

## 소프트웨어 공학

소프트웨어 공학이란? - 소프트웨어의 개발, 운용, 유지보수등의 생명주기 전반을 체계적이고 서술적이며 정량적으로 다루는 학문이다. 공학을 소프트웨어에 적용하는 것이다.

소프트웨어 개발 생명주기 모델 - 소프트웨어를 어떻게 개발할 것인가에 대한 전체적인 흐름, 주먹구구식, 폭포수, 프로토타이핑, 나선형  
소프트웨어 개발 방법론이란 - 소프트웨어를 생산하는데 필요한 반복적인 과정들을 정리한것(메뉴얼), 구조적프로그래밍, 객체지향, 고속개발 방법론, 익스트림 프로그래밍, 스크럴

- 애자일 개발 프로세스 : 특정한 개발 방법론을 가리키는 말은 아니다. 어떻게 하면 프로그램을 효율적으로 만들수 있을까 하는 방법론  
UML - 통합모델링언어, 표준화된 범용 모델링언어, 산출물을 명세화, 시각화, 문서화 할때 사용, 방법론 + 모델 + 다이어그램 = UML

TDD - 테스트주도개발, 매우 짧은 개발 사이클을 반복하는 소프트웨어 개발 프로세스

PDD - 계획기반개발, 계획을 세우고 그기획을 실천하는데에 많은 시간과 노력을 할애하는 개발법

형상관리 - sw개발 및 유지보수과정에서 발생하는 소스코드, 문서, 인터페이스 등 각종 결과물에 대해 형상을 만들고 이들 형상에 대한 변경을 체계적으로 관리, 제어하기위한활동

버전관리 - 형상관리의 일부, 일반적인 소프트웨어 소스코드만의 형상을 관리하는것, 로컬, 중앙집중, 분산 등이 있다.

## 프로그래밍 언어

컴파일언어 - 컴파일 0,1로 바꾸는 작업, 번역기가 필요없다, 빠르다, 미리 번역되어있어 바로 전송하기만 하면된다.

바이트코드언어 - 32,64비트 운영체제에 전부호환, 미리 컴파일 해둔다, 그 플랫폼에 맞게 번역해준다.

인터프리터 언어 - 동시통역, 실행 순간부터 번역시작(사용자 입장에서는 느리게 다가올수있음), 서버쪽 사용

프로그래밍 패러다임

객체지향 프로그래밍 패러다임 - 프로그램을 상호작용하는 객체들의 집합으로 표현, 프로그램 명령어의 목록으로 보는 시각에서 벗어나 여러개의 독립된 단위

함수형 프로그래밍 패러다임 - 프로그램을 상태값을 지니지않는 함수값들의 연속으로 표현

## 객체지향 프로그래밍 패러다임

객체 - 클래스의 속성과 행위를 부여한것이 객체

클래스 - 객체의 모음, 객체가 가질수 있는 속성과 행위를 정의하는 툴(템플릿, 설계도), 객체의 상위

객체의 생성 - 객체가 속성과 행위를 가짐으로서 실질적으로 구현할수있다.

## 프로그래밍 용어

software developer - 개발자

server / client - 서버(정보제공) > 클라이언트(정보받음)

from-end / back-end - 사용자편의, 시각적인 / 서버, 시각적인 후의 데이터관련

threed - 프로세스 내에서 작업이 실행되는 흐름의 단위, 멀티스래드 = 다중스래드

library - 특정기능을 수행할수있는 클래스 또는 함수의 집합체, 수학, 애니메이션, 문자열 라이브러리, 관련된 기능의 집합체

API - 응용소프트웨어 프로그래밍 접합부, 응용소프트웨어와 프레임워크 사이의 중간매체(방법)

framework - 포괄적인 추상구조, 이를 지원하는 소프트웨어 플랫폼

디자인패턴 - 프로그램 개발에서 자주나타나는 과제를 해결하기위한 방법 중 하나, 과거의 소프트웨어 개발과정에서 발견된 설계의 노하우에 이름을 붙여 이후에 재사용하기 좋은 형태로 묶어 정리한것

IDE - 통합개발환경(x-code, 파이썬)

SDK - 소프트웨어 개발에 필요한 도구의 모음