

# 프로그래밍의 본질

Dongjae Lee

2025.07.03

## Abstract

프로그래밍의 본질은 무엇일까? 자연어로 사람에게 명령하는 것과 컴퓨터에게 프로그래밍 언어로 명령하는 것의 근본적 차이는 무엇인가? 바이트 코딩이 이 시대의 버즈워드이자 논쟁의 중심이 된 지금 이 시점에 프로그래밍의 본질이 무엇인지 고민해봐야 한다. 이 글에서는 프로그래밍의 본질을 탐구하고, 왜 형식 언어 기반의 프로그래밍 언어가 불멸할 것인지 설명한다.

바이트 코딩 이전에는 노 코드가 있었고, 결국은 실패했다. 근본적인 이유는 복잡성의 한계, 유지보수의 어려움에 직면했기 때문이다. 간단한 프로토타입을 만들어 볼 수는 있지만 복잡한 서비스를 구축하고 보안성을 높이고 유지보수를 하는 것은 코드를 모르고서는 불가능했다. 바이트 코딩도 지금까지는 비슷한 길을 걷고 있다.

이러한 현상이 발생하는 이유는 자연어의 모호함에 있다. 자연어는 본질적으로 모호하다. 즉, 같은 문장도 여러 가지 방식으로 해석될 여지가 남아있으며, 그 해석조차 자연어이기 때문에 모호하다. 재귀적으로 모호한 요소들을 하나씩 제거하여 어떤 문장이 하나의 의미로만 해석될 수 있다면 이것은 자연어가 아닌 형식 언어다.

일반적인 작업의 경우 여러 사람들의 해석이 동일할 확률이 높으며, 이는 common-sense라는 개념으로 설명된다. 노 코드, 바이트 코딩이 뻥한 앱에서 잘 동작하는 이유다. 반면, 특수한 작업의 경우 자연어로 설명하는 것이 더욱 장황하며 오히려 엄밀한 형식 언어로 설명하는 것이 더 짧고 명확할 때가 많다.

프로그래밍의 본질은 자연어로 설명된 것을 형식언어로 번역하는 과정이다. 이 과정에서 모든 것이 엄밀하게 정의된 형식 언어로 표현함으로써 모호함이 자연스럽게 제거된다. 고객의 모호한 요구사항도 코드로 구현하려면 엄밀하게 정의할 수 밖에 없다. 하나의 의미로만 해석되는 요구사항을 강제하는 것이 형식 언어의 역할이다.

각자의 역할이 다르므로 자연어는 프로그래밍 언어를 대체할 수 없다. 본인의 의도를 완벽하게 설명하기 위해서는 형식 언어가 반드시 필요한데, 그것은 결국 코딩이다. 자연어로 의도를 설명하면 모호함이 발생하고, 형식 언어로

바뀌는 과정에서 의도와 다른 부분이 생길 가능성이 항상 존재한다. 따라서, 프로그래밍 언어는 자연어를 대체할 수 없다.