학습 목표

구조적 분석 기법의 주요 도구 중 하나인 소단위 명세서의 특성을 이해한다

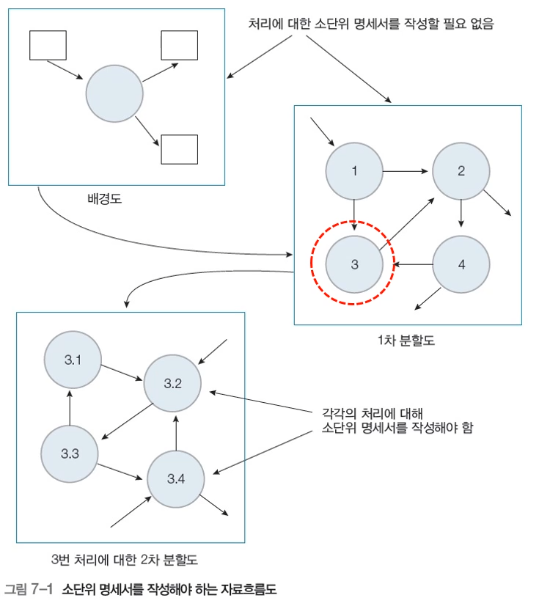
소단위 명세서를 작성하기 위한 구조적 언어에 대해 학습한다

소단위 명세서를 작성하기 위한 선후 조건문에 대해 학습한다

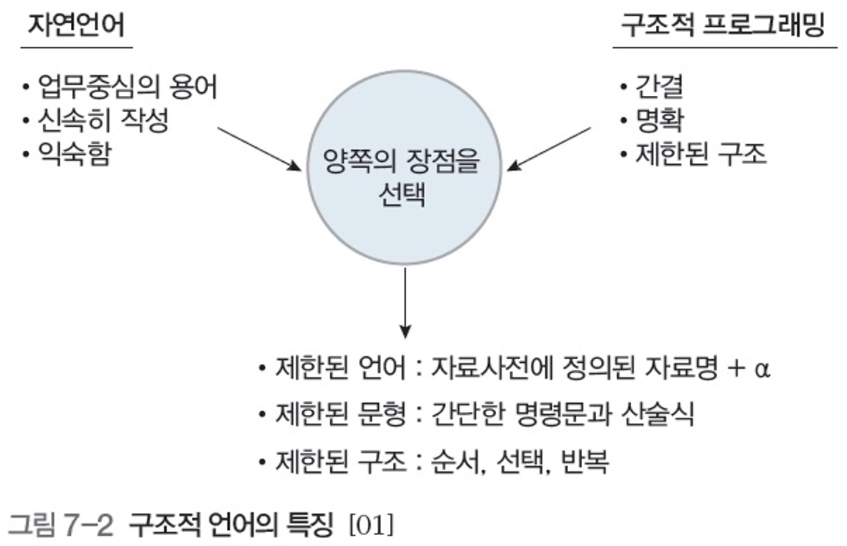
