

1과목 : 소프트웨어 설계

1. UML 다이어그램 중 순차 다이어그램에 대한 설명으로 틀린 것은?

- ① 객체 간의 동적 상호작용을 시간 개념을 중심으로 모델링하는 것이다.
- ② 주로 시스템의 정적 측면을 모델링하기 위해 사용한다.
- ③ 일반적으로 다이어그램의 수직 방향이 시간의 흐름을 나타낸다.
- ④ 회귀 메시지(Self-Message), 제어블록(Statement block) 등으로 구성된다.

<문제 해설>

- 순차 다이어그램은 행위 다이어그램이므로 동적이고, 순차적인 표현을 위한 다이어그램이다.

[해설작성자 : 두목넷]

2. 메시지 지향 미들웨어(Message-Oriented Middleware, MOM)에 대한 설명으로 틀린 것은?

- ① 느리고 안정적인 응답보다는 즉각적인 응답이 필요한 온라인 업무에 적합하다.
- ② 독립적인 애플리케이션을 하나의 통합된 시스템으로 묶기 위한 역할을 한다.
- ③ 송신측과 수신측의 연결 시 메시지 큐를 활용하는 방법이 있다.
- ④ 상이한 애플리케이션 간 통신을 비동기 방식으로 지원한다.

<문제 해설>

MOM(Message Oriented Middleware)

- 메시지 기반의 비동기형 메시지를 전달하는 방식의 미들웨어이다.
- 온라인 업무보다는 이기종 분산 데이터 시스템의 데이터 동기를 위해 많이 사용한다.

[해설작성자 : 두목넷]

MOM(Message Oriented Middleware)은 즉각적인 응답을 원하는 경우가 아니라 다소 느리고 안정적인 응답을 필요로 하는 경우에 많이 사용됩니다.

[해설작성자 : 세유니에용]

3. 익스트림 프로그래밍에 대한 설명으로 틀린 것은?

- ① 대표적인 구조적 방법론 중 하나이다.
- ② 소규모 개발 조직이 불확실하고 변경이 많은 요구를 접하였을 때 적절한 방법이다.
- ③ 익스트림 프로그래밍을 구동시키는 원리는 상식적인 원리와 경험을 최대한 끌어 올리는 것이다.
- ④ 구체적인 실천 방법을 정의하고 있으며, 개발 문서 보다는 소스코드에 중점을 둔다.

<문제 해설>

1. 구조적 방법론이 아니라 애자일 방법론 중 하나이다.

[해설작성자 : 지나가는 행인]

구조적 방법론은 계획 중심의 접근 방식을 따르는 반면, 익스트

림 프로그래밍은 반복적이고 점진적인 개발을 강조하여 변화에 유연하게 대응할 수 있도록 설계된 방법론으로 차이가 있다..

[해설작성자 : 노작가]

4. 유스케이스(Use Case)의 구성 요소 간의 관계에 포함되지 않는 것은?

- ① 연관
- ② 확장
- ③ 구체화
- ④ 일반화

<문제 해설>

유스케이스(Use Case)의 구성 요소 간의 관계

- 연관관계(Association) : 유스케이스와 액터간의 상호작용이 있음을 표현한다.
- 포함 관계(Include): 하나의 유스케이스가 다른 유스케이스의 실행을 전제로 할 때 형성되는 관계이다.
- 확장 관계(Extend): 확장 기능 유스케이스와 확장 대상 유스케이스 사이에 형성 되는 관계이다.
- 일반화 관계(Generalization) : 유사한 유스케이스 또는 액터를 모아 추상화한 유스케이스 또는 액터와 연결시켜 그룹을 만들어 이해도를 높이기 위한 관계이다.

[해설작성자 : 두목넷]

- 연관 (Association) : 유스케이스와 액터 간의 연결을 나타내며, 액터가 유스케이스를 사용하는 방식을 표시합니다.
- 확장 (Extend) : 하나의 유스케이스가 다른 유스케이스의 기능을 확장할 수 있도록 하는 관계입니다..확장되는 유스케이스는 추가적인 기능을 제공할 수 있지만, 기본 유스케이스 없이는 독립적으로 실행되지 않습니다.
- 구체화 (Realization) : 보통 유스케이스에서는 사용되지 않으며, 보다 일반적으로 소프트웨어에서 요구 사항이나 개념을 구체적인 설계나 구현으로 전환하는 과정을 설명할 때 사용됩니다.
- 일반화 (Generalization) : 한 유스케이스가 다른 유스케이스의 특수 형태인 경우, 이 관계를 사용하여 표현합니다..기본적으로 '상속' 관계와 유사하며, 일반 유스케이스의 기능을 확장하거나 재정의합니다.

[해설작성자 : 노작가]

UML 구성요소

- * 4 사물(구조,행동,그룹,주해)
- * 6 관계(연관vs의존,집합vs포함, 일반화vs실체화)
- * 13 다이어그램 (6개 구조다이어그램, 7개 행위 다이어그램)
 - ** 6 구조 UML 다이어그램 (클래스, 객체, 컴포넌트, 배치, 복합구조, 패키지)
 - ** 7 행위 UML 다이어그램 (유스케이스, 시퀀스(순차), 커뮤니케이션, 스테이트(상태), 액티비티(활동) 상호작용개요, 타이밍)

이 중 클래스, 유스케이스, 시퀀스에 대해 자세히 알아야됨

^^^ 이 문제의 유스케이스 관계에 대한 요약은 상기 내용 보시면 됨 **

유스케이스 다이어그램의 구성요소 : 시스템범위, 액터, 유스케

이스, 관계

유스케이스 다이어그램 관계 종류 : 연관, 포함, 확장, 일반화 4가지 ... UML 구성요소인 관계와 헷가리면 안됨
[해설작성자 : 이루기]

5. 요구사항 분석에서 비기능적(Nonfunctional) 요구에 대한 설명으로 옳은 것은?

- ① 시스템의 처리량(Throughput), 반응 시간 등의 성능 요구나 품질 요구는 비기능적 요구에 해당하지 않는다.
- ② '차량 대여 시스템이 제공하는 모든 화면이 3초 이내에 사용자에게 보여야 한다'는 비기능적 요구이다.
- ③ 시스템 구축과 관련된 안전, 보안에 대한 요구사항들은 비기능적 요구에 해당하지 않는다.
- ④ '금융 시스템은 조회, 인출, 입금, 송금의 기능이 있어야 한다'는 비기능적 요구이다.

<문제 해설>

기능적 요구사항 Vs 비기능적 요구사항

- 기능적 요구사항 : 시스템이 실제로 어떻게 동작하는지에 관점을 둔 요구사항
- 비기능적 요구사항 : 시스템 구축에 대한 성능, 보안, 품질, 안정 등에 대한 성능, 보안, 품질, 안정성등으로 실제 수행에 보조적인 요구사항

오답피하기

- 차량 대여 시스템이 제공하는 모든 화면이 3초 이내에 사용자에게 보여야 한다'는 성능에 해당 하므로 비기능적 요구사항에 해당한다.

[해설작성자 : 두목넷]

6. 정보공학 방법론에서 데이터베이스 설계의 표현으로 사용하는 모델링 언어는?

- ① Package Diagram
- ② State Transition Diagram
- ③ Deployment Diagram
- ④ Entity-Relationship Diagram

<문제 해설>

1,2,3 uml다이어그램 4 정보공학방법론

[해설작성자 : 공부]

7. 미들웨어(Middleware)에 대한 설명으로 틀린 것은?

- ① 여러 운영체제에서 응용 프로그램들 사이에 위치한 소프트웨어이다.
- ② 미들웨어의 서비스 이용을 위해 사용자가 정보 교환 방법 등의 내부 동작을 쉽게 확인할 수 있어야 한다.
- ③ 소프트웨어 컴포넌트를 연결하기 위한 준비된 인프라 구조를 제공한다.
- ④ 여러 컴포넌트를 1대 1, 1대 다, 다대 다 등 여러 가지 형태로 연결이 가능하다.

<문제 해설>

- 클라이언트와 서버 간의 통신을 담당하는 시스템 소프트웨어이다.
- 이기종 하드웨어, 소프트웨어, 네트워크, 프로토콜, PC 환경, 운영체제 환경 등에서 시스템 간의 표준화된 연결을 도와주는

소프트웨어이다.

- 표준화된 인터페이스를 통하여 시스템 간의 데이터 교환에 있어 일관성을 제공한다.
- 운영체제와 애플리케이션 사이에서 중간 매개 역할을 하는 다목적 소프트웨어이다.

오답

- 미들웨어 솔루션은 미들웨어의 서비스 이용을 위해 사용자가 정보 교환 방법 등의 내부 동작을 확인할 필요가 없다.

[해설작성자 : 행복한버거]

사용자가 미들웨어의 정보 교환 방법 등의 내부 동작을 쉽게 확인할 수 있다면, 보안의 위험이 될 수 있으므로 확인할 수 없도록 해야함.

[해설작성자 : 항공대미녀]

미들웨어(Middleware)는 사용자에게 초점이 맞추어져 소프트웨어 시스템에서 여러 구성 요소 간에 정보를 전달하고 상호 작용하는 역할을 하는 소프트웨어로 사용자는 내부 동작을 일일이 확인할 필요가 없다.

[해설작성자 : 선택결과]

사용자가 미들웨어의 내부 동작을 확인하기 위해서는 별도의 응용 소프트웨어가 사용되어야 함.

[해설작성자 : 공부하는중년]

8. UI의 설계 지침으로 틀린 것은?

- ① 이해하기 편하고 쉽게 사용할 수 있는 환경을 제공해야 한다.
- ② 주요 기능을 메인 화면에 노출하여 조작이 쉽도록 하여야 한다.
- ③ 치명적인 오류에 대한 부정적인 사항은 사용자가 인지할 수 없도록 한다.
- ④ 사용자의 직무, 연령, 성별 등 다양한 계층을 수용하여야 한다.

<문제 해설>

사용자 중심 : 사용자가 이해하기 쉽고 편하게 사용할 수 있는 환경을 제공해 실 사용자에게 대한 이해가 바탕이 되어야 함

일관성 : 버튼이나 조작 방법을 사용자가 기억하기 빠르고 쉽게 습득할 수 있도록 설계해야 함

단순성 : 조작 방법은 가장 간단하게 작동되도록 하여 인지적 부담 최소화

결과 예측 가능 : 작동시킬 기능만 보고도 결과 예측이 가능해야 함

가시성 : 주요 기능을 메인 화면에 노출하여 쉬운 조작이 가능해야 함

표준화 : 디자인을 표준화하여 기능 구조의 선행 학습 이후 쉽게 사용 가능해야 함

접근성 : 사용자의 직무, 연령, 성별 등이 고려된 다양한 계층을 수용해야 함

명확성 : 사용자가 개념적으로 쉽게 인지해야함

오류 발생 해결 : 사용자가 오류에 대한 상황을 정확하게 인지

할 수 있어야 함

[해설작성자 : pdh6941]

사용자가 UI를 통해서 프로그램을 사용할 때 발견할 수 있는 치명적 오류에 대한 부정적인 사항들을 인지할 수 있어야 프로그램의 개선이 가능하고 또 오류를 피할 수 있다.

[해설작성자 : 선택결과]

9. 객체지향 개념에서 다형성(Polymorphism)과 관련한 설명으로 틀린 것은?

- ① 다형성은 현재 코드를 변경하지 않고 새로운 클래스를 쉽게 추가할 수 있게 한다.
- ② 다형성이란 여러 가지 형태를 가지고 있다는 의미로, 여러 형태를 받아들일 수 있는 특징을 말한다.
- ③ 메소드 오버라이딩(Overriding)은 상위 클래스에서 정의한 일반 메소드의 구현을 하위 클래스에서 무시하고 재정의할 수 있다.
- ④ 메소드 오버로딩(Overloading)의 경우 매개 변수 타입은 동일하지만 메소드명을 다르게 함으로써 구현, 구분할 수 있다.

<문제 해설>

오버로딩 - 같은 이름의 메소드를 중복하여 정의하는 것

[해설작성자 : 지나가는 행인]

오버로딩 - 한 클래스 내에서 메서드의 이름은 동일하지만 매개 변수의 수나 타입을 다르게 하여 재정의 하는 것

오버라이딩 - 상속관계에서만 발생. 슈퍼클래스의 메서드를 서브클래스에서도 동일한 메서드를 재정의 하는 것

[해설작성자 : 칼방이]

오버로딩 • 함수명 같음. 리턴타입 관계없음. 매개변수타입 & 개수 다름

[해설작성자 : 공겸불자]

10. 소프트웨어 개발 영역을 결정하는 요소 중 다음 사항과 관계있는 것은?

- 소프트웨어에 의해 간접적으로 제어되는 장치와 소프트웨어를 실행하는 하드웨어
- 기존의 소프트웨어와 새로운 소프트웨어를 연결하는 소프트웨어
- 순서적 연산에 의해 소프트웨어를 실행하는 절차

- ① 기능(Function)
- ② 성능(Performance)
- ③ 제약 조건(Constraint)
- ④ 인터페이스(Interface)

<문제 해설>

인터페이스: 서로 다른 두 시스템이나 소프트웨어 등을 서로 이어주는 부분 또는 접속 장치를 의미

[해설작성자 : comcbt.com 이용자]

11. 객체에 대한 설명으로 틀린 것은?

- ① 객체는 상태, 동작, 고유 식별자를 가진 모든 것이라 할 수 있다.
- ② 객체는 공통 속성을 공유하는 클래스들의 집합이다.
- ③ 객체는 필요한 자료 구조와 이에 수행되는 함수들을 가진 하나의 독립된 존재이다.
- ④ 객체의 상태는 속성값에 의해 정의된다.

<문제 해설>

2. 클래스는 공통 속성을 공유하는 객체들의 집합이다.

[해설작성자 : 풋살은재있어]

프로그래밍을 해보면 객체를 생성할 때는 클래스를 정의한 후에 클래스 이름으로 객체를 생성한다..

쉽게 설명하면 클래스는 빵을 굽는 틀이다..그리고 객체는 그 틀에서 구워지는 각각의 빵이라고 생각하면 된다.

