

Lab 14

Q&A

<https://padlet.com/dcslabcp/0614-8v3iei1n9ifcvs5n>

오늘 수업 내용

- Java summary
- C++ vs. Java
- Programming
- SNUSearch

Java summary

Job interview에서 Java 역량을 확인할 때 던지는 시작 질문

#1: Java로 만들어 본 가장 큰 프로그램은?

#2: Overriding과 overloading의 차이점

#3: Java의 컴파일 과정에 대해 설명



Java summary

Job interview에서 Java 역량을 확인할 때 던지는 시작 질문

#1: Java로 만들어 본 가장 큰 프로그램은?

=> 후속 질문: 개발 과정에서 가장 어려웠던 점은? 해결 방법은?

#2: Overriding과 overloading의 차이점

=> 오버라이딩은 "메소드 재정의", 오버로딩은 "같은 이름 메소드 여러 개"

#3: Java의 컴파일 과정에 대해 설명

=> 컴파일러를 통해 바이트코드 생성하고 이를 JVM으로 실행
(JVM 내부 구조? JIT 개념은?)

* 신입 개발자 기술면접 질문 정리: <https://dev-coco.tistory.com/153>

Java summary

Java로 코딩 테스트를 준비한다면?

- String/StringBuilder, ArrayList 자유자재로 다루기
- 자료형 변환 자유자재로 다루기 (Wrapper classes, Upcasting/Downcasting)
- 유용한 Collection 적시 활용하기 (HashMap, HashSet)
- 입력을 받을 때는 Scanner보다는 BufferedReader 활용 ([link](#))
- 람다식과 정규표현식을 제대로 익혀두면 모든 언어에서 의미하게 활용 가능

[Pop quiz] 주어진 문자열을 Reverse 하는 방법은?



Java summary

Java로 코딩 테스트를 준비한다면?

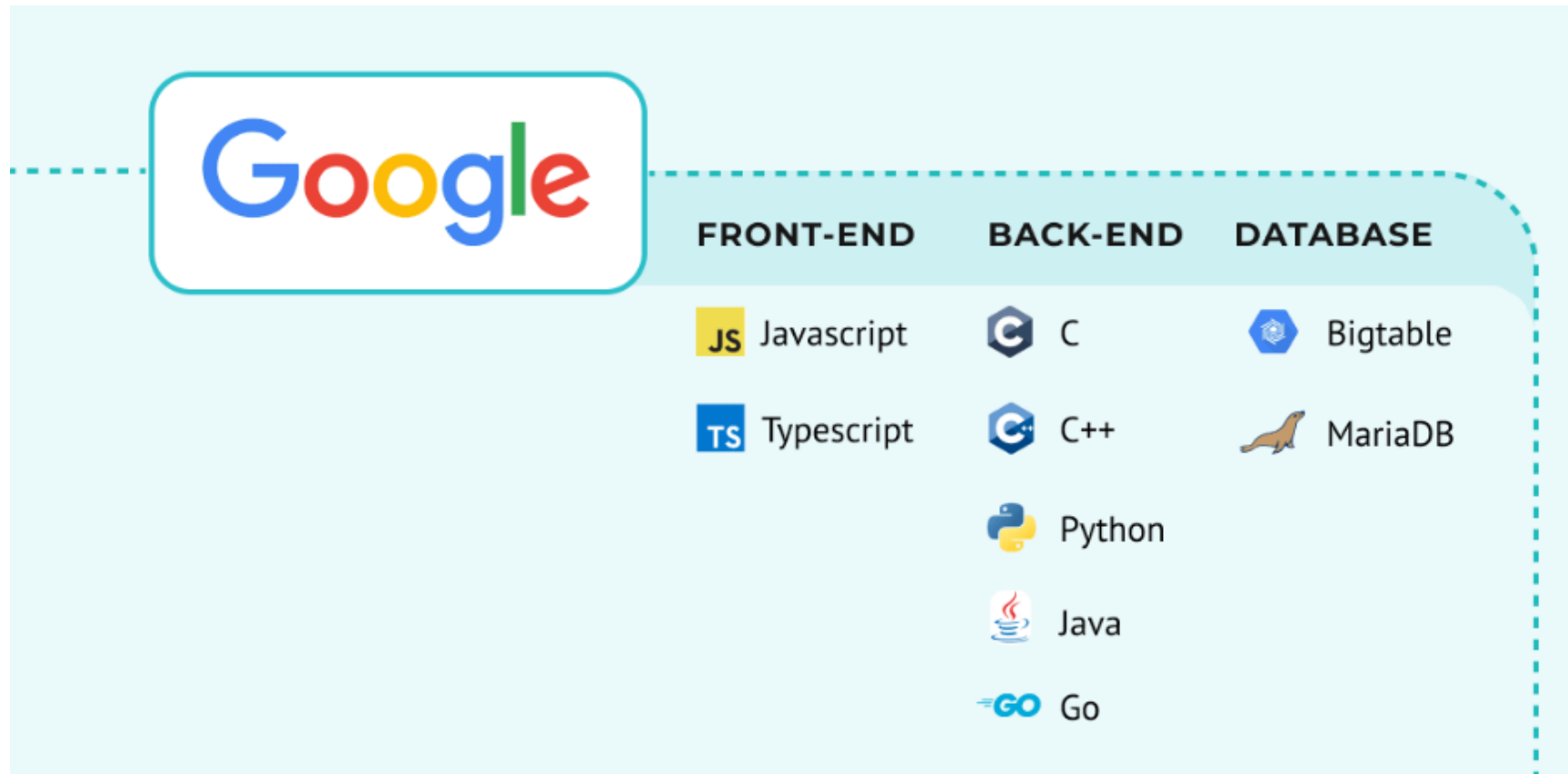
- String/StringBuilder, ArrayList 자유자재로 다루기
- 자료형 변환 자유자재로 다루기 (Wrapper classes, Upcasting/Downcasting)
- 유용한 Collection 적시 활용하기 (HashMap, HashSet)
- 입력을 받을 때는 Scanner보다는 BufferedReader 활용 ([link](#))
- 람다식과 정규표현식을 제대로 익혀두면 모든 언어에서 의미하게 활용 가능

