NEXT 학생들과 함께한 Ruby & RoR 스터디

2013년 11월 즈음 한 학생이 Ruby On Rails 스터디를 하자는 제안을 해왔다. 나는 NEXT에서 학생들과 함께한 자바 스터디 글에서 다루었듯이 이미 하나의 스터디를 하고 있었다. 또 하나의 스터디를 할 여력이 되지 않을 듯해서 약간 망설이다 "왜 스터디를 하려고 하느냐?"고 물어봤다. 그랬더니 돌아오는 답변이 나의 마음을 울렸다.

> NEXT 동기 중의 한명이 지금 슬럼프에 빠져 헤어나오지 못하고 있어요. 학교도 거의 오지 않고 있어 어떻게 구제할까 고민하다가 스터디를 하자고 제안했는데 하겠다고 하더라고요. 1학기 소프트웨어 공학 수업(2013학년 1학기 소프트웨어 공학 수업을 마치면서...)을 할 때 즐거웠던 경험이 있어서 교수님이 같이 하시면 적극적으로 참여할거 같은데...

나는 이 말을 들고 도저히 거절할 수가 없었다. 단, 내가 여력이 없기 때문에 완전히 새로운 주제로 진행하는데는 한계가 있겠다는 생각이 들어 NEXT에서 학생들과 함께한 자바 스터디 에서 진행하고 있는 지뢰찾기 게임을 Ruby로 만들어 보자고 제안했다. 동기의 슬럼프를 극복해 주고 싶은 한 친구의 순수한 우정이 또 하나의 스터디를 진행하게 만들었다. 그 스터디가 이번 주 수요일 끝났다. 더 하고 싶은 마음은 있지만 새로운 학기를 진행하는 상황에서 도저히 내가 여력이 되지 않아 끝내는 것으로 했다. 추후 여력이 되면 다시 시작하는 것으로 아쉬움을 달랬다. 몇 달 동안 달려왔던 스터디 과정을 공유하려 한다.

스터디 진행 방식

스터디 진행 방식은 NEXT에서 학생들과 함께한 자바 스터디와 같았다. 1차 Ruby 스터디 전하고 싶었던 내용은 객체 지향 설계, 단위 테스트의 필요성과 테스트 주도 개발 방식, 리팩토링을 깊이 있게 경험할 수 있도록 하는 것이었다. 학생들이 만들고 싶은 소프트웨어를 하나 정한 후 단계적으로 만들어가면서 위 내용을 소화하는 것을 목표로 했다.

스터디 진행은 다음과 같이 진행했다.

- 각 스터디 주차별로 요구사항 목록을 정한 후 학생 개개인이 해당 기능을 구현해 온다.
- 스터디 시간에는 해당 요구사항을 짝 프로그래밍으로 구현하면서 이슈 사항에 대해 토론하는 방식으로 진행했다.
- 다음 시간까지 구현할 요구사항을 정리한다.
- 스터디에 대한 회고를 진행한다.

매주 위와 같은 방식으로 진행했다. 1차 Ruby 스터디를 끝낸 후 Ruby On Rails(이하 RoR) 스터디를 다시 시작했다. RoR 스터디는 서비스 기획을 한 후 이 서비스를 단계적으로 완성하는 방식으로 진행했다.

Ruby 주차별 스터디 진행

- 루비-1주차-요구사항분석 : 1주차에는 구현할 소프트웨어를 정했다. 난이도나 규모로 봤을 때 지뢰찾기를 선택했다. 지뢰찾기 게임에 대한 요구사항을 분석했다. 요구사항 분석 과정에서 단계별로 어떻게 접근할 것인지에 대한 토론 과정이 있었다. 최초 5 X 5부터 시작할 것인지 여러 가지 방식으로 접근하다 1 X 1부터 2 X 2로 단계적으로 확장해 가는 방법으로 결론을 내고 요구사항을 정리했다.
- 루비-2주차 - 1 by 1 구현 : 개발 환경 정리하고, TDD에 대한 감을 찾아가면서 1 by 1에 대한 기능 구현했다. 이 주차의 가장 중요한 배움은 Ruby 기본 문법과 RubyMine이라는 IDE 사용 방법에 대한 학습이 가장 컸다. 아무래도 새로운 언어를 학습하는 측면에서 초반에는 개발 언어 문법과 개발 도구에 대한 이해가 중요하다.
- 루비-3주차 - 게임 엔진 기본 구현 완료 : 이름과 메소드명을 리팩토링하는 가장 낮은 수준의 리팩토링을 경험했다. 게임의 핵심 기능을 구현하고 대부분의 기능을 구현완료했다. 이 시점부터 서서히 리팩토링에 대한 관심도가 생기기 시작했다. 이 주차에 남긴 회고 내용을 보면 다음과 같다.

