



학부생으로서 어떻게 공부해야 하는가?

19학번 홍혁준



홍혁준

- Zeropage 29기
- 우아한테크코스 5기 BE
- 4-1 다니는 중
- 우테코 캠퍼스 스터디(초록 스터디) 운영
- AUSG 9기 운영진



학습 테스트로 배우는 스프링 입문 - 초록 스터디 2기

- 프로젝트 초기 설정
 - 프레임워크와 프로젝트
 - Gradle 프로젝트 구축하기
- Spring MVC
 - API Web Application 만들기
 - HTTP
 - Spring Annotation MVC
 - 학습 테스트 - MVC Response
 - 학습 테스트 - CRUD API
 - 1단계 - 종 휘연
 - 2단계 - 예약 주가 / 취소
 - 3단계 - 예약 처리
- Spring JDBC
 - Database 적용하기
 - Spring Database Access
 - 학습 테스트 - JdbcTemplate
 - 5단계 - 데이터베이스 적용하기
 - 6단계 - 데이터 조회하기
 - 7단계 - 데이터 추가/삭제하기
 - 8단계 - 시간 관리 기능
 - 9단계 - 기본 코드 수정
 - 10단계 - 계층화 리팩토링
- Spring Core
 - Spring Context 담당하기
 - Spring Bean과 Container
 - Layered Architecture
 - 학습 테스트 - Spring Bean
 - 3단계 - 기존 코드 수정

[Spring Core] 김선호 미션 제출합니다. #82

1-Merged boorownie merged 42 commits into next-step:shkims5 from SHKims5:step3 · on Feb 16

Conversation 36 · Commits 42 · Checks 0 · Files changed 20

SHKims5 commented on Jan 26

3주차 미션(Spring Core)

시간 데이터를 String으로 처리하다가 LocalTime으로 뒤늦게 바꿔 개발하여 PRO이 늦어졌습니다. 놓친 부분이나 개선해야 할 부분이 있다면 지적해주시면 감사하겠습니다.

학습 내용

- LocalTime 객체를 활용한 시간 데이터 처리와 Formatting
- SQL JOIN
- Spring Bean과 의존관계 주입(Constructor Injection, Field Injection)
- SOLID OOP Principle
- LayeredArchitecture와 관심사별 Logic의 분리
- SimpleJdbcInsert를 이용한 삽입 코드 작성
- Default Constructor의 사용



학부생으로서 어떻게 공부해야 하는가?



학부생으로서 어떻게 공부해야 하는가?



일반화

- $a \rightarrow b, c \rightarrow d$
- $P\{a, c, e, \dots\} \rightarrow S\{b, d, f, \dots\}$





일반화

- 안 봐도 비디오
- 하나를 알면 열을 안다
- 경향 추정
- 통찰력
- 편견
- ~~내 그럴 줄 알았다~~



...

어차피 1년만 지나면 **까먹을텐데** 왜 공부해야 하는가? 세세한 것들은 잊어 버리겠지만, 위에서 언급한 핵심은 아마도 여러분 상당수가 평생 사용하게 될 배경 지식이 될 것입니다.

잊은 것 같지만, **무의식 속에 남습니다.** 깊이 공부했을 수록, 고통을 많이 겪더냈을 수록 깊이 남습니다.

물론 여러분 중에 실제로 processor 아키텍트(설계자)가 될 사람은 많지 않을 것입니다.

그렇다 하더라도, CPU와 메모리가 대충 어떻게 동작한다는 것을 **이해함**으로서, 이를 바탕으로 SW 개발을 할 때 (무의식적으로라도) 좋은 판단을 하고 좋은 설계를 하길 **기대합니다**.

여러분이 짠 소프트웨어가 느리면 왜 느린지, 혹시 컴퓨터구조에 대한 생각을 안 했던 것은 아닌지, "**통찰력**"을 갖길 바랍니다.

