

★ 주요 키워드 ★

- (1) 요구사항 분석
- (2) 구조적 분석 도구 > DFD (자료흐름도)
- (3) 구조적 분석 도구 > DD (자료사전)
- (4) 구조적 분석 도구 > HIPO
- (5) 데이터 모델링 도구 > ERD
- (6) 구조적 설계
- (7) N-S 차트 (Nassi-Schneiderman Chart)
- (8) 모듈화 > 목적, 좋은 모듈
- (9) 모듈화 > 결합도
- (10) 모듈화 > 응집도
- (11) 구현
- (12) 화이트박스 검사, McCabe의 소프트웨어 복잡도
- (13) 블랙박스 검사
- (14) 검사 전략
- (15) 디버깅
- (16) 유지보수
- (17) 2008년 기출문제(중복제거)
- (18) 2009년 기출문제(중복제거)

(1) 요구사항 분석

[기-08년5월][기-03년8월][기-00년3월]

1. 구조적 분석 도구와 거리가 먼 것은?

- 가. 자료 사전 나. 자료 흐름도
다. 프로그램 명세서 라. 소단위 명세서

[기-05년3월][기-06년9월]

2. 시스템 개발을 위한 첫 단계는 사용자의 요구나 시스템에 대한 분석이라고 할 수 있다. 이 중 사용자의 요구 분석을 위해 주로 사용하는 기법이 아닌 것은?

- 가. 사용자 면접
나. 현재 사용 중인 각종 문서 검토
다. 설문 조사를 통한 의견 수렴
라. 통제 및 보안 분석

[기-05년3월]

3. 분석가(analyst)가 갖추어야 할 능력 중 가장 중요한 것은?

- 가. 추상적인 개념을 파악하여 논리적인 구성요소로 해할 수 있는 능력
나. 서로 상반되고 모호한 정보로부터 필요한 사항을 수 할 수 있는 능력
다. 관련된 하드웨어와 소프트웨어에 관한 최신 기술
라. 거시적 관점에서 세부적인 요소를 관찰할 수 있는 능력

(2) 구조적 분석 도구 > DFD (자료흐름도)

[기-06년3월]

4. 소단위 명세서(Mini-Specification)에 관한 내용 중 옳지 않은 것은?

- 가. 반 페이지나 한 페이지 정도의 크기로 세분화된 모듈을 작성할 때 사용한다.
나. DFD에서는 한 개의 처리 공정이 그 대상이 되지만,

한 공정의 기능이 두 가지 이상이거나 더 세분화함으로써 소단위 명세서를 이해하기가 쉬어진다면 더욱 세분화 될 수도 있다.

다. 소단위 명세서를 작성하는 도구에는 서술 문장, 의사결정나무, 의사 결정표, 표, 그래프 등이 있다.

라. 소단위 명세서는 구조적 언어를 사용하지 않고, 자연어를 사용하여 이해하기 쉽고 엄밀하게 기술한다.

[기-99년8월]

5. DFD에서 사용되는 기호 중 직사각형의 의미는?

- 가. 정보의 생산자와 소비자
나. 구축되어야 하는 소프트웨어 타입(TYPE)
다. 데이터 사전
라. 기능의 흐름

[기-99년10월][기-02년9월][기-07년5월]

6. 자료흐름도(DFD)의 구성요소가 아닌 것은?

- 가. 처리 나. 자료흐름 다. 단말 라. 기수

[기-00년3월][기-01년9월][기-04년3월]

7. 자료흐름도(DFD:Data Flow Diagram)의 구성요소 중 자료출처와 도착지를 나타내는 기호는?



[기-00년10월]

8. 기본 DFD의 특성이라고 볼 수 없는 것은?

- 가. 시스템내의 모든 자료 흐름은 4가지의 기본 기호로 표시된다.
나. 각 각의 변환(처리)에 대하여 개별적인 상세화가 가능하다.
다. 변환(처리)과정이 버블로 표현된다.
라. 배경도는 단 하나의 원으로 구성되어 level 1을 의미한다

[기-01년3월][기-02년3월][기-05년9월][기-04년9월]

9. 자료흐름도에서 구성요소에 대한 기호의 표현 연결이 옳지 않은 것은?

- 가. 자료흐름 : 화살표로 표시
나. 처리과정 : 마름모로 표시
다. 자료저장장소 : 직선(단선, 이중선)으로 표시
라. 종착지 : 사각형으로 표시

[기-01년6월][기-03년5월]

10. 자료흐름도의 구성 요소와 표시 기호의 연결이 옳지 않은 것은?

- 가. 종착지(terminator) : 오각형
나. 자료흐름(data flow) : 화살표
다. 처리과정(process) : 원
라. 자료저장소(data store) : 직선

[기-08년9월][기-03년8월]

11. 자료흐름도의 구성요소가 아닌 것은?

- 가. 소단위명세서 나. 단말
다. 프로세스 라. 자료저장소

[기-04년3월]

12. 자료흐름도(DFD)의 작성 지침이라고 볼 수 없는 것은?

- 가. 자료는 처리를 거쳐 변환될 때마다 새로운 명칭을

- 부여해야 한다.
나. 자료흐름도의 최하위 처리(process)는 소단위명세서를 갖는다.
다. 배경도(context diagram)에도 명칭과 번호를 부여해야 한다.
라. 어떤 처리(process)가 출력자료를 산출하기 위해서는 필요한 자료가 반드시 입력되어야 한다.

[기-04년5월]

13. 자료흐름도(DFD)를 작성하는데 지침이 될 수 없는 항목은?

- 가. 자료흐름은 처리(Process)를 거쳐 변환 될 때마다 새로운 이름을 부여한다.
나. 어떤 처리(Process)가 출력자료를 산출하기 위해서는 반드시 입력자료가 발생해야 한다.
다. 자료저장소에 입력 화살표가 있으면 반드시 출력 화살표도 표시되어야 한다.
라. 상위단계의 처리(Process)와 하위 자료흐름도의 자료 흐름은 서로 일치되어야 한다.

[기-05년5월]

14. DFD의 설명으로 옳지 않은 것은?

- 가. Bubble Chart라고도 부른다.
나. 구성 요소 중 종착지는 원으로 표시한다.
다. DFD 작성시 정확한 이름을 사용하고 자료 보존 법칙을 준수한다.
라. 처리 공정과 이들 간의 자료흐름을 그래프 형태로 도형화하여 표현한 것이다.

[기-06년3월]

15. DFD(data flow diagram)에 대한 설명으로 거리가 먼 것은?

- 가. 자료 흐름 그래프 또는 버블(bubble)차트라고 한다.
나. 구조적 분석기법에 이용된다.
다. 시간 흐름의 개념을 명확하게 표현할 수 있다.
라. DFD요소에는 화살표, 원, 사각형, 직선(단선/이중선)으로 표시된다.

[기-06년5월]

16. 자료흐름도(DFD)에서 처리(process)를 나타내는 기호는?

- 가. 원 나. 사각형 다. 화살표 라. 삼각형

[기-06년9월][기-05년3월][기-07년9월]

17. 데이터 흐름도(DFD)의 구성요소에 포함되지 않는 것은?

- 가. 처리공정(process)
나. 자료흐름(data flow)
다. 자료사전(data dictionary)
라. 자료저장소(data store)

[기-07년3월]

18. 구조적 분석도구인 자료흐름도의 구성 요소가 아닌 것은?

