

# 정보처리기사 필기 한권으로 끝내기



메인에듀 정보기술연구소 /  
김대영 공 편저



- 비전공자도 독학으로  
합격이 가능한 필수교재
- 합격에 필요한 핵심이론 완벽정리
- 단원별 예상문제 수록

#### 필기 과목

소프트웨어 설계  
소프트웨어 개발  
데이터베이스 구축  
프로그래밍어 활용  
정보시스템 구축관리



**NCS반영  
-  
출제기준  
전면개편**



# 머리말

대한민국 IT 대표 자격증 정보처리기사가 2020년부터 NCS 기반으로 대폭 개편되어 이전의 수험서와는 전혀 달라졌습니다.

정보처리기사는 2020년부터는 출제 기준이 새로 개정되어 국가 직무능력 표준(NCS) 방식으로 변경되었습니다.

새로 변경된 기사 시험은 개발자가 해야 될 것들을 특히 (소프트 웨어설계, 소프트웨어 개발 서버프로그램) 등을 공부해야 하는 시험으로 탈바꿈 됩니다. 특히 소프트웨어 엔지니어링 분야나 보안 그리고 ICT 트렌드가 반영되었습니다.

과거에는 기출문제를 외워 합격이 가능했었지만 새로운 개정판은 개발자로서 현장직무 중심의 능력을 보는 시험으로 바뀌겠다는 것입니다.

그래서 새로운 문제가 나올 것을 같이 준비를 해야 합니다.

정보처리기사는 IT 전공자와 전공을 하지 않는 비전공자도 많이 응시하는 자격증으로 본 수험서는 꼭 알아야 할 문제와 개념을 중심으로 설명을 하였습니다.

본 수험서로 공부하시는 모든 분들의 합격을 진심으로 기원합니다.

## 1. 정보처리기사 기본정보

- **직무내용** : 정보시스템 등의 개발 요구사항을 이해하여 각 업무에 맞는 소프트웨어의 기능에 관한 설계, 구현 및 테스트를 수행하고 사용자에게 배포하며, 버전관리를 통해 제품의 성능을 향상시키고 서비스를 개선하는 직무이다.
- **시 행 처** : 한국산업인력공단(www.q-net.or.kr)
- **관련학과** : 모든 학과 응시가능
- **시험과목 및 활용 국가직무능력표준(NCS)**  
국가기술자격의 현장성과 활용성 제고를 위해 국가직무능력표준(NCS)을 기반으로 자격의 내용(시험과목, 출제기준 등)을 직무 중심으로 개편하여 시행합니다.

### 1) 필기 시험

| 과목명         | 활용 NCS 능력단위   | NCS 세분류   |
|-------------|---------------|-----------|
| 소프트웨어 설계    | 요구사항 확인       | 응용SW엔지니어링 |
|             | 화면 설계         |           |
|             | 애플리케이션 설계     |           |
|             | 인터페이스 설계      |           |
| 소프트웨어 개발    | 데이터 입출력 구현    | 응용SW엔지니어링 |
|             | 통합 구현         |           |
|             | 제품소프트웨어 패키징   |           |
|             | 애플리케이션테스트 관리  |           |
|             | 인터페이스 구현      |           |
| 데이터베이스 구축   | SQL 응용        | DB엔지니어링   |
|             | SQL 활용        |           |
|             | 논리 데이터베이스 설계  |           |
|             | 물리 데이터베이스 설계  |           |
|             | 데이터 전환        |           |
| 프로그래밍 언어 활용 | 서버프로그램 구현     | 응용SW엔지니어링 |
|             | 프로그래밍 언어 활용   |           |
|             | 응용 SW 기초기술 활용 |           |

# 시험요강

|            |                     |           |
|------------|---------------------|-----------|
| 정보시스템 구축관리 | 소프트웨어개발 방법론 활용      | 응용SW엔지니어링 |
|            | IT프로젝트정보시스템<br>구축관리 | IT프로젝트관리  |
|            | 소프트웨어 개발보안 구축       | 보안엔지니어링   |
|            | 시스템 보안 구축           |           |

## 2) 실기 시험

| 과목명     | 활용 NCS 능력단위    | NCS 세분류   |
|---------|----------------|-----------|
| 정보처리 실무 | 요구사항 확인        | 응용SW엔지니어링 |
|         | 데이터 입출력 구현     |           |
|         | 통합 구현          |           |
|         | 제품소프트웨어 패키징    |           |
|         | 서버프로그램 구현      |           |
|         | 인터페이스 구현       |           |
|         | 프로그래밍 언어 활용    |           |
|         | 응용 SW 기초 기술 활용 |           |
|         | 화면 설계          |           |
|         | 애플리케이션 테스트 관리  |           |
|         | SQL 응용         | DB엔지니어링   |
|         | 소프트웨어 개발 보안 구축 | 보안엔지니어링   |

### ○ 점정방법

- 필기 : 객관식 4지 택일형, 과목당 20문항(과목당 30분)
- 실기 : 필답형(2시간30분)

### ○ 합격기준

- 필기 : 100점을 만점으로 하여 과목당 40점 이상, 전과목 평균 60점 이상.
- 실기 : 100점을 만점으로 하여 60점 이상.

## 2. 정보처리기사 진로 및 전망

- 기업체 전산실, 소프트웨어 개발업체, SI(system integrated)업체 (정보통신, 시스템 구축회사 등)정부기관, 언론기관, 교육 및 연구기관, 금융기관, 보험업, 병원 등 컴퓨터 시스템을 개발 및 운용하거나, 데이터 통신을 이용하여 정보처리를 시행하는 업체에서 활동하고 있다. 품질검사 전문기관 기술인력과 감리원 자격을 취득하여 감리 전문회사의 감리원으로 진출할 수 있다. 3년 이상의 실무경력이 있는 자는 측량분야 수치지도제작업의 정보처리 담당자로 진출가능하고, 정보통신부의 별정우체국 사무장, 사무주임, 사무보조 등 사무원으로 진출할 수 있다.
- 정보화사회로 이행함에 따라 지식과 정보의 양이 증대되어 작업량과 업무량이 급속하게 증가했다. 또한 각종 업무의 전산화 요구가 더욱 증대되어 사회 전문분야로 컴퓨터 사용이 보편화되면서 컴퓨터산업은 급속도로 확대되었다. 컴퓨터산업의 확대는 곧 이 분야의 전문인력에 대한 수요 증가로 이어졌다. 따라서 컴퓨터 관련 자격증에 대한 관심도 증가하고 있어 최근 응시자수와 합격자수가 증가하고 있는 추세이다. 또한 취업, 입학시 가산점을 주거나 병역특례 등 혜택이 있어 실생활에 널리 통용되고 인정받을 수 있다. (학점인정 30학점, 공무원 임용시험시 7급은 3%, 9급은 3% 가산특전이 있다.)

## 3.정보처리기사 필기 출제기준

### ○ 제1과목 소프트웨어 설계

| 주요항목       | 세부항목          | 세세항목                                                                                     |
|------------|---------------|------------------------------------------------------------------------------------------|
| 1. 요구사항 확인 | 1. 현행 시스템 분석  | 1. 플랫폼 기능 분석<br>2. 플랫폼 성능 특성 분석<br>3. 운영체제 분석<br>4. 네트워크 분석<br>5. DBMS 분석<br>6. 비즈니스융합분석 |
|            | 2. 요구사항 확인    | 1. 요구분석기법<br>2. UML<br>3. 애자일(Agile)                                                     |
|            | 3. 분석모델 확인    | 1. 모델링 기법<br>2. 분석자동화 도구<br>3. 요구사항 관리 도구                                                |
| 2. 화면 설계   | 1. UI 요구사항 확인 | 1. UI 표준<br>2. UI 지침<br>3. 스토리보드                                                         |
|            | 2. UI 설계      | 1. 감성공학<br>2. UI 설계 도구                                                                   |

# 시험요강

|              |                  |                                                                          |
|--------------|------------------|--------------------------------------------------------------------------|
| 3. 애플리케이션 설계 | 1. 공통 모듈 설계      | 1. 설계 모델링<br>2. 소프트웨어 아키텍처<br>3. 재사용<br>4. 모듈화<br>5. 결합도<br>6. 응집도       |
|              | 2. 객체지향 설계       | 1. 객체지향(OOP)<br>2. 디자인패턴                                                 |
| 4. 인터페이스 설계  | 1. 인터페이스 요구사항 확인 | 1. 내외부 인터페이스 요구사항<br>2. 요구공학                                             |
|              | 2. 인터페이스 대상 식별   | 1. 시스템 아키텍처<br>2. 인터페이스 시스템                                              |
|              | 3. 인터페이스 상세 설계   | 1. 내외부 송수신<br>2. 데이터 명세화<br>3. 오류 처리방안 명세화<br>4. 인터페이스 설계<br>5. 미들웨어 솔루션 |

