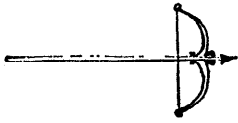


談話室



本格派スパゲッティ・プログラムの勧め†

小林 修†

1. ま え が き

goto 文を多用した旧タイプのプログラムに、スパゲッティ・コードという綽名が付けられたのは、20年ほど前のことと思われるが、スパゲッティにとっては、はなはだ不本意なことであったに違いない。

確かに、三流の料理店が出す「スパゲッティ」は、小さな皿に麺を積み上げ、フォークを挿して持ち上げると麺が一塊となって皿から浮き上がるような代物が多いが、これはスパゲッティにあらず。麺をフォークに絡ませてクルクルと回せば、麺がまとまって程よい大きさになる、スパゲッティはこうでなくてはならない。したがって、スパゲッティ・コードという言い方は、スパゲッティにとっては、濡れ衣であり、三流プログラマのとはっちりと言うべきである。

そこで、物言わぬスパゲッティに代わり、その名誉を回復すべく、本格派スパゲッティ・プログラムについて、蘊蓄を傾けることとした。

2. 料理の三要素

前章で三流料理店と書いたが、これはいわゆる大衆食堂を指して言っているのではない。そこは、まず腹を満たし必要な栄養価を摂取するための場所である。料理店とは、おのずと目的を異としており、一流三流の関係にはないのである。もっとも、昼時にサラリーマンでどった返す食堂は、味も良いことが多い。

ハードウェアに直接結びついたブート・ストリップのコードは、そんな店の一つと言えるかもしれない。たとえば、EDSAC の 40 ワードほどのコードを見るとよい¹⁾。それは、必要な栄養価

(プログラムの妥当性)を有し、味もよい(アルゴリズムも絶妙である)。ただし、必ずしも品良くまとまっているわけではない。盛りつけもぞんざいだし、食事を楽しむという雰囲気欠ける。つまり、「客をもてなす」サービス(プログラムの可読性)がないのである。しかし、食堂としての目的は十二分に満たしていると言うべきであろう。

それに対し、料理店は、そこで「楽しい食事」をする場所であり、料理の栄養価や味もさることながら、盛りつけや店の雰囲気までもがサービスを構成する重要な要因となっている。

そこでは、料理は三つの要素から成り立っている。すなわち、①栄養価(プログラムの妥当性)、②味(アルゴリズム)、③盛りつけ(プログラムの可読性)の三つである。

2.1 栄養価

トータルに考えて、必要な栄養価を供給できない食事は、欠陥品である。ここで、「トータルに考えて」とは、「1週間を平均して」あるいは「コースの単品を合計して」という意味である。単品の料理の多くは、栄養的にアンバランスであろうが、トータルすればバランスしているものである。良く造られたコースは、個々の料理を殺さずに、全体として程よく調和している。

同様に、プログラムもまた、所与の仕様を満足しなければならない。システムがいくつかのプログラムからなる場合には、個々のプログラムの機能が相互に調和している必要がある。たとえば、入力データの健全性をどの時点で保証するのか、などのモジュール間のインタフェースは、システムのアーキテクチャを設計する際の主要項目の一つである。単一のプログラムを設計するときもプログラム内のモジュールについて、オペレーティング・システムとの分担について、あるいはヒューマン・インタフェースについても、同様の

† The Authentic Spaghetti Program by Osamu KOBAYASHI (SANNŌ Junior College).

† 産能短期大学

ことが言える。

ちなみに、「たくあん」には、「身に付く」という意味での栄養はほとんどないが、整腸という大きな役割りがある。

C言語の行マクロなどは、さしずめ、プログラム・コードの「たくあん」と言えよう。マクロにすれば、展開形を複数個所にコードする手間が省けるとか、訂正が楽だとか言われるが、これらは実はほんの副次効果にすぎない。プログラムの妥当性を確保する上で、`isdigit(c)` や `n < MaxCount` という簡潔かつ自明な表記の効果は計り知れないのである。

2.2 味

料理は、味が良くなければならない。味を創るには三代かかるというが、それほどに微妙なものである。しかも、文化や個人の差が大きい。分析にかからぬ部分が大きいのは、名人芸に共通する特徴である。

