제 1장 시스템 개발과정의 이해

1.1 소프트웨어 공학

1.2 시스템과 시스템 개발자

1.3 SDLC 모형

1.4 프로토타입 모형

1.5 프로젝트 관리

학습 목표

* 소프트웨어 위기와 소프트웨어 공학의 출현 배경을 이해한다
* 시스템(소프트웨어) 개발 과정에 참여하는 사람들에 대해 알아본다
* 시스템 개발 단계를 이해하기 위해 SDLC 모형 및 프로토타입 모형을 학습한다
* 프로젝트 관리의 개념을 이해하고 절차를 학습한다

**1.1 소프트웨어 공학**

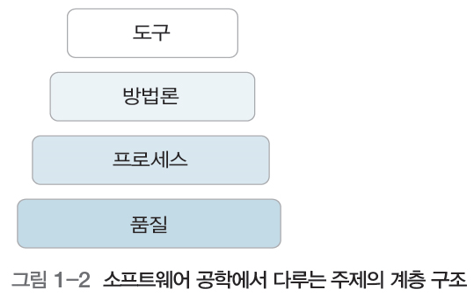
**소프트웨어 위기**

* 주먹구구식 개발로 개발 기간이 지연되거나 실패로 돌아감

**소프트웨어 공학의 출현**

* IEEE의 <소프트웨어 공학>에 대한 정의  
  소프트웨어 개발, 운용, 유지보수 및 파기에 대한 체계적인 접근 방법
* 품질이 좋은 소프트웨어를 최소한의 비용으로 계획된 일정에 맞추어 개발하는 것

**소프트웨어 공학의 계층 구조**



**도구**

* 프로그램 개발 과정에서 사용되는 여러 가지 방법을 자동화한 것
* CASE는 소프트웨어 개발 전 단계를 지원하는 대표적인 도구

**방법론**

* 프로세스 중심 방법론
  + 1970년대 제시됨
  + 자료의 변환 과정과 프로세스를 강조하여 프로그램을 개발하는 방법
* 자료 중심 방법론
  + 프로그램을 개발할 때 사용할 자료를 규명하고, 자료와 자료 간의 관계를 분석한 후 자료구조를 정의하고 이를 토대로 프로세스 구조를 고안하는 방법
  + 데이터베이스에 기반한 쿼리 중심의 프로그램 방식
* 객체지향 방법론
  + 객체를 캡슐화 함으로써 좀 더 쉽게 프로세스의 모듈화, 정보 은닉, 코드 재사용의 효율성을 꾀할 수 있는 방법
  + 프로세스 중심 방법론과 자료 중심 방법론의 장점을 묶어 진화한 방법론

**프로세스**

* 소프트웨어 개발에 필요한 작업 이름, 작업 내용, 결과물, 절차, 지시사항 등을 작업 사이의 선후 관계와 더불어 나타낸 것
* 다음 네 가지 영역으로 구분됨
  + 소프트웨어 명세
  + 소프트웨어 개발
  + 소프트웨어 검증
  + 소프트웨어 진화

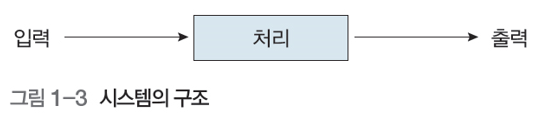
**품질**

* 소프트웨어 품질을 평가하는 기준은 다음과 같음
  + 정확성
  + 유지보수성
  + 무결성
  + 사용성

**1.2 시스템과 개발자**

**시스템의 개념**

* + 컴퓨터에 의해 처리가 가능한 형태로 자료를 변환하여 입력하고, 그 자료를 저장, 처리, 가공하여 필요한 시점에 정보를 출력할 수 있도록 설계되고 구현된 정보체계

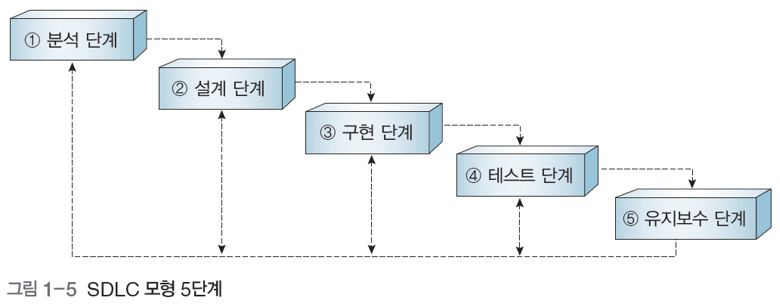


**시스템 개발에 참여하는 사람들**

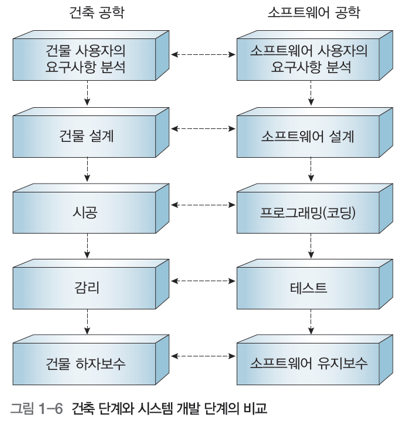


**1.3 SDLC 모형**

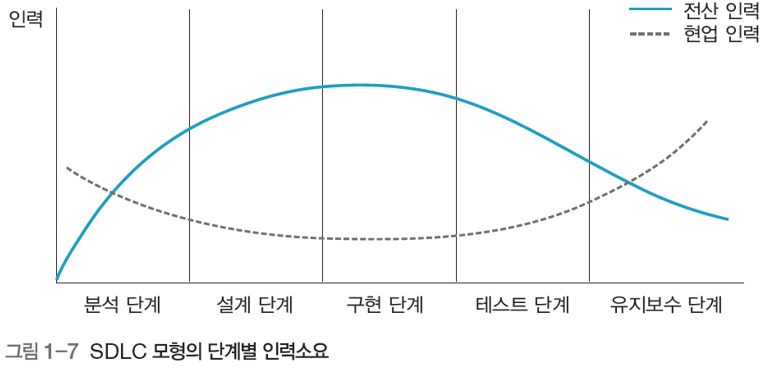
**SDLC 모형의 5단계**



**SDLC 모형과 건축 과정의 비교**



**SDLC 모형의 단계별 인력소요**



**SDLC 모형의 장점**

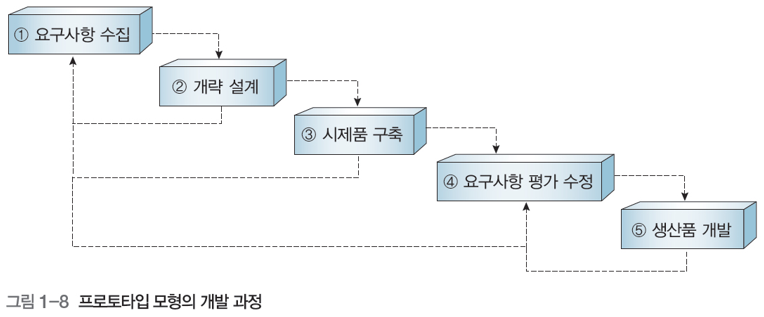
* 시스템 개발의 각 단계가 비교적 명확
* 각 단계들 간에 유기적인 연관성을 가지고 있어 쉽게 적용할 수 있음

**SDLC 모형의 단점**

* 충분한 분석을 기반으로 개발이 진행되지 않았을 경우 테스트 단계 또는 유지보수 단계에서 문제점이 노출되어 이를 개선하는 데 더 많은 비용과 시간이 소요됨
* 대형 프로젝트의 경우 긴 개발기간 동안 외부환경이나 내부 정책이 변화할 소지가 크고, 이를 개선하기 위해 이전 단계로 되돌아가 변경관리를 하므로 막대한 시간과 비용이 들어감

**1.4 프로토타입 모형**

**프로토타입 모형**



**프로토타입 모형의 장점**

* 개발 초기에 미리 결과물을 확인할 수 있다는 점에서 사용자의 이해를 도움
* 개발의 초기 단계에서 수정・보완할 사항을 미리 파악할 수 있음
* 분석 및 설계 과정에 사용자가 동참하여 즉각적인 피드백을 줄 수 있음

**프로토타입 모형의 단점**

* 일회적 프로젝트나 대규모 프로젝트의 개발에는 적용하기 쉽지 않음
* 불완전한 요구사항을 바탕으로 시제품이 만들어지기 때문에 결과적으로 불완전한 시스템을 산출하여 수정과 보완에 많은 인력과 시간이 투입됨

**1.5 프로젝트 관리**

**프로젝트 관리자의 활동**

* 제안서 작성
* 프로젝트 계획과 일정 수립
* 프로젝트 비용 산정
* 프로젝트 모니터링과 중간평가 실시
* 실무자 선정과 평가
* 보고서 작성과 발표

**프로젝트 계획과 함께 수립해야 할 계획**

* 품질 계획: 품질을 위한 계획과 표준을 설명
* 검증 계획: 시스템 검증을 위한 접근 방법, 자원, 일정을 설명
* 구성관리 계획: 구성관리 방안과 구조를 설명
* 유지보수 계획: 유지보수를 위한 요구사항, 비용, 노력을 설명
* 인력개발 계획: 팀원들의 경험 축적과 기술개발 설명

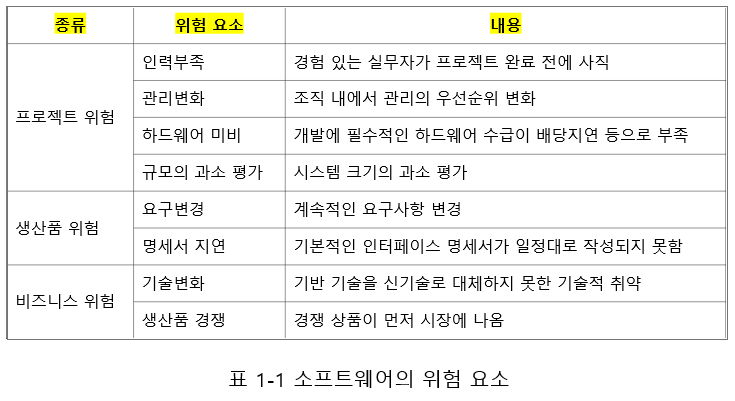
**품질 관리**

* 개발주기 전체에 걸쳐 시행하는 검열, 검토, 테스트 등을 말함

**품질은 설계 품질과 적합 품질로 나뉨**

* 설계 품질: 사용자의 요구를 반영하여 설계자가 시스템 설계 시 설계 문서에 적용하는 품질 특성
* 적합 품질: 구현할 때 설계 명세대로 이행하는 정도

**위험 관리**



**위험 관리 프로세스**

* 1단계 위험 인식
* 2단계 위험 분석
* 3단계 위험 계획
* 4단계 위험 모니터링

제 2장 시스템 분석/설계 방법론과 관련 문서

2.1 시스템 분석의 중요성

2.2 시스템 분석/설계 방법론

2.3 요구사항 분석

2.4 구조적 검토 회의

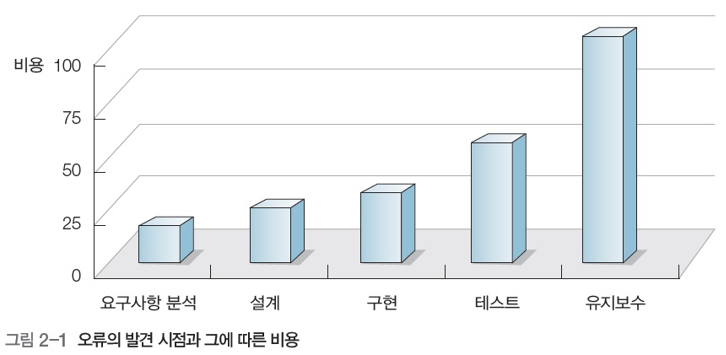
2.5 시스템 분석/설계 문서

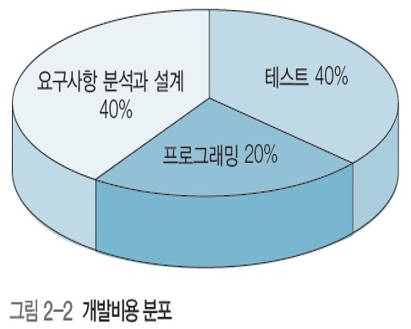
학습 목표

* 시스템 생명주기 모형의 첫 단계인 시스템 분석의 중요성을 인식한다
* 시스템 분석 및 설계 방법론의 개괄적인 검토를 통해 각 방법론의 특징과 장단점을 학습한다
* 시스템 분석 단계에서 해야 하는 중요한 절차인 요구사항 분석의 세부 내용을 이해한다
* 시스템 분석 및 설계 과정에서 산출되는 문서에 대해 알아본다