[Pop quiz] 주어진 문자열을 Reverse 하는 방법은?

```
public static void main(String[] args) {  
    String str = "abcdef";  
    StringBuilder sb = new StringBuilder(str);  
    System.out.println(sb.reverse());  
}
```

C++ vs. Java

Which coding languages are used by the world's top companies?



<https://www.siliconrepublic.com/advice/coding-languages-top-companies-infographic>

C++

- 빠르다. 어렵다. 메모리 관리 부담
- Modern C++ (C++17, C++20)
- 모토: 프로그래머를 믿어라! (높은 자유도)

C++

- 빠르다. 어렵다. 메모리 관리 부담
- Modern C++ (C++17, C++20)
- motto: 프로그래머를 믿어라! (높은 자유도)

[7] 그러나 실상 C++11 이후의 **모던 C++** 기능을 가르쳐주는 곳이 매우 드물어서 C++의 객체 지향만 조금 배울 수 있다. C++의 객체 지향은 다른 언어에 비해서 좀 더 복잡하다. 제대로 다루기 위해서는 별도의 서적을 참고하는 것이 권장되며, 실제 프로젝트의 코드에 대해 탐구하고 경험해보는 것이 좋다. C++를 모르면 2학년부터는 강의를 전혀 이해할 수 없기 때문에 의외로 많은 학생들이 중도포기를 하게 되며 컴공을 나왔는데도 프로그래밍을 전혀 못하는 학생들이 수두룩하게 발생하고 있다.

쓸 때는 C++라고 쓰지만 읽을 때는 시 플러스 플러스, 혹은 줄여서 시플플이라고 읽는다. 한국에서는 사실상 속어에 가까운 **씨뽕뽕**로 불린다. '씨뽕뽕'로 발음되는 경우도 많다. 나이가 꽤 있는 교수 중에는 간혹 '시 더블 플러스'라고 읽는 경우도 있다. 미국에서는 그냥 씨 플러스 플러스 또는 확장자 명인 CPP(시피피)라고 읽는다.

