평생을 학습하며 성장하기 위한 가장 중요한 것은???

소프트웨어 장인이 되겠다는 마음가짐이지 않을까?

> 소프트웨어 장인정신은 스스로가 선택한 직업에 책임감을 가지고, 지속적으로 새로운 도구와 기술을 익히며 발전하겠다는 마음가짐이다.
>
> 소프트웨어 장인정신은 책임감, 프로페셔널리즘, 실용주의, 소프트웨어 개발자로서의 자부심을 의미한다.
>
> 소프트웨어 장인 책 중에서 발췌

소프트웨어 장인: 프로페셔널리즘.실용주의.자부심

- 개발자로서 어떤 마음가짐으로 살아갈 것인지에 대한 가이드를 제시하는 책이다.
- 프로그래머라는 우리 업에 대한 자부심을 느끼고, 전문가로서의 책임과 역할에 대해 자세하게 다루고 있다.

소프트웨어 장인정신 매니페스토

- 동작하는 소프트웨어 뿐만 아니라, **정교하고 솜씨 있게 만들어진 작품을**,
- 변화에 대응하는 것뿐만 아니라, **계속해서 가치를 더하는 것을**,
- 개별적으로 협력하는 것뿐만 아니라, **프로페셔널 커뮤니티를 조성하는 것을**,
- 고객과 협업하는 것뿐만 아니라, **생산적인 동반자 관계를**,

이 왼쪽의 항목들을 추구하는 과정에서, 오른쪽 항목들이 꼭 필요함을 의미한다.

의식적인 연습

- 프리코스 3주 동안 진행한 방식은 의식적인 연습을 통해 효과적으로 학습하는 방식으로 설계해 진행했다.
- 무조건 연습을 많이 한다고 실력이 향상되지 않는다.
- 점진적으로 난이도를 높혀가고, 피드백으로 받으면서 의식적으로 연습할 때 빠르게 성장할 수 있다.

1만 시간의 재발견

- 의식적인 연습이 무엇이며, 의식적인 연습이 가능하도록 설계하는 방법에 대해 다룬다.

의식적인 연습의 7가지 원칙

- 첫째, 효과적인 훈련 기법이 수립되어 있는 기술 연마
- 둘째, 개인의 컴포트 존을 벗어난 지점에서 진행, 자신의 현재 능력을 살짝 넘어가는 작업을 지속적으로 시도
- 셋째, 명확하고 구체적인 목표를 가지고 진행
- 넷째, 신중하고 계획적이다. 즉, 개인이 온전히 집중하고 '의식적'으로 행동할 것을 요구
- 다섯째, 피드백과 피드백에 따른 행동 변경을 수반
- 여섯째, 효과적인 심적 표상을 만들어내는 한편으로 심적 표상에 의존
- 일곱째, 기존에 습득한 기술의 특정 부분을 집중적으로 개선함으로써 발전시키고, 수정하는 과정을 수반

의식적인 연습을 위한 도전꺼리 찾기

- 객체지향 생활 체조 원칙 지키기
- 클린코드 책에서 제시하는 원칙 지키기
    - 예를 들어 함수 인자 수와 관련해 다음과 같은 원칙으로 구현
    - 함수에서 이상적인 인수 개수는 0개(무항)이다. 다음은 1개이고, 다음은 2개이다.
    - 3개는 가능한 피하는 편이 좋다.
    - 4개 이상은 특별한 이유가 있어도 사용하면 안된다.
- **초보자일 때 가능하면 정성적인 원칙보다 정량적인 원칙으로 연습한다.**