- 가. process 나. data store 다. definition 라. terminator

(3) 구조적 분석 도구 > DD (자료사전)

[기-04년3월]

19. 자료사전(DD)에서 하나 이상의 선택이 필요할 때 사용하는 기호는?

- 가. () 나. { } 다. [] 라. < >

[기-09년3월][기-08년9월][기-08년3월][기-06년9월][기-07년9월]

20. 자료 사전(Data Dictionary)에서 반복을 의미하는 것은?

- 가. = 나. { } 다. + 라. ()

[기-07년5월]

21. 자료 사전에서 기호 “ { } ”의 의미는?

- 가. “optional” 나. “is composed of”
다. “iteration of” 라. “comment”

[기-09년8월][기-09년5월][기-99년4월]

22. 자료사전(Data Dictionary)에 사용되는 기호의 의미를 올바르게 나타낸 것으로 짝지어진 것은?

- 가. { } : 자료의 생략 가능, () : 자료의 선택
나. () : 자료의 설명, ** : 자료의 선택
다. = : 자료의 설명, ** : 자료의 정의
라. + : 자료의 연결, () : 자료의 생략 가능

[기-01년6월]

23. 다음 내용을 자료사전(data dictionary)의 형태로 옳게 표기한 것은?

고객명세는 고객성명, 고객번호, 고객주소로 구성되어 있으며, 고객성명과 고객번호는 둘 중 하나만 선택이 가능함

- 가. 고객명세 : <고객성명 | 고객번호> + 고객주소
나. 고객명세 = {고객명세 | 고객번호} + 고객주소
다. 고객명세 = [고객성명 | 고객번호] + 고객주소
라. 고객명세 : (고객성명 | 고객번호) + 고객주소

(4) 구조적 분석 도구 > HIPO

[기-00년7월][기-02년3월][기-04년9월][기-06년9월]

24. HIPO(hierarchy plus input process output)에 대한 설명으로 옳지 않은 것은?

- 가. HIPO 다이어그램에는 가시적 도표(visual table of contents), 총체적 다이어그램(overview diagram), 세부적 다이어그램(detail diagram)의 세 종류가 있다.
나. 가시적 도표(visual table of contents)는 시스템에 있는 어떤 특별한 기능을 담당하는 부분의 입력, 처리, 출력에 대한 전반적인 정보를 제공한다.
다. HIPO 다이어그램은 분석 및 설계 도구로서 사용된다.
라. HIPO는 시스템의 설계나 시스템 문서화용으로 사용되고 있는 기법이며, 기본 시스템 모델은 입력, 처리, 출력으로 구성된다.

[기-01년3월][기-04년3월]

25. 프로그램을 구성하는 기능을 기술한 것으로 입력, 처리, 출력을 기술하는 HIPO 패키지에 해당하는 것은?

- 가. Overview Diagram 나. Detail Diagram
다. Visual Table of contents 라. Index Diagram

[기-02년9월][기-05년9월]

26. HIPO(Hierarchy Input Process Output)에 대한 설명으로 옳지 않은 것은?

- 가. 상황식 소프트웨어 개발을 위한 문서화 도구이다.
나. 구조도, 개요 도표 집합, 상세 도표 집합으로 구성된다.
다. 기능과 자료의 의존 관계를 동시에 표현할 수 있다.
라. 보기 쉽고 이해하기 쉽다.

(5) 데이터 모델링 도구 > ERD

[기-99년10월][기-02년5월]

27. 데이터 모델링에 있어서 ERD(Entity Relationship Diagram)는 무엇을 나타내고자 하는가?

- 가. 데이터 흐름의 표현
- 나. 데이터 구조의 표현
- 다. 데이터 구조들과 그들간의 관계들을 표현
- 라. 데이터 사전을 표현

[기-00년10월]

28. 자료 모형화를 위해 ERD를 작성할 때 올바른 순서는?

- ㉠ 기본적인 엔티티와 주요키를 정의하며, 엔티티 사이의 관계를 정의한다
- ㉡ 주요키를 포함하는 엔티티의 속성을 모두 찾아낸다.
- ㉢ 1:M 관계를 단순화시키기 위해 속성 엔티티를 추가하며, 연관관계를 정의하여 M:N 관계를 표현한다.
- ㉣ 각 엔티티를 정규화, 누락된, 엔티티 점검 및 클래스 구조가 필요한지 결정한다.

- 가. ㉠-㉢-㉡-㉣
- 나. ㉠-㉡-㉣-㉢
- 다. ㉡-㉠-㉢-㉣
- 라. ㉠-㉡-㉢-㉣

(6) 구조적 설계

[기-99년10월][기-02년3월]

29. 자료 흐름 중심의 설계 절차들을 올바른 순서로 나열한 것은?

- 1. 자료흐름도를 프로그램 구조로 사상한다.
- 2. 흐름의 경계를 표시한다.
- 3. 정보흐름의 유형을 설정한다.
- 4. 제어계층을 분해시켜서 정의한다.
- 5. 경험적 방법으로 구체화시킨다.

- 가. 1-2-3-4-5
- 나. 3-2-1-4-5
- 다. 4-5-3-2-1
- 라. 4-5-1-2-3

[기-09년8월][기-99년8월][기-02년5월][기-07년5월]

30. 사용자 인터페이스 설계시 오류 메시지나 경고에 관한 다음의 지침 중 옳지 않은 것은 어느 것인가?

- 가. 메시지는 이해하기 쉬어야 한다.
- 나. 오류로부터 회복을 위한 구체적인 설명이 제공되어야 한다.
- 다. 오류로 인해 발생할 수 있는 부정적인 내용은 가급적 피한다.
- 라. 소리나 색 등을 이용하여 듣거나 보기 쉽게 의미 전달을 하도록 한다.

[기-99년10월]

31. 소프트웨어 설계에서 사용되는 대표적인 3가지 추상화기법에 대한 설명 중 틀린 것은?

- 가. 제어추상화 - 제어의 정확한 매카니즘을 정의하지 않고 원하는 효과를 정하는데 이용
- 나. 기능추상화 - 입력자료를 출력자료로 변환하는 과정을 추상화하는 방법

다. 정보는닉추상화 - 설계된 각 모듈은 자세한 처리내용이 시스템의 다른 부분으로부터 감추어져 있는 방법
라. 자료추상화 - 자료와 자료에 적용될 수 있는 기능을 함께 정의함으로써 자료 객체를 구성하는 방법

[기-09년3월][기-00년3월][기-04년5월]

32. 설계품질을 평가하기 위해서는 반드시 좋은 설계에 대한 기준을 세워야 한다. 다음 중 좋은 기준이라고 할 수 없는 것은?

- 가. 설계는 모듈적이어야 한다.
- 나. 설계는 자료와 프로시저에 대한 분명하고 분리된 표현을 포함해야 한다.
- 다. 소프트웨어 요소들간의 효과적 제어를 위해 설계에서 계층적 조직이 제시되어야 한다.
- 라. 설계는 서브루틴이나 프로시저가 전체적이고 통합적일 수 있도록 유도되어야 한다.

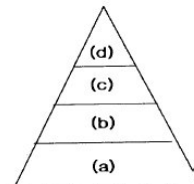
[기-00년7월][기-02년9월][기-03년3월]

33. 소프트웨어 설계를 위한 지침에 대한 설명으로 거리가 먼 것은?