## ○ 제2과목 소프트웨어 개발

| 주요항목          | 세부항목               | 세세항목                                                                     |
|---------------|--------------------|--------------------------------------------------------------------------|
| 1. 데이터 입출력 구현 | 1. 논리 데이터저장소 확인    | 1. 자료구조<br>2. 논리 데이터저장소                                                  |
|               | 2. 물리 데이터저장소 설계    | 1. 물리 데이터저장소<br>2. ORM(Object-Relational Mapping) 프레임워크<br>3. 트랜잭션 인터페이스 |
|               | 3. 데이터 조작 프로시저 작성  | 1. 프로시저<br>2. 프로그램 디버깅<br>3. 단위테스트 도구                                    |
|               | 4. 데이터 조작 프로시저 최적화 | 1. 쿼리(Query) 성능 측정<br>2. 소스코드 인스펙션                                       |
| 2. 통합 구현      | 1. 모듈 구현           | 1. 단위모듈 구현<br>2. 단위모듈 테스트                                                |
|               | 2. 통합 구현 관리        | 1. IDE 도구<br>2. 협업도구<br>3. 형상관리 도구                                       |

|                  |                     |                                                                 |
|------------------|---------------------|-----------------------------------------------------------------|
| 3. 제품소프트웨어 패키징   | 1. 제품소프트웨어 패키징      | 1. 애플리케이션 패키징<br>2. 애플리케이션 배포 도구<br>3. 애플리케이션 모니터링 도구<br>4. DRM |
|                  | 2. 제품소프트웨어 매뉴얼 작성   | 1. 제품소프트웨어 매뉴얼 작성<br>2. 국제 표준 제품 품질 특성                          |
|                  | 3. 제품소프트웨어 버전관리     | 1. 소프트웨어 버전관리 도구<br>2. 빌드 자동화 도구                                |
| 4. 애플리케이션 테스트 관리 | 1. 애플리케이션 테스트케이스 설계 | 1. 테스트 케이스<br>2. 테스트 레벨<br>3. 테스트 시나리오<br>4. 테스트 지식 체계          |
|                  | 2. 애플리케이션 통합 테스트    | 1. 결함관리 도구<br>2. 테스트 자동화 도구<br>3. 통합 테스트                        |
|                  | 3. 애플리케이션 성능 개선     | 1. 알고리즘<br>2. 소스코드 품질분석 도구<br>3. 코드 최적화                         |
| 5. 인터페이스 구현      | 1. 인터페이스 설계 확인      | 1. 인터페이스 기능 확인<br>2. 데이터 표준 확인                                  |
|                  | 2. 인터페이스 기능 구현      | 1. 인터페이스 보안<br>2. 소프트웨어 연계 테스트                                  |
|                  | 3. 인터페이스 구현 검증      | 1. 설계 산출물<br>2. 인터페이스 명세서                                       |

#### ○ 제3과목 데이터베이스 구축

| 주요항목      | 세부항목          | 세세항목                                                  |
|-----------|---------------|-------------------------------------------------------|
| 1. SQL 응용 | 1. 절차형 SQL 작성 | 1. 트리거<br>2. 이벤트<br>3. 사용자 정의 함수<br>4. SQL 문법         |
|           | 2. 응용 SQL 작성  | 1. DML<br>2. DCL<br>3. 윈도우 함수<br>4. 그룹 함수<br>5. 오류 처리 |

# 시험요강

|                 |                   |                                                                                      |
|-----------------|-------------------|--------------------------------------------------------------------------------------|
| 2. SQL 활용       | 1. 기본 SQL 작성      | 1. DDL<br>2. 관계형 데이터 모델<br>3. 트랜잭션<br>4. 테이블<br>5. 데이터 사전                            |
|                 | 2. 고급 SQL 작성      | 1. 뷰<br>2. 인덱스<br>3. 집합 연산자<br>4. 조인<br>5. 서브쿼리                                      |
| 3. 논리 데이터베이스 설계 | 관계데이터베이스 모델       | 1. 관계 데이터 모델<br>2. 관계데이터언어(관계대수, 관계해석)<br>3. 시스템카탈로그와 뷰                              |
|                 | 데이터모델링 및 설계       | 1. 데이터모델 개념<br>2. 개체-관계(E-R)모델<br>3. 논리적 데이터모델링<br>4. 데이터베이스 정규화<br>5. 논리 데이터모델 품질검증 |
| 4. 물리 데이터베이스 설계 | 1. 물리요소 조사 분석     | 1. 스토리지<br>2. 분산 데이터베이스<br>3. 데이터베이스 이중화 구성<br>4. 데이터베이스 암호화<br>5. 접근제어              |
|                 | 2. 데이터베이스 물리속성 설계 | 1. 파티셔닝<br>2. 클러스터링<br>3. 데이터베이스 백업<br>4. 테이블 저장 사이징<br>5. 데이터 지역화(locality)         |
|                 | 3. 물리 데이터베이스 모델링  | 1. 데이터베이스 무결성<br>2. 칼럼 속성<br>3. 키 종류<br>4. 반정규화                                      |
|                 | 4. 데이터베이스 반정규화    | 1. 정규화                                                                               |
|                 | 5. 물리데이터 모델 품질검토  | 1. 물리데이터 모델 품질 기준<br>2. 물리 E-R 다이어그램<br>3. CRUD 분석<br>4. SQL 성능 튜닝                   |

|           |              |                                                                           |
|-----------|--------------|---------------------------------------------------------------------------|
| 5. 데이터 전환 | 1. 데이터 전환 기술 | 1. 초기데이터 구축<br>2. ETL(Extraction, Transformation, Loading)<br>3. 파일 처리 기술 |
|           | 2. 데이터 전환 수행 | 1. 데이터 전환 수행 계획<br>2. 체크리스트<br>3. 데이터 검증                                  |
|           | 3. 데이터 정제    | 1. 데이터 정제<br>2. 데이터 품질 분석<br>3. 오류 데이터 측정                                 |

# 시험요강

## ○ 제4과목 프로그래밍언어 활용

| 주요항목              | 세부항목          | 세세항목                                                                  |
|-------------------|---------------|-----------------------------------------------------------------------|
| 1. 서버프로그램 구현      | 1. 개발환경 구축    | 1. 개발환경 구축<br>2. 서버 개발 프레임워크                                          |
|                   | 2. 서버 프로그램 구현 | 1. 보안 취약성 식별<br>2. API                                                |
|                   | 3. 배치 프로그램 구현 | 1. 배치 프로그램                                                            |
| 2. 프로그래밍 언어 활용    | 1. 기본문법 활용    | 1. 데이터 타입<br>2. 변수<br>3. 연산자                                          |
|                   | 2. 언어특성 활용    | 1. 절차적 프로그래밍 언어<br>2. 객체지향 프로그래밍 언어<br>3. 스크립트 언어<br>4. 선언형 언어        |
|                   | 3. 라이브러리 활용   | 1. 라이브러리<br>2. 데이터 입출력<br>3. 예외 처리<br>4. 프로토타입                        |
| 3. 응용 SW 기초 기술 활용 | 1. 운영체제 기초 활용 | 1. 운영체제 종류<br>2. 메모리 관리<br>3. 프로세스 스케줄링<br>4. 환경변수<br>5. shell script |
|                   | 2. 네트워크 기초 활용 | 1. 인터넷 구성의 개념<br>2. 네트워크 7 계층<br>3. IP<br>4. TCP/UDP                  |
|                   | 3. 기본 개발환경 구축 | 1. 웹서버<br>2. DB서버<br>3. 패키지                                           |

○ 제5과목 정보시스템 구축관리

| 주요항목                  | 세부항목                | 세세항목                                                                                                                         |
|-----------------------|---------------------|------------------------------------------------------------------------------------------------------------------------------|
| 1. 소프트웨어개발 방법론 활용     | 1. 소프트웨어개발 방법론 선정   | 1. 소프트웨어 생명주기 모델<br>2. 소프트웨어 개발 방법론<br>3. 요구공학 방법론<br>4. 비용산정 모델                                                             |
|                       | 2. 소프트웨어개발 방법론 테일러링 | 1. 소프트웨어 개발 표준<br>2. 테일러링 기준<br>3. 소프트웨어 개발 프레임워크                                                                            |
| 2. IT프로젝트 정보 시스템 구축관리 | 1. 네트워크 구축 관리       | 1. IT 신기술 및 네트워크 장비 트렌드 정보<br>2. 네트워크 장비(라우터, 백본 스위치 등)                                                                      |
|                       | 2. SW 구축 관리         | 1. IT 신기술 및 SW 개발 트렌드 정보<br>2. SW개발보안 정책                                                                                     |
|                       | 3. HW 구축 관리         | 1. IT 신기술 및 서버장비 트렌드 정보<br>2. 서버장비 운영(Secure-OS, 운영체제, NAS, DAS, SAN, 고 가용성 등)                                               |
|                       | 4. DB 구축 관리         | 1. IT 신기술 및 데이터베이스 기술 트렌드 정보<br>2. 데이터베이스 관리기능<br>3. 데이터베이스 표준화                                                              |
| 3. 소프트웨어 개발 보안 구축     | 1. SW개발 보안 설계       | 1. Secure SDLC(Software Development Life Cycle)<br>2. 입력데이터 검증 및 표현<br>3. 보안기능(인증, 접근제어, 기밀성, 권한 관리 등)<br>4. 에러처리<br>5. 세션통제 |
|                       | 2. SW개발 보안 구현       | 1. 암호 알고리즘<br>2. 코드오류<br>3. 캡슐화<br>4. API 오용                                                                                 |

