9 월 9 일(금)

프로그래밍 기초상식, SW 공학개론

소프트웨어 공학

- 소프트웨어의 개발, 운용, 유지보수 및 폐기에 대한 체계적인 접근방법

- **공학이란?**
  우리가 아는 지식을 동원해서 필요한 것을 만듬

소프트웨어 개발 생명주기 모델(Software Development Life Cycle Model)

- 소프트웨어를 어떻게 개발할것인가에 대한 전체적인 흐름
- 주먹구구식, 폭포수, 프로토타이핑, 나선형

- **폭포수**

Requirements Product	요구분석	Requirements Document
Design	문서화	Software Architecture
Implementation	소프트웨어 구성	Software
Verification	검증	
Maintenance	유지보수	

  문제점
  비효율적인 면이 있는것 같다. Design 에서의 시간끌기
  Design 이 잘못되면 다시 처음부터 해야됨
  요구사항의 변경 및 추가사항

  장점
  복잡하지 않고 단순하면 효율적

- **프로토타이핑**

  요구분석
  prototype 설계
  prototype 개발
  prototype 평가
  구현
  인수 및 설치

  문제점
  품질 저하
  작은 프로젝트에서는 비효율적일수 있다.
  시간을 기다리지 못한다
  비용문제
  요구가 많이 바뀌어서 완성을 못함

- **나선형 모델**

  계획 및 정의
  위험분석
  구축
  고객평가
  고객과의 의사소통
  폭포수를 작게 쪼갬

  문제점
  하나씩 만들다 보니 나중에 기술적 부채가 생김

소프트웨어 개발 방법론

- 소프트웨어를 생산하는데 필요한 반복적인 과정들을 정리한 것
- 구조적 프로그래밍, 객체지향, 고속 개발 방법론, 익스트림 프로그래밍(Agile), 스크럼(Scrum)…

- **애자일 개발 프로세스**
  다른 고전적인 방법론과 구별되는 가장 큰 차이점은 less document-oriented, 즉 문서를 통한 개발 방법이 아니라, code-oriented, 실질적인 코딩을 통한 방법론

  앞을 예측하며 개발을 하지 않고,
  일정한 주기를 가지고 끊임없이 프로토타입을 만들어내며 그때 그때 필요한 요구를 더하고 수정하여 하나의 커다란 소프트웨어를 개발해 나가는 Adaptive style

  특정 개발 방법론을 가리키는 말은 아니고
  "애자일(Agile= 기민한, 좋은것을 빠르고 낭비없게 만드는 것) 개발을 가능하게 해 주는 다양한 방법론 전체를 일컫는 말

- **익스트림 프로그래밍(eXtreme Programming, XP)**
  비즈니스 상의 요구가 시시각각 변동이 심한 경우에 적합한 개발 방법이다. 애자일 개발 프로세스라 불리는 개발 방법 중의 대표적인 하나로 꼽히며,비교적 적은 규모의 인원의 개발 프로젝트에 적용하기 좋다. 개발 문서 보다는 소스코드를, 조직적인 개발의 움직임 보다는 개개인의 책임과 용기에 중점을 둔다.

- **스크럼(Scrum)**
  애자일 소프트웨어 공학 중의 하나이다. 솔루션에 포함할 기능/개선점에 대한 우선 순위를 부여한다. 개발 주기는 30 일 정도로 조절하고 개발 주기마다 실제 동작할 수 있는 결과를 제공하라. 개발 주기마다 적용할 기능이나 개선에 대한 목록을 제공하라. 날마다 15 분 정도 회의를 가져라. 항상 팀 단위로 생각하라. 원활한 의사소통을 위하여, 구분 없는 열린 공간을 유지하라.

- **UML(Unified Modeling Language)**
  통합 모델링 언어
  표준화된 범용 모델링 언어
  객체 지향 소프트웨어 집약 시스템을 개발할 때 산출물을 명세화, 시각화, 문서화할 때 사용
  S/W 개발 방법론 + 모델 + 클래스 다이어그램

- **TDD(Test-driven development)**
  테스트 주도 개발
  매우 짧은 개발 사이클을 반복하는 소프트웨어 개발 프로세스.
  코드로 검증 해본다.

