**정보처리기사**

2021년 05월 15일 기출문제

**1과목**

**1.시스템의 구성요소로 볼 수 없는 것은?**

|  |  |
| --- | --- |
| 1. | Process |

|  |  |
| --- | --- |
| 2. | Feedback |

|  |  |
| --- | --- |
| 3. | Maintenance |

|  |  |
| --- | --- |
| 4. | Control |

<문제 해설>  
시스템 구성요소  
  
입력 (Input) : 처리 방법, 처리할 데이터, 조건을 시스템에 투입하는 것  
처리 (Process) : 입력된 데이터를 처리 방법과 조건에 따라 처리하는 것  
출력 (Output) : 처리된 결과를 시스템에서 산출하는 것  
제어 (Control) : 자료를 입력하여 출력될 때까지의 처리 과정이 올바르게 진행되는지 감독하는 것  
피드백 (Feedback) : 출력된 결과가 예정된 목표를 만족시키지 못할 경우 목표 달성을 위해 반복 처리하는 것  
  
Maintenance는 유지보수로서, 시스템 구성요소에는 포함되지 않음.

|  |  |
| --- | --- |
| **6.** | **GoF (Gangs of Four) 디자인 패턴에 대한 설명으로 틀린 것은?** |

* 교재\_132

|  |  |
| --- | --- |
| 1. | factory method pattern은 상위클래스에서 객체를 생성하는 인터페이스를 정의하고, 실질적으론 하위(서브)클래스에서 인스턴스(객체)를 생성하도록 하는 방식이다. |

|  |  |
| --- | --- |
| 2. | prototype pattern은 prototype을 먼저 생성하고 인스턴스를 복제하여 사용하는 구조이다. |

|  |  |
| --- | --- |
| 3. | bridge pattern은 기존에 구현되어 있는 클래스에 기능 발생 시 기존 클래스를 재사용할 수 있도록 중간에서 맞춰주는 역할을 한다. |

|  |  |
| --- | --- |
| 4. | mediator pattern은 객체간의 통제와 지시의 역할을 하는 중재자를 두어 객체지향의 목표를 달성하게 해준다. |

<문제 해설>  
bridge pattern은 구현부(구현)에서 추상층(기능)을 분리하여 각자 독립적으로 확장이 가능하게 하는 패턴이다. 3은 Adapter pattern의 설명

|  |  |
| --- | --- |
| **8.** | **소프트웨어 아키텍처 설계에서 시스템 품질속성이 아닌 것은?** |

* 교재\_108

|  |  |
| --- | --- |
| 1. | 가용성 (Availability) |

|  |  |
| --- | --- |
| 2. | 독립성 (Isolation) |

|  |  |
| --- | --- |
| 3. | 변경 용이성 (Modifiability) |

|  |  |
| --- | --- |
| 4. | 사용성(Usability) |

|  |  |
| --- | --- |
|  | <문제 해설> 시스템 품질 속성 |

성능, 기능성, 가용성, 사용성, 사용편의성,

변경용이성, 확장성, 보안성,

시험용의성, 테스트 용이성, 배치성, 안정성 등(2)

|  |  |
| --- | --- |
| **10.** | **CASE(Computer-Aided Software Engineering)의 원천 기술이 아닌 것은?** |

|  |  |
| --- | --- |
| 1. | 구조적 기법 |

|  |  |
| --- | --- |
| 2. | 프로토타이핑 기술 |

|  |  |
| --- | --- |
| 3. | 정보 저장소 기술 |

|  |  |
| --- | --- |
| 4. | 일괄처리 기술 |

|  |  |
| --- | --- |
|  | <문제 해설> CASE의 원천 기술은 구조적 기법, 프로토타이핑 기술, 자동프로그래밍 기술, 정보 저장소 기술, 분산 처리 기술이 있다.(4) |

|  |  |
| --- | --- |
| **12.** | **서브시스템이 입력 데이터를 받아 처리하고 결과를 다른 시스템에 보내는 작업이 반복되는 아키텍처 스타일은?** |