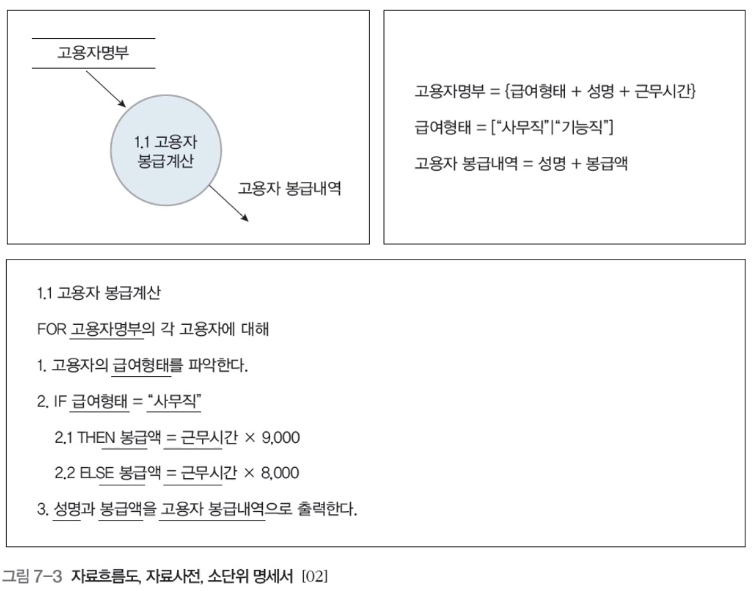
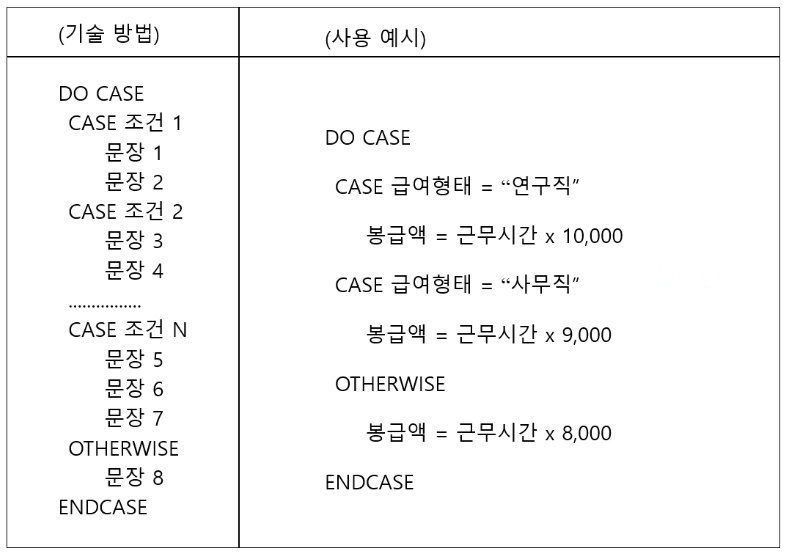
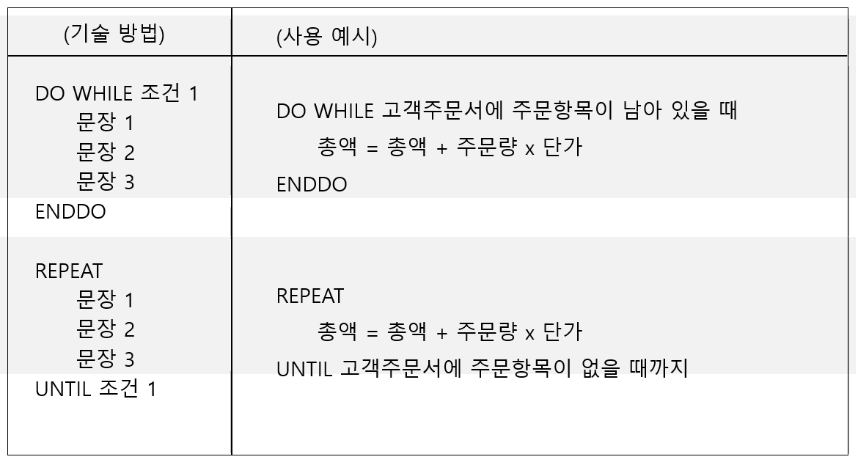
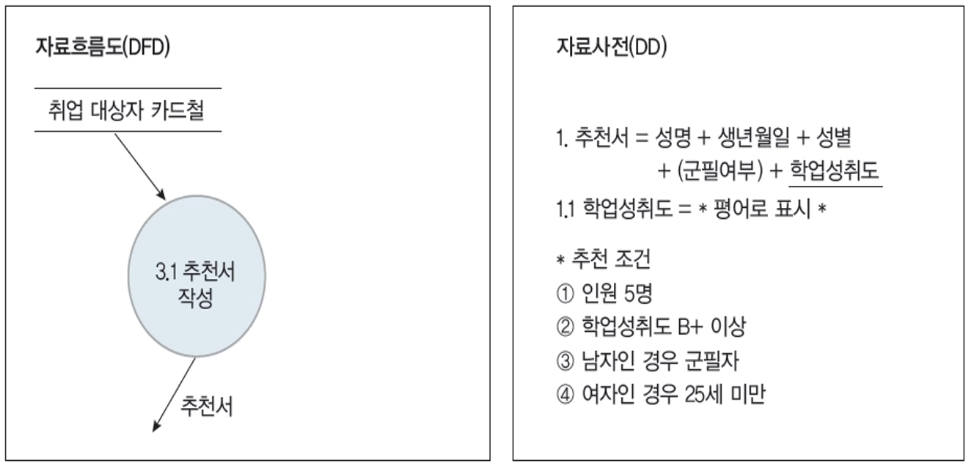
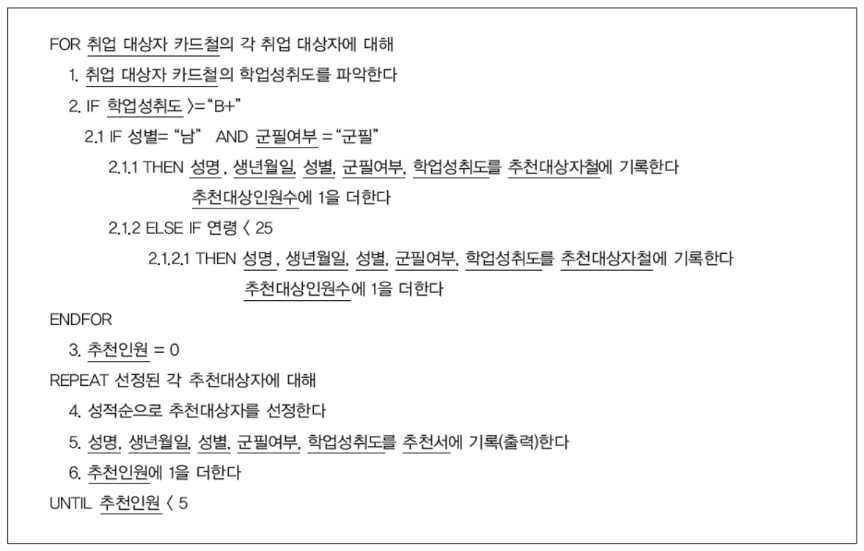
소단위 명세서의 작성법 가운데 하나인 의사결정표를 학습한다