- 가. 소프트웨어 요소간의 효과적 제어를 위해 설계에서 계층적 자료조건이 제시되어야 한다.
- 나. 설계는 종속적인 기능적 특성을 가진 모듈화로 유도되어야 한다.
- 다. 소프트웨어는 논리적으로 특별한 기능과 부기능을 수행하는 요소들로 나누어져야 한다.
- 라. 설계는 자료와 프로시저에 대한 분명하고 분리된 표현을 포함해야 한다.

[기-03년5월]

34. 다음은 소프트웨어 설계 모형의 구조도이다. (a)(b)(c)(d)에 들어갈 항목을 순서대로 나열한 것은?



- 가. 데이터설계 - 아키텍처설계 - 절차설계 - 인터페이스설계
- 나. 아키텍처설계 - 데이터설계 - 절차설계 - 인터페이스설계
- 다. 아키텍처설계 - 데이터설계 - 인터페이스설계 - 절차설계
- 라. 데이터설계 - 아키텍처설계 - 인터페이스설계 - 절차설계

[기-00년3월]

35. 프로그램 구조에서 fan-in은 무엇을 의미하는가?

- 가. 얼마나 많은 모듈이 주어진 모듈을 호출하는가를 나타냄
- 나. 주어진 모듈이 호출하는 모듈의 개수를 나타냄
- 다. 같은 등극(level)의 모듈 수를 나타냄
- 라. 최상위 모듈에서 주어진 모듈까지의 깊이를 나타냄

[기-03년3월]

36. 다음은 프로그램 구조를 나타낸다. 모듈 F에서의 fan-in과 fan-out의 수는 얼마인가?



- 가. fan-in: 2 fan-out:3 나. fan-in: 3 fan-out:2
다. fan-in: 1 fan-out:2 라. fan-in: 2 fan-out:1

[기-00년10월][기-04년3월]

37. 소프트웨어 구조와 관련된 용어로, 주어진 한 모듈을 제어하는 상위 모듈 수를 나타내는 것은?

- 가. Modularity 나. Subordinate
다. Fan-in 라. Super-Ordinate

[기-02년3월][기-05년3월][기-06년5월]

38. 구조적 프로그래밍에서 사용하는 기본적인 제어구조에 해당하지 않는 것은?

- 가. 순차(sequence) 나. 반복(iteration)
다. 호출(call) 라. 선택(selection)

(7) N-S 차트 (Nassi-Schneiderman Chart)

[기-99년8월][기-07년3월][기-05년3월][기-01년6월]

39. 나씨-슈나이더만(Nassi-Schneiderman) 도표는 구조적 프로그램을 표현하기 위해 고안되었다. 이 방법에서 알고리즘의 제어구조는 3가지로 충분히 표현될 수 있는데 이에 속하지 않는 것은 어느 것인가?

- 가. 선택, 다중선택(if ~ then ~ else, case)
나. 반복 (repeat ~ until, while, for)
다. 분기(goto, return)
라. 연속(sequential)

[기-00년7월]

40. 모듈(Module) 설계시 문서화의 기법 중 N-S차트(Nassi-Schneiderman)에 관한 설명으로 가장 적절한 것은?

- 가. 도표나 그림이 아닌 if then else, while do end 등과 같은 키워드를 사용하여 작성하는 방법이다.
나. 모듈의 내부 기능과 논리를 문법적 제약없이 명시한 명세서이다.
다. 논리의 기술에 중점을 둔 도형식 표현 방법이다.
라. 모듈의 처리과정을 수직적인 것과 수평적인 것의 두차원으로 표현하는 방법이다.

[기-03년8월][기-05년5월]

41. NS(Nassi-Schneiderman) chart에 대한 설명으로 거리가 먼 것은?

- 가. 논리의 기술에 중점을 둔 도형식 표현 방법이다.
나. 연속, 선택 및 다중 선택, 반복 등의 제어논리 구조로 표현한다.
다. 주로 화살표를 사용하여 논리적인 제어구조로 흐름을 표현 한다.
라. 조건이 복합되어 있는 곳의 처리를 시각적으로 명확히 식별하는데 적합하다.

(8) 모듈화 > 목적, 좋은 모듈

[기-09년8월][기-08년9월][기-08년3월][기-04년3월][기-03년3월][기-07년5월][기-07년3월][기-01년3월]

42. 좋은 모듈이 되기 위한 응집도와 결합도에 대한 설명으로 옳은 것은?

- 가. 모듈의 응집도와 결합도 모두가 높아야 한다.
나. 모듈의 응집도는 높아야 하고 결합도는 낮아야 한다.
다. 모듈의 응집도는 낮아야 하고 결합도는 높아야 한다.
라. 모듈의 응집도와 결합도 모두가 낮아야 한다.

[기-05년9월][기-99년8월]

43. 효과적인 모듈화 설계 방안이 아닌 것은?

- 가. 응집도를 높인다.
나. 결합도를 낮춘다.
다. 복잡도와 중복을 피한다.
라. 예측 불가능하도록 정의한다.

[기-05년9월]

44. 모듈화 설계의 장점에 해당하지 않는 것은?

- 가. 확장성 나. 융통성 다. 복잡성 라. 경제성

[기-06년3월]

45. 하나의 프로그램을 몇 개의 작은 부분으로 분할하는 경우, 그 분할 단위를 일반적으로 모듈(Module)이라고 한다. 다음 중 모듈에 대한 설명으로 옳은 것은?

- 가. 모듈의 독립성을 높여주기 위해서는 각 모듈간의 관련성을 최소로 하며, 이 경우에 응집도(Cohesion)는 최소가 된다.
나. 모듈의 관련성을 최대로 하면 모듈의 독립성은 저하되며, 이 경우에 모듈의 결합도(Coupling)는 최소가 된다.
다. 복잡성을 감소시키는 수단으로 독립성의 개념이 많이 적용되고 있으며, 모듈의 독립성 척도로서 결합도는 고려 대상이 아니며, 응집도만 적용된다.
라. 모듈의 결합도는 자료 결합도(Data Coupling)로, 모듈의 응집도는 기능적 응집도(Functional Cohesion)로 하는 것이 가장 바람직하다.

(9) 모듈화 > 결합도

[기-01년6월]

46. 한 모듈이 다른 모듈의 내부 기능 및 그 내부 자료를 조회하는 경우의 결합도에 해당하는 것은?

- 가. data coupling 나. stamp coupling
다. common coupling 라. content coupling

[기-01년9월]

47. 모듈의 결합도를 높은 순서대로 옳게 표시한 것은?

㉠ 외부결합	㉡ 제어결합	㉢ 스템프결합
㉣ 자료결합	㉤ 내용결합	

- 가. ㉠>㉡>㉢>㉤ 나. ㉤>㉠>㉡>㉢
다. ㉡>㉢>㉤>㉠ 라. ㉢>㉤>㉠>㉡

[기-02년3월][기-03년5월][기-04년9월][기-00년7월]

48. 결합도(coupling)가 강한 순서대로 옳게 나열된 것은?

- 가. 내용 결합도>공통 결합도>제어 결합도>스탬프 결합도>데이터 결합도
나. 공통 결합도>내용 결합도>제어 결합도>데이터 결합도>스탬프 결합도
다. 데이터 결합도>내용 결합도>제어 결합도>공통 결합도>스탬프 결합도

라. 공통 결합도>내용 결합도>제어 결합도>스탬프 결합도>데이터 결합도

[기-02년9월]

49. 단일 파일 또는 동종 테이블을 매개변수로 통신하는 모듈 간의 결합성을 의미하는 것은?