|  |  |
| --- | --- |
| 1. | 클라이언트 서버 구조 |

|  |  |
| --- | --- |
| 2. | 계층 구조 |

|  |  |
| --- | --- |
| 3. | MVC 구조 |

|  |  |
| --- | --- |
| 4. | 파이프 필터 구조 |

|  |  |
| --- | --- |
|  | <문제 해설> 1. 클라이언트 서버 구조 : 컴포넌트가 다른 컴포넌트에게 서비스를 요청. 데이터가 여러 컴포넌트를 거치며 처리. 2. 계층구조 : 모듈들로 응집된 계층 단위로 SW를 구성. 계층간에 사용 가능의 관계로 표현 3. MVC 구조 : 모델-뷰-컨트롤러, 기능을 분리한 아키텍처 4. 파이프 필터 : 파이프를 통해 받은 데이터를 변경시키고 그 결과를 파이프로 전송 |

|  |  |
| --- | --- |
| **13.** | **럼바우(Rumbaugh)의 객체지향 분석에서 사용하는 분석 활동으로 옳은 것은?** |

|  |  |
| --- | --- |
| 1. | 객체 모델링, 동적 모델링, 정적 모델링 |

|  |  |
| --- | --- |
| 2. | 객체 모델링, 동적 모델링, 기능 모델링 |

|  |  |
| --- | --- |
| 3. | 동적 모델링, 기능 모델링, 정적 모델링 |

|  |  |
| --- | --- |
| 4. | 정적 모델링, 객체 모델링, 기능 모델링 |

|  |  |
| --- | --- |
|  | <문제 해설> 럼바우는 객체 모델링, 동적 모델링, 기능 모델링 : 객동기 추가로 동적 모델링-상태도, 기능 모델링-자료 흐름도 : 동상기자로 암기 |

|  |  |
| --- | --- |
| **14.** | **UML 다이어그램이 아닌 것은?** |

* 교재\_56

|  |  |
| --- | --- |
| 1. | 액티비티 다이어그램(Activity diagram) |

|  |  |
| --- | --- |
| 2. | 절차 다이어그램(Procedural diagram) |

|  |  |
| --- | --- |
| 3. | 클래스 다이어그램(Class diagram) |

|  |  |
| --- | --- |
| 4. | 시퀀스 다이어그램(Sequence diagram) |

|  |  |
| --- | --- |
|  | <문제 해설> 업무의 흐름을 모델링하거나 객체의 생명 주기를 표현하는 Activity 다이어그램 객체 간의 메시지 전달을 시간적 흐름에서 분석하는 Sequence 다이어그램 기업 환경의 구성과 컴포넌트들 간의 관계를 그린 Deployment 다이어그램 객체와 객체가 주고받는 메시지 중심의 작성 동적 다이어그램인 Collaboration 다이어그램 소프트웨어 구조가 그리는 Component 다이어그램 시스템의 구조적인 모습을 그리는 Class 다이어그램(2)  액시디콜콤클   * UML = Unified Modeling Language(객체지향모델링언어, 표기법) |

|  |  |
| --- | --- |
| **15.** | **UML 모델에서 한 객체가 다른 객체에게 오퍼레이션을 수행하도록 지정하는 의미적 관계로 옳은 것은?** |

|  |  |
| --- | --- |
| 1. | Dependency |

|  |  |
| --- | --- |
| 2. | Realization |

|  |  |
| --- | --- |
| 3. | Generalization |

|  |  |
| --- | --- |
| 4. | Association |

|  |  |
| --- | --- |
|  | <문제 해설> Dependency(의존) : 한 사물의 명세서가 바뀌면 그것을 사용하는 다른 사물에게 영향을 끼치는 것을 말합니다 (Cascade 생각하셈) Realization(실체화) : 한 객체가 다른 객체에 의해 오퍼레이션을 수행하도록 지정 Generalization(일반화) : 일반화된 사물과 좀 더 특수화된 사물 사이의 관계를 말합니다.('is-a')관계 Association(연관) : 두 사물간의 구조적 관계로, 어느 한 사물 객체가 다른 사물 객체와 연결되어 있음을 말함 ('has-a')관계라고도 합니다(2) |

