

# 1장. 프로그래밍 언어 활용

## 001. 데이터 타입

변수(Variavle)에 저장될 데이터의 형식으로 정수, 부동 소수점, 문자, 문자열, 불린, 배열 타입이 존재

## 002. 변수와 기억 클래스

변수 : 변할 수 있는 값으로 정수형, 실수형, 문자형, 포인트 형 등으로 구분

기억 클래스 : 기억영역을 결정하는 작업으로 C언어는 자동, 외부, 정적, 레지스터 변수를 제공

## 003. scanf() 함수 / printf() 함수

scanf("%3d", &a); : C언어의 표준 입력 함수	<p>%, 서식 문자임을 지정</p> <p>3, 입력 자릿수를 3자리로 지정</p> <p>d, 10진수로 입력</p> <p>&amp;a, 입력받은 데이터를 변수 a의 주소에 저장</p>
printf("%-8.2f", 200.2); : C언어의 표준 출력 함수	<p>%, 서식 문자임을 지정</p> <p>-, 왼쪽부터 출력</p> <p>8, 출력 자릿수를 8자리로 지정</p> <p>2, 소수점 이하를 2자리로 지정</p> <p>f, 실수로 출력</p>

## 004. Java에서의 표준 입/출력

입력 : Scanner 클래스를 이용해 키보드로 입력

출력 : System.out.printf(서식 문자열, 변수)

