

웹 백엔드 학습 로드맵

평생을 학습하며 성장하기 위한 가장 중요한 것은?

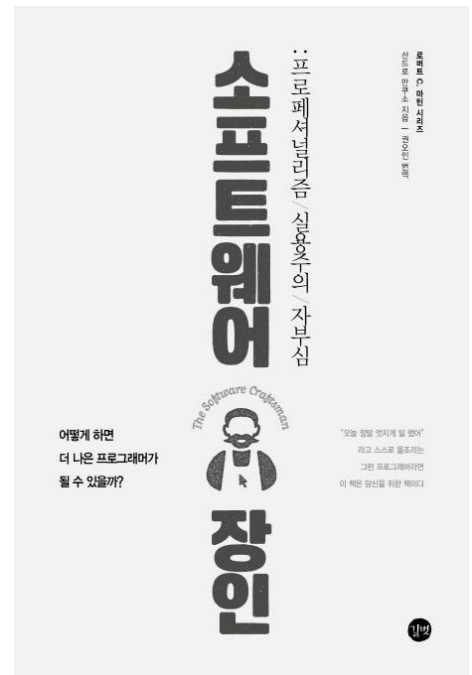
소프트웨어 장인이 되겠다는 마음가짐이지 않을까?

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

- 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 숨씨 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

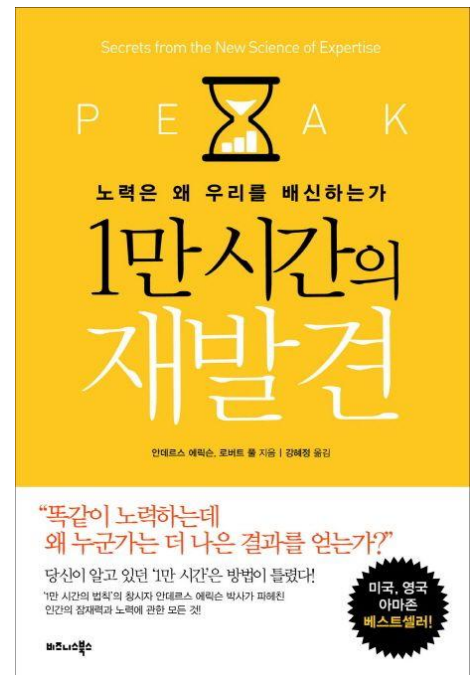
- 프리코스 4주 동안 진행한 방식은 의식적인 연습을 통해 효과적으로 학습하는 방식으로 설계해 진행했다.
- 무조건 연습을 많이 한다고 실력이 향상되지 않는다.
- 점진적으로 난이도를 높여 가고, 피드백으로 받으면서 의식적으로 연습할 때 빠르게 성장할 수 있다.

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

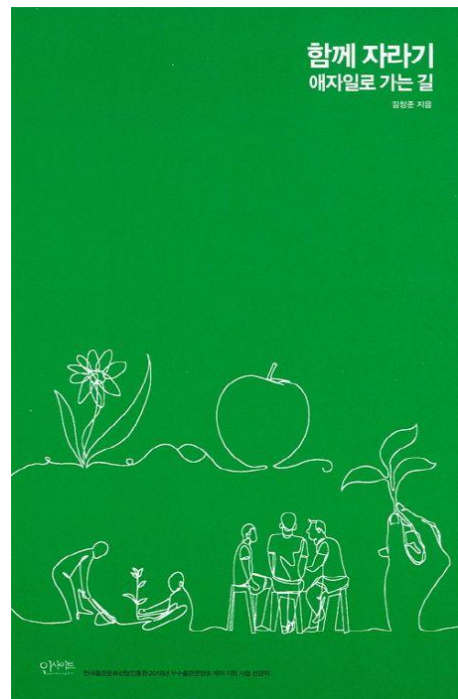


의식적인 연습을 위한 도전거리 찾기

- 클린코드 책에서 제시하는 원칙 지키기
 - 함수는 한 가지를 해야한다. 그 한 가지를 잘 해야 한다. 그 한 가지만을 해야 한다.
 - 중첩 구조가 생길만큼 함수가 커져서는 안된다. 그러므로 함수에서 들여쓰기 수준은 1단이나 2단을 넘어서면 안 된다.
- 초보자일 때 가능하면 정성적인 원칙보다 정량적인 원칙으로 연습한다.

