

정보처리기사 핵심내용 정리

협약(Contract)에 의한 설계

다음 () 안에 들어갈 내용으로 옳은 것은?

컴포넌트 설계 시 “()에 의한 설계”를 따를 경우, 해당 명세에서는

- (1) 컴포넌트의 오퍼레이션 사용 전에 참이 되어야 할 선행조건
- (2) 사용 후 만족되어야 할 결과조건
- (3) 오퍼레이션이 실행되는 동안 항상 만족되어야 할 불변조건 등이 포함되어야 한다.

20.8

핵심 042 협약(Contract)에 의한 설계



컴포넌트를 설계할 때 클래스에 대한 여러 가정을 공유할 수 있도록 명세한 것으로, 소프트웨어 컴포넌트에 대한 정확한 인터페이스를 명세한다.

- 협약에 의한 설계 시 명세에 포함될 조건에는 선행 조건, 결과 조건, 불변 조건이 있다.

선행 조건 (Precondition)	오퍼레이션이 호출되기 전에 참이 되어야 할 조건
결과 조건 (Postcondition)	오퍼레이션이 수행된 후 만족되어야 할 조건
불변 조건 (Invariant)	오퍼레이션이 실행되는 동안 항상 만족되어야 할 조건

- 협약 (Contract)에 의한 설계

GoF 디자인 패턴 중 생성 패턴 종류



추상 팩토리 (Abstract Factory)	<ul style="list-style-type: none">구체적인 클래스에 의존하지 않고, 인터페이스를 통해 서로 연관·의존하는 객체들의 그룹으로 생성하여 추상적으로 표현함연관된 서브 클래스를 묶어 한 번에 교체하는 것이 가능함
빌더 (Builder)	<ul style="list-style-type: none">작게 분리된 인스턴스를 건축 하듯이 조합하여 객체를 생성함객체의 생성 과정과 표현 방법을 분리하고 있어, 동일한 객체 생성에서도 서로 다른 결과를 만들어 낼 수 있음
팩토리 메소드 (Factory Method)	<ul style="list-style-type: none">객체 생성을 서브 클래스에서 처리하도록 분리하여 캡슐화한 패턴상위 클래스에서 인터페이스만 정의하고 실제 생성은 서브 클래스가 담당함가상 생성자(Virtual Constructor) 패턴이라고도 함
프로토타입 (Prototype)	<ul style="list-style-type: none">원본 객체를 복제하는 방법으로 객체를 생성하는 패턴일반적인 방법으로 객체를 생성하여, 비용이 큰 경우 주로 이용함
싱글톤 (Singleton)	<ul style="list-style-type: none">하나의 객체를 생성하면 생성된 객체를 어디서든 참조할 수 있지만, 여러 프로세스가 동시에 참조할 수는 없음클래스 내에서 인스턴스가 하나뿐임을 보장하며, 불필요한 메모리 낭비를 최소화 할 수 있음

- 추상 팩토리(Abstract Factory)
- 빌더(Builder)
- 팩토리 메소드(Factory Method)
- 프로토타입(Prototype)
- 싱글톤(Singleton)

럼바우 객체지향 분석 기법

럼바우(Rumbaugh) 객체지향 분석 기법에서 동적 모델링에 활용되는 다이어그램은?

① 객체 다이어그램(Object Diagram)	맞추다 내 선택률 23%
② 패키지 다이어그램(Package Diagram)	맞추다 내 선택률 5%
③ 자료 흐름도(Data Flow Diagram)	맞추다 내 선택률 50%
④ 상태 다이어그램(State Diagram)	맞추다 내 선택률 21%

핵심 053

럼바우(Rumbaugh)의 분석 기법

2402403

럼바우의 분석 기법은 모든 소프트웨어 구성 요소를 그래픽 표기법을 이용하여 모델링하는 기법으로, 객체 모델링 기법(OMT, Object-Modeling Technique)이라고도 한다.

- 분석 활동은 '객체 모델링 → 동적 모델링 → 기능 모델링' 순으로 통해 이루어진다.

객체 모델링 (Object Modeling)	정보 모델링이라고도 하며, 시스템에서 요구되는 객체를 찾아내어 속성과 연관 식별 및 객체들 간의 관계를 규정하여 객체 다이어그램으로 표시하는 것
동적 모델링 (Dynamic Modeling)	상태 다이어그램(상태도)을 이용하여 시간의 흐름에 따른 객체들 간의 제어 흐름, 상호 작용, 동작 순서 등의 동적인 행위를 표현하는 모델링
기능 모델링 (Functional Modeling)	자료 흐름도(DFD)를 이용하여 다수의 프로세스들 간의 자료 흐름을 중심으로 처리 과정을 표현한 모델링

- 객체 모델링 : 객체 다이어그램으로 표시
- 동적 모델링 : 상태 다이어그램을 이용하여 표현
- 기능 모델링: 자료 흐름도(DFD)를 이용하여 표현

애자일 방법론

215, 209

핵심 005 애자일 모형(Agile Model)



애자일은 '민첩한', '기민한'이라는 의미로, 고객의 요구사항 변화에 유연하게 대응할 수 있도록 일정한 주기를 반복하면서 개발과정을 진행한다.

- 애자일 모형은 어느 특정 개발 방법론이 아니라 좋은 것을 빠르고 낭비 없게 만들기 위해 고객과의 소통에 초점을 맞춘 방법론을 통칭한다.
- 애자일 모형은 기업 활동 전반에 걸쳐 사용된다.
- 애자일 모형을 기반으로 하는 소프트웨어 개발 모형에는 스크럼(Scrum), XP(eXtreme Programming), 칸반(Kanban), Lean, 크리스탈(Crystal), ASD(Adaptive Software Development), 기능 중심 개발(FDD; Feature Driven Development), DSDM(Dynamic System Development Method), DAD(Disciplined Agile Delivery) 등이 있다.

- 스크럼
- XP(eXtream Programming)
- 기능 중심 개발(FDD)

요구사항 분석을 위한 CASE(자동화 도구)


- 요구사항을 자동으로 분석
- 요구사항 분석 명세서를 기술하도록 개발된 도구

CASE의 원천 기술

- 구조적 기법
- 프로토타이핑 기술
- 자동 프로그래밍 기술
- 정보 저장소 기능
- 분산처리 기술

XP(eXtreme Programming)

- 고객 참여, 개발 과정의 반복을 극대화
- 고객의 요구사항에 유연하게 대응


<div><div>핵심 010</div><div>XP의 주요 실천 방법 (Practice)</div></div> <div></div>	
Pair Programming (짝 프로그래밍)	다른 사람과 함께 프로그래밍을 수행함으로써 개발에 대한 책임을 공동으로 나눠 갖는 환경을 조성함
Collective Ownership (공동 코드 소유)	개발 코드에 대한 권한과 책임을 공동으로 소유함
Test-Driven Development (테스트 주도 개발)	<ul style="list-style-type: none">• 개발자가 실제 코드를 작성하기 전에 테스트 케이스를 먼저 작성하므로 자신이 무엇을 해야할지를 정확히 파악함• 테스트가 지속적으로 진행될 수 있도록 자동화된 테스트 도구(구조, 프레임워크)를 사용함
Whole Team (전체 팀)	개발에 참여하는 모든 구성원(고객 포함)들은 각자 자신의 역할이 있고 그 역할에 대한 책임을 가져야 함
Continuous Integration (계속적인 통합)	모듈 단위로 나눠서 개발된 코드들은 하나의 작업이 마무리될 때마다 지속적으로 통합됨
Design Improvement (디자인 개선) 또는 Refactoring(리팩토링)	프로그램 기능의 변경 없이, 단순화, 유연성 강화 등을 통해 시스템을 재구성함
Small Releases (소규모 릴리즈)	릴리즈 기간을 짧게 반복함으로써 고객의 요구 변화에 신속히 대응할 수 있음