[해설작성자 : 선택결과]

12. 속성과 관련된 연산(Operation)을 클래스 안에 묶어서 하나로 취급하는 것을 의미하는 객체지향 개념은?

- ① Inheritance
- ② Class
- ③ Encapsulation
- ④ Association

<문제 해설>

클래스: 하나 이상의 유사한 객체들을 묶는다.

캡슐화: 서로 관련성이 많은 데이터들과 연산들을 묶는다.

[해설작성자 : comcbt.com 이용자]

상속(Inheritance) : 이미 정의된 상위 클래스(부모 클래스)의 모든 속성과 연산을 하위 클래스(자식 클래스)가 물려받는 것

클래스: 하나 이상의 유사한 객체들을 묶는다.

캡슐화(Encapsulation): 서로 관련성이 많은 데이터들과 연산들을 묶는다.

연관관계 (Association) : 2개 이상의 사물이 서로 관련되어 있음을 표현한다

[해설작성자 : comcbt.com 이용자]

[해설작성자 : 박지인]

Encapsulation -> 인캡슐 -> 캡슐 안에 넣다 -> 클래스 안에 속성과 연산들을 묶어서 넣는다.

이렇게 이해하면 쉽다.

[해설작성자 : 선택결과]

13. 애자일(Agile) 프로세스 모델에 대한 설명으로 틀린 것은?

- ① 변화에 대한 대응보다는 자세한 계획을 중심으로 소프트웨어를 개발한다.
- ② 프로세스와 도구 중심이 아닌 개개인과 상호소통을 통해 의견을 수렴한다.
- ③ 협상과 계약보다는 고객과의 협력을 중시한다.
- ④ 문서 중심이 아닌, 실행 가능한 소프트웨어를 중시한다.

<문제 해설>

애자일 개발 프로세스 : 개발에 대한 개념적 방법론으로, 개발 프로젝트 기간을 짧은 주기로 나눠 반복적인 개발을 하는 것이 특징

1. 고객관점의 효율적이고 민첩한 변화 대응을 중시

[애자일 선언문]

" 프로세스, 도구 보다는 (사람과 상호작용)을
광범위한 문서 보다는 (실제 작동하는 제품)을
계약 협상보다는 (고객 협력)을
계획을 따르기 보다는 (변화 대응)을"

[해설작성자 : 서처]

14. 명백한 역할을 가지고 독립적으로 존재할 수 있는 시스템의
부분으로 넓은 의미에서는 재사용되는 모든 단위라고 볼 수
있으며, 인터페이스를 통해서만 접근할 수 있는 것은?

- ① Model ② Sheet
③ Component ④ Cell

<문제 해설>

컴포넌트

- 프로그래밍에 있어 재사용이 가능한 각각의 독립된 모듈
- 특정 기능 수행을 위해 독립적으로 분리

[해설작성자 : 보호주의]

15. GoF(Gang of Four) 디자인 패턴을 생성, 구조, 행동 패턴
의 세 그룹으로 분류할 때, 구조 패턴이 아닌 것은?

- ① Adapter 패턴 ② Bridge 패턴
③ Builder 패턴 ④ Proxy 패턴

<문제 해설>

[구조패턴]

- 어댑터
- 브릿지
- 컴포지트
- 데코레이터
- 퍼사드
- 플라이웨이트
- 프록시

[해설작성자 : 아기재혁]

생성패턴 : 객체를 생성하는 것에 대한 패턴

- 추상 팩토리 패턴(abstract factory), 빌더 패턴(builder), 팩토리 메소드 패턴(factory method), 프로토타입 패턴(prototype), 싱글톤 패턴(singleton)

구조패턴 : 구조를 통해 확장성을 꾀하는 패턴

- 어댑터 패턴(adapter), 브릿지 패턴(bridge), 컴포지트 패턴(composite), 데코레이터 패턴(decorator), 퍼사드 패턴(facade), 플라이웨이트 패턴(flyweight), 프록시 패턴(proxy)

행위 패턴 : 행위의 변경, 수정 등을 위한 패턴

- 역할 사슬 패턴(책임 연쇄, chain of responsibility), 커맨드 패턴(command), 인터프리터 패턴(interpreter), 이터레이터 패턴(iterator), 미디에이터 패턴(중재자, mediator), 메멘토 패턴(memento), 옵저버 패턴(observer), 상태 패턴(state), 전략 패턴(strategy), 템플릿 메소드 패턴(template method), 비지터 패턴(visitor)

[해설작성자 : xoxo]

16. UI와 관련된 기본 개념 중 하나로, 시스템의 상태와 사용자
의 지시에 대한 효과를 보여주어 사용자가 명령에 대한 진행
상황과 표시된 내용을 해석할 수 있도록 도와주는 것은?

- ① Feedback ② Posture
③ Module ④ Hash

<문제 해설>

피드백(feedback): 처리된 결과를 측정하고 목표에 도달되었는
가를 검사하며 불충분할 경우 다시 입력하는 요소로 사용.

[해설작성자 : 내일시험임..]

17. UI의 종류로 멀티 터치(Multi-touch), 동작 인식(Gesture
Recognition) 등 사용자의 자연스러운 움직임을 인식하여
서로 주고받는 정보를 제공하는 사용자 인터페이스를 의미
하는 것은?

- ① GUI(Graphical User Interface)
② OUI(Organic User Interface)
③ NUI(Natural User Interface)
④ CLI(Command Line Interface)

<문제 해설>

사용자 인터페이스(UI, User Interface)의 구분

- CLI(Command Line Interface) : 텍스트 형태 인터페이스
- GUI(Graphical User Interface) : 마우스로 선택하여 작업하는 그래픽 환경 인터페이스
- NUI(Natural User Interface) : 사용자의 말이나 행동으로 기기 조작하는 인터페이스
- VUI(Voice User Interface) : 사람의 음성으로 기기 조작하는 인터페이스
- OUI(Organic User Interface) : 모든 사물과 사용자 간의 상호작용을 위한 인터페이스

[해설작성자 : 방진]

NUI : 사용자의 자연스러운 행동(터치, 제스처, 음성 등)을 통해
시스템과 상호작용하는 인터페이스

ex)

- 멀티터치 스마트폰(두 손가락 이상의 터치가 가능한 스마트폰)
- kinect(마이크로소프트가 개발한 동작 인식_모션 캡처_장치)
- Siri(VUI도 맞음)

[해설작성자 : 내가공부중]

18. 소프트웨어 모델링과 관련한 설명으로 틀린 것은?

- ① 모델링 작업의 결과물은 다른 모델링 작업에 영향을 줄 수 없다.
② 구조적 방법론에서는 DFD(Data Flow Diagram), DD(Data Dictionary) 등을 사용하여 요구 사항의 결과를 표현한다.
③ 객체지향 방법론에서는 UML 표기법을 사용한다.
④ 소프트웨어 모델을 사용할 경우 개발될 소프트웨어에 대한 이해도 및 이해 당사자 간의 의사소통 향상에 도움이 된다.

<문제 해설>

모델링 작업의 결과물은 다른 모델링 작업에 영향을 줄 수 있음.

ex) 유스케이스 명세로부터 필요한 요소들을 가져와 도메인 모델로 정리할 수 있음.

[해설작성자 : 이예림]

19. 유스케이스 다이어그램(Use Case Diagram)에 관련된 내용으로 틀린 것은?

- ① 시스템과 상호작용하는 외부시스템은 액터로 파악해서는 안된다.
- ② 유스케이스는 사용자 측면에서의 요구사항으로, 사용자가 원하는 목표를 달성하기 위해 수행할 내용을 기술한다.
- ③ 시스템 액터는 다른 프로젝트에서 이미 개발되어 사용되고 있으며, 본 시스템과 데이터를 주고받는 등 서로 연동되는 시스템을 말한다.
- ④ 액터가 인식할 수 없는 시스템 내부의 기능을 하나의 유스케이스로 파악해서는 안된다.

<문제 해설>

액터: 시스템과 상호작용하는 모든 것(사람, 기계, 시스템 등)

[해설작성자 : 칼방이]

액터: 시스템과 상호작용하는 모든 외부요소로, 사람이나 외부시스템을 의미함

[해설작성자 : comcbt.com 이용자]

20. 소프트웨어 아키텍처 모델 중 MVC(Model-View-Controller)와 관련된 설명으로 틀린 것은?

- ① MVC 모델은 사용자 인터페이스를 담당하는 계층의 응집도를 높일 수 있고, 여러 개의 다른 UI를 만들어 그 사이에 결합도를 낮출 수 있다.
- ② 모델(Model)은 뷰(View)와 제어(Controller) 사이에서 전달자 역할을 하며, 뷰마다 모델 서브시스템이 각각 하나씩 연결된다.
- ③ 뷰(View)는 모델(Model)에 있는 데이터를 사용자 인터페이스에 보이는 역할을 담당한다.
- ④ 제어(Controller)는 모델(Model)에 명령을 보냄으로써 모델의 상태를 변경할 수 있다.

<문제 해설>

한 개의 모델에 대해 여러 개의 뷰를 만들 수 있음

[해설작성자 : 방진]

모델은 서브 시스템의 핵심 기능과 데이터를 보관하며 MVC 패턴에서는 여러 개의 뷰를 만들 수 있으므로 한 개의 모델에 대해 여러 개의 뷰를 필요로 하는 대화형 어플리케이션에 적합하다.

[해설작성자 : 여름]

전달자 역할은 모델(Model)이 아닌 제어(Controller)이다.

[해설작성자 : 좋은데이]

제어(controller)는 뷰(view)와 모델(model)사이에서 전달자 역할을 수행한다.

[해설작성자 : comcbt.com 이용자]

2과목 : 소프트웨어 개발

21. 통합 테스트(Integration Test)와 관련한 설명으로 틀린 것은?

- ① 시스템을 구성하는 모듈의 인터페이스와 결합을 테스트하는 것이다.
- ② 하향식 통합 테스트의 경우 넓이 우선(Breadth First) 방식으로 테스트를 할 모듈을 선택할 수 있다.
- ③ 상향식 통합 테스트의 경우 시스템 구조도의 최상위에 있는 모듈을 먼저 구현하고 테스트한다.
- ④ 모듈 간의 인터페이스와 시스템의 동작이 정상적으로 잘 되고 있는지를 빨리 파악하고자 할 때 상향식 보다는 하향식 통합 테스트를 사용하는 것이 좋다.

<문제 해설>

상향식 통합 테스트 (Bottom Up Integration Test)

- 프로그램의 하위 모듈에서 상위 모듈 방향으로 통합
- 하나의 주요 제어 모듈과 관련된 종속 모듈의 그룹인 클러스터(Cluster)필요

하향식 통합 테스트 (Top Down Integration Test)

- 상위 모듈에서 하위 모듈 방향으로 통합
- 깊이 우선 통합법, 넓이 우선 통합법 사용
- 초기부터 사용자에게 시스템 구조를 보여줌

[해설작성자 : 전자공학도]

상향식 테스트 : 드라이버(Driver) 이용(하위 모듈 호출, 매개변수 전달, 테스트 후 결과 도출)

하향식 테스트 : 스텝(Stub) 이용(모듈 기능만 수행)

[해설작성자 : 공부하자]

22. 다음과 같이 레코드가 구성되어 있을 때, 이진 검색 방법으로 14를 찾을 경우 비교되는 횟수는?

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

- ① 2
- ② 3
- ③ 4
- ④ 5

<문제 해설>

이진 탐색 동작 방식

1. 배열의 중간 값을 찾음
2. 중간 값과 검색 값을 비교
- 2-1. 중간 값이 검색 값과 같으면 끝.
- 2-2. 중간 값보다 검색 값이 크다면 오른쪽 구간을 탐색
- 2-3. 중간 값보다 검색 값이 작다면 왼쪽 구간을 탐색
3. 값을 찾거나 간격이 비었을 때까지 반복

[해설작성자 : 전자공학도]

다음은 이진 탐색 관련 내용을 정리한 문장입니다:

"low 값이 0이고, 배열이 1부터 15까지 순차적으로 증가한다면 'high = 14'는 이미 값 15를 가리키는 것 아니냐"는 의문이 있을 수 있습니다..그러나 이진 탐색은 값이 아닌 인덱스를 기준으로 동작하기 때문에, 'high = 14'는 단순히 인덱스 14를 의미하며, 해당 값이 이미 선택되었다는 뜻은 아닙니다.

이진 탐색에서 $\text{mid} = (\text{low} + \text{high}) // 2$ 는 배열의 중간값이 아니라 **중간 인덱스**를 의미하며, 실제 비교는 `arr[mid]`와 `target` 값 사이에서 이루어집니다.

이 문제가 1부터 15까지 등차 수열 형태의 '친절한' 배열이기 때문에 중간 인덱스가 가리키는 값도 실제 중간값처럼 보일 수 있지만, 등차 수열이 아닌 경우에는 같은 방식으로 계산된 `mid`가 가리키는 값은 논리적 '중간값'과 다를 수 있습니다.

따라서, 이진 탐색은 "정렬된 배열"이라는 전제가 성립할 때 인덱스를 기준으로 범위를 좁혀가는 알고리즘이며, 배열이 등차 수열인지 여부는 탐색의 정확성과는 무관합니다.

[해설작성자 : 엽이]

23. 소프트웨어 공학에서 워크스루(Walkthrough)에 대한 설명으로 틀린 것은?

- ① 사용사례를 확장하여 명세하거나 설계 다이어그램, 원시 코드, 테스트 케이스 등에 적용할 수 있다.
- ② 복잡한 알고리즘 또는 반복, 실시간 동작, 병행 처리와 같은 기능이나 동작을 이해하려고 할 때 유용하다.
- ③ 인스펙션(Inspection)과 동일한 의미를 가진다.
- ④ 단순한 테스트 케이스를 이용하여 프로덕트를 수작업으로 수행해 보는 것이다.

<문제 해설>

워크스루: 요구사항 명세서 작성자를 포함하여 사전 검토한 후에 짧은 검토 회의를 통해 결함을 발견

인스펙션: 요구사항 명세서 작성자를 제외한 다른 검토 전문가들이 요구사항 명세서를 확인하면서 결함을 발견.