## 005. 연산자 우선순위

단항, 산술, 시프트, 관계, 비트 논리, 조건, 대입, 순서 연산자

## 006. 포인터

변수의 주소이며 배열을 포인터 변수에 저장한 후 포인터를 이용해 배열의 요소에 접근할 수 있음

## 007. Java의 클래스와 메소드

클래스는 객체 생성을 위한 필드(속성)와 메소드(함수)를 정의하는 설계도

## 008. Python의 기초

input() 함수 : 표준 입력 함수

print() 함수 : 표준 출력 함수

List : C, Java는 배열을 사용하나 Python은 List 사용

Range : 연속된 숫자를 생성

Slice : 문자열이나 리스트 객체에서 일부를 잘라 반환

## 009. 라이브러리 - C

stdio.h : 데이터 입/출력 기능 제공

math.h : 수학 함수 제공

string.h : 문자열 처리 기능 제공

stdlib.h : 자료형 변환, 난수 발생, 메모리 할당 기능 제공

time.h : 시간 처리 기능 제공

## 010. 라이브러리 - Java

**java.lang** : import문 없이 사용 가능

**java.util** : 날짜, 난수, 복잡한 문자열 처리 등의 기능 제공

**java.io** : 파일 입/출력 기능 제공

**java.net** : 네트워크에 관련된 기능 제공

**java.awt** : UI와 관련된 기능 제공

## 011. 라이브러리 - Python

**내장함수** : import문 없이 사용 가능

**os** : 운영체제와 상호 작용하기 위한 기능 제공

**re** : 고급 문자열 처리를 위한 기능 제공

**math** : 복잡한 수학 연산을 위한 기능 제공

**random** : 무작위 선택을 위한 기능 제공

**statistics** : 통계값 산출을 위한 기능 제공

**datetime** : 날짜와 시간 조작을 위한 기능 제공

## 012. 절차적 프로그래밍 언어

**C** : 자료의 주소를 조작할 수 있는 포인터 제공    **ALGOL** : 수치 계산이나 논리 연산을 위한 과학 기술 계산용 언어

**COBOL** : 사무 처리용 언어

**FORTAN** : 과학 기술 계산용 언어

## 013. 객체지향 프로그래밍 언어

**Java** : 운영체제 및 하드웨어에 독립적이며 이식성이 강함

**C++** : 모든 문제를 객체로 모델링하여 표현

**Smalltalk** : 최초로 GUI를 제공한 언어

## 014. 객체지향 프로그래밍 언어의 구성 요소

**객체(Object)** : 데이터(속성)와 이를 처리하기 위한 연산(메소드)을 결합시킨 실체

**클래스(Class)** : 두 개 이상의 유사한 객체들을 묶어서 하나의 공통된 특성을 표현하는 요소

**메시지(Message)** : 객체들 간에 상호작용을 하는데 사용되는 수단

## 015. 스크립트 언어

**HTML** 문서 안에 직접 프로그래밍 언어를 삽입하여 사용하는 것

기계어로 컴파일 되지 않고 별도의 번역기가 소스를 분석하여 동작

**서버용 스크립트 언어** : 서버에서 해석되어 실행된 후 결과만 클라이언트로 보내는 스크립트 언어

**클라이언트용 스크립트 언어** : 클라이언트의 웹 브라우저에서 해석되어 실행되는 스크립트 언어

## 016. 스크립트 언어의 종류

**자바 스크립트** : 웹 페이지의 동작을 제어하며 변수 선언이 필요

**ASP** : 서버 측에서 동작으로 수행되는 페이지를 만들기 위한 언어

**JSP** : 서버용 스크립트 언어로 Linux, Unix, Windows 운영체제에서 사용

**파이썬** : 객체지향 기능을 지원하는 대화형 인터프리터 언어

017. 선언형 언어

- 함수형 언어 : 수학적 함수를 조합하여 문제를 해결하는 언어
- 논리형 언어 : 기호 논리학에 기반을 둔 언어
- HTML : 인터넷의 표준 문서인 하이퍼텍스트 문서를 만들기 위해 사용하는 언어
- LISP : 인공지능 분야에 사용되는 언어
- PROLOG : 논리학을 기초로 한 고급 언어
- XML : 기존 HTML의 단점을 보완하여 웹에서 다양한 문서들을 상호 교환할 수 있도록 설계된 언어
- Haskell : 함수형 프로그래밍 언어로 부작용이 없고 코드가 간결하여 에러 발생이 낮음

018. 예외처리

하드웨어 문제, 운영체제의 설정 실수, 라이브러리 손상, 사용자의 입력 실수, 받아들일 수 없는 연산, 할당하지 못하는 기억장치 접근 등에 대한 문제를 프로그래머가 작성해놓은 처리 루틴을 통해 해결하도록 하는 것

2장. 요구사항 확인

019. 현행 시스템 파악 절차

시스템 구성 파악 → 시스템 기능 파악 → 시스템 인터페이스 파악 → 아키텍처 구성 파악 → 소프트웨어 구성 파악 → 하드웨어 구성 파악 → 네트워크 구성 파악

020. 개발 기술 환경

- |                                       |                     |
|---------------------------------------|---------------------|
| 운영체제(OS, Operating System)            | 데이터베이스 관리 시스템(DBMS) |
| 웹 애플리케이션(WAS, Web Application Server) | 오픈 소스(Open Source)  |

021. 요구사항 유형

기능 요구사항	비기능 요구사항	사용자 요구사항	시스템 요구사항
---------	----------	----------	----------

022. 요구사항 개발 프로세스

요구사항 도출	요구사항 분석	요구사항 명세	요구사항 확인
---------	---------	---------	---------

023. 요구사항 분석 기법

- |         |         |
|---------|---------|
| 요구사항 분류 | 정형 분석   |
| 요구사항 할당 | 요구사항 협상 |
- 개념 모델링
- 유스케이스 다이어그램, 데이터 흐름 모델, 상태 모델, 목표기반 모델, 사용자 인터랙션, 객체 모델, 데이터 모델

024. 자료 흐름도

프로세스	자료 흐름	자료 저장소	단말
------	-------	--------	----

025. 요구사항 확인 기법

요구사항 검토	프로토타이핑	모델 검증	인수 테스트
---------	--------	-------	--------

026. UML

- 사물
- 구조 사물, 행동 사물, 그룹 사물, 주해 사물
- 관계
- 연관 관계, 집합 관계, 포함 관계, 일반화 관계, 의존 관계, 실체화 관계
- 다이어그램
- 정적 모델링, 동적 모델링