한국에서 **컴퓨터공학과**를 다닐 경우 배울 수 있는 프로그래밍 언어 3개 중에 속한다. 나머지 둘은 C와 Java.[7] 학교마다 다르지만 1학년때 아예 다른 언어를 배우고 2학년 1학기때 Java, 2학기때 C++ 과목이 생기는 곳도 있다. 예시로 **서울대학교**의 경우에는 1학년 1학기에 Python으로 코딩을 입문하고, 1학년 2학기에 C언어로 어느 정도 큰 프로젝트를 짜 보고, 2학년 1학기에 C++과 Java를 배우면서 객체지향 프로그래밍을 배우게 된다.

<https://namu.wiki/w/C++>

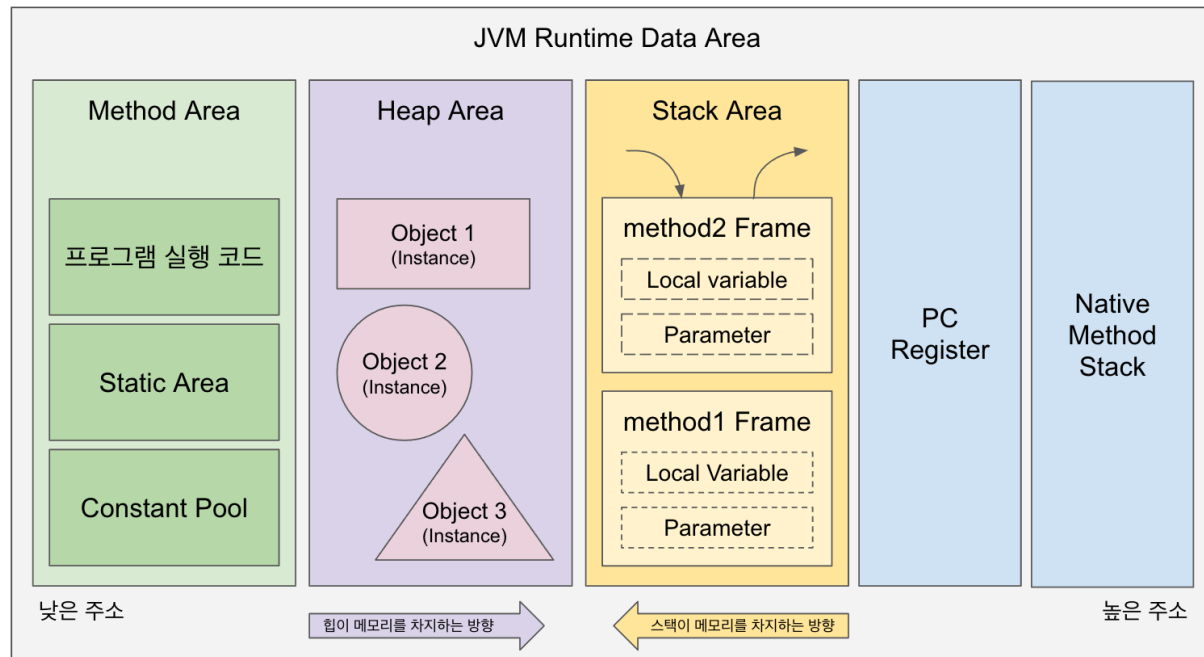
Java

- 덜 빠르다. 덜 어렵다. 메모리 부담 X
- 유료다. (오라클 Since 2010)
- OOP, JVM, JNI, GC, JIT, Spring, Maven, JavaFx