[해설작성자 : 칼방이]

24. 소프트웨어의 개발과정에서 소프트웨어의 변경사항을 관리하기 위해 개발된 일련의 활동을 뜻하는 것은?

- ① 복호화 ② 형상관리
- ③ 저작권 ④ 크랙

<문제 해설>

복호화 - 디코딩, 즉 부호화된 데이터를 부호화 되기 전 형태로 바꾸어, 사람이 읽을 수 있는 형태로 되돌려 놓는것

저작권 - 창작물을 만든이가 자기 저작물에 대해 가지는 법적권리

크랙 - 소프트웨어를 수정하여 소프트웨어를 크랙하는 사람이 원하지 않는 기능들, 보통은 수정 방식(복사 보호, 소프트웨어 조작 보호)을 비활성화하거나 제거하는 일

[해설작성자 : comcbt.com 이용자]

25. 테스트 케이스와 관련한 설명으로 틀린 것은?

- ① 테스트의 목표 및 테스트 방법을 결정하기 전에 테스트 케이스를 작성해야 한다.
- ② 프로그램에 결함이 있더라도 입력에 대해 정상적인 결과를 낼 수 있기 때문에 결함을 검사할 수 있는 테스트 케이스를 찾는 것이 중요하다.
- ③ 개발된 서비스가 정의된 요구 사항을 준수하는지 확인하기 위한 입력 값과 실행 조건, 예상 결과의 집합으로 볼 수 있다.
- ④ 테스트 케이스 실행이 통과되었는지 실패하였는지 판단하기 위한 기준을 테스트 오라클(Test Oracle)이라고 한다.

<문제 해설>

테스트케이스(Test Case)

- 구현된 소프트웨어가 사용자의 요구사항을 정확하게 준수했는지를 확인하기 위해 설계된 입력값, 실행 조건, 기대 결과 등으로 구성된 테스트 항목에 대한 명세서를 의미한다.

- 테스트의 목표 및 테스트 방법을 결정하고 테스트 케이스를 작성해야 한다.

[해설작성자 : 두목넷]

26. 객체지향 개념을 활용한 소프트웨어 구현과 관련한 설명 중 틀린 것은?

- ① 객체(Object)란 필요한 자료 구조와 수행되는 함수들을 가진 하나의 독립된 존재이다.
- ② JAVA에서 정보은닉(Information Hiding)을 표기할 때 `private`의 의미는 '공개'이다.
- ③ 상속(Inheritance)은 개별 클래스를 상속 관계로 묶음으로써 클래스 간의 체계화된 전체 구조를 파악하기 쉽다는 장점이 있다.
- ④ 같은 클래스에 속하는 개개의 객체이자 하나의 클래스에서 생성된 객체를 인스턴스(Instance)라고 한다.

<문제 해설>

- JAVA에서 정보은닉(InformationHiding)을 표기할 때 `private`의 의미는 외부에서 클래스 내부 정보에 접근하지 못하도록 하는 '접근금지'이다.

[해설작성자 : comcbt.com 이용자]

27. DRM(Digital Rights Management)과 관련한 설명으로 틀린 것은?

- ① 디지털 콘텐츠와 디바이스의 사용을 제한하기 위해 하드웨어 제조업자, 저작권자, 출판업자 등이 사용할 수 있는 접근 제어 기술을 의미한다.
- ② 디지털 미디어의 생명 주기 동안 발생하는 사용 권한 관리, 과금, 유통 단계를 관리하는 기술로도 볼 수 있다.
- ③ 클리어링 하우스(Clearing House)는 사용자에게 콘텐츠 라이선스를 발급하고 권한을 부여해주는 시스템을 말한다.
- ④ 원본을 안전하게 유통하기 위한 전자적 보안은 고려하지 않기 때문에 불법 유통과 복제의 방지는 불가능하다.

<문제 해설>

DRM(Digital Right Management)

- 디지털 콘텐츠의 지적재산권보호, 관리 기능 및 안전한 유통

과 배포를 보장하는 솔루션이다.

- 디지털 콘텐츠의 지적재산권을 보호하는 권한통제기술, 사용 권한제어 기술, 패키징 기술, 라이선스 관리를 포함한 유통체계이다.

- 디지털 콘텐츠와 디바이스의 사용을 제한하기 위해 하드웨어 제조업자, 저작권자, 출판업자 등이 사용할 수 있는 접근 제어 기술을 의미한다.

[해설작성자 : comcbt.com 이용자]

28. 위험 모니터링의 의미로 옳은 것은?

- ① 위험을 이해하는 것
- ② 첫 번째 조치로 위험을 피할 수 있도록 하는 것
- ③ 위험 발생 후 즉시 조치하는 것
- ④ 위험 요소 징후들에 대하여 계속적으로 인지하는 것

<문제 해설>

위험 모니터링(monitring)의 의미

-> 위험 요소 징후들에 대하여 계속적으로 인지하는 것

[해설작성자 : 내일정치기합격기원]

29. 동시에 소스를 수정하는 것을 방지하며 다른 방향으로 진행된 개발 결과를 합치거나 변경 내용을 추적할 수 있는 소프트웨어 버전 관리 도구는?

- ① RCS(Revision Control System)
- ② RTS(Reliable Transfer Service)
- ③ RPC(Remote Procedure Call)
- ④ RVS(Relative Version System)

<문제 해설>

RCS(Revision Control System)

- CVS와의 차이점은 소스 파일의 수정을 한 사람만으로 제한한다.

- 다수의 사용자가 동시에 파일 수정을 할 수 없도록 파일 잠금 방식으로 버전을 관리하는 도구이다.

- 다른 방향으로 진행된 개발 결과를 합치거나 변경 내용을 추적할 수 있다.

[해설작성자 : 두목넷]

RPC(Remote Procedure call)

- 별도의 원격 제어를 위한 코딩 없이 다른 주소 공간에서 리모트의 함수나 프로시저를 실행 할 수 있게 해주는 프로세스간 통신

[해설작성자 : comcbt.com 이용자]

RTS : 데이터 통신에서 신뢰성이 보장된 전송 서비스를 제공하는 프로토콜

RPC : 네트워크상에서 원격지의 다른 프로세스나 컴퓨터에서 실행중인 함수나 프로시저를 호출하는 프로토콜

RVS : 파일의 상대적인 버전을 관리하는 도구로 각 파일 버전의 차이를 저장해 파일 복원과 변경 이력을 추적

[해설작성자 : 정처기 공부중]

30. 화이트박스 테스트와 관련한 설명으로 틀린 것은?

- ① 화이트박스 테스트의 이해를 위해 논리흐름도

(Logic-Flow Diagram)를 이용할 수 있다.

- ② 테스트 데이터를 이용해 실제 프로그램을 실행함으로써 오류를 찾는 동적 테스트(Dynamic Test)에 해당한다.
- ③ 프로그램의 구조를 고려하지 않기 때문에 테스트 케이스는 프로그램 또는 모듈의 요구나 명세를 기초로 결정한다.
- ④ 테스트 데이터를 선택하기 위하여 검증기준(Test Coverage)을 정한다.

<문제 해설>

3. 프로그램 구조를 고려하지 않는 것은 == 블랙박스 테스트

[해설작성자 : comcbt.com 이용자]

화이트박스 테스트는 정적 분석뿐만 아니라 동적 분석에도 사용될 수 있습니다..정적 분석에서는 코드를 실행하지 않고 검토하여 문제점을 찾지만, 동적 분석에서는 실제로 코드를 실행시켜 테스트를 진행합니다..화이트박스 테스트의 경우, 프로그램의 내부 구조를 알고 있는 상태에서 테스트를 진행하기 때문에, 로직 흐름도를 이용하여 로직 검증을 할 수 있고, 동적 테스트를 통해 실제 실행 중인 프로그램의 동작을 검증할 수도 있습니다.

제시된 문제에서 언급된 "프로그램의 구조를 고려하지 않고 때때로 테스트 케이스는 프로그램 코드 또는 모듈의 우연한 면을 기준으로 결정한다"는 주장은 화이트박스 테스트에 부합하지 않습니다..화이트박스 테스트는 프로그램의 내부 구조에 기반하여 체계적으로 테스트 케이스를 설계하기 때문에 우연한 면을 기준으로 테스트 케이스를 설계하지 않습니다.

[해설작성자 : comcbt 사용자]

31. 알고리즘과 관련한 설명으로 틀린 것은?

- ① 주어진 작업을 수행하는 컴퓨터 명령어를 순서대로 나열한 것으로 볼 수 있다.
- ② 검색(Searching)은 정렬이 되지 않은 데이터 혹은 정렬이 된 데이터 중에서 키값에 해당되는 데이터를 찾는 알고리즘이다.
- ③ 정렬(Sorting)은 흩어져있는 데이터를 키값을 이용하여 순서대로 열거하는 알고리즘이다.
- ④ 선형 검색은 검색을 수행하기 전에 반드시 데이터의 집합이 정렬되어 있어야 한다.

<문제 해설>

선형 검색 - 처음부터 끝까지 하나씩 순서대로 비교하며 원하는 값을 찾아내는 검색

[해설작성자 : 전자공학도]

이진검색 - 검색을 수행하기 전에 반드시 데이터의 집합이 정렬되어 있어야 한다.

[해설작성자 : 문은식]

검색 알고리즘

- 선형 검색(Linear Search) : 데이터가 모인 집합(배열, 리스트 등)의 처음부터 끝까지 하나씩 순서대로 비교하며 검색(순차검색)

- 이진 검색(Binary Search) : 데이터가 오름차순 혹은 내림차

순으로 정렬된 배열에서 검색하는 알고리즘(이분 검색)
 [해설작성자 : 공부하자]

32. 버블 정렬을 이용하여 다음 자료를 오름차순으로 정렬할 경우 PASS 1의 결과는?

9, 6, 7, 3, 5

- ① 6, 9, 7, 3, 5 ② 3, 9, 6, 7, 5
 ③ 3, 6, 7, 9, 5 ④ 6, 7, 3, 5, 9

<문제 해설>

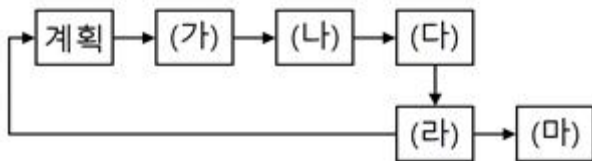
버블 정렬은 첫위치부터 시작해서 오른쪽값과 비교

PASS 1 첫 버블 정렬이 끝난상태

96 비교 -> 69735 -> 97비교 -> 67935 -> 93비교 ->
 67395 -> 95비교 -> 67359

[해설작성자 : 합격기원]

33. 다음은 인스펙션(Inspection) 과정을 표현한 것이다. (가)~(마)에 들어갈 말을 보기에서 찾아 바르게 연결한 것은?



- | | |
|-----------|---------|
| ㉠ 준비 | ㉡ 사전 교육 |
| ㉢ 인스펙션 회의 | ㉣ 수정 |
| ㉤ 후속조치 | |

- ① (가) - ㉡, (나) - ㉢
 ② (나) - ㉠, (다) - ㉣
 ③ (다) - ㉢, (라) - ㉤
 ④ (라) - ㉤, (마) - ㉣

<문제 해설>

가 - 사전교육

나 - 준비

다 - 인스펙션 회의

라 - 수정

마 - 후속조치

[해설작성자 : ㅎㅎ]

34. 소프트웨어를 보다 쉽게 이해할 수 있고 적은 비용으로 수정할 수 있도록 겉으로 보이는 동작의 변화 없이 내부구조를 변경하는 것은?

- ① Refactoring ② Architecting
 ③ Specification ④ Renewal

<문제 해설>

1. Refactoring - 코드의 구조 재조정

2. Architecting - 설계관련

3. Specification - 명세서

4. Renewal - 유지보수

* 리팩토링과 유지보수는 다른 개념이다.

[해설작성자 : 익명]

리팩토링: 소프트웨어를 보다 쉽게 이해할 수 있고 적은 비용으로 수정할 수 있도록 겉으로 보이는 동작의 변화 없이 내부구조를 변경하는 것으로, 코드 스멜(code smell)을 고치고 다듬는 과정이다.

[해설작성자 : 재발합격]

35. 단위 테스트(Unit Test)와 관련한 설명으로 틀린 것은?

- ① 구현 단계에서 각 모듈의 개발을 완료한 후 개발자가 명세서의 내용대로 정확히 구현되었는지 테스트한다.
 ② 모듈 내부의 구조를 구체적으로 볼 수 있는 구조적 테스트를 주로 시행한다.
 ③ 필요 데이터를 인자를 통해 넘겨주고, 테스트 완료 후 그 결과값을 받는 역할을 하는 가상의 모듈을 테스트 스텝(Stub)이라고 한다.
 ④ 테스트할 모듈을 호출하는 모듈도 있고, 테스트할 모듈이 호출하는 모듈도 있다.

<문제 해설>

필요 데이터를 인자를 통해 넘겨주고, 테스트 완료 후 그 결과값을 받는 역할을 하는 가상의 모듈을 테스트 드라이버(test driver)라고 한다..반대로 테스트 스텝(test stub)은 인자를 통해 받은 값을 가지고 수행한 후 그 결과를 테스트할 모듈에 넘겨주는 역할을 한다.

[해설작성자 : 물집메고맘이있다]

36. IDE(Integrated Development Environment) 도구의 각 기능에 대한 설명으로 틀린 것은?

- ① Coding - 프로그래밍 언어를 가지고 컴퓨터 프로그램을 작성할 수 있는 환경을 제공
 ② Compile - 저급언어의 프로그램을 고급언어 프로그램으로 변환하는 기능
 ③ Debugging - 프로그램에서 발견되는 버그를 찾아 수정할 수 있는 기능
 ④ Deployment - 소프트웨어를 최종 사용자에게 전달하기 위한 기능

<문제 해설>

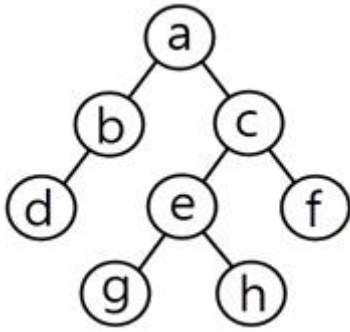
Compile - 주어진 언어로 작성된 컴퓨터 프로그램을 다른 언어의 동등한 프로그램으로 변환하는 기능

[해설작성자 : comcbt.com 이용자]

컴파일은 고급언어(사람이 인식하는 언어)에서 저급언어(기계어로 바꾸는 변환하는 기능

[해설작성자 : 나진관예혁인대]

37. 아래 Tree 구조에 대하여 후위 순회(Postorder) 한 결과는?



