…› 久野　靖（筑波大学）

プログラム書法（第 2 版）

Brian W. Kernighan，P. J. Plauger 著
木村 泉 訳　共立出版（1982）　ISBN : 978-4-320-02085-6

「ビューティフルコード」

やや旧聞になるが，2012 年の夏のプログラミングシンポジウム（通称プロシン）は「ビューティフルコード programming should be fun ; programs should be beautiful」と称して「コードの美しさ」をテーマに開催された．

私はその名も「ビューティフルコード [1]」という本の翻訳者であるためか，何か喋れということになり，コードの美しさについて自分はこう思う，のような話をさせていただいた [2]．もともとこの話題については昔から関心を持っているところだし，プログラミング言語を研究対象としている中でも「書きやすさ／読みやすさ」はとりわけ重要だと思っていることもあった．

とは言っても，勝手な自説だけで 30 分も喋るわけにいかないわけで，半分くらいは過去に遭遇したコードの美しさに関する書籍や文献の紹介をした．その中でトップバッターとして取り上げたのがここに紹介させていただく「プログラム書法（第 2 版）」である．

「プログラム書法」とは…

「プログラム書法」はその原著が 1974 年（原著の和訳が 1976 年）に刊行され，かなり評判となったことから（と思う），わずか 4 年後の 1978 年にここで取り上げる第 2 版が出ている．和訳の刊行は 1982 年であるが，30 年（！!）経った今日でも現役であり，少し大きい書店に行くと棚に並んでいるという，変遷の激しいソフトウェア技術分野としてはバケモノみたいな本である．

その中身であるが，名前通りプログラムの「書法」つまり「好ましい書き方」についてさまざまな側面から記されている．その特徴は，好ましい書き方についての指針が 1 ～ 2 行くらいの「規則」として提示されていて，それを掲げては具体例を示して説明する，という体裁になっていることである．

ちなみにこの本の原題は「The Elements of Programming Style」で，好ましい英文の書き方について書かれた超有名な本である「The Elements of Style」にちなんでいる．規則を掲げて説明という方式もこの「お手本」から借りてきている感じであるが，プログラミング関係の書籍でこのようなスタイルのものは初めてだったので，私が本書に出会ったときは非常に新鮮だった（というよりそもそも，プログラムを美しく書こうという本に初めて出会ったことが衝撃だったのかもしれないが）．

出だしから…

この本の出だしは，次のように始まる．

プログラムの一部分に次のようなところがあったとする．

```
      DO 14 I=1, N
      DO 14 J=1, N
   14 V (I, J) = (I/J) * (J/I)
…
```

いきなり 2 行目からプログラムコードが出てくる本というのも相当である☆1．そしてこれが何をするかの種明かし（V(I,J) には I と J が等しいときのみ 1，それ以外は 0 が入るから，V は単位行列になる）があり，その後「多くの人はこういうコードをなんとなく眺めて先に進むからよくない，もっと単位行列になることを明確に書くべきだ」という風に話が進んで，最初の規則が提示される．

わかりやすく書こう．
--- うますぎるプログラムはいけない

この先もこの本のスタイルは大体同様であり，まず

☆1 ちなみにこのコードは Fortran で，あと本書では PL/I も使われている．いずれも今ではマイナーな言語だが，この本が今でも売られているということは，この本を買う人にとってはどの言語ということはあまり問題ではないということなのだろう．

問題のあるコードが出てきて，それを具体的に改良してみせ，規則を提示して一見落着，という形が繰り返されている．

わかりやすく書こう

ちなみに，ダッシュが使われている規則はこの本では 2 つだけで，もう 1 つは次のものである．

わかりやすく書こう
---「効率」のために
わかりやすさを犠牲にしてはいけない

これも冒頭から比較的すぐのところに出てくるので，この「わかりやすく書こう」ということが，この本の重要なテーマだと分かる．

自分では，このことはこの本でも目にしたし，恩師の木村 泉先生（この本の訳者でもある）にも常々言われていたことなので，「プログラムを書くときの常識的な心がまえ」なのだと思っていたのだが，後にネットなどでさまざまな人と議論するようになると，必ずしもそのように思っていない人が多数いることがわかってきた．