## 027. 구조적 다이어그램

클래스 다이어그램 : 클래스, 제약조건, 관계  
컴포넌트 다이어그램  
복합체 구조 다이어그램

객체 다이어그램  
배치 다이어그램  
패키지 다이어그램

## 028. 시퀀스 다이어그램

액터      객체      라이프라인      활성화 상자      메시지      객체 소멸      프레임

## 029. 커뮤니케이션 다이어그램

액터      객체      라이프라인      활성화 상자      메시지      객체 소멸      프레임      링크

## 030. 상태 다이어그램

상태      이벤트      상태 전환      프레임

## 031. 유스케이스 다이어그램

시스템 범위      액터      유스케이스      관계

## 032. 활동 다이어그램

액션 / 액티비티      노드      스웜레인

# 3장. 데이터 입/출력 구현

## 033. 데이터 모델

개체      속성      관계

## 034. 데이터 모델의 종류

개념적 데이터 모델      논리적 데이터 모델      물리적 데이터 모델

## 035. 데이터 모델에 표시할 요소

구조      연산      제약 조건

## 036. 이상

일부 속성들의 종속으로 데이터의 중복이 발생      종류 : 삽입 이상, 삭제 이상, 갱신 이상

## 036. 함수적 종속

$X \rightarrow Y$  : X는 결정자, Y는 종속자      완전 함수적 종속      부분 함수적 종속

## 037. 정규화

비정규 릴레이션  $\rightarrow$  1NF(도메인이 원자값)  $\rightarrow$  2NF(부분적 함수 종속 제거)  $\rightarrow$  3NF(이행적 함수 종속 제거)  $\rightarrow$  BCNF(결정자이면서 후보키가 아닌 것 제거)  $\rightarrow$  4NF(다치 종속)  $\rightarrow$  5NF(조인 종속성 이용)

## 038. 논리 데이터 모델의 물리 데이터 모델로 변환

엔티티를 테이블로 변환      슈퍼타입 기준 테이블 변환  
서브타입 기준 테이블 변환      개별타입 기준 테이블 변환  
속성을 컬럼으로 변환      관계를 외래키로 변환

## 039. 반정규화

테이블 통합      테이블 분할      중복 테이블 추가      중복 속성 추가

040. 인덱스

트리 기반 인덱스      비트맵 인덱스      함수 기반 인덱스      비트맵 조인 인덱스      도메인 인덱스

041. 뷰

사용자에게 접근이 허용된 자료만을 제한적으로 보여주며, 존재하는 것처럼 간주 되는 가상 테이블

042. 트랜잭션

원자성(Atomicity)      일관성(Consistency)      독립성(Isolation)      지속성(Durability)

043. 클러스터

동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법  
클러스터링키로 지정된 컬럼 값의 순서대로 저장되고 여러 개의 테이블이 하나의 클러스터에 저장

044. 파티션

범위 분할      해시 분할      조합 분할

045. 데이터베이스 용량 설계

테이블의 테이블 스페이스와 인덱스의 테이블 스페이스를 분리하여 구성  
테이블 스페이스와 임시 테이블 스페이스를 분리하여 구성  
테이블을 마스터 테이블과 트랜잭션 테이블로 분류

4장. 통합 구현

046. 통합 구현

송신 시스템과 모듈    수신 시스템과 모듈    중계 시스템    연계 데이터    네트워크

047. 연계 요구사항 분석

시스템 구성도    코드 정의서    응용 애플리케이션 구성    테이블 정의서    연계 요구사항 분석서(정의서)

048. 연계 데이터 식별 및 표준화

연계 범위 및 항목 정의      연계 코드 변환 및 매핑  
연계 데이터 식별자와 변경 구분 추가      연계 데이터 표현 방법 정의  
연계 정의서 및 명세서 작성

049. 연계 매커니즘

송신 시스템 : 운영 DB로부터 인터페이스 테이블이나 파일 형식으로 연계 데이터를 생성하여 송신  
수신 시스템 : 송신 시스템으로부터 전송된 데이터를 수신 시스템에 맞는 데이터로 변환한 후 운영 DB에 반영  
송/수신 시스템 사이에 중계 시스템을 둘 수 있으며 직접 연계방식과 간접 연계방식으로 나뉨

050. 연계 매커니즘의 직접 연계 방식

장점	단점
연계 및 통합 구현이 단순하고 용이, 중간 매개체 없이 직접 연계되므로 데이터 연계 처리 성능이 뛰어나, 개발 비용이 저렴, 개발 기간이 짧음	송/수신 시스템 간 결합도가 높아 시스템 변경 시 오류가 발생할 수 있음, 보안을 위한 암호화 처리와 비즈니스 로직을 적용하기 어려움, 연계 및 통합이 가능한 시스템 환경이 제한적임
종류	
DB Link, API, DB Connection, JDBC	

051. 연계 매커니즘의 간접 연계 방식

장점

서로 다른 네트워크, 프로토콜 등 다양한 환경의 연계 및 통합이 가능, 송/수신 시스템 간 인터페이스가 변경되어도 오류 없이 서비스 가능, 보안 품질 보장과 비즈니스 처리를 위한 로직을 쉽게 반영할 수 있음