소단위 명세서의 작성 사례들을 평가하고 개선할 수 있도록 실습 예제를 풀어본다

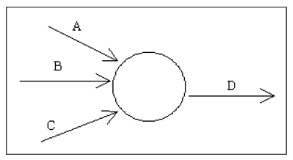
**7.1 소단위 명세서의 특성**

1. 소단위 명세서 정의
   1. 입력 자료흐름을 출력 자료흐름으로 변환하기 위해 중간에 수행하는 각 처리들의 업무 절차를 상세히 작성해 놓은 것
   2. 구조적 언어, 선후 조건문, 의사 결정표와 같은 도구들이 주로 사용됨  
      다음 중 소단위 명세서를 하기 위한 도구가 아닌 것은 -> 시험 문제
2. 소단위 명세서 작성 도구의 조건
   1. 사용자와 시스템 분석가가 검증 가능한 형태로 표현
   2. 여러 계층의 사람들이 의사소통 할 수 있는 형태로 표현
   3. 설계와 구현사항을 임의로 결정하지 않도록 유의
3. 소단위 명세서의 역할
   1. 소단위 명세서에서는 여러 계층의 사람들이 효과적으로 의사소통 할 수 있는 형태로 표현되어야 함
   2. 소단위 명세서를 작성하는 것은 시스템 분석가이지만, 사용자, 관리자, 감사자, 품질보증 담당자 등 다양한 사람들이 읽고 이해하고 검증할 수 있어야 함
   3. 설계와 구현사항에 대해 임의 결정은 금자
4. 소단위 명세서의 특성
   1. 오직 최하위 단계의 처리만 기술
   2. 항상 간결, 명료, 완전하여 의미에 모호함이 없어야 함
5. 소단위 명세서의 작성 대상  
   