- ① a → b → d → c → e → g → h → f
- ② d → b → g → h → e → f → c → a
- ③ d → b → a → g → e → h → c → f
- ④ a → b → d → g → e → h → c → f

<문제 해설>

전위 순회 (PreOrder) => Root, Left, Right 순서
 중위 순회 (InOrder) => Left, Root, Right 순서
 후위 순회 (PostOrder) => Left, Right, Root 순서
 [해설작성자 : Eclatz]

pre_order (전위순회) : 뿌리 먼저 방문
 in_order (중위순회) : 왼쪽 하위 노드 방문 후 뿌리 방문
 post_order (후위순회) : 하위 노드 모두 방문 후 뿌리 방문
 [해설작성자 : 트리맨]

38. 인터페이스 구현 시 사용하는 기술로 속성-값 쌍 (Attribute-Value Pairs)으로 이루어진 데이터 오브젝트를 전달하기 위해 사용하는 개방형 표준 포맷은?

- ① JSON
- ② HTML
- ③ AVPN
- ④ DOF

<문제 해설>

JSON(JavaScript Object Notation)
 - 속성-값 쌍(Attribute-Value Pairs)으로 이루어진 데이터 오브젝트를 전달하기 위해 사용하는 개방형 표준 포맷이다.
 - AJAX(Asynchronous Javascript and XML)에서 많이 사용되고 XML을 대체하는 주요 데이터 포맷이다.
 - 언어 독립형 데이터 포맷으로 다양한 프로그래밍 언어에서 사용되고 있다.
 [해설작성자 : comcbt.com 이용자]

39. 순서가 있는 리스트에서 데이터의 삽입(Push), 삭제(Pop)가 한 쪽 끝에서 일어나며 LIFO(Last-In-First-Out)의 특징을 가지는 자료구조는?

- ① Tree
- ② Graph
- ③ Stack
- ④ Queue

<문제 해설>

1. Stack = LIFO, 한 쪽 끝에서 일어남
 2. Queue = FIFO, 양방향에서 일어남
 3. Tree & Graph 는 비선형구조
 [해설작성자 : IJH]

40. 다음 중 단위 테스트 도구로 사용될 수 없는 것은?

- ① CppUnit
- ② JUnit

③ HttpUnit

④ IgpUnit

<문제 해설>

CppUnit - C++ 프로그래밍 언어용 단위 테스트 도구
 JUnit - 자바 프로그래밍 언어용 단위 테스트 도구
 [해설작성자 : comcbt.com 이용자]

HttpUnit - 웹 브라우저 없이 웹 사이트 테스트를 수행하는 데 사용되는 오픈 소스 소프트웨어 테스트 프레임워크 입니다.
 [해설작성자 : 상동닭집]

IgpUnit은 존재하지 않는 유닛입니다.

[해설작성자 : 왕재수]

3과목 : 데이터베이스 구축

41. 다음 조건을 모두 만족하는 정규형은?

- 테이블 R에 속한 모든 도메인이 원자값만으로 구성되어 있다.
- 테이블 R에서 키가 아닌 모든 필드가 키에 대해 함수적으로 종속되며, 키의 부분 집합이 결정자가 되는 부분 종속이 존재하지 않는다.
- 테이블 R에 존재하는 모든 함수적 종속에서 결정자가 후보키이다.

- ① BCNF
- ② 제1정규형
- ③ 제2정규형
- ④ 제3정규형

<문제 해설>

정규화 과정 [도부이결다조]

- 1NF : 모든 *도*메인이 원자 값으로만 구성
- 2NF : 기본키가 아닌 속성이 기본키에 대한 완전 함수적 종속을 만족, *부*분적 함수 종속을 제거한 정규형
- 3NF : 기본키가 아닌 모든 속성이 기본키에 대해 *이*행적 함수 종속 관계를 만족하지 않는 정규형
- BCNF : 모든 *결*정자가 후보키인 정규형, 종속성 보존X
- 4NF : *다*치 종속이 성립하는 경우, R의 모든 속성이 A에 함수적 종속 관계를 만족
- 5NF : 모든 *조*인 종속이 R의 후보키를 통해서만 성립되는 정규형

결정자라는 키워드가 있기 때문에 BCNF

[해설작성자 : 열라면]

42. 데이터베이스의 트랜잭션 성질들 중에서 다음 설명에 해당하는 것은?

- 트랜잭션의 모든 연산들이 정상적으로 수행 완료되거나 아니면 전혀 어떠한 연산도 수행되지 않은 원래 상태가 되도록 해야 한다.

- ① Atomicity
- ② Consistency
- ③ Isolation
- ④ Durability

<문제 해설>

트랜잭션의 특성

- 원자성(Atomicity) : 완전하게 수행 완료되지 않으면 전혀 수행되지 않아야 함
 - 일관성(Consistency) : 시스템의 고정 요소는 트랜잭션 수행 전후에 같아야 함
 - 격리성(Isolation, 고립성) : 트랜잭션 실행 시 다른 트랜잭션의 간섭을 받지 않아야 함
 - 영속성(Durability, 지속성) : 트랜잭션의 완료 결과가 데이터베이스에 영구히 기억됨
- [해설작성자 : 두목넷]

43. 분산 데이터베이스 시스템과 관련한 설명으로 틀린 것은?

- ① 물리적으로 분산된 데이터베이스 시스템을 논리적으로 하나의 데이터베이스 시스템처럼 사용할 수 있도록 한 것이다.
- ② 물리적으로 분산되어 지역별로 필요한 데이터를 처리할 수 있는 지역 컴퓨터(Local Computer)를 분산 처리기(Distributed Processor)라고 한다.
- ③ 분산 데이터베이스 시스템을 위한 통신 네트워크 구조가 데이터 통신에 영향을 주므로 효율적으로 설계해야 한다.
- ④ 데이터베이스가 분산되어 있음을 사용자가 인식할 수 있도록 분산 투명성(Distribution Transparency)을 배제해야 한다.

<문제 해설>

분산 데이터베이스의 목표

- 위치 투명성(Location Transparency) : 하드웨어와 소프트웨어의 물리적 위치를 사용자가 알 필요가 없다.
- 중복(복제) 투명성(Replication Transparency) : 사용자에게 통지할 필요없이 시스템 안에 파일들과 자원들의 부가적인 복사를 자유롭게 할 수 있다.
- 병행 투명성(Concurrency Transparency) : 다중 사용자들이 자원들을 자동으로 공유할 수 있다.
- 장애 투명성(Failure Transparency) : 사용자들은 어느 위치의 시스템에 장애가 발생했는지 알 필요가 없다.

[해설작성자 : 두목넷 정보처리]

분산 데이터베이스는 '분산 투명성'을 고려해야 한다..

분산 데이터베이스 목표는 투명성이며 위치 투명성, 중복 투명성, 병행 투명성, 장애 투명성 4가지의 세부 목표를 가진다..

[해설작성자 : 처기찌래기]

44. 다음 테이블을 보고 강남지점의 판매량이 많은 제품부터 출력되도록 할 때 다음 중 가장 적절한 SQL 구문은? (단, 출력은 제품명과 판매량이 출력되도록 한다.)

<푸드> 테이블

지점명	제품명	판매량
강남지점	비빔밥	500
강북지점	도시락	300
강남지점	도시락	200
강남지점	미역국	550
수원지점	비빔밥	600
인천지점	비빔밥	800
강남지점	잡채밥	250

- ① SELECT 제품명, 판매량 FROM 푸드 ORDER BY 판매량 ASC;
- ② SELECT 제품명, 판매량 FROM 푸드 ORDER BY 판매량 DESC;
- ③ SELECT 제품명, 판매량 FROM 푸드 WHERE 지점명 = '강남지점' ORDER BY 판매량 ASC;
- ④ SELECT 제품명, 판매량 FROM 푸드 WHERE 지점명 = '강남지점' ORDER BY 판매량 DESC;

<문제 해설>

- 판매량이 많은 제품부터 출력되도록 하려면 내림차순 정렬을 적용해야 한다..SQL에서 정렬은 ORDER BY를 사용하며 내림차순은 DESC를 사용한다..오름차순의 경우 생략이나 ASC를 사용한다.

[해설작성자 : 두목넷]

DESC : descending 내림차순

ASC : ascending 오름차순

[해설작성자 : 주거니아빠]

45. 데이터베이스의 인덱스와 관련한 설명으로 틀린 것은?

- ① 문헌의 색인, 사전과 같이 데이터를 쉽고 빠르게 찾을 수 있도록 만든 데이터 구조이다.
- ② 테이블에 붙여진 색인으로 데이터 검색 시 처리 속도 향상에 도움이 된다.
- ③ 인덱스의 추가, 삭제 명령어는 각각 ADD, DELETE이다.
- ④ 대부분의 데이터베이스에서 테이블을 삭제하면 인덱스도 같이 삭제된다.

<문제 해설>

인덱스(Index)

- 데이터베이스 성능에 많은 영향을 주는 DBMS의 구성 요소로 테이블과 클러스터에 연관되어 독립적인 저장 공간을 보유하며, 데이터베이스에 저장된 자료를 더욱 빠르게 조회하기 위하여 별도로 구성한 순서 데이터를 말한다.
- 대부분의 데이터베이스에서 테이블을 삭제하면 인덱스도 같이 삭제된다.

오답피하기

- 인덱스 생성 : CREATE

- 인덱스 삭제 : DROP
[해설작성자 : 두목넷]

46. 물리적 데이터베이스 구조의 기본 데이터 단위인 저장 레코드의 양식을 설계할 때 고려 사항이 아닌 것은?

- ① 데이터 타입
- ② 데이터 값의 분포
- ③ 트랜잭션 모델링
- ④ 접근 빈도

<문제 해설>

저장 레코드 양식 설계 시 고려 사항

- 데이터 타입, 데이터 값의 분포, 접근 빈도

오답피하기 | 트랜잭션 모델링

- 트랜잭션을 개념적 시스템 독립적으로 정의한다.

- 트랜잭션의 입출력 기능, 형태만 정의한다.

- 검색, 갱신, 혼합(검색, 갱신)

[해설작성자 : 두목넷]

3번 보기 트랜잭션 모델링은 개념적 설계 단계에서 하는것

[해설작성자 : ㅇㅈㅈ]

47. SQL의 기능에 따른 분류 중에서 REVOKE문과 같이 데이터의 사용 권한을 관리하는데 사용하는 언어는?

- ① DDL(Data Definition Language)
- ② DML(Data Manipulation Language)
- ③ DCL(Data Control Language)
- ④ DUL(Data User Language)

<문제 해설>

DCL 종류

- COMMIT : 명령어로 수행된 결과를 실제 물리적 디스크로 저장하고, 명령어로 수행을 성공적으로 완료하였음을 선언한다.

- ROLLBACK : 명령어로 수행을 실패하였음을 알리고, 수행된 결과를 원상복구시킨다.

- GRANT : 데이터베이스 사용자에게 사용 권한 부여한다.

- REVOKE : 데이터베이스 사용자로부터 사용 권한 취소한다.

[해설작성자 : 두목넷]

DDL

- CREATE : SCHEMA(스키마), DOMAIN(도메인), TABLE(테이블), VIEW(뷰), INDEX(인덱스)를 정의

- ALTER : TABLE에 대한 정의를 변경하는 데 사용

- DROP : SCHEMA(스키마), DOMAIN(도메인), TABLE(테이블), VIEW(뷰), INDEX(인덱스)를 삭제

DML

- SELECT : 테이블에서 조건에 맞는 튜플을 검색

- INSERT : 테이블에 새로운 튜플을 삽입

- DELETE : 테이블에서 조건에 맞는 튜플을 삭제

- UPDATE : 테이블에서 조건에 맞는 튜플의 내용 변경

[해설작성자 : 전자공학도]

48. 데이터 사전에 대한 설명으로 틀린 것은?

- ① 시스템 카탈로그 또는 시스템 데이터베이스라고도 한다.

② 데이터 사전 역시 데이터베이스의 일종이므로 일반 사용자가 생성, 유지 및 수정 할 수 있다.

③ 데이터베이스에 대한 데이터인 메타데이터(Metadata)를 저장하고 있다.

④ 데이터 사전에 있는 데이터에 실제로 접근하는 데 필요한 위치 정보는 데이터 디렉토리(Data Directory)라는 곳에서 관리한다.

<문제 해설>

데이터 사전(Data Dictionary)

- 시스템 자신이 필요로 하는 여러 가지 객체(기본 테이블, 뷰, 인덱스, 데이터베이스, 패키지, 접근 권한 등)에 관한 정보를 포함하고 있는 시스템 데이터베이스이다.

- 시스템 카탈로그(System Catalog), 메타 데이터(Meta Data)라고도 한다.

- 시스템 카탈로그 자체도 시스템 테이블로 구성되어 있어 SQL 문을 이용하여 내용 검색이 가능하다.

- 사용자가 시스템 카탈로그를 직접 갱신할 수 없다.

- SQL 문으로 여러 가지 객체에 변화를 주면 시스템이 자동으로 갱신한다.

[해설작성자 : 두목넷]

49. 데이터베이스에서 릴레이션에 대한 설명으로 틀린 것은?

- ① 모든 튜플은 서로 다른 값을 가지고 있다.
- ② 하나의 릴레이션에서 튜플은 특정한 순서를 가진다.
- ③ 각 속성은 릴레이션 내에서 유일한 이름을 가진다.
- ④ 모든 속성 값은 원자 값(atomic value)을 가진다.