たとえば JavaScript の話だが，今でも 2 ちゃんねるなどでは，「x に入っている実数を整数に切り捨てて整数にするには ~~x を使う」だとか☆2，「配列 a の後ろに要素を追加するのには「a[a.length] = 値 ; が高速だ」，みたいな書き込みにお目にかかる．そんなのより「Math.floor（x）」「a.push（値）;」と「わかりやすく」書く方がよっぽど価値があるはずなのに…．

ちなみに，プログラム書法の教えは，そういう人間にとってわかりにくいコードは（間違いを誘発するなどの結果）プログラムの完成を遅れさせてしまうので，結局は損である，というものである☆3．このことは本書が書かれた時代以前から今日に至るまでずっと真実であるはずなのだが，前記のことからすると，この真実の教えを受けないままにプログラミングを学び，プログラムを書いている人が世の中にはたくさんいるわけである．これは我が国のプログラミング文化のためには困った事態ではないだろうか…．

☆2 ビット反転演算子「~」を使うとまず x を 32 ビット整数に変換してから反転するので，さらにもう 1 度反転すると反転前の整数が得られる．
☆3 どうしても性能のため読みやすさを犠牲にする部分もあるかもしれないが，それは最内側ループ内など，コードの中の非常に限られた個所だけのはずである．

その他の規則例

紙面がなくなってきたので，あとは規則の中からこれはと思うものをいくつか挙げるので，味わっていただきたい（当たり前だよねという人も多いはずだけど）．

- *多方向の枝分かれを作るときには IF…ELSE IF…ELSE を使おう．*
- *まずわかりやすい擬似言語で書いて，そのあと目的の言語に翻訳しよう．*
- *プログラムが簡単になるようなデータ表現を選ぼう．*
- *第 1 版ができたところでやめてしまってはいけない．*
- *だめなプログラムを修正するのはやめて，全部書き直そう．*
- *0.1 の 10.0 倍はまず決して 1.0 にはならない．*
- *浮動小数点表示の数を等しいかどうかに着目して比較してはならない．*
- *速くする前に正しくしよう．*
- *速くしたかったら単純さを保とう．*
- *だめなプログラムに注釈をつけるのはよそう．プログラムの書き方を変えよう．*
- *注釈をつけすぎないようにしよう．*

「よいコード」にもっと注目を

ソフトウェアの世界では得てして「動けばよい」「動いているものは壊すな」という考えのため，汚いコードがはびこりやすい．私としてはぜひとも，そういう考えが改まって，「よいコード」がもっと注目されるようになってほしい．その意味でも「プログラム書法」はこれからも読み継がれてほしい本である．なお，「よいコード」本の新顔として「リーダブルコード[3]」というのも最近刊行されている．こちらもおすすめである．

参考文献

1）Kernighan, B., Bentley, J., まつもとゆきひろ他 著，Oram, A. 他編，久野禎子，久野　靖 訳：ビューティフルコード，オライリー（2008）．
2）久野　靖：ビューティフルコードのための N 個の指針，夏のプログラミングシンポジウム 2012 報告集，pp.35-46（2012）．
3）Boswell, D., Foucher, T. 著，角　征典 訳：リーダブルコード，オライリー（2012）．

（2013 年 8 月 1 日受付）

久野　靖（正会員）　kuno@gssm.otsuka.tsukuba.ac.jp
1984 年東京工業大学理工学研究科情報科学専攻単位取得退学．同年同大理学部情報科学科助手．筑波大学講師，助教授を経て現在，同大学ビジネスサイエンス系教授．理学博士．プログラミング言語，ユーザインタフェース，情報教育に関心を持つ．