믿으셔도 좋습니다. 이 컴공/SW 바닥에서 먹고 살아간다면 분명 도움이 될 내용입니다.

...



앎의 3단계

1. 지식

2. 지혜

3. 수리



앎의 3단계

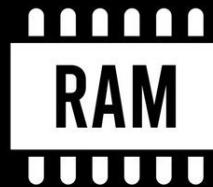
1. 지식의 누적





앎의 3단계

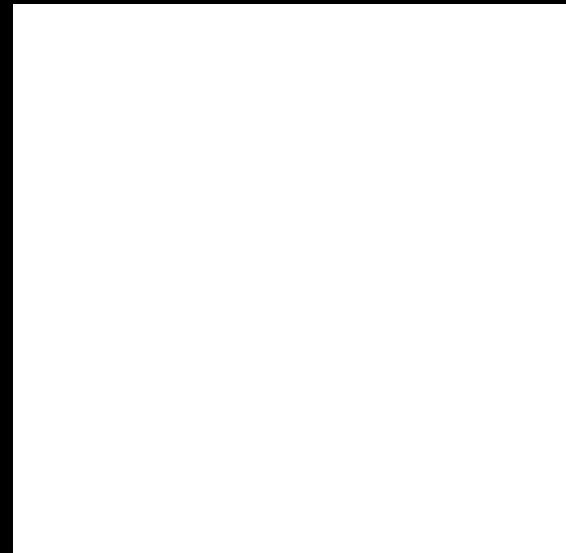
1. 지식의 누적





앎의 3단계

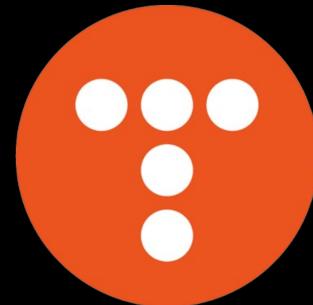
1. 지식의 누적





앎의 3단계

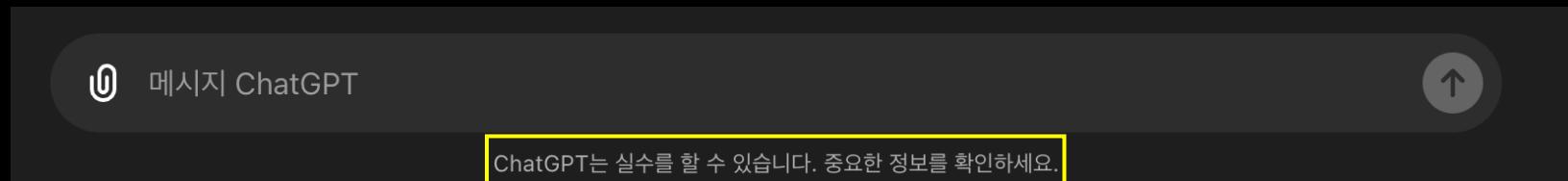
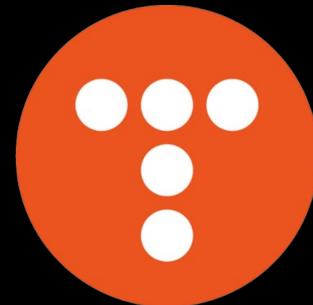
1. 지식의 누적





앎의 3단계

1. 지식의 누적





앎의 3단계

2. 지혜(일반화)



앎의 3단계

2. 지혜(일반화)

The screenshot shows a code editor with Java test code and a Git commit history.