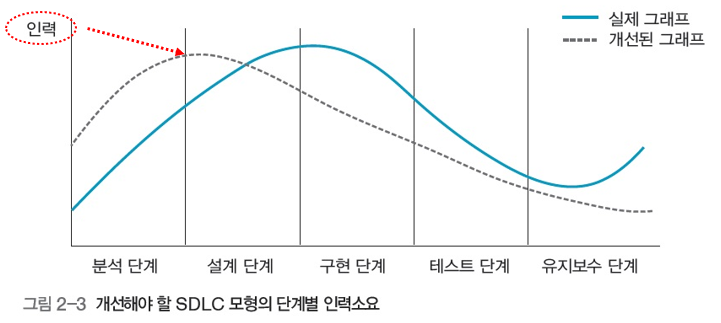
**2.1 시스템 분석의 중요성**

**요구사항 분석과 설계의 중요성**

* 유지보수 단계에서 오류가 발생하면 더 많은 추가비용이 발생함  
  
* 소프트웨어 개발 비용은 프로그래밍 이전 단계에서 40~50%가 소요됨
  + 소프트웨어 개발에 있어 요구사항 분석과 설계가 체계적으로 이루어지지 않으면 좋은 품질을 기대하기 어려움



**이상적인 SDLC 모형**

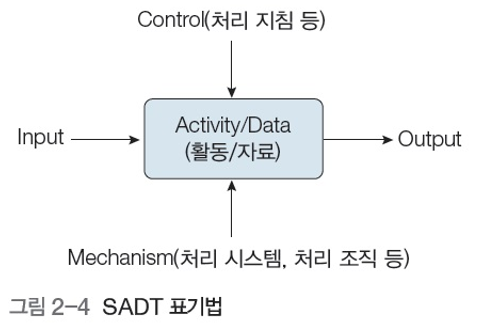


**2.2 시스템 분석/설계 방법론**

**기능 모델링: 구조적 분석 방법론**

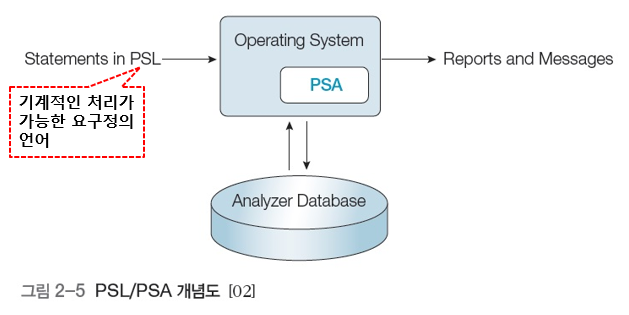
* 구조적 분석 방법론(by DeMarco)
  + 1980년대부터 널리 활용되기 시작
  + 현재 요구사항 분석에 가장 많이 활용하는 기법
  + 사용하는 도구로는 자료흐름도(DFD), 자료사전(DD), 소단위 명세서 등이 있음
* 구조적 분석 방법론의 특징
  + 매우 간결함(Concise)
  + 이해하기 쉬움(Understandability)
  + 검증이 가능함(Verifiable)
  + 체계적임(Organized)

**기능 모델링: SADT**

* SADT(Structured Analysis and Design Technique) by Softech사
  + 본질적으로는 그래프 언어이며 시스템 구조를 계층적으로 기술
* SADT는 다음과 같은 사항을 수행하기 위한 방법론을 제공
  + 대규모이고 복잡한 문제를 구조적으로 생각하게 됨
  + 각 작업자의 노력과 역할을 효과적으로 나누고 또 통합해서 팀으로서 효과적으로 활동하게 함
  + 명료하고 정확한 표기법에 의해서 인터뷰, 분석, 설계의 결과를 전달
* SADT 표기법  
  

**기능 모델링: PSL/PSA by Michigan Univ. for IBM**

* PSL/PSA(Problem Statement Language/Problem Statement Analyzer)
  + 정보처리 시스템에 대한 요구사항 분석과 문서화를 지원하는 시스템

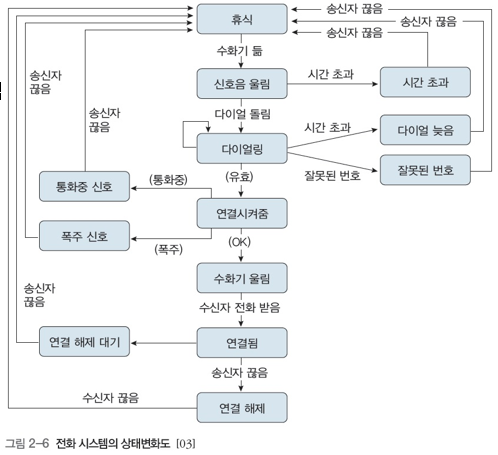


**동적 모델링: 실시간 시스템**

* 실시간 시스템
  + 제한된 시간 내에 외부에서 주어진 사건에 응답하고 자료를 처리하는 시스템
  + 시스템의 제어흐름, 상호작용, 동작의 순서 다루기 등이 포함되어 있음
* 시스템 시스템의 예
  + 통신 시스템, 비행기 운행 관리 시스템, 자동차 속도 조절장치, 원자력 발전소의 원자로 제어장치, 군사용 미사일 시스템 등

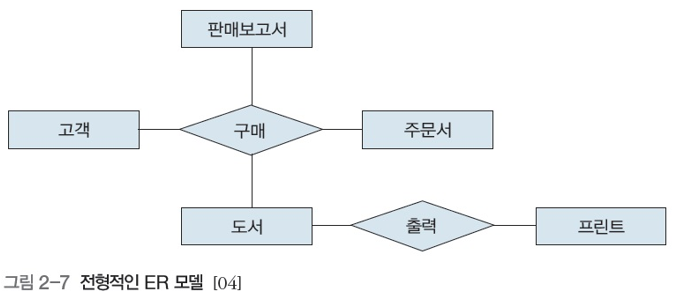
**동적 모델링: 상태 변화도(State Transition Diagram)**

* 상태 변화도(SDT)
  + 시스템의 제어흐름과 동작의 순서를 나타낸 도식



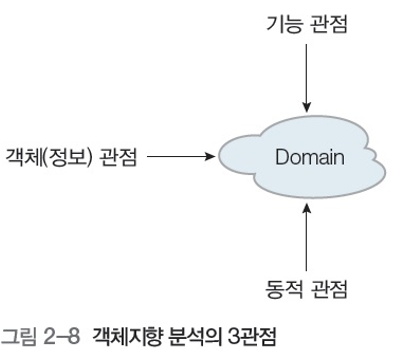
**정보 모델링(Information Modeling)**

* 시스템에 필요한 엔티티를 정의하고 이들 엔티티 사이의 연관성을 규명
* 대표적인 도구: ERR(Enhanced Entity-Relationship) 모델
  + 1976년 피터 첸에 의해 제안된 ER 모델에 데이터의 계층 구조를 추가하여 확장시킨 것



**객체지향 모델링(Object-Oriented Modeling)**

* 데이터와 행위를 하나로 묶어 객체를 정의하고 추상화 시키는 작업



**2.3 요구사항 분석**

**요구사항 조사 방법**

* 관찰 조사
  + 실제 현업 부서를 방문하여 부서의 작업 환경, 현업의 처리 절차, 개선할 사항 등을 관찰하여 정량적인 정보(빈도, 수량, 비용 등)를 수집하는 방법
* 질문지 조사
  + 체계적으로 설계된 질문지를 이용해 필요한 정보를 수집하는 방법
  + 직접 관찰하거나 면담하기 어려운 부서의 담당자에게서도 손쉽게 정보를 수집할 수 있음
* 면담(인터뷰) 조사
  + 가장 보편적이며 중요한 정보수집 방법
  + 시스템 분석가와 현업부서 담당자 간의 직접 대화를 통해 현행 시스템의 문제점 및 개선 요구사항 등을 파악할 수 있는 방법

**요구사항 조사 내용**

* 조직에 대한 정보
  + 조직의 연혁, 조직도, 업무 분장 및 규정 등을 수집・분석함
* 현재 사용 중인 제반 서식
  + 부서에서 현재 사용 중인 제반 서식을 빠짐없이 수집・분석함
    - 데이터베이스 설계 및 입력과 출력 설계의 기본이 되는 정보를 제공함
* 시스템 인프라
  + 서버의 가용 자원, 성능 등을 비롯하여 네트워크 구축 상태 및 데이터베이스 사용 등을 조사・분석함
* 현재 운영 중인 시스템
  + 현재 운영 중인 시스템이 있는 경우, 시스템의 지원 범위를 비롯하여 운영자 매뉴얼 등을 수집・분석함

**2.4 구조적 검토회의**

**기존 검토회의의 문제점**

* 참석자의 역할과 책임이 불명확
* 검토회의의 효율적인 진행법 부재
* 산출물보다 사람 평가 경향
* 검토회의 목적이 불분명

**구조적 검토회의의 효과**

* 역할과 책임을 분명히 정의
* 검토회의 이전 단계, 진행 단계, 이후 단계로 구분되어 작업 수행
* 참여자들의 심리적 갈등 해소
* 분명한 목표
* 개발 초기 산출물이 안고 있는 문제점 발견 가능
* 산출물의 완전성, 일관성, 이해 가능도 확인
* 각자가 가지고 있는 개념과 기법의 상호 교환 가능
* 프로젝트 진척도 측정 가능
* 공동 책임 의식 고취

**구조적 검토회의 참석자의 역할**

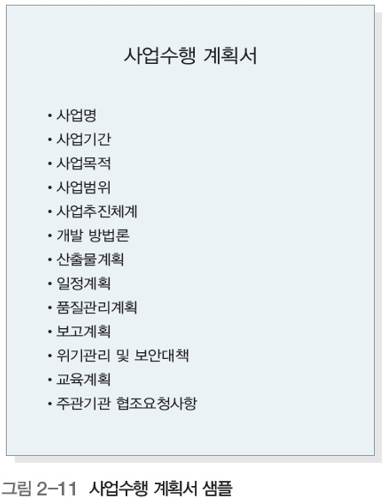
* 산출물 발표자
  + 검토회의 참석자들에게 산출물을 설명함
* 중재자
  + 검토회의가 효율적이고 순조롭게 진행되도록 회의를 계획하고 회의 진행을 조정함
* 서기
  + 검토회의에서 발견된 오류나 기타 문제점들을 기록
* 산출물 검토자
  + 장래의 유지 관점에서 산출물의 검토
  + 표준화 요원과 유지보수 요원이 있음
* 사용자 대표
  + 요구사항이 충족되었는지 확인, 프로젝트 진척 사항 피드백과 질적 문제에 대한 조언을 함

**2.5 시스템 분석/설계 문서**

**제안 요청서**

* 전문 개발업체에서 개발을 의뢰할 경우 작성하는 문서
* 제안 요청서의 구성요소
  + 제안업체의 일반사항 - 개발 방법론
  + 제안목적 - 일정계획
  + 사업명 - 투입인력 계획
  + 사업기간 - 기술이전 계획
  + 사업목적 - 제안업체의 사업수행 실적
  + 사업범위 - 제안금액(별지)
  + 사업추진체계

**사업수행 계획서**

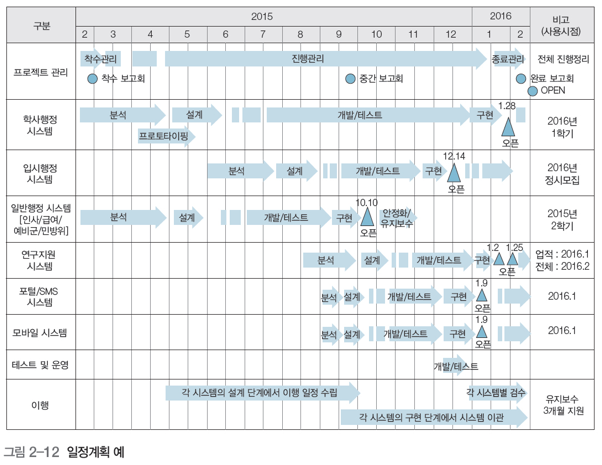
* 제안요청서를 바탕으로 사업수행에 필요한 제반 계획사항들을 명확히 기술하는 문서  
  

**사업수행 계획서의 산출물 계획**





**사업수행 계획서의 일정계획**



**사업수행 계획서의 품질관리계획**



**사업수행 계획서의 보고계획**



**그 외의 사업수행 계획서 내용**

* + 위기관리 및 보안대책
  + 교육계획
  + 주관기관 협조요청사항

**요구사항 명세서**

* 요구사항 명세서의 구성요소
  + 기능 요구사항 - 문서화 요구사항
  + 성능 요구사항 - 보안 요구사항
  + 인터페이스 요구사항 - 이식성 요구사항
  + 운영 요구사항 - 품질 요구사항
  + 자원 요구사항 - 신뢰성 요구사항
  + 검증 요구사항 - 유지보수성 요구사항
  + 인수 테스트 요구사항 - 안전 요구사항

