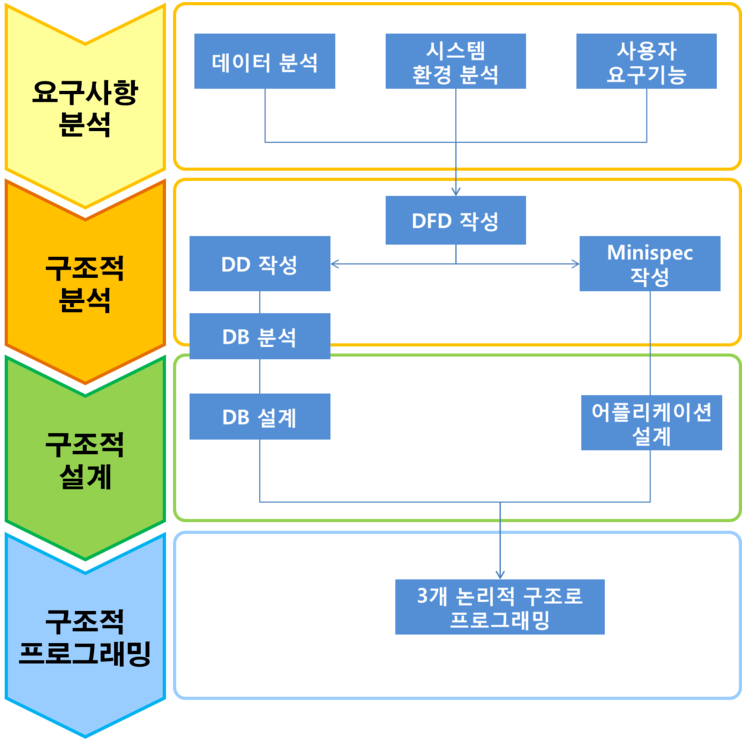
5. 비즈니스 환경에서의 정보시스템들을 개발하기 위해 개념적 방법론, 구조적 분석설계의 방법론, 객체지향시스템분석설계의 방법론에 대하여 작성하세요.

개념적 방법론

소프트웨어 개발에 있어 고객이 요구하는 내용을 조사, 분석하여 컴퓨터 프로그램에 반영하기 위해 먼저 소프트웨어 개발절차를 이해하고, 분석과 설계 단계의 중요성을 이해하여야 한다. 분석 단계는 사용자의 문제를 구체적으로 이해하고, 소프트웨어가 담당해야 하는 정보의 영역을 정의한다. 또 사용자가 원하는 기능, 프로그램의 목표 성능, 프로그램의 신뢰도와 같은 요구들은 요구 명세서(requirements specifications)로 문서화된다. 이러한 과정을 원활하게 진행하기 위해서는 사용자와 개발자의 의사소통 기술이 필수적이다. 설계 단계는 소프트웨어의 구조와 그 컴포넌트들을 명확하게 밝혀 구현을 어떻게 할지 정하는 단계이다. 설계 단계는 크게 외부 설계와 내부 설계 두단계로 나눌 수 있다. 외부 설계는 시스템과 사용자와의 인터페이스를 중시하는 설계이고, 내부 설계는 시스템의 내부 동작을 중시하는 설계이다. 설계 단계에서의 결과는 설계 명세로 산출되어야 하고, 이 명세를 토대로 사용자 메뉴얼과 소프트웨어 테스트 계획이 작성되어야 한다. 앞에서 설명한 분석과 설계방법은 많은 연구와 경험을 통하여 여러가지 방법론으로 정리되어 있다. 여기서는 구조적 분석설계의 방법론과, 객체지향시스템 분석설계를 상세히 설명한다.

구조적 분석설계의 방법론

구조적 분석설계란 소프트웨어의 품질이 일정하지 못하고 하드웨어의 생산과 발전 속도에 비하여 소프트웨어의 생산 분야에서는 오래전부터 침체되어 왔다. 프로그램의 규모가 커질수록 불량율이 늘고 유지 보수에도 많은 어려움이 발생하였기 때문이다. 이를 해결하기 위하여 순차, 선택, 반복의 구조로만 프로그램을 만들면 로직을 이해하기가 쉬어 좋은 프로그램이 된다는 것이 바로 구조적 프로그래밍의 원리이다. 하지만 구조적인 프로그래밍 보다도 분석, 설계 등의 프로그래밍 이전 과정에서도 구조적인 접근 방법이 동원된다면 소프트웨어 개발의 생산성과 품질을 높일 수 있다는 이론이 나왔다. 이것이 바로 구조적 분석과 설계이다.