Code Editor Content:

```
19 }
20
21 @
22     public static AddProductRequest 상품등록요청_생성() { 7 usages hong-sile
23         final String name = "상품명";
24         final int price = 1000;
25         final DiscountPolicy policy = DiscountPolicy.NONE;
26         return new AddProductRequest(name, price, policy);
27     }
28 @
29     public static GetProductResponse 상품등록_생성(final Long id) { 2 usages hong-sile
30         final String name = "상품명";
31         final int price = 1000;
32         final DiscountPolicy policy = DiscountPolicy.NONE;
33         return new GetProductResponse(id, name, price, policy);
34
35 @
36     public static UpdateProductRequest 상품수정요청_생성() { 2 usages hong-sile
37         return new UpdateProductRequest( name: "상품 수정", price: 5000, DiscountPolicy.NONE );
38
39     public static ExtractableResponse<Response> 상품조회요청(final Long id) { 3 usages hong-sile
40         return PostAuthenticated given() RequestSpecification
```

Git Commit History:

Commit	Author	Date
docs: 리드미 완성	origin & main hong-sile	2/28/24, 06:23
test: 테스트 환경 구성	hong-sile	2/28/24, 06:23
chore: data.sql 로 초기데이터를 삽입하도록 설정	hong-sile	2/28/24, 06:13
merge: clug ably clone coding	hong-sile*	2/28/24, 05:21
docs: 모자란 설명 추가	hong-sile	2/28/24, 05:20
docs: test 부분 설명 추가	hong-sile	2/28/24, 05:09
docs: rest docs 간단 설명 추가	hong-sile	2/28/24, 04:59
feat: no offset 기반 페이지 네이션 구현	hong-sile	2/28/24, 04:50
feat: Item 좋아요 수 추가	hong-sile	2/28/24, 04:27
docs: 단건 조회 API 명세 추가	hong-sile	2/28/24, 04:21



앎의 3단계

2. 지혜(일반화)





앎의 3단계

2. 지혜(일반화)





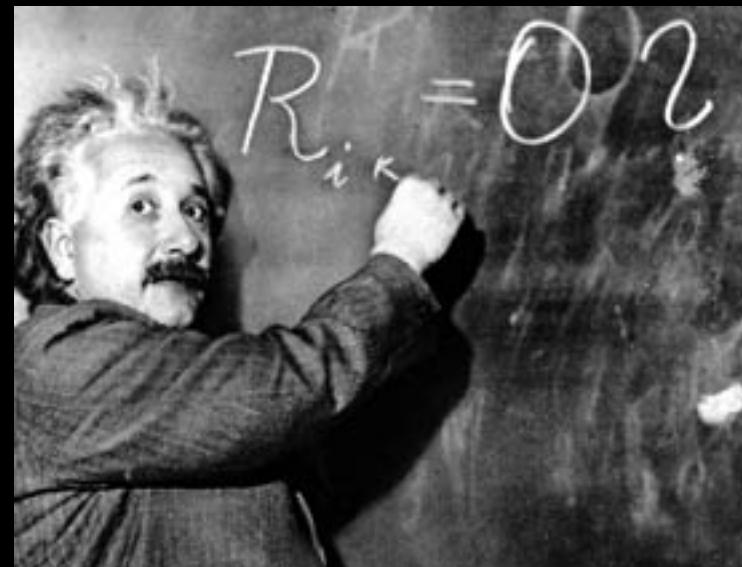
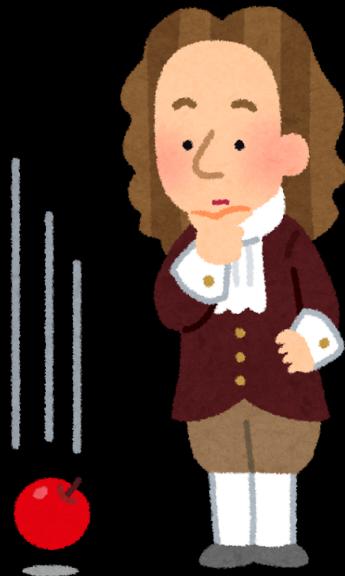
앎의 3단계

2. 지혜(일반화)



앎의 3단계

3. 수리





앎의 3단계

3. 수리

- 안 봐도 비디오
- 하나를 알면 열을 안다
- 귀납적 추론
- 통찰력
- 편견
- ~~내 그럴 줄 알았다~~



앎의 3단계

1. 지식

2. 지혜

3. 수리



학부생으로서 어떻게 공부해야 하는가?