- 가. 자료결합도(data coupling)
- 나. 제어결합도(control coupling)
- 다. 공통결합도(common coupling)
- 라. 내용결합도(content coupling)

[기-03년3월][기-00년3월]

50. 한 모듈이 다른 모듈의 내부 기능 및 그 내부 자료를 참조하는 경우, 이를 무슨 결합이라고 하는가?

- 가. 내용 결합 나. 제어 결합
- 다. 공통 결합 라. 스탬프 결합

[기-03년8월]

51. 모듈이 파라미터나 인수로 다른 모듈에게 데이터를 넘겨주고 호출받은 모듈은 받은 데이터에 대한 처리 결과를 다시 돌려주는 유형의 모듈 결합도(coupling)를 무엇이라고 하는가?

- 가. 내용 결합도 나. 외부 결합도
- 다. 제어 결합도 라. 데이터 결합도

[기-05년3월][기-99년10월]

52. 모듈 결합도(Coupling)에 관한 설명으로 옳지 않은 것은?

- 가. 자료결합(Data Coupling) - 모듈간의 인터페이스가 자료요소로만 구성된 경우
- 나. 스탬프결합(Stamp Coupling) - 모듈간의 인터페이스로 배열이나 레코드 등의 자료구조가 전달된 경우
- 다. 내용결합(Content Coupling) - 한 모듈이 다른 모듈의 일부분을 참조 또는 수정하는 경우
- 라. 제어결합(Control Coupling) - 한 모듈이 다른 모듈에게 제어요소를 전달하고 여러 모듈이 공통 자료영역을 사용하는 경우

[기-05년5월]

53. 결합도 단계 순서(약 -> 강)를 바르게 표시한 것은?

- 가. stamp coupling -> data coupling -> control coupling -> common coupling -> content coupling
- 나. data coupling -> stamp coupling -> control coupling -> common coupling -> content coupling
- 다. content coupling -> stamp coupling -> control coupling -> common coupling -> data coupling
- 라. control coupling -> data coupling -> stamp coupling -> common coupling -> content coupling

[기-06년9월][기-03년8월][기-00년10월]

54. 한 모듈과 다른 모듈간의 상호 의존도 또는 두 모듈 사이의 연관 관계를 의미하는 것은?

- 가. 신뢰도 나. 충실도 다. 응집도 라. 결합도

[기-99년4월]

55. 다음 중 모듈(module) 설계에서 가장 좋은 결합도 상태는 어느 것인가?

- 가. control coupling 나. stamp coupling
- 다. common coupling 라. coutent coupling

[기-01년3월][기-05년3월]

56. 두 모듈이 동일한 자료구조를 조회하는 경우의 결합성이며 자료구조의 어떠한 변화, 즉 포맷이나 구조의 변화는 그것을 조회하는 모든 모듈 및 변화되는 필드를 실제로 조회하지 않는 모듈에까지도 영향을 미치게 되는 결합성은?

- 가. data coupling 나. stamp coupling
- 다. control coupling 라. content coupling

(10) 모듈화 > 응집도

[기-09년3월][기-01년6월][기-04년3월][기-02년3월][기-99년8월]

57. 응집력이 강한 것부터 약한 순서로 옳게 나열된 것은?

- 가. sequential → functional → procedural → coincidental → logical
- 나. procedural → coincidental → functional → sequential → logical
- 다. functional → sequential → proedural → logical → coincidental
- 라. logical → coincidental → functional → sequential → procedural

[기-02년5월]

58. 모듈의 응집력(Cohesion)에 대한 설명 중 틀린 것은?

- 가. 모듈의 응집도란 모듈 안의 요소들이 서로 관련되어 있는 정도를 말한다.
- 나. 기능적 응집도(Functional Cohesion)는 한 모듈 내부의 한 기능요소에 의한 출력자료가 다음 기능 원소의 입력자료로서 제공되는 형태이다.
- 다. 교환적 응집도(Communication Cohesion)는 동일한 입력과 출력을 사용하는 소 작업들이 모인 모듈에서 볼 수 있다.
- 라. 논리적 응집도(Logical Cohesion)는 유사한 성격을 갖거나 특정형태로 분류되는 처리요소들로 하나의 모듈이 형성되는 경우이다.

[기-03년3월][기-06년5월]

59. 모듈을 이루고 있는 각 요소들이 공통의 목적을 달성하기 위하여 얼마나 관련이 있는가를 나타내는 것을 무엇이라고 하는가?

- 가. 결합도(coupling) 나. 응집도(cohesion)
- 다. 구조도(structure) 라. 일치도(unity)

[기-03년5월]

60. 데이터 설계에 있어서 응집력(cohesion)의 의미로 가장 적절한 것은?

- 가. 데이터 구조들이 시스템 전반에 얼마나 연관관계를 가지고 있는가 하는 정도
- 나. 모듈이 개발 단계별로 얼마나 잘 정의되어 있는가 하는 정도
- 다. 모듈이 독립적인 기능으로 잘 정의되어 있는 정도
- 라. 모듈들간의 상호 연관성의 정도

[기-03년5월]

61. 모듈의 구성 요소가 하나의 활동으로부터 나온 출력

자료를 그 다음 활동의 입력 자료로 사용하는 같은 모듈 내에서의 응집의 정도를 나타내는 것은?

- 가. 절차적(procedural) 응집도
- 나. 논리적(logical) 응집도
- 다. 기능적(functional) 응집도
- 라. 순차적(sequential) 응집도

[기-03년8월]

62. 모듈(module)의 응집도(cohesion)가 약한것 부터 강한 순서로 옳게 나열된 것은?

- 가. 기능적응집 => 시간적응집 => 논리적응집
- 나. 시간적응집 => 기능적응집 => 논리적응집
- 다. 논리적응집 => 시간적응집 => 기능적응집
- 라. 논리적응집 => 기능적응집 => 시간적응집

[기-05년9월]

63. Myers의 응집력 단계 순서(강 --> 약)를 바르게 표시한 것은?

- 가. functional cohesion -> communication cohesion
-> procedural cohesion -> temporal cohesion
-> logical cohesion
- 나. functional cohesion -> procedural cohesion ->
communication cohesion -> temporal cohesion
-> logical cohesion
- 다. procedural cohesion -> functional cohesion ->
communication cohesion -> temporal cohesion
-> logical cohesion
- 라. logical cohesion -> procedural cohesion ->
functional cohesion -> communication cohesion
-> temporal cohesion

[기-99년4월]

64. 다음 중 Myers가 구분한 응집력(cohesion)의 정도에서 가장 낮은 응집력을 갖는 단계는?

- 가. sequential cohesion 나. coincidental cohesion
- 다. temporal cohesion 라. functional cohesion

[기-01년3월]

65. 데이터 설계에 있어서, 응집력(cohesion)의 의미로 가장 적절한 것은?

- 가. 데이터 구조들이 시스템 전반에 얼마나 연관관계를 가지고 있는가 하는 정도
- 나. 모듈이 개발 단계별로 얼마나 잘 정의되어 있는가 하는 정도
- 다. 모듈이 독립적인 기능으로 잘 정의되어 있는 정도
- 라. 데이터들간의 상호 연관성의 정도

(11) 구현

[기-99년10월][기-05년5월]

66. 소프트웨어 개발을 위한 프로그래밍 언어의 선정기준으로 거리가 먼 것은?