단점

연계 매커니즘과 아키텍처가 복잡함, 중간 매개체로 인해 성능 저하 가능, 개발 및 적용 기간이 비교적 김

종류

연계 솔루션, ESB, Scket, Web Service

052. 연계 매커니즘의 구성

연계 데이터 생성 및 추출 → 코드 매핑 및 데이터 변환 → 인터페이스 테이블 또는 파일 생성 → 로그(Log) 기록 → 연계 서버 또는 송/수신 어댑터 → 전송 → 운영 DB에 연계 데이터 반영

053. 연계 매커니즘 구간별 오류 발생 시점

송신 시스템	코드 매핑 및 데이터 변환 시, 인터페이스 테이블 또는 파일 등록 시
수신 시스템	연계 데이터 로드 시, 코드 매핑 및 데이터 변환 시, 운영 DB에 반영 시
연계 서버	연계 데이터 송/수신 시, 수신 시스템의 데이터 형식으로 변환 및 로드 시

054. 장애 및 오류의 확인과 처리 절차

오류 원인 및 발생 현황 확인, 불가능하다면 오류 로그 테이블 또는 파일로 원인 분석, 발생한 오류에 대한 원인이 확인되면 원인에 따른 적절한 대응

055. 장애 및 오류의 정의와 설계

장애 및 오류 관리 대상	관리 대상의 장애 및 오류 코드와 메시지
장애 및 오류 코드와 메시지 관리 방식	장애 및 오류 기록 방식

056. 연계 데이터 보안

전송 구간 보안

VPN

인터넷 등 통신 사업자의 공중 네트워크와 암호화 기술을 이용하여 사용자가 마치 자신의 전용 회선을 사용하는 것처럼 해주는 보안 솔루션

데이터 보안

연계 솔루션

EAI 서버와 송/수신 시스템에 설치되는 클라이언트를 이용하는 간접 연결 방식

057. 연계 모듈의 구현 환경

EAI : Point-to-Point, Hub & Spoke, Message Bus, Bus Hybrid    ESB    웹 서비스 : SOAP, UDD, WSDL

058. XML

웹 브라우저 간 HTML 문제와 SGML의 문제로 개발된 **다목적 마크업 언어**, **유니코드**를 기반으로 다국어 지원  
JSON, 비동기 처리에 사용되는 AJAX에서 XML을 대체하여 사용