> - 구현과 리팩토링을 분리해서 생각하는 것이 훨씬 더 명확하고 효율적이었다. 한번에 하나에 집중하는 연습이 된다.
> - 리팩토링하는 시점에 리팩토링만 집중하다보니 더 나은 해결책을 찾는 것에 명확해짐. 즉, 일단 구현해서 마음의 평정심을 가진 후 여유를 가지고 리팩토링을 하니 사고가 훨씬 더 열리게 되는 느낌이 들었다.
> - 위 코드를 리팩토링하면서 수업 시간에 배웠던 code block의 개념을 구체적으로 이해하는 것을 느꼈으며, 자바스크립트에서 배웠던 closure 개념을 위 예를 통해 학생들이 명확하게 이해하는 느낌이 들었다.
> - 여러 명이 같이 개발하다보니 이름을 정하는 것에 있어 좋은 이름을 찾을 수 있었다.
> - 활발한 논의를 하면서 구현하다보니 집중에서 구현하는 것을 경험했다.

- 루비-4주차 - 리팩토링 : 핵심 기능만 구현한 상태에서 기능을 추가하지 않고 이 상태에서 리팩토링 작업만 진행했다. 학생들이 리팩토링을 진행하려고 하는 시점에 한 가지 제약사항을 두었다. 기존에 개발한 테스트 코드가 깨지지 않는 상태에서 소스 코드를 리팩토링해야 한다는 것이었다. 이 제약사항은 NEXT에서 학생들과 함께한 자바 스터디 에서도 시도해 큰 배움을 얻었던 것이었다. 이 과정은 정말 step by step으로 소스 코드를 개선하면서 리팩토링을 진행했다. 추후 회고에서 이 스터디가 학생들에게 큰 영감을 주었던 것으로 생각한다. 이 주차에 정리한 학생의 회고는 다음과 같다.

> 학생 A
>
> 객체를 뽑아내는 고민을 하면서 이제 점점 객체지향적 사고방식을 점점 알아가는 것 같습니다. 정말로 신이 존재한다면 그도 이런 사고로 세상을 창조하지 않았을까요? 하지만 아직 수련이 부족하여 객체지향적인 사고에 자신감은 없네요.
>
> 한편, 리팩토링을 할때 이전에 만들어 둔 테스트가 버텨주고 있으니 정말 든든하고 편했습니다. 테스트의 그린라이트를 볼때는 정말 뿌듯하고 진짜 공부를 한듯한 느낌을 받네요.

학생 B

- 지금 당장 눈앞에 있는 코드에서 벗어나 더욱 더 멀리, 넓게 볼 수 있는 프로그래밍적 관점을 배운 시간이였다.
- 특히 injection과 객체화에 대한 토론에서 많은 insight를 얻을 수 있었던것 같다.

  injection을 논의할 때에는 지금까지 test코드를 바라보던 관점들인 '서비스코드와 실시간으로 상호작용을 할 수 있다, 에러처리가 빠르다. 소스코드 수정 및 리팩토링에 자신감을 줄 수 있다'는 것에서 발전시켜 코드설계나 객체지향의 시점을 알려주는 선각자가 된다는 것을 배웠다. (수업중에 논의되었던 '테스트코드가 복잡한건 실서비스 항목에서 리팩토링이 필요한 것이다'나, '테스트코드에서 중복되는건 객체화를 고민해볼 시기라는 것') 등과 같은 항목들이 인상깊었다.

- 현재 학교수업을 통해 여러 프로젝트를 진행하고 있지만, 루비스터디가 가장 즐겁다. 테스트코드와 todo list가 함께하기 때문인것 같다. 보통 다른 프로젝트의 경우 한번 시작하면 빠져들어 재밌게 코딩하지만, 한번 시작하기가 어렵다. 하지만 루비 스터디는 앞서 이야기 한것과 같이 테스트코드와 todo list덕분인지 코딩의 시작과 끝이 부담되지 않는것 같다.

