

# 7장. 경험주의: 현실을 자각하자

## 경험주의

- 과학 철학에서는 경험주의를 “특히 실험에서 발견한 증거를 강조하는 것”으로 정의됩니다.

## 자기기만은 우리의 적

- 경험주의자가 되기 위해서는 현실에서 수집한 신호를 어떻게 조합하고 실험을 통해 테스트할 수 있는 이론으로 조립하는지에 대해 체계화 할 필요가 있습니다.
- 인간은 스스로를 속이도록 진화했기 때문에 온갖 추측을 통해 성급하게 결론을 내리고 있습니다.
- 리처드 파인먼은 과학의 특징을 다음과 같이 설명한 것으로 유명합니다.
  - “첫 번째 원칙은 자신을 속이지 말아야 한다는 것인데, 속이기 가장 쉬운 사람이 바로 자신이다.”

## 신화와 현실 분리하기 : 예시

몇몇 특이한 작업의 경우 병렬 실행이 해답이지만, 여러 실행 스레드의 정보를 다시 한데 모아야 할 필요가 생기면 상황은 달라집니다. 병렬화가 정답이라고 설불리 결론을 내리는 대신 몇몇 데이터를 수집해 보면 아래와 같습니다.

[예제] 간단한 정수를 5억번 증가시키는 알고리즘 작성

방법	시간(밀리초)
단일 스레드	300
잠금을 사용한 단일 스레드	10,000
잠금을 사용한 두 스레드	224,000
CAS를 사용한 단일 스레드	5,700
CAS를 사용한 두 스레드	30,000

“동기화를 위한 코드를 도입하자마자 (동시성 전문가가 아닌 이상) 예상하지 못했던 비용을 목격하기 시작한다. 여전히 단일 스레드에서 모든 작업을 수행하지만 다른 스레드에서 결과를 사용할 수 있도록 잠금을 추가하면 비용에 9,700밀리초가 추가된다. 잠금은 매우 비싸다! 작업을 2개의 스레드로만 나누고 작업 결과를 동기화하기로 결정하면 단일 스레드에서 작업을 수행하는 것보다 746배 느린다

따라서 잠금은 매우 비싸다. 스레드 사이의 작업을 조정하기 위해, 사용은 더 어렵지만 더 효율적인 방법이 있다. 이를 위해 가장 효율적인 방법은 비교와 교환 (compare-and-swap, CAS)라는 저수준의 동시성 접근 방식이다. 안타깝게도 이 접근 방식도 단일 스레드에서 작업하는 경우보다 100배 더 느린다.

이런 피드백을 바탕으로 더 많은 정보와 근거에 기반한 결정을 내릴 수 있다. 알고리즘의 진행 속도를 극대화하려면, 진행하면서 결과를 결코 다시 합칠 필요가 없는 경우에 가능한 한 많은 작업을 단일 스레드에 유지하려고 노력해야만 한다”

## 추측보다는 실험: 현실에 입각한 경험주의

- 추측, 가정, 자만심을 바탕으로 진행하는 대신 회의적인 태도로 자신의 아이디어를 확인하는 것은 힘들지만 진정한 진전을 이룰 수 있는 유일한 방법입니다.
- 가장 좋은 시작 방법은 자신이 알고 있다고 생각하는 것이 틀렸을 가능성이 있다고 가정하고 나서, ‘어떻게 틀렸는지’를 알아내는 방법을 찾아내는 것입니다.

## 정리

- 순수 과학이 아닌 공학은 해법의 실용성을 고려하게끔 요구 합니다. 바로 여기서 경험 주의가 작동하게 됩니다.
- 이는 잘못된 과학이자 잘못된 공학입니다. 하지만 공학은 실용적인 학문입니다. 따라서 우리는 우리의 추측과 이를 테스트하기 위해 만든 실험에 대해 끊임없이 회의적인 태도를 취하고 현실의 경험과 비교하면서 확인해야만 합니다.

### 💡 7장: 경험주의: 현실을 자각하자

7장에서는 소프트웨어 개발에서 현실에 대한 증거와 관찰을 바탕으로 의사결정을 내리는 경험주의의 중요성을 강조합니다. 경험주의는 합리적인 발전을 위한 필수적인 요소로, 빠르게 바뀌는 세상에서 우리가 '학습의 전문가'가 되어야 한다는 책의 핵심 주제와 연결됩니다.

#### 💡 경험주의의 핵심 원칙 및 실천 방안

경험주의는 단순히 수동적으로 사건을 경험하는 것을 넘어, 현실을 자각하고 이를 통해 학습하는 능동적이고 의식적인 실천 행위입니다.

##### • 현실 기반의 결정

- 경험주의는 추측(짐작)에만 근거하여 진행하는 대신, **실험의 일부로 수집된 정보를 사용하여 의사결정을 내리는 방식입니다.**
- 소프트웨어 개발이 탐구와 발견의 과정이므로, 우리는 **실수가 불가피하게 발생할 것이라는 가정**에서 출발하고 실수로 인한 비용을 완화하는 방식으로 작업해야 합니다.

##### • 실험과 경험의 분리

- 경험주의자는 아이디어의 결과를 덜 형식적으로 관찰함으로써, 다음 실험을 생각할 때 **현재 상황에 대한 특성화의 품질을 개선할 수 있습니다.**
- 이는 다음 장에서 다룰 **실험주의** (과학적 사고와 실천)를 대체하는 것이 아니라 보완하는 방법입니다.

#### 자기기만과의 투쟁: 현실을 왜곡하는 습성

저자는 경험주의자가 되는 것은 보기보다 더 복잡하며 훈련이 필요하다고 지적합니다. 인간은 생물학적으로 성급하게 결론을 내리도록 진화했으며, 이 때문에 현실에 근거하지 않은 함정에 빠지기 쉽습니다.

- **명백해 보이는 오류의 함정:** 저자는 고성능 금융 거래소에서 겪었던 경험을 예시로 제시합니다. 팀원들은 API 테스트 중 오류가 발생하자, 겉으로 드러난 증상만을 보고 새로운 메시징 코드가 문제라고 성급하게 결론 내렸습니다.
- **사실 수집의 중요성:** 이들은 잠시 멈춰 서서 **기존 메시징 코드가 일주일 넘게 성공적으로 실행되어 왔다는 사실**을 인지한 후에야, 이 문제가 '메시징 문제'가 아니라 새로운 코드의 스레딩 버그임을 깨달았습니다. 팀은 선부른 결론(추측) 위에 새로운 추측을 쌓아 올리는 실수를 저질렀습니다.
- **과학적 회의주의:** 문제 해결을 위한 과학적이고 합리적인 접근 방식의 중요한 측면은 **회의주의입니다**. 노벨상 수상자인 리처드 파인먼 (Richard Feynman)은 "자신을 속이지 말아야 한다"고 경고하며, **가설을 세우고 이를 증명하거나 반증하는 실험을 수행해야 함**을 강조했습니다.
- **추측 대신 실험:** 병렬화가 코드를 빠르게 만든다는 학계의 일반적인 지혜(미신)가 실제 실험(측정) 결과, 오히려 단일 스레드 코드보다 훨씬 느리고 복잡한 코드를 만들어냈다는 사례도 제시되었습니다. **추측보다는 실험에 입각한 경험주의가 진정한 진전을 이룰 수 있는 유일한 방법입니다.**

요컨대, 7장 경험주의는 소프트웨어 개발의 모든 단계에서 **추측이나 가정 대신 현실에서 수집된 증거와 피드백을 바탕으로 학습하고 진전을 이루어야 한다는 철학적 토대를 제공합니다.**