# 시험요강

|              |              |                                                                         |
|--------------|--------------|-------------------------------------------------------------------------|
| 4. 시스템 보안 구축 | 1. 시스템 보안 설계 | 1. 서비스 공격 유형<br>2. 서버 인증<br>3. 서버 접근통제<br>4. 보안 아키텍처<br>5. 보안 Framework |
|              | 2. 시스템 보안 구현 | 1. 로그 분석<br>2. 보안 솔루션<br>3. 취약점 분석                                      |

## 제1과목 소프트웨어 설계 19

|                   |                                  |    |
|-------------------|----------------------------------|----|
| <b>Chapter 01</b> | <b>요구사항 확인</b>                   | 21 |
| 1.                | 현행 시스템 분석                        | 21 |
| 2.                | 요구사항 확인                          | 25 |
| 3.                | 분석모델 확인                          | 35 |
| <b>Chapter 02</b> | <b>화면설계</b>                      | 49 |
| 1.                | UI 요구사항 확인                       | 49 |
| 2.                | UI 설계                            | 52 |
| <b>Chapter 03</b> | <b>애플리케이션 설계</b>                 | 61 |
| 1.                | 공통 모듈 설계                         | 61 |
| 2.                | 객체지향 설계                          | 70 |
| <b>Chapter 04</b> | <b>인터페이스 설계</b>                  | 87 |
| 1.                | 인터페이스 요구사항 확인                    | 87 |
| 2.                | 요구 공학 (Requirements Engineering) | 88 |
| 3.                | 인터페이스 대상 식별                      | 91 |
| 4.                | 인터페이스 상세 설계                      | 93 |

## 제2과목 소프트웨어 개발 109

|                   |                   |     |
|-------------------|-------------------|-----|
| <b>Chapter 01</b> | <b>데이터 입출력 구현</b> | 111 |
| 1.                | 자료구조              | 111 |
| 2.                | 데이터 조작 프로시저 작성    | 123 |
| 3.                | 데이터 조작 프로시저 최적화   | 124 |
| <b>Chapter 02</b> | <b>통합 구현</b>      | 133 |
| 1.                | 모듈 구현             | 133 |
| 2.                | 통합 구현 관리          | 135 |

|                                 |     |
|---------------------------------|-----|
| <b>Chapter 03 제품소프트웨어 패키징</b>   | 144 |
| 1. 제품소프트웨어 패키징                  | 144 |
| 2. 애플리케이션 매뉴얼 작성                | 149 |
| 3. 국제 표준 제품 품질 특성               | 150 |
| 3. 제품 소프트웨어 버전 관리               | 151 |
| <b>Chapter 04 애플리케이션 테스트 관리</b> | 161 |
| 1. 애플리케이션 테스트 케이스 설계            | 161 |
| 2. 애플리케이션 통합 테스트                | 169 |
| 3. 애플리케이션 성능 개선                 | 173 |
| <b>Chapter 05 인터페이스 구현</b>      | 189 |
| 1. 인터페이스 설계 확인                  | 189 |
| 2. 인터페이스 기능 구현                  | 192 |
| 3. 인터페이스 구현 검증                  | 194 |

## 제3과목 데이터베이스

203

|                                |     |
|--------------------------------|-----|
| <b>Chapter 01 SQL 응용</b>       | 205 |
| 1. 절차형 SQL 작성                  | 205 |
| 2. 응용 SQL 작성                   | 210 |
| <b>Chapter 02 SQL 활용</b>       | 225 |
| 1. 기본 SQL 작성                   | 225 |
| 2. 고급 SQL 작성                   | 231 |
| <b>Chapter 03 논리 데이터베이스 설계</b> | 247 |
| 1. 관계데이터베이스 모델                 | 247 |
| 2. 관계 데이터 언어 (관계대수, 관계해석)      | 251 |
| 3. 시스템 카탈로그와 뷰                 | 254 |
| 2. 데이터모델링 및 설계                 | 255 |

**Chapter 04 물리 데이터베이스 설계 .....270**

- 1. 물리 요소 조사 분석 .....270
- 2. 데이터베이스 물리 속성 설계 .....273
- 3. 물리 데이터베이스 모델링 .....275
- 4. 물리데이터 모델 품질검토 .....278

**Chapter 05 데이터 전환 .....289**

- 1. 데이터 전환 기술 .....289
- 2. 데이터 전환 수행 .....291
- 3. 데이터 정제 .....293

**제4과목 프로그래밍언어활용 299**

**Chapter 01 서버프로그램 구현 .....301**

- 1. 개발환경 구축 .....301
- 2. 서버 프로그램 구현 .....303
- 5. 배치 프로그램 구현 .....304

**Chapter 02 프로그래밍 언어 활용 .....309**

- 1. 기본문법 활용 .....309
- 2. 언어 특성 활용 .....313
- 3. 라이브러리 활용 .....327

**Chapter 03 응용 SW 기초 기술 활용 .....342**

- 1. 운영체제 기초 활용 .....342
- 2. 네트워크 기초 활용 .....357
- 3. 기본 개발환경 구축 .....365

|                   |                            |     |
|-------------------|----------------------------|-----|
| <b>Chapter 01</b> | <b>소프트웨어 개발 방법론 활용</b>     | 379 |
| 1.                | 소프트웨어 개발 방법론 선정            | 379 |
| 2.                | 소프트웨어 개발 방법론 테일러링          | 386 |
| <b>Chapter 02</b> | <b>IT 프로젝트 정보시스템 구축 관리</b> | 398 |
| 1.                | 네트워크 구축관리                  | 398 |
| 2.                | SW 구축 관리                   | 404 |
| 3.                | HW 구축 관리                   | 405 |
| 4.                | DB 구축 관리                   | 408 |
| <b>Chapter 03</b> | <b>소프트웨어 개발 보안 구축</b>      | 417 |
| 1.                | SW 개발 보안 설계                | 417 |
| 2.                | SW개발 보안 구현                 | 422 |
| <b>Chapter 04</b> | <b>시스템 보안 구축</b>           | 433 |
| 1.                | 시스템 보안 설계                  | 433 |
| 2.                | 시스템 보안 구현                  | 438 |



# 제1과목

## 소프트웨어 설계



## Chapter 01 요구사항 확인

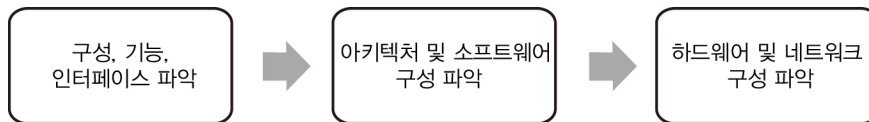
### 1. 현행 시스템 분석

#### (1) 현행 시스템 파악

##### 1) 현행 시스템 파악

현행 시스템의 구성, 제공기능, 시스템들 간의 정보 파악, 사용 기술, 소프트웨어, 하드웨어, 네트워크 구성 등을 파악하는 활동을 말한다. 향후 개발하고자 하는 시스템의 개발범위와 이행 방향성 설정에 도움을 준다.

##### 2) 현행 시스템 파악 절차



##### ① 시스템 구성, 기능, 인터페이스 파악

|           |                                                   |
|-----------|---------------------------------------------------|
| 시스템 구성    | 주요 업무를 처리하는 기간 업무 시스템과 지원 업무 시스템으로 구분하여 표시        |
| 시스템 기능    | 주요기능, 하부기능, 세부기능으로 구분하여 계층적으로 표시                  |
| 시스템 인터페이스 | 단위 업무시스템 간 전달되는 데이터 종류, 형식, 프로토콜, 연계 유형, 주기 등을 표시 |

##### ② 아키텍처 및 소프트웨어 구성 파악

|          |                                           |
|----------|-------------------------------------------|
| 아키텍처 구성  | 최상위 수준에서 계층별로 표현한 아키텍처 구성도를 작성            |
| 소프트웨어 구성 | 소프트웨어들의 제품명, 용도, 라이선스 적용 방식, 라이선스 수 등을 명시 |

### ③ 하드웨어 및 네트워크 구성 파악

|         |                                                         |
|---------|---------------------------------------------------------|
| 하드웨어 구성 | 단위 업무 시스템들이 운용되는 서버의 주요 사양과 수량, 그리고 서버의 이중화의 적용 여부를 명시함 |
| 네트워크 구성 | 서버의 위치, 서버 간의 네트워크 연결 방식을 네트워크 구성도로 작성함                 |

## (2) 플랫폼 기능 분석

### 1) 플랫폼의 정의

- ① 플랫폼은 다양한 서비스를 주고받기 위한 공통의 기반 구조이다.
- ② 공급자와 수요자가 직접 참여해서 각자가 얻고자 하는 가치를 거래할 수 있도록 만들어진 환경이다.
- ③ 소프트웨어적 관점의 플랫폼이란 다양한 애플리케이션을 실행하고 상호 호환이 쉽게 이루어질 수 있도록 만들어진 기반 환경을 의미한다.