- 가. 개발담당자의 경험과 지식 나. 대상업무의 성격
- 다. 과거의 개발실적 라. 4세대 언어 여부

[기-00년7월]

67. 구조화 프로그래밍 방법론 중 선택과 반복 구조를 사용하는 것은?

- 가. Yourdon 방법론 나. Jackson 방법론

다. Dijkstra 방법론

라. Warnier-orr 방법론

[기-01년9월][기-02년3월][기-04년9월][기-07년5월]

68. 소프트웨어 개발 방법론에서 구현(Implementation)에 대한 설명으로 가장 적절한 것은?

- 가. 요구사항 분석 과정 중 모아진 요구사항을 옮기는 것
- 나. 시스템이 무슨 기능을 수행하는지에 대한 시스템의 목표기술
- 다. 프로그래밍 또는 코딩이라고 불리며 설계 명세서가 컴퓨터가 알 수 있는 모습으로 변환되는 과정
- 라. 시스템이나 소프트웨어 요구 사항을 정의하는 과정

[기-05년3월]

69. 시스템의 설계 명세서를 바탕으로 모듈 단위의 코딩과 디버깅 및 단위 테스트가 이루어지는 소프트웨어 개발 단계는?

- 가. 코딩 나. 구현 다. 테스트 라. 프로그램 설계

(12) 화이트박스 검사, McCabe의 소프트웨어 복잡도

[기-03년3월]

70. 화이트박스 테스트 기법에 해당되는 것은?

- 가. Equivalence partitioning
- 나. Boundary value analysis
- 다. Cause and effect graphing
- 라. Condition coverage

[기-06년5월]

71. 소프트웨어 테스트에 사용되는 방식으로 모듈의 논리적 구조를 체계적으로 점검하는 구조 테스트이며, 이 방식의 유형에는 기초 경로 검사, 조건 검사, 데이터 흐름 검사, 루프 검사 등이 있는 것은?

- 가. 화이트 박스 테스트 나. 블랙 박스 테스트
- 다. 블루 박스 테스트 라. 레드 박스 테스트

[기-09년5월][기-99년8월]

72. 화이트 박스 시험(white box testing)의 설명으로 옳지 않은 것은 어느 것인가?

- 가. 프로그램의 제어구조에 따라 선택, 반복 등의 부분들을 수행함으로써 논리적 경로를 제어한다.
- 나. 모듈 안의 작동을 직접 관찰할 수 있다.
- 다. 소프트웨어 산물의 각 기능별로 적절한 정보영역을 정하여 적합한 입력에 대한 출력의 정확성을 점검한다.
- 라. 원시 코드의 모든 문장을 한 번 이상 수행함으로써 수행된다.

[기-00년7월][기-01년9월][기-06년3월]

73. 모듈안의 작동을 자세히 관찰할 수 있으며, 프로그램 원시 코드의 논리적인 구조를 커버(cover)하도록 테스트 케이스를 설계하는 프로그램 테스트 방법은?

- 가. 블랙 박스 테스트 나. 화이트 박스 테스트
- 다. 알파 테스트 라. 베타 테스트

[기-04년9월]

74. 소프트웨어 테스트에서 화이트박스 기법의 설명에 해당하는 것은?

- 가. 프로그램을 눈으로 보면서 확인하는 것
- 나. 프로그램의 구조에 의거하여 테스트하는 것
- 다. 프로그램의 외부사양에 대하여 테스트하는 것

라. 프로그램의 외관상 관계를 파악하는 것

[기-05년3월]

75. 소프트웨어의 시험 중 화이트박스 시험의 과정이 아닌 것은?

- 가. 조건 테스트 나. 모든 실행문 테스트
다. 경계값 분석 라. 분기점 테스트

[기-05년9월]

76. 화이트박스 검사(test)에 대한 설명 중 잘못된 것은?

- 가. 모듈 안의 논리적인 구조를 검사한다.
나. 동치 분할(equivalence partitioning)이라는 기법을 사용한다.
다. 검사대상의 가능한 경로를 어느 정도 통과하는지의 적용 범위성을 측정기준으로 한다.
라. Nassi-Shneiderman 도표를 사용하여 검정기준을 작성할 수 있다.

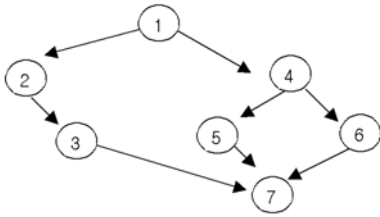
[기-00년7월][기-05년3월]

77. McCabe에 의해 제안된 소프트웨어의 복잡성 측정에 대한 설명으로 옳지 않은 것은?

- 가. 영역은 그래프의 평면에서 둘러 쌓여진 부분으로 묘사될 수 있다.
나. 영역의 수는 경계된 영역들과 그래프 외부의 비 경계지역의 수를 계산한다.
다. 모듈크기의 실제 상한선은 존재하지 않는다.
라. $V(G)$ 는 영역의 수를 결정함으로써 계산되어 진다.

[기-02년5월]

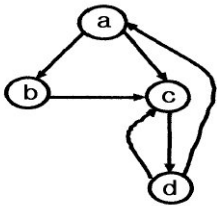
78. McCabe 방법에 의한 다음 그래프의 $V(G)$ 의 크기는?



- 가. 1 나. 3 다. 7 라. 13

[기-09년3월][기-04년5월]

79. 제어흐름 그래프가 다음과 같을 때 McCabe의 cyclomatic 수는 얼마인가?



- 가. 3 나. 4 다. 5 라. 6

[기-07년9월]

80. 화이트박스 검사로 찾기 힘든 오류는?

- 가. 논리흐름도 나. 루프구조 다. 자료구조 라. 순환복잡도

(13) 블랙박스 검사

[기-99년4월][기-04년5월]

81. 블랙박스 테스트를 통해 발견하기 힘든 오류는?

- 가. 성능 오류 나. 부정확한 기능
다. 인터페이스 오류 라. 논리구조상의 오류

[기-01년6월][기-03년5월]

82. 블랙박스 검사에 해당하지 않는 것은?

- 가. 데이터 흐름 검사(data flow testing)
나. 동치 분할 검사(equivalence partitioning testing)
다. 원인 효과 그래픽 기법(cause effect graphic-technique)
라. 비교 검사(comparison testing)

[기-02년5월]

83. 블랙박스 테스트 기법에 해당하는 것은?

- 가. mutation testing(fault based testing)
나. cause-effect graphing testing
다. data flow testing
라. control structure testing

[기-02년9월]

84. 소프트웨어 시험 기법 중 블랙박스 시험기법이 아닌 것은?

- 가. 구조시험(Structure testing)
나. 동등분할(Equivalence partitioning)
다. 경계값 분석(Boundary value analysis)
라. 원인-결과 그래프(Cause-effect graphing)

[기-01년3월][기-05년3월]

85. 제품이 수행할 특정 기능을 알기 위해서 각 기능이 완전히 작동되는 것을 입증하는 검사로서, 기능 검사라고도 하는 것은?

- 가. 블랙박스 검사 나. 그린박스 검사
다. 블루박스 검사 라. 화이트박스 검사

[기-05년5월]

86. 블랙박스 검사에 관하여 기술한 것 중 잘못된 것은?

- 가. 모듈의 구조보다 기능을 검사한다.
나. 동치 분할(equivalence partitioning)이라는 기법을 사용한다.
다. Nassi-Shneiderman 도표를 사용하여 검정기준을 작성할 수 있다.
라. 원인-결과 그래프(cause and effect graph)로 테스트케이스를 작성할 수 있다.

