

스터디 11.19.(화) 21:30

The Programmer's Brain

프로그래머의 뇌

훌륭한 프로그래머가 알아야 할
인지과학의 모든 것

펠리너 헤르만스 지음
차건희 옮김



MANNING

Jpub
제이퍼브

저는 오늘 이 책을 소개하고자 해요!

가끔 마음을 다지고자 읽는 책인데, 5번쯤 완독을 했던 것 같은데, 전부 따라가지는 못해서.
최대한 따라할 수 있는 건 따라하려고 노력합니다..!

PART | 코드 더 잘 읽기

CHAPTER 1 코딩 중 겪는 혼란에 대한 이해 3

1.1 코드가 초래하는 세 가지 종류의 혼란 4

1.1.1 혼란의 첫 번째 유형: 지식의 부족 5

1.1.2 혼란의 두 번째 유형: 정보의 부족 6

1.1.3 혼란의 세 번째 유형: 처리 능력의 부족 6

1.2 코딩에 영향을 주는 인지 과정 7

1.2.1 LTM과 프로그래밍 8

1.2.2 STM과 프로그래밍 8

1.2.3 작업 기억 공간과 프로그래밍 10

1.3 인지 과정들의 상호작용 10

1.3.1 인지 과정들이 어떻게 상호작용하는지에 대한 간단한 설명 10

1.3.2 프로그래밍 업무와 관련한 인지 과정 11

요약 13

처음 발표를 하는 것이다 보니 저 같은 경우는 그냥 이런 원론적인 내용들을 공유하고 싶었어요.

앞으로 프로그래밍 언어를 더 배워야 하는 상황이 왔을 때를 대비하는 차원에서도 중요하다고 생각을 했어요.

우리가 코드를 봤을 때나, 파일을 읽을 때 혼란스러운 경우는 세 가지가 있을 거예요.

MECE 원칙이라고 들어보셨나요? 중복없이 빠짐없이 분류를 하는건데. 중복없이 빠짐없이 분류를 하는 건 여러분들의 머리를 좀 더 정리하는데 도움이 될 거라고 생각을 했어요.

과학기술학의 유명한 학자로 라투르라는 학자가 있어요.

이 사람의 블랙박스를 같이 언급하고 싶은데. 우리가 알고 확실하다고 믿는 걸 블랙박스라고 표현을 해요. 사실 더 알고 싶다면 관련 저서를 읽어야 하는데, 어렵다하여 많은 사람들이 읽기 힘들어하긴 한데.

블랙박스라는 개념을 간단하게 말하면, 안정화된 상자인데, 여기를 많은 사람들이 열어보지 않아요. 열어봄으로써 더 많은 것을 알게 되는 것이거든요.

혼란의 세 가지 유형

1. 지식의 부족

- 필요한 프로그래밍 개념이나 기술에 대한 이해가 부족할 때 발생합니다.
- 특정 알고리즘의 작동 방식을 모르면 해당 코드를 이해하기 어렵습니다.

2. 정보의 부족

- 변수명이나 함수명이 명확하지 않거나 주석이 부족할 때, 코드의 의도를 파악하기 어려워집니다.
- 의미 없는 변수명이나 불충분한 주석으로 인해 코드의 목적을 이해하기 힘든 경우.

3. 처리 능력의 부족

- 복잡한 로직이나 구조로 인해 코드를 분석하고 이해하는 데 어려움을 겪을 때 발생합니다.
- 중첩된 조건문이나 복잡한 데이터 흐름이 있는 코드.

이 유형에 따라서 처리하는 방법이 달라질 것 같아요. 필요한 프로그래밍 개념이 부족한 경우라면 그 개념을 학습해서 채워야하는 것.

정보가 부족하다면, 의도 파악하기 위해서 정보를 좀 더 찾아보는 것.

복잡한 것 때문이라면 단순하게라든지, 처리하기 위한 생각을 바로 잡아보는 등의 행위를 하는 것이 필요할 것이라고 생각해요.

CHAPTER 4 복잡한 코드 읽는 방법 51

4.1 복잡한 코드를 이해하는 것이 왜 어려울까? 52

4.1.1 작업 기억 공간과 STM의 차이 53

4.1.2 프로그래밍과 관련한 인지 부하의 종류 53

4.2 인지 부하를 줄이기 위한 기법 55

4.2.1 리팩터링 55

4.2.2 생소한 언어 구성 요소를 다른 것으로 대체하기 57

4.2.3 플래시카드에 코드 동의어 추가 60

4.3 작업 기억 공간에 부하가 오면 사용할 수 있는 기억 보조 수단 61

4.3.1 의존 그래프 생성 61

4.3.2 상태표 사용 64

4.3.3 의존 그래프와 상태표의 혼용 66

요약 68

인지 부하의 종류

1. 내재적 인지 부하

- 문제 자체의 복잡성에서 오는 부하
- 복잡한 알고리즘을 이해하려면 많은 인지 자원이 필요.

2. 외재적 인지 부하

- 코드의 표현 방식이나 지식 부족에서 오는 부하.
- 비효율적인 코드 구조나 불명확한 변수명은 외재적 인지 부하를 증가시킴.

우리는 작성하면서도 최대한 인지 자원을 적게 쓰면서 효율적으로 활용해야 할 거예요.

책에서 소개하는 인지 부하를 줄이는 방법.

- **리팩터링**

- 코드를 좀 더 이해하기 쉽게 구조를 변경.
- 긴 함수를 여러 개의 작은 함수로 나누면 코드의 가독성이 높아짐.

- **의존 그래프 & 상태표 사용**

- 코드 간의 관계를 시각적으로 표현하여 이해에 도움이 됨.

CHAPTER 10: 복잡한 문제 해결을 더 잘하려면

- **문제 해결 과정과 기억의 역할**

- 암시적 기억과 명시적 기억이 문제 해결에 중요한 역할을 합니다.
- 암시적 기억은 자동화된 작업을, 명시적 기억은 능동적으로 호출되는 정보를 의미
- 암시적 기억은 자주 사용하는 단축키나 알고리즘의 기본 구조를 기억하는 것이고, 명시적 기억은 특정 함수의 사용법을 기억하는 것.

- **자동화와 암시적 기억**

- 타이핑이나 단축키 암기와 같은 자동화된 기술을 통해 암시적 기억을 강화하면 문제 해결 속도가 빨라짐.
- 자주 사용하는 함수나 라이브러리의 사용법을 암기하면 코딩 시 더 빠르게 작성할 수 있음.

→ 책을 홍보하려는 목적은 아니고, 인지과학적으로 프로그래밍을 한다면 이렇게 접근을 할 수 있을 것 같아서. 혹시나 내가 왜 이것 이해하지 못하지? 라고 한다면 이런 접근방법도 괜찮을 것 같다는 생각을 해서 이번 발표는 이렇게 진행을 했습니다..

이 책의 전부를 소화한 것은 아니지만, 올해 5월부터는 이렇게 하려고 노력은 해왔거든요. 초심을 다지고자 읽는 책이 되어버려서 나름 애장하는 도서 중에 하나가 되었습니다. 책은 한번 읽는 것이 아니라 두 세 번 읽어야 와닿는다는 입장을 가지고 있어서 더 그렇습니다..!