### 2) 플랫폼의 기능 분석

- ① 플랫폼과 애플리케이션 연결 기능
- ② 소프트웨어 개발 및 운영 비용 감소 기능
- ③ 소프트웨어 개발 생산성 향상 기기능
- ④ 커뮤니티 형성에 의한 네트워크 효과 유발 기능

### 3) 플랫폼의 성능 특성 분석

|                      |                           |
|----------------------|---------------------------|
| 처리량 (Through put)    | 단위 시간당 처리하는 트랜잭션의 수       |
| 응답시간 (Response time) | 결과를 요청 후 응답을 받기까지 걸리는 시간  |
| 가용성 (Availability)   | 플랫폼을 필요할 때 즉시 이용할 수 있는 특성 |
| 사용률 (Utilization)    | 플랫폼 이용 시 사용되는 자원의 사용량     |

## (3) 운영체제 분석

### 1) 운영체제의 정의

- ① 운영체제(OS)는 하드웨어와 소프트웨어 자원을 관리하고 컴퓨터의 공통 서비스를 제공하는 시스템 소프트웨어를 말한다.



- ② 마이크로소프트 윈도우즈(Microsoft Windows), 유닉스(UNIX), 리눅스(Linux), 아이오에스(iOS), 안드로이드(Android)등이 해당한다.

## 2) 운영체제 요구사항 식별 시 고려사항

| 구분    | 내용                                                                                                                                                             |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 신뢰도   | <ul style="list-style-type: none"> <li>• 장기간 시스템 운영 시 운영체제 고유의 장애 발생 가능성</li> <li>• 운영체제의 버그 등으로 인한 패치 설치를 위한 재가동</li> <li>• 메모리 누수로 인한 성능 저하 및 재기동</li> </ul> |
| 성능    | <ul style="list-style-type: none"> <li>• 대규모 및 대량 파일 작업 처리</li> <li>• 대규모 동시 사용자 요청 처리</li> <li>• 지원 가능한 메모리 크기(32bit, 64bit)</li> </ul>                       |
| 기술 지원 | <ul style="list-style-type: none"> <li>• 공급업체들의 안정적인 기술 지원</li> </ul>                                                                                          |
| 주변 기기 | <ul style="list-style-type: none"> <li>• 다수의 주변 기기 지원 여부</li> </ul>                                                                                            |
| 구축 비용 | <ul style="list-style-type: none"> <li>• 설치할 응용프로그램의 라이선스 정책 및 비용</li> <li>• 지원 가능한 하드웨어 비용, 유지 및 관리 비용</li> </ul>                                             |

## 3) 운영체제의 종류 및 특징

| 종류                | 저작자            | 특징                                  |
|-------------------|----------------|-------------------------------------|
| 윈도우즈<br>(Windows) | Microsoft      | 중/소규모 서버, 일반 PC 등 유지 및 관리 비용 장<br>점 |
| 유닉스(UNIX)         | IBM, HP, SUN   | 대용량 처리, 안정성 높은 엔터프라이즈 급 서버          |
| 리눅스(Linux)        | Linux Torvalds | 중/대규모 서버 대상, 높은 보안성 제공              |
| 아이오에스(iOS)        | Apple          | 스마트폰, 태블릿PC, 다양한 기기의 호환성 제공         |
| 안드로이드(Android)    | Google         | 스마트폰, 태블릿PC, 높은 보안성과 고성능 제공         |

## (4) DBMS 분석

### 1) DBMS의 정의

DBMS(Data Base Management System)이란 사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성해주고 데이터베이스를 관리해주는 시스템 소프트웨어를 말한다.

## 2) DBMS의 종류 및 특징

| 종류      | 저작자              | 주요 용도                              |
|---------|------------------|------------------------------------|
| Oracle  | Oracle           | 대규모, 대량 데이터의 안정적 처리                |
| MS-SQL  | Micorosoft       | 중소 규모 데이터의 안정적 처리                  |
| MySQL   | MySQL AB, Oracle | 오픈 소스 RDBMS                        |
| SQLite  | D.Richard Hipp   | 스마트폰, 태블릿 PC 등의 Embedded Database용 |
| MongoDB | MongoDB Inc.     | 오픈 소스 NoSQL 데이터베이스                 |

## 3) DBMS의 필수 기능

- ① 정의(Definition) 기능 : 생성할 데이터의 유형과 구조에 대한 정의, 이용 방식, 제약 조건 등을 명세하여 응용 프로그램과 데이터베이스 사이의 인터페이스를 제공하는 기능
- ② 조작(Manipulation) 기능 : 데이터 검색, 갱신, 삽입, 삭제 등을 체계적으로 처리하기 위하여 사용자와 데이터베이스 사이의 인터페이스 수단을 제공하는 기능
- ③ 제어(Control) 기능 : 데이터의 정확성과 안전성을 유지하는 기능(무결성, 보안, 병행 수행 제어, 회복 등)

## 4) DBMS 분석 시 고려사항

| 고려 관점 | 고려사항   | 내용                                                                                                                                        |
|-------|--------|-------------------------------------------------------------------------------------------------------------------------------------------|
| 성능 측면 | 가용성    | <ul style="list-style-type: none"> <li>시스템 장기간 운영 시 장애 발생 가능성</li> <li>DBMS의 버그 등으로 인한 패치 설치를 위한 재기동</li> <li>DBMS 이중화 및 복제 지원</li> </ul> |
|       | 성능     | <ul style="list-style-type: none"> <li>대량 거래 처리 기능</li> <li>비용 기반 최적화 지원 및 설정의 최소화</li> <li>다양한 튜닝 옵션 지원 여부</li> </ul>                    |
|       | 기술 지원  | <ul style="list-style-type: none"> <li>공급업체들의 안정적인 기술 지원</li> <li>오픈 소스 여부</li> </ul>                                                     |
| 지원 측면 | 상호 호환성 | <ul style="list-style-type: none"> <li>설치 가능한 OS 종류</li> <li>다양한 OS에서 지원되는 JDBC, ODBC</li> </ul>                                          |
|       | 구축 비용  | <ul style="list-style-type: none"> <li>라이선스 정책 및 비용</li> <li>유지 및 관리 비용</li> <li>총 소유 비용(TCO)</li> </ul>                                  |



## 2. 요구사항 확인

### (1) 요구분석기법

- ① 요구 분석은 사용자 요구사항을 이해하고 정리하여 소프트웨어 개발 시 어떤 기능이 필요한지, 기능을 수행하는데 제약사항은 무엇인지를 분석하는 것을 의미한다.
- ② 구조적 분석 방식, 객체지향 분석 방식이 사용된다.

### (2) 구조적 요구 분석

- ① 구조적 분석은 시스템을 하향식 분할하여 분석의 중복을 배제하고 도형 중심 도구를 사용하여 전체 시스템을 일관성 있게 이해할 수 있다.
- ② 구조적 분석 도구로 자료흐름도, 자료사전, 소단위 명세서, E-R 다이어그램 등이 사용된다.

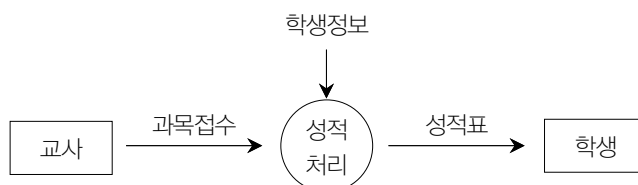
#### 1) 자료흐름도(DFD ; Data Flow Diagram)

요구사항 분석에서 자료흐름과 변화 과정을 도형 중심으로 기술하는 방법이다.

#### ▼ 자료 흐름도 구성요소

| 기호                 | 의미                     | 표기법 |
|--------------------|------------------------|-----|
| 프로세스(Process)      | 자료를 변환시키는 처리 과정을 나타낸다. |     |
| 자료 흐름(Flow)        | 자료의 이동(흐름)을 나타낸다.      |     |
| 자료 저장소(Data Store) | 시스템에서의 자료 저장소를 나타낸다.   |     |
| 단말(Terminator)     | 시스템과 교신하는 외부 개체를 나타낸다. |     |

예



## 2) 자료사전 (DD ; Data Dictionary)

자료흐름도에 있는 모든 자료와 자료 속성을 정의하는 메타데이터(Meta Data)로 기호를 사용하여 표기한다.

| 기호    | 의미                                   |
|-------|--------------------------------------|
| =     | 자료의 정의 : ~로 구성되어 있다.(is composed of) |
| +     | 자료의 연결 : 그리고(and)                    |
| ( )   | 자료의 생략 : 생략 가능한 자료(Optional)         |
| [   ] | 자료의 선택 : 또는(or)                      |
| { }   | 자료의 반복 : Iteration of                |
| **    | 자료의 설명 : 주석(Comment)                 |

예) 급여파일 = {사원명부 + 기본급 + 수당 + (상여금)}  
 결제방법 = [현금 | 계좌이체 | 신용카드]

## 3) 소단위 명세서 (Mini - Specification)

소단위 명세서는 세분화된 자료흐름도에서 처리 절차를 구조적 언어 등으로 기술한 것이다.