  예) 사운드를 넣고싶다
  1.결함을 점검하는 자동화된 테스트 케이스를 작성
  2.케이스를 통과하기 위한 최소한의 양의 코드를 생성
  3.새 코드를 표준에 맞도록 리팩토링

  리팩토링
  기존의 코드를 지금의 상황에 맞게 변환
  컴퓨터에게 테스팅 할 수 있게 코드 작성
  최소한의 오류작성

  문제점
  손이 많이감

- **PDD(Plan – Driven Development)**
  계획 기반 개발
  계획을 세우고 그 계획을 실천하는데에 많은 시간과 노력을 할애하는 개발 방법

형상관리

- SW 개발 및 유지보수 과정에서 발생하는 소스코드, 문서, 인터페이스 등 각종 결과물에 대해 형상을 만들고, 이들 형상에 대한 변경을 체계적으로 관리, 제어하기 위한 활동
- 새로 합류한 사람에게 이해하기 쉽게 하려고
- 예) 이전 버전들을 관리
- rollback
  데이터베이스에서 업데이트에 오류가 발생할 때, 이전 상태로 되돌리는 것을 말한다.

- **버전관리**
  형상관리의 일부
  일반적인 소프트웨어 소스 코드만의 형상을 관리하는 것
  형상관리이면서 버전관리인 SVN(Subversion), Git, Mercurial, CVS….

- **프로젝트관리**
  문서, 일정, 예산, 인력, 고객, 위험, 품질 관리

프로그래밍

- 프로그램을 만드는데 사용하는 언어
- 기존에 있는 언어의 문제점을 해결하기 위해서 많은 프로그래밍 언어가 탄생됐다.

프로그래밍 언어의 종류

- 고급언어(사람이 알아듣고 작성할 수있는 높은수준의 언어)에서 저급언어로 변환되는 과정에 따른 분류

- **컴파일 언어 C, C++, Go**
  (컴파일을 하고 난 뒤 실행파일을 준다 사용자에게 배포)
  빠르다 요청한 사람이 바로 번역본을 받을수 있으니
  번역기가 필요 없다

리버싱 하기가 어렵다
번역하는데 오래걸린다.
수정이 어렵다.
번복하기가 어렵다.(패치)
게임에서 주로 사용

- **바이트코드 언어 Java, C#**
  번역가가 알아들을수 있는 말로 미리 컴파일해놓고 요구에 맡게 번역을 해줌
  플랫폼에 맡는 컴파일

  문제점
  중간단계 컴파일이 있어 느림
  번역기가 그 시스템(자바가상머신 JVM (Java virtual machine))이 꼭 있어야함
  리버싱하기가 쉽다 보안 x
  무겁다

- **인터프리터 언어 BASIC, JavaScript, Python, Ruby**
  동시통역 사람들이 필요로 할때 번역
  서버쪽, 유지보수가 쉽다.

프로그래밍 패러다임(관점)에 따른 분류

- **객체지향 프로그래밍 패러다임**
  프로그램을 상호작용하는 객체들의 집합으로 표현

- **함수형 프로그래밍 패러다임**
  프로그램을 상태값을 지니지 않는 함수값들의 연속으로표현

객체지향 프로그래밍 패러다임

- 컴퓨터 프로그램을 명령어의 목록으로 보는 시각에서 벗어나 여러 개의 독립된 단위, 즉 "객체"들의 모임으로 파악하여, 객체간의 상호작용으로 프로그램의 동작을 구현하고자 하는 것

- 컴퓨터의 목적이 다양해짐 다양한 모습의 프로그램이 필요해 졌다.
- 모든것이 객체다

- **클래스와 객체**
  클래스 객체(실질적인 형태)(클래스의 객체화)

- **클래스 Class**
  객체가 가질 수 있는 속성과 행위를 정의하는 틀
  (템플릿, 설계도)
  공통적인 특징을 가지고 있는 것
  정의를 할 수 있다

- **object 객체, 물건**
  의식이나 행동의 대상
  형체, 물체 ,물질
  상태(속성,변수)와 행위(기능,함수)를 가지는 형체
  객체는 클래스에 의해 주기억장치에서 생성 된다.
  객체마다 속성,변수가 다를 수 있다.

프로그래밍 용어

- **개발자-Developer**
  무언가를 만들어 내는 사람

- **Server/Client**
  서버 > 데이터 제공 > 클라이언트
  정보를 제공하는 주체와 받는 주체
  상대적인것
  Back-End
  뒷단

- **Front-End**
  앞단

EndUser, Client 와 맞닿아 있는 것
서버안에서도 Front-end(API) Back-end 가 있다.