- 루비-5주차 - 리팩토링 및 회고 : 4주차에 이어 리팩토링을 진행했다. 4주차의 리팩토링에서 한발 더 나아가 새로운 객체를 추출하고 새롭게 추출한 객체에 역할을 위임하는 과정까지 경험했다.
- 루비-6주차 - 랜덤구현 및 회고 : 게임 구현의 마지막 작업으로 각자 지뢰찾기 게임을 구현한 후 소스 코드 리뷰하는 방식으로 진행했다.
- 루비-7주차 - 테스트 자동화 : 6주차 소스 코드 리뷰 후 소스 코드 개선과 Ruby에서 테스트 자동화에 대해 논의하는 시간을 가졌다. RSpec, guard, spork를 활용해 테스트를 어떻게 자동화하는지에 대한 학생 중의 한명이 준비해 온 후 공유하는 시간을 가졌다. 이 스터디를 마지막으로 Ruby 스터디를 끝내고 Ruby on Rails 스터디를 다시 시작하는 것으로 결정했다.

스터디 중간 회고

Ruby 스터디가 끝났다. Ruby 스터디 시작은 나 + 학생 2명으로 시작했다. 스터디 진행 중에 같이 합류하고 싶다는 친구가 있어 나 + 학생 3명으로 진행하고 있었다. 모두 남자들만 모인 칙칙한 스터디였다. 그런데 RoR 스터디를 진행하는 시점에 여학생 한명이 합류하면서 스터디의 꽃이 되었다. 여학생이 합류하면서 스터디의 상당히 부드러워지면서 토론도 활발해진 느낌이 들었다. 여자의 힘을 다시 한번 느낄 수 있는 계기가 되었다.

스터디를 진행하는 중 NEXT 2기 신입생이 입학하게 되었다. 그런데 신입생 오리엔테이션 모임에서 이 스터디에 대한 소식을 들었다면서 같이 합류하고 싶다는 의사를 밝혀 또 다시 한명의 스터디원이 늘어났다. 3명으로 시작했던 스터디가 끝나는 시점에 6명이 되어 있었다.

같은 인원으로 스터디를 진행하는 것이 좋은 점도 있지만 일정 시점이 지나면 자극이 사라진다는 단점이 있다. 그런데 이 스터디의 경우 중간 중간에 새로운 팀원이 계속해서 합류하면서 스터디의 새로운 자극이 되었다. 학생들 서로 서로가 자극을 주면서 스터디를 활성화하는 계기가 되었다. 이 스터디의 경우 새로운 팀원이 합류될 때마다 그런 느낌을 강하게 받을 수 있었다.