예) 고용자 명부의 각 고용 형태에 대하여 다음과 같이 급여 계산을 수행한다.

- 고용자 급여 형태를 파악한다.
- 급여 형태에 따른 급여액을 계산한다.
  - 사무직 급여액 = 근무시간 × 8,000원
  - 기능직 급여액 = 근무시간 × 8,500원
- 성명과 계산된 급여액을 급여 내역서에 기록한다.

|    |                                                    |
|----|----------------------------------------------------|
| 정답 | ④                                                  |
| 해설 | 현행 시스템 분석시 고려사항 : DBMS 분석, 네트워크 분석, 운영체제분석, 플랫폼 분석 |



# 제2과목

## 소프트웨어 개발



## Chapter 01 데이터 입출력 구현

## 1. 자료구조

## (1) 자료구조(Data structure)

자료구조란 자료를 기억공간에 어떻게 표현하고 저장할 것인가를 결정하는 것으로 객체나 객체 집합뿐만 아니라 그들의 관계까지 기술하는 것을 의미한다.

자료구조는 선형 구조와 비선형 구조로 나뉜다.



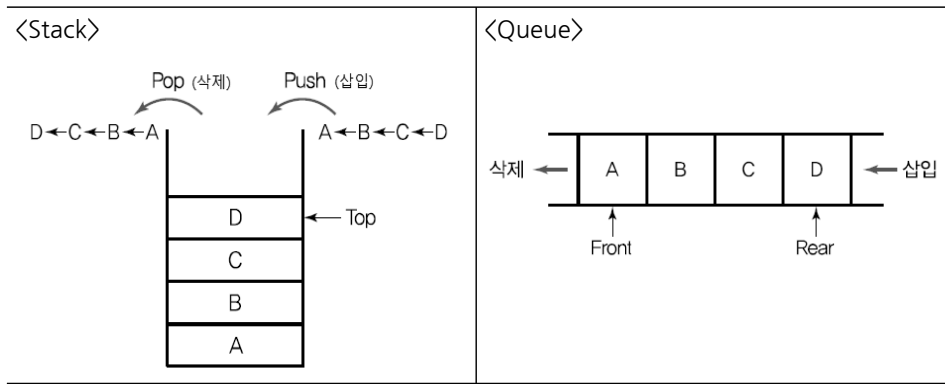
## (2) 선형구조

## 1) 배열(Array)

- ① 배열이란 연속되는 기억 공간에 자료가 저장되는 구조이다.
- ② 기억 공간의 밀도가 좋고 액세스 시간이 빠르다.
- ③ 삽입, 삭제가 어렵고 기억 공간이 인접해 있어야 한다.

## 2) 스택(stack)

- ① 리스트의 한쪽 끝에서만 자료의 삽입(push)과 삭제(pop)가 이루어진다.
- ② 마지막에 입력한 데이터가 먼저 출력되는 후입선출(LIFO ; Last In First Out) 구조이다.
- ③ 자료의 위치를 지정하기 위해 TOP 포인터(스택 포인터)를 사용한다.
- ④ 스택의 크기보다 삽입 자료가 많으면 오버플로우(overflow), 더 이상 삭제할 자료가 없으면 언더플로우(underflow)가 발생한다.
- ⑤ 응용 : 서브루틴 호출 시 복귀주소저장, 후위표기식 연산, 0번지 주소지정



## 3) 큐 (Queue)

- ① 리스트의 한쪽에서 자료 삽입이, 반대쪽에서 삭제가 이루어지는 구조이다.
- ② 먼저 입력한 자료가 먼저 출력되는 선입선출 (FIFO ; First In First Out) 구조를 갖는다.
- ③ 자료의 위치를 지정하기 위해 Front 포인터와 Rear 포인터를 이용한다.
- ④ 응용 : 운영체제의 작업 스케줄링

## 4) 데크 (Deque)

- ① 리스트의 양쪽 끝에서 삽입과 삭제가 모두 가능한 구조이다.
- ② 스택과 큐의 장점을 이용해 구현되었다.

## 5) 연결 리스트 (linked list)

- ① 연속공간이 아닌 임의의 공간에 자료 저장이 가능한 선형 구조이다.
- ② 자료값 뿐 아니라 다음 자료 주소를 가리키는 포인터로 구성된다.
- ③ 포인터의 변경만으로 데이터의 삽입과 삭제가 용이하다.

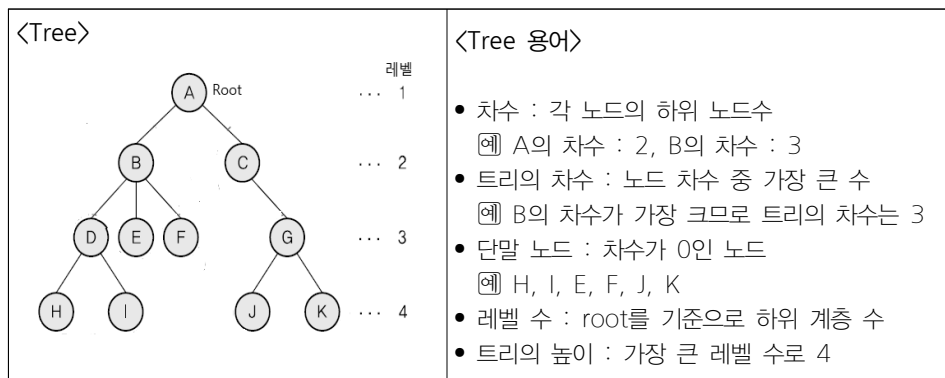


- ④ 단점 : 포인터를 연결하는 시간이 소요되므로 접근속도가 느리다.

## (2) 비선형 구조

### 1) 트리(Tree)

- ① 트리는 비선형 자료구조이다.
- ② 트리는 정점과 선분으로 형성된 계층구조로 사이클을 허용하지 않는다.
- ③ 하나의 루트노드는 반드시 존재해야 하며 루트를 제외한 노드는 서브트리로 구성된다.



### 2) 이진트리(Binary Tree)

#### ① 이진트리 특성

- 이진트리는 모든 노드의 차수가 2를 넘지 않는 트리이다.
- 레벨  $i$ 에서의 최대 노드 수 :  $2^{i-1}$
- 높이가  $K$ 인 2진 트리가 가질 수 있는 최대 노드 수 :  $2^K - 1$   
예) 트리의 높이가 7일 때 이진트리의 최대 노드수는  $2^7 - 1 = 128 - 1 = 127$

#### ② 이진트리 순회

- 이진트리의 각 노드를 한 번씩만 방문하는 방법이다,
- 루트노드의 탐색순서에 따라 전위순회(preorder), 중위순회(inorder), 후위순회(postorder) 방식으로 구분된다.

## 1. 다음 중 선형 구조와 비선형 구조가 옳게 짝지어진 것은?

- |              |                      |
|--------------|----------------------|
| ㉠ 스택(stack)  | ㉡ 큐(Queue)           |
| ㉢ 트리(Tree)   | ㉣ 연결리스트(Linked List) |
| ㉤ 그래프(Graph) |                      |

- ① 비선형 구조 : ㉠, ㉡, ㉤      선형구조 : ㉢, ㉣  
 ② 비선형 구조 : ㉢, ㉤      선형구조 : ㉠, ㉡, ㉣  
 ③ 비선형 구조 : ㉠, ㉡, ㉢      선형구조 : ㉣, ㉤  
 ④ 비선형 구조 : ㉢      선형구조 : ㉠, ㉡, ㉣, ㉤

|    |                                                                                         |
|----|-----------------------------------------------------------------------------------------|
| 정답 | ②                                                                                       |
| 해설 | 선형 구조: 배열(Array), 스택(Stack), 큐(Queue), 데크(Deque), 연결리스트<br>비선형 구조: 그래프(Graph), 트리(Tree) |

## 2. 스택에 대한 설명으로 틀린 것은?

- ① 입출력이 한쪽 끝으로만 제한된 리스트이다.  
 ② Head(front)와 Tail(rear)의 2개 포인터를 갖고 있다.  
 ③ LIFO 구조이다.  
 ④ 더이상 삭제할 데이터가 없는 상태에서 데이터를 삭제하면 언더플로(Underflow)가 발생한다.

|    |                                                    |
|----|----------------------------------------------------|
| 정답 | ②                                                  |
| 해설 | Head(front)와 Tail(rear)의 2개 포인터를 갖는 것은 큐(queue)이다. |

## 3. 자료구조에 대한 설명으로 틀린 것은?

- ① 큐는 포인터(pointer)를 이용해 삽입, 삭제를 쉽게 처리한다.  
 ② 큐는 운영체제 스케줄링에 응용된다.  
 ③ 스택은 Last In - First out 처리를 수행한다.  
 ④ 스택은 서브루틴 호출, 인터럽트 처리, 수식 계산 및 수식 표기법에 응용된다.