The longest hello world program

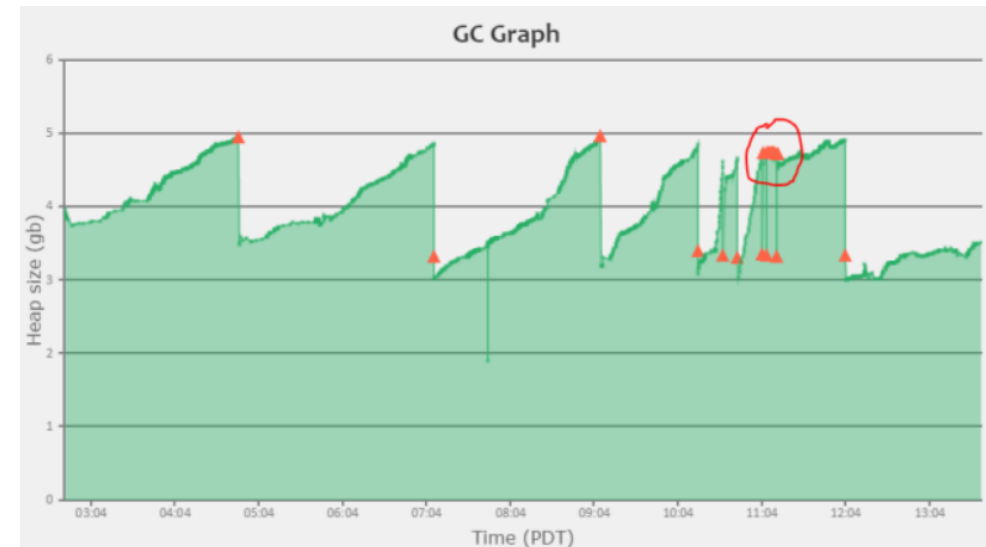
```
class Main {  
    public static void main(String args[]) {  
        System.out.println("Hello");  
    }  
}
```

<https://namu.wiki/w/Java/평가>



<https://www.devkuma.com/docs/jvm/memory-structure/>

Java GC graph example



C++ vs. Java with Go

- C++과 GO를 여러 방면에서 비교해봤을 때 장점이 많음

	C++		GO
성능	매우 좋음	≥	C++과 유사함
러닝 커브	어려운 편에 속함	<	쉬운 편에 속함
라이브러리 추가	일일이 추가	<	명령어 한줄로 쉽게 추가
메모리 관리	관리가 필요함	<	GC 지원
유닛 테스트	직접 구축	<	기본 지원
숙련도에 따른 편차	큰 편에 속함	<	적은 편에 속함

<https://devview.kr/2020/sessions/379>

C++ vs. Java with Rust

Q. C++과 Rust의 대표적인 차이점은 어떤 것이 있을까요?

- 악의적인 C++ 코드 작성을 통해 손쉽게 네트워크 카드를 불태울 수 있음
- Rust는 악의적인 코드가 실행되지 않도록 원천 차단하는 내부 매커니즘 제공 (결국은 컴파일러)
- C++은 런타임에 문제 상황이 곧잘 발생하는 반면,
- Rust는 컴파일 타임에 예상되는 문제를 훌륭하게 잡아줌 (단점: 컴파일 시간이 오래걸림)
- 이를 가능하게 하는 것은, Rust의 훌륭한 컴파일러 선생님의 활약 (에러 메시지 내 해결책도 제공)

Memory Safety problem!
(e.g. UAF, Use after free)

Q. Java와 Rust의 대표적인 차이점은 어떤 것이 있을까요?

- Java는 기본적으로 메모리 관리를 런타임에 GC가 담당하여 안전성 이슈가 없음
- Java에서 Iterator를 구현하면 이는 파일 시스템 차원으로 문제 안생기게 보장함
- Rust는 GC 역할 없이도, 메모리 관리 부분을 컴파일러가 사전에 효과적으로 제어함
- 사실 HW 사양에 큰 제약이 없는 경우라면 Rust 대신 Java를 쓰면 좋음 (HotSpot JVM 매우 훌륭) (Rust의 경쟁자는 C++이 아닌 Java)
- 관점에 따라 deallocation을 모아서 하는 GC 방식의 효율이 더 좋을 때도 있음

<https://docs.google.com/presentation/d/15Ar0IUqIXknB7wIZ2xPwiY-5Y1-aj3CaF5tSo63NLZo/>

Programming

프로그래밍을 잘하려면?