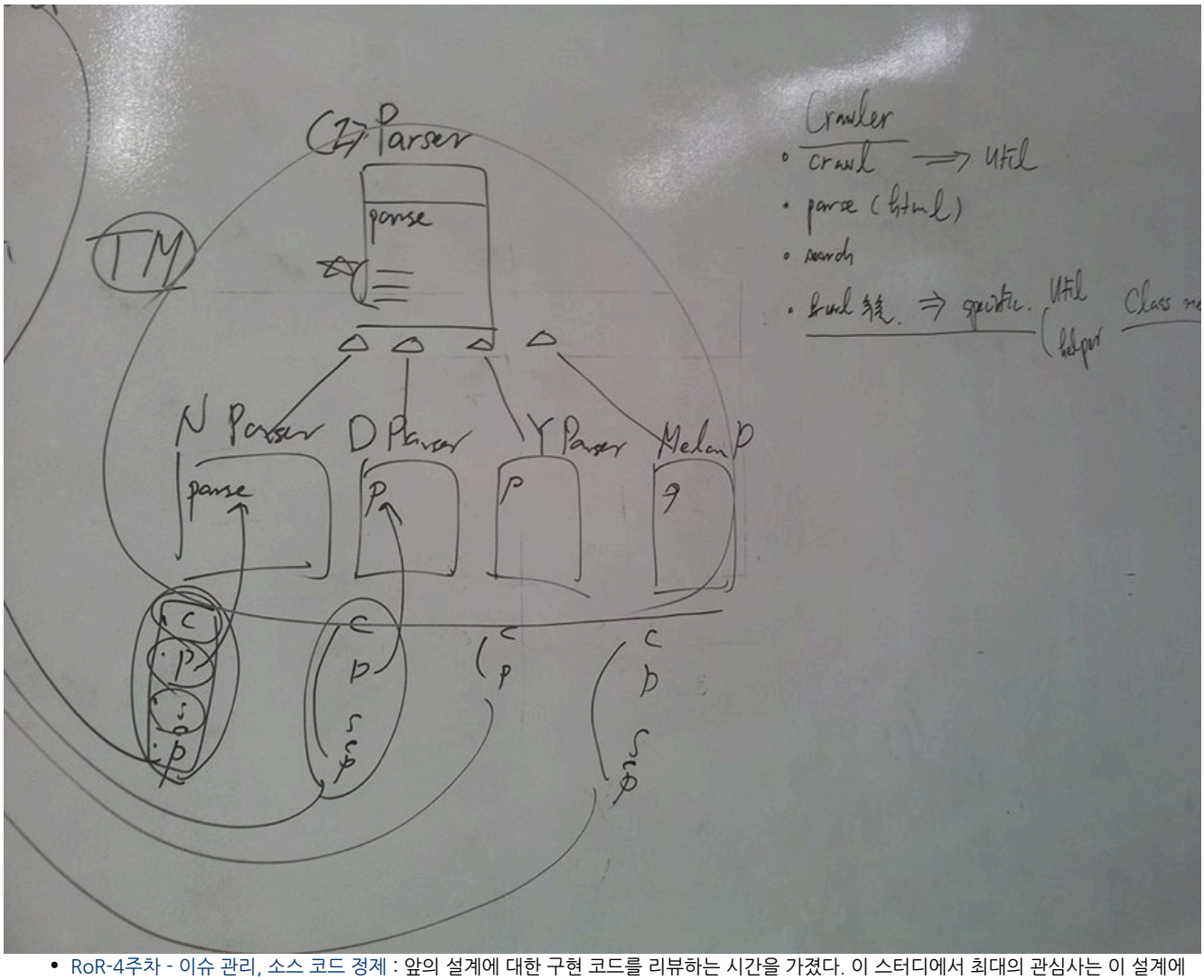
Ruby on Rails 주차별 스터디 진행

RoR 스터디부터 스터디 내용을 분석하고 기록하고 싶은 마음이 들어 스터디 내용을 동영상으로 기록하고 정리하는 과정을 거쳤다. 스터디 내용을 기록하고 정리하는 작업은 학생 중의 한명이 담당했다. NEXT는 근로를 제공하고 그에 따르 장학금을 받는 제도가 있다. 한 친구에게 근로도 하면서 스터디까지 같이 진행하면서 배움을 만들어갈 수 있도록 했다. 이 때부터 스터디 내용이 좀 더 구체적으로 정리되어 있다.

- RoR-1주차 - 서비스 기획 및 준비 : RoR 기반으로 만들 서비스를 기획했다. 서비스 기획을 하면서 스터디에 참여하는 목적과 서로의 차이점을 이해할 수 있는 계기가 되었다. 단, 그 과정 속에서 서로를 이해하고 다름을 알 수 있는 많은 토론이 있었던 것이 인상적이었다. 서로 의식하지 않고, 배려하면서 토론하는 모습을 볼 수 있었다. 학생간, 학생 vs 교수 간의 친밀도를 높이기 위해 닉네임을 만들고 부르는 것으로 약속했지만 일주일에 한번 만나는 상황이라 끝내 성공하지 못했다.
- RoR-2주차 - 환경설정 및 추천음악 Parsing : 각자 구현한 소스 코드를 리뷰하는 방식으로 진행했다. RoR 스터디는 스터디에서 직접 구현하는데 많은 시간을 투자하기 보다는 각자 구현한 소스코드를 리뷰하고 설계를 논의하는 시간을 많이 가졌다. 수업을 관찰하는 학생이 다음과 같이 관찰 결과를 남겼다. 이 주차부터 지금까지 소프트웨어를 학습하면서 어떻게 학습했는지, 자신만의 노하우를 공유하는 시간을 가졌다. 이 과정 또한 서로를 알아가는데 많은 도움이 되었다.

