

[실전 모의고사 4회]

1	2	3	4	5	6	7	8	9	10
2	4	3	2	1	2	4	2	2	4
11	12	13	14	15	16	17	18	19	20
2	1	2	2	1	3	4	3	2	1
21	22	23	24	25	26	27	28	29	30
2	2	2	4	1	2	3	1	4	4
31	32	33	34	35	36	37	38	39	40
3	1	4	1	4	2	3	3	1	4
41	42	43	44	45	46	47	48	49	50
3	1	3	2	4	4	1	2	2	1
51	52	53	54	55	56	57	58	59	60
2	2	3	4	3	1	4	2	4	1
61	62	63	64	65	66	67	68	69	70
3	4	3	1	2	4	4	1	2	4
71	72	73	74	75	76	77	78	79	80
2	3	2	4	1	1	2	2	3	4
81	82	83	84	85	86	87	88	89	90
1	3	3	3	2	1	1	4	4	3
91	92	93	94	95	96	97	98	99	100
4	3	2	4	2	2	1	1	3	4

1. ②

- 브리지(Bridge): 두 개의 LAN이 데이터 링크 계층에서 서로 결합되어 있는 경우에 이들을 연결하는 요소로 OSI 2계층 장비이다.

2. ④

- 데이터베이스 설계 순서
: 요구조건 분석→개념적 설계→논리적 설계→물리적 설계→구현→운영→감시 및 개선

3. ③

③ 경영 시스템의 이해 관계자로 프로게이머(Professional Gamer)는 거리가 멀다.

4. ②

행위 다이어그램은 동적 모델링에 속하며, 구조 다이어그램은 정적 모델링에 속한다.
- 구조 다이어그램: 클래스 다이어그램, 객체 다이어그램, 패키지 다이어그램, 컴포넌트 다이어그램, 복합 구조 다이어그램, 배치 다이어그램, 프로파일 다이어그램
- 행위 다이어그램: 유스케이스 다이어그램, 활동 다이어그램, 상태 다이어그램, 시퀀스 다이어그램, 통신 다이어그램, 타이밍 다이어그램, 상호작용 개요 다이어그램

5. ①

XP 방법론은 짧고 반복적인 개발 주기, 단순한 설계, 고객의 적극적인 참여를 통해 소프트웨어를 빠르게 개발하는 것을 목적으로 한다.

6. ②

CASE는 소프트웨어 생명주기 전 과정을 자동화할 수 있도록 지원하는 자동화 도구이다.

7. ④

- 분석 클래스의 클래스 유형

분석 클래스 유형	스테레오 타입	아이콘
경계 클래스	<<boundary>>	
제어 클래스	<<control>>	
엔티티 클래스	<<entity>>	

8. ②

컴퓨터 사용자 인터페이스(Computer User Interface)에 대한 설명이다.

9. ②

② 디자이너와 개발자가 최종적으로 참고하는 작업 지침서로, 정책, 프로세스, 콘텐츠 구성, 와이어프레임, 기능 정의 등 서비스 구축을 위한 모든 정보가 들어 있다.

10. ④

④ 비기능적 요구사항이다.

11. ②

② UI 흐름 설계에 대한 내용이다.

12. ①

② 감성공학 2류에 대한 설명이다.
③ 감성공학 2류에 대한 설명이다.
④ 감성공학 3류에 대한 설명이다.

13. ②

② 전체 프로그램의 기능 중 공통적으로 사용할 수 있는 모듈을 공통 모듈로 모듈화 할 수 있으므로 프로그램 코딩 양이 적어진다.

14. ②

② 설계 모델링은 소프트웨어의 내부 기능 외에 구조 및 동적 행위들을 모델링하여 표현, 분석, 검증하는 과정이다.

15. ①

소프트웨어 아키텍처 4+1 View의 논리 뷰(Logical View)에 대한 설명이다.

16. ③

캡슐화(encapsulation)에 대한 설명이다.

17. ④

④ 구조 패턴에 대한 설명이다.

18. ③

③ 검증가능성에 대한 내용이다.

- 일관성: 명세 내용 간의 상호간 모순이 없어야 한다.

19. ②



20. ①

인터페이스에 대한 설명이다.

21. ②

큐(Queue)의 응용 분야

- 운영체제의 작업 스케줄링, 키보드 버퍼 이용 시, 스펴(pool) 운용 시

22. ②

- 트리의 차수: 트리의 최대 차수(A의 차수: 3)

23. ②

동적 인덱스 방법에 대한 설명이다.