<문제 해설>

* 릴레이션의 특징

- 튜플의 유일성 : 모든 튜플은 서로 다른값을 갖는다

- 튜플의 무순서성 : 하나의 릴레이션에 튜플에 순서는 없다

- 속성의 원자성 : 속성값은 원자값을 갖는다

- 속성의 무순서성 : 각 속성은 릴레이션내에서 유일한 이름을 가지며, 순서는 큰 의미가 없다

[해설작성자 : 합격드가자~]

50. 데이터베이스에서의 뷰(View)에 대한 설명으로 틀린 것은?

- ① 뷰는 다른 뷰를 기반으로 새로운 뷰를 만들 수 있다.
- ② 뷰는 일종의 가상 테이블이며, update에는 제약이 따른다.
- ③ 뷰는 기본 테이블을 만드는 것처럼 create view를 사용하여 만들 수 있다.
- ④ 뷰는 논리적으로 존재하는 기본 테이블과 다르게 물리적으로만 존재하며 카탈로그에 저장된다.

<문제 해설>

뷰는 논리적으로만 존재한다.

[해설작성자 : 칼방이]

51. 트랜잭션의 상태 중 트랜잭션의 마지막 연산이 실행된 직후의 상태로, 모든 연산의 처리는 끝났지만 트랜잭션이 수행한 최종 결과를 데이터베이스에 반영하지 않은 상태는?

- ① Active
- ② Partially Committed
- ③ Committed
- ④ Aborted

<문제 해설>

Partially Committed: 마지막 연산이 실행된 직후의 상태로 아직 Commit 연산 실행 전
Committed: 트랜잭션이 실행을 성공적으로 완료하여 Commit 연산을 수행한 상태
[해설작성자 : 칼방이]

Partially Committed: 트랜잭션이 마지막 명령문을 실행한 직후의 상태
Active: 트랜잭션이 실행을 시작하여 실행 중인 상태
Committed: 트랜잭션이 실행을 성공적으로 완료하여 COMMIT 연산을 수행한 상태
Aborted: 트랜잭션이 실행에 실패하여 ROLLBACK연산을 수행한 상태
[해설작성자 : 제발합격]

52. SQL의 명령을 사용 용도에 따라 DDL, DML, DCL로 구분할 경우, 그 성격이 나머지 셋과 다른 것은?

- ① SELECT ② UPDATE
- ③ INSERT ④ GRANT

<문제 해설>

* DML

- 1. SELECT : 테이블에서 조건에 맞는 튜플 검색
- 2. INSERT : 테이블에 새로운 튜플 삽입
- 3. UPDATE : 테이블에서 조건에 맞는 튜플의 내용 갱신
- 4. DELETE : 테이블에서 조건에 맞는 튜플 삭제

* DCL

- 1. COMMIT : 명령에 의해 수행된 결과를 실제 물리적 디스크로 저장하고, 데이터베이스 조작 작업이 정상적으로 완료됐음을 알려주는 명령어
- 2. ROLLBACK : 아직 COMMIT 되지 않은 변경된 모든내용 들을 취소하고 , 데이터베이스를 이전 상태로 되돌리는 명령어
- 3. GRANT : 데이터베이스 사용자에게 사용 권한 부여
- 4. REVOKE : 데이터베이스 사용자의 사용 권한 취소

[해설작성자 : 루니]

53. 키의 종류 중 유일성과 최소성을 만족하는 속성 또는 속성들의 집합은?

- ① Atomic key ② Super key
- ③ Candidate key ④ Test key

<문제 해설>

Super key(슈퍼키) - 속성들의 집합으로 구성된 키, 릴레이션을 구성하는 모든 튜플들 중 슈퍼키로 구성된 속성의 집합과 동일한 값은 나타나지 않는다.
Candidate key(후보키) - 튜플을 유일하게 식별하기 위해 사용하는 속성들의 부분집합
[해설작성자 : 전자공학도]

Candidate key(후보키)는 기본키가 될 수 있는 것들이기 때문에 유일성과 최소성을 만족하지만 Super key(슈퍼키)는 동일한 것이 없기에 유일성은 만족하나 속성의 집합이기 때문에 최소성을

만족하지 않음

[해설작성자 : 정처기 파이팅합시다]

후보키(Candidate Key): 기본키로 사용 가능한 속성, 유일성/최소성 만족
기본키(Primary Key): 후보키 중에서 선택됨, 중복된 값과 Null 값 가질 수 없음. 반드시 필요한 키
슈퍼키(Super Key): 한 릴레이션 내 속성들의 집합으로 구성된 키, 유일성○, 최소성X
[해설작성자 : 합격가자]

54. 데이터베이스에서 개념적 설계 단계에 대한 설명으로 틀린 것은?

- ① 산출물로 E-R Diagram을 만들 수 있다.
- ② DBMS에 독립적인 개념 스키마를 설계한다.
- ③ 트랜잭션 인터페이스를 설계 및 작성한다.
- ④ 논리적 설계 단계의 앞 단계에서 수행된다.

<문제 해설>

데이터베이스 설계단계에서의 트랜잭션 설계단계

- 개념설계 : 트랜잭션 모델링
- 논리설계 : 트랜잭션 인터페이스 설계
- 물리설계 : 트랜잭션 세부설계

[해설작성자 : 두목넷]

개념적 설계 : 트랜잭션 모델링/ 독립적인 개념 스키마 모델링/ 개념스키마 설계/ E-R 다이어그램
논리적 설계 : 트랜잭션 인터페이스 설계/ DBMS에 맞는 논리스키마 설계/ 테이블 설계/ 논리적 매핑/ 스키마 평가 및 정제
물리적 설계 : 레코드 집중의 분석,설계/ 저장레코드 양식 설계 / 저장 구조 및 액세스 경로 설정
[해설작성자 : 브라더진 내일 시험본다..]

55. 테이블의 기본키(Primary Key)로 지정된 속성에 관한 설명으로 가장 거리가 먼 것은?

- ① NOT NULL로 널 값을 가지지 않는다.
- ② 릴레이션에서 튜플을 구별할 수 있다.
- ③ 외래키로 참조될 수 있다.
- ④ 검색할 때 반드시 필요하다.

<문제 해설>

기본키 - 후보키 중에서 선정된 주키(MAIN KEY)로 중복된 값을 가질 수 없다.
- 한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성
- 기본키는 후보키의 성질을 갖는다..즉 튜플을 식별하기 위해 반드시 필요한 키
- 기본키는 NULL 값을 가질 수 없다..즉 기본키로 설정된 속성에는 NULL값이 있어서는 안된다.
[해설작성자 : comcbt.com 이용자]

- 검색할때 반드시 필요한 것은 아니다
- 하지만 기본키로 검색하면 유일한 튜플을 조회할 수 있다.
[해설작성자 : 정처기가 필요한 개발자]

56. 데이터 모델의 구성 요소 중 데이터 구조에 따라 개념 세계 나 컴퓨터 세계에서 실제로 표현된 값들을 처리하는 작업을 의미하는 것은?

- ① Relation ② Data Structure
 ③ Constraint ④ Operation

<문제 해설>

데이터 모델의 구성 요소

- 구조(Structure):논리적인 개체 타입들 간의 관계, 데이터 구조 및 정적 성질을 표현

- 연산(Operation):실제 데이터를 처리하는 작업에 대한 명세로, 조작하는 기본 도구

- 제약 조건(Constraint):DB에 저장될 수 있는 실제 데이터의 논리적인 제약 조건

[해설작성자 : 제발 한번에 합격하고 싶어욤]

데이터 모델링의 구성요소

- 논리적구조(Structure)

- 연산(Operation)

- 제약조건(Constraint)

논리 데이터 모델의 구성요소

- 개체

- 속성

- 관계

[해설작성자 : ㅁㅁ]

57. 다음 [조건]에 부합하는 SQL문을 작성하고자 할 때, [SQL문]의 빈칸에 들어갈 내용으로 옳은 것은? (단, '팀코드' 및 '이름'은 속성이며, '직원'은 테이블이다.)

[조건]

이름이 '정도일'인 팀원이 소속된 팀코드를 이용하여 해당 팀에 소속된 팀원들의 이름을 출력하는 SQL문 작성

[SQL문]

SELECT 이름
 FROM 직원
 WHERE 팀코드 = ();

- ① WHERE 이름 = '정도일'
 ② SELECT 팀코드 FROM 이름
 WHERE 직원 = '정도일'
 ③ WHERE 직원 = '정도일'
 ④ SELECT 팀코드 FROM 직원
 WHERE 이름 = '정도일'

<문제 해설>

- 하위 질의문은 하위 질의를 먼저 처리하고 검색된 결과는 상위 질의에 적용되어 검색된다.

- 직원 테이블에서 “정도일” 팀원의 팀 코드를 검색하여 상위 질의에 반환한다.

[해설작성자 : 두목넷]

58. 무결성 제약조건 중 개체 무결성 제약조건에 대한 설명으로 옳은 것은?

- ① 릴레이션 내의 튜플들이 각 속성의 도메인에 정해진 값만을 가져야 한다.
 ② 기본키는 NULL 값을 가져서는 안되며 릴레이션 내에 오직 하나의 값만 존재해야 한다.
 ③ 자식 릴레이션의 외래키는 부모 릴레이션의 기본키와 도메인이 동일해야 한다.
 ④ 자식 릴레이션의 값이 변경될 때 부모 릴레이션의 제약을 받는다.

<문제 해설>

기본키(Primary Key)

- 테이블의 각 레코드를 고유하게 식별하는 필드나 필드의 집합이다.

- 테이블에 기본키 설정은 필수가 아니다.

- 기본키를 설정하지 않고도 다른 테이블과의 관계를 설정할 수 있다.

- 기본키는 혼합속성(다수 필드)에 설정할 수 있다.

[해설작성자 : 두목넷]

1.개체 무결성 : 각 릴레이션의 기본키를 구성하는 속성은 널(NULL) 값이나 중복된 값을 가질 수 없습니다.

2.참조무결성 : 외래키 값은 NULL이거나 참조하는 릴레이션의 기본키 값과 동일해야 합니다.

3.도메인 무결성 : 속성들의 값은 정의된 도메인에 속한 값이어야 합니다.

4.고유 무결성 : 특정 속성에 대해 고유한 값을 가지도록 조건이 주어진 경우, 릴레이션의 각 튜플이 가지는 속성 값들은 서로 달라야 합니다.

5.NULL 무결성 : 릴레이션의 특정 속성 값은 NULL 될 수 없습니다.

6.키 무결성 : 각 릴레이션은 최소한 한 개 이상의 키가 존재해야 합니다.

[해설작성자 : comcbt.com 이용자]

59. 관계 데이터 모델에서 릴레이션(Relation)에 포함되어 있는 튜플(Tuple)의 수를 무엇이라고 하는가?

- ① Degree ② Cardinality
 ③ Attribute ④ Cartesian product

<문제 해설>

- 차수(Degree) : 속성의 수

- 카디널리티(Cardinality) : 튜플의 수 (기수)

[해설작성자 : 두목넷]

60. 사용자 'PARK'에게 테이블을 생성할 수 있는 권한을 부여하기 위한 SQL문의 구성으로 빈칸에 적합한 내용은?

[SQL 문]

GRANT () PARK ;

- ① CREATE TABLE TO ② CREATE TO
 ③ CREATE FROM ④ CREATE TABLE FROM

<문제 해설>

PARK 에게 [CREATE TABLE] 권한 취소

REVOKE CREATE TABLE FROM PARK;

[해설작성자 : comcbt.com 이용자]

4과목 : 프로그래밍 언어 활용

61. C언어에서 문자열 처리 함수의 서식과 그 기능의 연결로 틀린 것은?

- ① strlen(s) - s의 길이를 구한다.
- ② strcpy(s1, s2) - s2를 s1으로 복사한다.
- ③ strcmp(s1, s2) - s1과 s2를 연결한다.
- ④ strrev(s) - s를 거꾸로 변환한다.

<문제 해설>

strcat - 문자열 연결 함수

[해설작성자 : 수도공고갤러리 켈주]

strcmp - 문자열 비교

[해설작성자 : comcbt.com 이용자]

함수명에 용도를 의미하는 약어가 포함되어 있음

str -> string

len -> length

cpy -> copy

cmp -> compare

rev -> reverse

[해설작성자 : Jinni]

62. 다음 C언어 프로그램이 실행되었을 때, 실행 결과는?

```
#include <stdio.h>
int main(int argc, char *argv[]) {
    int a = 5, b = 3, c = 12;
    int t1, t2, t3;
    t1 = a && b;
    t2 = a || b;
    t3 = !c;
    printf("%d", t1 + t2 + t3);
    return 0;
}
```

- ① 0 ② 2
- ③ 5 ④ 14

<문제 해설>

a && b -> 1

AND연산

a || b -> 1

OR연산

!c -> 0

NOT연산

1 + 1 + 0 = 2 (최종정답)

C언어 연산자에서는 0이 아닌 모든값은 1(TRUE)로 본다.

그래서 a=1 b=1 c=1 로 바뀌어서 대입해도 같은결과가 나온다

[해설작성자 : 루니]

Int a,b,c에 정의된 값들을 조건절에 대입.

t1= 1&&2 *&&는 좌우가 모두 참이어야 함, 좌우에 모든 값이 존재함으로 1=참

t2=1||2 *||는 좌우 중 1개만 참이어도 참, 좌우 모든 값이 존재함으로 1=참

t3=!c *!는 NOT연산으로 3의 값이 없음을 말하기에 0=거짓
 그래서 1+1+0 =2

[해설작성자 : 사람]

비트연산자랑 헷갈릴수 있는데 비트연산자는 & 하나만 있어야함

[해설작성자 : 빛나는 하루]

63. 다음 C언어 프로그램이 실행되었을 때, 실행 결과는?

```
#include <stdio.h>
struct st{
    int a;
    int c[10];
};

int main (int argc, char *argv[]) {
    int i = 0;
    struct st ob1;
    struct st ob2;
    ob1.a = 0;
    ob2.a = 0;

    for(i = 0; i < 10; i++) {
        ob1.c[i] = i;
        ob2.c[i] = ob1.c[i] + i;
    }

    for(i = 0; i < 10; i = i+2) {
        ob1.a = ob1.a + ob1.c[i];
        ob2.a = ob2.a + ob2.c[i];
    }

    printf("%d", ob1.a + ob2.a);
    return 0;
}
```

- ① 30 ② 60
- ③ 80 ④ 120

<문제 해설>

ob1.c[i] = i

ob2.c[i] = i + i

ob1.a = ob1.a + ob1.c[i] = 0+2+4+6 + 8 = 20

ob2.a = ob2.a + ob2.c[i] = 0+4+8+12+16 = 40

20 + 40 = 60

[해설작성자 : 루니]

64. IP 프로토콜에서 사용하는 필드와 해당 필드에 대한 설명으로 틀린 것은?

- ① Header Length는 IP 프로토콜의 헤더 길이를 32비트 위

드 단위로 표시한다.