# 제3과목

## 데이터베이스



## Chapter 01 SQL 응용

### 1. 절차형 SQL 작성

- 절차형 SQL은 일반 개발 언어처럼 프로그래밍이 가능한 SQL을 의미한다.
- 절차형 SQL의 생성 모듈로는 트리거, 이벤트, 사용자 정의 함수 등이 있다

#### (1) 트리거(TRIGGER)

##### 1) 트리거(Triiger) 개요

- ① 데이터베이스의 이벤트 프로그래밍으로 일정 조건이 충족될 경우 이벤트(데이터 삽입, 수정, 삭제 등)가 발생하는 것이다.
- ② 사용자가 직접 호출하는 것이 아니라, 데이터베이스에서 자동적으로 호출되어 실행되는 것을 말한다.
- ③ 데이터 무결성 유지와 로그 메시지 출력 등의 처리를 위해 사용된다.

##### 2) 트리거 기본문법

|   |                                                                |
|---|----------------------------------------------------------------|
| ① | CREATE [OR REPLACE] TRIGGER 트리거명                               |
| ② | [BEFORE   AFTER] ON                                            |
| ③ | [FOR EACH ROW]<br>[WHEN (condition)]<br>DECLARE(변수선언)<br>BEGIN |
| ④ | {SQL 명령 작성}<br>{EXCEPTION}<br>END;                             |

- ① 트리거 선언
- ② BEFORE: INSERT, UPDATE, DELETE 문이 실행되기 전에 트리거 실행명령이다.
- ③ AFTER: INSERT, UPDATE, DELETE 문이 실행 후 트리거가 실행된다.
- ④ FOR EACH ROW : 행 트리거 옵션이다.

⑤ 필요한 SQL 명령어 작성

3) 트리거 예제

|   |                          |
|---|--------------------------|
| ① | CREATE TRIGGER T_SUBJECT |
| ② | BEFORE INSERT ON SUBJECT |
|   | BEGIN                    |
| ③ | IF(MAT < 60) THEN        |
| ④ | RAISE_APPLICATION_ERROR  |
| ⑤ | (-12345, '과락 입니다');      |
|   | END IF;                  |
|   | END;                     |

- ① T\_SUBJECT(과목) 트리거 명 선언
- ② 삽입(INSERT) 명령어 실행 전
- ③ 수학 점수가 60점 미만일 경우
- ④ 에러가 발생
- ⑤ 12345번 에러 발생하며 '과락입니다' 표시

(2) 이벤트 (EVENT)

1) 이벤트 개요

- ① 특정 시간대에 일정 기능을 실행시키는 기능이다.
- ② 특정 기능이란 프로시저, 함수, SQL 쿼리 등을 포함한다.
- ③ 이벤트가 발생하는 순서를 알면 매크로 또는 이벤트 프로시저가 실행되는 방법 및 시기에 영향을 줄 수 있다.

2) 이벤트 기본문법

|   |                                |
|---|--------------------------------|
| ① | CREATE EVENT 이벤트명              |
| ② | ON SCHEDULE                    |
|   | [ON COMPLETION [NOT] PRESERVE] |
|   | [ENABLE   DISABLE]             |
|   | [COMMENT '주석문']                |
|   | DO                             |
|   | [BEGIN]                        |



|  |             |
|--|-------------|
|  | {SQL 명령 작성} |
|  | [END]       |

- ① 이벤트 선언
- ② 스케줄 시작

### 3) 이벤트 예제

|   |                                                     |
|---|-----------------------------------------------------|
| - | CREATE EVENT T_EVENT                                |
| - | ON SCHEDULE                                         |
| - | EVERY 1 MINUTE                                      |
| - | DO INSERT INTO TEST(ID, PW) VALUES('park', '1234'); |

- ① T\_EVENT 이벤트 명 선언
- ② 스케줄 시작
- ③ 1분 마다 한번씩
- ④ INSERT 명령어 실행

## (3) 사용자 정의 함수

### 1) SQL 데이터베이스에서 사용자 정의 함수 개요

- ① SQL 문에서 평가할 수 있는 함수를 추가함으로써 데이터베이스 서버의 기능을 확장하기 위한 장치를 제공한다.
- ② 일련의 SQL명령의 결과를 단일 값으로 변환한다.
- ③ 사용자가 직접 정의하고 작성 가능하다.
- ④ SQL 표준은 스칼라와 테이블 함수를 구별한다.

### 2) 사용자 정의 함수 기본문법

|   |                              |
|---|------------------------------|
| ① | CREATE [OR REPLACE] FUNCTION |
| ② | IS                           |
|   | BEGIN                        |
| ③ | RETURN[VALUE]                |
|   | END;                         |

1. 데이터베이스 시스템에서 삽입, 갱신, 삭제 등의 이벤트가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL은?

- ① 트리거(trigger)                      ② 무결성(integrity)  
③ 잠금(lock)                            ④ 복귀(rollback)

정답 ①

- 무결성 : 데이터베이스의 정확성을 의미
- 잠금 : 병행제어를 수행하기 위한 데이터 접근 수단
- 복귀 : 트랜잭션 수행이 취소되었을 때 원래 상태로 돌아가는 것

2. 이벤트(EVENT)에 대한 설명으로 잘못된 것은?

- ① 특정 시간대에 일정 기능을 실행시키는 기능이다.
- ② 특정 기능이란 프로시저, 함수, SQL 쿼리문 등을 포함한다.
- ③ 사용자에게 의해 호출되어 실행된다.
- ④ 이벤트가 발생하는 순서를 알면 매크로 또는 이벤트 프로시저가 실행되는 방법 및 시기에 영향을 줄 수 있다.

정답 ③ 이벤트(EVENT)는 특정 시간대에 알아서 실행하는 것이지 사용자에게 의해서 호출되는 것이 아니다.

3. 사용자 정의함수에 대한 설명으로 잘못된 것은?

- ① 일련의 SQL 명령의 결과를 단일 값으로 변환한다.
- ② 사용자가 직접 정의하고 작성 가능하다.
- ③ 데이터베이스를 조작하는 명령어이다.
- ④ SQL 문에서 평가할 수 있는 함수를 추가함으로써 데이터베이스 서버의 기능을 확장하기 위한 장치를 제공한다.

|    |                           |
|----|---------------------------|
| 정답 | ③                         |
| 해설 | 데이터베이스를 조작하는 명령어는 DML 이다. |



# 제4과목

## 프로그래밍언어활용



## Chapter 01 서버프로그램 구현

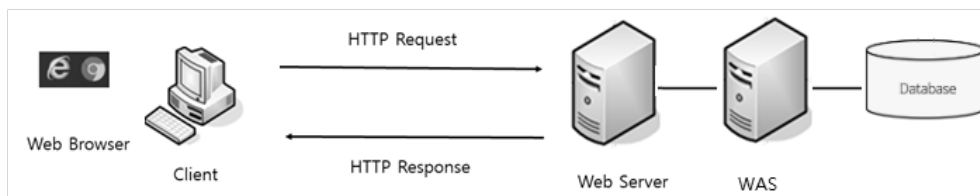
### 1. 개발환경 구축

#### (1) 개발환경 구축

##### 1) 개발환경 구축의 개념

- ① 개발환경은 물리적인 하드웨어 환경과 소프트웨어 환경으로 구성된다.
- ② 소프트웨어 환경은 서버 운영 소프트웨어와 프로그램 개발 소프트웨어로 구성된다.

##### 2) 서버 운영 도구



- ① Web Server (웹서버)
  - 클라이언트로부터 요청받은 정적 데이터를 처리하는 서버
  - Apache, IIS 등
- ② WAS (웹 애플리케이션 서버)
  - 클라이언트로부터 요청받은 동적 데이터를 처리하는 서버
  - Tomcat, JEUS, Websphere 등
- ③ DB Server (데이터베이스 서버)
  - 데이터베이스와 DBMS를 운영하는 서버
  - Oracle, Mysql, MSsql

### 3) 프로그램 개발도구

#### ① 구현 도구

- 프로그램의 소스 작성, 디버깅, 수정을 할 때 주로 사용하는 도구이다.
- Visual Studio, Eclipse, Edit Plus

#### ② 빌드 도구

- 코드의 컴파일, 링크 및 실행 그리고 배포를 하는 도구이다.
- ant, maven, gradle

#### ③ 테스트 도구

- 프로그램 소스의 품질에 관한 정보를 제공하는 조사하는 도구이다.
- GUITAR, Coverity, Defensics

#### ④ 형상관리 도구

- 소스 코드나 문서의 버전 관리, 이력 관리, 추적 등 변경 사항을 체계적으로 관리하는 기능을 제공하는 도구이다.
- CVS, SVN, Git

## (2) 서버 개발 프레임워크

### 1) 프레임워크 개념

- 프레임워크는 소프트웨어 개발에 있어 기본 구조를 제공한다.
- 반제품 상태의 제품을 토대로 필요한 서비스 컴포넌트를 사용하여 재사용성과 성능을 보장받을 수 있게 한다.
- 개발해야 할 애플리케이션의 일부분이 이미 구현되어 있어 동일한 로직의 반복 작업을 줄일 수 있다.
- 라이브러리처럼 사용하기 때문에 코드의 흐름을 제어 가능하다.
- 생산성 향상과 유지보수성 향상 등의 장점이 있다.