**설계 명세서**

* 설계 과정에서 산출된 각종 설계 문서를 뜻함
* 기본적으로 시스템 구조도, 데이터베이스 설계 문서, 프로그램 작성 지침, 인터페이스 설계 문서 등이 포함됨

제 3장 구조적 분석 방법론

3.1 구조적 분석의 원리

3.2 모형화 도구의 특성

3.3 구조적 분석 모형화 도구

3.4 구조적 분석 절차

학습 목표

* 구조적 분석 방법론의 일반적 원리를 이해한다
* 구조적 분석 방법론에서 사용하는 모형화 도구의 특성을 이해한다
* 구조적 분석 방법론에서 사용하는 모형화 도구인 자료흐름도, 자료사전, 소단위 명세서에 대해 살펴본다
* 구조적 분석 방법론에 의한 시스템 분석의 4단계를 살펴본다

**3.1 구조적 분석의 원리**

**구조적 분석 방법론의 개요**

* 구조적 분석 방법론
  + 도형화된 도구를 이용해 정형화된 분석 절차에 따라 사용자 요구사항을 파악하고 문서화하는 분석 기법
  + 요던(Yourdon) 등에 의해 개발되어 보급된 이후 지금도 널리 사용되고 있음
  + 사용하는 도구로는 자료흐름도, 자료사전, 소단위 명세서 등이 대표적임
  + 하향식 기능 분해 기법 등을 사용하는 특성이 있음

**구조적 분석의 기본원리**

* 추상화(Principle of Abstract) 원칙:  
  특정 대상에 대해 실체로부터 분리된 개념이나 관점  
  ‘어떻게’가 아닌 ‘무엇’으로 정의하는 것
* 정형화(Principle of Formality) 원칙:  
  형식화를 하여 프로젝트 제어에 활용
* 분할 정복의 개념(Divide-and-Conquer Concept):  
  복잡한 프로세스를 좀 더 작고 독립적인 서브시스템으로 분할
* 계층적 구조의 개념(Hierachical Structure Concept):  
  분할된 서브시스템들의 상하관계 및 좌우관계를 정의하는 것

**3.2 모형화 도구의 특성**

**모형화 도구를 사용하는 이유**

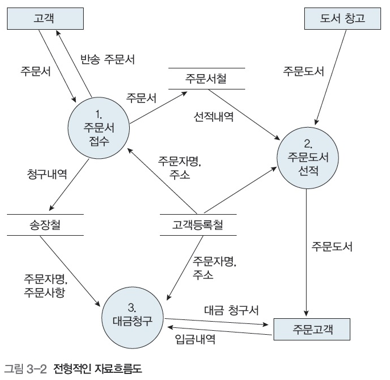
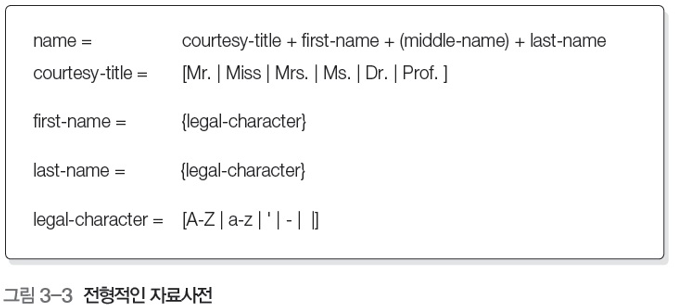
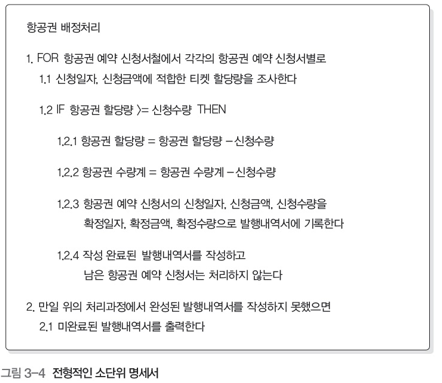
* 비용을 줄이고 위험도를 최소화할 수 있음
* 시스템 분석가가 사용자의 환경을 정확히 이해하고 문서화하였는지 검증할 수 있음
* 실제 시스템을 구축하고 설치하는 것보다 낮은 비용으로 모형을 구축할 수 있음
* 시스템에 대한 깊은 지식이 없더라도 쉽게 이해하도록 할 수 있음
* 시스템을 모형화하고자 하는 사람의 생각을 정형화할 수 있음
* 비유 사례: 신차 개발 프로세스 중 클레이 모형을 만들어 점검하는 것

**모형화 도구의 특성**

* 도형적 모형: 시스템을 설명할 때, 텍스트보다는 도형을 통해 더 잘 설명할 수 있음
* 하향식 분할 모형: 시스템 각각의 구성 부분을 독자적으로 표시하고, 시스템 모형을 한 부분에서 다른 부분으로 간단히 연결할 수 있어야 함  
  사례) 큰 상세지도의 구석에 전체지도를 표기하는 방법
* 최소 중복 모형: 중복을 최소화하여 시스템을 모형화함
* 투명적 모형: 좋은 모형의 조건은 이해하기 쉬워야 하며, 추상화한 모형을 보면서 실세계를 자연스럽게 인식할 수 있어야 함
* 다양한 모형: 시스템의 특성에 따라 더 적합한 모형화 도구를 선택해 사용해야 함

**3.3 구조적 분석 모형화 도구**

**구조적 방법론의 3가지 모형화 도구**

1. 자료흐름도(DFD: Data Flow Diagram):  
   구조적 시스템 분석의 가장 중요한 모형화 도구로 네트워크형 구조를 가짐  
   
2. 자료사전(DD: Data Dictionary):  
   자료흐름도에 기술된 모든 자료들에 대한 사항을 정의하는 도구  
   
3. 소단위 명세서(Mini-Spec):  
   입력 자료를 출력 자료로 변환하기 위해 수행되어야 하는 정책이나 규칙을 구체적으로 기술하는 도구 -> Use Case라고도 함  
   

**3.4 구조적 분석 절차**

**구조적 분석의 4단계 절차**

* 1단계: 현 물리적 모형화(CPM, Current Physical Modeling)  
  사용자의 업무수행 절차 및 환경을 있는 그대로 모형화하는 단계  
  사례) 앞의 DFD 도면을 참고하면, 화살표로 표시된 행위들을 대상(Object)을 참조하여 업무처리 내역을 기술
* 2단계: 현 논리적 모형화(CLM, Current Logical Modeling)  
  현 물리적 모형에 존재하는 구현 의존적인 물리적 특성을 제거해 모형화하는 단계  
  사례) 앞의 DFD를 참고하면, 업무처리 내역을 분류하여 단순화하고 관련 서류들의 공통점을 찾아 새로운 시스템상의 이름을 부여
* 3단계: 신 논리적 모형화(NLM, New Logical Modeling)  
  새로운 시스템에서 수행될 모든 기능 및 이에 필요한 자료에 대한 모형을 구축하는 단계  
  사례) 도서목록을 DB화하여 문서에 추가 기록하고, 바코드 시스템을 도입하여 프로세스를 기존 논리구조에 추가한다
* 4단계: 신 물리적 모형화(NPM, New Physical Modeling)  
  현실적인 물리적 환경을 감안해 최종 적용할 모형을 제시하는 단계  
  사례) 3단계의 신 논리적 모형을 참조하여 실제 사용할 DB선정 및 적용, 신분증 및 도서에 도입할 바코드 시스템의 구체화 및 새로운 시스템에서의 프로세스를 구체화하여 기술한다

제 4장 자료 흐름도

4.1 자료흐름도의 특징

4.2 자료흐름도의 구성요소

4.3 자료흐름도의 작성 원칙

4.4 자료흐름도의 작성 절차

학습 목표

* 구조적 분석 방법론의 모형화 도구인 자료흐름도의 특징을 이해한다
* 자료흐름도의 구성요소인 네 개의 심볼을 식별하여 작성할 수 있도록 학습한다
* 자료흐름도의 작성 과정에서 놓치기 쉬운 작성 원칙들을 살펴본 후 적용한다
* 자료흐름도의 작성 사례를 평가하고 개선할 수 있도록 다양한 사례를 검토한다

**4.1 자료흐름도의 특징**

**자료흐름도의 특징**

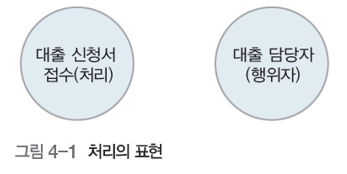
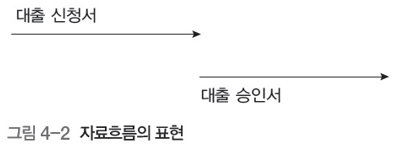
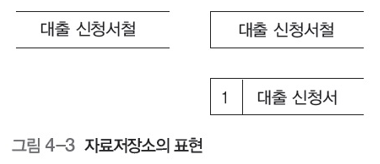
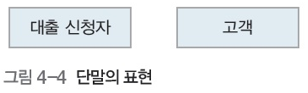
* 도형을 이용한 그림 중심의 표현
* 하향식 분할의 원리를 적용
* 다차원적
* 자료의 흐름에 중점을 두는 분석용 도구
* 제어의 흐름은 중요시하지 않음 즉, 데이터 관점에서 표현

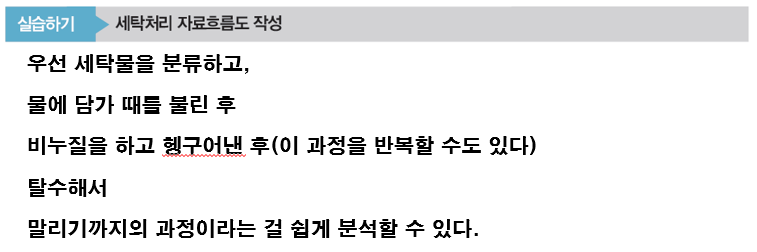
**자료흐름도의 작성 효과**

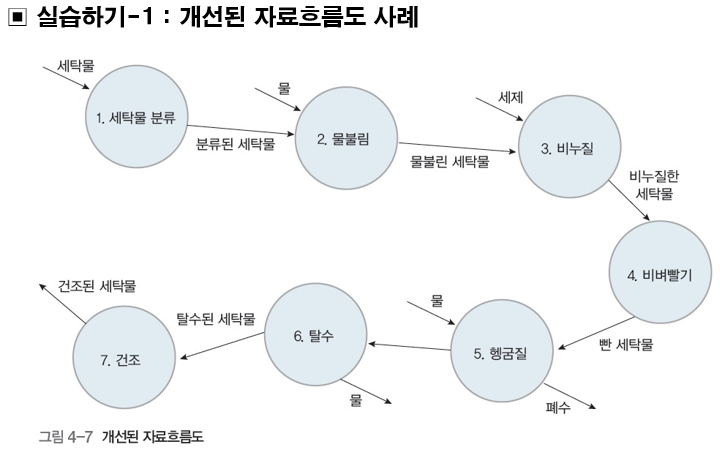
* 사용자의 업무 및 요구사항을 쉽게 문서화할 수 있음
* 사용자와 분석가 사이의 의사소통을 위한 공용어 역할을 함
* 일관성 있고 정확한 사용자의 요구사항을 파악할 수 있는 요구분석용 도구의 역할을 수행