2과목 : 소프트웨어 개발

소프트웨어 패키징 도구 활용 시 고려 사항

2019, 2018, 2016

핵심 100 패키징 시 고려사항



- 사용자의 시스템 환경, 즉 운영체제(OS), CPU, 메모리 등에 필요한 최소 환경을 정의한다.
- UI(User Interface)는 사용자가 눈으로 직접 확인할 수 있도록 시각적인 자료와 함께 제공하고 메뉴얼과 일치시켜 패키징한다.
- 소프트웨어는 단순히 패키징하여 배포하는 것으로 끝나는 것이 아니라 하드웨어와 함께 관리될 수 있도록 Managed Service 형태로 제공하는 것이 좋다.
- 사용자에게 배포되는 소프트웨어이므로 내부 콘텐츠에 대한 암호화 및 보안을 고려한다.
- 다른 여러 콘텐츠 및 단말기 간 DRM(디지털 저작권 관리) 연동을 고려한다.
- 사용자의 편의성을 위한 복잡성 및 비효율성 문제를 고려한다.
- 제품 소프트웨어 종류에 적합한 암호화 알고리즘을 적용한다.

소스코드 품질분석 도구 중 정적분석 도구의 종류

소스코드 품질분석 도구 중 정적분석 도구가 아닌 것은?

1	pmd	맞추다 내 선지 선택률	15%
2	cppcheck	맞추다 내 선지 선택률	10%
3	valMeter	맞추다 내 선지 선택률	62%
4	checkstyle	맞추다 내 선지 선택률	13%

정적 분석 도구 - 실행하지 않고 스타일,결함 등을 확인하는 도구

- **pmd**
- **cppcheck**
- SonarQube
- **checkstyle**
- ccm
- cobertura

테스트 케이스에 일반적으로 포함되는 항목

테스트 케이스에 일반적으로 포함되는 항목이 아닌 것은?

1	테스트 조건	맞추다 내 선지 선택률	1%
2	테스트 데이터	맞추다 내 선지 선택률	3%
3	테스트 비용	맞추다 내 선지 선택률	86%
4	예상 결과	맞추다 내 선지 선택률	11%

- 입력 데이터
- 테스트 조건
- 예상 결과

소프트웨어 패키징

소프트웨어 패키징에 대한 설명으로 틀린 것은?	
1 패키징은 개발자 중심으로 진행된다.	맞추다 내 선택 항목 93%
2 신규 및 변경 개념소스를 식별하고, 이를 모듈화하여 상품제품으로 패키징 한다.	맞추다 내 선택 항목 1%
3 고객의 편의성을 위해 메뉴얼 및 버전관리를 지속적으로 한다.	맞추다 내 선택 항목 3%
4 범용 환경에서 사용이 가능하도록 일반적인 배포 형태로 패키징이 진행된다.	맞추다 내 선택 항목 3%

- 모듈별로 생성한 실행 파일들을 묶어 배포용 설치 파일을 만드는 것
- **개발자가 아니라 사용자 중심으로 진행**
- 소스 코드는 향후 관리를 고려하여 모듈화하여 패키징
- 다양한 환경에서 손쉽게 사용할 수 있도록 일반적인 배포 형태로 패키징

블랙박스 테스트의 유형

테스트를 목적에 따라 분류했을 때,강도(Stress) 테스트에 대한 설명으로 옳은것은?

- 1

시스템에 고의로 실패를 유도하고 시스템이정상적으로 복귀하는지 테스트한다.

맞추다 내 선지 선택률 3%
- 2

시스템에 과다 정보량을 부과하여 과부하 시에도 시스템이 정상적으로 작동되는지를 테스트한다.

맞추다 내 선지 선택률 78%
- 3

사용자의 이벤트에 시스템이 응답하는 시간,특정 시간 내에 처리하는 업무량, 사용자 요구에 시스템이 반응하는 속도 등을 테스트한다.

맞추다 내 선지 선택률 11%
- 4

부당하고 불법적인 침입을 시도하여 보안시스템이 불법적인 침투를 잘 막아내는지 테스트한다.

맞추다 내 선지 선택률 8%

- 블랙박스 테스트는 소프트웨어가 수행할 특정 기능을 알기 위해 완전히 작동되는 것을 입증하는 테스트이다.(기능 테스트 라고도 한다)

블랙박스 테스트의 종류

동치 분할 검사 (Equivalence Partitioning Testing, 동치 클래스 분해)	<ul style="list-style-type: none">입력 자료에 초점을 맞춰 테스트 케이스(동치 클래스)를 만들고 검사하는 방법으로 동등 분할 기법이라고도 함프로그램의 입력 조건에 타당한 입력 자료와 타당하지 않은 입력 자료의 개수를 균등하게 하여 테스트 케이스를 정하고, 해당 입력 자료에 맞는 결과가 출력되는지 확인하는 기법
경계값 분석 (Boundary Value Analysis)	<ul style="list-style-type: none">입력 자료에만 치중한 동치 분할 기법을 보완하기 위한 기법입력 조건의 중간값보다 경계값에서 오류가 발생할 확률이 높다는 점을 이용하여 입력 조건의 경계값을 테스트 케이스로 선정하여 검사하는 기법
원인-효과 그래프 검사 (Cause-Effect Graphing Testing)	입력 데이터 간의 관계와 출력에 영향을 미치는 상황을 체계적으로 분석한 다음 효율성이 높은 테스트 케이스를 선정하여 검사하는 기법
오류 예측 검사 (Error Guessing)	<ul style="list-style-type: none">과거의 경험이나 확인자의 감각으로 테스트하는 기법다른 블랙 박스 테스트 기법으로는 찾아낼 수 없는 오류를 찾아내는 일련의 보충적 검사 기법이며, 데이터 확인 검사라고도 함
비교 검사 (Comparison Testing)	여러 버전의 프로그램에 동일한 테스트 자료를 제공하여 동일한 결과가 출력되는지 테스트하는 기법

목적에 따른 테스트

테스트를 목적에 따라 분류했을 때 강도(Stress) 테스트에 대한 설명으로 옳은것은?	
① 시스템에 고의로 실패를 유도하고 시스템이정상적으로 복구하는지 테스트한다.	맞추다 내 선택지를 3%
② 시스템에 과다 정보량을 부과하여 과부하 시에도 시스템이 정상적으로 작동되는지를 테스트한다.	맞추다 내 선택지를 78%
③ 사용자의 이벤트에 시스템이 응답하는 시간,특정 시간 내에 처리하는 업무량, 사용자 요구에 시스템이 반응하는 속도 등을 테스트한다.	맞추다 내 선택지를 11%
④ 부당하고 불법적인 접근을 시도하여 보안시스템이 불법적인 침투를 잘 막아내는지 테스트한다.	맞추다 내 선택지를 8%