さて、アルゴリズムはどうか。これも、名人芸に属すると言ってもよいのではなからうか。アルゴリズムの定義からして、チューリング機械による定義や帰納的関数による定義など幾つかあり、数学的に同値であることが分かっているが²⁾、もう一つピンとこない。これに対し、たとえば、実数の古典的な定義には、カントルの縮小区間列やデデキントの切断、ワイエルストラスの上限・下限によるものなどがあり、どれも同値であることが示されている。そして、そのどれもが「びっしりと隙間なく詰まっている」という実数のイメージとよく一致しているのである。

翻って、アルゴリズムに関しては、「良いアルゴリズムとは何か」も状況により人により異なるし、計算量が最小のアルゴリズムの構成も、経験がものを言う名人芸の世界のようである。

もっとも、計算機が生まれて半世紀足らず、三代には遠く及ばない。アルゴリズム・センスが定着しないのも当然かも知れない。

これまでに述べた二点 (①栄養価と②味) だけを満たすスパゲッティも、そうざらにあるわけではない。看板を頼りに店に入っても、うまいスパゲッティにありつけるとは限らないのである。

プログラム (の妥当性とアルゴリズム) もまた同様であり、「パソコンのプログラムはバグ付きで売っている」と喝破した人がいたが、そう言わ

れてもやむをえない状況にあったことは事実である (今もある?)。この状況は、大型機の場合も変わらず、「スパゲッティ・コード」と蔑む以前の問題として、プログラマにアルゴリズム概念がないのではないかと疑わせるようなプログラムも散在する。

2.3 盛りつけ

料理がもてなしの心であるとするれば、料理の盛りつけもまた、重要な要素の一つとなる。短くちぎれた麺と具をかき混ぜて椀に盛って出されても食べる気にはなるまい。あるいは、いったいどこからどうやって食べたらよいのか皆目分からぬ洋菓子もある。造った当人はご自慢であろうが、食べさせられるほうはたまらない。ナイフとフォークで洋菓子の腑分けをする羽目になる。

プログラミングの場合も同じことが言える。プログラム・コードには、インタフェースが二つある。プログラム・コードは、最終的には機械が読むのであるから、機械との間に一つの界面をもつのは当然である (コンパイラの視点)。しかし、それはまた、人が読むためのものでもある。プログラミング中は絶えず、それを書き読む。完成後も、改訂や修正、流用のために読み書く。プログラム・コードは、人との間にも界面をもっているのである (プログラム言語の視点)。こう言うと、「自分しか使わないから」という反論が聞こえてきそうだが、試みに一カ月前の自分のプログラム・コードを読んでもみるとよい。何の抵抗もなく読めることは稀である。他人事ではない、「明日は我が身」なのである。プログラム言語は、元来、人がプログラムを書くために考案されたものである。それをわざわざ分かりにくく書くこともあるまい³⁾。

良くできたプログラムは、見た目も綺麗なものである (もちろん逆は真ならず。これは人の外見と同じである)。段付けや変数名が処理内容に合っていて、解くべき問題とよく整合している、プログラム・コードからディテールを除いたスケルトンだけでプログラムの構造が読み取れる、などなど。たとえば、手続き名 `MAESYORI 1`, `MAESHORI 2` と `INIT-FOR-TOTAL`, `INIT-FOR-GRAND-TOTAL` とでは、どちらが見やすいか、わざわざ一考するにも及ばない⁴⁾。

話は変わるが、ベーコン・エッグという料理が

ある。言わずと知れた、半熟の目玉焼きにベーコンを添えた料理である。この目玉焼きを食べるのに、まず周りの白身を攻め落とし、残る黄身を人目を盗んで一気に頬張る若い女性がいるそうである。もしコックがそのような食べ方を期待したとすれば、彼は料理人失格と言うべきであろう。食べにくい料理は、料理ではないのだ。しかし、目玉焼きについて言えば、黄身はソースやベーコンなどに絡めて食べるものである。黄身とソースを厨房で混ぜてしまうのではなく、食べるそのときに自然に混ざる（言わば、調理の最後のステップである）ように仕組まれているのである。実に見事な構成だ。一方、このとき黄身を丸ごと拘うためのスプーンを用意したとしたら、食べやすくなるかも知れないが、料理の価値はなくなることだろう。（ちなみに、茹で卵はスプーンで食べる。）