[기-07년3월]

87. 다음 설명에 해당하는 소프트웨어 테스트 기법은?

- 소프트웨어 인터페이스에서 실시되는 검사로 설계된 모든 기능들이 정상적으로 수행되는지 확인한다.
- 소프트웨어의 기능이 의도대로 작동하고 있는지, 입력은 적절하게 받아들여졌는지, 출력은 정확하게 생성되는지를 보여주는 데 사용된다.
- Equivalence Partitioning Testing, Boundary Value Analysis 등이 이 기법에 해당한다.

- 가. 화이트박스 테스트 나. 블랙박스 테스트
다. 레드박스 테스트 라. 블루박스 테스트

[기-07년5월]

88. 다음 검사의 기법 중 종류가 다른 하나는 무엇인가?

- 가. 동치 분할 검사 나. 원인 효과 그래픽 검사

다. 비교 검사 라. 데이터 흐름 검사

[기-09년8월][기-07년9월]

89. 블랙 박스 검사에 대한 설명으로 옳지 않은 것은?

- 가. 각 기능이 완전히 작동되는 것을 입증하는 검사이다.
- 나. 인터페이스 결함, 성능 결함, 초기화와 종료 이상 결함 등을 찾아 낸다.
- 다. 동치 분할 검사는 논리적인 조건과 대응하는 행동을 간략히 표현할 수 있도록 하는 검사 사례 설계 기법이다.
- 라. 경계값 분석은 입력의 경계값에서 발생하는 오류를 제거하기 위한 검사 기법이다.

(14) 검사 전략

[기-99년10월][기-04년3월]

90. 상향식 통합테스트(Bottom-up Integration Test)의 과정이 옳게 나열된 것은?

- | |
|---------------------------|
| ① 드라이버라는 제어프로그램의 작성 |
| ② 낮은 수준의 모듈들을 클러스터로 결합 |
| ③ 클러스터의 검사 |
| ④ 드라이버를 제거하고 클러스터를 상위로 결합 |

- 가. ①→②→③→④ 나. ②→①→③→④
- 다. ②→③→①→④ 라. ①→②→④→③

[기-00년3월][기-02년5월]

91. 소프트웨어 시험의 목적은 오류를 찾아내는데 있다. 이의 종류로는 단위시험, 통합시험, 검증 시험, 그리고 시스템 시험이 있는데 이 중에서 소프트웨어가 요구사항에 맞는지 추적해 보는데 중점을 두고 있는 시험방법은 무엇인가?

- 가. 단위 시험 나. 통합 시험
- 다. 검증 시험 라. 시스템 시험

[기-00년10월][기-02년3월]

92. 소프트웨어 검사 단계를 올바른 순서로 나열한 것은?

- | | |
|---------|-----------|
| ㉠ 설계 검사 | ㉡ 요구사항 검사 |
| ㉢ 코드 검사 | ㉣ 시스템 검사 |

- 가. ㉠-㉡-㉢-㉣ 나. ㉢-㉠-㉡-㉣
- 다. ㉡-㉢-㉣-㉠ 라. ㉡-㉣-㉠-㉢

[기-00년10월][기-03년3월]

93. 검증 시험을 하는데 있어 알파 테스트란?

- 가. 사용자의 장소에서 개발자가 직접 시험을 한다.
- 나. 사용자의 장소에서 개발자와 사용자가 실 데이터를 가지고 공동으로 시험한다.
- 다. 개발자의 장소에서 개발자가 시험을 하고 사용자는 지켜본다.
- 라. 개발자의 장소에서 사용자가 시험을 하고 개발자는 뒤에서 결과를 지켜본다.

[기-02년5월][기-05년5월]

94. 검증시험(Validation Test)을 하는데 있어 Beta Test에 대한 설명으로 옳은 것은?

- 가. 사용 부서에서 개발 담당자가 시험한다.
- 나. 개발부서와 사용 부서가 공동으로 시험한다.
- 다. 개발 부서에서 개발자가 시험을 한다.
- 라. 실 업무를 가지고 사용자가 직접 시험한다.

[기-03년5월][기-05년5월]

95. 하향식 통합에 있어서 모듈간의 통합시험을 하기 위해 일시적으로 필요한 조건만을 가지고 임시로 제공하는 시험용 모듈을 무엇이라고 하는가?

- 가. Driver 나. Stub
- 다. Sub-Program 라. Dummy-Program

[기-04년5월]

96. 소프트웨어 개발단계와 그에 따른 테스트 전략의 결함이 적절한 것은?

- 가. 분석단계 - 결합(Integration) Test
- 나. 설계단계 - 검증(Validation) Test
- 다. 구현단계 - 단위(Unit) Test
- 라. 유지보수단계 - 시스템(System) Test

[기-04년5월]

97. 시스템 테스트 단계의 순서가 적절하게 이루어진 것은?

- 가. 단위 테스트 - 통합 테스트 - 시스템 테스트 - 수용 테스트
- 나. 수용 테스트 - 단위 테스트 - 통합 테스트 - 시스템 테스트
- 다. 단위 테스트 - 통합 테스트 - 수용 테스트 - 시스템 테스트
- 라. 수용 테스트 - 시스템 테스트 - 단위 테스트 - 통합 테스트

[기-05년3월]

98. 소프트웨어 개발 단계와 테스트 전략이 옳게 연결된 항목은?

- 가. 설계 단계 <----> 시스템테스트
- 나. 요구사항 분석 단계 <----> 검증테스트
- 다. 코딩 단계 <----> 통합테스트
- 라. 시스템엔지니어링 단계 <----> 단위테스트

(15) 디버깅

[기-99년10월]

99. 디버깅에 관한 설명 중거리가 먼 것은?

- 가. 디버깅은 성공적인 테스트의 결과로 발생한다.
- 나. 디버깅은 징후로부터 원인을 찾아 수정하는 과정이다.
- 다. 디버깅이 힘든 이유는 심리적인 요소가 많이 관여하기 때문이다.
- 라. 디버깅에 대한 체계적인 접근은 아직까지 제안되고 있지 않다.

(16) 유지보수

[기-07년5월][기-06년5월][기-01년9월][기-03년3월][기-03년8월][기-99년4월]

100. 소프트웨어 개발 단계에서 가장 많은 비용이 소요되는 단계는?

- 가. 계획 단계 나. 분석 단계
- 다. 구현 단계 라. 유지보수 단계

[기-05년5월]

101. 유지보수의 활동 종류로 볼 수 없는 것은?

- 가. 정정(Corrective) 보수 나. 품질(Quality) 보수
- 다. 적응(Adaptive) 보수 라. 예방(Preventive) 보수

[기-05년9월][기-01년6월]

102. 소프트웨어 유지보수의 유형에 해당하지 않는 것은?

- 가. 수정보수(Corrective maintenance)

- 나. 기능보수(Functional maintenance)
다. 완전화보수(Perfective maintenance)
라. 예방보수(Preventive maintenance)

[기-06년3월][기-03년5월][기-00년7월]

103. 소프트웨어 유지보수에 관련된 설명으로 옳지 않은 것은?

- 가. 유지보수는 소프트웨어가 인수, 설치된 후 발생하는 모든 공학적 작업을 말한다.
나. 유지보수는 원인에 따라 수리(corrective)보수, 적응(adaptive)보수, 완전화(perfective)보수, 예방(preventive)보수 등이 있다.
다. 소프트웨어에 가해지는 연결을 제어 관리하는 것을 형상관리(configuration management)라고 한다.
라. 소프트웨어 비용 중 유지보수 비용은 개발비용 보다 적다.