함께 자라기

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

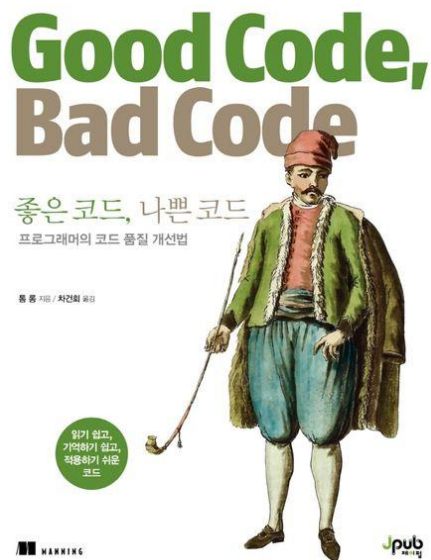


클린 코드

프리코스 과정에서 느꼈지만 현장에서는 읽기 좋은 코드, 유지 보수하기 좋은 코드를 구현하는 역량이 중요하다. 클린 코드를 구현하는 연습은 개발자로 살아가는 평생 해야 한다. 클린 코드를 구현하는 좋은 연습은 단위 테스트를 기반으로 지속적인 리팩터링을 통해 향상할 수 있다.

좋은 코드, 나쁜 코드

나쁜 코드에 대해 이유를 설명하고, 좋은 코드로 바뀌가는 과정을 볼 수 있다. 다양하고 구체적인 코드 예시를 통해 이론으로 배운 코드 작성 방법을 실제로 어떻게 적용하는지 알 수 있다. 책을 읽으며 좋은 코드에 대한 본인의 기준을 세우고, 기준에 작성한 코드에 적용하면서 학습하는 것을 추천한다.

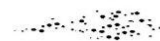


Clean Code 클린 코드

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.

Program
Programming
Programmer

클린 코드
재미있고 소스 코드를 읽는 방법
클린 코드를 위한 다양한 규칙들



Clean
Code

테스트 주도 개발 시작하기

TDD(Test-Driven Development)는 테스트부터 시작한다. 구현을 먼저 하고 나중에 테스트하는 것이 아니라 먼저 테스트를 하고 그다음에 구현한다. 구현 코드가 없는데 어떻게 테스트할 수 있을까? 여기서 테스트를 먼저 한다는 것은 기능이 올바르게 동작하는지 검증하는 테스트 코드를 작성한다는 것을 의미한다. 기능을 검증하는 테스트 코드를 먼저 작성하고 테스트를 통과시키기 위해 개발을 진행한다.

테스트 주도 개발

TDD의 구체적인 사례와 패턴을 제시하고 있다. TDD를 제대로 연습하고 경험하고 싶다면 반드시 읽고 실습해 봐야 할 책이다.

테스트 주도
개발 시작하기

TEST-DRIVEN DEVELOPMENT
적용하는 다양한 코드를 만드는 데 필요한 습관

최영근 지



KIM - 김민준 - 김민준

Program
Programming
Programmer

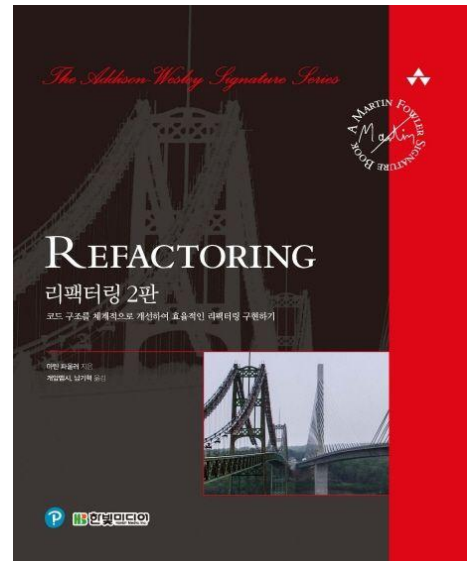
테스트 주도 개발
재미있고 소스 코드를 읽는 방법
클린 코드를 위한 다양한 규칙들



Test-Driven
Development
: By Example

리팩터링 2판

리팩터링 개념, 리팩터링이 필요한 이유와 다양한 리팩터링 카탈로그를 제공하고 있다. 책의 카탈로그가 나오기 전까지가 핵심적인 내용이다. 카탈로그는 필요한 시점에 참고하는 용도로 사용한다.