- ② Packet Length는 IP 헤더를 제외한 패킷 전체의 길이를 나타내며 최대 크기는 $2^{32} - 1$ 비트이다.
- ③ Time To Live는 송신 호스트가 패킷을 전송하기 전 네트워크에서 생존할 수 있는 시간을 지정한 것이다.
- ④ Version Number는 IP 프로토콜의 버전번호를 나타낸다.

<문제 해설>

Total Packet Length (16 bits): IP 헤더 및 데이터를 포함한 IP 패킷 전체 길이를 바이트 단위로 길이를 표시. (최대값은 $65,535 = 2^{16} - 1$)

[해설작성자 : 루니]

65. 다음 Python 프로그램의 실행 결과가 [실행결과]와 같을 때, 빈칸에 적합한 것은?

```
x = 20

if x == 10 :
    print('10')
(   ) x == 20 :
    print('20')
else :
    print('other')
```

[실행결과]

20

- ① either ② elif
- ③ else if ④ else

<문제 해설>

python 조건문 if, elif, else

else if 는 없습니다 ..ㅠㅠ

[해설작성자 : 쉐랏]

66. RIP 라우팅 프로토콜에 대한 설명으로 틀린 것은?

- ① 경로 선택 메트릭은 홉 카운트(hop count)이다.
- ② 라우팅 프로토콜을 IGP와 EGP로 분류했을 때 EGP에 해당한다.
- ③ 최단 경로 탐색에 Bellman-Ford 알고리즘을 사용한다.
- ④ 각 라우터는 이웃 라우터들로부터 수신한 정보를 이용하여 라우팅 표를 갱신한다.

<문제 해설>

RIP(Routing Information Protocol)

- 최단 경로 탐색에는 Bellman-Ford 알고리즘을 사용하는 거리 벡터 라우팅 프로토콜

- 라우팅 프로토콜을 IGP와 EGP로 분류했을 때 IGP에 해당한다.

- 최단경로탐색에는 Bellman-Ford 알고리즘을 사용한다.

- 최적의 경로를 산출하기 위한 정보로서 홉(거리 값)만을 고려하므로, RIP을 선택한 경로가 최적의 경로가 아닌 경우가 많이 발생할 수 있다.

- 소규모 네트워크 환경에 적합하다.

- 최대 홉 카운트를 15홉 이하로 한정하고 있다.

[해설작성자 : 두목넷]

보기 2번 추가 내용 :

* 자율 시스템(AS : Autonomous System)

- 하나의 기관 내부를 나타내는 것으로 라우터들과 통신기기들의 집합. 예를 들면 하나의 학내망

* IGP(Interior Gateway Protocol, 내부 게이트웨이 프로토콜)

- 학내망 내부에서 라우팅하는것. 즉 하나의 AS 내에서의 라우팅에 사용되는 프로토콜

* EGP(Exterior Gateway Protocol, 외부 게이트웨이 프로토콜)

- 학내망과 다른 학내망의 라우팅에 사용. 즉 AS간의 라우팅에 사용되는 프로토콜

RIP는 OSPF와 함께 IGP(내부 게이트웨이 프로토콜)에 해당한다

[해설작성자 : 합격드가자~]

RIP = 홉 기반 + 소규모 + Bellman-Ford + 최대 15홉

IGP = 내부용 (RIP, OSPF)

EGP = 외부용 (BGP)

[해설작성자 : 황승현]

67. 다음에서 설명하는 프로세스 스케줄링은?

최소 작업 우선(SJF) 기법의 약점을 보완한 비선점 스케줄링 기법으로 다음과 같은 식을 이용해 우선순위를 판별한다.

우선순위 = $\frac{\text{대기한 시간} + \text{서비스를 받을 시간}}{\text{서비스를 받을 시간}}$

- ① FIFO 스케줄링 ② RR 스케줄링
- ③ HRN 스케줄링 ④ MQ 스케줄링

<문제 해설>

HRN 스케줄링 방식

- 비선점 스케줄링

실행시간이 긴 프로세스에 불리한 SJF 을 보완하기 위해 대기시간 및 서비스 시간을 이용

긴 작업과 짧은 작업 간의 지나친 불평등을 해소할 수 있다.

우선순위를 계산 숫자가 높은것부터 낮은순으로 순위 부여

(대기시간 + 서비스시간) / 서비스시간 = 우선순위값) 값이

클수록 우선순위가 높다.

[해설작성자 : 수도공고갤러리 갤주]

HRN 스케줄링 : Highest Response Ratio Next

[해설작성자 : comcbt.com 이용자]

68. UNIX 운영체제에 관한 특징으로 틀린 것은?

- ① 하나 이상의 작업에 대하여 백그라운드에서 수행이 가능하다.
- ② Multi-User는 지원하지만 Multi-Tasking은 지원하지 않는다.
- ③ 트리 구조의 파일 시스템을 갖는다.
- ④ 이식성이 높으며 장치 간의 호환성이 높다.

<문제 해설>

UNIX 운영체제

주로 서버용 컴퓨터에서 사용됨

time sharing system을 위해 설계된 대화식 운영체제

c언어로 작성되어 이식성 높고, 장치간 호환성 높다.

Multi user, Multi tasking 모두 지원.

트리 구조의 파일시스템

[해설작성자 : 합격하자]

69. UDP 프로토콜의 특징이 아닌 것은?

- ① 비연결형 서비스를 제공한다.
- ② 단순한 헤더 구조로 오버헤드가 적다.
- ③ 주로 주소를 지정하고, 경로를 설정하는 기능을 한다.
- ④ TCP와 같이 트랜스포트 계층에 존재한다.

<문제 해설>

UDP(User Datagram Protocol)

- 비연결형 및 비신뢰성 전송 서비스를 제공한다.
- 흐름 제어나 순서 제어가 없어 전송 속도가 빠르다.
- 수신된 데이터의 순서 재조정 기능을 지원하지 않는다.
- 복구 기능을 제공하지 않는다.

오답피하기

- ③ 번은 IP Protocol에 관한 설명이다.

[해설작성자 : 두목넷]

70. Python 데이터 타입 중 시퀀스(Sequence) 데이터 타입에 해당하며 다양한 데이터 타입들을 주어진 순서에 따라 저장할 수 있으나 저장된 내용을 변경할 수 없는 것은?

- ① 복소수(complex) 타입
- ② 리스트(list) 타입
- ③ 사전(dict) 타입
- ④ 튜플(tuple) 타입

<문제 해설>

리스트 타입 : 가변형

튜플 타입 : 불변형

[해설작성자 : comcbt.com 이용자]

시퀀스- 리스트 - 순서있고,가변 [1,2,3]

시퀀스- 튜플 - 순서있고,불변 (1,2,3)

세트 - 세트 - 순서없고,중복x {1,2,3}

맵 - 딕셔너리- 순서없고,key-value쌍 {'a':1,'b':2,'c':3}

[해설작성자 : comcbt.com 이용자]

71. 다음 JAVA 프로그램이 실행되었을 때, 실행결과?

```
public class Rarr {
    static int[] marr() {
        int temp[] = new int[4];
        for(int i=0; i < temp.length; i++)
            temp[i] = i;
        return temp;
    }
    public static void main(String[] args) {
        int iarr[];
        iarr = marr();
        for(int i = 0; i < iarr.length; i++)
            System.out.print(iarr[i] + " ");
    }
}
```

- ① 1 2 3 4
- ② 0 1 2 3
- ③ 1 2 3
- ④ 0 1 2

<문제 해설>

0부터 시작해서 4보다 작은 정수

+는 1씩 증가

[해설작성자 : 널시형]

72. 다음 JAVA 프로그램이 실행되었을 때의 결과는?

```
public class ovr {
    public static void main(String[] args) {
        int a = 1, b = 2, c = 3, d = 4;
        int mx, mn;
        mx = a < b ? b : a;
        if(mx==1) {
            mn = a > mx ? b : a;
        }
        else {
            mn = b < mx ? d : c;
        }
        System.out.println(mn);
    }
}
```

- ① 1
- ② 2
- ③ 3
- ④ 4

<문제 해설>

mx = a < b ? b : a;

a < b 조건문이 참일 경우 전자인 b, 거짓일 경우 후자인 a 가 mx의 값이 됨

여기선 a = 1, b = 2 이기 때문에 a < b 는 참이므로 mx = b = 2

그 다음

if 절의 조건 mx == 1 은 mx = 2 이기 때문에 거짓, else 문으로 넘어감

mn = b < mx ? d : c;

b = 2, mx = 2 이기 때문에 b < mx 조건은 거짓, 그러므로 후

자인 c 의 값이 mn에 대입됨

mn = c = 3

결과적으로, mn을 출력하게 되면 c의 값 3이 나온다

[해설작성자 : 공부중]

mx = 1 < 2 ? 2 : 1;

상항연산 결과가 mx = 2가 들어가는게 맞구요

else 분기에 타게되어, mn = 2 < mx(2) ? 4 : 3;

mn 결과 3 맞는데요?

[해설작성자 : comcbt.com 이용자]

a = 1, b = 2, c = 3, d = 4

mx = a < b ? b : a

mx = 1 < 2 ? 2 : 1

mx = 2

mx는 1이기 때문에 else 실행

mn = b < mx ? d : c

mn = 2 < 2 ? 4 : 3

mn = 3

[해설작성자 : cl0ud]

mx = a < b ? b : a;

→ 1 < 2 이므로 참 → mx = 2

if (mx == 1)

→ mx = 2 이므로 조건 거짓 → else 실행

mn = b < mx ? d : c;

→ 2 < 2는 거짓 → mn = c = 3

System.out.println(mn);

→ 출력: 3

[해설작성자 : 헛갈리실까봐 다시 정리해드려요]

73. 다음 중 Myers가 구분한 응집도(Cohesion)의 정도에서 가장 낮은 응집도를 갖는 단계는?

- ① 순차적 응집도(Sequential Cohesion)
- ② 기능적 응집도(Functional Cohesion)
- ③ 시간적 응집도(Temporal Cohesion)
- ④ 우연적 응집도(Coincidental Cohesion)

<문제 해설>

응집도

- (강함) 기능적 응집도 > 순차적 응집도 > 교환적 응집도 > 절차적 응집도 > 시간적 응집도 > 논리적 응집도 > 우연적 응집도 (약함)

[해설작성자 : 두목넷]

74. 다음 C언어 프로그램이 실행되었을 때, 실행 결과는?

```
#include <stdio.h>
int main(int argc, char *argv[]) {
    int n1=1, n2=2, n3=3;
    int r1, r2, r3;

    r1=(n2<=2) || (n3>3);
    r2=!n3;
    r3=(n1>1) && (n2<3);

    printf("%d", r3-r2+r1);
    return 0;
}
```

- ① 0
- ② 1
- ③ 2
- ④ 3

<문제 해설>

r1 - or 연산 한쪽만 참이어도 1 반환 n2<=2(참) -> 1

r2 - != 부정연산자 3 부정 0 반환 -> 0

r3 - and 연산 둘다 참이어야 1반환 (n1>1) (거짓) -> 0

0 - 0 + 1 = 1

[해설작성자 : 수도공고갤러리 갤주]

!n3 -> 논리 부정 연산자 피연산자가 0이면 1을, 나머지 0을 반환

[해설작성자 : 차이카와]

75. IP 프로토콜의 주요 특징에 해당하지 않는 것은?

- ① 체크섬(Checksum) 기능으로 데이터 체크섬(Data Checksum)만 제공한다.
- ② 패킷을 분할, 병합하는 기능을 수행하기도 한다.
- ③ 비연결형 서비스를 제공한다.
- ④ Best Effort 원칙에 따른 전송 기능을 제공한다.

<문제 해설>

데이터 체크섬은 제공하지 않고, 헤더 체크섬만 제공한다.

[해설작성자 : 루저]

76. 4개의 페이지를 수용할 수 있는 주기억장치가 있으며, 초기에는 모두 비어 있다고 가정한다. 다음의 순서로 페이지 참조가 발생할 때, LRU 페이지 교체 알고리즘을 사용할 경우 몇 번의 페이지 결함이 발생하는가?

페이지 참조 순서 1, 2, 3, 1, 2, 4, 1, 2, 5

- ① 5회
- ② 6회
- ③ 7회
- ④ 8회

<문제 해설>

1 | 1 [결함(부재)]

2 | 1 2 [결]

3 | 1 2 3 [결]

1 | 1 2 3 [적재되어 있는 1 참조]

2 | 1 2 3 [적재되어 있는 2 참조]

4 | 1 2 3 4 [결]

1 | 1 2 3 4 [적재되어 있는 1 참조]

2 | 1 2 3 4 [적재되어 있는 2 참조]

5 | 1 2 5 4 [결], [최근까지 남아있었던(LRU기법) 3을 제거, 5 적재]

총 결함 발생 횟수는 5회입니다.

[해설작성자 : COOKIE]

마지막 부분을 가장 오랫동안 사용하지 않은 페이지 제거 (LRU 기법)로 수정하면 더 좋을 것 같습니다.

[해설작성자 : boan]

77. 사용자 수준에서 지원되는 스레드(thread)가 커널에서 지원되는 스레드에 비해 가지는 장점으로 옳은 것은?