엔지니어로서 어떻게 공부해야 하는가?



학자형(카테고리)

VS

야생형(문제)



학자형(카테고리)

- 컴퓨터 구조 (캐시, CPU, 램, ...)
- 운영체제 (스케줄링, 락, 프로세스, ...)
- 네트워크 (OSI 7, HTTP, DNS, CDN, ...)
- 알고리즘 (BF, Greedy, DP, ...)
- 데이터베이스(버퍼, 인덱스, I/O, ...)



야생형(문제)

- 인프라를 구축해야 한다.
- 서버 속도가 느리다.
- 동시성 이슈가 발생한다
- 실시간이 중요한 서비스를 구현해야 한다.
- 이미지 업로드가 느리다.



야생형(문제)

- 인프라를 구축해야 한다.
- 서버 속도가 느리다.
- 동시성 이슈가 발생한다
- 실시간이 중요한 서비스를 구현해야 한다.
- 이미지 업로드가 느리다.



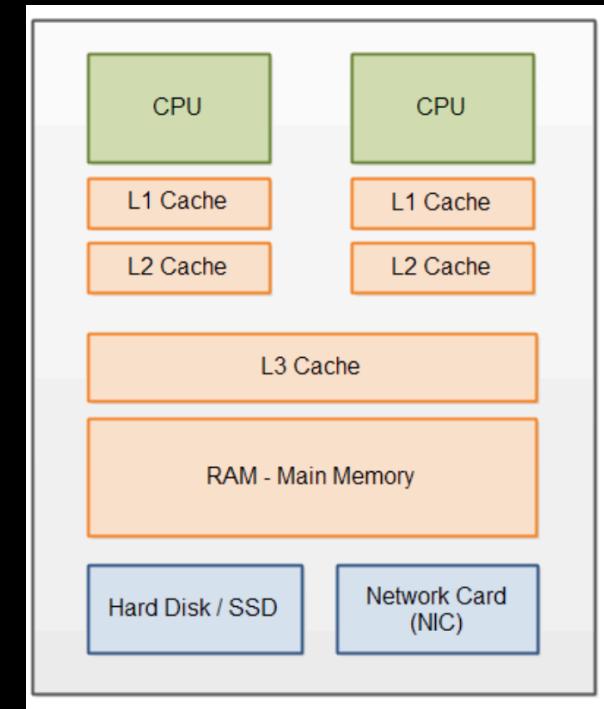
문제 해결방법 탐색 – 서버 속도가 느리다.

- 컴퓨터 구조 (**캐시**, CPU, 램, ...)
- 운영체제 (스케줄링, **락**, 프로세스, ...)
- 네트워크 (OSI 7, **HTTP**, DNS, CDN, ...)
- 알고리즘 (BF, Greedy, DP, ...)
- 데이터베이스(**버퍼**, 인덱스, I/O, ...)