|  |  |
| --- | --- |
| **16.** | **다음 중 상위 CASE 도구가 지원하는 주요기능으로 볼 수 없는 것은?** |

|  |  |
| --- | --- |
| 1. | 모델들 사이의 모순검사 기능 |

|  |  |
| --- | --- |
| 2. | 전체 소스코드 생성 기능 |

|  |  |
| --- | --- |
| 3. | 모델의 오류검증 기능 |

|  |  |
| --- | --- |
| 4. | 자료흐름도 작성 기능 |

|  |  |
| --- | --- |
|  | <문제 해설> CASE: 시스템 개발과정의 일부 또는 전체를 자동화시킨 것 - 소프트웨어 생명주기의 전체 단계를 연결해 주고 자동화 해주는 통합된 도구를 제공 - 소프트웨어, 하드웨어, 데이터베이스, 테스트 등을 통합하여 소프트웨어를 개발하는 환경을 제공  상위 CASE: 요구 분석과 설계 단계를 지원 - 모델들 사이의 모순검사 기능 - 모델의 오류 검증 기능 - 자료흐름도 작성 기능  하위 CASE: 코드를 작성하고 테스트하며 문서화하는 과정 지원 - 원시코드 생성 기능  통합 CASE: 소프트웨어 개발 주기 전체과정을 지원 |

|  |  |
| --- | --- |
| **17.** | **요구사항 관리 도구의 필요성으로 틀린 것은?** |

|  |  |
| --- | --- |
| 1. | 요구사항 변경으로 인한 비용 편익 분석 |

|  |  |
| --- | --- |
| 2. | 기존 시스템과 신규 시스템의 성능 비교 |

|  |  |
| --- | --- |
| 3. | 요구사항 변경의 추적 |

|  |  |
| --- | --- |
| 4. | 요구사항 변경에 따른 영향 평가 |

|  |  |
| --- | --- |
|  | <문제 해설> 2번은 개발, 설계 등 구현단계 때 |

|  |  |
| --- | --- |
| **18.** | **애자일 개발 방법론이 아닌 것은?** |

|  |  |
| --- | --- |
| 1. | 스크럼(Scrum) |

|  |  |
| --- | --- |
| 2. | 익스트림 프로그래밍(XP, eXtreme Programming) |

|  |  |
| --- | --- |
| 3. | 기능 주도 개발(FDD, Feature Driven Development) |

|  |  |
| --- | --- |
| 4. | 하둡(Hadoop) |

|  |  |
| --- | --- |
|  | <문제 해설> 애자일 개발 방법론 - 익스트림 프로그래밍(XP , Extreme Programming) - 스크럼크리스털 패밀리 - 기능 주도 개발(FDD , Feature-Driven Development) - 적응형 소프트웨어 개발 (ASD , Adaptive Software Development( - 익스트림 모델링  \* 하둡(Hadoop) 분산 환경에서 빅 데이터를 저장하고 처리할 수 있는 자바 기반의 오픈 소스 프레임 워크. |

|  |  |
| --- | --- |
| **19.** | **GoF(Gangs of Four) 디자인 패턴 중 생성패턴으로 옳은 것은?** |