회복 (Recovery) 테스트	시스템에 여러 가지 결함을 주어 실패하도록 한 후 올바르게 복구되는지를 확인하는 테스트
안전(Security) 테스트	시스템에 설치된 시스템 보호 도구가 불법적인 침입으로부터 시스템을 보호할 수 있는지를 확인하는 테스트
강도(Stress) 테스트	시스템에 과도한 정보량이나 빈도 등을 부과하여 과부하 시에도 소프트웨어가 정상적으로 실행되는지를 확인하는 테스트
성능 (Performance) 테스트	소프트웨어의 실시간 성능이나 전체적인 효율성을 진단하는 테스트로, 소프트웨어의 응답 시간, 처리량 등을 테스트
구조(Structure) 테스트	소프트웨어 내부의 논리적인 경로, 소스 코드의 복잡도 등을 평가하는 테스트
회귀 (Regression) 테스트	소프트웨어의 변경 또는 수정된 코드에 새로운 결함이 없음을 확인하는 테스트
병행(Parallel) 테스트	변경된 소프트웨어와 기존 소프트웨어에 동일한 데이터를 입력하여 결과를 비교하는 테스트

- 강도 테스트: 시스템에 과도한 정보량이나 빈도를 부과하여 과부하 시에도 정상적으로 실행하는지 확인

인터페이스 구현 검증 도구

- xUnit
- STAF
- FitNesse
- NTAF
- Selenium
- watir

3과목: 데이터베이스 구축

반정규화

- 반정규화란 시스템의 성능 향상, 개발 및 운영의 편의성 등을 위해 정규화된 데이터 모델을 통합, 중복, 분리하는 과정.
→ 의도적으로 정규화 원칙을 위배하는 행위
- 반정규화 방법

테이블 통합	<ul style="list-style-type: none"> 두 개의 테이블이 조인(Join)되는 경우가 많아 하나의 테이블로 합쳐 사용하는 것이 성능 향상에 도움이 될 경우 수행함 두 개의 테이블에서 발생하는 프로세스가 동일하게 자주 처리되는 경우, 두 개의 테이블을 이용하여 항상 조화를 수행하는 경우 테이블 통합을 고려함
테이블 분할	<ul style="list-style-type: none"> 테이블을 수직 또는 수평으로 분할하는 것 수평 분할(Horizontal Partitioning) : 레코드(Record)를 기준으로 테이블을 분할하는 것 수직 분할(Vertical Partitioning) : 하나의 테이블에 속성이 너무 많을 경우 속성을 기준으로 테이블을 분할하는 것
중복 테이블 추가	<ul style="list-style-type: none"> 여러 테이블에서 데이터를 추출해서 사용해야 하거나 다른 서버에 저장된 테이블을 이용해야 하는 경우 중복 테이블을 추가하여 작업의 효율성을 향상시킬 수 있음 중복 테이블 추가 방법 : 집계 테이블의 추가, 진행 테이블의 추가, 특정 부분만을 포함하는 테이블의 추가
중복 속성 추가	조인해서 데이터를 처리할 때 데이터를 조회하는 경로를 단축하기 위해 자주 사용하는 속성을 하나 더 추가하는 것

시스템 카탈로그

시스템 카탈로그에 대한 설명으로 옳지 않은 것은?

- 1 사용자 직접 시스템 카탈로그의 내용을 갱신하여 데이터베이스 무결성을 유지한다. 맞추다 내 선택 비율 70%
- 2 시스템 자신이 필요로 하는 스키마 및 메타데이터에 관한 정보를 포함하고 있는 시스템 데이터베이스이다. 맞추다 내 선택 비율 5%
- 3 시스템 카탈로그에 저장되는 내용을 메타데이터라고 한다. 맞추다 내 선택 비율 4%
- 4 시스템 카탈로그는 DBMS가 스스로 생성하고 유지한다. 맞추다 내 선택 비율 21%

- 시스템에 관련이 있는 다양한 객체에 관한 정보를 포함하는 시스템 데이터베이스
- 시스템 카탈로그에 저장된 정보를 메타 데이터라고 한다.
- 카탈로그는 DBMS가 직접 생성하고 유지한다.
- 시스템 테이블로 구성되어 있어 일반 사용자도 SQL을 이용하여 검색이 가능하다.
- INSERT, DELETE, UPDATE문으로 카탈로그를 갱신하는 것은 허용되지 않는다.

회복 기법

데이터베이스 로그(log)를 필요로 하는 회복 기법은?

①	즉각 갱신 기법	맞추다 내 선택지 선택률 43%
②	대수적 코딩 방법	맞추다 내 선택지 선택률 6%
③	타임 스탬프 기법	맞추다 내 선택지 선택률 38%
④	롤딩 기법	맞추다 내 선택지 선택률 13%

- 데이터베이스가 손상되었을 때 손상되기 이전의 정상 상태로 복구하는 작업

회복 기법의 종류

- 연기 갱신 기법
 - log 필요로 함
 - Redo 작업만 가능함
- 즉각 갱신 기법
 - log 필요로 함
 - Redo, Undo 모두 사용 가능함
- 그림자 페이지 대체 기법
 - log, Undo, Redo 알고리즘 필요 X
- 검사적 기법
 - 로그에 보관해 두고 장애 발생시 검사점부터 회복 작업

분산 데이터베이스의 목표

분산 데이터베이스 목표 중 "데이터베이스의 분산된 물리적 환경에서 특정 지역의 컴퓨터 시스템이나 네트워크에 장애가 발생해도 데이터 무결성이 보장된다"는 것과 관계있는 것은?

1 장애 투명성	맞추다 내 선지 선택률 73%
2 병행 투명성	맞추다 내 선지 선택률 11%
3 위치 투명성	맞추다 내 선지 선택률 14%
4 중복 투명성	맞추다 내 선지 선택률 3%

- 위치 투명성(Location Transparency) : 액세스하려는 데이터베이스의 실제 위치를 알 필요 없이 단지 데이터베이스의 논리적인 명칭만으로 액세스할 수 있음
- 중복 투명성(Replication Transparency) : 동일 데이터가 여러 곳에 중복되어 있더라도 사용자는 마치 하나의 데이터만 존재하는 것처럼 사용하고, 시스템은 자동으로 여러 자료에 대한 작업을 수행함
- 병행 투명성(Concurrency Transparency) : 분산 데이터베이스와 관련된 다수의 트랜잭션들이 동시에 실행되더라도 그 트랜잭션의 결과는 영향을 받지 않음
- 장애 투명성(Failure Transparency) : 트랜잭션, DBMS, 네트워크, 컴퓨터 장애에도 불구하고 트랜잭션을 정확하게 처리함

관계 대수

- 관계형 DB에서 원하는 정보를 검색하기 위해 어떻게 유도 하는가를 기술하는 **절차적인 언어**
- 릴레이션을 처리하기위해 연산자와 연산 규칙을 제공하는 언어로 **피연산자가 릴레이션이고, 결과도 릴레이션이다.**
- 일반 집합 연산자와 순수 관계 연산으로 구분된다.

- 순수 관계 연산자
 - Select
 - Project
 - Join
 - Division
- 일반 집합 연산자
 - UNION(합집합)
 - INTERSECTION(교집합)
 - DIFFERENCE(차집합)
 - CARTESIAN PRODUCT(교차곱)

- 질의에 대한 해를 구하기 위해 수행해야 할 **연산의 순서를 명시한다.**

관계 대수에 대한 설명으로 틀린 것은?

- 1

원하는 릴레이션을 정의하는 방법을 제공하며 비절차적 언어이다.

