

여러분의 코드는 어느 문에 해당할까? 여러분의 팀이나 회사는? 왜 그렇게 되었는가? 그저 일상적인 코드 검토 회의인가? 아니면 출시 직후 끔찍한 문제가 드러났는가? 돌아간다고 생각했던 코드를 허접지점 디버깅하는가? 고객이 무더기로 이탈하고, 관리자가 압박을 가하는가? 난관에 부딪힐 때 옳은 문 뒤에 있으려면 어떻게 해야 할까? 답은 장인 정신이다.

장인 정신을 익히는 과정은 두 단계로 나뉜다. 바로 이론과 실전이다. 첫째, 장인에게 필요한 원칙, 패턴, 기법, 경험이라는 지식을 습득해야 한다. 둘째, 열심히 일하고 연습해 지식을 몸과 마음으로 체득해야 한다.

나는 여러분에게 자전거 타기에 관한 물리적인 지식은 가르칠 수 있다. 실제로도 고전 수학은 별로 복잡하지 않다. 중력, 마찰력, 각 운동량, 질량 중심 등은 수식 한 쪽 정도로 설명이 가능하다. 공식을 이용해 자전거 타기가 가능하다는 사실을 증명하고 자전거 타기에 필요한 지식 전부를 알려줄 수도 있다. 그래도 자전거를 처음 타는 사람이라면 100% 넘어진다.

구현도 마찬가지다. 깨끗한 코드를 만드는 “그럴듯한” 원칙을 모두 적어준 후 알아서 하라고 버려둔다면 (즉, 자전거를 타다가 넘어져도 개의치 않는다면) 그게 무슨 선생이겠는가? 그런 선생한테서 어떤 학생이 나오겠는가?

아니다. 이 책은 그냥 원칙만 가르치고 끝나지 않는다.

깨끗한 코드를 작성하는 방법은 배우기 어렵다. 단순히 원칙과 패턴을 안다고 깨끗한 코드가 나오지 않는다. 고생을 해야 한다. 스스로 연습하고 실패도 맛봐야 한다. 남들이 시도하다 실패하는 모습도 봐야 한다. 그들이 넘어지고 일어서는 모습도 봐야 한다. 결정을 내리느라 고민하는 모습, 잘못된 결정으로 대가를 치르는 모습도 봐야 한다.

이 책을 읽는 동안 마음 고생할 준비를 하기 바란다. 비행기 안에서 심심풀이로 읽어보는 “기분 좋은” 책이 아니다. 열심히, 아주 열심히 독파해야 하는 책이다. 여러분 앞에 어떤 여정이 펼쳐질까? 우선, 읽을 코드가 아주 많다. 코드를 읽으며 무엇이 옳은지 무엇이 그른지 생각을 해야 한다. 모듈을 분해했다 다시 조립하는 과정도 이해해야 한다. 이 모든 활동에는 시간과 노력이 들지만 그만한 값어치가 있다고 생각한다.

이 책은 세 부분으로 나뉜다. 처음 몇 장은 깨끗한 코드를 작성하는 원칙,



패턴, 실기를 설명한다. 코드가 많아 읽기가 힘들지도 모르지만 둘째 부분을 준비하는 단계다. 첫 부분만 읽고서 책을 내려놓는다면..... 행운을 빈다.

둘째 부분은 좀 더 어렵다. 여러 사례 연구를 소개하는데, 복잡도는 점점 더 높아진다. 각 사례 연구는 코드를 깨끗하게 고치는, 즉 문제가 있는 코드를 문제가 더 적은 코드로 바꾸는 연습이다. 상세히 살펴보려면 집중력이 필요하다. 설명과 코드를 번갈아 뒤적여야 한다. 코드를 분석하고 이해하며 코드에 가하는 변경과 이유를 납득해야 한다. 그러려면 여러 날이 걸리므로 시간도 충분히 투자해야 한다.

셋째 부분은 결말이다. 사례 연구를 만들면서 수집한 냄새와 휴리스틱(heuristic)을 마지막 장에서 열거한다. 사례 연구에서 코드를 분석하고 정리하면서 우리는 우리 행위의 모든 이유를 휴리스틱이나 냄새로 정리했다. 코드를 분석하고 고치며 우리가 느끼는 감정을 이해하려 애썼고, 그렇게 느끼는 이유와 그렇게 고치는 이유를 잡아내려 애썼다. 코드를 짜고, 읽고, 정리하는 관점에서, 우리가 생각하는 방식을 묘사한 지식 기반을 구축했다.

이렇게 구축한 지식 기반은 이 책 둘째 부분의 사례 연구를 주의 깊게 읽어보지 않으면 가치를 발휘하지 못한다. 사례 연구에서 우리는 코드를 변경할 때마다 마지막 장에 나오는 휴리스틱 번호를 표기했다. [H22]라는 식이다. 이들은 휴리스틱을 적용해 작성한 맥락을 보여준다. 여기서는 휴리스틱 자체가 아니라 사례 연구에서 코드를 정리하면서 내린 각 결정과 휴리스틱 사이의 관계가 중요하다.

둘 사이 관계를 파악하기 쉽도록 책 마지막에 부록으로 교차 참조표도 실었다. 교차 참조표는 각 휴리스틱이 쓰이는 쪽 수를 열거한다. 특정 휴리스틱이 적용된 코드를 보고 싶다면 교차 참조표를 활용하기 바란다.

첫 부분과 셋째 부분만 읽고 사례 연구를 건너뛰다면 여러분은 좋은 소프트웨어를 작성하는 “기분 좋은 책” 하나를 더 읽었을 뿐이다. 하지만 시간을 들여 사례 연구를 검토하고, 모든 결정과 단계를 이해하고, 우리 입장에 서서 우리가 생각한 방식을 이해하려 애쓴다면, 이 책에서 제시하는 원칙과 패턴과 실기와 휴리스틱을 자신의 지식으로 만들 수 있으리라 생각한다. 이 책이 그저 “기분 좋은 책”으로 머물지 않기를 바란다. 손으로 몸으로 마음으로 익히기 바란다. 자전거