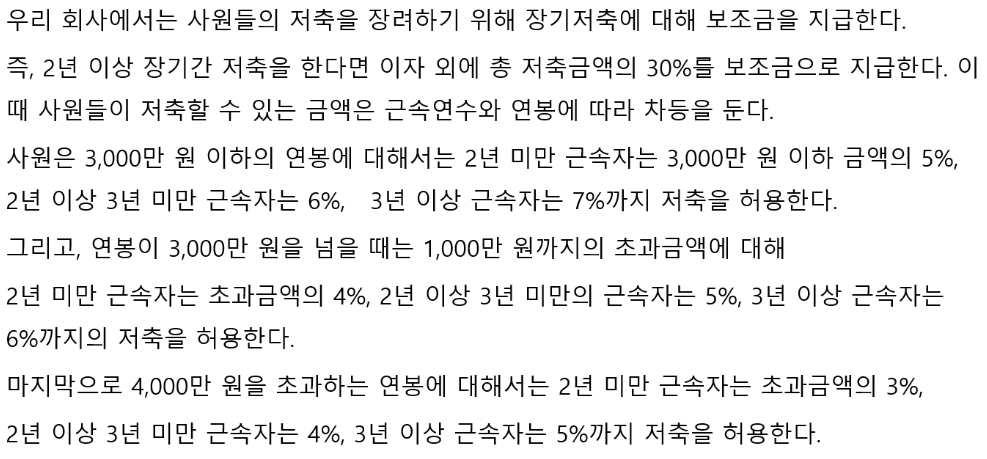
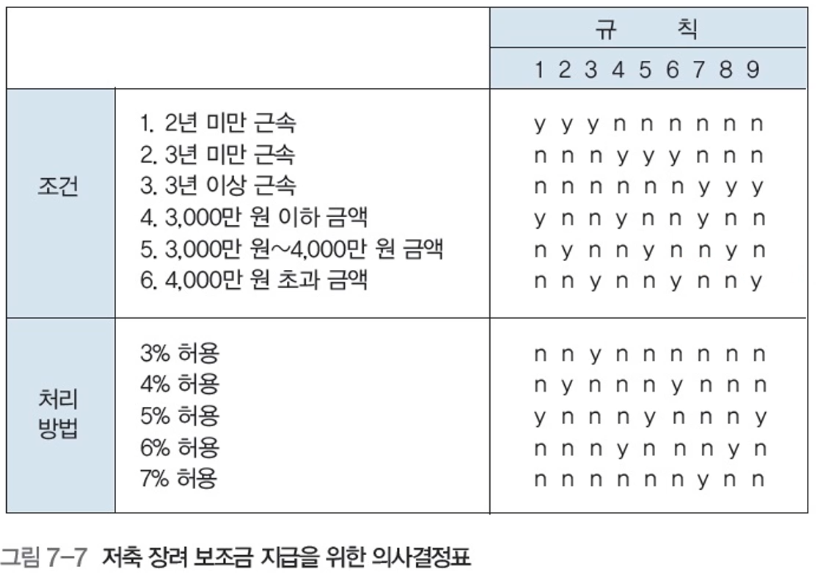
**7.2 구조적 언어**