맞추다 내 선택지 선택률 63%
- 2

릴레이션 조작을 위한 연산의 집합으로 피연산자와 결과가 모두 릴레이션이다.

맞추다 내 선택지 선택률 15%
- 3

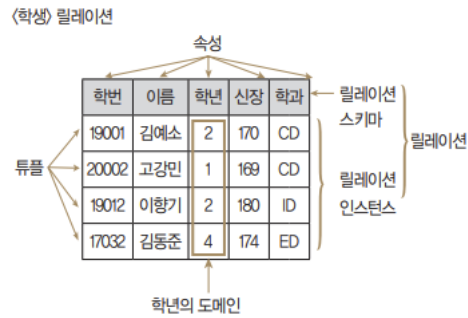
일반 집합 연산과 순수 관계 연산으로 구분된다.

맞추다 내 선택지 선택률 6%
- 4

질의에 대한 해를 구하기 위해 수행해야 할 연산의 순서를 명시한다.

맞추다 내 선택지 선택률 16%

관계형 데이터베이스의 Relation 구조



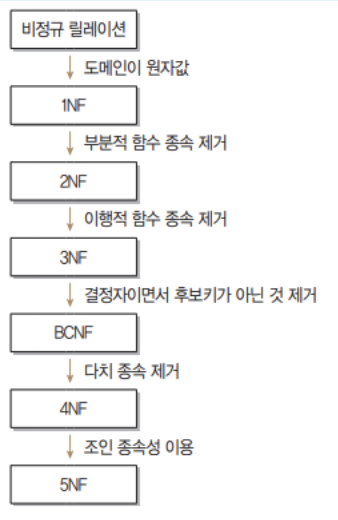
- 튜플 : 릴레이션을 구성하는 각각의 행
 - 튜플의 수를 **카디널리티(Cardinality)**라고 한다.
- 속성
 - 속성의 수를 **Degree** 라고 한다.
- 도메인
 - 하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자값들의 집합
 - ex) 성별의 애트리뷰트의 도메인은 '남' 과 '여'

무결성(Integrity)

- 데이터베이스에 저장된 값과 그것이 표현하는 실제값이 일치하는 정확성

- 개체 무결성(Entity Integrity, 실제 무결성) : 기본 테이블의 기본키를 구성하는 어떤 속성도 Null 값이나 중복값을 가질 수 없다는 규정
- 도메인 무결성(Domain Integrity, 영역 무결성) : 주어진 속성 값이 정의된 도메인에 속한 값이어야 한다는 규정
- 참조 무결성(Referential Integrity) : 외래키 값은 Null이거나 참조 릴레이션의 기본키 값과 동일해야 한다. 즉 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다는 규정
- 사용자 정의 무결성(User-Defined Integrity) : 속성 값들이 사용자가 정의한 제약조건에 만족해야 한다는 규정

정규화 과정



정규화 단계 알기 요령

두부를 좋아하는 정규화가 두부가게에 가서 가게에 있는 두부를 다 달라고 말하니 주인이 깜짝 놀라며 말했다.

두부이걸다줘? ≈ 도부이걸다조

- 도메인이 원자값
- 부분적 함수 종속 제거
- 이행적 함수 종속 제거
- 결정자이면서 후보키가 아닌 것 제거
- 다치 종속 제거
- 조인 종속성 이용

어떤 릴레이션 R의 모든 **조인 종속성**의 만족이 R의 후보 키를 통해서만 만족될 때, 이 릴레이션 R이 해당하는 정규형은?

- | | |
|---------|---------------|
| 1 제5정규형 | 맞추다 내 선택률 60% |
| 2 제4정규형 | 맞추다 내 선택률 11% |
| 3 제3정규형 | 맞추다 내 선택률 20% |
| 4 제1정규형 | 맞추다 내 선택률 9% |

• 조인 종속성 → 5NF

릴레이션 R의 모든 결정자(determinant)가 후보키이면 그 릴레이션 R은 어떤 정규형에 속하는가?

1 제 1 정규형	맞추다 내 선택률 13%
2 제 2 정규형	맞추다 내 선택률 13%
3 보이스/코드 정규형	맞추다 내 선택률 63%
4 제 4 정규형	맞추다 내 선택률 11%

- 결정자 → BCNF

SQL의 기능에 따른 분류

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

① DDL(Data Definition Language)	맞추다 내 선택 비율 9%
② DML(Data Manipulation Language)	맞추다 내 선택 비율 12%
③ DCL(Data Control Language)	맞추다 내 선택 비율 76%
④ DUL(Data User Language)	맞추다 내 선택 비율 3%

- DDL(Data Define Language)
 - SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의하거나 변경 또는 삭제할 때 사용하는 언어
 - **CREATE, ALTER, DROP**
- DML(Data Manipulation Language)
 - 저장된 데이터를 실질적으로 처리하는데 사용되는 언어
 - **SELECT, INSERT, DELETE, UPDATE**
- DCL(Data Control Language)
 - 데이터의 보안, 무결성, 회복, 병행 수행 제어 등 정의
 - **COMMIT, ROLLBACK, GRANT, REVOKE**

4과목 : 프로그래밍 언어 활용

결합도(Coupling)

- 모듈 간에 상호 의존하는 정도 또는 두 모듈 사이의 연관 관계

- 다양한 결합으로 모듈을 구성할 수 있으나 결합도가 약할수록 품질이 높고, 강할수록 품질이 낮다.
- 결합도가 강하면 시스템 구현 및 유지보수 작업이 어렵다.
- 결합도의 종류에는 자료 결합도, 스템프 결합도, 제어 결합도, 외부 결합도, 공통 결합도, 내용 결합도가 있으며 결합도가 약함에서 강함순으로 정리하면 다음과 같다.

모듈의 독립성을 높이기 위한 결합도(Coupling)와 관련한 설명으로 틀린 것은?

- ① 오류가 발생했을 때 전파되어 다른 오류의 원인이 되는 파문 효과(Ripple Effect)를 최소화해야 한다.

맞추다 내 선택률 3%

- ② 인터페이스가 정확히 설정되어 있지 않을 경우 불필요한 인터페이스가 나타나 모듈 사이의 의존도는 높아지고 결합도가 증가한다.

맞추다 내 선택률 22%

- ③ 모듈들이 변수를 공유하여 사용하게 하거나 제어 정보를 교류하게 함으로써 결합도를 낮추어야 한다.

맞추다 내 선택률 58%

- ④ 다른 모듈과 데이터 교류가 필요한 경우 전역변수(Global Variable)보다는 매개변수(Parameter)를 사용하는 것이 결합도를 낮추는 데 도움이 된다.

맞추다 내 선택률 18%

- 결합도가 낮을수록 독립성이 높아져 품질이 좋다.

OSI 7 layers

physical layer
data link layer
network layer
transport layer
session layer
presentation layer
application layer

- 각 계층의 주요 프로토콜에 대하여 알아야 한다.