|  |  |
| --- | --- |
| 1. | singleton pattern |

|  |  |
| --- | --- |
| 2. | adapter pattern |

|  |  |
| --- | --- |
| 3. | decorator pattern |

|  |  |
| --- | --- |
| 4. | state pattern |

|  |  |
| --- | --- |
|  | <문제 해설> 생성패턴: 객체를 생성하는 것에 대한 패턴 추상팩토리 패턴, 빌더 패턴, 팩토리 메소드 패턴, 프로토타입 패턴, 싱글톤 패턴  구조 패턴: 구조를 통해 확장성을 꾀하는 패턴 어댑터 패턴, 브릿지 패턴, 컴포지트 패턴, 데코레이터 패턴, 퍼사드 패턴, 플라이웨잇 패턴, 프록시 패턴  행위 패턴: 행위의 변경, 수정 등을 위한 패턴 역할 사슬 패턴, 커맨드 패턴, 인터프리터 패턴, 이터레이터 패턴, 미디에이터 패턴, 메멘토 패턴, 옵저버 패턴, 스테이트 패턴, 스트래티지 패턴, 템플릿 메소드 패턴, 비지터 패턴  1. singleton pattern(단일체): 한 클래스에 한 객체만 존재하도록 제한한다(생성패턴) 2. adapter pattern(적응자): 인터페이스가 호환되지 않는 클래스들을 함께 이용할 수 있도록, 타 클래스의 인터페이스를 기존 인터페이스에 덧씌운다(구조패턴) 3. decorator pattern(장식자): 0개, 1개 혹은 그 이상의 객체를 묶어 하나의 객체로 이용할 수 있다(구조패턴) 4. state pattern(상태): 동일한 동작을 객체의 상태에 따라 다르게 처리해야 할 때 사용하는 디자인 패턴(행위패턴) |

**2과목**

|  |  |
| --- | --- |
| **31.** | **다음 자료를 버블 정렬을 이용하여 오름차순으로 정렬할 경우 Pass 2의 결과는?** |

https://img.comcbt.com/cbt/data/iz/iz20210515/iz20210515m31.gif?1621458069

|  |  |
| --- | --- |
| 1. | 3, 5, 6, 7, 9 |

|  |  |
| --- | --- |
| 2. | 6, 7, 3, 5, 9 |

|  |  |
| --- | --- |
| 3. | 3, 5, 9, 6, 7 |

|  |  |
| --- | --- |
| 4. | 6, 3, 5, 7, 9 |

|  |  |
| --- | --- |
|  | <문제 해설> Pass 1  9 6 7 3 5  →  6 9 7 3 5  →  6 7 9 3 5  →  6 7 3 9 5  →  6 7 3 5 9  Pass 2  6 7 3 5 9  →  6 3 7 5 9  →  6 3 5 7 9 (4)  [해설작성자 : comcbt.com 이용자]  버블 정렬은 첫 번째 자료와 두 번째 자료를, 두 번째 자료와 세 번째 자료를, 세 번째와 네 번째를, … 이런 식으로 (마지막-1)번째 자료와 마지막 자료를 비교하여 교환하면서 자료를 정렬한다. |

|  |  |
| --- | --- |
| **35.** | **인터페이스 구현 검증 도구가 아닌 것은?** |

|  |  |
| --- | --- |
| 1. | Foxbase |

|  |  |
| --- | --- |
| 2. | STAF |

|  |  |
| --- | --- |
| 3. | watir |

|  |  |
| --- | --- |
| 4. | xUnit |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | <문제 해설> 인터페이스 구현 검증 도구는 xUnit, STAF, Fitnesse, NTAF, Selenium, watir 등이 있습니다.  Foxbase는 프로그래밍 언어입니다. | |
| **40.** | **다음 중 스택을 이용한 연산과 거리가 먼 것은?** | |

|  |  |
| --- | --- |
| 1. | 선택정렬 |

|  |  |
| --- | --- |
| 2. | 재귀호출 |

|  |  |
| --- | --- |
| 3. | 후위표현(Post-fix expression)의 연산 |

|  |  |
| --- | --- |
| 4. | 깊이우선탐색 |
|  | <문제 해설> 선택정렬은 큐를 이용해서 정렬합니다(1) |

**3과목**

|  |  |
| --- | --- |
| **41.** | **병렬 데이터베이스 환경 중 수평 분할에서 활용되는 분할 기법이 아닌 것은?** |