캐시 – 컴퓨터 구조

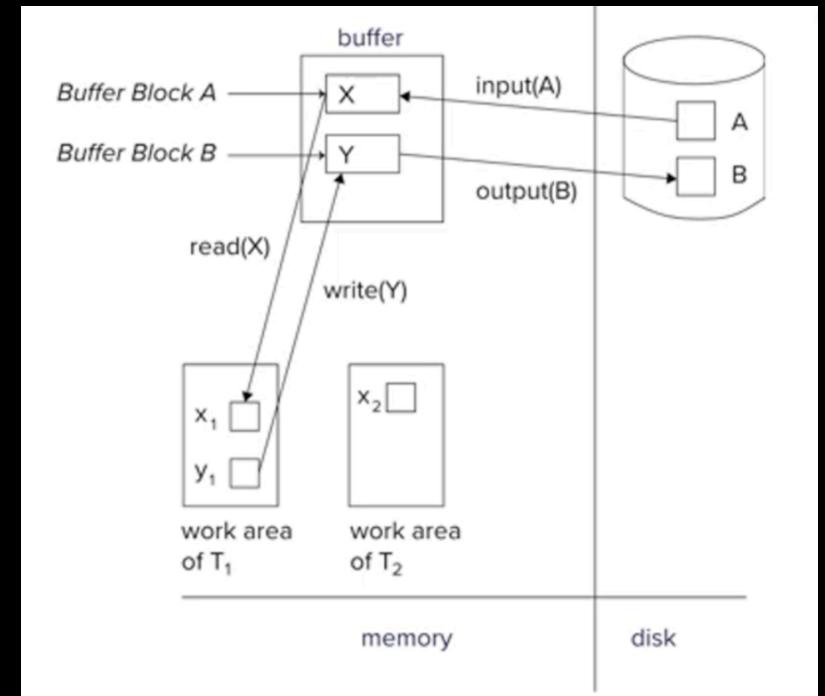
- 자주 사용하는 데이터나 값을 미리 복사해놓는 임시 저장소
- Core 별 일관성 문제를 위해 여러 개의 캐시를 둔다.





버퍼 – 데이터베이스

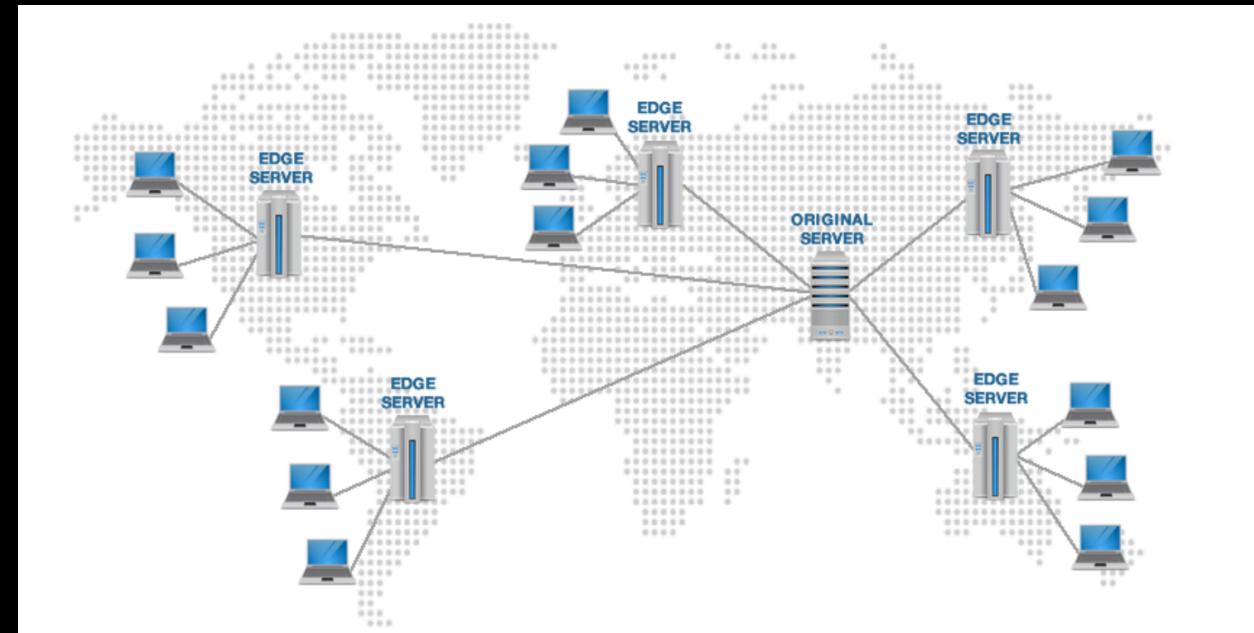
- 각 transaction 별로 데이터를 접근하는 메모리 영역을 갖는다.
- tx별 일관성 문제 고려하면서 성능을 높이기 위해 버퍼영역