응용 계층의 주요 프로토콜

FTP (File Transfer Protocol)	컴퓨터와 컴퓨터 또는 컴퓨터와 인터넷 사이에서 파일을 주고받을 수 있도록 하는 원격 파일 전송 프로토콜
SMTP(Simple Mail Transfer Protocol)	전자 우편을 교환하는 서비스
TELNET	<ul style="list-style-type: none"> • 멀리 떨어져 있는 컴퓨터에 접속하여 자신의 컴퓨터처럼 사용할 수 있도록 해주는 서비스 • 프로그램을 실행하는 등 시스템 관리 작업을 할 수 있는 가상의 터미널(Virtual Terminal) 기능을 수행
SNMP(Simple Network Management Protocol)	TCP/IP의 네트워크 관리 프로토콜로, 라우터나 허브 등 네트워크 기기의 네트워크 정보를 네트워크 관리 시스템에 보내는 데 사용되는 표준 통신 규약
DNS (Domain Name System)	도메인 이름을 IP 주소로 매핑(Mapping)하는 시스템
HTTP(HyperText Transfer Protocol)	월드 와이드 웹(WWW)에서 HTML 문서를 송수신 하기 위한 표준 프로토콜

전송 계층의 주요 프로토콜

TCP(Transmission Control Protocol)	<ul style="list-style-type: none"> • 양방향 연결(Full Duplex Connection)형 서비스를 제공함 • 스트림 위주의 전달(패킷 단위를 함 • 신뢰성 있는 경로를 확립하고 메시지 전송을 감독함 • 순서 제어, 오류 제어, 흐름 제어 기능을 함 • TCP 프로토콜의 헤더는 기본적으로 20Byte에서 60Byte까지 사용할 수 있는데, 선택적으로 40Byte를 더 추가할 수 있으므로 최대 100Byte까지 크기를 확장할 수 있음
UDP (User Datagram Protocol)	<ul style="list-style-type: none"> • 데이터 전송 전에 연결을 설정하지 않는 비연결형 서비스를 제공함 • TCP에 비해 상대적으로 단순한 헤더 구조를 가지므로, 오버헤드가 적고, 흐름제어나 순서 제어가 없어 전송 속도가 빠름 • 실시간 전송에 유리하며, 신뢰성보다는 속도가 중요시되는 네트워크에서 사용됨
RTCP(Real-Time Control Protocol)	<ul style="list-style-type: none"> • RTP(Real-time Transport Protocol) 패킷의 전송 품질을 제어하기 위한 제어 프로토콜 • 세션(Session)에 참여한 각 참여자들에게 주기적으로 제어 정보를 전송함

인터넷 계층의 주요 프로토콜

IP: 비연결형인 데이터그램 방식 사용

ARP: IP주소를 MAC 주소로 변환

TCP/IP 네트워크에서 IP 주소를 MAC 주소로 변환하는 프로토콜은?

- 1

UDP

맞추다 내 선택률 6%
- 2

ARP

맞추다 내 선택률 63%
- 3

TCP

맞추다 내 선택률 17%
- 4

ICMP

맞추다 내 선택률 14%

IP(Internet Protocol)	<ul style="list-style-type: none">전송할 데이터에 주소를 지정하고, 경로를 설정하는 기능을 함비연결형인 데이터그램 방식을 사용하는 것으로 신뢰성이 보장되지 않음
ICMP (Internet Control Message Protocol, 인터넷 제어 메시지 프로토콜)	IP와 조합하여 통신중에 발생하는 오류의 처리와 전송 경로 변경 등을 위한 제어 메시지를 관리하는 역할을 하며, 헤더는 8Byte로 구성됨
IGMP(Internet Group Management Protocol, 인터넷 그룹 관리 프로토콜)	멀티캐스트를 지원하는 호스트나 라우터 사이에서 멀티캐스트 그룹 유지를 위해 사용됨
ARP (Address Resolution Protocol, 주소 분석 프로토콜)	호스트의 IP 주소를 호스트와 연결된 네트워크 접속 장치의 물리적 주소(MAC Address)로 바꿈
RARP(Reverse Address Resolution Protocol)	ARP와 반대로 물리적 주소를 IP 주소로 변환하는 기능을 함

주요 스케줄링 알고리즘

- FCFS(First Come First Service)
 - 준비상태 큐에 도착한 순서에 따라 차례로 CPU를 할당하는 기법
- SJF(Shortest Job First)
 - 실행 시간이 가장 짧은 프로세스에게 먼저 CPU를 할당하는 기법
- HRN(Highest Response-ratio Next)
 - 우선순위 계산 공식을 이용하여 우선순위를 주어 CPU를 할당한다.
 - 계산하여 숫자가 가장 높은 것부터 낮은 순으로 우선순위가 부여된다.

• 우선순위 계산식

$$\text{우선순위 계산식} = \frac{\text{대기 시간} + \text{서비스 시간}}{\text{서비스 시간}}$$

HRN 방식으로 스케줄링 할 경우, 입력된 작업이 다음과 같을 때 처리되는 작업 순서로 옳은 것은?

작업	대기시간	서비스(실행)시간
A	5	20
B	40	20
C	15	45
D	20	2

① A→B→C→D

맞추다 내 선택률 5%

② A→C→B→D

맞추다 내 선택률 14%

③ D→B→C→A

맞추다 내 선택률 56%

④ D→A→B→C

맞추다 내 선택률 25%

- A = 25 / 20
- B = 60 / 20
- C = 60 / 45
- D = 22 / 2

숫자가 큰 순서대로 D → B → C → A

프레임워크의 특성

모듈화 (Modularity)	프레임워크는 캡슐화를 통해 모듈화를 강화하고 설계 및 구현의 변경에 따른 영향을 최소화함으로써 소프트웨어의 품질을 향상시킨다. 프레임워크는 개발표준에 의한 모듈화로 인해 유지보수가 용이함
재사용성 (Reusability)	프레임워크는 재사용 가능한 모듈들을 제공함으로써 예산 절감, 생산성 향상, 품질 보증이 가능함
확장성 (Extensibility)	프레임워크는 다형성(Polymorphism)을 통한 인터페이스 확장이 가능하여 다양한 형태와 기능을 가진 애플리케이션 개발이 가능함
제어의 역흐름 (Inversion of Control)	개발자가 관리하고 통제해야 하는 객체들의 제어를 프레임워크에 넘김으로써 생산성을 향상시킨다

페이지 교체 알고리즘 (Page fault 횟수 계산 문제)

- Page Fault가 발생했을 때 주기억장치의 모든 페이지 프레임이 사용중이면 어떤 페이지 프레임을 선택하여 교체할 것인지 결정하는 기법

OPT (OPTimal replacement, 최적 교체)	<ul style="list-style-type: none">• 앞으로 가장 오랫동안 사용하지 않을 페이지를 교체하는 기법• 벨레이디(Belady)가 제안한 것으로, 페이지 부재 횟수가 가장 적게 발생하는 가장 효율적인 알고리즘
FIFO(First In First Out)	<ul style="list-style-type: none">• 각 페이지가 주기억장치에 적재될 때마다 그때의 시간을 기억시켜 가장 먼저 들어와서 가장 오래 있었던 페이지를 교체하는 기법• 이해하기 쉽고, 프로그래밍 및 설계가 간단함
LRU(Least Recently Used)	<ul style="list-style-type: none">• 최근에 가장 오랫동안 사용하지 않은 페이지를 교체하는 기법• 각 페이지마다 계수기(Counter)나 스택(Stack)을 두어 현 시점에서 가장 오랫동안 사용하지 않은, 즉 가장 오래 전에 사용된 페이지를 교체함
LFU (Least Frequently Used)	<ul style="list-style-type: none">• 사용 빈도가 가장 적은 페이지를 교체하는 기법• 활발하게 사용되는 페이지는 사용 횟수가 많아 교체되지 않고 사용됨
SCR(Second Chance Replacement, 2차 기회 교체)	가장 오랫동안 주기억장치에 있던 페이지 중 자주 사용되는 페이지의 교체를 방지하기 위한 것으로, FIFO 기법의 단점을 보완하는 기법
NUR (Not Used Recently)	<ul style="list-style-type: none">• LRU와 비슷한 알고리즘으로, 최근에 사용하지 않은 페이지를 교체하는 기법• 최근에 사용되지 않은 페이지는 향후에도 사용되지 않을 가능성이 높다는 것을 전제로, LRU에서 나타나는 시간적인 오버헤드를 줄일 수 있음• 최근의 사용 여부를 확인하기 위해서 각 페이지마다 두 개의 비트, 즉 참조 비트(Reference Bit)와 변형 비트(Modified Bit, Dirty Bit)가 사용됨