함께 자라기 - 김창준

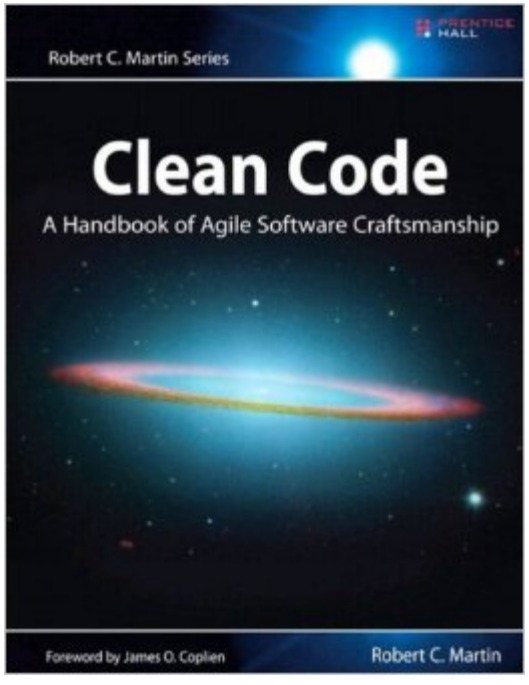
- 내가 성장하는 방법, 우리가 함께 성장하는 방법, 매일매일 성장하는 방법에 대해 다루고 있다.
- 나와 팀이 같이 성장하는 방법을 알고 싶다면 반드시 읽어볼 것을 추천한다.

클린코드

- 프리코스 과정에서 느꼈지만 현장에서는 읽기 좋은 코드, 유지보수하기 좋은 코드를 구현하는 역량이 중요하다.
- 클린코드를 구현하는 연습은 개발자로 살아가는 평생해야 한다.
- 클린코드를 구현하는 좋은 연습은 단위 테스트를 기반으로 지속적인 리팩토링을 통해 향상할 수 있다.

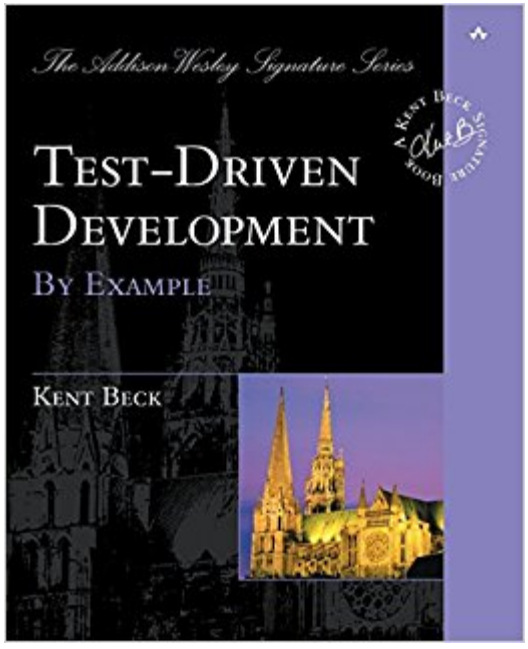
Clean Code - Robert C. Martin

- 클린 코드를 구현하기 위한 다양한 규칙들을 설명하고 있다. 책 예제 코드가 자바 기반으로 구현되어 있으며, 다양한 예제를 통해 설명하고 있다.
- 클린 코드에 관심이 있는 개발자라면 반드시 읽어야 할 책이다.
- `Clean Code(인사이트)` 라는 이름으로 번역서가 있음.



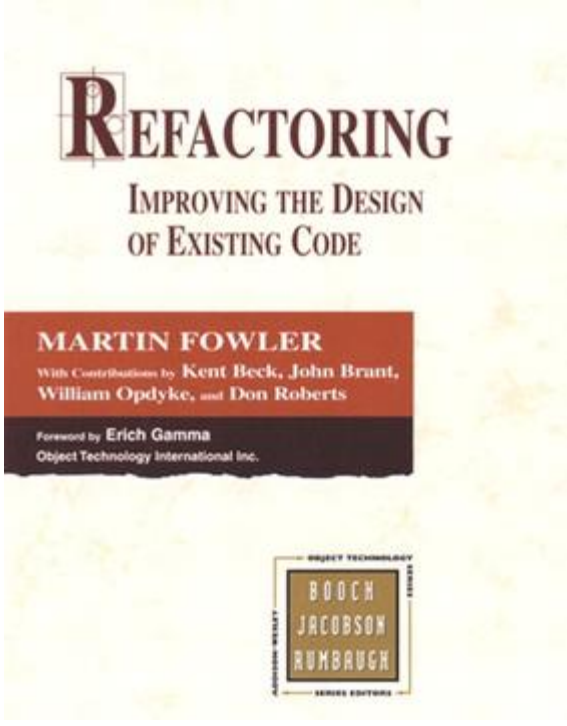
Test Driven Development by example - Kent Beck

