

2주차 1차시 소프트웨어 생명주기

【학습목표】

1. 소프트웨어의 단계적 생명 주기를 설명할 수 있다.
2. 요구분석을 단계별로 구분할 수 있다.

학습내용1 : 소프트웨어

1. 개요

* 소프트웨어란?

- 단순 컴퓨터 프로그램이 아닌 프로그램의 개발과 운영 및 유지보수에 관련된 모든 정보와 작업 포함
- 얼마나 정확하고 효율적으로 개발 및 사용, 관리가 이루어지는가가 중요

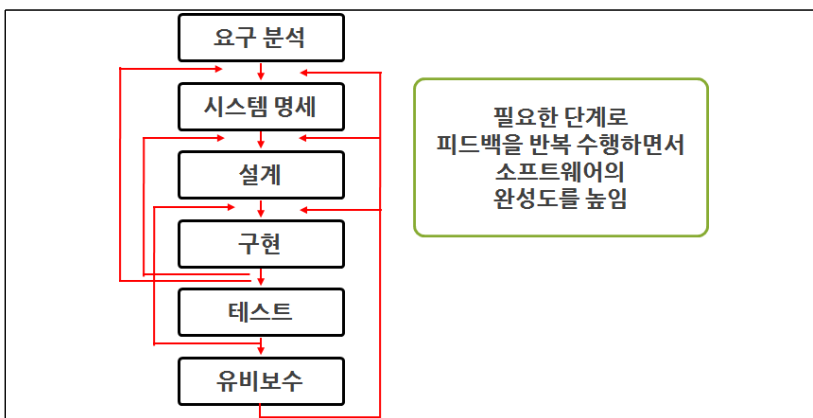
2. 성공적인 소프트웨어 개발은

- ① 개발할 소프트웨어를 정확히 이해
- ② 사용할 자료와 자료 간의 연산 관계를 분석하여 자료구조를 정의
- ③ 가장 효율적인 구조를 고려하여 개발

학습내용2 : 소프트웨어 생명주기

* 소프트웨어 생명주기(Software Life Cycle)

- 소프트웨어를 체계적으로 개발하고 관리하기 위해서 개발 과정을 단계별로 나누어 구분한 것으로 일반적으로 6단계로 구분



1. 요구분석 단계

- ① 문제 분석 단계
- ② 개발할 소프트웨어의 기능과 제약조건, 목표 등을 소프트웨어 사용자와 함께 명확히 정의하는 단계
- ③ 개발할 소프트웨어의 성격을 정확히 이해하고 개발 방법과 필요한 개발 자원 및 예산 측정
- ④ 결과물로 요구명세서 작성

2. 시스템 명세

- ① 시스템이 무엇을 수행해야 하는가를 정의하는 단계
- ② 입력 자료, 처리 내용, 생성되는 출력을 정의
- ③ 시스템 기능 명세서를 가능한 정확히 작성 의견 차이나 재개발 방지

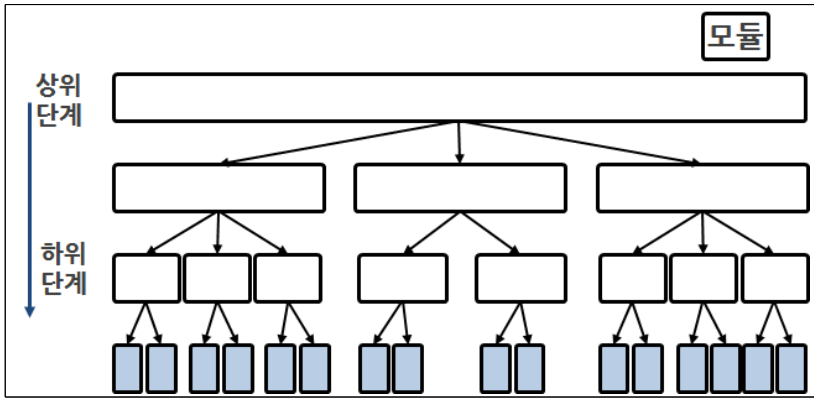
3. 설계 단계

- ① 시스템 명세 단계에서 정의한 기능을 실제 수행할 수 있도록 수행 방법을 결정
- ② 개발 목표 기능을 어떻게 구현할 것인지 구체적으로 정의



