름을 가진 함수를 다른 여러 가지 방법으로 구현할 수 있다.

15. 객체지향 방법에서 객체를 식별(identification)하는 방법 중 옳지 않은 것은? (5)
① 사건, 상호교류, 위치 등의 구체적인 개체를 사용한다.
② 시스템의 전반적인 행동방식을 이해하도록 하는 행동적(behavioral) 접근법을 사용한다.
③ 시나리오 중심적 분석을 사용한다.
④ 과거의 경험과 기술을 적극 활용한다.
⑤ 객체와 속성은 동사로, 연산과 서비스는 명사로 식별하는 문법적인 분석을 이용한다.

16. 객체지향 기법에서 오브젝트의 상태는 무엇을 파악함으로써 알 수 있는가? (2)

① 오퍼레이션 ② 애트리뷰트
③ 트리거 ④ 메소드

17. 객체지향 시스템에서 전통적 시스템의 함수 또는 프로시저에 해당하는 연산 기능을 무엇이라고 하는가? (4)

① 메시지 ② 모듈
③ 패키지 ④ 메소드

18. 객체지향 기법에서 메소드는 어느 시점에서 시작되어지는가? (4)
① 사용자 명령어가 입력될 때
② OS에 의하여 인터럽트가 감지될 때
③ 특별한 데이터 값을 만날 때
④ 오브젝트로부터 메시지를 받을 때

19. 객체지향 기법에 대한 설명으로 거리가 먼 것은? (1)
① 프로시저에 근간을 두고 프로그래밍을 구현하는 기법이다.
② 현실 세계를 모형화하여 사용자와 개발자가 쉽게 이해할 수 있다.
③ 소프트웨어의 재사용율이 높아진다.
④ 소프트웨어의 유지보수성이 향상된다.

20. 다음 중 ( ) 안에 공통으로 들어갈 내용은? (2)

모든 객체들은 ( )의 멤버이고, 그 ( )에 대하여 이미 정의된 개발 자료 구조와 연산이 상속된다. 그 때문에 개별 객체는 ( )의 인스턴스가 된다.

① 메시지 ② 클래스
③ 상속성 ④ 정보

21. 객체지향 개념 중 하나 이상의 유사한 객체들을 묶어 공통된 특성을 표현한 데이터 추상화를 의미하는 것은? (2)

① 메소드 ② 클래스
③ 상속성 ④ 추상화

22. 객체지향 기법에서 메시지의 전달은 어떻게 이루어지는가? (3)
① 애트리뷰트(attribute)에서 애트리뷰트로
② 오브젝트에서 애트리뷰트로
③ 오브젝트에서 오브젝트로
④ 클래스에서 데이터로

23. 오브젝트가 메시지를 받으면 어떤 행위가 발생하는가? (2)
① 애트리뷰트를 부른다(invoke)
② 메소드를 부른다.
③ 연관 없이 흘러 보낸다.
④ 메시지를 되돌려 보낸다.

24. 객체지향 개념에서 오퍼레이션은 무엇을 변화시키는가? (1)

① 애트리뷰트 ② 클래스
③ 오브젝트 ④ 메시지

25. 객체지향 프로그램의 장점으로 거리가 먼 것은? (2)
① 자연적인 모델링이 가능하다.
② 실행 속도가 빨라진다.
③ 소프트웨어의 재사용률이 높아진다.
④ 소프트웨어의 유지보수성이 향상된다.

26. 객체지향의 기본 개념 중 객체가 메시지를 받아 실행해야 할 객체의 구체적인 연산을 정의한 것은? (1)

① 메소드 ② 추상화
③ 상속성 ④ 캡슐화

27. 기존의 소프트웨어 공학 기법들과 차별화될 수 있는 객체지향 개념이 아닌 것은 어느 것인가? (4)

① 캡슐화 ② 상속성
③ 다형성 ④ 모듈화

28. 객체지향 기법에서 상속의 결과로 얻을 수 있는 가장 중요한 이점은? (3)
① 모듈 라이브러리의 재이용
② 객체지향 DB를 사용할 수 있는 능력
③ 클래스와 오브젝트들을 재사용할 수 있는 능력
④ 프로젝트들을 보다 효과적으로 관리할 수 있는 능력