### 2) 서버 개발 프레임워크

- 서버 프로그램 개발 시 아키텍처 모델, 네트워크 설정, 요청/응답 등 구체적인 부분에 해당하는 설계와 구현을 손쉽게 처리하도록 기본적인 클래스와 인터페이스를 제공하는 소프트웨어이다.
- 대부분의 서버 개발 프레임은 MVC 패턴을 기반으로 하고 있다.



- 프로그램 언어에 기반하여 개발되고 제공되므로 제한적이다.

### 3) 대표적인 서버 개발 프레임워크

|               |                                                 |
|---------------|-------------------------------------------------|
| Spring        | JAVA를 기반으로 만들어진 프레임워크로 전자정부 표준 프레임워크의 기반 기술로 사용 |
| Node.js       | JavaScript를 기반으로 만들어진 프레임워크                     |
| Django        | Python을 기반으로 만들어진 프레임워크                         |
| Codeigniter   | PHP를 기반으로 만들어진 프레임워크                            |
| Ruby on Rails | Ruby를 기반으로 만들어진 프레임워크                           |

## 2. 서버 프로그램 구현

### (1) 보안 취약성 식별

#### 1) 보안 취약점 개요

- ① 보안 위협으로부터 안전한 소프트웨어를 개발하기 위해서는 개발과정에서 발생할 수 있는 보안 취약점을 식별하고 예방하여 위험을 최소화해야 한다.
- ② 보안 취약성은 소프트웨어 개발 보안 가이드를 참고하여 보안의 취약점을 점검한다.

#### 2) 보안 취약점 항목 (소프트웨어 개발 보안 가이드)

- ① 입력 데이터 검증 및 표현 : 프로그램 입력값에 대한 검증 누락 또는 부적절한 검증, 데이터의 잘못된 형식지정, 일관되지 않은 언어셋 사용 등으로 인해 발생하는 보안 약점
- ② 보안 기능 : 보안기능(인증, 접근제어, 기밀성, 암호화, 권한관리 등)을 부적절하게 구현 시 발생할 수 있는 보안 약점
- ③ 시간 및 상태 : 병렬 시스템이나 하나 이상의 프로세스가 동작되는 환경에서 시간 및 상태를 부적절하게 관리하여 발생할 수 있는 보안약점
- ④ 에러처리 : 에러를 처리하지 않거나, 불충분하게 처리하여 에러 정보에 중요정보가 포함될 때, 발생할 수 있는 보안약점
- ⑤ 코드 오류 : 타입 변환 오류, 자원(메모리 등)의 부적절한 반환 등과 같이 개발자가 범할 수 있는 코딩 오류로 인해 유발되는 보안약점

- ⑥ 캡슐화 : 중요한 데이터 또는 기능성을 불충분하게 캡슐화하거나 잘못 사용함으로써 발생하는 보안약점
- ⑦ API 오용 : 의도된 사용에 반하는 방법으로 API를 사용하거나, 보안에 취약한 API를 사용하여 발생할 수 있는 보안약점

## (2) API(Application Programming Interface)

### 1) API 개념

- ① API는 응용 프로그램 개발 시 운영체제나 프로그래밍 언어가 제공하는 소프트웨어 라이브러리를 이용할 수 있도록 정의해 놓은 인터페이스를 뜻한다.
- ② API는 애플리케이션을 만들기 위한 하위 함수, 프로토콜, 도구들의 집합으로 명확하게 정의된 다양한 컴포넌트 간의 통신 방법이다.
- ③ API는 프로그램 내에서 실행되기 위한 특정 서브루틴(Subroutine)에 연결을 제공하는 함수를 호출하는 것으로 구현된다.
- ④ 응용 프로그램은 API를 사용하여 운영체제 등이 가지고 있는 다양한 기능을 이용할 수 있으며, 같은 API를 사용해 만든 프로그램은 비슷한 인터페이스를 갖추고 있어, 사용자 입장에서는 새로운 프로그램의 사용법을 배우기 쉽다는 장점이 있다.

### 2) API 유형

- ① 비공개 API (Close API) : 권한이 있는 일부 사용자들에게만 주어진 API
- ② 공개 API (Open API) : 누구나 사용할 수 있도록 공개된 API

## 5. 배치 프로그램 구현

### (1) 배치 프로그램

#### 1) 배치 프로그램(일괄처리 : Batch Processing) 개념

- ① 컴퓨터 프로그램 흐름에 따라 순차적으로 자료를 처리하는 방식을 말한다.
- ② 요청이 있을 때마다 데이터 처리를 하는 것이 아니라 일정시간 데이터를 모았다가 일괄적으로 처리할 수 있다.
- ③ 배치(Batch)는 보통 정해진 특정한 시간(트리거)에 실행되는 경우도 많다.



- ④ 세금, 급여, 전화요금, 전기요금, 성적처리 등이 특정 시간대에 몰아서 처리하는 일에 주로 이용된다.

## 2) 배치 프로그램의 필수 요소

- ① 대용량 데이터 : 대용량의 데이터를 처리할 수 있어야 한다.
- ② 자동화 : 심각한 오류 상황 외에는 사용자의 개입 없이 동작해야 한다.
- ③ 견고함 : 유효하지 않은 데이터의 경우도 처리해서 비정상적인 동작 중단이 발생하지 않아야 한다.
- ④ 안정성 : 어떤 문제가 생겼는지, 언제 발생했는지 등을 추적할 수 있어야 한다.
- ⑤ 성능 : 주어진 시간 내에 처리를 완료할 수 있어야 하고, 동시에 동작하고 있는 다른 애플리케이션을 방해하지 말아야 한다.

## 3) 배치 프로그램 유형

- ① 정기 배치 : 정해진 시간에 처리된다. 주로 야간이나 컴퓨터가 쉬는 시점에 실행된다.
- ② 이벤트 배치 : 사전에 정의한 조건이 충족되면 자동으로 실행된다(트리거)
- ③ On-Demand 배치 : 사용자의 명시적인 요구가 있을 때마다 실행된다.





# 제5과목

## 정보시스템 구축관리



## Chapter 01 소프트웨어 개발 방법론 활용

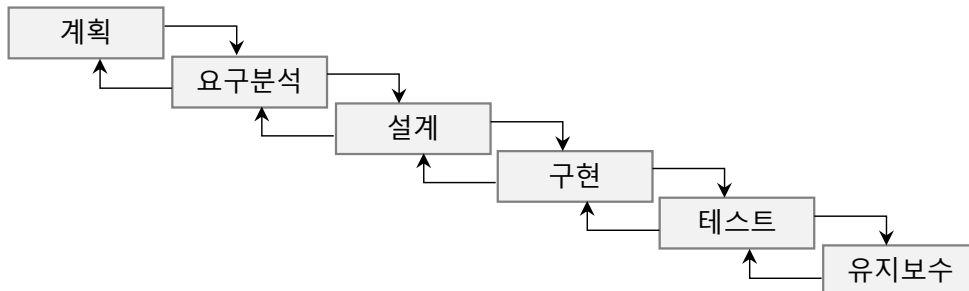
### 1. 소프트웨어 개발 방법론 선정

#### (1) 소프트웨어 생명주기 모형

소프트웨어 생명주기(Software Life Cycle)란 소프트웨어의 개발 및 운용, 유지보수 등의 각 단계를 진행하는 것으로 소프트웨어의 생성부터 소멸까지 변환되는 과정이다.