다음의 페이지 참조 열(Page reference string)에 대해 페이지 교체 기법으로 선입선을 알고리즘을 사용할 경우 페이지 부재(Page Fault) 횟수는? (단, 할당된 페이지 프레임 수는 3이고, 처음에는 모든 프레임이 비어 있다.)

<페이지 참조 열>

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0

1	13	맞추다 내 선지 선택률 20%
2	14	맞추다 내 선지 선택률 55%
3	15	맞추다 내 선지 선택률 20%
4	20	맞추다 내 선지 선택률 5%

- 페이지 프레임 수가 3
- 표를 그려 계산
- 넣으려는 페이지가 이미 주기억 장치에 있다면 다시 적재할 필요가 없기 때문에 페이지 부재가 발생하지 않는다.

참조 페이지	7	0	1	2	0	3	0
주기억 장치	7	7	7	2	2	2	2
		0	0	0	0	3	3
			1	1	1	1	0
Page fault	0	0	0	0	x	0	0

- 세로로 FIFO 알고리즘에 따라 계산. 주기억장치에 없으면 page fault 발생 → 14회 발생
- 시험문제로 주로 FIFO가 나오지만, LRU가 나오는 경우도 있다.

- LRU인 경우에는 최근에 사용되지 않는 페이지를 교체해야 한다.

모듈이 되기위한 특징

- 다른 것들과 구분될 수 있는 독립적인 기능을 가진다
- 유일한 이름을 사용해야 한다
- 모듈 자체로서 재사용 될수 있다
- 독립적인 컴파일이 가능하다
- 다른 모듈에서 호출하여 사용할 수 있다.

스레드

- 프로세스 내의 작업 단위. 사용자 수준의 스레드와 커널 수준의 스레드로 나뉜다.

스레드의 분류

사용자 수준의 스레드	<ul style="list-style-type: none"> • 사용자가 만든 라이브러리를 사용하여 스레드를 운용함 • 속도는 빠르지만 구현이 어려움
커널 수준의 스레드	<ul style="list-style-type: none"> • 운영체제의 커널에 의해 스레드를 운용함 • 구현이 쉽지만 속도가 느림

- 사용자 스레드는 커널 내부에 존재하지만 커널의 통제권 안에 없을 뿐이다.

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.

맞추다 내 선택률 12%

- ② 동시에 여러 스레드가 커널에 접근할 수 있으므로 여러 스레드가 시스템 호출을 동시에 사용할 수 있다.

맞추다 내 선택률 16%

- ③ 각 스레드를 개별적으로 관리할 수 있으므로 스레드의 독립적인 스케줄링이 가능하다.

맞추다 내 선택률 20%

- ④ 커널 모드로의 전환 없이 스레드 교환이 가능하므로 오버헤드가 줄어든다.

맞추다 내 선택률 52%

- context switching이 발생하지 않아서 커널 스레드보다 오버헤드가 적다.

페이지 크기

페이지 크기가 작을 경우

- 페이지 단편화가 감소
- 기억장치 효율이 높아짐
- 디스크 접근 횟수가 많아져서 입출력 시간 증가
- 페이지 정보를 가지는 페이지 맵 테이블의 크기가 커지고, 매핑 속도가 늦어짐

페이지 크기가 클 경우

- 페이지 맵 테이블의 크기가 작아지고, 매핑 속도가 빨라짐
- 디스크 접근 횟수가 적어져서 입출력이 빨라짐
- 페이지 단편화가 증가
- 불필요한 내용까지 적재될 수 있음

교착상태(Dead Lock)

상호 배제에 나타나는 문제점. 둘 이상의 프로세스들이 자원을 점유한 상태에서 서로 다른 프로세스가 가지고 있는 자원을 요구하며 무한정 기다리는 현상

- 교착 상태 발생의 필요 충분 조건

상호 배제 (Mutual Exclusion)	한 번에 한 개의 프로세스만이 공유 자원을 사용할 수 있어야 함
점유와 대기 (Hold and Wait)	최소한 하나의 자원을 점유하고 있으면서 다른 프로세스에 할당되어 사용되고 있는 자원을 추가로 점유하기 위해 대기하는 프로세스가 있어야 함
비선점 (Non-preemption)	다른 프로세스에 할당된 자원은 사용이 끝날 때까지 강제로 빼앗을 수 없어야 함
환형 대기 (Circular Wait)	공유 자원과 공유 자원을 사용하기 위해 대기하는 프로세스들이 원형으로 구성되어 있어 자신에게 할당된 자원을 점유하면서 앞이나 뒤에 있는 프로세스의 자원을 요구해야 함

- 교착 상태의 해결 방법

예방 기법 (Prevention)	<ul style="list-style-type: none"> 교착상태가 발생하지 않도록 사전에 시스템을 제어하는 방법으로, 교착상태 발생의 네 가지 조건 중에서 어느 하나를 제거(부정)함으로써 수행됨 자원의 낭비가 가장 심한 기법
회피 기법 (Avoidance)	<ul style="list-style-type: none"> 교착상태가 발생할 가능성을 배제하지 않고 교착상태가 발생하면 적절히 피해나가는 방법으로, 주로 은행원 알고리즘(Banker's Algorithm)이 사용됨 은행원 알고리즘(Banker's Algorithm) : E. J. Dijkstra가 제안한 것으로, 은행에서 모든 고객의 요구가 충족되도록 현금을 할당하는 데서 유래한 기법
발견 기법 (Detection)	<ul style="list-style-type: none"> 시스템에 교착상태가 발생했는지 점검하여 교착상태에 있는 프로세스와 자원을 발견하는 것을 의미함 교착상태 발견 알고리즘과 자원 할당 그래프 등을 사용할 수 있음
회복 기법 (Recovery)	교착상태를 일으킨 프로세스를 종료하거나 교착상태의 프로세스에 할당된 자원을 선점하여 프로세스나 자원을 회복하는 것을 의미함

- Banker's Algorithm → 회피 기법.