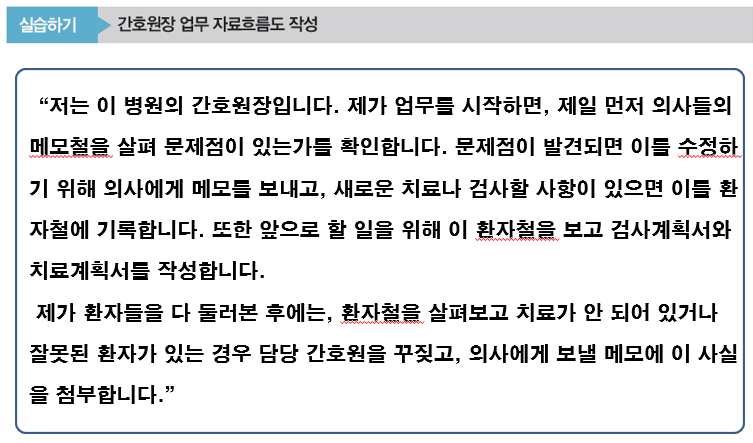
**4.2 자료흐름도의 구성요소**

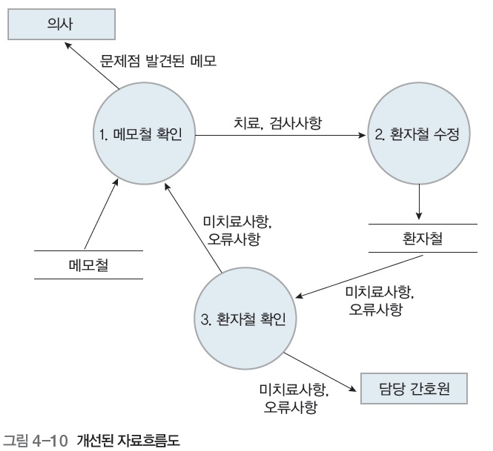
**자료흐름도의 4가지 구성요소**

* 처리: 입력되는 자료흐름을 출력되는 자료흐름으로 변환하는 것  
  
* 자료흐름: 자료흐름도에서 구성요소들 간의 접속관계를 나타냄  
  
* 자료저장소: 머물고 있는 자료군의 집합  
  
* 단말: 상세한 자료흐름도를 이해할 수 있게 사각형의 단말을 사용함  
  



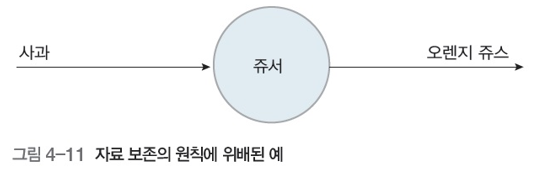
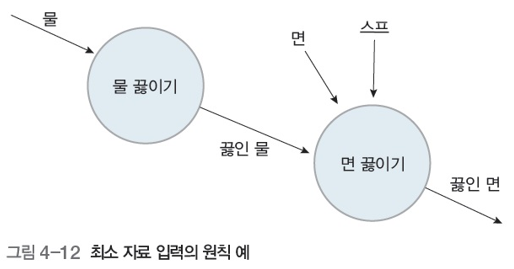
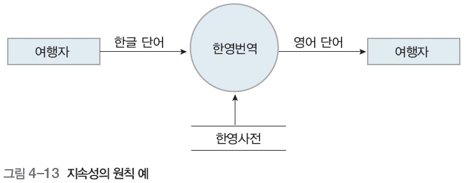
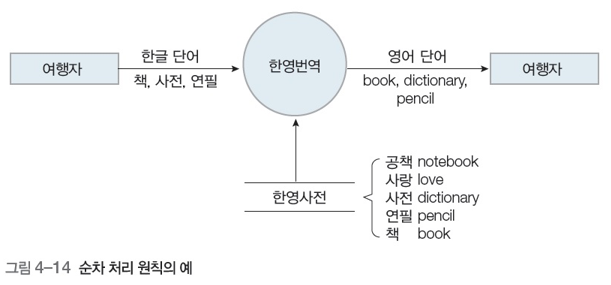
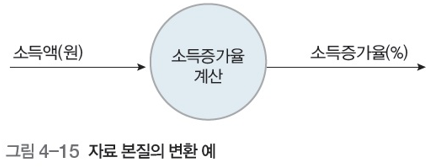
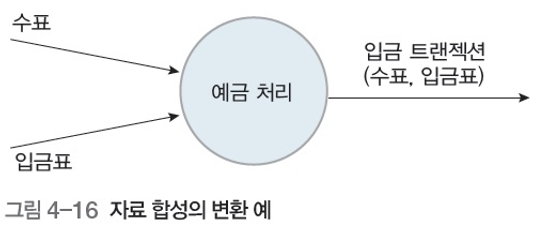
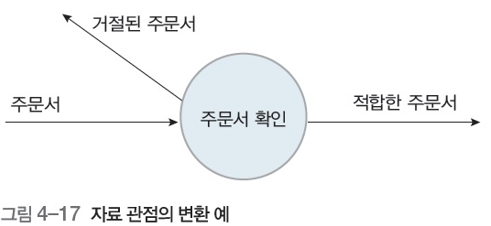
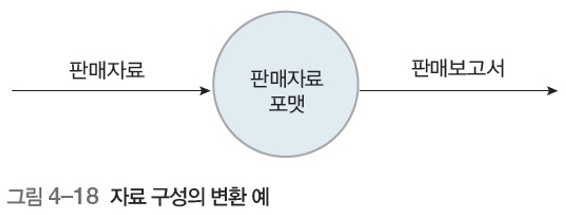




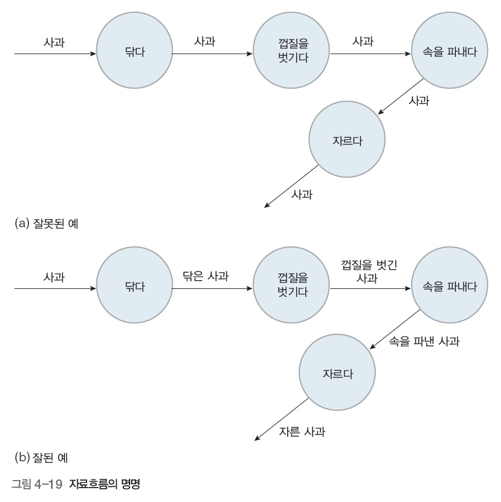
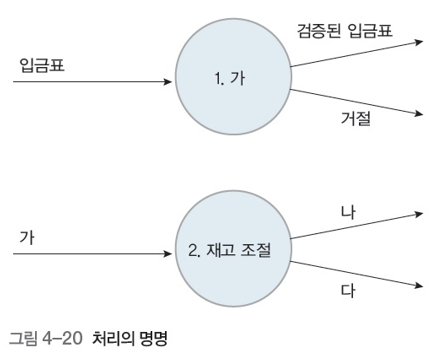


**4.3 자료흐름도의 작성 원칙**

**자료흐름도 작성의 7가지 원칙**

* 자료 보존의 원칙(Conservation Rule)
  + 어떤 처리의 출력 자료흐름은 반드시 입력 자료흐름을 이용해 생성해야 함  
    
* 최소 자료 입력의 원칙(Parsimony Rule)
  + 어떤 처리가 출력 자료흐름을 산출하는 데 반드시 필요로 하는 최소의 자료흐름만 입력해야 하는 것  
    
* 독립성의 원칙(Independence Rule)
  + 자기의 처리는 오직 자신의 입력 자료와 출력 자료 자체에 대해서만 알면
  + 독립성의 원칙은 유지보수가 쉬운 시스템을 산출할 수 있음
* 지속성의 원칙(Persistence Rule)
  + 어떤 자료흐름을 기다릴 때를 제외하고는 다시 시작하거나 멈춰서는 안 됨  
    
* 순차 처리의 원칙(Ordering Rule)
  + 처리에 입력되는 자료흐름의 순서는 출력되는 자료흐름에서도 지켜져야 함  
    
* 영구성의 원칙(Permanence Rule)
  + 자료저장소의 자료는 제거되지 않아야 함
* 자료 변환의 원칙(Nature of Change)
  + 자료 본질의 변환: 일반적으로 입력 자료흐름에 편집, 계산 등을 해 출력 자료흐름을 산출하는 것  
    
  + 자료 합성의 변환: 두 개 이상의 입력 자료흐름에 대해 자료합성의 변환이 발생할 수 있음  
    
  + 자료 관점의 변환: 입력 자료흐름이 동일하게 출력자료흐름으로 나타나게 됨  
    
  + 자료 구성의 변환:  
    출력 자료가 입력 자료와 동일하지만, 자료의 구성 형태가 변환됨  
    구성의 변환은 포맷팅 또는 정렬 등을 위한 처리가 필요함  
    

**4.4 자료흐름도의 작성 절차**

* 시스템 경계의 입출력 식별
  + 순수 입력과 출력을 선정하는 것은 분석의 대상이 무엇이어야 하는가에 대한 결정과 관련됨
* 시스템 경계 내부의 작성
  + 자료흐름도를 작성하는 것은 현재 사용자 영역을 최초로 문서화하는 것
  + 현재의 업무수행 방식을 그대로 기술해야 함
* 자료흐름의 명명
  + 각각의 자료흐름에 대해 새로운 명칭을 부여
  + 명칭을 부여할 때는 전체의 자료흐름에 적용될 수 있는 이름을 부여
  + ‘자료’, ‘정보’ 등과 같이 의미 없는 명칭은 부여하지 않음
  + 전체로 통합될 수 없는 자료항목을 하나의 자료흐름으로 만들지 않음  
    
* 처리의 명명
  + 하향식 접근방법: 자료흐름에 중점을 두어 명명하고, 처리에 관한 명명하는
  + 상향식 접근방법: 처리에 먼저 중점을 두는 접근방식  
    
  + 처리의 명칭은 처리내용에 적합하도록 명명해야 함
  + 처리의 이름은 동사형 명사와 단일 직접 목적어를 사용함(두 개의 동사가 필요하다면 처리를 분할해야 함)
  + 어떤 경우에도 다 적용될 수 있는 포괄적인 명칭은 피해야 함
  + 명칭부여가 불가능한 처리가 없도록 분할함

**자료흐름도 작성 시 주의사항**

* 초기화와 종료화는 고려대상에서 제외
* 사소한 오류처리는 생략  
  사소한 오류 발견시에는 일단 표시해 놓고 전체그림을 작성해 나아간다  
  단, 오류 발생시 이전에 수행한 변경이나 자료저장소의 상태를 이전의 상태로 되돌릴 정도면 무시하지 말 것
* 제어흐름은 표시하지 않음(자료처리기에 대한 의식의 흐름은 제거)

**자료흐름도 검토 및 개선**

* 주어진 일을 불완전하게 실행한 후, 그것에 대해 개선하는 방식이 좋음
* 대체할 때마다 개선이 이루어지고 더 나은 자료흐름도가 완성되므로 여러 번 반복할 각오를 해야 한다

제 5장 자료 흐름도 작성 단계

5.1 자료흐름도의 단계화

5.2 자료흐름도의 분할 방법

5.3 논리적 모형의 구축

학습 목표

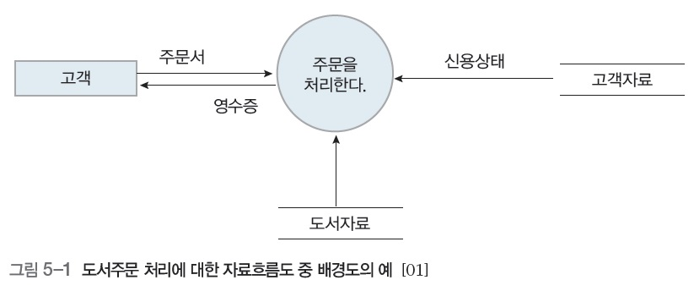
* 자료흐름도의 작성 단계를 학습한 후 배경도와 분할도를 직접 작성해 본다
* 자료흐름도의 분할 방법을 사례를 통해 학습한다
* 시스템의 물리적 모형과 논리적 모형에 대해 이해한다
* 논리적 모형의 구축을 위해 알아야 할 개념들을 학습한다
* 자료흐름도의 작성 사례들을 평가하고 개선할 수 있도록 다양한 사례들을 검토한다

**5.1 자료흐름도의 단계화**

**단계화 된 자료흐름도의 이점**

* 단계화 된 자료흐름도는 분석을 하향식으로 수행하므로 시스템을 상위로부터 하위로 조망하면서 전체적인 윤곽을 파악할 수 있음
* 분할된 페이지와 페이지를 연결하는 연결점이 필요 없음 (한 장의 자료흐름도가 특정 업무영역을 완전히 표현하게 되기 때문)
* 한 장의 종이는 일곱 개 전후로 적절한 개수의 처리를 포함하고 있어 이해하기 쉬움

**배경도**

* 분석하고자 하는 시스템과 외부 세계와의 접속관계를 식별하기 위한 것
* 시스템 분석의 범위를 결정함  
  

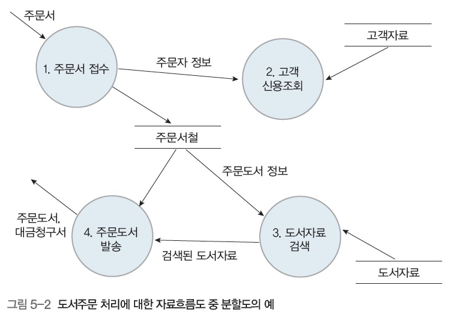
**분할도**

* 시스템의 복잡도에 따라 세분화된 자료흐름도를 통칭함

**분할을 어느 정도까지 하는 것이 좋은가에 대한 일반적인 지침**

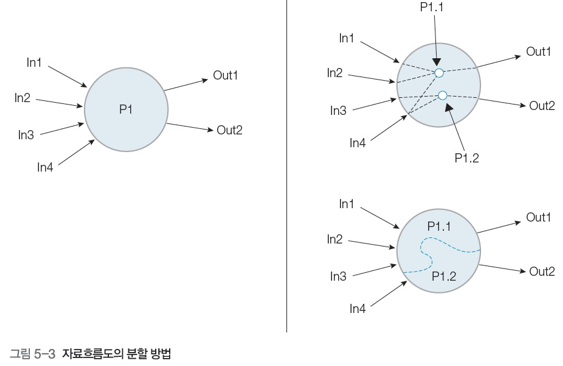
* 자료흐름에 주목하여 반영
* 개념적으로 의미 있는 접속관계가 이루어지도록 분할함
* 상위 단계의 분할은 하위 단계보다 많게 이루어져도 무방함
* 자료흐름도의 분할은 이해도를 저하시키지 않는 한 많이 하는 것이 좋음
* 일곱 개 전후로 분할된 자료흐름도가 이해 및 작업하기가 용이함
* 절대적인 분할원칙을 고수하는 것보다는 자료흐름도를 명확히 표현하여 이해가 쉽게 하는 것이 가장 좋음