- ① 한 프로세스가 운영체제를 호출할 때 전체 프로세스가 대기할 필요가 없으므로 시스템 성능을 높일 수 있다.
- ② 동시에 여러 스레드가 커널에 접근할 수 있으므로 여러 스레드가 시스템 호출을 동시에 사용할 수 있다.
- ③ 각 스레드를 개별적으로 관리할 수 있으므로 스레드의 독립적인 스케줄링이 가능하다.
- ④ 커널 모드로의 전환 없이 스레드 교환이 가능하므로 오버헤드가 줄어든다.

<문제 해설>

1,2,3 : 커널수준 스레드의 장점

[해설작성자 : 익명]

78. 한 모듈이 다른 모듈의 내부 기능 및 그 내부 자료를 참조하는 경우의 결합도는?

- ① 내용 결합도(Content Coupling)
- ② 제어 결합도(Control Coupling)
- ③ 공통 결합도(Common Coupling)
- ④ 스탬프 결합도(Stamp Coupling)

<문제 해설>

스탬프 결합도 - 두 모듈이 매개변수로 자료를 전달할 때, 자료 구조 형태로 전달되어 이용될 때 데이터가 결합되어 있다고 한다.

내용 결합도 - 하나의 모듈이 직접적으로 다른 모듈의 내용을 참조할 때 두 모듈은 내용적으로 결합되어 있다고 한다.

공통 결합도 - 두 모듈이 동일한 전역 데이터를 접근한다면 공통결합 되어 있다고 한다.

결합도 - 두 모듈간의 상호작용, 또는 의존도 정도를 나타내는 것이다

[해설작성자 : 수도공고갤러리 갈주]

내용 결합도

-하나의 모듈이 직접적으로 다른 모듈의 내용을 참조할 때 두 모듈은 내용적으로 결합되어 있는 경우의 결합도

TIP : 썰라 썰라~ 내용 참조, 자료 참조 (단어가 있으면)

내용 결합도

공통 결합도

-파라미터가 아닌 모듈 밖에 선언되어 있는 전역 변수를 참조하고 전역 변수를 갱신하는 식으로 상호작용하는 경우의 결합도

TIP : 썰라 썰라 "전역변수" (단어가 있으면) 공통 결합도!

[해설작성자 : 팀만 보세요]

79. a[0]의 주소값이 10일 경우 다음 C언어 프로그램이 실행되었을 때의 결과는? (단, int 형의 크기는 4Byte로 가정한다.)

```
#include <stdio.h>
int main(int argc, char *argv[]) {
    int a[] = {14, 22, 30, 38};
    printf("%u, ", &a[2]);
    printf("%u", a);
    return 0;
}
```

- ① 14, 10 ② 14, 14
- ③ 18, 10 ④ 18, 14

<문제 해설>

배열은 자료형이 같은 변수를 메모리에 연속으로 할당합니다. int형인 배열 요소 14, 22, 30, 38은 제시된 문제에 따라 메모리 10번지부터 시작하여 4바이트 간격으로 각각 10, 14, 18, 22번지의 주소값을 할당 받습니다..

%u <--- 부호 없는 10진수로 출력하는 변환 문자입니다..

printf("%u, ", &a[2]); <--- &a[2]는 &a + 2 또는 10 + (2 * sizeof(int)) 이며 18입니다..18번지부터 21번지까지의 총 4바이트 저장 공간인 주소 값을 뜻합니다..

printf("%u", a); <--- 배열명은 첫 번째 배열 요소(a[0])의 주소값 입니다..

"%u, "에는 쉼표와 띄어쓰기가 있으므로 실행 결과는 18, 10 입니다..

[해설작성자 : COOKIE]

80. 모듈화(Modularity)와 관련한 설명으로 틀린 것은?

- ① 시스템을 모듈로 분할하면 각각의 모듈을 별개로 만들고 수정할 수 있기 때문에 좋은 구조가 된다.
- ② 응집도는 모듈과 모듈 사이의 상호의존 또는 연관 정도를 의미한다.
- ③ 모듈 간의 결합도가 약해야 독립적인 모듈이 될 수 있다.
- ④ 모듈 내 구성 요소들 간의 응집도가 강해야 좋은 모듈 설계이다.

<문제 해설>

응집도 : 하나의 모듈이 하나의 기능을 수행하는 요소들간의 연관성 척도, 독립적인 모듈이 되기 위해서는 응집도가 강해야 한다.(결합도는 약해야 한다.)

[해설작성자 : 수도공고갤러리 갈주]

결합도: 모듈과 모듈 사이의 상호의존 또는 연관 정도를 의미한다.

응집도: 모듈이 독립적인 기능으로 정의되어 있는 정도를 의미한다.

[해설작성자 : 칼방이]

81. 소프트웨어 개발에서 정보보안 3요소에 해당 하지 않는 설명은?

- ① 기밀성 : 인가된 사용자에게 대해서만 자원 접근이 가능하다.
- ② 무결성 : 인가된 사용자에게 대해서만 자원 수정이 가능하며 전송중인 정보는 수정되지 않는다.
- ③ 가용성 : 인가된 사용자는 가지고 있는 권한 범위 내에서 언제든지 자원 접근이 가능하다.
- ④ 휘발성 : 인가된 사용자가 수행한 데이터는 처리 완료 즉시 폐기 되어야 한다.

<문제 해설>

정보보안 3요소 : 무결성, 기밀성, 가용성 (무기가)

[해설작성자 : comcbt.com 이용자]

82. 어떤 외부 컴퓨터가 접속되면 접속 인가 여부를 점검해서 인가된 경우에는 접속이 허용되고, 그 반대의 경우에는 거부할 수 있는 접근제어 유틸리티는?

- ① tcp wrapper ② trace checker
- ③ token finder ④ change detector

<문제 해설>

tcp/ip 통신에서, tcp는 3-way handshake 를 통해 통신이 원활히 이뤄질 수 있는지를 확인합니다.

[해설작성자 : 김구름]

tcp wrapper는 tcp 3-way handshake 인증 과정 직후에 수행됩니다.

[해설작성자 : 얼뽕]

83. 기기를 키오스크에 갖다 대면 원하는 데이터를 바로 가져올 수 있는 기술로 10cm 이내 근접 거리에서 기가급 속도로 데이터 전송이 가능한 초고속 근접무선통신(NFC : Near Field Communication) 기술은?

- ① BcN(Broadband Convergence Network)
- ② Zing
- ③ Marine Navi
- ④ C-V2X(Cellular Vehicle To Everything)

<문제 해설>

근거리로 가까이 가면 지이이이잉 하고 소리가 난다고 외우세요.

[해설작성자 : 연상암기황제]

1번 BcN은 음성·데이터·유무선 등 통신·방송·인터넷이 융합된 품질 보장형 광대역 멀티미디어 서비스를 공간에 제한없이 실시간으로 이용할 수 있는 차세대 네트워크 환경이다.

3번 Marine Navi는 소형 선박에 설치된 GPS 기반 선박자동 식별장치(AIS: Automatic Identification System)를 기본으로 주변 선박의 이동 속도와 위치를 파악한다.

4번 엘티이(LTE), 5G와 같은 셀룰러 이동통신망을 통해 차량이 다른 차량이나 교통 인프라, 보행자, 네트워크 등과 정보를 서로 주고받는 차량 통신 기술.

[해설작성자 : 곰팡이친구먼지]

84. 취약점 관리를 위한 응용 프로그램의 보안 설정과 가장 거리가 먼 것은?

- ① 서버 관리실 출입 통제
- ② 실행 프로세스 권한 설정
- ③ 운영체제의 접근 제한
- ④ 운영체제의 정보 수집 제한

<문제 해설>

응용프로그램 취약점 관리

- 패치 관리 : 응용프로그램에 대한 패치 또는 서비스 팩 적용
- 응용프로그램 실행권한의 제한 필요
- 운영체제의 접근 제한
- 정보 수집 제한
- 불필요한 서비스 및 악성 프로그램의 확인 및 제거
- 시스템 무결성 검사 : 주요 파일을 검사하여 변경 내역 확인

[해설작성자 : 한비로]

취약점 분석

환경 및 시설: 도난, 정전-오작동, 천재지변

하드웨어: 고장-오작동, 데이터 유실

소프트웨어: 소프트웨어 장애, 정보유출, 데이터 소실

[해설작성자 : 카피바라팬]

85. 소프트웨어 개발 프레임워크와 관련한 설명으로 가장 적절하지 않은 것은?

- ① 반제품 상태의 제품을 토대로 도메인별로 필요한 서비스 컴포넌트를 사용하여 재사용성 확대와 성능을 보장 받을 수 있게 하는 개발 소프트웨어이다.
- ② 라이브러리와는 달리 사용자 코드에서 프레임워크를 호출해서 사용하고, 그에 대한 제어도 사용자 코드가 가지는 방식이다.
- ③ 설계 관점에 개발 방식을 패턴화시키기 위한 노력의 결과물인 소프트웨어 디자인 패턴을 반제품 소프트웨어 상태로 집적화시킨 것으로 볼 수 있다.
- ④ 프레임워크의 동작 원리를 그 제어 흐름의 일반적인 프로그램 흐름과 반대로 동작한다고 해서 IoC(Inversion of Control)이라고 설명하기도 한다.

<문제 해설>

사용자 코드에서 호출해서 사용하고 그에 대한 제어를 할 수 있는 것은 라이브러리이다..프레임워크는 이미 정해진 코드를 호출해 사용하고 자체적인 흐름을 가지고 있음.

[해설작성자 : 물?루]

86. 클라우드 기반 HSM(Cloud-based Hardware Security Module)에 대한 설명으로 틀린 것은?

- ① 클라우드(데이터센터) 기반 암호화 키 생성, 처리, 저장 등을 하는 보안 기기이다.
- ② 국내에서는 공인인증제의 폐지와 전자서명법 개정을 추진하면서 클라우드 HSM 용어가 자주 등장하였다.
- ③ 클라우드에 인증서를 저장하므로 기존 HSM 기기나 휴대폰에 인증서를 저장해 다닐 필요가 없다.
- ④ 하드웨어가 아닌 소프트웨어적으로만 구현되기 때문에

소프트웨어식 암호 기술에 내재된 보안 취약점을 해결할 수 없다는 것이 주요 단점이다.

<문제 해설>

하드웨어적으로 구현되므로 소프트웨어식 암호 기술에 내재된 보안 취약점을 해결할 수 있다.

[해설작성자 : 코봉이]

87. 다음 내용이 설명하는 기술로 가장 적절한 것은?

- 다른 국을 향하는 호출이 종계에 의하지 않고 직접 접속되는 그물 모양의 네트워크이다.
- 통신량이 많은 비교적 소수의 국 사이에 구성될 경우 경제적이며 간편하지만, 다수의 국 사이에는 회선이 세분화되며 비경제적일 수도 있다.
- 해당 형태의 무선 네트워크의 경우 대용량을 빠르고 안전하게 전달할 수 있어 행사장이나 군 등에서 많이 활용된다.

- ① Virtual Local Area Network
- ② Simple Station Network
- ③ Mesh Network
- ④ Modem Network

<문제 해설>

그물 = mesh

[해설작성자 : 산익]

88. 물리적 위협으로 인한 문제에 해당하지 않는 것은?

- ① 화재, 홍수 등 천재지변으로 인한 위협
- ② 하드웨어 파손, 고장으로 인한 장애
- ③ 방화, 테러로 인한 하드웨어와 기록장치를 물리적으로 파괴하는 행위
- ④ 방화벽 설정의 잘못된 조작으로 인한 네트워크, 서버 보안 위협

<문제 해설>

방화벽 설정의 잘못된 조작으로 인한 네트워크, 서버 보안 위협 - 기술적

[해설작성자 : 너굴맨]

89. 악성코드의 유형 중 다른 컴퓨터의 취약점을 이용하여 스스로 전파하거나 메일로 전파되며 스스로를 증식하는 것은?

- ① Worm
- ② Rogue Ware
- ③ Adware
- ④ Reflection Attack

<문제 해설>

2. Rogue Ware(Rogue security software; 가짜 백신 소프트웨어) : 사용자가 컴퓨터에 바이러스가 있다고 잘못 믿게 하고 컴퓨터에 실제로 악성 프로그램을 설치토록 하거나, 가짜 악성 프로그램 제거 도구에 대한 비용을 지불하도록 설득함. 공포심을 통해 사용자를 조종

3.Adware : 특정 소프트웨어를 실행할 때 또는 자동으로 활성화되는 광고프로그램으로 이 자체는 악성코드로 보기는 힘들지만, 무분별한 광고 팝업을 뜨게 하는 등의 악용 위험성으로 악성코드로 분류되기도 함.

4. Reflection Attack(반사공격) : 송신자가 생성한 메시지를 가로챈 공격자가 그 메시지를 다시 송신자에게 재전송하여 접근 권한을 얻는 형태의 공격 방법.

---(4번 상세 설명)

사전에 암호 키를 공유한 송신자와 수신자는 상대방 식별을 위하여 각각 난수값을 생성하여 전송하고 이에 대한 암호값을 요청한 후 수신된 암호값을 복호화하여 그 결과 값이 자신이 송신하였던 난수값과 일치하는지 여부를 확인하여 상대방을 인증한다..이때 공격자는 송신자가 보낸 난수값에 대한 암호값을 알기 위해 송신자가 보낸 난수값을 송신자에게 재전송하고, 이를 수신자의 난수값으로 인식한 송신자는 이에 대한 암호값을 공격자에게 전송하게 된다..공격자는 이 암호값을 다시 송신자에게 전송함으로써 자신을 인증시키고, 그 결과 접근 권한을 획득

[해설작성자 : 설명덕보기]

90. 다음 설명에 해당하는 공격기법은?

시스템 공격 기법 중 하나로 허용범위 미상의 ICMP 패킷을 전송하며 대상 시스템의 네트워크를 마비시킨다.