059. 연계 테스트

연계 테스트 케이스 작성    연계 테스트 환경 구축    연계 테스트 수행    연계 테스트 수행 결과 검증

5장. 서버 프로그램 구현

060. 개발 환경 구축 - 하드웨어 환경

웹 서버    웹 애플리케이션 서버    데이터베이스 서버    파일 서버

## 061. 개발 환경 구축 - 소프트웨어 환경

요구사항 관리 도구      설계/모델링 도구      구현 도구      빌드 도구      테스트 도구      형상 관리 도구

## 062. 모듈

### 모듈화

모듈 간 결합도 최소화, 응집도 최대화가 목표

## 063. 결합도

결합도가 약할수록 품질이 높고, 강할수록 품질이 낮으며 시스템 구현 및 유지보수 작업이 어려움

자료 결합도, 스탬프(검인) 결합도, 제어 결합도, 외부 결합도, 공통(공유) 결합도, 내용 결합도

## 064. 응집도

응집도가 강할수록 품질이 높고, 강할수록 품질이 낮음

기능적 응집도, 순차적 응집도, 교환(통신)적 응집도, 절차적 응집도, 시간적 응집도, 논리적 응집도, 우연적 응집도

## 065. 팬인(Fan-In) / 팬아웃(Fan-Out)

### 팬인

어떤 모듈을 제어(호출)하는 모듈의 수

### 팬아웃

어떤 모듈에 의해 제어(호출)되는 모듈의 수

## 066. 공통 모듈

정확성

명확성

완전성

추적성

## 067. 재사용

재사용되는 대상은 외부 모듈과의 결합도는 낮고, 응집도는 높아야 함

재사용 규모에 따른 분류

함수와 객체, 컴포넌트, 애플리케이션

## 068. DBMS 접속 기술

JDBC

ODBC

MyBatis

## 069. SQL

### 동적 SQL(Dynamic SQL)

다양한 조건에 따라 SQL 구문을 동적으로 변경하여  
처리하는 방식

### 정적 SQL(Static SQL)

SQL 코드를 변수에 담지 않고 코드 사이에 직접  
기술하는 방식

## 070. 서버 개발

### 서버 개발 프레임워크

spring, Node.js, Django, Codeigniter, Ruby on Rails

### 프레임워크의 특징

모듈화, 재사용성, 확장성, 제어의 역흐름

## 071. 서버 개발 과정

DTO 구현

SQL 구현

DAO 구현

Service 구현

Controller 구현

## 072. 배치 프로그램

### 수행 주기에 따른 분류

정기 배치, 이벤트성 배치, On-Demand 배치

### 필수 요소

대용량 데이터, 자동화, 견고성, 안정성/신뢰성, 성능

### 배치 스케줄러

일괄 처리 작업이 설정된 주기에 맞춰 수행되도록 지원해주는 도구로 Spring Batch, Quartz, Cron이 있음

6장. 화면 설계

073. 사용자 인터페이스

CLI (Command Line Interface)	GUI (Graphical User Interface)	NUI (Natural User Interface)
------------------------------	--------------------------------	------------------------------

074. 사용자 인터페이스의 기본 원칙

직관성	유효성	학습성	유연성
-----	-----	-----	-----

075. 사용자 인터페이스의 설계 지침

사용자 중심 표준	일관성 접근성	단순성	결과 예측 가능 명확성	가시성 오류 발생 해결
--------------	------------	-----	-----------------	-----------------

076. UI 설계 도구

와이어프레임 (Wireframe)	목업 (Mockup)	스토리보드 (Story Board)	프로토타입 (Prototype)	유스케이스 (Use Case)
-----------------------	----------------	------------------------	----------------------	---------------------

077. UI 스타일 가이드 작성

구동 환경 정의	레이아웃 정의	네비게이션 정의	기능 정의	구성 요소 정의
----------	---------	----------	-------	----------

078. UI 요구사항 확인 순서

목표 정의	인터뷰 유의사항 가능하면 개별적으로 진행 / 한 시간 미만 / 반드시 사용자 리서치를 시작하기 전 실시			
활동 사항 정의	조사항 요구사항을 토대로 활동 사항 정의			
UI 요구사항 작성	요구사항 요소 확인 → 정황 시나리오 → 요구사항 작성			

079. UI 흐름 설계

기능 작성	입력 요소 확인	유스 케이스 설계	기능 및 양식 확인 텍스트 박스, 콤보 박스, 라디오 박스, 체크 박스
-------	----------	-----------	---

080. UI 상세 설계

요구사항 확인	UI 설계서 표지 및 개정 이력 작성	UI 구조 설계	메뉴 구조 설계	화면 설계
---------	----------------------	----------	----------	-------

7장. 애플리케이션 테스트

081. 애플리케이션 테스트

완벽한 테스트 불가능	결함 집중	살충제 패러독스	정황 의존	오류-부재의 궤변
-------------	-------	----------	-------	-----------

082. 애플리케이션 테스트의 분류

프로그램 실행 여부에 따른 테스트	정적 테스트, 동적 테스트
테스트 기반에 따른 테스트	명세 기반 테스트, 구조 기반 테스트, 경험 기반 테스트
시각에 따른 테스트	검증 테스트, 확인 테스트
목적에 따른 테스트	회복 테스트, 안전 테스트, 강도 테스트, 성능 테스트, 구조 테스트, 회귀 테스트, 병행 테스트

083. 화이트박스 테스트

테스트 종류	테스트 검증 기준
기초 경로 검사, 제어구조 검사	문장 검증 기준, 분기 검증 기준, 조건 검증 기준, 분기/조건 기준



084. 블랙박스 테스트

테스트 종류

동치분할 검사, 경계값 분석, 원인-효과 그래프 검사, 오류 예측 검사, 비교 검사

085. 개발 단계에 따른 애플리케이션 테스트

단위 테스트                      통합 테스트                      시스템 테스트                      인수 테스트

086. 통합 테스트

비점진적 통합 방식                      점진적 통합 방식                      회귀 테스트  
빅뱅 통합 테스트                      하향식 통합 테스트, 상향식 통합 테스트, 혼합식 통합 테스트