|  |  |
| --- | --- |
| 1. | 라운드-로빈 |

|  |  |
| --- | --- |
| 2. | 범위 분할 |

|  |  |
| --- | --- |
| 3. | 예측 분할 |

|  |  |
| --- | --- |
| 4. | 해시 분할 |

     입력한 답 : 2      정답 : [3]       정답률 : 33%

|  |  |  |  |
| --- | --- | --- | --- |
|  | | <문제 해설> 아래와 같은 오류 신고가 있었습니다. 여러분들의 많은 의견 부탁 드립니다. 추후 여러분들의 의견을 반영하여 정답을 수정하도록 하겠습니다. 참고로 정답 변경은 오류 신고 5회 이상일 경우 수정합니다.  [오류 신고 내용] 1. 범위 분할(range partitioning) : 분할 키 값이 범위 내에 있는지 여부로 구분한다. 2. 목록 분할(list partitioning) : 값 목록에 파티션을 할당 분할 키 값을 그 목록에 비추어 파티션을 선택한다. 3. 해시 분할(hash partitioning) : 해시 함수의 값에 따라 파티션에 포함할지 여주를 결정한다. 4. 합성 분할(composite partitioning) : 상기 기술을 결합하는 것  라운드로빈은 스케줄링에 나옴 암튼 아님 [해설작성자 : 가치증명하기]  [관리자 입니다. 문제지 사진원본 확인해 봤는데 확정답안은 3번이네요. 다른 문제집 확인 가능한분 계시면 확인 부탁 드립니다. 신고시 출판사명까지 기제 부탁 드립니다.]  [오류신고 반론] 수평분할 - 하나의 테이블의 각 행을 다른 테이블에 분산시키는 것  라운드로빈 분할 - 라운드로빈 분할은 파티션에 행의 고른 분포를 원할 때 사용한다. 그러나, 해시분할과 달리 분할 칼럼을 명시할 필요가 없다. 라운드 로빈 분할로 회전하면서 새로운 행이 파티션에 할당된다. 테이블은 기본키가 반드시 필요하진 않다. [해설작성자 : 오씨] | |
|  | | |  |  | | --- | --- | | **42.** | **시스템 카탈로그에 대한 설명으로 옳지 않은 것은?** |      |  |  | | --- | --- | | 1. | 사용자가 직접 시스템 카탈로그의 내용을 갱신하여 데이터베이스 무결성을 유지한다. |  |  |  | | --- | --- | | 2. | 시스템 자신이 필요로 하는 스키마 및 여러 가지 객체에 관한 정보를 포함하고 있는 시스템 데이터베이스이다. |  |  |  | | --- | --- | | 3. | 시스템 카탈로그에 저장되는 내용을 메타데이터라고도 한다. |  |  |  | | --- | --- | | 4. | 시스템 카탈로그는 DBMS가 스스로 생성하고 유지한다. |      |  |  | | --- | --- | |  | <문제 해설> 시스템 카탈로그는 DBMS가 스스로 생성하고 유지하기 때문에, INSERT, DELETE, UPDATE문으로 시스템 카탈로그를 갱신하는 것은 허용되지 않습니다. (1) | | |
| **43.** | **SQL 문에서 SELECT에 대한 설명으로 옳지 않은 것은?** | |

|  |  |
| --- | --- |
| 1. | FROM 절에는 질의에 의해 검색될 데이터들을 포함하는 테이블명을 기술한다. |

|  |  |
| --- | --- |
| 2. | 검색결과에 중복되는 레코드를 없애기 위해서는 WHERE 절에 'DISTINCT'키워드를 사용한다. |

|  |  |
| --- | --- |
| 3. | HAVING 절은 GROUP BY 절과 함께 사용되며, 그룹에 대한 조건을 지정한다. |

|  |  |
| --- | --- |
| 4. | ORDER BY 절은 특정 속성을 기준으로 정렬하여 검색할 때 사용한다. |

|  |  |
| --- | --- |
|  | <문제 해설> 검색결과에 중복되는 레코드를 없애기 위해서는 SELECT 명령 뒤에 'DISTINCT'키워드를 사용한다. WHERE절 뒤는 원하는 조건.  검색결과에 중복되는 레코드를 없애기 위해서는 SELECT 명령 뒤에 'DISTINCTROW' 키워드를 사용한다. DISTINCT : 중복된 속성이 조화될 경우 그 중 한 개만 검색(SELECT 뒤에 명시된 속성이 중복될 경우 한 개만 검색) |