29. 객체지향 기술에서 다형성(polymorphism)의 의미로 가장 적절한 것은? (4)
① 다중 메시지를 수행하기 위하여 이용되는 기술
② 동일한 일을 수행하기 위하여 상이한 메소드 이름을 이용하는 능력
③ 상이한 일을 수행하기 위하여 동일한 메시지 형태를 이용하는 능력
④ 많은 상이한 클래스들이 동일한 메소드 이름을 이용하는 능력

30. 객체지향 소프트웨어 공학의 상속성에 대해 바르게 설명한 것은? (1)
① 상위 클래스의 메소드와 속성을 하위 클래스가 물려받는 것을 말한다.
② 데이터와 데이터를 조작하는 연산을 하나로 묶는 것을 말한다.
③ 객체 클래스로부터 만들어진 하나의 인스턴스이다.
④ 변수가 취할 수 있는 여러 가지 특성 중의 하나를 결정 받는 것을 말한다.

31. 다음 설명에 적합한 용어는? (3)

원(Circle)의 면적을 구하는 getArea() 함수를 갖고 있는 객체, 직사각형(Rectangle)의 면적을 구하는 getArea() 함수를 갖고 있는 객체, 삼각형(Triangle)의 면적을 구하는 getArea() 함수를 갖고 있는 객체들은 getArea()라는 함수를 가지고 있으므로 면적을 구하기 위해서는 'getArea()'란 메시지를 받으면 수행된다. 그러나 각각의 함수에서 면적을 구하는 방법은 모두 다를 것이다.

① 캡슐화(Encapsulation)　② 상속성(Inheritance)
③ 다형성(Polymorphism)　④ 추상화(Abstraction)

32. 다음 중 UML 다이어그램이 아닌 것은? (2)

① 클래스 다이어그램(class diagram)　② 속성 다이어그램(attribute diagram)
③ 사용사례 다이어그램(use-case diagram)　④ 순차 다이어그램(sequence diagram)

33. UML 다이어그램 중 분석 단계에서 사용하기에 가장 적합하지 않는 것은? (2)

① 액티비티 다이어그램　② 컴포넌트 다이어그램
③ 유스케이스 다이어그램　④ 클래스 다이어그램

34. UML은 시스템의 정적인 부분과 동적인 부분을 표현하기 위하여 여러 다이어그램들을 제공한다. 정적인 부분을 표현하는 다이어그램들은 시스템의 구조를 나타내기 위해서 사용되며, 동적인 부분을 표현하는 다이어그램들은 시스템의 행위를 나타내기 위해 사용된다. 다음 중 성격이 다른 다이어그램은? (1)

① 클래스 다이어그램(class diagram)　② 통신 다이어그램(communication diagram)
③ 상태 다이어그램(state diagram)　④ 활동 다이어그램(activity diagram)

35. UML 다이어그램에 대한 설명 중 적합하지 않은 것은? (2)

① 유스케이스 다이어그램은 사용자 요구사항을 분석하기 위한 도구이다.
② 시퀀스 다이어그램은 시간이 지남에 따른 클래스간의 상호작용을 기술한다.
③ 클래스 다이어그램은 시스템이 갖는 정적인 정보들의 관계를 설명해 준다.
④ 상태 다이어그램은 객체가 보유하는 상태와 상태가 전이하는 제어흐름을 표현해 준다.

36. UML에서 유스케이스를 사용하는 이점은? (1)

① 사용자 관점의 요구추출이 용이하다.
② 재사용성이 향상된다.
③ 시스템의 시험이 쉽다.
④ 프로젝트 관리가 용이하다.

37. 유스케이스 다이어그램에 대한 설명으로 옳지 않은 것은? (4)

① 문제 영역에 대해 개발자는 사용자 입장에서 시스템의 기능 요구를 추출하고 분석하는데 사용한다.

② 유스케이스는 분석, 설계, 검사 등 전 과정에 이용될 수 있다.
③ 전체 시스템을 유스케이스 단위로 분할하고 분할된 유스케이스는 다시 분할하여 시스템에 대한 전반적인 이해를 높인다.
④ 유스케이스 다이어그램은 클래스의 내부적인 구조나 클래스들 간의 관계에 의한 정적인 관계를 나타내며 시간적인 정보를 보여주지는 않는다.

38. 객체지향 분석 설계에서 유스케이스가 만족해야 하는 특성을 설명하는 것 중 가장 거리가 먼 것은? (4)
① 하나의 유스케이스는 일련의 액션들로 표현된다.
② 유스케이스는 시스템이 제공하는 기능이다.
③ 유스케이스는 액터가 관찰할 수 있는 결과를 산출한다.
④ 하나의 유스케이스는 여러 variants를 포함하지 않는다.

