

개발자로서 고민되는 것 하나가 변수의 이름을 짓는 것이다. 나의 경우에는 거의 혼자서 작업을 했기에 내가 알아볼 수 있을 정도로만 해 놓았다. 그러나, 실무에서는 많은 협업이 이루어진다. 다른 사람이 작성한 코드를 수정해야 할 때 변수 명이 제멋대로라면, 이해하는데 훨씬 많은 시간이 걸릴 것이다. 어쩌면 내가 작성한 코드를 내가 모를 수도 있다. 그렇기에, 개발자들은 암묵적인 '명명 규칙'을 만들었다. 나 역시 이것에 대해 여러 번 들었고, 기본적인 규칙은 지키고 있다고 생각했다. 그러나, 이 책은 더 세분화된 규칙을 알려 주었다. 가장 공감이 되었던 부분은 바로 '검색이 가능한 이름'이다. 예를 들어, 나의 경우 중복된 단어가 있는 변수를 자주 사용한다. Customer/CustomerService/CustomerStatus와 같이 말이다. 그러다 보니 변수를 검색해도 원하는 변수를 바로 찾기가 어려웠다. 또한 한 글자로 된 변수도 종종 사용하여, 어느 줄에 위치하는지 검색 자체가 힘든 경우도 있었다. 그나마 visual code 등에서는 intellisense 같은 부가 기능을 제공하기에 불편함이 덜했다. 그러나, 이 기능이 없었다면 코딩하는데 훨씬 많은 시간이 걸렸을 것이라 생각한다. 이번 학기의 과제부터 시작하여 좋은 변수 명을 사용해야 겠다.

3장의 내용은 함수 구현에 관련된 것이었다. 나는 여기서 중요한 두 가지의 개발 원칙을 알게 되었다. 바로 SRP와 OCP이다. 하나의 함수에는 '한 가지 기능'만 존재해야 하며 '확장'을 가능하게 하되 '수정'을 피하자는 것이다. 1학년때 나는 어싸인 등에서 구현해야 하는 함수에 모든 기능을 밀어넣었다. 추가적으로 정의한 함수는 없었으며, 그럴 필요를 느끼지 못했다. 그러나, '함수 쪼개기'를 하고 나니 코드가 훨씬 간결해진다는 사실을 알게 되었다. 뿐만 아니라 중복되는 부분을 크게 줄이고 재사용이 쉽다는 것도 느끼게 되었다. 이것은 나에게 'divide-and-conquer'을 떠올리게 했다. 책을 읽으며, 점점 내가 가지고 있던 고민이 해소되는 느낌이었다. 더욱 '좋은 코딩'을 향해 앞으로도 계속 나아갈 것이다.