5과목 : 정보시스템 구축관리

블루투스 관련 공격

블루버그 (BlueBug)	블루투스 장비 사이의 취약한 연결 관리를 악용한 공격으로, 휴대폰을 원격 조정하거나 통화를 감청할 수 있음
블루스나이프 (BlueSnarf)	블루투스의 취약점을 활용하여 장비의 파일에 접근하는 공격으로, 인증없이 간편하게 정보를 교환할 수 있는 OPP(Object Push Profile)를 사용하여 정보를 열람함
블루프린팅 (BluePrinting)	공격 대상이 될 블루투스 장비를 검색하는 활동
블루재킹 (BlueJacking)	블루투스를 이용해 스팸처럼 메시지를 익명으로 퍼뜨리는 공격

COCOMO 모델의 프로젝트 유형

- 조직형(Organic)
- 반분리형(Semi-detached)
- 내장형(Embedded)

보안 관련 용어

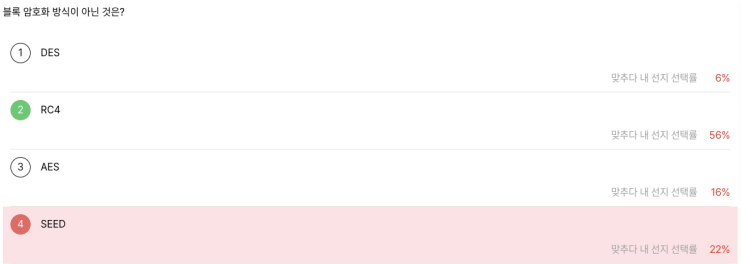
서비스형 블록체 인(BaaS; Blockchain as a Service)	<ul style="list-style-type: none"> • 블록체인(Blockchain) 앱의 개발 환경을 클라우드로 기반으로 제공하는 서비스 • 블록체인 네트워크에 노드의 추가 및 제거가 용이하고, 블록체인 플랫폼마다 다른 블록체인 기술을 보다 편리하게 사용할 수 있게 함
OWASP(the Open Web Application Security Project)	<ul style="list-style-type: none"> • 웹 정보 노출이나 악성 코드, 스크립트, 보안이 취약한 부분을 연구하는 비영리 단체 • 보안 취약점 중 보안에 미치는 영향이 큰 것을 기준으로 선정한 10대 웹 애플리케이션 취약점을 3~4년에 한 번씩 발표하고 있음
TCP 래퍼(TCP Wrapper)	<ul style="list-style-type: none"> • 외부 컴퓨터의 접속 인가 여부를 점검하여 접속을 허용 및 거부하는 보안용 도구 • 네트워크에 접속하면 로그인한 다른 컴퓨터 사용자의 ID 및 로그를 조회하여 악용이 가능한 데 이것을 방지하기 위한 방화벽 역할을 수행함
허니팟 (Honeypot)	<ul style="list-style-type: none"> • 비정상적인 접근을 탐지하기 위해 설치해 둔 시스템 • 침입자를 속여 실제 공격을 당하는 것처럼 보여 줌으로써 추적 및 공격기법에 대한 정보를 수집함
DPI(Deep Packet Inspection)	OSI 7 Layer 전 계층의 프로토콜과 패킷 내부의 콘텐츠를 파악하여 침입 시도, 해킹 등을 탐지하고, 트래픽을 조정하기 위한 패킷 분석 기술

- DPI: OSI 7 Layer 전 계층의 프로토콜과 패킷 내부의 콘텐츠를 파악하여 침입 시도, 해킹 등을 탐지하고 트래픽을 조정하기 위한 패킷 분석 기술

개인키 암호화 기법

- 동일한 키로 데이터를 암호화하고 복호화하는 기법.
- 블록 암호화 방식과 비트 단위로 암호화 하는 스트림 암호화 방식으로 나뉜다.

- 종류
 - 블록 암호화 방식 : DES, SEED, AES, ARIA
 - 스트림 암호화 방식 : LFSR, RC4



정보 보안 침해 공격 관련 용어

웜 (Worm)	네트워크를 통해 연속적으로 자신을 복제하여 시스템의 부하를 높임으로써 결국 시스템을 다운시키는 바이러스의 일종으로, 분산 서비스 거부 공격, 버퍼 오버플로 공격, 슬래머 등이 웜 공격의 한 형태
제로 데이 공격 (Zero Day Attack)	보안 취약점이 발견되었을 때 발견된 취약점의 존재 자체가 널리 공표되기도 전에 해당 취약점을 통하여 이루어지는 보안 공격으로, 공격의 신속성을 의미함
키로거 공격 (Key Logger Attack)	컴퓨터 사용자의 키보드 움직임을 탐지해 ID, 패스워드, 계좌번호, 카드번호 등과 같은 개인의 중요한 정보를 몰래 빼가는 해킹 공격
랜섬웨어 (Ransomware)	인터넷 사용자의 컴퓨터에 잠입해 내부 문서나 파일 등을 암호화해 사용자가 열지 못하게 하는 프로그램으로, 암호 해독용 프로그램의 전달을 조건으로 사용자에게 돈을 요구하기도 함
백도어 (Back Door, Trap Door)	<ul style="list-style-type: none">• 시스템 설계자가 서비스 기술자나 유지 보수 프로그램 작성자(Programmer)의 액세스 편의를 위해 시스템 보안을 제거하여 만들어놓은 비밀 통로로, 컴퓨터 범죄에 악용되기도 함• 백도어 탐지 방법 : 무결성 검사, 열린 포트 확인, 로그 분석, SetUID 파일 검사 등

리눅스의 커널 로그

다음 내용이 설명하는 로그 파일은?

- 리눅스 시스템에서 사용자의 성공한 로그인/로그아웃 정보기록

- 시스템의 종료/시작 시간 기록

- 1

tapping

맞추다 내 선택지 선택률 7%
- 2

xtslog

맞추다 내 선택지 선택률 27%
- 3

linuxer

맞추다 내 선택지 선택률 9%
- 4

wtmp

맞추다 내 선택지 선택률 56%

대문	파일명	내용
kernel	/dev/console	커널에 관련된 내용을 관리자에게 알리기 위해 파일로 저장하지 않고 지정된 장치에 표시함
	var/log/wtmp	• 성공한 로그인/로그아웃에 대한 로그를 기록함 • 시스템의 시작/종료 시간에 대한 로그를 기록함
	var/run/utmp	현재 로그인한 사용자의 상태에 대한 로그를 기록함
	var/log/btmp	실패한 로그인에 대한 로그를 기록함
	var/log/lastlog	마지막으로 성공한 로그인에 대한 로그를 기록함

소프트웨어 개발 방법론 테일러링

- 프로젝트의 상황 및 특성에 맞도록 정의된 소프트웨어 개발 방법론의 절차, 사용기법 등을 수정 및 보완하는 작업
- 테일러링 작업 시 고려해야 할 사항에는 내부적 기준과 외부적 기준이 있다.

내부적 기준	<ul style="list-style-type: none">• 목표 환경 : 시스템의 개발 환경과 유형이 서로 다른 경우 테일러링이 필요함• 요구사항 : 프로젝트의 생명 주기 활동에서 개발, 운영, 유지보수 등 프로젝트에서 우선적으로 고려할 요구사항이 서로 다른 경우 테일러링이 필요함• 프로젝트 규모 : 비용, 인력, 기간 등 프로젝트의 규모가 서로 다른 경우 테일러링이 필요함• 보유 기술 : 프로세스, 개발 방법론, 산출물, 구성원의 능력 등이 서로 다른 경우 테일러링이 필요함
외부적 기준	<ul style="list-style-type: none">• 법적 제약사항 : 프로젝트별로 적용될 IT Compliance가 서로 다른 경우 테일러링이 필요함• 표준 품질 기준 : 금융, 제도 등 분야별 표준 품질 기준이 서로 다른 경우 테일러링이 필요함

소프트웨어 개발 방법론의 테일러링(Tailoring)과 관련한 설명으로 틀린 것은?