39. UML의 유스케이스 다이어그램에서 표현되지 않는 것은? (4)

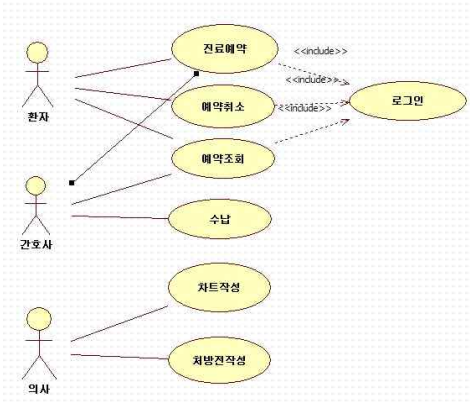
① 행위자(Actor)
② 일반화(Generalization)
③ 포함(Include)
④ 구획면(Swimlane)

40. 유스케이스 다이어그램에 대한 설명으로 옳지 않은 것은? (2)
① 유스케이스 다이어그램을 통하여 시스템의 범위를 파악할 수 있다.
② 시스템과 상호작용을 하는 외부시스템은 액터로 파악해서는 안 된다.
③ 액터가 인식할 수 없는 시스템 내부의 기능을 하나의 유스케이스로 파악해서는 안 된다.
④ 유스케이스는 이름으로부터 해당 유스케이스가 나타내는 시스템의 기능을 명확하게 표현할 수 있어야 한다.

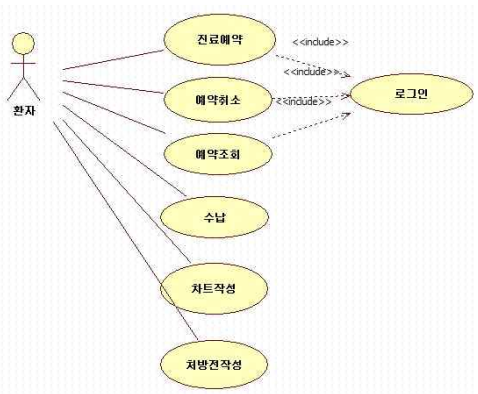
42. 다음은 병원의 “진료 예약 시스템”에 대한 기능 요구사항 명세서이다. 이를 통해 작성된 유스케이스 다이어그램으로 적합한 것은? (1)

A병원은 환자들의 진료를 보다 원활히 하기 위해서 진료 예약 시스템을 개발하기로 하였다, 환자들은 진료를 받기 전에 예약을 할 수 있으며, 당일에는 예약을 할 수 없다. 예약한 환자들 가운데 사정상 진료 받기 어려운 경우에는 예약을 취소할 수 있으며, 로그인 상태에서 자신의 예약정보를 조회할 수도 있다. 환자가 예약 관련 업무를 수행하기 위해서는 간호사가 환자 대신에 예약 관련 업무를 수행할 수도 있다. 의사는 환자를 진료하면서 환자의 진료정보를 시스템에 기록하고, 처방전을 작성한다. 진료가 끝나면, 간호사는 처방전을 환자에게 제공하고, 환자에게 진료비를 청구한다. 환자는 수납이 끝나면 다음 진료를 예약한다.

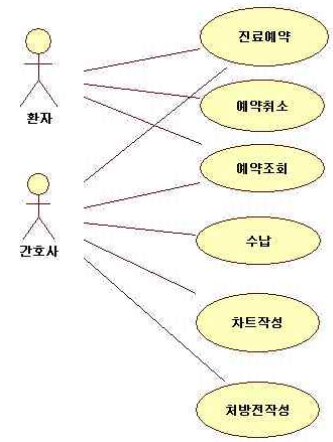
①



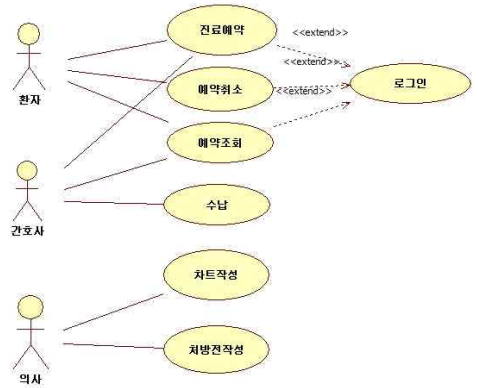
②



③



④



43. 다음은 유스케이스 모델이다. 설명이 틀린 것은? (1)