1. 구조적 언어의 개념  
   
2. 구조적 분석에 따른 자료흐름도, 자료사전, 소단위 명세서  
   
3. 구조적 언어의 구조
   1. 순서문: 문장들을 단순히 순서대로 작성한 것  
      
   2. 선택문: IF 문 또는 CASE 문을 사용함  
      
   3. 반복문: DO WHILE 문, REPEAT UNTIL 문, FOR 문이 있음  
      
4. 구조적 언어의 작성 지침
   1. 구조적 언어가 한 페이지를 초과하지 않도록
      1. 한 페이지를 초과하게 되면 다른 방식으로 알고리즘을 간략히 기술하는 것을 생각
      2. 그러나, 마땅한 대안이 없다면 처리 자체를 하위 단계로 분할  
         (이건 기억해두기)
   2. IF 문 또는 CASE 문의 제어구조를 사용할 때는 중첩도(Nesting)가 3단계를 초과하지 않도록 한다
   3. 제어구조를 중첩해 사용할 때는 중첩에 따라 요철모양(Indentation)을 사용
   4. 사용자가 이해하기 쉬운 단어 사용 및 수정 용이
   5. 작성형식의 혼합 가능(구조적 언어, 의사결정표 등과 함께)
   6. 순차, 반복, 선택 구조 등이 명확할 것
   7. 비공식적으로 융통성 있게 작성(즉, 공식적으로 표준화 시키지 않도록)
5. 소단위 명세서 작성 실습(취업 추천서)  
     
   

**7.3 선후 조건문**

1. 선후 조건문을 작성하는 이유
   1. 어떤 처리에 대한 알고리즘이나 절차를 기술하지 않고, 그 처리가 수행해야 할 기능을 기술하는 방법
2. 선후 조건문이 유용한 경우
   1. 사용자가 자신이 오랫동안 사용하던 특유의 알고리즘으로 처리할 때
   2. 적용 가능한 알고리즘들이 여럿 존재할 때
   3. 다수의 알고리즘을 프로그래머가 선택하길 원할 때
3. 선후 조건문의 작성 방법
   1. 선조건문: 처리가 수행되기 전 만족해야 할 모든 사항들을 기술
   2. 선조건문의 기술사항
      1. 어떤 입력이 존재해야 하는지 기술
      2. 입력 자료 내의 관계 또는 입력 자료 간의 관계를 기술
      3. 입력 자료와 자료저장소 간의 관계를 기술
      4. 자료저장소들 간의 관계 또는 자료저장소 내에서의 관계를 기술  
         
   3. 후조건문: 처리 작업을 마쳤을 때 만족해야 할 사항들을 기술
   4. 후조건문 기술사항
      1. 처리가 산출하는 출력을 기술
      2. 출력 값과 입력 값 사이에 존재해야 하는 관계를 기술
      3. 출력 값과 자료저장소 값들 간에 존재해야 할 관계를 기술
   5. 자료저장소의 변경사항들을 기술  
      ・새로운 항목의 추가  
      ・기존 항목의 수정, 삭제 등
4. 선후 조건문 작성 시 고려사항
   1. 선후 조건문을 작성할 때는 정상적인 처리 상황을 우선적으로 기술
   2. 여러 가지의 정상적인 상황이 존재할 때는 이들을 제각기 다른 선조건문으로 기술
   3. 각각의 선조건문에 대해 출력이 산출되었거나 자료저장소가 수정되었을 때의 처리 상황을 후조건문으로 기술해야 함
   4. 정상적인 처리상황이 기술되었으면 다음에는 오류의 경우나 비정상적인 경우에 대해 적절한 선후조건문을 작성해야 함
   5. 입력과 출력자료 사이에 복잡한 변환관계가 존재하면 구조적 언어로 표현방법 변경

**7.4 의사결정표**

1. 의사결정표를 사용하는 경우
   1. 처리가 산출하는 출력이 복잡한 의사결정에 의해 좌우될 때
   2. 의사결정이 수많은 입력 자료에 의해 좌우될 때
   3. 입력 자료가 광범위한 값을 가질 때  
      
2. 의사결정표 작성의 장점
   1. 사용자가 전혀 생각하지 못했던 변수들의 조합 발견 가능
   2. 각각의 규칙에 대해 집중 분석을 할 수 있음
   3. 특정 처리에 대한 구체적인 알고리즘을 명시하지 않음
3. 의사결정표의 효과
   1. 사용자가 분석결과에 대해 검토할 때 이해하기 쉽다
   2. 분석가는 규칙의 경우의 수를 정확히 알 수 있다
   3. 누락이나 중복을 피할 수 있다
   4. 사용자와 함께 오류나 불확실한 것을 찾아낼 수 있다
4. 의사결정표의 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
5. 의사결정표 작성 사례 연구 1)  
     
   
6. 의사결정표 작성 사례 연구 2)  
   