#1. 한 언어를 깊이 있게 이해하기 (컴파일 하는 언어 중 하나, C++ or Java)

=> 그러면 다른 언어는 나의 주 언어와 비교해가며 쉽고 빠르게 습득 가능

=> 이를 위해서는 규모 있는 프로그램 개발 경험이 필수적 (토이 프로젝트)

#2. 복잡하고 좋은 코드 많이 보기 (오픈 소스)

#3. 실력자 멘토 만나기 (인턴, 동호회)

=> 오픈 소스 멘티 모집: <https://www.oss.kr/notice/show/7b08b07d-b566-42cb-9193-da299ea4197c>

#4. 개발 관련 input을 꾸준히 축적해가자

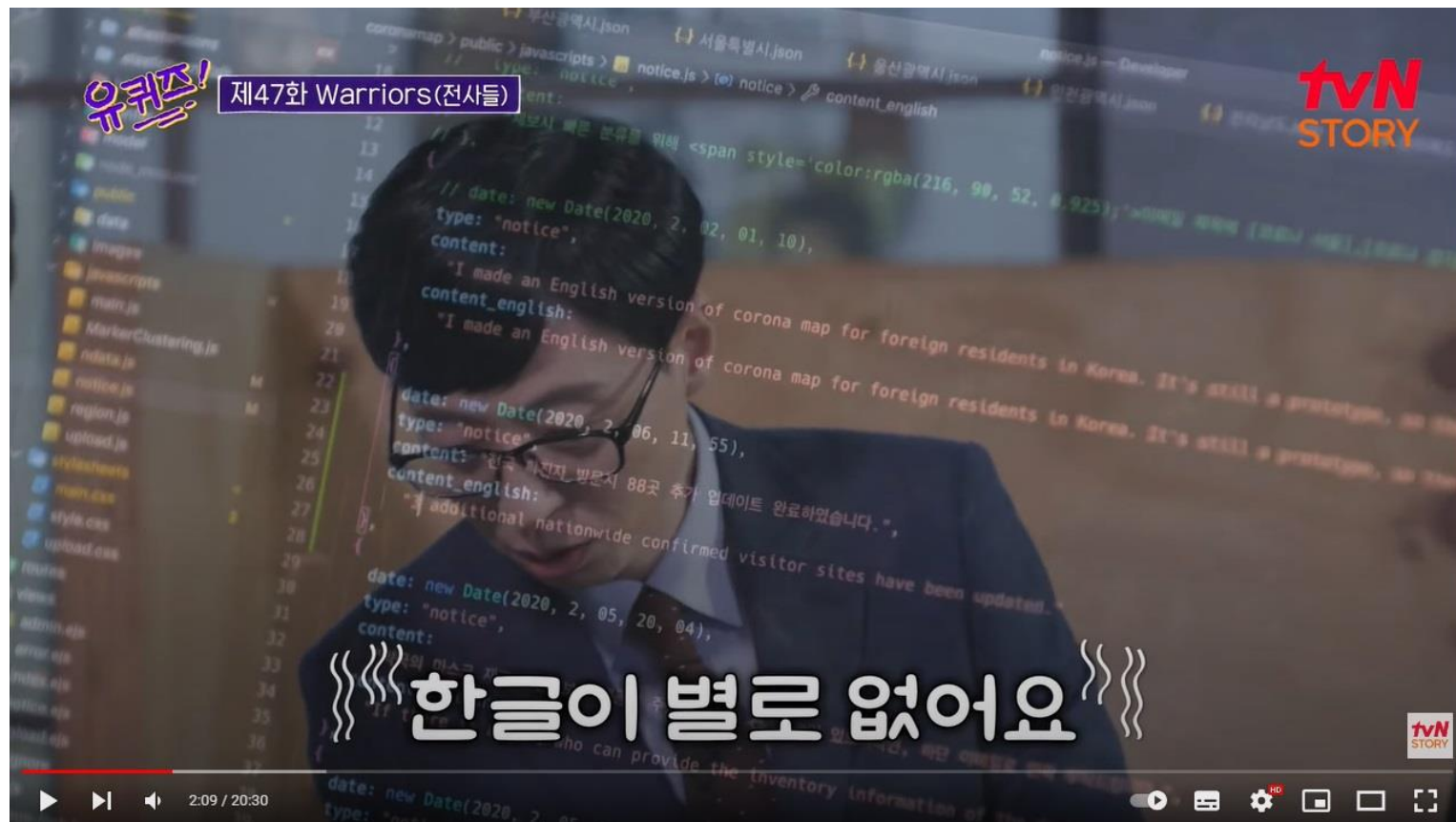
=> 추천 뉴스레터 GeekNews Weekly: <https://news.hada.io/weekly>