- TDD의 구체적 사례와 패턴을 제시하고 있다.
- TDD를 제대로 연습하고 경험하고 싶다면 반드시 읽고 실습해 봐야할 책
- `테스트 주도 개발(인사이트)` 이라는 이름으로 번역서가 있음.



Refactoring - Martin Fowler

- 리팩토링 개념, 리팩토링이 필요한 이유와 다양한 리팩토링 카탈로그를 제공하고 있다.
- 책의 카탈로그가 나오기 전까지가 핵심적인 내용이다. 카탈로그는 필요한 시점에 참고하는 용도로 사용한다.
- `리팩토링(한빛미디어)` 이라는 이름으로 번역서가 있음.



객체지향 설계

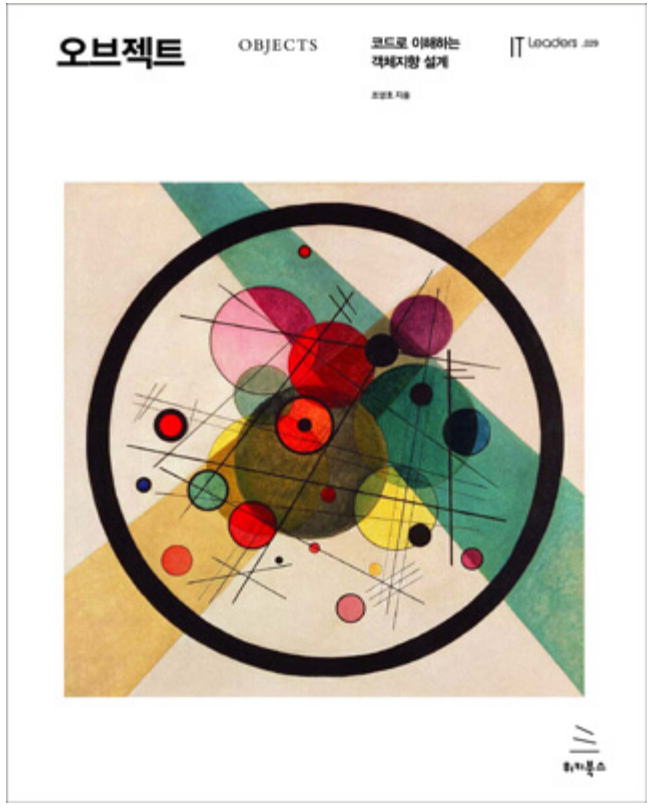
- 현재 대부분의 언어가 객체지향 패러다임을 기반으로 하고 있다. 따라서 객체지향 연습은 필수이다.
- 객체지향 설계 역량이 쌓이면 다음 단계로 함수형 프로그래밍에 도전해 보는 것도 좋겠다.

객체지향의 사실과 오해 - 조영호

- 객체지향의 핵심이라 할 수 있는 역할, 책임, 협력에 대해 쉽게 설명하고 있다.
- 객체지향 초보로서 개념적인 부분에 대해 관심이 있다면 읽어볼 것을 추천

오브젝트 - 조영호

- `객체지향의 사실과 오해` 책이 이론적인 내용을 다루고 있다면 이 책은 실 예제를 통해 객체 지향을 설명하고 있다.



객체 지향과 디자인 패턴 - 최범균

- 실 예제를 통해 객체 지향 개념에 대해 전달하고 있다.
- `객체지향의 사실과 오해` 책이 이론적인 내용을 다루고 있다면 이 책은 실 예제를 통해 객체 지향과 디자인 패턴을 설명하고 있다.