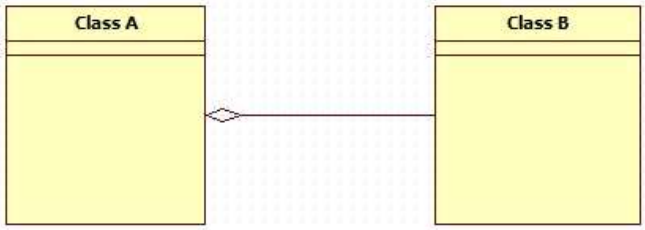
① include는 확장 유스케이스에서 어떤 조건을 만나면 선택적으로 자신의 행동이 다른 유스케이스 행동으로 이동한다.

② 〈〈 〉〉는 스테레오 타입을 의미하며 구현과 관련된 사항은 언급하지 않고 요소의 역할에 추가적인 정보를 제공한다.

③ 예금인출은 실제로 예금계좌를 갱신하지 않고 계좌정리에 위임하여 처리한다.

④ 고객과 예금인출은 연관관계이며 행위자와 유스케이스가 메시지를 교환함을 뜻한다.

44. UML 클래스 다이어그램이 다음과 같을 때 Class A의 JAVA 코드로 옳은 것은? (3)

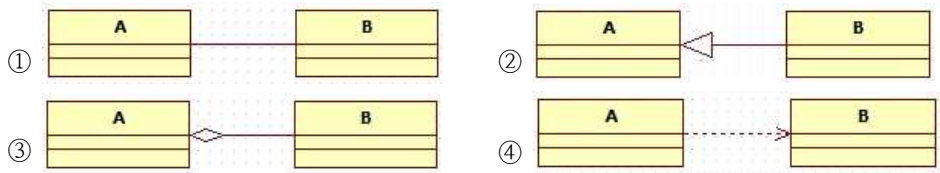


① Public class Class A {
}
② Public class Class A extends Class B{
}
③ Public class Class A {private Class B Link Class B{
}
④ Public class Class A extends Class B{private Link Class B{
}

45. UML의 클래스 다이어그램에서 표현하기 어려운 개념은? (3)

① 전체 부분(whole-part)
② 일반화(generalization)
③ 분할(fork)과 결합(join)
④ 연관(association)

46. '프린터는 출력장치이다'를 UML 클래스 다이어그램으로 표현하고자 한다. A는 출력장치, B는 프린터를 나타내는 클래스라고 할 때 다음 중 가장 적절한 것은? (2)

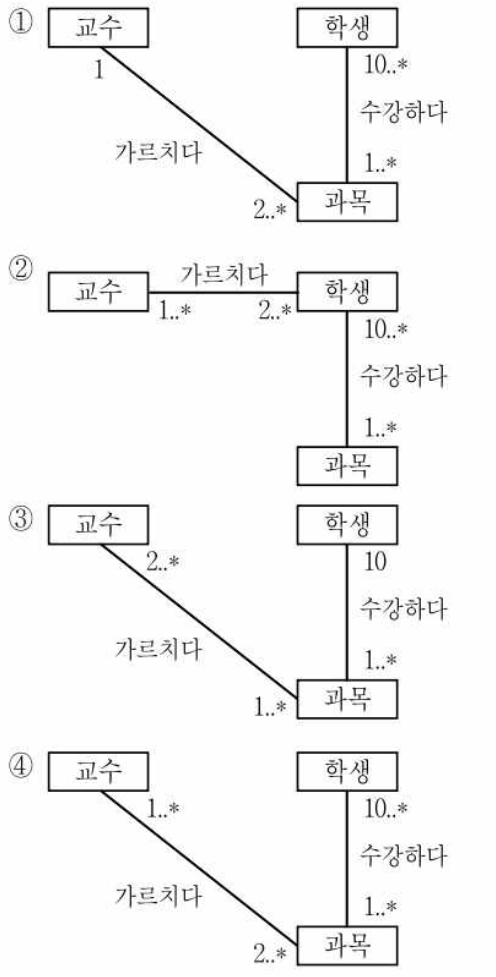


47. UML의 순차(sequence) 다이어그램과 가장 관계가 먼 것은? (4)

① 객체
② 생명선(lifeline)
③ 실행(activation)
④ 구획면(Swimlane)

48. 다음 사항을 UML 클래스 다이어그램으로 가장 잘 나타낸 것은? (4)

○ '교수'는 적어도 두 '과목' 이상을 가르쳐야 한다. ○ '과목'은 한 명 이상의 '교수'가 가르쳐야 한다. ○ '과목'은 열 명 이상의 '학생'들이 수강해야 한다. ○ '학생'은 한 '과목' 이상을 수강해야 한다.