Programming

토이 프로젝트 사례 소개

라이브 코로나 맵

<https://livecorona.co.kr/> (code)



"수익은 거절했어요" 대학생이 하루만에 개발한 '코로나 맵' 어플 개발자를 만나다

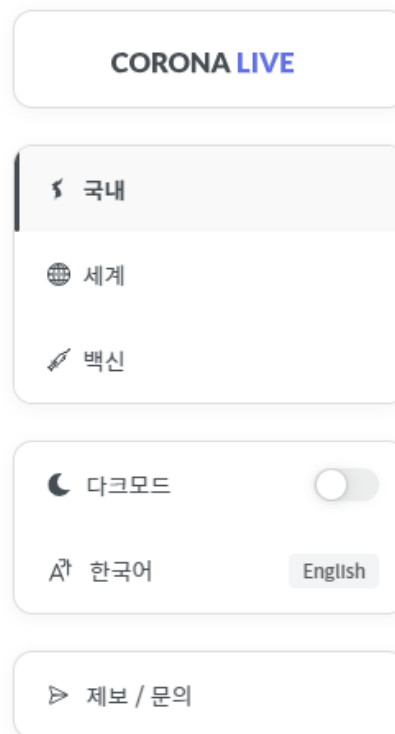
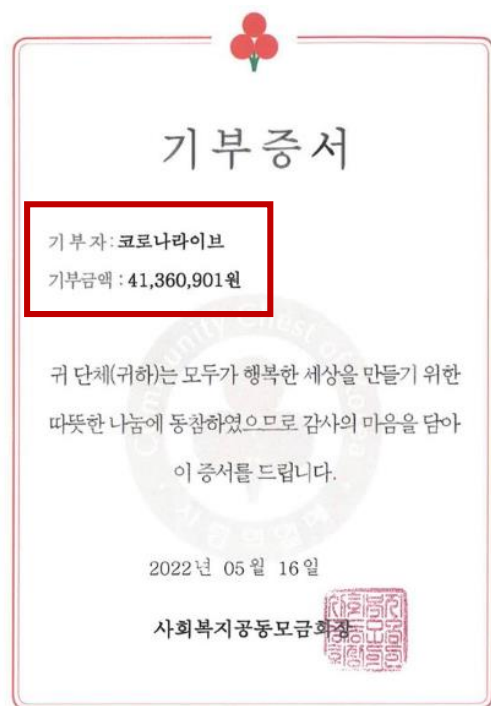
<https://youtu.be/o-MxaG5QrSM>

Programming

토이 프로젝트 사례 소개

코로나 라이브

<https://corona-live.com/>



Programming

토이 프로젝트를 위한 도메인/호스팅 추천

#1. <https://hosting.kr> (Java 지원)

#2. <https://cafe24.com> (가격 경쟁력)

Slim

소규모 웹사이트, 블로그 운영에 적합한
시작 패키지

1,000원

신청하기

- ✓ 웹사이트 1개
- ✓ 도메인 1개
- ✓ 300MB SSD
- ✓ 월 10GB 트래픽 사용
- ✓ MySQL DB 1개
- ✓ 무료 도메인 (.page.place)
- ✓ 무료 SSL 인증서