- ① Ping of Death
- ② Session Hijacking
- ③ Piggyback Attack
- ④ XSS

<문제 해설>

ICMP가 패킷을 많이 쏘니 핑 하고 죽어뵈네. 핑 오브 데스

[해설작성자 : 연상암기황제]

2. Session Hijacking(세션 가로채기) : 컴퓨터 시스템의 정보나 서비스에 무단으로 접근하기 위해 유효한 컴퓨터 세션(세션 키라고도 함)을 이용하는 것 일반적인 예로 두 컴퓨터 간에 활성화된 상태(즉, 로그인된 상태)에서 공격자가 피공격자의 로그인 정보를 활용하여 자신에게 필요한 행위를 하는 것을 뜻한다..

3.Piggyback Attack(피그백 공격) : 공격자가 다른 사용자의 연결에서 계정을 사용하지 않는 비활성 기간(비활성 간격)을 이용하여 시스템에 액세스(접근)한다..이는 간선(회선 간) 공격이라고도 불린다.

-> 쉽게 설명하자면, 시스템에 대한 합법적인 권한을 가진 사용자가 시스템에 접근할 때, 활성화된 기간(직접 로그인 인증을 받아야 하는 상황)에는 접근 권한이 없는 공격자가 비활성화된 기간(합법적 사용자가 시스템에 접근 인증을 받은 상황)에 마치 사용자와 관련있는 사람인 듯이 태그를 붙여 몰래 뒤따라 들어가는 것이라 할 수 있다.

[해설작성자 : 설명덕보기]

XSS (크로스 사이트 스크립팅) : 웹사이트에 악성 스크립트를 주입하는 행위. 공격자가 상대방의 브라우저에 스크립트가 실행되도록 해 사용자의 세션을 가로채거나, 웹사이트를 변조하거나, 악의적 콘텐츠를 삽입하거나, 피싱 공격을 진행하는 것.

[해설작성자 : 소곰]

91. 다음 설명에 해당하는 소프트웨어는?

- 개발해야 할 애플리케이션의 일부분이 이미 내장된 클래스 라이브러리로 구현이 되어 있다.
- 따라서, 그 기반이 되는 이미 존재하는 부분을 확장 및 이용하는 것으로 볼 수 있다.
- JAVA 기반의 대표적인 소프트웨어로는 스프링(Spring)이 있다.

- ① 전역 함수 라이브러리
- ② 소프트웨어 개발 프레임워크
- ③ 컨테이너 아키텍처
- ④ 어휘 분석기

<문제 해설>

프레임워크는 뼈대 골조를 의미하고 특정 기능을 수행하기 위한 "클래스"나 "인터페이스"를 모아둔 집합체

[해설작성자 : 산익]

92. 소프트웨어 개발 방법론 중 애자일(Agile) 방법론의 특징과 가장 거리가 먼 것은?

- ① 각 단계의 결과가 완전히 확인된 후 다음 단계 진행
- ② 소프트웨어 개발에 참여하는 구성원들 간의 의사소통 중시
- ③ 환경 변화에 대한 즉시 대응
- ④ 프로젝트 상황에 따른 주기적 조정

<문제 해설>

1번은 폭포수 모형의 특징

[해설작성자 : 코봉이]

93. 대칭 암호 알고리즘과 비대칭 암호 알고리즘에 대한 설명으로 틀린 것은?

- ① 대칭 암호 알고리즘은 비교적 실행 속도가 빠르기 때문에 다양한 암호의 핵심 함수로 사용될 수 있다.
- ② 대칭 암호 알고리즘은 비밀키 전달을 위한 키 교환이 필요하지 않아 암호화 및 복호화의 속도가 빠르다.
- ③ 비대칭 암호 알고리즘은 자신만이 보관하는 비밀키를 이용하여 인증, 전자서명 등에 적용이 가능하다.
- ④ 대표적인 대칭키 암호 알고리즘으로는 AES, IDEA 등이 있다.

<문제 해설>

대칭 암호화 알고리즘은 키 교환이 필요하기 때문에 키를 교환하는 중 키가 탈취될 수 있다는 문제가 있다.

공개 암호화 방식에 비해 속도가 빠르다.

[해설작성자 : 코봉이]

대칭 암호 알고리즘은 비밀키(Private Key)를 공유해야 하므로 키 교환이 필요함

[해설작성자 : 공부하자]

94. 두 명의 개발자가 5개월에 걸쳐 10000 라인의 코드를 개발하였을 때, 월별(man-month) 생산성 측정을 위한 계산 방식으로 가장 적합한 것은?

- ① 10000 / 2
- ② 10000 / (5×2)

③ 10000 / 5

④ (2×10000) / 5

<문제 해설>

생산성 = LOC/노력(인월)

10000/(5×2)

[해설작성자 : comcbt.com 이용자]

[비용 산정 계산법임]

노력 측정 = 개발기간 X 투입인원

개발비용 측정 = 개발기간 X 투입인원 x 단위비용

개발기간 측정 = 예측된 LOC(라인) / (투입인원 X 1인당 월평균 LOC)

생산성 측정 = 개발된 LOC / (투입인원 X 개발기간)

[해설작성자 : 우지]

95. 접근 통제 방법 중 조직 내에서 직무, 직책 등 개인의 역할에 따라 결정하여 부여하는 접근 정책은?

- ① RBAC
- ② DAC
- ③ MAC
- ④ QAC

<문제 해설>

(Role Based Access Control)

역할 기반 접근통제

권한이 있는 사용자들만 접근할 수 있는 보안 방법

[해설작성자 : 우리스카이]

DAC 임의접근통제 사용자또는 신분

MAC 강제접근통제. 미리 정해진 정책과 보안 등급

[해설작성자 : 산익]

MAC (Mandatory Access Control)

- 강제 접근 통제

- 미리 정해진 자원의 보안 레벨과 사용자에게 허락된 접근 권한 비교

DAC (Discretionary Access Control)

- 임의적 접근통제

- 자원에 대한 접근을 사용자나 그룹의 신분에 따라 제한

- 자원의 소유권을 가진 사람이 다른 사람의 접근을 허용하거나 제한할 수 있음

RBAC (Role Based Access Control)

- 사용자 역할에 따른 접근 통제

- 개별적인 신분이 아니라 조직 내 그룹 / 역할에 따라 부여

[해설작성자 : LBS]

96. COCOMO(Constructive Cost Model) 모형의 특징이 아닌 것은?

- ① 프로젝트를 완성하는데 필요한 man-month로 산정 결과를 나타낼 수 있다.
- ② Boehm이 제안한 것으로 원시코드 라인 수에 의한 비용 산정 기법이다.
- ③ 비교적 작은 규모의 프로젝트 기록을 통계 분석하여 얻은 결과를 반영한 모델이며 중소 규모 소프트웨어 프로

젝트 비용 추정에 적합하다.

- ④ 프로젝트 개발유형에 따라 object, dynamic, function의 3가지 모드로 구분한다.

<문제 해설>

개발유형에 따라 조직형(Organic), 반분리형(Semi-Detached), 내장형(Embedded)으로 구분한다.

[해설작성자 : 코봉이]

<cocomo 모델>

보엠이 고안한 소프트웨어 개발비용 산정 방법.

man-month = 투입 노력

개발 유형에 따른 3종류 - organic, semidetached, embedded (기본, 중간, 내장형 모드)

[해설작성자 : 함격하자]

개발 유형에 따른 3종류 - organic, semidetached, embedded (기본, 중간, 내장형 모드) -> (조직형, 반분리형, 내장형)

[해설작성자 : 김인직]

COCOMO 모형

- 원시 프로그램 규모(LOC)에 의한 비용 산정 기법
- 보헴(Boehm)이 제안
- 비교적 작은 규모의 프로젝트들을 통계 분석한 결과를 반영한 모델 -> 중소 규모 소프트웨어 프로젝트 비용 추정에 적합
- 비용 산정 결과는 Man-Month로 표현

소프트웨어 개발유형에 따른 분류

조직형(Organic)

- 5만 라인 이하
- 사무 처리용, 업무용, 과학용 소프트웨어 개발에 적합

반분리형(Semi-Detached)

- 30만 라인 이하
- 컴파일러, 인터프리터 개발에 적합

내장형(Embedded)

- 30만 라인 이상
- 미사일 유도 시스템, 실시간 처리 시스템 개발에 적합

[해설작성자 : LBS]

1~3은 COCOMO, 4는 럼바우(Rumbaugh) => 객체, 동적, 기능(object, dynamic, function)

[해설작성자 : 내일시험]

97. 각 사용자 인증의 유형에 대한 설명으로 가장 적절하지 않은 것은?

- ① 지식 : 주체는 '그가 알고 있는 것'을 보여주며 예시로는 패스워드, PIN 등이 있다.
- ② 소유 : 주체는 '그가 가지고 있는 것'을 보여주며 예시로는 토큰, 스마트카드 등이 있다.
- ③ 존재 : 주체는 '그를 대체하는 것'을 보여주며 예시로는 패턴, QR 등이 있다.

- ④ 행위 : 주체는 '그가 하는 것'을 보여주며 예시로는 서명, 움직임, 음성 등이 있다.

<문제 해설>

1. 지식 - 본인이 알고 있는 것 (ex. 패스워드, PIN 등)
2. 소유 - 본인이 가지고 있는 것 (ex. 토큰, 스마트카드 등)
3. 존재 - 본인을 나타내는 것 (ex. 홍채, 지문 등)
4. 행위 - 본인이 하는 것 (ex. 서명, 움직임, 음성 등)

[해설작성자 : 지나가는 행인]

'존재'라는 사용자 인증 유형은 없습니다.

1. 지식: 사용자 기반 인증 수행
ex) 패스워드, i-PIN
2. 소유: 사용자가 소유하고 있는 것을 기반으로 인증 수행
ex) 신분증, 메모리 카드, 스마트 카드, OTP
3. 생체: 사용자의 고유한 생체 정보를 기반으로 인증 수행
ex) 지문, 홍채, 얼굴, 음성, 정맥
4. 행위: 사용자의 행동정보를 기반으로 인증 수행
ex) 서명, 동작, 음성

5. 위치: 인증을 시도하는 위치의 적절성
ex) 콜백, GPS or IP 이용

[해설작성자 : 한기대 정치기 레츠고]

98. 시스템의 사용자가 로그인하여 명령을 내리는 과정에 대한 시스템의 동작 중 다음 설명에 해당하는 것은?

- 자신의 신원(Identity)을 시스템에 증명하는 과정이다.
- 아이디와 패스워드를 입력하는 과정이 가장 일반적인 예시라고 볼 수 있다.

- ① Aging
- ② Accounting
- ③ Authorization
- ④ Authentication

<문제 해설>

3번은 권한부여입니다.

[해설작성자 : 김구름]

자신의 신원을 시스템에 증명하는 과정을 인증(Authentication)이라고 합니다.

추가적으로 3번 보기는 인가(Authorization)를 의미하며 보기 4번인 인증하고 용어가 유사하니 혼돈하지 않을 정도로 숙지 하시면 됩니다.

[해설작성자 : 24년 정치기 공부]

99. 다음에서 설명하는 IT 기술은?

- 네트워크를 제어부, 데이터 전달부로 분리하여 네트워크 관리자가 보다 효율적으로 네트워크를 제어, 관리할 수 있는 기술
- 기존의 라우터, 스위치 등과 같이 하드웨어에 의존하는 네트워크 체계에서 안정성, 속도, 보안 등을 소프트웨어로 제어, 관리하기 위해 개발됨
- 네트워크 장비의 펌웨어 업그레이드를 통해 사용자의 직접적인 데이터 전송 경로 관리가 가능하고, 기존 네트워크에는 영향을 주지 않으면서 특정 서비스의 전송 경로 수정을 통하여 인터넷상에서 발생하는 문제를 처리할 수 있음

- ① SDN(Software Defined Networking)
- ② NFS(Network File System)
- ③ Network Mapper
- ④ AOE Network

<문제 해설>

NFS(Network File System) : 네트워크상에서 공유되는 파일 시스템. 다른 원격 호스트의 파일 시스템을 로컬 디스크에 접근하듯 간단하게 접근하여 자신의 디렉토리처럼 사용할 수 있다

Network Mapper : 네트워크 보안을 위한 유틸리티. 네트워크의 보안을 위해 물리적 연결과 어떤 서버와 운영 체제가 작동 중인 지 따위를 조사하는 응용 프로그램

AOE Network(Activity On Edge Network) : 어떤 프로젝트를 마치기까지 수행되는 작업의 각 단계(상태)를 그래프의 정점(Vertex)으로 표현하고, 작업 하나가 완료되어 다음 단계로 넘어가는 시간을 그래프의 간선(Edge)으로 나타낸 방향 그래프
 [해설작성자 : 한국공학대학교]

100. 프로젝트 일정 관리 시 사용하는 PERT 차트에 대한 설명에 해당하는 것은?

- ① 각 작업들이 언제 시작하고 언제 종료되는지에 대한 일정을 막대 도표를 이용하여 표시한다.
- ② 시간선(Time-line) 차트라고도 한다.
- ③ 수평 막대의 길이는 각 작업의 기간을 나타낸다.
- ④ 작업들 간의 상호 관련성, 결정경로, 경계시간, 자원할당 등을 제시한다.

<문제 해설>

1. 막대 도표가 아니라 네트워크 도표를 작성한다.

※ 1,2,3은 GANTT 차트에 대한 설명이다..

[해설작성자 : 익명]

PERT : Program Evaluation & Review Technique

[해설작성자 : comcbt.com 이용자]

본 해설집의 저작권은 www.comcbt.com에 있으며 카페, 블로그 등의 업로드 및 개인적 활용 이외에 문서의 수정 및 DB 저장, 기타 금전적 이익을 취하는 일체의 행위를 금지 합니다.

전자문제집 CBT 홈페이지 : www.comcbt.com

기출문제 및 해설집 다운로드 : www.comcbt.com/x

전자문제집 CBT 앱(구글플레이) : [\[다운로드\]](#)

전자문제집 CBT란? 인터넷으로 종이 없이 문제를 풀고 자동채점하는 프로그램으로 워드, 컴활, 기능사 등의 상설검정에서 사용하는 실제 프로그램 방식입니다.

해설을 제공하며 PC 버전 및 모바일 버전 완벽 연동 교사용/학생용 관리기능도 제공합니다.

오답 및 오탈자가 수정된 최신 자료와 해설은 전자문제집 CBT에서 확인하세요.

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