**최하위 단계를 결정하는 데 도움이 되는 지침**

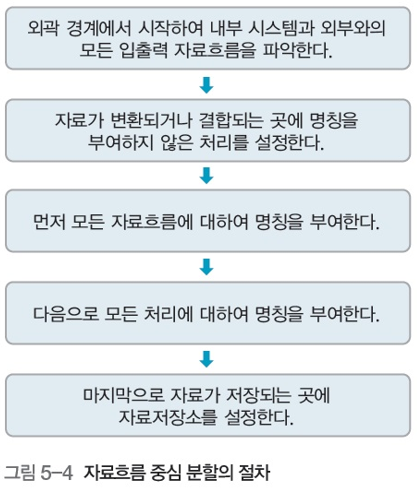
* 소단위 명세서로 한 페이지 이내에 기술할 수 있을 때까지 분할
* 처리에 대한 입력 자료흐름과 출력 자료흐름이 오직 하나씩 남을 때까지 분할
* 입력과 출력 사이에 일대일 또는 다대일의 관계를 갖는다면 분할이 충분히 이루어진 것임  
  

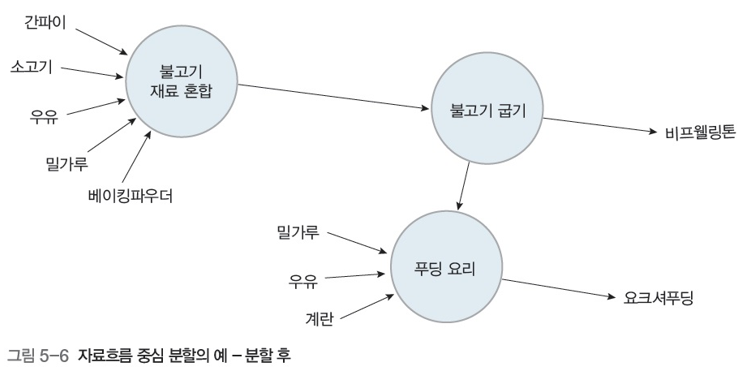
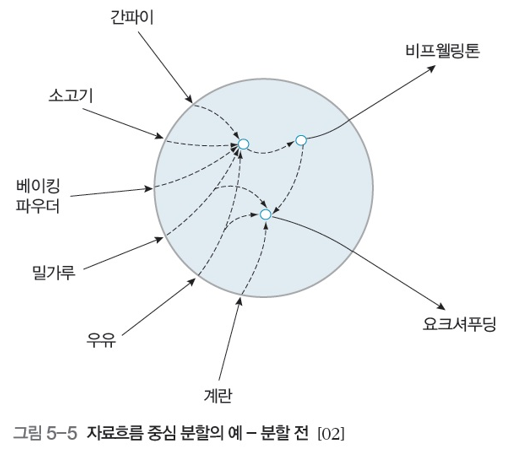
**5.2 자료흐름도의 분할 방법**

**자료흐름도를 세분화하는 방법**

1. 자료흐름 중심 분할, 접속점 분할: 자료흐름 중심으로 분할하는 방법
2. 처리 중심 분할, 조각그림 짜맞추기 분할: 처리 중심으로 분할하는 방법  
   

**자료흐름 중심 분할**

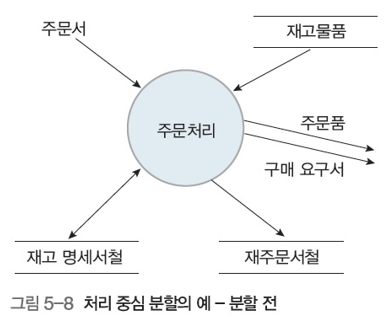
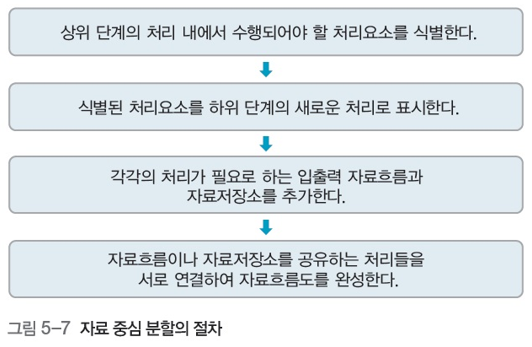
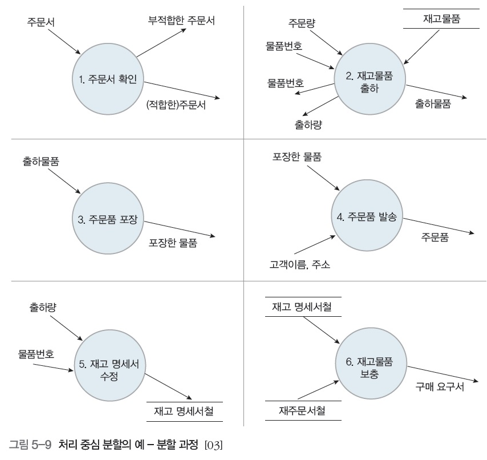
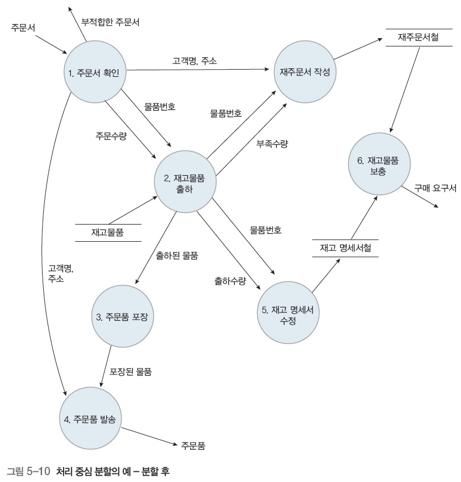




**자료흐름 중심 분할의 특징**

* 자료흐름 관점을 기반으로 분할도를 작성
* 처리순서에 따라 요구되는 자료들을 식별할 수 있음
* 입력 자료흐름과 출력 자료흐름 사이의 균형을 고려함
* 모호한 처리들을 구별할 수 있도록 도와줌
* 상대적으로 소규모 시스템에 적용하기에 적합함

**처리 중심 분할**

**처리 중심 분할의 특징**

* 처리기 관점에서 분할도를 작성함
* 활동의 처리 순서를 기준으로 순차적으로 분할함
* 처리 범위가 명확하게 정의된 보다 대규모의 시스템에 적용하기에 적합함
* 많고 상세한 처리기들 탓으로 너무 복잡해질 수 있음
* 자료흐름에 대한 적절한 이름이 사용되지 않은 경우 각각의 조각을 하나로 짜맞추기가 어려움

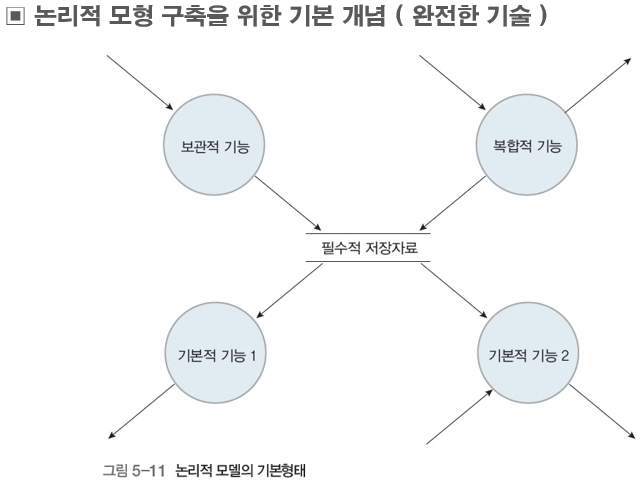
**5.3 논리적 모형의 구축**

**논리적 모형 구축을 위한 기본 개념(완전한 기술)**

* 완전한 기술: 분석가가 요구사항을 논리적 관점에서 파악할 수 있도록 도움
* 기술:
  1. 인간이 소기의 목적을 달성하기 위해 사용하는 수단
  2. 처리기와 저장기라는 두 가지 요소로 구성
* 완전한 기술 = 완전한 처리기 + 완전한 저장기

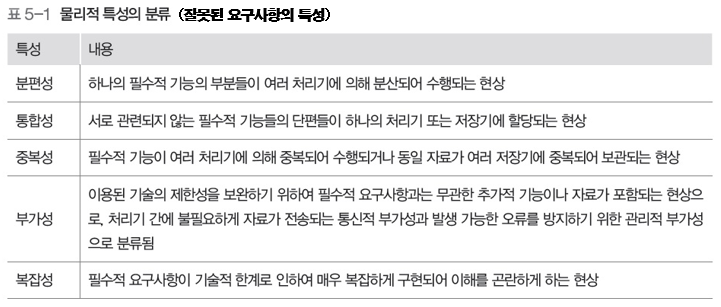
**완전한 기술의 관점에서 필수적 요구사항 정의**

* 필수적 요구사항: 완전한 기술을 이용하여 구현하더라도 시스템에 존재해야 하는 활동 및 자료
* 필수적 요구사항 = 필수적 기능 + 필수적 저장자료
* 필수적 기능: 기본적 기능과 보관적 기능으로 구분됨



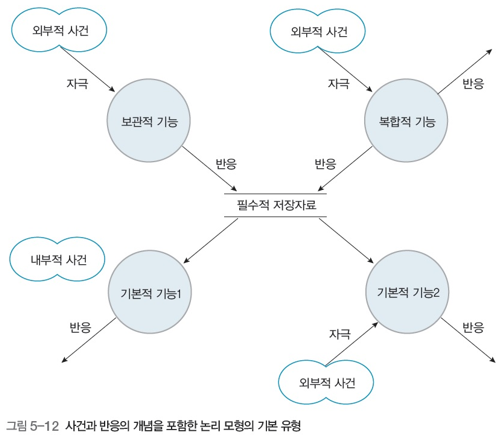
* 논리적 모형은 시스템의 현존 그대로를 표현한 물리적 모형을 구축한 후, 이 중 완전한 기술을 이용하여 잘못된 요구사항을 제거함으로써 구축할 수 있음

**물리적 특성의 분류**

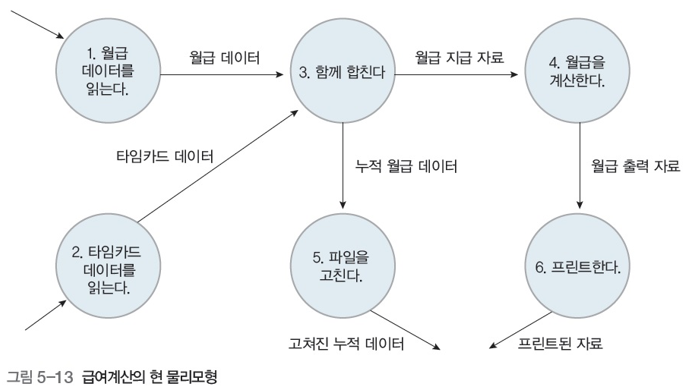
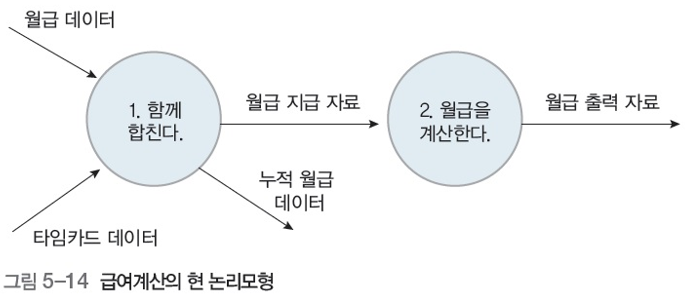


