

ぶっくす

武市正人 著

プログラミング言語

岩波講座 ソフトウェア科学 4, 岩波書店

なにも **bit** に限ったことではないが、日本の書評は概して甘口である。たまに批判めいたことを書く場合も、その表現は婉曲をきわめる。しかし、この本についてはそのような余分な配慮をせずともよいのは、評者にとって幸いである。文句なしに素晴らしいし、およそ情報科学を志す人、ソフトウェアに関心のある人なら、誰にでも薦められる。

この本は表題通り「プログラミング言語」を扱ったものだが、このテーマでまとまりのある1冊の本を書くのは、難題のはずである。ある特定のプログラミング言語についてなら、すでにいくらでも本がある。多様なプログラミング言語を集めて横並びに論じてみても、寄せ集めのゴッタ煮になりかねない。それらの共通項を一般化した抽象言語で議論しようにも、どのような言語要素を取り込み、どのような記法を用いるかが大きな問題だし、一方で具体的なプログラミングとの結びつきが弱くならざるをえない。

著者は見事な工夫でこの問題を解決し、一貫性のある書をなした。その方針は次のようである。プログラミング言語について説明するためのメタ言語として、プログラミング言語を用いる。これは循環的な状況を生む恐れがあるが、メタ言語として数学的に性質が明快な関数型プログラミング言語（具体的には **Gofer**）を採用することにより、その可能性を最小限とする。言語の説明とはその構文と意味であり、それをプログラミング言語で記述することにより、形式的で厳密な議論ができるとともに、処理系によって具体的な動作の確認もできる。これを著者は言語プログラミングと名付けている。つまり抽象的な言語機能を取り扱いながら、具体的なプログラミングも学べるというわけだ。

着想はよい。それをどう実現するかが、さらに著者の腕の見せどころだ。まず、第2章で「言語プログラミング」のための言語としての **Gofer** を紹介している。これが簡にして要をえた解説で、関数型プログラミングには門外漢の評者にも、以降の章を理解するのに十分な基礎を与えてくれる。この説明は非形式的で、次章以降の形式的議論と対照をなし、それだけでも本書の構成を面白くしている。

第3章で、プログラミング言語を定義するとはどういうことかを、これまた簡明に、しかし適切な例によって具体的に記述した上で、構文規則と意味の定義に関数型プログラミング言語をどう用いていくかを示して、以降の記述の基本路線を明確に提示している。

第4章以降で、典型的な構文要素を順に取り上げて、構文を与え、その意味を表示的意味論に従って定めていく。取り上げられるのは、式と値、環境、代入、記憶域と変数、制御構造、継続、手続き、型づけ、データ構造（リスト、配列、直積、直和、ポインタ、関数）、型検査、閉包、単一化機構、多様型関数などである。これを一覧するだけでも、簡単どころだけで済ませてお茶を濁そうなどという意図はまったくなく、よく分かる

だろう。

とかく表示的意味論というのとつきにくい。しかし、本書はその難解な部分を巧みに避けている。一方で、意味論の入門書によくあるような、適用対象を式と簡単な制御文だけですませるというつまらなさからも、見事に免れている。方法としては、これらの構文要素を含む統一的な仮想的言語の仕様全体をいきなり与えるというやり方をとっていない。それぞれの構文要素はそのつど導入定義され、説明される。評者にはこの方法も、とても読み易くてよい工夫と感じられた。

最後の第 9 章で、プログラムの翻訳実行までを同じ枠組みで説明している。そしてこれだけの内容をわずか 220 ページほどにまとめていることも、読者としてはありがたい。しかもこの短さは、面倒なところを端折ったり、形式化に徹して骨だけに削りあげたことからくるものではない。

Gofer の処理系や、本書にでてくるプログラムの入手方法まで、親切に示してある。書評の常道としては、「処理系を入手してプログラムを試してみると、さらに理解が深まろう」などというところであろうが、評者は自分でそれを実行していないので言う資格がない。むしろ Gofer を知らなかった評者でも、処理系に頼らずともプログラムがみなよく読めたということを強調しておきたい。

(玉井哲雄 東京大学 教養学部)