##### 1) 폭포수 모델

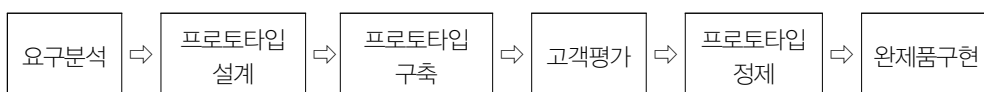
폭포수 모델은 고전적인 소프트웨어 생명주기 모델로 각 단계를 순차적으로 진행하고 단계별 마무리가 되어야 다음 단계로 진행하는 방식이다.



|    |                                                                                                                                            |
|----|--------------------------------------------------------------------------------------------------------------------------------------------|
| 장점 | <ul style="list-style-type: none"> <li>• 순차적으로 소프트웨어를 개발하는 전통적 모델로 사용사례와 성공사례가 많다.</li> <li>• 각 단계별 정의와 활동이 분명하고 단계별 산출물이 명확하다.</li> </ul> |
| 단점 | <ul style="list-style-type: none"> <li>• 개발 과정에 발생하는 사용자 요구 사항의 반영이 어렵다.</li> <li>• 단계별 오류없이 다음 단계로 진행한다는 것이 현실적으로 불가능하다.</li> </ul>       |

##### 2) 프로토타입 모델

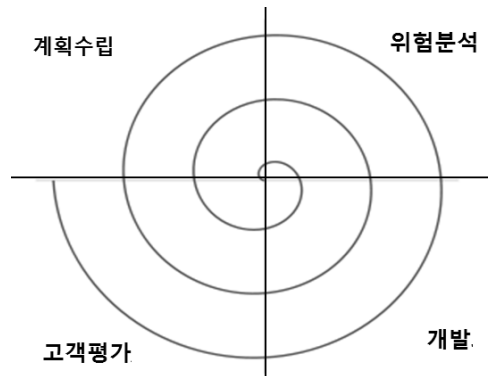
프로토타입 모델은 개발 과정에서 사용자의 요구사항을 잘 반영하기 위해서 개발중인 소프트웨어에 대한 시제품(prototype)을 만들어 제공하는 방식이다.



|    |                                                                                                                                                                               |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 장점 | <ul style="list-style-type: none"> <li>• 사용자 요구사항의 반영이 용이하다.</li> <li>• 의뢰자가 최종 결과물이 나오기 전에도 결과물의 일부나 모형을 볼 수 있다.</li> <li>• 프로토타입은 개발자나 의뢰자 모두에게 공동의 참조 모델을 제공한다.</li> </ul> |
| 단점 | <ul style="list-style-type: none"> <li>• 반제품의 프로토타입과 실제 소프트웨어와의 차이가 발생할 수 있다.</li> <li>• 단기간에 개발되어야 하기 때문에 비효율적인 언어나 알고리즘을 사용할 수 있다</li> </ul>                                |

### 3) 나선형 모델

- ① 나선형 모델(Spiral Model)은 가장 현실적인 모델로 폭포수 모델과 프로토타입 모델의 장점에 위험 분석 기능을 추가하였다.
- ② 나선 모양처럼 소프트웨어 개발과정을 반복하여 점진적으로 최종 소프트웨어를 개발하는 것으로, 점진적 모델이라고도 한다.
- ③ 소프트웨어 개발 과정에서 발생하는 위험을 관리하고 위험을 최소화하는 것을 목적으로 한다.



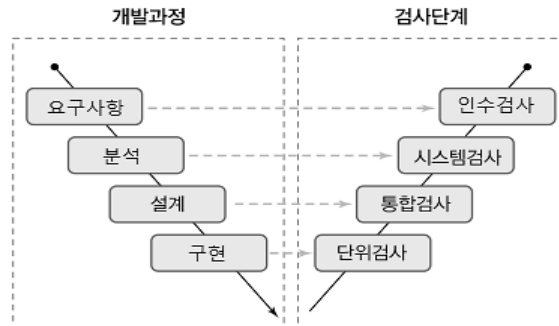
|    |                                                                                                                                                                                      |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 장점 | <ul style="list-style-type: none"> <li>• 비용이 많이 들거나 시간이 소요되는 대규모 시스템 개발에 적합하다.</li> <li>• 점진적인 반복 작업을 통해 새로운 요구사항을 추가할 수 있다.</li> <li>• 위험 요소를 사전에 제거함으로써 유지보수 과정이 필요 없다.</li> </ul> |
| 단점 | <ul style="list-style-type: none"> <li>• 위험 요소를 사전에 발견하지 못하면 개발 후 더 큰 문제가 발생할 수 있다.</li> </ul>                                                                                       |

### 4) V 모델

- ① Perry에 의해 제안되었으며 세부적인 테스트 과정으로 구성되어 신뢰도 높은 시스템을 개발하는데 효과적이다.
- ② 개발 작업과 검증 작업 사이의 관계를 명확히 들어내 놓은 폭포수 모델의 변형이라고 볼 수 있다.

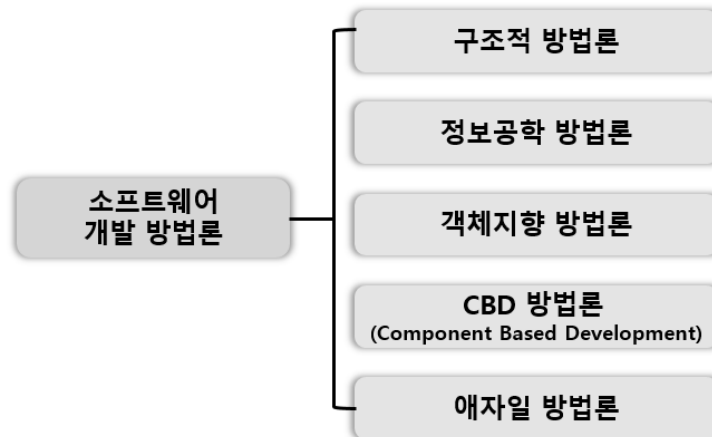


- ③ 폭포수 모델이 산출물 중심이라면 V 모델은 작업과 결과의 검증에 초점을 둔다.



## (2) 소프트웨어 개발 방법론

- ① 소프트웨어 개발 방법론은 소프트웨어 생명 주기의 각 작업을 어떻게 수행하느냐를 정의한 것이다.
- ② 소프트웨어 개발 과정의 수행 방법과 입력자료 및 산출물의 표현에 필요한 각종 기법과 도구들을 체계적으로 정리, 표준화한 것이다.



### 1) 구조적 방법론

- ① 정형화된 분석 절차에 따라 사용자 요구사항을 파악하여 문서화하는 방법론이다.
- ② 1960년대까지 가장 많이 사용되었던 소프트웨어 개발 방법론이다.
- ③ 쉬운 이해 및 검증이 가능한 프로그램 코드 생성이 목적이다.
- ④ 분할과 정복(Divide and Conquer) 원리를 적용한 하향식 방식이다.

|                |         |                               |
|----------------|---------|-------------------------------|
| 제5과목<br>데이터베이스 | 적중 예상문제 | Chapter 01<br>소프트웨어 개발 방법론 활용 |
|----------------|---------|-------------------------------|

1. 생명주기 모형 중 가장 오래된 모형으로 많은 적용 사례가 있지만 요구사항의 변경이 어렵고 각 단계의 결과가 확인 되어야 다음 단계로 넘어갈 수 있는 선형 순차적, 고전적 생명주기 모형이라고도 하는 것은?

① Waterfall Model

③ Cocomo Model

② Prototype Model

④ Spiral Model

정답  
해설

① 폭포수 모형(Waterfall Model) : 고전적 생명주기 모형으로 선형 순차적으로 진행한다. 각 단계별 산출물이 명확하고 적용 사례가 많다.

2. 폭포수 모형의 특징으로 거리가 먼 것은?

① 개발 중 발생한 요구사항을 쉽게 반영할 수 있다.

② 순차적인 접근방법을 이용한다.

③ 단계적 정의와 산출물이 명확하다.

④ 모형의 적용 경험과 성공사례가 많다.

정답  
해설

① 폭포수 모형(Waterfall Model)의 단점은 개발 중 발생하는 사용자 요구사항 반영이 어렵다.

3. 소프트웨어 생명주기 모델 중 나선형 모델(Spiral Model)과 관련한 설명으로 틀린 것은?

① 소프트웨어 개발 프로세스를 위험 관리(Risk Management) 측면에서 본 모델이다.

② 위험 분석(Risk Analysis)은 반복적인 개발 진행 후 주기의 마지막 단계에서 최종적으로 한 번 수행해야 한다.

③ 시스템을 여러 부분으로 나누어 여러 번의 개발 주기를 거치면서 시스템이 완성된다.

④ 요구사항이나 아키텍처를 이해하기 어렵다거나 중심이 되는 기술에 문제가 있는 경우 적합한 모델이다.

정답  
해설

② 나선형 모형에서 위험 분석은 마지막에 한번만 수행하는 것이 아니라 매 주기마다 수행되어야 한다.

## 정보처리기사 필기 한권으로 끝내기

편저자 메인에듀 정보기술연구소 편저  
제작유통 메인에듀(주)  
초판발행 2024년 04월 02일  
초판인쇄 2024년 04월 02일  
마케팅 메인에듀(주)  
주소 서울시 강동구 성안로 115, 3층  
전화 1544-8513  
정가 29,000원

I S B N 979-11-89357-56-6

# 기술자격시험 합격은 메인에듀

01

NCS반영  
-  
출제기준  
전면개편

02

비전공자도  
합격이 가능  
한 필수교재

03

핵심이론  
단원별예상문제  
수록

## 메인에듀 동영상강의

메인에듀는 전문국가 자격증 사회복지사, 요양보호사, 손해평가사, 청소년상담사, 관광동역안내사, 물류관리사, 재경관리사, 기술/IT 자격증 정보처리기사, 컴퓨터 활용능력, 그리고 유통관리사, CS리더스관리사, 텔레마케팅관리사, 스포츠지도사, SMAT(서비스경영자격), 한국어, 한자, 직업상담사, 전산세무회계, 사회조사분석사 등 학위취득에 필요한 국가공인 자격증 과정을 운영하고 있습니다. 향후 더 많은 국가공인 자격증 과정을 제공할 예정입니다.

## 인터넷/모바일