49. 객체 사이의 상호작용을 나타내는 UML 다이어그램은? (3)

① 유스케이스 다이어그램
② 클래스 다이어그램
③ 순차(sequence) 다이어그램
④ 상태 다이어그램
⑤ 액티비티 다이어그램

50. 객체지향 기법에서 분석 객체 모델과 분석 유스케이스 실현 모델은 각각 클래스 다이어그램과 순차(sequence) 다이어그램으로 표현되지만 동일한 하나의 시스템을 모델링하는 것이므로 서로 만족해야 할 조건들이 있다. 다음 중 가장 거리가 먼 것은? (4)

① 객체 모델의 클래스와 유스케이스 실현 모델의 객체 사이의 일관성
② 객체 모델의 연산과 유스케이스 실현 모델의 메시지 사이의 일관성
③ 객체 모델의 연관과 유스케이스 실현 모델의 메시지 사이의 일관성
④ 객체 모델의 클래스와 유스케이스 실현 모델의 메시지 사이의 일관성

51. 상태(state) 다이어그램 작성 방법을 순서대로 가장 적절하게 나열한 것은? (2)

가) 객체가 어떤 상태들을 갖는지 찾아낸다. 나) 표현하려는 대상의 시작, 종료 상태를 파악한다. 다) 상태 머신이 올바로 작성되었는지 검증한다. 라) 상태 다이어그램으로 작성하고자 하는 범위를 파악한다. 마) 상태 변환과 관련된 이벤트, 액션, 조건을 찾는다.

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

52. 상태전이도(STD)에 대한 설명 중 틀린 것은? (4)

① 상태(state)
② 전이(transition)
③ 입출력
④ 연관(association)

53. 다음은 UML의 한 다이어그램에 대한 설명이다. 무엇에 대한 설명인가? (3)

시스템의 동적인 모습을 나타내는 이 다이어그램은 사건에 따라 순차적으로 발생하는 한 객체의 상태변화를 표현한다.

① 컴포넌트 다이어그램(component diagram)
② 통신 다이어그램(communication diagram)
③ 스테이트 다이어그램(state diagram)
④ 액티비티 다이어그램(activity diagram)

54. 다음의 UML 다이어그램들 중에서 통신 다이어그램(communication diagram)과 인터랙션(interaction) 측면에서 같은 정보를 담고 있는 다이어그램은 무엇인가? (2)

시스템의 동적인 모습을 나타내는 이 다이어그램은 사건에 따라 순차적으로 발생하는 한 객체의 상태변화를 표현한다.

① 클래스 다이어그램(class diagram)
② 순차 다이어그램(sequence diagram)
③ 상태 다이어그램(state diagram)
④ 액티비티 다이어그램(activity diagram)

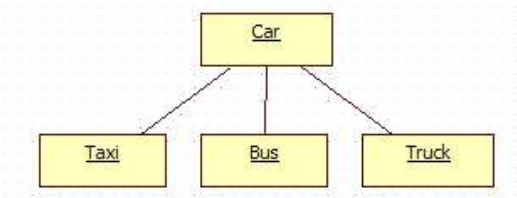
55. UML 다이어그램의 설명이 옳지 않은 것은? (2)

① 사용 사례 다이어그램(use-case diagram) - 시스템의 기능을 모델링
② 상태 다이어그램(state diagram) - 클래스 사이의 메시지 교환을 시간의 흐름에 따라 표현
③ 클래스 다이어그램(class diagram) - 시스템의 정적인 구조를 나타냄
④ 액티비티 다이어그램(activity diagram) - 시스템의 동적 특징을 나타냄

56. UML의 관계 표시법으로 옳은 것은? (4)

① ——————— : 의존(dependency)
② ----------〉 : 실체화(realization)
③ ----------▷ : 연관(association)
④ ———————▷ : 일반화(generalization)

57. UML의 다이어그램에서 관계를 완성하고자 한다. 다음 관계의 표현으로 가장 적합한 것은? (2)



① 의존관계(dependency)
② 일반화관계(generalization)
③ 집단화관계(aggregation)
④ 연관관계(association)

58. 다음 설명을 UML 클래스 다이어그램으로 표현할 때 가장 적합한 용어는? (3)

○ 컴퓨터는 여러 개의 부품으로 구성된다. ○ 컴퓨터를 더 이상 사용할 수 없게 되면 그 부품들도 다른 곳에서 재사용할 수 없다.