**논리적 모형 구축을 위한 기본 개념(사건과 반응)**

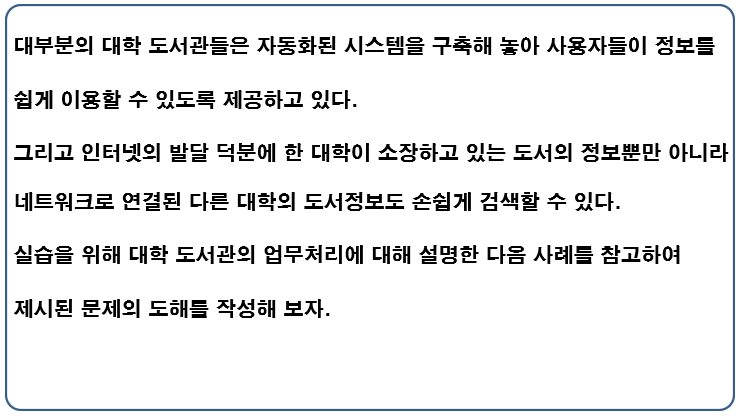
* 사건(Event): 시스템의 내부 및 외부에서 발생하는 상태변화, 내부적 사건과 외부적 사건이 있음
* 반응(Response): 특정 사건이 발생할 때마다 시스템에 의해 수행되어야 할 일련의 동작과 그 결과로써 필수적 기능을 형성

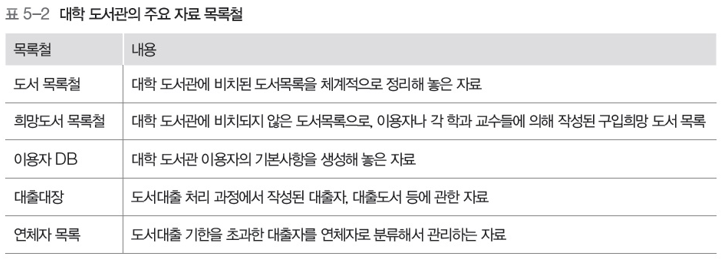


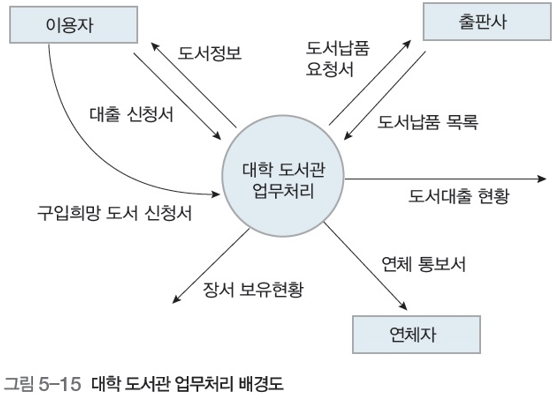
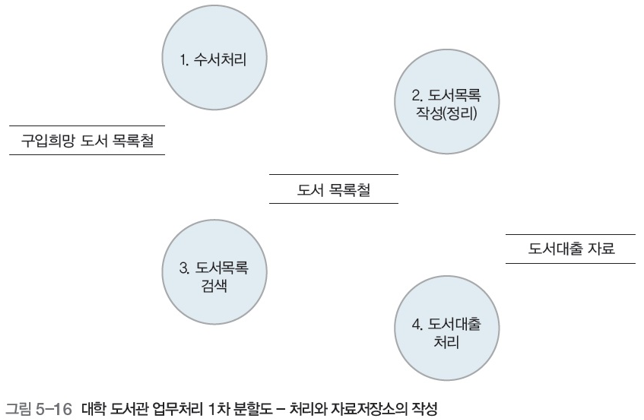
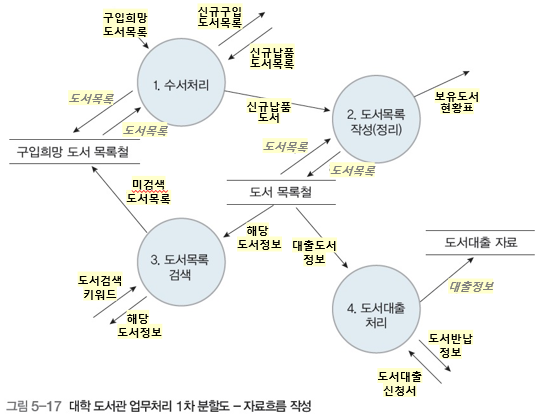
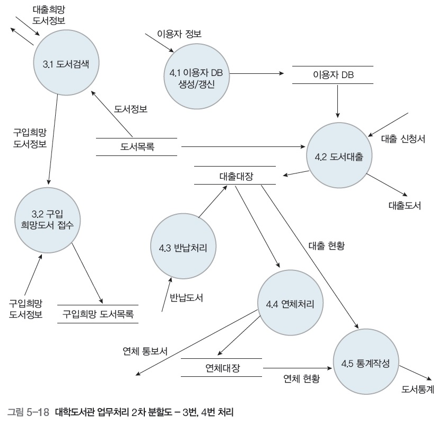
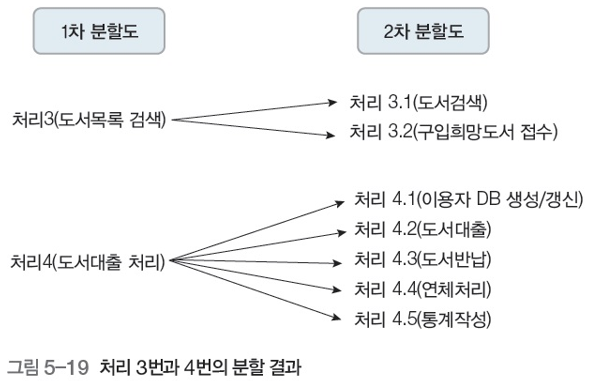
**물리적 모형 작성과 논리적 모형 작성**

* 현 논리 모형: 현업의 업무를 중심으로 최초로 작성되는 자료흐름도 (1/2)  
  
* 현 논리 모형: 순수하게 업무에 관계된 기능만 중심으로 재분석 작업을 통해 만들어진 모델 (2/2)  
  

**배경도 및 분할도 작성 실습**



1. 수서: 신규도서의 구매를 담당하는 부서
2. 정리: 신규도서에 대한 분류, 목록 작성, 라벨 등의 일을 담당하는 부서
3. 열람: 도서목록의 조회 서비스, 대출 서비스 등을 담당하는 부서  
   

* 대학 도서관 업무에 대한 배경도 작성  
  
* 대학 도서관 업무처리에 대한 1차 분할도 작성 (1/2)  
  
* 대학 도서관 업무처리에 대한 1차 분할도 작성 (2/2)  
  
* 대학 도서관 업무처리에 대한 2차 분할도 작성 (1/2)  
  
* 대학 도서관 업무처리에 대한 2차 분할도 작성 (2/2)  
  

제 6장 자료 사전

6.1 자료사전의 특성

6.2 자료사전 표기법

6.3 자료사전 작성 원칙

학습 목표

* 구조적 분석 기법의 주요 도구 중 하나인 자료사전의 특성을 이해한다
* 자료사전 작성 시 사용하는 기호를 식별할 수 있도록 학습한다
* 자료사전의 작성 원칙을 학습한다
* 자료사전의 작성 사례를 평가하고 개선할 수 있도록 다양한 사례를 검토한다

**6.1 자료사전의 특성**

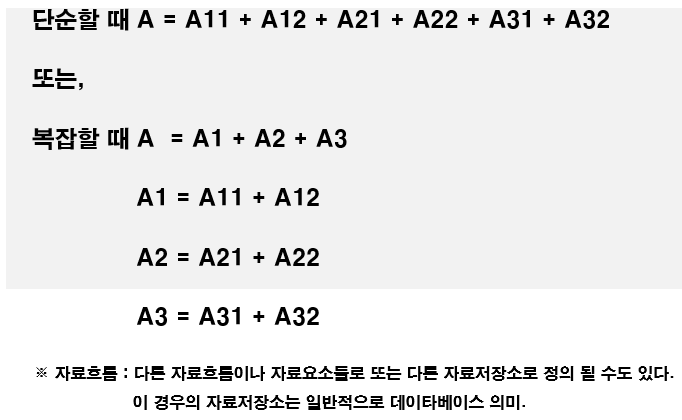
**자료사전의 역할**

* 자료사전
  + 자료흐름도에 기술된 모든 자료에 대해 다음 사항들을 정의
  + 자료흐름을 구성하는 자료항목
  + 자료에 대한 의미
  + 자료저장소를 구성하는 자료항목
  + 자료 원소의 단위 및 값

**자료사전과 자료흐름도 관계**



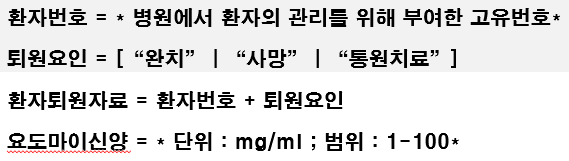
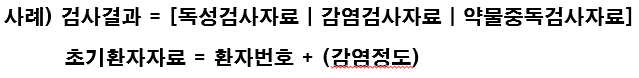
**자료의 하향식 분할**



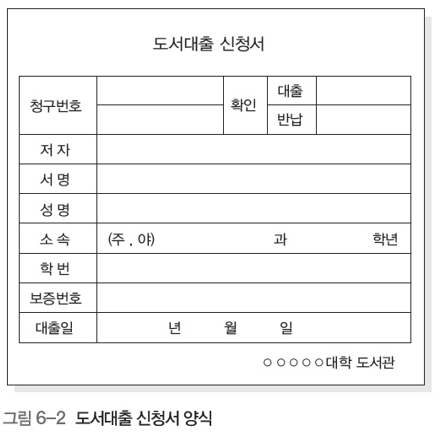
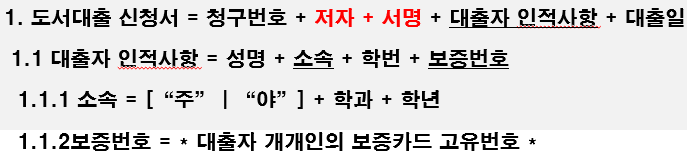
**6.2 자료사전 표기법**

**자료사전 작성법**



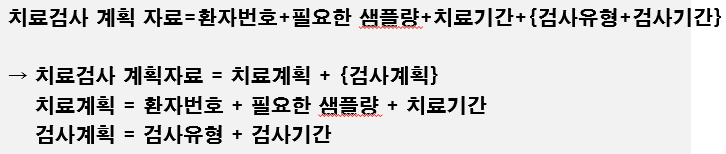
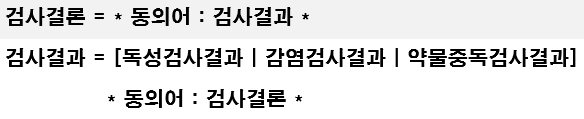
* 정의
  + 주석을 사용하여 의미를 기술
  + 자료흐름과 자료저장소에 대한 구성내역을 설명
  + 자료원소에 대하여 값이나 단위를 나타냄  
    
* 반복
  + 여러 번 반복되는 자료항목은 { } 안에 기술
  + { }의 좌측에는 최소 반복횟수를 기록하고, 우측에는 최대 반복횟수를 기록
  + 반복횟수를 기록하지 않을 때는 디폴트로 최소는 0, 최대는 무한대를 나타냄  
    
* 선택과 생략가능
  + 선택기호 [ | ]: ‘ | ‘ 로 분리된 항목들 중 하나가 선택되었다는 것을 표시
  + 생략 가능 기호 ( ): 괄호 안의 자료항목이 기술될 수도, 생략될 수도 있음을 표시  
    
* 자료원소
  + 더 이상 분할되지 않는 자료항목으로 특정한 값이나 값의 범위를 취함  
    

**자료사전 작성 사례**

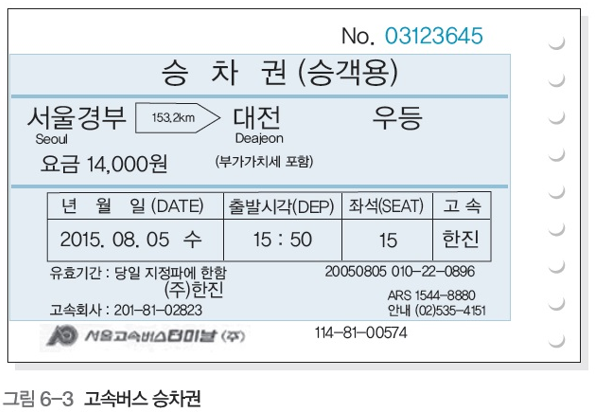
* 도서대출 신청서 양식을 자료사전으로 작성하기  
  
* 도서대출 신청서의 자료사전을 작성한 사례 1)  
  