24. ④

모델링은 크게 세 가지 관점인 데이터관점, 프로세스관점, 데이터와 프로세스의 상관관점으로 구분하여 설명할 수 있다.

- 1) 데이터관점 : 업무가 어떤 데이터와 관련이 있는지 또는 데이터간의 관계는 무엇인지에 대해서 모델링하는 방법(What, Data)
- 2) 프로세스관점 : 업무가 실제로 하고 있는 일은 무엇인지 또는 무엇을 해야 하는지를 모델링하는 방법(How, Process)
- 3) 데이터와 프로세스의 상관관점 : 업무가 처리하는 일의 방법에 따라 데이터는 어떻게 영향을 받고 있는지 모델링하는 방법(Interaction)

25. ①

- 이상(anomaly)

: 릴레이션에서 일부 속성들의 종속으로 인해 데이터의 중복이 발생하여 테이블 조작 시 불일치가 발생하는 것

26. ②

* DDL(데이터 정의어)

- 도메인, 테이블, 뷰, 인덱스를 정의, 변경, 제거하는 언어
- 종류: CREATE, ALTER, DROP

27. ③

③ 반드시 반환될 값의 데이터 타입을 Return문에 선언해야 한다.

28. ①

- 원자성(Automicity): 트랜잭션에 포함된 작업들은 모두 수행되거나 아니면 전혀 수행되지 않아야 한다.(all or nothing)

29. ④

- SQL 성능 개선 순서

: 문제 있는 SQL 식별→옵티마이저(Optimizer) 통계 확인→SQL 문 재구성→인덱스 재구성→실행계획 유지관리

30. ④

④ Github는 형상관리 도구이다.

31. ③

③ 사용자 환경과 똑같은 환경에서 패키징 적용을 해본다.

32. ①

DRM(Digital Rights Management)에 대한 설명이다.

33. ④

* 제품소프트웨어 설치 매뉴얼의 기본 작성 항목

- 목차 및 개요: 매뉴얼 전체 내용에 대한 요약
- 문서 이력 정보: 매뉴얼 변경 이력 정보
- 설치 도구 구성: exe/dll/ini/chm 등 설치 관련 파일 설명
- 설치 위치 지정: 설치 폴더와 설치 프로그램 실행 파일 설명
- 설치를 위한 환경 체크 항목

34. ①

② 클라이언트/서버 방식이다.

③ 분산 저장소 방식이다.

④ 클라이언트/서버 방식이다.

35. ④

④ CI는 소스 코드 일관성을 위해 소스 관리 시스템이 필요하므로, 비용이 증가될 수 있다.

36. ②

* 테스트 레벨

- 1) 단위 테스트
- 2) 통합 테스트
- 3) 시스템 테스트
- 4) 인수 테스트: 알파 테스트, 베타 테스트

37. ③

③ 베타 시험에 대한 설명이다.

* 인수 테스트

- 알파 테스트: 개발자 환경에서 테스트하는 기법
- 베타 테스트: 사용자 환경에서 테스트하는 기법

38. ③

③ 인수 테스트 방법이다.

39. ①

* Factory Method(팩토리 메서드)

: 객체 생성을 위한 인터페이스를 정의한 후 상속한 서브 클래스를 이용하여 객체를 생성한다.

40. ④

* 인터페이스 기능 확인 방법

- 상세 인터페이스 기능은 인터페이스 설계서(정의서)를 보고 확인한다.
- 시스템 정적/동적 기능분석을 통하여 내/외부 인터페이스 기능을 확인한다.
- 인터페이스 되는 데이터 유형, 값의 범위, 예외처리 규칙을 확인한다.

41. ③

③ 분산 처리에 대한 설명이다.

42. ①

① 사용자는 자신이 원하는 바(what)만 명시하며, 원하는 것을 처리하는 방법(how)은 명시하지 않는다.

43. ③

③ DML 명령어이다.

* DDL 명령어

- CREATE: 데이터베이스 객체 생성
- ALTER: 데이터베이스 객체 변경
- DROP: 데이터베이스 객체 삭제
- TRUNCATE: 데이터베이스 객체 내용 삭제

44. ②

* 트랜잭션의 특징

- 원자성(Atomicity) - 일관성(Consistency)
- 고립성(Isolation) - 영속성(Durability)

45. ④

뷰(VIEW)에 대한 설명이다.

46. ④

④ ANY에 대한 설명이다.