① Generalization
② Inheritance
③ Composition
④ Dependency

59. UML 2.2에서 UML 다이어그램은 다이어그램에서 표현하고자 하는 기본 개념의 유형에 따라서 구조 다이어그램과 행위 다이어그램으로 분류될 수 있다. 이러한 관점에서 볼 때 다음의 UML 다이어그램 중에서 다른 3가지 다이어그램과 다른 유형의 다이어그램은 어느 것인가? (1)

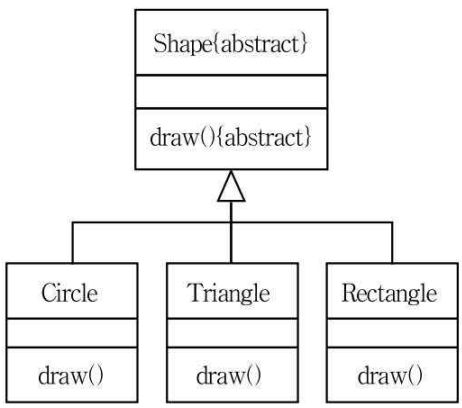
① 컴포넌트 다이어그램(component diagram)　② 시퀀스 다이어그램(sequence diagram)
③ 유스케이스 다이어그램(usecase diagram)　④ 활동 다이어그램(activity diagram)

60. 시스템 또는 시스템을 구성하는 요소들의 동적인 행위를 표현하기 위한 UML 다이어그램이 아닌 것은? (1)

① 배치(deployment) 다이어그램　② 상태(state) 다이어그램
③ 시퀀스(sequence) 다이어그램　④ 타이밍(timing) 다이어그램

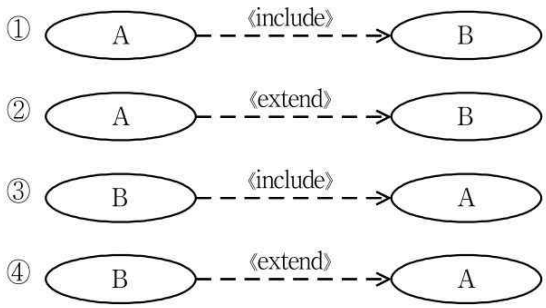
61. 컴포넌트에 대한 설명으로 옳지 않은 것은? (1)
① 컴포넌트는 인스턴스를 정의하는데 사용되는 템플릿으로 타입을 정의한다.
② 컴포넌트는 배치 가능한 개체로서 실행 플랫폼에 직접 설치된다.
③ 컴포넌트를 정의하는 요소는 인터페이스이다.
④ 컴포넌트는 언어 독립적이고, 객체지향 프로그래밍 언어가 아닌 언어로도 구현할 수 있다.

62. 다음 클래스 다이어그램에서 적용되지 않은 개념은? (3)