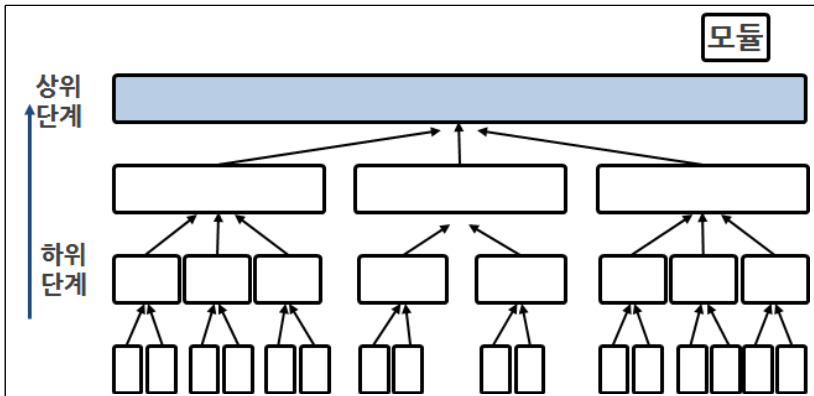
1) 설계 방법

- ① 하향식 설계 방법
 - 문제 정의 단계에서 시작하여 각 단계를 내려가면서 구체적으로 정의하고 세분화하는 방법
 - 분할 정복 방식의 설계 방법



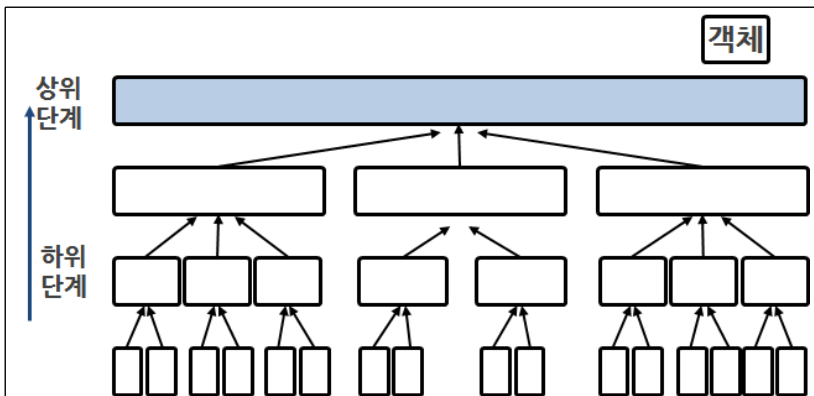
② 상향식 설계 방법

- 하위단계의 작은 단위의 문제를 먼저 해결하고 이를 이용하여 상위단계의 큰 단위의 문제를 해결하는 방법
- 하위단계 개발 시 기존 알고리즘을 재사용하는 경우 개발기관과 비용 단축



③ 객체지향 설계 방법

- 하위단위의 문제해결 도구를 객체로 만들어 재사용하는 방법으로 전체 문제를 해결하는 방법



4. 구현 단계

* 구현단계란?

- 설계 단계에서 결정한 문제 해결 방법(알고리즘)을 특정 프로그래밍 언어를 사용하여 실제 프로그램을 작성하는 단계
- 프로그래밍언어, 프로그래밍 기법과 스타일 및 순서 등 결정 필요

1) 프로그래밍 기법

① 구조화 프로그래밍

- 지정문과 조건문, 반복문 만을 사용하여 프로그램을 작성
- 순차구조, 선택구조, 반복구조의 세가지 제어구조로 표현
- 구조가 명확하여 정확성 검증과 테스트 및 유지보수 용이

② 모듈러 프로그래밍

- 프로그램을 여러 개의 작은 모듈로 나누어 계층 관계로 구성하는 프로그래밍 기법
- 각각의 모듈은 구조화 프로그래밍 기법으로 작성, 하나의 기능만 수행토록 구성
- 모듈별로 개발과 테스트 및 유지보수 가능, 모듈의 재사용 가능

5. 테스트 단계

* 테스트 단계란?