* Multi Row(다중 행) 서브쿼리

- IN: 컬럼의 값이 서브쿼리의 결과 값들의 집합에 속하는지 확인
- ANY: 컬럼의 값이 서브쿼리의 결과 값들의 집합에 속하는 하나 이상의 값들과 어떤 관계를 갖는지 확인
- ALL: 컬럼의 값이 서브쿼리의 결과 값들의 집합에 속하는 모든 값들과 어떤 관계를 갖는지 확인
- EXISTS: 서브쿼리의 결과가 존재하는지 확인

47. ①

① 한 개의 애트리뷰트에 적용되어 단일 값을 반환한다.

* 집계 함수 특징

- 한 개의 애트리뷰트에 적용되어 단일 값을 반환한다.
- SELECT절과 HAVING절에만 나타날 수 있다.
- COUNT(*)를 제외한 모든 집계 함수는 널 값을 제거한 후 계산한다.
- COUNT(*)는 결과 릴레이션의 모든 행들의 총 개수를 반환한다.
- COUNT(애트리뷰트)는 널 값이 아닌 값들의 개수를 반환한다.
- DISTINCT가 집단 함수 앞에 사용되면 집단 함수가 적용되기 전에 중복을 제거한다.

48. ②

트리거(Trigger)에 대한 설명이다.

49. ②

* 관계형 데이터 모델 구성 요소

- 구조(Structure) - 연산(Operation)
- 제약조건(Constraint)

50. ①

① 데이터베이스의 논리적 구조가 트리(tree) 형태이다.

51. ②

튜플(Tuple)에 대한 설명이다.

52. ②

데이터 모델링에 대한 설명이다.

53. ③

③ 새로운 형태의 데이터가 삽입될 때 관계를 재구성할 필요성을 줄인다.

54. ④

④ 불규칙한 응답 속도

55. ③

③ 특정 데이터가 어느 Hash Partition에 있는지 판단 불가

56. ①

Mapping Rule에 대한 설명이다.

57. ④

* 물리 데이터 모델 품질 기준

- 정확성 - 완전성
- 준거성 - 최신성
- 일관성 - 활용성

58. ②

ETL에 대한 설명이다.

59. ④

* 데이터 전환 절차

: 데이터 전환 계획 및 요건 정의→데이터 전환설계→데이터 전환개발→데이터 전환 테스트 및 검증→데이터 전환

60. ①

직접 파일에 대한 설명이다.

61. ③

SSD(Solid-State Drive)에 대한 설명이다.

62. ④

Add plug-In에 대한 설명이다.

63. ③

③ 코드 재사용은 장황한 작업에 소비하는 시간과 에너지를 절약하는 전형적 기법이다.

64. ①

* 응집도 단계(높→낮)

: 기능적 응집도-순차적 응집도-통신적 응집도-절차적 응집도-시간적 응집도-논리적 응집도-우연적 응집도

65. ②

② 입력 확인 오류에 의한 취약점이다.

66. ④

④ 윈도 셸에 대한 설명이다.

- 웹: 인터넷 익스플로러 웹 브라우저 또한 응용 프로그램에 자주 쓰이는 수많은 API를 노출하며 이러한 것들은 윈도 API의 일부로 간주할 수 있다.

67. ④

④ C/C++에서 사용하는 변수 타입이다.

* Java에서 사용하는 변수 타입

- byte - short
- int - long

68. ①

① 비트 연산자이다.

* 산술 연산자 종류: +, -, *, /, %

69. ②

② GOTO문, JUMP문 사용이 아닌 쉬운 흐름 파악이 가능하다.

* GOTO문, JUMP문 (실행 순서 제어)

- 절차적 프로그래밍 이전 주로 사용하던 제어 방식 [코드 이해 어려움(가독성X)→유지 보수 쉽지 않음]
- GOTO문이나 JUMP문을 많이 쓰면 크고 복잡한 프로그램이 일명 스파게티 코드가 될 수도 있다.

70. ④

④ 절차지향 프로그래밍에 대한 설명이다.

71. ②

② 짧은 소스 코드 파일이나 REPL로 상호작용하는 데 적합하다.

72. ③

③ \b에 대한 설명이다.

* 제어 문자

- \n: new line : 커서를 다음 줄로 바꾼다.
- \r: carriage return : 커서를 그 줄의 맨 앞으로 이동
- \f: form feed : 한 페이지를 넘긴다.
- \b: backspace : 커서를 그 줄의 1문자만큼 앞으로 이동
- \t: tab : 커서를 그 줄의 tab 만큼 이동

73. ②

OLAP(On-Line Analytical Processing, 온라인 분석 처리)에 대한 설명이다.

74. ④

④ 실행 가능한 목적(object) 프로그램 생성(X)→컴파일러

75. ①

세그멘테이션(Segmentation) 기법에 대한 설명이다.

