Essense of Programming

Dongjae Lee

2025.07.03

Abstract

What is the essence of programming? What is the fundamental difference between giving instructions to a person in natural language and giving instructions to a computer in a programming language? As "vibe coding" becomes a buzzword and a main topic of debate in our time, it is worth reflecting on the true nature of programming. This article explores the essence of programming.

Before vibe coding, there was no-code, which ultimately failed. The fundamental reason was the limits of complexity and the difficulties of maintenance. While it was possible to build simple prototypes, creating complex services, ensuring security, and maintaining systems were impossible without understanding code. Vibe coding, so far, is following a similar path.

The reason for this phenomenon lies in the ambiguity of natural language. Natural language is inherently ambiguous. That is, the same sentence can be interpreted in multiple ways, and the interpretation itself is natural language, so it is ambiguous too. If we recursively remove the ambiguous elements one by one, and if a sentence can be interpreted as one meaning only, then it is not natural language but a formal language.

The essence of programming is the process of translating natural language descriptions into formal language. In this process, the ambiguity is naturally removed because of the strict definition of the formal language. Forcing a requirement to be interpreted as one meaning only is the main role of a formal language.

Since each has a different role, natural language cannot replace programming languages. To describe someone's intention with only one interpretation, a formal language is necessary, and this is coding in the end. If we explain the intention in natural language, the ambiguity will occur, and there is always a possibility that the intention will be differently interpreted during the process of translation. Therefore, programming languages cannot replace natural languages.