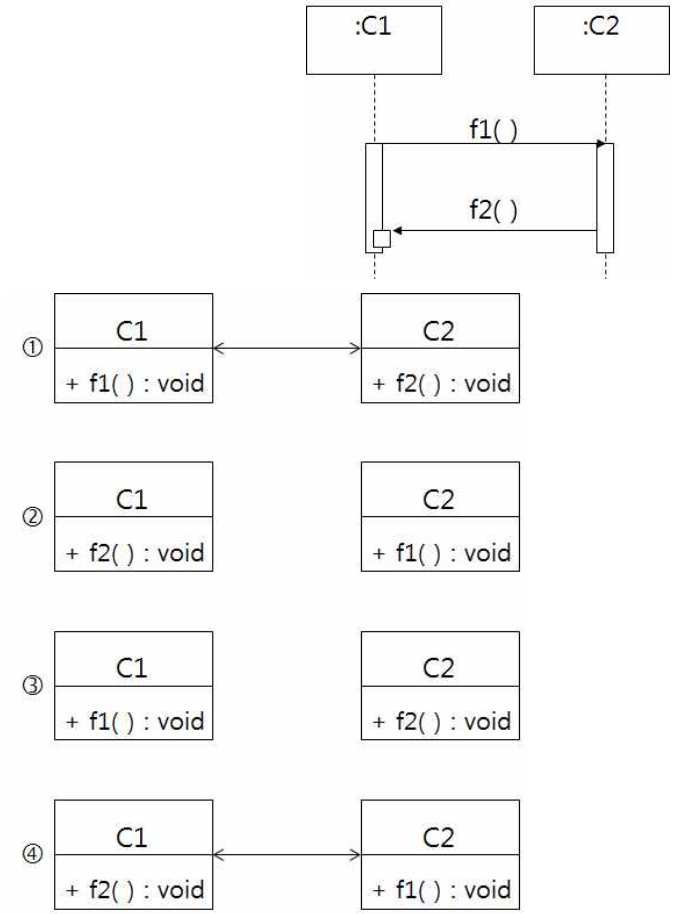


① 일반화(Generalization)　② 상속(Inheritance)
③ 합성(Composition)　④ 다형성(Polymorphism)

63. 유스케이스 다이어그램에서 A 유스케이스를 수행하는 도중에 특정 조건을 만족하면 B 유스케이스를 수행한다. A 유스케이스와 B 유스케이스 간의 관계로 옳은 것은? (4)



64. 다음은 UML 시퀀스 다이어그램을 보여준다. 이 시퀀스 다이어그램과 가장 일치하는 클래스 다이어그램은? (4)



65. 객체지향 분석 단계에서 클래스 모델링을 위해 사용하는 CRC 카드 활용에 대한 설명으로 옳지 않은 것은? (4)

① 클래스의 기능성(responsibility)을 카드에 기재한다.
② CRC 기법의 특징은 클래스의 책임을 가지는 팀 멤버들에게 해당 카드를 배포하는 것이다.
③ CRC 카드를 팀이 활용할 경우, 멤버들 간의 상호작용으로 속성이나 메소드 등이 클래스에서 누락되었는지를 밝혀준다.
④ CRC 카드 활용법은 팀 멤버들이 관련 업무영역에 충분한 경험을 갖고 있지 않더라도 적절한 것이 장점이다.

66. 다음 중 밑줄 친 곳에 들어갈 것은? (4)

CRC 모델은 클래스들을 나타내는 표준 인덱스 카드들의 모임이다. 카드의 위 부분에는 클래스의 이름이 기록된다. 카드의 좌측에는 ㉠ 이(가), 우측에는 ㉡ 들이 기재된다.

① ㉠ 수신자, ㉡ 생성자
② ㉠ 수신자, ㉡ 협력자
③ ㉠ 책임, ㉡ 생산자
④ ㉠ 책임, ㉡ 협력자

67. 클래스 간의 관계에 대한 설명으로 옳지 않은 것은? (4)

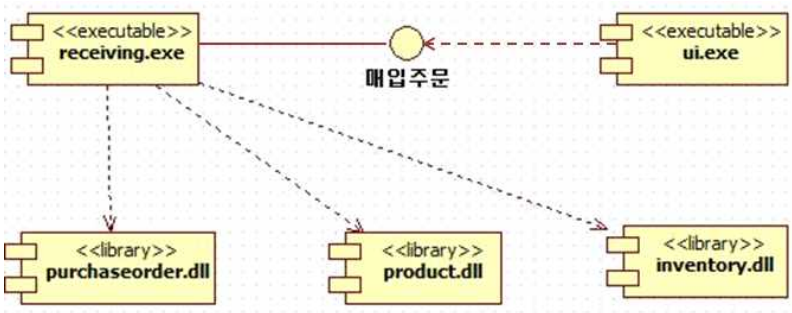
① 연관 관계의 다중성(multiplicity)은 두 클래스의 연관 관계에서 실제로 연관을 가지는 객체의 수를 나타낸다.
② 집합(Aggregation) 관계는 전체와 그 객체의 구성 요소 사이의 관계를 나타내며, whole-part 관계를 나타낸다.
③ 일반화(Generalization) 관계는 일반적인 클래스와 구체적인 클래스 간의 관계 즉 상속 개념을 나타내며, is a kind of의 관계를 나타낸다.
④ 의존 관계는 하나의 클래스에 있는 멤버 함수의 인자가 변해도 다른 클래스에 영향을 미치지 않는 관계를 의미한다.