「プログラムが、解くべき問題とよく整合している」というのは、このように両者の構造が隙間なく一致していることを指している。このようなプログラムを「構造的プログラム」と呼ぶ。これに対して、goto文（フォーク）をdo while文（スプーン）に代えただけのプログラムは、「構造化プログラム」と言えようか。

3. 本格派スパゲッティ

パスタを茹でるときは、必ず大きな鍋で茹でる。重さでいうと、パスタの量の十数倍の水が必要である。この水は、もちろん、食卓に上ることではないが、この大量の水がなければ、美味しく茹でることはできない。パスタを入れる前から、そして茹でているときも、必ず沸騰させておく。スパゲッティを入れるときは、折れないように注意する：「論文は短く、スパゲッティは長く！」である。茹で加減は、「噛んで少し固さの残る程度」が適当であり、茹で上がった後、すぐに水を切る。このとき、冷水をかけて冷ましてはならない。冷めないうちに、具を絡めて（このとき具に火が通る）、食卓に出す。こうすれば、スパゲッティが絡まることはない。皿は少し大きめのほうがよい。小さな皿に山盛りするのは、見苦しい。料理にも適度の余白が必要なのである。

プログラミングという作業の中で、コーディングは最後の一瞬に過ぎない。目に見えるコードの一つ一つの裏には、莫大な量の考察があり、その

ほとんどは検討され捨て去られ後には残らない。問題をよく吟味し、その構造を理解し、それに沿ってアルゴリズムを構成する。コーディングは、その後の、最後の仕上げ（盛りつけ）である。アルゴリズムを写すのに精確であるのは当然として、それは見やすくなければならない。ここでも適度な余白（段付けや変数のネーミング法）が必要なのである。

4. あとがき

料理になぞらえて、プログラミングについての私見を述べた。私見とはいっても、その多くは、過去に繰り返し語られたことであり、現在の研究者の常識と大筋で一致していると思う。ただ、残念なことに、こうしたプログラミングの理念の届かぬ場所が今でも見受けられる。そのような場所に、プログラミング教育についての話題を提供できれば本稿の目的は達したことになる。

最後に、本稿を読んで「不真面目である」と思われた生真面目過ぎる方に三言申し述べて終わりにしたい。

2.～3. でみたように、スパゲッティの調理とプログラミングには、構造の一致がみられる。これを読み取ることは、構造的プログラミングの要諦の一つである。具体のみをみて、構造をみずに、構造的プログラミングが書けるわけがない。

面白いことに、スパゲッティ (spaghetti) はコード (cord) を意味する。原語では異なるが、片仮名にすると、プログラム・コードのコード (code) と、偶然にも、一致する。それに、コード (cord) は、紐の意であるから、手続き型言語のコードの形態を指していると言えぬこともない。

また、よく知られているように、プログラム (program) は pro (before) + gram (write) であるから、ストアード・プログラム方式の概念を実によく表している。一方、調理の手順は、レシピ (recipe) と呼ばれ、receive との類似からも知れるように、take という意味のラテン語に由来する。解くべき問題を良く理解 (take) することから始まるプログラミングでは、プログラムはレシピと呼ばれるほうが適切なのかも知れない。

参 考 文 献

- 1) 高橋延匡：計算機科学の温故知新，電子情報通信学会昭和63年度生涯教育講座Cコースソフトウェア技術資料（1988）。
- 2) 小林孝次郎：計算可能性入門，近代科学社，東京（1980）。
- 3) Price, D., 竹市正人，角田博保訳：Pascal：思いやりプログラミング第2版，近代科学社，東京（1988）。
- 4) 小林 修：初等プログラミング教育の視点，情報処理学会計算機教育シンポジウム報告書，pp. 27-38（1991）。

（平成5年1月14日受付）



小林 修（正会員）

1954年生。1976年東京工業大学理学部物理学科卒業。1978年東北大学大学院博士課程前期修了。理学修士。1987年産能短期大学入職。1988年～1993年産能短期大学講師。情報処理教育，教育工学に関心をもつ。電子情報通信学会，教育工学会各会員。