절약형

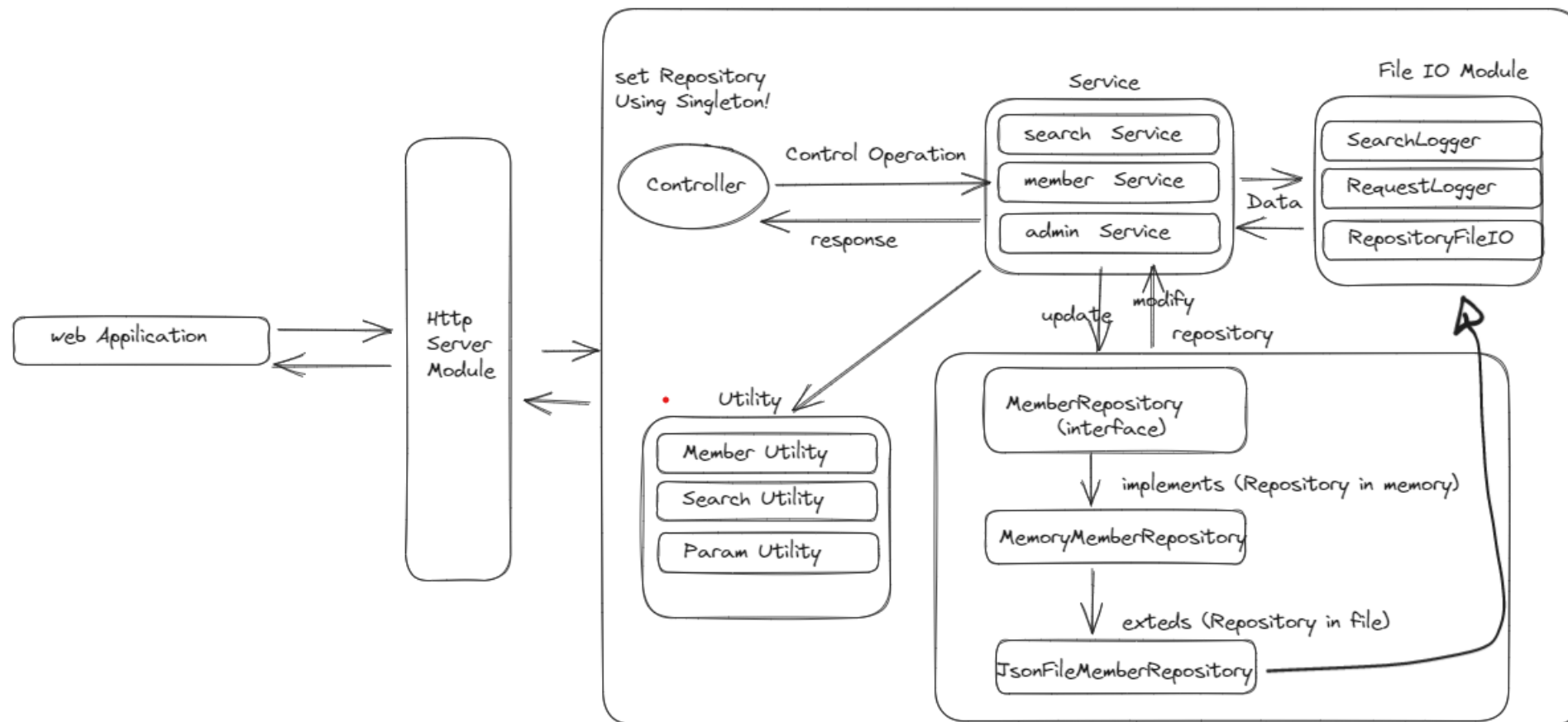
500원/월

누구나 부담 없는 시작

- ✓ 700MB 웹용량
- ✓ 1.6GB 트래픽 용량
- ✓ 무제한 데이터베이스

SNUSearch

한 학생이 코드와 함께 제출한 디자인 그림



SNUSearch

A few things to think

- 구현보다 방향성 잡기와 설계 그리기가 더 중요
- 좋은 코드를 많이 보는 것과 직접 복잡한 코드를 작성하는 것 병행 필요성
- 나의 강점은 R&D 중 어디에 있는지 발견하기 (Research vs. Development)

한 학기 동안 고생 많으셨습니다.