087. 애플리케이션 테스트 프로세스

테스트 계획 → 테스트 분석 및 디자인 → 테스트 케이스 및 시나리오 작성 → 테스트 수행 → 테스트 결과  
평가 및 리포팅 → 결함 추적 및 관리

088. 테스트 케이스

테스트 계획 검토 및 자료 확보 → 위험 평가 및 우선순위 결정 → 테스트 요구사항 정의 → 테스트 구조 설계  
및 테스트 방법 결정 → 테스트 케이스 정의 → 테스트 케이스 타당성 확인 및 유지 보수

089. 테스트 시나리오

테스트 케이스를 적용하는 순서에 따라 여러 테스트 케이스들을 묶은 집합

090. 테스트 오라클

참 오라클                      샘플링 오라클                      추정 오라클                      일관성 검사

091. 테스트 자동화 도구 유형

휴먼 에러                      정적 분석 도구                      테스트 실행 도구  
성능 테스트                      테스트 통제 도구                      테스트 하네스 도구

092. 결함 관리

프로세스 처리 순서                      결함 관리 계획 → 결함 기록 → 결함 검토 → 결함 수정 → 결함 재확인 → 결함 상태  
추적 및 모니터링 활동 → 최종 결함 분석 및 보고서 작성  
측정 지표                      결함 분포, 결함 추세, 결함 에이징  
결함 추적 순서                      결함 등록 → 결함 검토 → 결함 할당 → 결함 수정 → 결함 종료 → 결함 해제  
결함 분류                      시스템 결함, 기능 결함, GUI 결함, 문서 결함  
결함 심각도                      High, Medium, Low

093. 애플리케이션 성능

처리량                      응답 시간                      경과 시간                      자원 사용률

094. 소스 코드 최적화

클린 코드                      누구나 쉽게 이해하고 수정 및 추가할 수 있는 단순, 명료한 코드  
나쁜 코드                      프로그램의 로직이 복잡하고 이해하기 어려운 코드  
클린 코드 작성 원칙                      가독성, 단순성, 의존성 배제, 중복성 최소화, 추상화  
소스 코드 최적화 유형                      클래스 분할 배치, 느슨한 결합

8장. SQL 응용 - Skip

9장. 소프트웨어 개발 보안 구축

095. 소프트웨어 개발의 비용 산정 기법

하향식 비용 산정 기법	전문가 감정 기법, 델파이 기법
상향식 비용 산정 기법	LOC 기법, 개발 단계별 인월수 기법

096. 서비스 거부 공격

Ping of Death	Smufing	SYN Flooding
TearDrop	Land	DDoS

097. 네트워크 침해 공격 관련 용어

Smishing	Spear Phishing	APT
무작위 대입 공격	Qshing	SQL 삽입 공격

098. 정보 보안 침해 공격 관련 용어

좀비(Zombie) PC	C&C 서버	봇넷(Botnet)
웜(Worm)	제로 데이 공격(Zero Day Attack)	키로거 공격(Key Logger Attack)
랜섬웨어(Ransomware)	백도어(Back Door, Trap Door)	트로이 목마(Trojan Horse)

099. 소프트웨어 개발 보안

행정안전부	한국인터넷진흥원	발주기관	사업자	감리법인
-------	----------	------	-----	------

100. 소프트웨어 개발 보안 활동 관련 법령

개인정보 보호법	정보통신망 이용촉진 및 정보보호 등에 관한 법률
신용정보의 이용 및 보호에 관한 법률	위치정보의 보호 및 이용 등에 관한 법률
표준 개인정보 보호 지침	개인정보의 안전성 확보 조치 기준
개인정보 영향평가에 관한 고시	

101. Secure SDLC

요구사항 분석 단계	설계 단계	구현 단계	테스트 단계	유지 보수 단계
------------	-------	-------	--------	----------

10장. 소프트웨어 개발 보안 구축

102. 보안 요소

기밀성	무결성	가용성	인증	부인 방지
-----	-----	-----	----	-------

103. 보안 점검 - 세션 통제

불충분한 세션 관리	잘못된 세션에 의한 정보 노출
------------	------------------

104. 보안 점검 - 입력 데이터 검증 및 표현

SQL 삽입	경로 조작 및 자원 삽입	크로스사이트 스크립팅
운영체제 명령어 삽입	위험한 형식 파일 업로드	신뢰되지 않는 URL 주소로 자동접속 연결