Thread

- 하나의 프로레스 내에서 작업이 실행되는 흐름의 단위
- 여러가지 작업을 진행하는것 처럼보임 이들의 작업단위

멀티 스레드

- Multi Thread – 다중 스레드
- 멀티 스레드(한개의 프로세스 다른 기능을 순식간에 번갈아가며 실행 예: 애니 영상,소리) vs 멀티 태스크(책을 읽으며 음악을 듣는다 여러가지의 프로세스)

Library

- 도서관
- 특정 기능을 수행할 수 있는 클래스 또는 함수의 집합체
- 수학 라이브러리
- 애니메이션 라이브러리
- 문자열 라이브러리

API(Application Programming Interface)

- 응용 소프트웨어 프로그래밍 접합부
- 응용 소프트웨어와 프레임워크 사이의 중간매체(방법)
- 소프트웨어 간의 통신을 위해 메시지를 전달하는 방식 등이 결정된 것
- 통로 접합부(날씨 API 로 접근하면 구글이 전달해줄게)
- 운영체제(시스템)에게 파일을 저장해주세요 할 때도 API 를 통해서 요청함
- 서버 프로그램으로 부터 API 를 통해 데이터 전달

Framework

- 구조적으로 고정된 부분을 재사용할 수 있도록 하고, 응용별 특정 기능을 추가적인 사용자 작성 코드에 의해 선택적으로 구현가능하도록 하는 포괄적인 추상 구조, 그리고 이를 지원하는 소프트웨어 플랫폼
- 응용소프트웨어를 만드는 밑바탕(플랫폼)

예를들면

프레임워크 – 국가
프레임워크의 하위 프레임워크 – 정부기관
API-국가기관
라이브러리- 직무전문가
프로그래밍 언어- 국가 통용 언어
레퍼런스 문서- 국가 조직도 및 기능설명서

회사를 설립하기 위해서 기반이 필요
소프트웨어를 만들기위한 기반
요청하는 창구를 API
어떠한 일을 하기위한 정보 Framework

Cocoa Touch Framework +obj
Coco 라는 나라 모양에 내가 원하는 일을 할 수 있는 기업을 세우는 것
Coco 라는 나라에서는 obj 또는 Swift 라는 언어를 통해 의사소통 할 수 있다.

디자인패턴

- 프로그램 개발에서 자주 나타나는 과제를 해결하기 위한 방법 중 하나
- 과거의 소프트웨어 개발 과정에서 발견된 설계의 노하우에 이름을 붙여 이후에 재이용하기 좋은 형태로 묶어서 정리한 것
- 디자인 노하우 MVC(Model-View-Controller)패턴(android, ios 에서 주로 사용), MVVM(Model-View-Controller- Model), Observer, Singleton, Prototype 등등 프로그램을 설계하는데에 있어 테크닉
  예) 클래스 는 이런 클래스가 있으면 좋더라

Reference Document

- 레퍼런스 문서
- API 에 대해 서술해 놓은 문서
- 프로그램 전반적인 설명

IDE

- Integrated Development Environment
- 통합 개발 환경
- 개발할 때 필요한 도구의 모음(소스편집기, 컴파일 등등 빌드 링크)

SDK(Software Development Kit)

- IDE 도 포함
- 소프트웨어 개발에 필요한 도구의 모음
- IDE + Framework +Tools…
- 플랫폼 제공자들이 시장 장악을 위해

학습링크

프로그래밍 패러다임
https://goo.gl/JyjX1H

객체지향 프로그래밍
https://goo.gl/AtI3vN, https://goo.gl/84dP6n, https://goo.gl/0wFeSZ, https://goo.gl/yX4y52

소프트웨어 개발자
https://goo.gl/1SwDO7

스레드, 태스크
https://goo.gl/e9lyOd, https://goo.gl/dEpfwb, https://goo.gl/9tnyy9

풀스택
https://goo.gl/Miarbd, https://goo.gl/chOiXq, https://goo.gl/5pKLi6, https://goo.gl/4rQw6R

웹 프로그래머
https://goo.gl/DsDQMv, https://goo.gl/bvAfCO

프로그래머가 되려면
https://goo.gl/D9Axo7, https://goo.gl/ixn1zW

과제 내용

- 버전관리란 무엇이고, 버전관리 방법에는 어떤 것들이 있는지 알아보기