- 1

프로젝트 수행 시 예상되는 변화를 배제하고 신속히 진행하여야 한다.

맞추다 내 선지 선택률 68%
- 2

프로젝트에 최적화된 개발 방법론을 적용하기 위해 절차, 산출물 등을 적절히 변경하는 활동이다.

맞추다 내 선지 선택률 9%
- 3

관리 측면에서의 목적 중 하나는 최단기간에 안정적인 프로젝트 진행을 위한 사전 위험을 식별하고 제거하는 것이다.

맞추다 내 선지 선택률 13%
- 4

기술적 측면에서의 목적 중 하나는 프로젝트에 최적화된 기술 요소를 도입하여 프로젝트 특성에 맞는 최적의 기법과 도구를 사용하는 것이다.

맞추다 내 선지 선택률 10%

테일러링은 변화를 배제하는게 아니라 수정 및 보완하는 작업이다

테일러링(Tailoring) 개발 방법론의 내부 기준에 해당하지 않는 것은?

- 1

납기/비용

맞추다 내 선지 선택률 11%
- 2

기술환경

맞추다 내 선지 선택률 4%
- 3

구성원 능력

맞추다 내 선지 선택률 26%
- 4

국제표준 품질기준

맞추다 내 선지 선택률 60%

품질기준은 외부적 기준에 해당함

스토리지 시스템 종류

- DAS, NAS

23.7, 22.3, 20.9


핵심 201 DAS (Direct Attached Storage)



DAS는 서버와 저장장치를 전용 케이블로 직접 연결하는 방식으로, 일반 가정에서 컴퓨터에 외장하드를 연결하는 것이 여기에 해당된다.

- 서버에서 저장장치를 관리한다.
- 저장장치를 직접 연결하므로 속도가 빠르고 설치 및 운영이 쉽다.
- 초기 구축 비용 및 유지보수 비용이 저렴하다.
- 직접 연결 방식이므로 다른 서버에서 접근할 수 없고 파일을 공유할 수 없다.
- 확장성 및 유연성이 상대적으로 떨어진다.
- 저장 데이터가 적고 공유가 필요 없는 환경에 적합하다.

핵심 202 NAS (Network Attached Storage)



- NAS는 서버와 저장장치를 네트워크를 통해 연결하는 방식이다.
- 별도의 파일 관리 기능이 있는 NAS Storage가 내장된 저장장치를 직접 관리한다.
- Ethernet 스위치를 통해 다른 서버에서도 스토리지에 접근할 수 있어 파일 공유가 가능하고, 장소에 구애받지 않고 저장장치에 쉽게 접근할 수 있다.
- DAS에 비해 확장성 및 유연성이 우수하다.
- 접속 증가 시 성능이 저하될 수 있다.

다음 내용이 설명하는 스토리지 시스템은?

- 하드디스크와 같은 데이터 저장장치를 호스트 버스 어댑터에 직접 연결하는 방식
 - 저장장치와 호스트 기기 사이에 네트워크 디바이스 없이 직접 연결하는 방식으로 구성

1	DAS	맞추다 내 선지 선택률	71%
2	NAS	맞추다 내 선지 선택률	17%
3	BSA	맞추다 내 선지 선택률	4%
4	NFC	맞추다 내 선지 선택률	8%

- DAS는 서버와 저장장치를 **케이블**로 직접 연결
- NAS는 서버와 저장장치를 **네트워크**를 통해 연결

DDoS 공격

- 여러 곳에 분산된 공격 지점에서 한 곳의 서버에 대해 분산 서비스 공격을 수행하는것.

Ping of Death (죽음의 핑)	Ping 명령을 전송할 때 패킷의 크기를 인터넷 프로토콜 허용 범위 이상으로 전송하여 공격 대상의 네트워크를 마비시키는 서비스 거부 공격 방법
SMURFING (스머핑)	IP나 ICMP의 특성을 악용하여 엄청난 양의 데이터를 한 사이트에 집중적으로 보냄으로써 네트워크를 불능 상태로 만드는 공격 방법
스마싱 (Smishing)	문자 메시지(SMS)를 이용해 사용자의 개인 신용 정보를 빼내는 공격 방법
피싱 (Phishing)	개인 정보(Private Data)와 낚시(Fishing)의 합성어로, 이메일이나 메신저 등을 통해 공공기관이나 금융기관을 사칭하여 개인 정보를 빼내는 기법
Ping Flood	특정 사이트에 매우 많은 ICMP 메시지를 보내 이에 대한 응답(Respond)으로 시스템 자원을 모두 사용하게 해 시스템이 정상적으로 동작하지 못하도록 하는 공격 방법
Evil Twin Attack	실제 존재하는 동일한 이름의 무선 Wifi 신호를 송출하여 로그인한 사람들의 계정 정보나 신용 정보 등을 빼내는 기법
스위치 재밍 (Switch Jamming)	위조된 매체 접근 제어(MAC) 주소를 지속적으로 네트워크로 흘려보내, 스위치 MAC 주소 테이블의 저장 기능을 혼란시켜 더미 허브(Dummy Hub)처럼 작동하게 하는 공격 방법

DoS(Denial of Service) 공격과 관련한 내용으로 틀린 것은?

1

Ping of Death 공격은 정상 크기보다 큰 ICMP 패킷을 작은 조각(Fragment)으로 쪼개어 공격 대상이 조각화 된 패킷을 처리하게 만드는 공격 방법이다.

맞추다 내 선택률 16%

2

Smurf 공격은 멀티캐스트(Multicast)를 활용하여 공격 대상이 네트워크의 임의의 시스템에 패킷을 보내게 만드는 공격이다.

맞추다 내 선택률 46%

3

SYN Flooding은 존재하지 않는 클라이언트가 서버별로 한정된 접속 가능 공간에 접속한 것처럼 속여 다른 사용자가 서비스를 이용하지 못하게 하는 것이다.

맞추다 내 선택률 25%

4

Land 공격은 패킷 전송 시 출발지 IP주소와 목적지 IP주소 값을 똑같이 만들어서 공격 대상에게 보내는 공격 방법이다.

맞추다 내 선택률 12%

- Smurf 공격은 엄청난 양의 데이터를 한 사이트에 집중적으로 보냄으로써 네트워크를 불능 상태로 만드는 공격 방법

소셜 네트워크에서 악의적인 사용자가 지인 또는 특정 유명인으로 활동하여 활동하는 공격 기법은? [문제오류 / 1번 정답처리]

1

Evil Twin Attack

맞추다 내 선택률 71%

2

Phishing

맞추다 내 선택률 20%

3

Logic Bomb

맞추다 내 선택률 1%

4

Cyberbullying 1

맞추다 내 선택률 9%

- 피싱: 개인 정보로 이메일이나 메신저 등을 이용해 공공기관 또는 금융기관을 사칭하여 개인 정보를 빼내는 기법
- Evil Twin Attack: 실제 존재하는 동일한 이름의 Wifi 신호를 송출하여 로그인한 사람들의 계정 정보나 신용 정보 등을 빼내는 기법

세션 하이재킹

- 서버에 접속하고 있는 클라이언트들의 세션 정보를 가로채는 공격 기법
- 정상적인 연결을 RST 패킷을 통해 종료시킨 후 재연결 시 공격자에게 연결
- 탐지 방법
 - 비동기화 상태 탐지
 - ACK Storm 탐지
 - 패킷의 유실 탐지
 - 예상치 못한 접속의 리셋 탐지