[기-06년3월][기-00년7월]

104. 소프트웨어 유지보수 작업의 목적으로 부적절한 것은?

- 가. 하자보수 나. 환경적응 다. 예방조치 라. 설계수정

[기-07년3월][기-05년3월]

105. 소프트웨어의 문서(document) 표준이 되었을 때, 개발자가 얻는 이득으로 가장 거리가 먼 것은?

- 가. 시스템 개발을 위한 분석과 설계가 용이하다.
나. 프로그램 유지보수가 용이하다.
다. 프로그램의 확장성이 있다.
라. 프로그램 개발 인력이 감소된다.

[기-03년3월][기-01년9월]

106. 외계인 코드(Alien Code)를 방지하기 위한 방법으로 가장 적합한 것은?

- 가. 프로그램 내에 문서화(Documentation)를 철저하게 해두어야 한다.
나. 자료흐름도(DFD)를 상세히 그려야 한다.
다. 프로그램 완성시 testing을 확실하게 해야 한다.
라. 프로그램시 반드시 visual tool을 사용해야 한다

[기-02년3월][기-04년3월][기-99년8월]

107. 소프트웨어 라이프사이클 단계중 가장 오랜 시간이 걸리며, 대부분의 비용을 차지하는 단계는?

- 가. 타당성 검토 단계 나. 운용 및 유지 보수 단계
다. 기본설계 단계 라. 실행 단계

[기-02년9월][기-01년3월]

108. 소프트웨어 유지보수에 대한 설명으로 옳지 않은 것은?

- 가. 소프트웨어 유지보수 비용은 개발비용보다 일반적으로 적다.
나. 소프트웨어 유지보수를 용이하게 하려면 시험용이성,이해성,수정용이성, 이식성이 고려되어야 한다.
다. 소프트웨어 유지보수의 과정은 유지보수요구,현시스템에 대한 이해, 수정 및 시험순을 반복하여 일어난다.
라. 소프트웨어 유지보수는 기능개선,하자보수, 환경적응, 예방조치를 목적으로 소프트웨어의 수명을 연장시키는 작업이다.

[기-03년8월]

109. 유지보수의 종류 중 소프트웨어 산물의 수명 기간 중에 발생하는 환경의 변화를 기존의 소프트웨어 산물에 반영하기

위하여 수행하는 활동을 의미하는 것은?

- 가. 적응(adaptive) 유지보수 나. 완전(perfective) 유지보수
다. 정정(corrective) 유지보수 라. 예방(preventive) 유지보수

[기-04년9월]

110. 소프트웨어 유지보수 유형 중 현재 수행 중인 기능의 수정, 새로운 기능의 추가, 전반적인 기능 개선 등의 요구를 사용자로부터 받았을 때 수행되는 유형으로서, 유지보수 유형별 비용 비율 중 약 50%를 차지하는 것은?

- 가. Preventive maintenance 나. Adaptive maintenance
다. Corrective maintenance 라. Perfective maintenance

[기-04년9월]

111. 소프트웨어 유지보수의 부작용 중 자료코드에 대한 변경이 설계문서나 사용자가 사용하는 메뉴얼에 적용되지 않을 때에 발생하는 부작용은 무엇인가?

- 가. 코딩 부작용 나. 자료 부작용
다. 문서화 부작용 라. 유지보수 부작용

[기-05년3월]

112. 유지보수(Maintenance) 작업의 분류상 가장 큰 비중(업무량 및 비용)을 차지하는 부분은?

- 가. 교정정비(Corrective Maintenance)
나. 조정정비(Adaptive Maintenance)
다. 예방정비(Preventive Maintenance)
라. 완전정비(Perfective Maintenance)

[기-00년10월]

113. 유지보수 비용산정을 구하는 공식이 옳은 것은?

- (단. ① M:유지보수를 위한 노력(인원/월)
② P :생산적인 활동에 드는 비용
③ K :통계값에서 구한 상수
④ c :복잡도
⑤ d :소프트웨어에 대한 지식의 정도)

- 가. $M=P+Ke^{(c-d)}$ 나. $M=P+Ke^{(c+d)}$
다. $M=P*Ke^{(c-d)}$ 라. $M=P*Ke^{(c+d)}$

[기-00년10월]

114. 소프트웨어 유지 보수의 비용분포 중 가장 많은 부분을 차지하는 것은?

- 가. 수정 보수 나. 적응 보수 다. 기능 보수 라. 예방 보수

[기-01년3월]

115. 유지보수(Maintenance) 작업의 분류상 가장 큰 비중(업무량 및 비용)을 차지하는 부분은?

- 가. 교정정비(Corrective Maintenance)
나. 조정정비(Adaptive Maintenance)
다. 예방정비(Preventive Maintenance)
라. 완전정비(Perfective Maintenance)

[기-01년6월]

116. 외계인 코드(Alien Code)에 대한 설명으로 옳은 것은?

- 가. 프로그램의 로직이 복잡하여 이해하기 어려운 프로그램을 의미한다.
나. 아주 오래되어(15년 정도 이상) 유지보수 작업이 어려운 프로그램을 의미한다.
다. 오류(Error)가 없이 완벽하게 수정된 프로그램을 의미한다.
라. 4세대 언어로 사용자가 직접 작성한 프로그램을 의미한다.

[기-07년9월]

117. 소프트웨어 유지보수 유형 중 현재 수행 중인 기능의 수정, 새로운 기능의 추가, 전반적인 기능 개선 등의 요구를 사용자로부터 받았을 때 수행되는 유형으로서, 유지보수 유형 중 제일 많은 비용이 소요되는 것은?

- 가. Preventive maintenance 나. Adaptive maintenance
다. Corrective maintenance 라. Perfective maintenance

(17) 2008년 기출문제(중복제거)

[기-08년3월]

118. 자료흐름도(DFD)의 각 요소별 표기 형태의 연결이 잘못된 것은?

- 가. Process : 원 나. Data Flow : 화살표
다. Data Store : 삼각형 라. Terminator : 사각형

[기-08년3월]

119. 화이트 박스 테스트 기법으로만 짝지어진 것은?

- ① Equivalence Partitioning Test
- ② Comparison Test
- ③ Basic Path Test
- ④ Condition Test
- ⑤ Data Flow Test
- ⑥ Cause-Effect Graphing Test
- ⑦ Loop Test

- 가. ①, ②, ⑦ 나. ②, ③, ④, ⑥, ⑦
다. ①, ②, ⑥ 라. ③, ④, ⑤, ⑦

[기-08년5월]

120. 소프트웨어 설계시 고려 사항으로 거리가 먼 것은?

- 가. 전체적이고 포괄적인 개념을 설계한 후 차례로 세분화하여 구체화시켜 나간다.
나. 요구사항을 모두 구현해야 하고 유지보수가 용이해야 한다.
다. 모듈은 독립적인 기능을 갖도록 설계해야 한다.
라. 모듈간의 상관성은 높이고 변경이 쉬워야 한다.

[기-08년5월]

121. 블랙박스 테스트를 이용하여 발견할 수 있는 오류의 경우로 거리가 먼 것은?