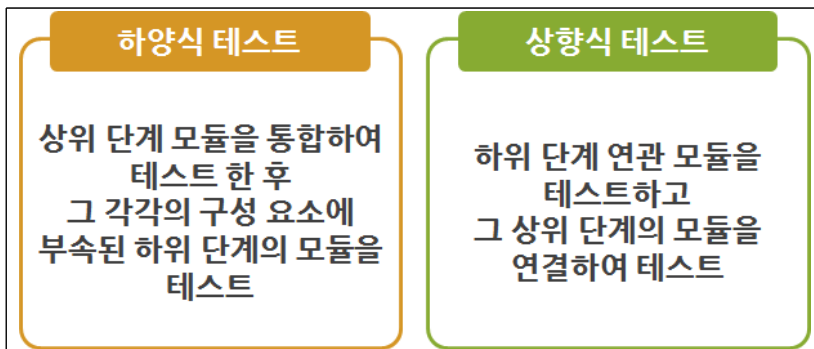
- 개발한 시스템이 요구사항을 만족하는지, 실행결과가 예상한 결과와 정확하게 맞는지를 검사하고 평가하는 일련의 과정
- 숨어있는 오류를 최대한 찾아내어 시스템의 완성도를 높이는 단계

① 1단계 - 단위 테스트(Unit Test)

- 시스템의 최소 구성요소가 되는 모듈에 대해서 개별적으로 시행

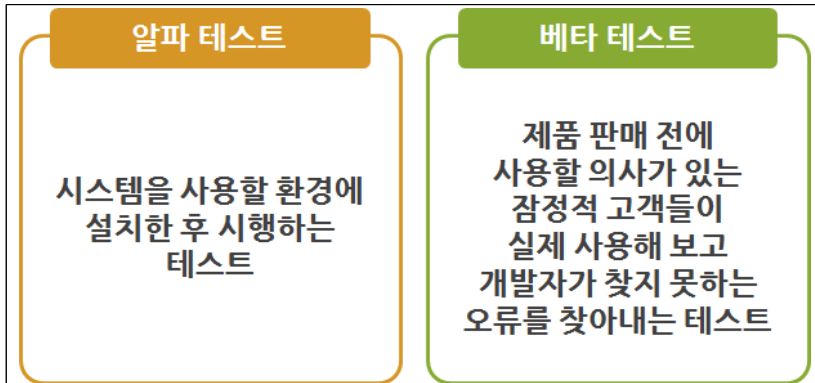
② 2단계 -통합테스트(Integration Test)

- 단위 테스트를 통과한 모듈을 통합하여 전체 시스템을 완성하기 전에 모듈 간의 인터페이스를 통합적으로 테스트
- 구성요소 연결을 점진적으로 확장하면서 테스트 시행



③ 3단계 -인수테스트

- 완성된 시스템을 인수하기 위해서 실제 자료를 사용한 최종 테스트
- 시스템이 실질적으로 사용되기 전에 마지막으로 수행되는 테스트



6. 유지보수 단계

〈시스템이 인수되고 설치된 후 일어나는 모든 활동〉

- 프로그램 오류 수정에서 디자인 수정, 새로운 기능 추가 등 포함
- 소프트웨어 생명주기에서 가장 긴 기간

1) 유지보수의 유형

① 수정형 유지보수

- 사용 중에 발견한 프로그램의 오류 수정 작업

② 적응형 유지보수

- 시스템과 관련한 환경적 변화에 적응하기 위한 재조정 작업

③ 완전형 유지보수

- 시스템의 성능을 향상시키기 위한 개선 작업

④ 예방형 유지보수

- 앞으로 발생할지 모를 변경 사항을 수용하기 위한 대비 작업

7. 개발된 소프트웨어 품질 평가

정확성	요구되는 기능들을 정확하게 수행하는 정도를 평가
유지 보수성	효율적 유지 보수의 정도를 평가
무결성	바이러스 등의 외부 공격에 대한 보안성 평가
사용성	사용자가 쉽고 편리하게 사용할 수 있는가에 대한 평가

【학습정리】

1. 소프트웨어 생명주기란 소프트웨어 개발단계와 방법을 이해하고 체계적으로 개발 · 관리하기 위해서 소프트웨어 개발과정을 몇 단계로 구분한 것이다.
2. 일반적으로 요구분석, 시스템 명세, 설계, 구현, 테스트, 유지보수의 6단계로 구분한다.
3. 소프트웨어의 품질은 정확성, 유지보수성, 무결성, 사용성 등에 따라 평가된다.