  - 1주차에는 각자 갖고 있는 경험의 정도가 너무 달라서 잘 어우러져서 스터디를 할 수 있을까 걱정했지만, 생각보다 스터디가 잘 진행된 것 같다.
  - '왜 사용하나'를 아는 것이 단순히 '~한 용도로 쓰인다.'로 배우는 것 보다 좋은 것 같다.
    - RoR에서 guard 등을 왜 사용하나에 각목이 답함으로써 active record pattern을 알게 되고, 자바진영 패턴과의 비교를 자연스럽게 이해할 수 있었다. 각각 따로 배운다면 그 유기적인 관계를 이해하기 어려웠을 것 같다.
  - 라이브러리 사용에 대해서도 차이점이 나타나는 것 같다.
    - 위에서 말한 레일즈의 숙련도에 따라서 루비에 대한 관심이 나타난다.
    - 주디(덜 숙련된 사람)의 경우
      - 아직 레일즈 자체에 관심이 많았기 때문에 우선 서비스가 돌아가는 것을 첫번째로 봤다.
      - 라이브러리를 사용해서 데이터 파싱에 따른 뷰를 빨리 보여주고 싶었다.
      - 데이터 파싱 기능을 하는 클래스를 만들 수 있겠다는 생각을 하긴 했었지만 우선 순위에서 뒤로 밀려있었다.
    - 각목(숙련된 사람)의 경우
      - 루비를 최대한 활용한 방향으로 코딩을 하였다.
    - 라이브러리에 대한 개인적인 생각이 다르기 때문에 라이브러리 사용에 차이점이 나타났을 수도 있을 것 같다.
      - Q. 라이브러리에 대한 생각은 숙련도에 따라 어떻게 달라질까?

- RoR-3주차 - 추천음악 Parsing, 협업 환경 공유 : 지금까지 구현한 소스 코드의 중복을 제거하기 위해 구조를 어떻게 개선하는 것이 좋을 것인지에 대해 토론하고 설계안을 만드는 과정을 거쳤다. 학생들이 설계에 대한 아이디어를 내고 내가 이를 그림으로 그려 나가는 과정을 가졌다. 이 과정에서 설계한 내용을 바탕으로 다음 스터디 시간까지 구현해 오는 것이 과제로 제출되었다.



- RoR-4주차 - 이슈 관리, 소스 코드 정제 : 앞의 설계에 대한 구현 코드를 리뷰하는 시간을 가졌다. 이 스터디에서 최대의 관심사는 이 설계에 대한 구현 리뷰도 있었지만 지난 주부터 새롭게 합류한 친구의 학습 과정을 듣는 시간이었다. 이 친구가 학습 과정을 공유하면서 다른 친구들에게도 많은 자극이 되었다. 특히 vi를 최고의 에디터로 활용하며, 손코딩을 즐겨하고, 문법과 API를 모두 외운다. 프로그래밍을 시작한지 1년도 되지 않는데 밖으로 보여지는 실력은 장난아니라는 느낌이 들었다. 기대되는 친구가 2기로 들어왔다. 내가 감당할 수 있을지 모르겠다.
- RoR-5주차 - 테스트 및 마지막 스터디 : 정말 긴 시간을 지나 마지막 스터디까지 왔다. 이 날은 마지막 스터디인 만큼 지식을 전달하는데 집중하지 않았다. 단지 학생 발표 중 이슈 사항들에게 대해 꼬리에 꼬리를 무는 방식으로 질문을 지속했다. 그런데 이런 식의 학습이 상당히 흥미롭고 좋았다는 피드백이 있었다. 추후 수업 설계할 때 참고하면 좋겠다. 또 하나 흥미로웠던 활동은 교수들의 공개 뒷담화... 안 당한 사람은 모릅니다 에서 영감을 얻어 우리도 서로 간의 뒷담화를 해보기로 했다. 예상외로 반응은 폭발적이었다. 서로를 알아가고 신뢰를 쌓는 좋은 시간이었다. 마지막 스터디인 만큼 최종 회고를 하고 스터디를 마쳤다.

My 회고

이 스터디와 NEXT에서 학생들과 함께한 자바 스터디에서 다룬 자바 스터디 두 개를 진행하면서 경험한 나만의 느낌을 정리해 본다.