[그림 1. 구조적 분석과 설계 절차]

구조적 분석과 설계는 구조적인 프로그래밍을 기반으로 하였기 때문에, 당연히 구조적인 프로그래밍을 유도하도록 분석과 설계가 진행된다. 구조적 분석과 설계의 절차는 [그림 1]과 같이 이루어진다. 또한 철저하게 모듈화 하여 프로그래밍이 이루어 진다는 점이 특징이다. [그림 1]에서 DFD는 데이터 흐름도(Data Flow Diagram)이고, DD는 자료사전(Data Dictionary)이다. 또 Mini-spec이란 소단위 명세라고도 하며 기능을 쪼갤 수 없을 정까지의 기능으로 분리한 뒤, 해당 기능에 대한 명세를 하는 것을 말한다.

구조적 분석과 설계의 5가지 원리는 다음과 같다. **첫 번째로 추상화(Abstraction)**이라는 원리가 있다. 추상화는 문제나 데이터를 표현하기 위해, 세부사항을 모두 기술하기 보다는 추상화시켜 복잡한 것을 단순하게 보이도록 하는 것을 말한다. **두 번째로 정보 은닉(Information Hiding)** 이라는 개념은 각 모듈들이 다른 모듈들에 대하여 독립적이며, 한 쪽 모듈을 변경하여도 다른 모듈의 내용에는 영향을 미치지 않는다는 것이다. **세 번째로 구조화(Structuring)**이다 소프트웨어의 구성에 계층적인 구조를 사용함으로써, 상위 모듈이 하위 모듈을 활용하는 것을 말한다. 네 번째로 **단계적 상세화(Stepwise Refinement)**이다. 프로그램이 하향식으로 진행됨에 따라 점점 내용이 구체화되는 것이다. **다섯 번째로 모듈화(Modulization)**이다. 하나의 시스템을 여러가지 프로그램, 서브시스템, 모듈 등으로 나누어 정의하고, 이것들을 개별적으로 설계한다. 여기서 모듈이란 최소한의 프로그램 단위라고 볼 수 있다.

객체지향시스템 분석 설계

객체지향 분석 설계를 이해하기 위해 먼저 객체라는 것을 알아야한다. ‘객체’란 실제 현실에서 어떤 구체적인 의미를 구성하는 하나하나의 실체 단위로 특정 사물이나 개념이며 클래스의 인스턴스를 말한다. 또는 필요로 하는 데이터와 그 위에서 수행되는 함수들을 가진 작은 모듈이라고도 말한다. 즉 객체 지향적이라는 것은 현실 세계에 실재하는 사물, 즉 객체를 지향한다는 것이다. 이 말의 뜻은 객체들을 프로그램에 응용하여 객체 지향적인 시스템을 설계한다는 의미이다. 객체 지향 분석 및 설계 프로세스에서는 객체를 의미, 인터페이스, 응용, 베이스 클래스 들로 분류하여 나타낸다. 클래스는 표현 대상들의 공통적인 특징에 대한 서술(추상)을 말한다.

객체지향 분석 및 설계의 단계는 크게 4개로 나눌 수 있다. **첫 번째로** 문제영역에서 클래스로 묶을 수 있는 것들을 식별한다. **두 번째로** 이 것들을 클래스나 오브젝트로 일반화한다. **세 번째로** 대상들 간의 관계(상속 관계 등)을 설정하고. **마지막으로** 이들을 서로 상호작용하도록 하게 한다. [그림 2]는 앞에서 기술한 단계들을 세분화한 것이다. 계획->분석->설계->구현->테스트 및 검증 순으로 다른 소프트웨어 방법론과 개발 순서들을 가지지만, 분석 작업을 할 때 객체 모델링-> 동적 모델링 -> 기능 모델링 순으로 모델링을 한다는 점, 설계 단계에서 객체를 설계한다는 점을 차이점이라고 할 수 있다.



[그림 2. 객체지향 분석 및 설계의 단계]

객체 지향 분석 및 설계의 장점은 규모가 큰 대형 프로젝트 개발에 적당하다는 점, 소프트웨어의 재 사용률, 확장성이 높아 생산성이 증가한다는 점, 객체를 재사용하며 프로그래밍할 수 있어 빠르게 개발이 가능하다는 점, 유지보수성이 향상되고, 자연적인 모델링이 가능하다는 점 등 여러가지 장점이 있다. 하지만, 객체를 설계하기 어렵다는 단점과, 객체의 규모가 크기 때문에 실행 속도가 느리다는 단점이 있다.