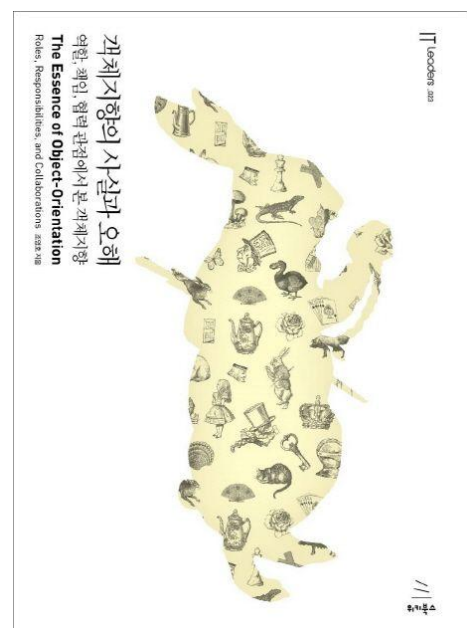


객체지향 설계

현재 대부분의 언어가 객체지향 패러다임을 기반으로 하고 있다. 따라서 객체지향 연습은 필수이다.

객체지향의 사실과 오해

객체지향이란 무엇인가라는 원론적 면에서도 다소 위험한 질문에 답하기 위해 쓰인 책이다. 안타깝게도 많은 사람들이 객체지향의 본질을 오해하고 있다. 가장 널리 퍼져있는 오해는 클래스가 객체지향 프로그래밍의 중심이라는 것이다. 객체지향으로 향하는 첫걸음은 클래스가 아니라 객체를 바라보는 것에서부터 시작한다.



자바

모던 자바 인 액션

이 책은 자바 최신 기능을 애플리케이션에 실용적으로 적용하는 방법을 알려준다. 기존 핵심 자바의 지식뿐 아니라 스트림 API, 자바 모듈 시스템 같은 최신 추가 기능, 동시성의 새로운 접근 방법, 함수형 개념 등 읽기 쉽고 유지 보수하기 쉬운 코드를 구현하는 데 어떻게 도움이 되는지 배울 수 있다.

람다, 스트림, 함수형, 리액티브 프로그래밍으로 새로워진 자바 마스터하기

Modern Java in Action

모던 자바 인 액션

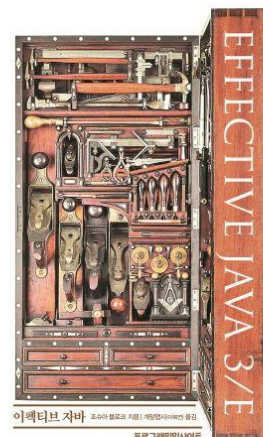
전문가를 위한 자바 8, 9, 10 기법 가이드



리플-가장먼저 읽을 무렵에, 마리오 루소, 알만 마인크로프트 지음
우영은 옮김

이펙티브 자바

이 책은 자바 프로그래밍의 과제들에 대해 실전적인 최적의 해결책을 제공하고 있다. 책의 내용이 어려울 수 있으니, 앞에 나온 책을 학습 후 역량을 쌓고 보는 것을 추천한다. 재사용 가능한 코드를 작성하기 위해 자바 프로그래밍 언어를 좀 더 깊게 이해할 수 있는 책으로 소프트웨어 분야의 최고 영예인 Jolt Award 상을 수상하였다.



이펙티브 자바 조슈아 로보트 지음 / 최만갑(이재민) 옮김
프로그래밍한사이어트