- 지금까지 습관적으로 작업하던 방식에 약간의 제약을 가하고, 이 제약을 해결하기 위한 방법을 찾는 과정에서 교수, 학생 모두 상당히 큰 희열감을 맛보는 경험을 했다.
- 교수, 학생 간의 즐겁게 웃고, 떠드는 다소 시끄러운 상황에서도 배움이 일어난다는 것을 느꼈다. 약간의 소음은 오히려 학생들을 소통하게 하고 토론하게 만드는 것을 경험했다. 조용하던 친구들도 자신의 목소리를 내는 경험을 종종 했다.
- 나는 각 스터디별로 과제와 목표가 명확하다고 생각했는데 학생들이 느끼는 온도 차이는 컸다. 약간 더디더라도 목표를 명확히 하고 거북이 걸음으로 접근하는 것에서 더 큰 배움을 얻는 경험을 했다. 많은 것을 전달하려고 욕심 부리는 학생들의 만족도는 낮아졌다.
- 소수의 인원으로 친밀감 있게 소통하면서 지식만을 습득하는 것이 아니라 지식 외적인 부분(자기 주도적 공부 습관, 토론하는 방법, 협업하는 방법, 코드 리뷰 등)에 대한 다양한 부분을 배운다는 것을 경험했다.
- 교수에게서 배우는 것도 많지만 학생 서로 간에 질문과 답변을 주고 받으면서 더 많은 것을 배운다는 것을 경험했다. 교수는 단지 학생들이 학습할 수 있는 환경을 잘 만들어 주고, 가끔 제대로 가고 있는지 확인, 새로운 개념에 대한 정리, 토론하는 내용에 대해 정리하는 역할만 해도 학생들은 자연스럽게 배워나간다는 것을 느꼈다.
- 평가가 없어지는 순간 학생들이 배움에 집중한다는 것을 경험했다. 평가가 없는 자발적 참여에 의해(즉, 동기 부여가 있는 상태에서 참여) 학습이 이루어질 경우 배움의 본질에 집중하는 모습을 봤다. 외부 환경에 영향을 덜 받음으로써 좀 더 집중력 있는 배움이 발생했다고 생각한다.

학생 회고

희열감을 맛보게 하기도 하고, 실망을 안겨주기도 했던 친구들. 그들이 스터디에서 느낀 점을 정리해 본다. 전체적으로 즐거움이 더 가득했던

스터디였다.

학생들의 목소리를 생생하게 들을 수 있는 회고 내용도 들어보면 재미있다.

스터디를 하면서 가장 큰 배움을 얻은 시점은 언제였으며, 어떤 배움을 얻었는가?

- 학생 A
  - Step by Step으로 리팩토링하는 과정에서 테스트의 묘미를 느꼈다.
  - 마지막 스터디의 뒷담화
- 학생 B
  - 단일 책임 원칙, 추상화 개념을 코드와 같이 논의할 때
  - 다른 사람들과 전체를 보면서, 어떤 방법이 효율적일지 생각해 보는 것
  - 나의 생각을 개선하는 방법 + 추상화에 대한 지식
  - 낯선 개념들에 대해 전체적인 흐름 파악
- 학생 C
  - 코드 리뷰에 대한 즐거움, 협업에 대한 즐거움을 느꼈다.
  - 무엇보다 프로그래밍을 주제로 스터디를 하는 즐거움을 느꼈다.
- 학생 D
  - 짝 프로그래밍
  - 여유있는 개발 시간 -> 개발 역량이 높아졌다는 개인적인 느낌
  - 교수님을 본받아 기록의 필요성을 느꼈다. 현재 기록하고 있다.
- 학생 E
  - 나는 이번 스터디에서 기능 구현보다 TDD 프로세스에 집중했다. 화폐 예제로 가장 큰 배움이 있었던 순간은 "테스트 주도 개발" 책을 읽고, 프로세스를 경험하고 그 과정을 실제로 스터디에 적용해 보는 순간이었다.

더 좋은 스터디를 하기 위해 개선해야 할 부분은 무엇인가?

