

Binary Classification

프로그래밍언어란?

① 언어란 정보의 매개체

숫자/문자 : Primitive 언어 x

한글/English : 자연어 o 나(개방자) 언어 → 대상(Human) ⇒ Word, Grammar (뜻이유됨)

JAVA/J.S/C... : 프로그래밍언어 o 나(개방자) 언어 → 대상(JVM) ⇒ Term, Syntax (조금이라도 틀리면 Error)

② Boolean (true vs false)

- George Ball

기호 논리 학자 / 수학자 / 철학자.

'사로의 불의 사상 (참 vs 거짓) → 컴퓨터 과학 발전을 위한 기초.

⇒ 프로그래밍을 공부할때 이것을 Base로 한다.

ex) 디렉토리 vs 파일

확장자 (.exe ...) 가 있냐 없냐

Object?

ex)



가상 공간 → 화면

홍길동

존재 or Not?

존재하는 것은 다 거역제이다.

① 공간을 점유? or Not.

(메소드)

- 머들/쭈/주다사 : **Real_world_Object** (공부라는 기능수행) : 기능 수행을 위한 설계도 : class 가 설계되면 거역/
- 우기 : **Software_Object** (") → 설계한 것은 **Instance**.

- 공간 : **Context** (Bit 502)

⇒ 공간을 이동한다는 것 : **Context switching**.

종류	설계	저장공간	가변성
Real_world_Object	hard(우미)	디스크	X
Software_Object	Virtual(우미X)	메모리	O

변동 O vs 변동 X

- 변동 O : **Variable** → Console (load 해야함) / 비수한 개념 : **Dynamic** (동적)

- 변동 X : **Construct** → Code (save 해야함) / " : **Static** (정적)

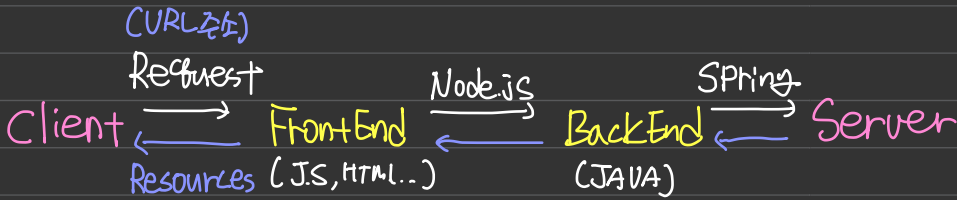
ex) 자바스크립트

ex) HTML

상수/변수는 항상 같이 존재

ex) UI 개판

Client Server Model

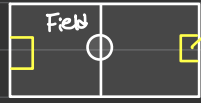


Ex) WWW, Email

객체지향 4대원칙

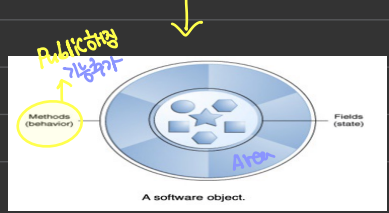
객체: 디스크에 있는 Real-world-Object (클래스가 객체)
 객체는 기능, 속성의 집합이다 (캡슐화)
 속성만 있으면? ex) class Person { int age, string name }
 ⇒ Entity (가 객체) 이다.
 : 데이터를 저장만 하수있지만, 의미는 모름 (컴퓨터 입장)
 마케팅과 같음, Entity 관계명은 1975 DB 시절부터 존재
 속성에 기능을 추가가 방법 } → Object → Collection
 - Method
 클래스에는 Object가 가져야 할 구성멤버가 선언되어야 함
 구성멤버: 필드/생성자/메소드

* Field vs Area



변환 O : Area
 변환 + : Field

가상머신으로 돌아가면



→ Method로 은닉 : Capsulation!

Ex) DOC vs DOM

DOC: 속성만 검색
 DOM: JS 추가로 기능 add ⇒ 객체지향 변환.

- 상속 (extends / implement)

객체지향에서 상속은 상위클래스의 특성을
 하위클래스에서 상속하고 더 필요한 속성을 확장해서 사용
 (계종관정 x 불꽃도 관정)
 * 구현 : is able to 인터페이스
 * 상위 클래스: 문제권 특성 상속 Good
 인터페이스: 구현강제 메서드 상속 Good.

- 추상화

구체적인 기존 관성영역에 있는 특성만을 가지고 재조합 하는 것
 ex) 병원에서 class Person은 환자 관련 도메인 (Entity) ↑
 상점 " " 고객 " "

- 다형성

Overriding / overloading
 ↓ ↓
 재정의 메소드 이름만 동일

스프링 4대원칙