105. 보안 점검 - 보안 기능

적절한 인증 없이 중요기능 허용	부적절한 인가	중요한 자원에 대한 잘못된 관련 설정
취약한 암호화 알고리즘 사용	하드코드 된 비밀번호	중요 정보 평문 저장 및 전송



118. UNIX / LINUX의 CLI 기본 명령어

cat	cd	chmod	chown	cp	rm
find	fsck	kill	killall	ls	mkdir
rmdir	mv	ps	pwd	top	who

119. 기억장치 관리 전략

반입 전략	배치 전략	교체 전략
<u>요구 반입</u> , <u>예상 반입</u>	<u>최초 적합</u> , <u>최적 적합</u> , <u>최악 적합</u>	<u>FIFO</u> , <u>OPT</u> , <u>LRU</u> , <u>LFU</u> , <u>NUR</u> , <u>SCR</u>

120. 가상 기억장치

페이징	세그멘테이션
-----	--------

121. 가상기억장치 기타 관리 사항

국부성(Locality)	페이지 부재(Page Fault)
<u>시간 구역성</u> , <u>공간 구역성</u>	<u>페이지 부재 빈도</u>
워킹 셋(Working Set)	스래싱(Thrashing)

122. 페이지 교체 알고리즘

OPT(최적 교체)	FIFO	LRU
LFU	NUR	SCR(2차 기획 교체)

123. 프로세스(Process) 관리

제출(Submit)	접수(Hold)	준비(Ready)
실행(Run)	대기(Wait), <u>블록(Block)</u>	종료(Terminated, Exit)
	<u>Dispatch</u> , <u>Wake Up</u> , <u>Spooling</u>	

124. 비선점(Non-preemptive) 스케줄링

FOFS(선입선출)	SJF(단기 작업 우선)	HRN
Deadline(기한부)	Priority(우선순위)	<u>Aging(기법)</u>

125. 선점(Preemptive) 스케줄링

선점 우선순위	SRT	RR(라운드 로빈)
MQ(다단계 큐)	MFQ(다단계 피드백 큐)	

126. 교착상태(Dead Lock)

상호 배제	점유와 대기	비선점	환형 대기
-------	--------	-----	-------

127. 교착 상태 해결 방법

회피 기법	발견 기법	회복 기법
	예방 기법	
<u>상호 배제 부정</u> , <u>점유 및 대기 부정</u> , <u>비선점 부정</u> , <u>환형 대기 부정</u>		

128. 데이터베이스

통합된 데이터	저장된 데이터	운영 데이터	공용 데이터
---------	---------	--------	--------

129. DBMS(Database Management System, 데이터베이스 관리 시스템)

DBMS의 필수 기능	DBMS의 종류
<u>정의 기능</u> , <u>조작 기능</u> , <u>제어 기능</u>	<u>계층형 DBMS</u> , <u>망형 DBMS</u> , <u>관계형 DBMS</u>

130. 분산 데이터베이스(Distributed Database)

위치 투명성	중복 투명성	병행 투명성	장애 투명성
--------	--------	--------	--------

131. 고급 데이터베이스

데이터 웨어하우스	데이터 마트	데이터 마이닝	OLAP	OLTP
-----------	--------	---------	------	------

132. ER(Entity Relationship) 모델 - ER 도형

개체 타입	관계 타입	속성	기본키 속성
개체 타입과 속성 연결	다중값 속성	개체 타입 간의 연관성	개체 타입 간의 연관성

133. ER 모델 - 관계 및 관계xkdldq

차수에 따른 관계의 종류 단항 관계, 이항 관계, 삼항 관계, n항 관계	대응 카디널리티에 따른 관계의 종류 1:1 관계, 1:N 관계, N:M 관계
---	---

134. 관계 데이터베이스의 Relation 구조

튜플 튜플의 수 = 카디널리티(Cardinality) = 기수 = 대응수 도메인	속성 속성의 수 = 디그리(Degree) = 차수 릴레이션 인스턴스
--	---

135. ER 모델을 관계형 데이터 모델로 변환

관계 Y가 1:1 관계	관계 Y가 1:N 관계	관계 Y가 N:M 관계
--------------	--------------	--------------

136. 키(Key)의 개념 및 종류

슈퍼키(Super Key)	후보키(Candidate Key)	기본키(Primary Key)	대체키(Alternate Key)	외래키(Foreign Key)
----------------	--------------------	------------------	--------------------	------------------