- 학생 A
  - 각자의 흥미에 맞는 목표
  - 교수님이 질문을 정리해 주는 부분이 좋았다. 나머지는 그 질문에만 집중하게 하는 방식
- 학생 B
  - 준비를 많이 해오지 않은 것, 미리 언어에 익숙해 지는 것
  - 서로 어디까지 진행하는지 스터디 외 시간에도 공유하는 것이 필요하다고 생각
- 학생 C
  - 관심을 조금 더 끌 수 있는 요소가 필요하다.
  - 스터디 pair에 대한 선의의 경쟁심 유발
- 학생 D
  - 매주 스터디에 대한 강제성을 높였으면 한다.
  - 짝 프로그래밍의 비율을 높이자.
  - 주제(TDD 등)를 더 명확히 잡으면 좋겠다.
- 학생 E
  - Rails 스터디를 하면서 테스트보다 Rails 동작 원리에 집중했던 것 같다.
  - 단순한 기능 구현이 아니라 구성원 모두 테스트를 통한 구현, TDD에 의해 만들어지는 아키텍처에 대해 토론하는 과정이 있었으면 좋겠다.
  - 활발한 토론을 위해서는 명확한 목적 & 구성원들의 참여가 필요하다.

NEXT 수업을 우리가 진행한 Ruby & RoR 스터디와 같이 진행할 경우 적합한 부분과 우려되는 점은 무엇인가?

- 학생 A
  - 적합 : 학생들 1:1 케어가 더 잘된 것 같다.
  - 우려 : 평가라는 요소가 들어가게 될 경우 이런 분위기가 형성이 안될 것 같다. 자신이 모르는 것을 모른다고 하기 어려울 듯하다.
- 학생 B
  - 팀 플레이에서 얻을 수 있는 상대방의 생각. 서로 지나쳤던 부분
  - 체계적인(주도적으로) 학생들이 진행해 나갈 수 있을까? 놓치는 부분이 많을 것 같다.
- 학생 C
  - 시간이 너무 많기 때문에 코드 리뷰 & 수업 준비를 각자에게 자율적으로 맡기는 점이 우려된다.
  - 정말 Needs Base로 수업이 개설되어야 하지 않을까?
- 학생 D
  - 적합 : 열심히만 한다면 개발 역량을 높일 수 있다. 빠른 학습 속도
  - 우려 : 분명히 따라가지 못하는 학생이 생긴다. 낙오되는 학생을 어떻게 해결할 것인가?
- 학생 E
  - 앞의 내용과 중복

NEXT 수업과 Ruby & RoR 스터디에서 차이점이 있다면 무엇일까?

- 학생 A
  - 자발적 참여
  - 무평가
  - 교수님의 적극적 참여
- 학생 B
  - 지식을 얻는 방향

        - NEXT 수업은 교수 -> 학생(나)
        - RoR 스터디는 학생들 -> 나
    - 다른 사람에 의해서 새로운 것을 알 때보다 스스로 모르는 것을 알아갈 때.
    - 주도적으로 더 깊이 있는 지식까지 찾아가는 것 같다.
- 학생 C
    - 즐거움, 유대감, 지속성
- 학생 D
    - 없음
- 학생 E
    - 질문을 잘못 이해함.
    - Ruby 스터디는 스터디에서 해야할 일이 명확했던 것 같고, Rails의 경우에는 조금씩 차이가 있었던 것 같다.
    - Ruby는 최신 코드를 이해하고 다 같이 만들었지만 Rails는 나만의 개인 프로젝트였던 느낌이 난다.

Ruby & RoR 스터디에 대한 전체 회고를 한다면...

- 학생 A
    - Ruby 스터디가 더 좋았다. Ruby 스터디는 코딩에도 적극적으로 참여한 점이 좋았다.
- 학생 B
    - 짧은 기간이어서 아쉬웠고, 처음 다짐했던 것보다 스스로 준비가 부족해서 더 아쉬웠다.
    - 각각 다른 사람들이 모였기 때문에 그 사람들을 경험해 보는 것도 좋았다.
- 학생 C
    - 정말 많은 부분을 배웠고, 항상 친근하게 이끌어 주어서 고맙다.
    - 프로그래밍적인 부분 뿐만 아니라 개인의 경험이나 학습 방법 등 포괄적인 내용을 공유했던 것도 많이 기억에 남네요.
    - 소중한 인연 계속 이어갔으면 좋겠습니다.
- 학생 D
    - 짧은 시간이었지만 너무나 많은 점을 배웠다. 더 열심히 하겠다.
- 학생 E
    - 더 열심히 하지 못했던 것이 아쉽다.
    - 웹 애플리케이션에서 OOP의 맛을 제대로 경험해 보지 못해서 아쉽다.