* 도서대출 신청서의 자료사전을 작성한 사례 2) – 저자와 서명 생략  
  

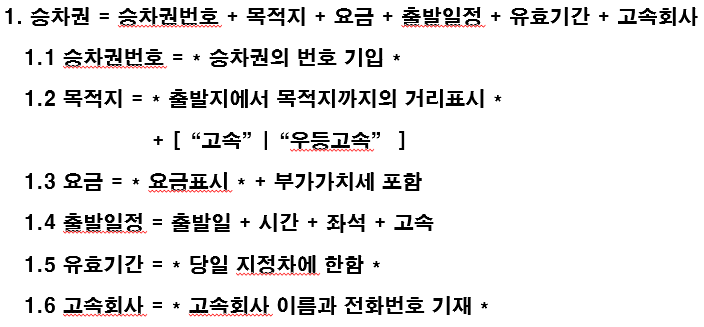
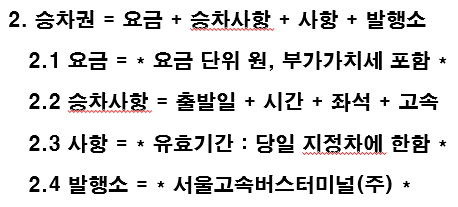
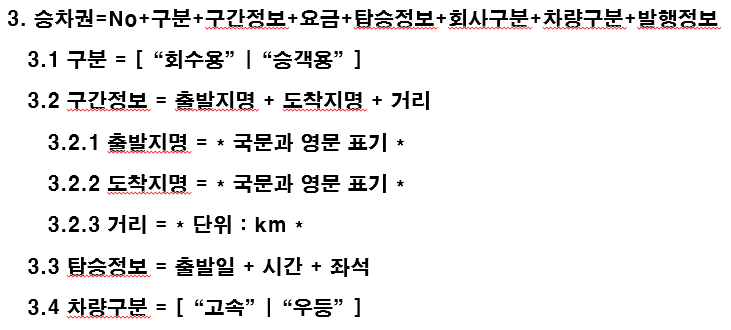
**6.3 자료사전 작성 원칙**

**자료사전 작성시 고려사항**

* 자료의 의미 기술
  + 자료의 의미는 주석을 통해서 기술
* 자료 구성항목의 기술
  + 구성항목을 그룹으로 묶음
  + 각 그룹에 대해 의미 있는 이름을 부여
  + 이름이 붙여진 각 그룹을 다시 정의  
    
* 동의어(Alias)
  + 자료사전에 이미 정의된 자료항목에 대한 또 다른 이름
  + 동의어가 많아지면 자료의 명칭에 혼동이 생길 우려가 있음  
    
* 자료정의의 중복 제거
  + 자료정의는 중복성을 제거하고 간단명료하게 하는 것이 좋음

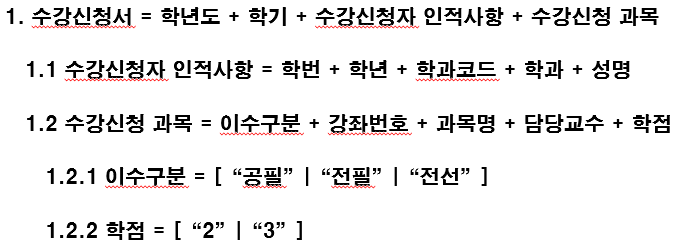
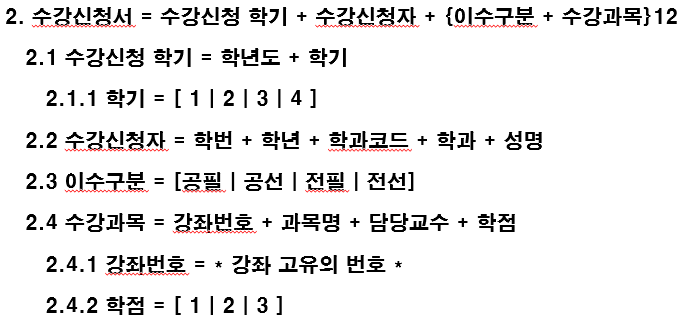
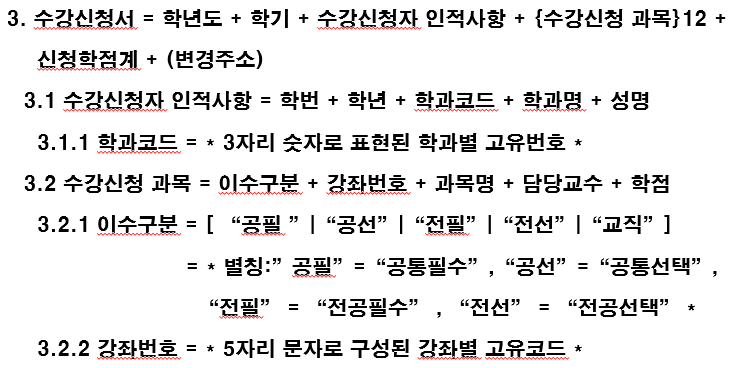
**자료사전 작성 실습 – 1**



* 작성 예제 1 – 1  
  
* 작성 예제 1 – 2  
  
* 모범답안  
  

**자료사전 작성 실습 – 2**



* 작성 예제 2 – 1  
  
* 작성 예제 2 – 2  
  
* 개선된 자료사전  
  

제 7장 소단위 명세서

7.1 소단위 명세서의 특징

7.2 구조적 언어

7.3 선후 결정문

7.4 의사결정표

학습 목표

* 구조적 분석 기법의 주요 도구 중 하나인 소단위 명세서의 특성을 이해한다
* 소단위 명세서를 작성하기 위한 구조적 언어에 대해 학습한다
* 소단위 명세서를 작성하기 위한 선후 조건문에 대해 학습한다
* 소단위 명세서의 작성법 가운데 하나인 의사결정표를 학습한다
* 소단위 명세서의 작성 사례들을 평가하고 개선할 수 있도록 실습 예제를 풀어본다

**7.1 소단위 명세서의 특성**

**소단위 명세서 정의**

* 입력 자료흐름을 출력 자료흐름으로 변환하기 위해 중간에 수행하는 각 처리들의 업무절차를 상세히 작성해 놓은 것
* 구조적 언어, 선후 조건문, 의사 결정표와 같은 도구들이 주로 사용됨

**소단위 명세서 작성 도구의 조건**

* 사용자와 시스템 분석가가 검증 가능한 형태로 표현
* 여러 계층의 사람들이 의사소통 할 수 있는 형태로 표현
* 설계와 구현사항을 임의로 결정하지 않도록 유의

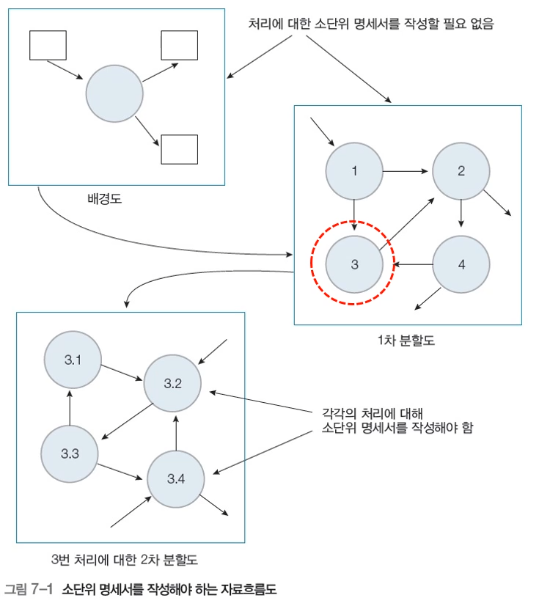
**소단위 명세서의 역할**

* 소단위 명세서에서는 여러 계층의 사람들이 효과적으로 의사소통을 할 수 있는 형태로 표현되어야 함
* 소단위 명세서를 작성하는 것은 시스템 분석가이지만, 사용자, 관리자, 감사자, 품질보증 담당자 등 다양한 사람들이 읽고 이해하고 검증할 수 있어야 함
* 설계와 구현사항에 대해 임의 결정은 금지

**소단위 명세서의 특성**

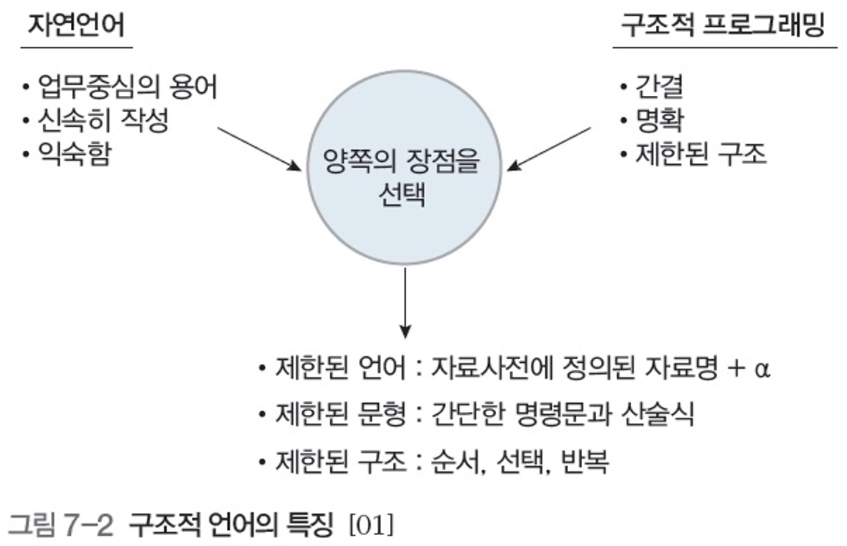
* 오직 최하위 단계의 처리만 기술
* 항상 간결, 명료, 완전하여 의미에 모호함이 없어야 함

**소단위 명세서의 작성 대상**

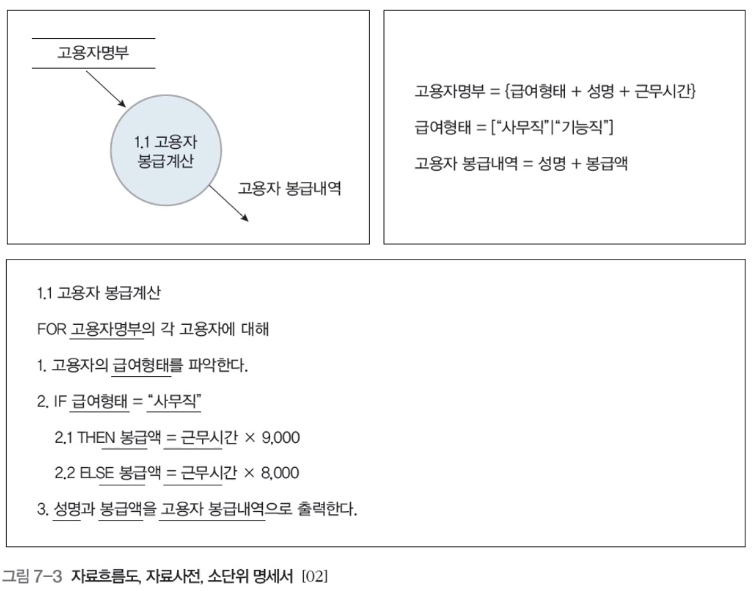


**7.2 구조적 언어**

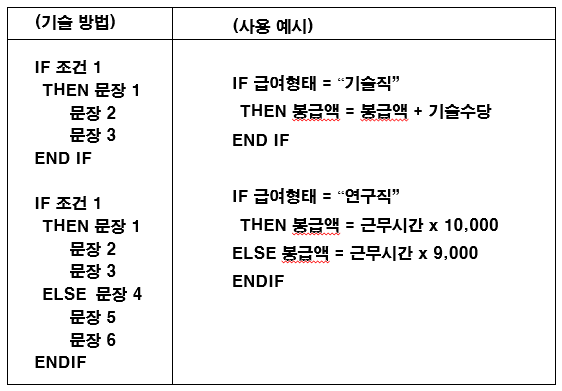
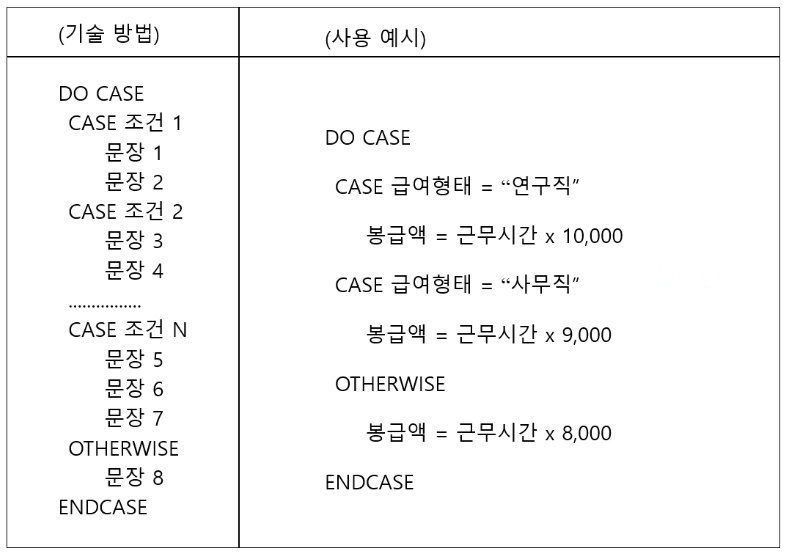
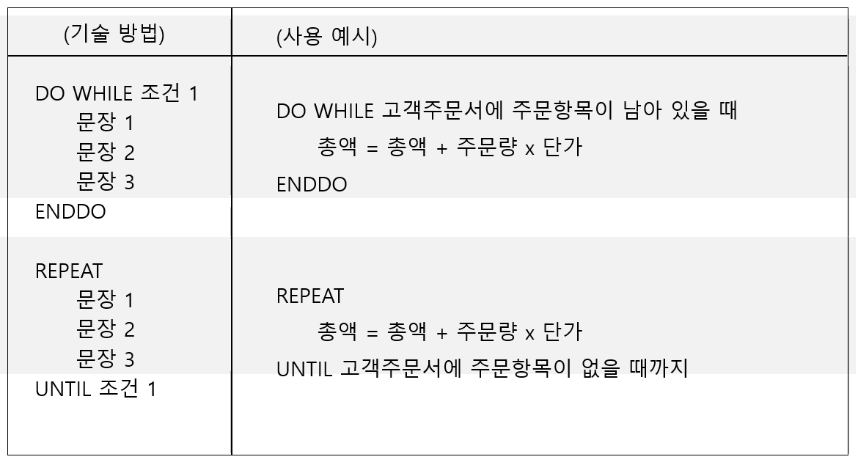
**구조적 언어의 개념**