137. 무결성(Integrity)

개체 무결성 NULL 무결성	도메인 무결성 고유 무결성	참조 무결성 키 무결성	사용자 정의 무결성 관계 무결성
--------------------	-------------------	-----------------	----------------------

138. 네트워크

근거리 통신망(LAN, Local Area Network)	광대역 통신망(WAN, Wid Area Network)
----------------------------------	--------------------------------

139. IP 주소(Internet Protocol Address)

Class A	Class B	Class C	Class D	Class E
---------	---------	---------	---------	---------

140. IPv6(Internet Protocol version 6)

유니캐스트	멀티캐스트	애니캐스트
-------	-------	-------

141. OSI(Open System Interconnection) 참조 모델

물리 계층	데이터 계층	네트워크 계층	
전송 계층	세션 계층	표현 계층	응용 계층

142. 네트워크 관련 장비

허브 더미 허브, 스위칭 허브	리피터	브리지
라우터	게이트웨이	스위치 LS, L3, L4, L7

143. TCP/IP(Transmission Control Protocol/Internet Protocol)

TCP	IP
-----	----

## 144. 프로토콜

구문(Syntax)

의미(Semantics)

시간(Time)

## 145. TCP/IP의 구조

응용계층

(OSI : 응용 계층, 표현계층, 세션 계층)  
TELNET, FTP, SMTP, SNMP, DNS, HTTP

인터넷 계층

(OSI : 네트워크 계층)  
IP, ICMP, IGMP, ARP, RARP

전송계층

(OSI : 전송 계층)  
TCP, UDP, RTCP

네트워크 액세스 계층

(OSI : 데이터 링크 계층, 물리 계층)  
Ethernet, IEEE 802, HDLC, X.25, RS-232C, ARQ

## 146. 회선 교환 방식(Circuit Switching)

공간 분할 교환 방식(SDS, Space Division Switching)

시분할 교환 방식(TDS, Time Division Switching)

## 146. 패킷 교환 방식(Packet Switching)

가상 회선 방식

데이터그램 방식

## 147. 라우팅(Routing)

라우팅 프로토콜

RIP(Routing Information Protocol),  
IGRP(Interior Gateway Routing Protocol),  
OSPF(Open Shortest Path First Protocol),  
BGP(Border Gateway Protocol)

라우팅 알고리즘

거리 벡터 알고리즘(Distance Vector Algorithm),  
링크 상태 알고리즘(Link State Algorithm)

# 11장. 제품 소프트웨어 패키징

## 148. 소프트웨어 패키징

기능 식별 → 모듈화 → 빌드 진행 → 사용자 환경 분석 → 패키징 및 적용 시험 → 패키징 변경 개선 → 배포

## 149. 릴리즈 노트(Release Note)

모듈 식별 → 릴리즈 정보 확인 → 릴리즈 노트 개요 작성 → 영향도 체크 → 정식 릴리즈 노트 작성 → 추가  
개선 항목 식별

## 150. 디지털 저작권 관리(DRM)

암호화  
저작권 표현

키 관리  
정책 관리

암호화 파일 생성  
크랙 방지

식별 기술  
인증

## 151. 소프트웨어 설치 매뉴얼 작성

기능 식별 → UI 분류 → 설치/백업 파일 확인 → Uninstall 절차 확인 → 이상 Case 확인 → 최종 매뉴얼 적용

## 152. 소프트웨어 사용자 매뉴얼 작성

기능 식별 → 사용자 화면 분류 → 사용자 환경 파일 확인 → 초기화 절차 확인 → 이상 Case 확인 → 최종  
매뉴얼 적용

## 153. 소프트웨어 패키징의 형상 관리(SCM)

형상 식별 → 버전 제어 → 형상 통제(변경 관리) → 형상 감사 → 형상 기록(상태 보고)

## 154. 소프트웨어 버전 등록 과정

가져오기(Import)

인출(Check-Out)

예치(Commit)

동기화(Update)

차이(Diff)

155. 소프트웨어 버전 관리 도구

공유 폴더 방식	클라이언트/서버 방식	분산 저장소 방식	Subversion(SVN)	Git
----------	-------------	-----------	-----------------	-----

156. 빌드 자동화 도구

Jenkins	Gradle
---------	--------