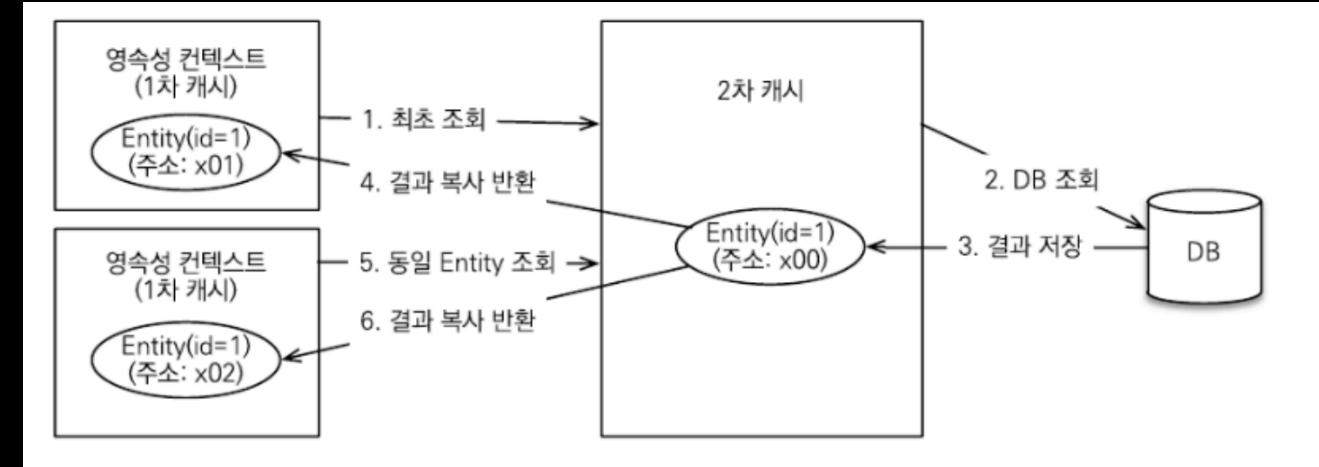
CDN – 웹

- Origin server의 데이터를 보관하고 있는 cdn의 엣지 서버
- 엣지 서버간의 데이터 일관성 문제 발생



영속성 컨텍스트 – Spring JPA

- 데이터베이스 조회를 줄이기 위해
1차 캐시 활용
- 1차 캐시의 일관성 해결하며 속도를 높이기
위해 2차 캐시 활용





데이터 I/O가 오래 걸리니, 값을 가까이에 복사해둔다.

- 컴퓨터 구조 캐시
- 데이터베이스 버퍼
- CDN
- JPA 캐시
- 브라우저의 캐시
- etc...



"저는 프론트엔드 할거라서 운영체제 같은 CS 지식은 필요 없어요."

"프레임워크 잘하는게 CS 지식 아는거보다 훨씬 낫다고 생각해요."



"저는 프론트엔드 할거라서 운영체제 같은 CS 지식은 필요 없어요."

"프레임워크 잘하는게 CS 지식 아는거보다 훨씬 낫다고 생각해요."



학점이 중요하지 않다?



학점이 중요하지 않다?

- 학점 잘 안주지만 좋은 강의들(백정엽 교수님, 김대원 교수님, ...)
- 지금은 안 중요하다. 하지만 n년 뒤에는..?



마지막으로

- CS는 열심히
- 단순히 그런 게 있구나 하고 넘어가지 않기
- 최대한 기존 지식과 엮어보려 노력하자

아이콘 출처



- Hdd 아이콘 제작자: Oz Mora - Flaticon
- Brain icons created by Freepik - Flaticon
- Note icons created by Freepik - Flaticon
- 알츠하이머 아이콘 제작자: Vector Squad - Flaticon
- 기분 전환 아이콘 제작자: bsd - Flaticon