76. ①

* 프로세스 제어 블록의 저장 정보

- 프로세스의 현 상태
- 프로세스의 우선 순위
- 프로세스 식별자(고유 번호)
- 레지스터 저장 장소
- 관련 레지스터 정보
- 할당된 자원에 대한 포인터

77. ②

IP 주소(Internet Protocol Address)에 대한 설명이다.

78. ②

* 프로토콜의 기본 요소

- 구문 (Syntax) : 데이터 형식
- 타이밍 (Timing) : 순서, 속도 조절
- 의미 (Semantics) : 오류, 제어 정보

79. ③

③ Replication에 대한 설명이다.

* MySQL의 기능

- Cross-platform support: 다양한 종류의 플랫폼에 대해 최적화된 바이너리 버전을 제공한다.
- Independent storage engines: 독립적인 스토리지 엔진을 이용하여 사용자의 구체적인 요구에 적합한 스토리지 방식을 사용한다.

- Transactions: MySQL 데이터베이스 서버는 InnoDB 또는 Berkeley DB(BDB) 스토리지 엔진을 이용해 트랜잭션을 지원한다.
- Query caching: 추가적인 프로그래밍 없이 일반적으로 자주 사용되는 쿼리의 성능을 증가시켜준다.
- Replication: 하나의 “주” 서버를 복제한 다수의 “종속” 서버를 구성하여 빠른 성능을 가진 시스템을 구축할 수 있다.

80. ④

④ 준비(Preparation) 단계에 대한 설명이다.

81. ①

폭포수 모델에 대한 설명이다.

82. ③

③ 객체지향 방법론에 대한 설명이다.

83. ③

* 애자일 방법론의 특징

- 변화대응
- 동작소프트웨어
- 고객협력: 계약협상 중심에서 탈피
- 개인과 상호작용

84. ③

③ 정량적 관리 단계: 예측 가능한 프로세스

* CMMI 프로세스

- 초기(initial) 단계: 프로세스 없음
- 관리(managed) 단계: 규칙화된 프로세스
- 정의(defined) 단계: 표준화된 프로세스
- 정량적 관리(quantitatively managed) 단계: 예측 가능한 프로세스
- 최적화(optimizing) 단계: 지속적 개선 프로세스

85. ②

② 자동화에 따른 테일러링

86. ①

버스형 구조에 대한 설명이다.

87. ①

① 애플리케이션 개발자 및 보안 담당자는 시험계획서에 따라 보안 기능을 시험한다.

88. ④

RFID(Radio Frequency IDentification)에 대한 설명이다.

89. ④

④ 시스템 활동 최대

* 동시성 제어 기능의 목적

- 트랜잭션의 직렬성 보장

- 공유도 최대, 응답 시간 최소, 시스템 활동의 최대 보장
- 데이터의 무결성 및 일관성 보장

90. ③

③ 데이터 형식 사용 시 고려사항이다.

91. ④

④ 배포·운영에 대한 설명이다.

* MS-SDL 절차

- 교육: 소프트웨어 보안 교육
- 계획·분석: 소프트웨어의 질과 버그 경제 정의, 보안과 프라이버시 위험분석
- 설계: 공격영역분석, 위험모델링
- 구현: 도구명세, 금지된 함수 사용제한, 정적분석
- 시험·검증: 동적/Fuzzing 테스트, 공격영역/위험모델 검증
- 배포·운영: 사고대응계획, 최종보안검토, 기록보관
- 대응: 사고대응수행

92. ③

③ 보안기능이다.

* 입력데이터 검증 및 표현

- SQL삽입
- 크로스사이트스크립트(XSS)
- 크로스사이트 요청 위조(CSRF)
- 위험한 형식 파일 업로드
- 디렉토리 경로 조작

93. ②

한계 검사(limit check)에 대한 설명이다.

94. ④

④ 대표적인 알고리즘으로 DES, AES가 있다.

- RSA는 비대칭키 방식의 대표적인 알고리즘이다.

95. ②

지능형 가상 비서(IPA)에 대한 설명이다.

96. ②

TearDrop 공격에 대한 설명이다.

97. ①

소유기반 인증에 대한 설명이다.

98. ①

① 개인의 물리적 접근 제어에 대한 설명이다.

- 서버보안에서의 접근 제어: 사람이거나 프로세스가 시스템이나 파일에 읽기, 쓰기, 실행 등의 접근 여부를 허가하거나 거부하는 기능을 말한다.

99. ③

③ 시스템로그 - /var/log/messages

메일로그 - /var/log/maillog

100. ㉔

Anti-DDos에 대한 설명이다.