68. 다음과 같은 Java 코드를 클래스 다이어그램으로 표현하고자 한다. 생성된 클래스 다이어그램에 표현되는 객체 사이의 관계와 가장 거리가 먼 것은? (3)

```
public TempServiceImpl extends TempService {
    private TempDao tempDao;
    public void addTemp(Temp tmp) {
        tempDao.add(tmp);
    }
}
```

① 연관(Association) 관계
② 상속(Inheritance) 관계
③ 집합(Composition) 관계
④ 의존(Dependency) 관계

69. 접수(receiving), 물품(product), 매입주문(purchaseorder), 재고목록(inventory), 사용자인터페이스(ui) 컴포넌트를 사용하는 클래스를 구현한 컴포넌트 다이어그램에 대한 설명으로 가장 적절하지 않은 것은? (4)



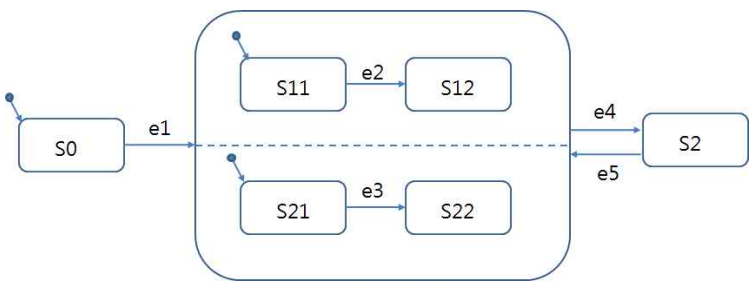
① 접수 컴포넌트(receiving,exe)는 물품 컴포넌트(product.dll)를 참조하여 물품의 상태를 수정할 수 있다.

② 접수 컴포넌트(receiving,exe)와 매입주문 컴포넌트(purchaseorder.dll), 물픔 컴포넌트(product.dll), 재고목록 컴포넌트(inventory.dll) 사이에는 의존관계가 존재하므로 점선 하살표로 표시하였다.

③ 사용자 인터페이스 애플리케이션 컴포넌트(ui.exe)와 매입주문 인터페이스 사이의 화살표는 UI가 접수 컴포넌트(receiving,exe)에 접근할 경우, 반드시 매입주문 인터페이스를 거쳐야 한다는 것을 나타낸다.

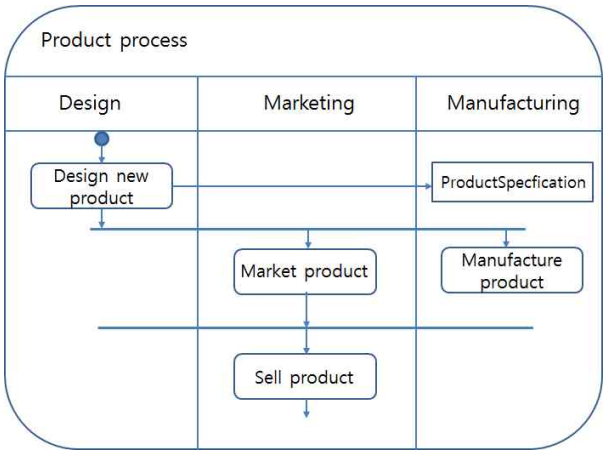
④ 매입주문 컴포넌트(purchaseorder.dll)가 접수 컴포넌트(receiving,exe)를 참조하기 때문에 화살표의 방향은 접수 컴포넌트(receiving,exe)에서 매입주문 컴포넌트(purchaseorder.dll) 쪽을 향하고 있다.

70. 다음은 UML 2.0의 상태 다이어그램을 나타내고 있다. 만약 e1, e2, e3, e4, e5 순으로 수행되었을 때 최종 상태로 가장 적절한 것은? (2)



① S12, S22 ② S11, S21
③ S11, S22 ④ S12, S21

71. 다음 그림은 UML 다이어그램의 일부이다. 그림에 대한 설명으로 가장 거리가 먼 것은? (4)



① 여러 부서의 역할을 swim lane으로 표시하고 있다.
② fork와 join 노드가 사용되고 있다.
③ 객체 흐름이 잘 나타나 있다.
④ 객체가 가질 수 있는 상태의 변화를 잘 나타내고 있다.