|  |  |
| --- | --- |
| **49.** | **병행제어 기법의 종류가 아닌 것은?** |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | 로킹 기법 |  |  |

|  |  |
| --- | --- |
| 2. | 시분할 기법 |

|  |  |
| --- | --- |
| 3. | 타임 스탬프 기법 |

|  |  |  |
| --- | --- | --- |
| 4. | | 다중 버전 기법  <문제 해설> 병행제어는 동시에 여러 개의 트랜잭션을 병행 수행할 때, DB의 일관성을 파괴하지 않도록 제어하는 것이다.  시분할 기법은 사용자가 여러 개의 작업이 동시에 수행하는 것처럼 느끼도록 분할하는 운영체제 운영기법이다. |
|  |  | | |
| **53.** | **릴레이션 R의 차수가 4이고 카디널리티가 5이며, 릴레이션 S의 차수가 6이고 카디널리티가 7일 때, 두 개의 릴레이션을 카티션 프로덕트한 결과의 새로운 릴레이 션의 차수와 카디널리티는 얼마인가?** | | | |

|  |  |
| --- | --- |
| 1. | 24, 35 |

|  |  |
| --- | --- |
| 2. | 24, 12 |

|  |  |
| --- | --- |
| 3. | 10, 35 |

|  |  |
| --- | --- |
| 4. | 10, 12 |

|  |  |
| --- | --- |
|  | <문제 해설> 차수는 +를 해주고 카디널리티는 \*를 해주면 해당 결과가 나옵니다. 차수 = 4+6=10 / 카디널리티 = 5\*7=35 |

|  |  |
| --- | --- |
| **55.** | **다음 SQL 문에서 ( ) 안에 들어갈 내용으로 옳은 것은?** |

https://img.comcbt.com/cbt/data/iz/iz20210515/iz20210515m55.gif?1621458069

|  |  |
| --- | --- |
| 1. | SET |

|  |  |
| --- | --- |
| 2. | FROM |

|  |  |
| --- | --- |
| 3. | INTO |

|  |  |
| --- | --- |
| 4. | IN |

|  |  |
| --- | --- |
|  | <문제 해설> update문의 기본형식  update (테이블) set (열=변경할값) where (조건) |

|  |  |
| --- | --- |
| **59.** | **데이터 속성 간의 종속성에 대한 엄밀한 고려없이 잘못 설계된 데이터베이스에서는 데이터 처리 연산 수행 시 각종 이상 현상이 발생할 수 있는데, 이러한 이상 현상이 아닌 것은?** |

|  |  |
| --- | --- |
| 1. | 검색 이상 |

|  |  |
| --- | --- |
| 2. | 삽입 이상 |

|  |  |
| --- | --- |
| 3. | 삭제 이상 |

|  |  |
| --- | --- |
| 4. | 갱신이상 |

|  |  |
| --- | --- |
|  | <문제 해설> 이상 현상은 갱신 이상(Modification Anomaly), 삽입 이상(Insertion Anomaly), 삭제 이상(Deletion Anomaly) 으로 구성된다. 각각을 간략하게 설명하면 다음과 같다.  갱신 이상(Modification Anomaly): 반복된 데이터 중에 일부를 갱신 할 시 데이터의 불일치가 발생한다. 삽입 이상(Insertion Anomaly): 불필요한 정보를 함께 저장하지 않고서는 어떤 정보를 저장하는 것이 불가능하다. 삭제 이상(Deletion Anomaly): 필요한 정보를 함께 삭제하지 않고서는 어떤 정보를 삭제하는 것이 불가능하다. |

**4과목**

**5과목**