- 가. 비정상적인 자료를 입력해도 오류처리를 수행하지 않는 경우
나. 정상적인 자료를 입력해도 요구된 기능이 제대로 수행되지 않는 경우
다. 반복 조건을 만족하는데도 루프 내의 문장이 수행되지 않는 경우
라. 경계값을 입력할 경우 요구된 출력 결과가 나오지 않는 경우

[기-08년5월]

122. 응집도의 종류 중 서로간에 어떠한 의미 있는 연관관계도 지니지 않은 기능요소로 구성되는 경우이며, 서로 다른 기능을 수행하는 경우의 응집도는?

- 가. Coincidental Cohesion 나. Functional Cohesion
다. Sequential Cohesion 라. Logical Cohesion

[기-08년9월]

123. 다음 사항과 관계되는 결합도는 무엇인가?

- 한 모듈에서 다른 모듈의 내부로 제어 이동
- 한 모듈이 다른 모듈 내부 자료의 조회 또는 변경
- 두 모듈이 동일한 문자(Literals)의 공유

- 가. Data Coupling 나. Content Coupling
다. Control Coupling 라. Stamp Coupling

[기-08년9월]

124. 다음의 소프트웨어 검사 기법 중 성격이 나머지 셋과 다른 하나는?

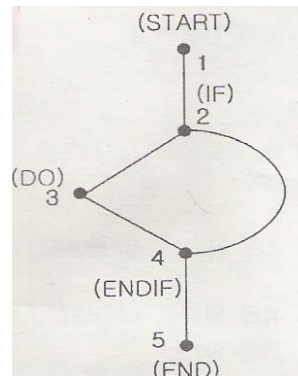
- 가. Equivalence partitioning test
나. Boundary value analysis
다. Comparison test
라. Loop test

[기-08년9월]

125. 어떤 프로그램을 재공학 기술을 적용하여 보수하고자 할 때 Flow Graph가 사용될 수 있다. 다음의 샘플 프로그램에 대한 Flow Graph가 다음 그림과 같을 때 McCabe식의 Cyclomatic complexity를 구하면?

```

1 START
2 IF <condition> THEN
3 DO A
4 ENDIF
5 END
    
```



- 가. 1 나. 2 다. 3 라. 4

[기-09년5월][기-08년9월]

126. 유지보수의 종류 중 소프트웨어 테스트 중 밝혀지지 않는 모든 잠재적인 오류를 찾아 수정하는 활동에 해당하는 것은?

- 가. Corrective Maintenance 나. Adaptive Maintenance
다. Perfective Maintenance 라. Preventive Maintenance

(18) 2009년 기출문제(중복제거)

[기-09년3월]

127. DFD(Data Flow Diagram)에 대한 설명으로 거리가 먼 것은?

- 가. 단말(Terminator)은 원으로 표기한다.
 나. 구조적 분석 기법에 이용된다.
 다. 자료 흐름과 기능을 자세히 표현하기 위해 단계적으로 세분화된다.
 라. 자료 흐름 그래프 또는 버블(Bubble) 차트라고도 한다.

[기-09년8월][기-09년3월]

128. 유지보수의 종류 중 다음 설명에 해당하는 것은?

소프트웨어를 운용하는 환경 변화에 대응하여 소프트웨어를 변경하는 경우로써 운영체제나 컴파일러와 같은 프로그래밍 환경의 변화와 주변장치 또는 다른 시스템 요소가 향상되거나 변경될 때 대처할 수 있는 유지보수이다.

- 가. Preventive maintenance
 나. Corrective maintenance
 다. Perfective maintenance
 라. Adaptive maintenance

[기-09년3월]

129. 화이트박스 테스트에 대한 설명으로 옳지 않은 것은?

- 가. 제품의 내부 요소들이 명세서에 따라 수행되고 충분히 실행되는가를 보장하기 위한 검사이다.
 나. 모듈 안의 작동을 직접 관찰한다.
 다. 프로그램 원시 코드의 논리적인 구조를 커버하도록 테스트 케이스를 설계한다.
 라. 화이트박스 테스트 기법에는 조건 검사, 루프 검사, 비교 검사 등이 있다.

[기-09년5월]

130. 설계 기법 중 하향식 설계 방법과 상향식 설계 방법에 대한 비교 설명으로 옳지 않은 것은?

- 가. 하향식 설계에서는 통합 검사시 인터페이스가 이미 정의되어 있어 통합이 간단하다.
 나. 하향식 설계에서 레벨이 낮은 데이터 구조의 세부 사항은 설계 초기 단계에서 필요하다.
 다. 상향식 설계는 최하위 수준에서 각각의 모듈들을 설계하고 이러한 모듈이 완성되면 이들을 결합하여 검사한다.
 라. 상향식 설계에서는 인터페이스가 이미 성립되어 있지 않더라도 기능 추가가 쉽다.

[기-09년5월]

131. 프로그램 설계도의 하나인 NS(Nassi-Schneiderman) chart에 대한 설명으로 옳지 않은 것은?

- 가. 논리의 기술에 중점을 두고 도형을 이용한 표현 방법이다.
 나. 박스, 다이아몬드, 화살표 등의 기호를 사용하므로 읽고 작성하기가 매우 쉽다.
 다. 이해하기 쉽고 코드로 변환이 용이하다.
 라. 연속, 선택, 반복 등의 제어 논리 구조를 표현한다.

[기-09년5월]

132. 시스템에서 모듈 사이의 결합도(Coupling)에 대한 설명으로 옳은 것은?

- 가. 한 모듈 내에 있는 처리요소들 사이의 기능적인 연관 정도를 나타낸다.
 나. 결합도가 높으면 시스템을 구현하고 유지보수 작업이 쉽다.
 다. 모듈간의 결합도를 약하게 하면 모듈 독립성이 향상된다.
 라. 자료결합도는 내용결합도 보다 결합도가 높다.

[기-09년8월]

133. 다음 중 검증시험(Validation Test)과 거리가 먼 것은?

- 가. 알파(Alpha) 테스트
 나. 베타(Beta) 테스트
 다. 블랙박스 (Black-Box) 테스트
 라. 화이트박스 (White-Box) 테스트

[SE03-구조적 개발 방법론]

1	2	3	4	5	6	7	8	9	10
다	라	라	라	가	라	라	라	나	가
11	12	13	14	15	16	17	18	19	20
가	다	다	나	다	가	다	다	다	나
21	22	23	24	25	26	27	28	29	30
다	라	다	나	가	가	다	다	나	다
31	32	33	34	35	36	37	38	39	40
다	라	나	라	가	나	다	다	다	다
41	42	43	44	45	46	47	48	49	50
다	나	라	다	라	라	나	가	가	가
51	52	53	54	55	56	57	58	59	60
라	라	나	라	나	나	다	나	나	다
61	62	63	64	65	66	67	68	69	70
라	다	가	나	다	라	다	다	나	라
71	72	73	74	75	76	77	78	79	80
가	다	나	나	다	나	다	나	나	다
81	82	83	84	85	86	87	88	89	90
라	가	나	가	가	다	나	라	다	나
91	92	93	94	95	96	97	98	99	100
다	나	라	라	나	다	가	나	라	라
101	102	103	104	105	106	107	108	109	110
나	나	라	라	라	가	나	가	가	라
111	112	113	114	115	116	117	118	119	120
다	라	가	다	라	나	라	다	라	라
121	122	123	124	125	126	127	128	129	130
다	가	나	라	나	가	가	라	라	라
131	132	133	134	135	136	137	138	139	140
나	다	라							