**구조적 분석에 따른 자료흐름도, 자료사전, 소단위 명세서**



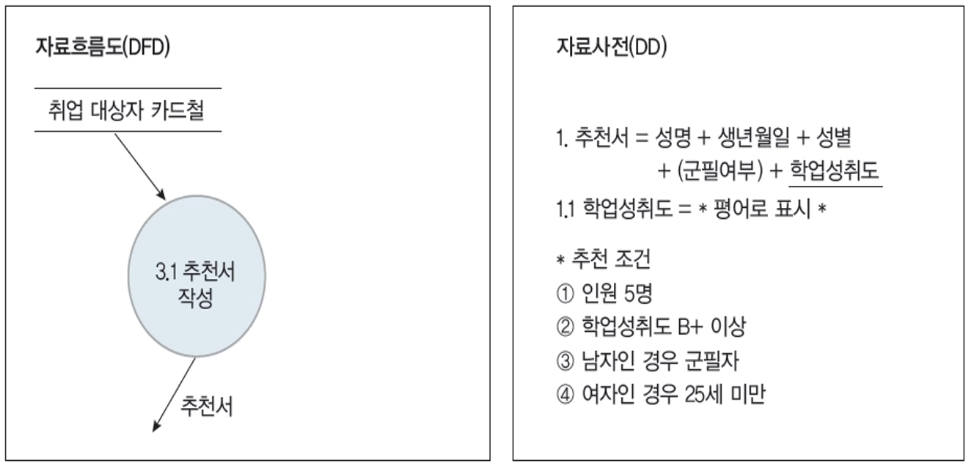
**구조적 언어의 구조**

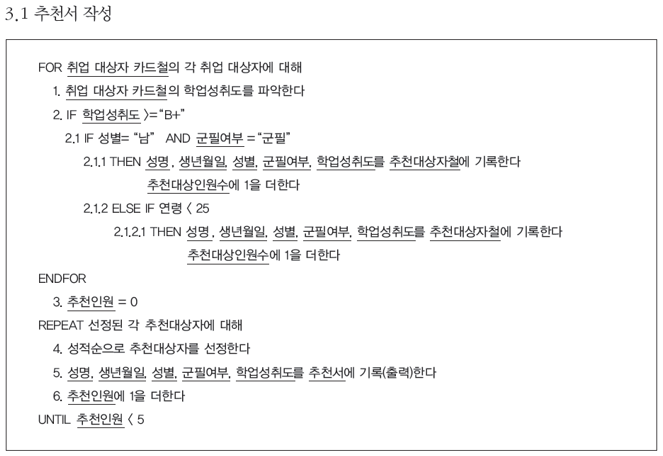
* 순서문: 문장들을 단순히 순서대로 작성한 것  
  
* 선택문: IF 문 또는 CASE 문을 사용함  
    
  
* 반복문: DO WHILE 문, REPEAT UNTIL 문, FOR 문이 있음  
  

**구조적 언어의 작성 지침**

* 구조적 언어가 한 페이지를 초과하지 않도록
  + 한 페이지를 초과하게 되면 다른 방식으로 알고리즘을 간략히 기술하는 것을 생략
  + 그러나, 마땅한 대안이 없다면 처리 자체를 하위단계로 분할
* IF 문 또는 CASE 문의 제어구조를 사용할 때는 중첩도(Nesting)가 3단계를 초과하지 않도록 한다
* 제어구조를 중첩해 사용할 때는 중첩에 따라 요철모양(Indentation)을 사용
* 사용자가 이해하기 쉬운 단어 사용 및 수정 용이
* 작성형식의 혼합 가능(구조적언어, 의사결정표 등과 함께)
* 순차, 반복, 선택 구조 등이 명확할 것
* 비공식적으로 융통성 있게 작성(즉, 공식적으로 표준화 시키지 않도록)

**소단위 명세서 작성 실습(취업 추천서)**





**7.3 선후 조건문**

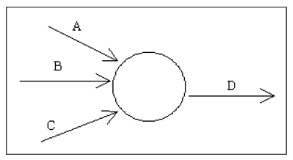
**선후 조건문을 작성하는 이유**

* 어떤 처리에 대한 알고리즘이나 절차를 기술하지 않고, 그 처리가 수행해야 할 기능을 기술하는 방법

**선후 조건문이 유용한 경우**

* 사용자가 자신이 오랫동안 사용하던 특유의 알고리즘으로 처리할 때
* 적용 가능한 알고리즘들이 여럿 존재할 때
* 다수의 알고리즘을 프로그래머가 선택하길 원할 때

**선후 조건문의 작성 방법**

* 선조건문: 처리가 수행되기 전에 만족해야 할 모든 사항들을 기술
* 선조건문의 기술사항
  + 어떤 입력이 존재해야 하는지 기술
  + 입력 자료 내의 관계 또는 입력 자료 간의 관계를 기술
  + 입력 자료와 자료저장소 간의 관계를 기술
  + 자료저장소들 간의 관계 또는 자료저장소 내에서의 관계를 기술  
    
* 후조건문: 처리 작업을 마쳤을 때 만족해야 할 사항들을 기술
* 후조건문의 기술사항
  + 처리가 산출하는 출력을 기술
  + 출력 값과 입력 값 사이에 존재해야 하는 관계를 기술
  + 출력 값과 자료저장소 값들 간에 존재해야 할 관계를 기술
  + 자료저장소 변경사항을 기술  
    ・새로운 항목의 추가  
    ・기존 항목의 수정, 삭제 등

**선후 조건문 작성 시 고려사항**

* 선후 조건문을 작성할 때는 정상적인 처리 상황을 우선적으로 기술
* 여러 가지의 정상적인 상황이 존재할 때는 이들을 제각기 다른 선조건문으로 기술
* 각각의 선조건문에 대해 출력이 산출되었거나 자료저장소가 수정되었을 때의 처리 상황을 후조건문으로 기술해야 함
* 정상적인 처리상황이 기술되었으면 다음에는 오류의 경우나 비정상적인 경우에 대해 적절한 선후조건문을 작성해야 함
* 입력과 출력자료 사이에 복잡한 변환관계가 존재하면 구조적 언어로 표현방법 변경

**7.4 의사결정표**

**의사결정표를 사용하는 경우**

* 처리가 산출하는 출력이 복잡한 의사결정에 의해 좌우될 때
* 의사결정이 수많은 입력 자료에 의해 좌우될 때
* 입력 자료가 광범위한 값을 가질 때  
  

**의사결정표 작성의 장점**

* 사용자가 전혀 생각하지 못했던 변수들의 조합 발견 가능
* 각각의 규칙에 대해 집중 분석할 수 있음
* 특정 처리에 대한 구체적인 알고리즘을 명시하지 않음

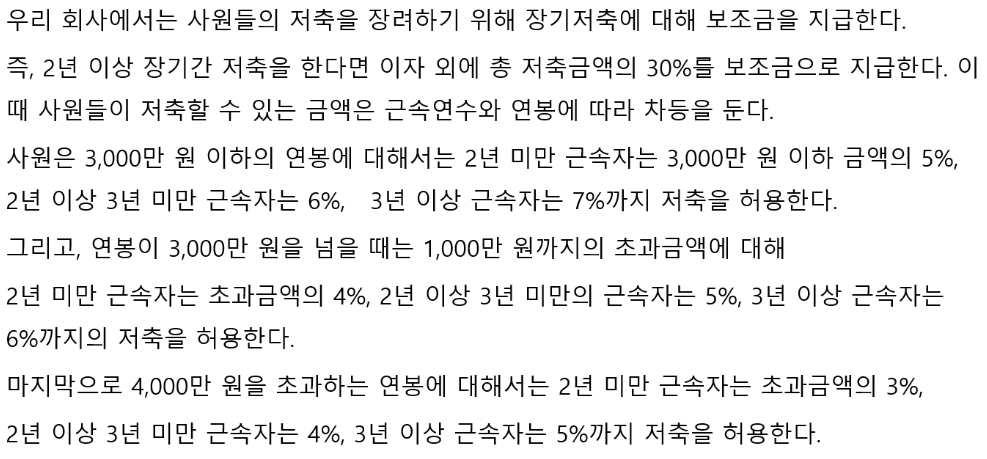
**의사결정표의 효과**

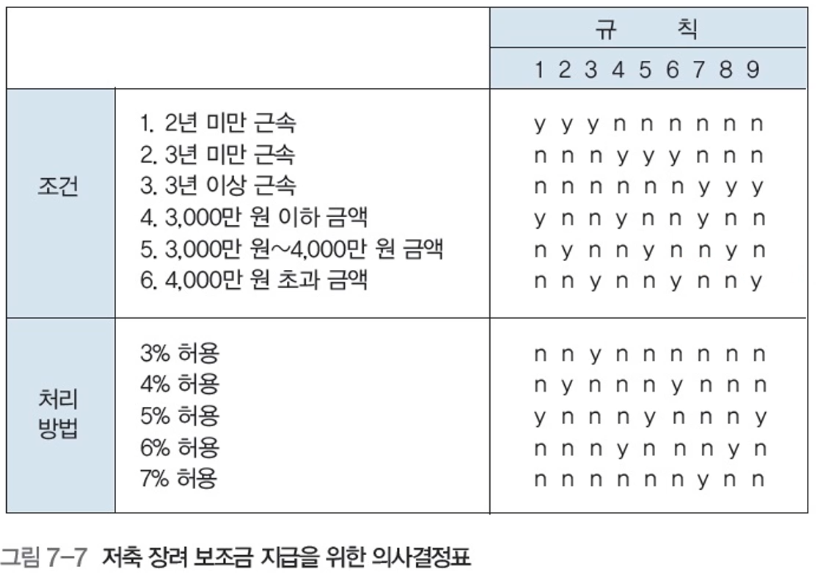
* 사용자가 분석결과에 대해 검토할 때 이해하기 쉽다
* 분석가는 규칙의 경우의 수를 정확히 알 수 있다
* 누락이나 중복을 피할 수 있다
* 사용자와 함께 오류나 불확실한 것을 찾아낼 수 있다

**의사결정표 작성 절차**

1. 명세할 모든 조건과 변수를 식별하고, 각각의 변수가 취할 수 있는 모든 값을 확인
2. 조건들의 조합의 수를 계산
3. 의사결정표에 요구되는 가능한 모든 처리방법을 식별
4. 표의 왼쪽에 조건들과 처리방법들을 표시하고, 오른쪽에는 조건들의 조합인 규칙에 번호를 부여함
5. 표의 세로로 이루어진 칸에 조건들의 조합을 하나씩 채워 넣음
6. 각각의 규칙에 대해 적절한 처리방법을 식별
7. 의사결정표에서 생략되는 부분, 모순되는 부분, 모호한 부분 등의 유무 확인
8. 사용자와 생략되는 부분, 모순되는 부분, 모호한 부분 등을 토론해 의사결정표에 반영시킴

**의사결정표 작성 사례 연구 1)**





**의사결정표 작성 사례 연구 2)**

