



はじめてのプログラミング言語

田中二郎

筑波大学 電子・情報工学系

情報系の大学に入ってくる学生に、プログラミング言語として何を最初に教えたらよいか？ また、これからプログラミング言語はどのように変わっていくか、また変わっていかねばならないかについて論じる。

最初に教えるプログラミング言語

今もって世界で最も多く使われているプログラミング言語はCOBOLとFORTRANだそうである。計算の性能が極限まで要求される数値計算や科学技術計算の分野ではFORTRANが主流である。FORTRANはおそらく今後100年たっても世界最高のプログラミング言語であり続けるのだろう。そうだとすれば、FORTRANを最初に教えるというのも1つの考え方であろう。いかにも古すぎる（なにしろ構文解析という概念もないのである）ということを除けば、FORTRANは、構文が簡潔にして効率的な言語である。一方、COBOLの方であるが、これは事務処理用のプログラミング言語である。構文も多様であり、言語処理系も巨大であるので最初に教える教育用のプログラミング言語としては不適切かもしれない。

歴史的に見た場合、大学生に教える教育用のプログラミング言語としては、FORTRAN, Basic, Algol, Pascalなどを教えた例が多かったのではないだろうか。FORTRANは、要するに使われているから教えた、ほかに適切な言語がなかったから教えたというスタンスであろう。また、Basicはパソコンなどで

よく使われていた言語だからであろう。Algol, Pascalなどの場合には、アルゴリズムの記述などとの親和性がよい点が買われたと思われる。

現在、大学などではプログラミング言語Cを教える場合が多い。筆者の属する筑波大学でも、従来はPascalを教えていたが4年ほど前からCに切り替えた。Cを教える理由としては、この手の手続き型言語としては現れたのが比較的新しく（1973年）、言語デザインと実装技術の調和もとれていること、ポインタなどの構文要素を用いて比較的低レベルの記述が可能なので、ワークステーション上のオペレーティングシステムであるUNIXとの相性もよく、プログラムやデータ構造、システムプログラムなどについても教えやすいなどの点が挙げられよう。

一部に、最初から高級言語を教えるのではなく、まずはアセンブラ言語から教えたらどうかという考え方もあるだろう。残念ながらこうした試みが成功したとは聞かない。アセンブラ言語を教えることには、プログラミングを教えるというよりも計算機の仕組みを教える、計算機アーキテクチャを理解させるという側面が強いように思われる。

また、1980年代には、MITでは大学

1年生にLispの一種であるSchemeを教えるなどの試みを行ったり、日本でも、大学生の一般情報処理でPrologを教えるなどの試みを行ったが、プログラミング教育としてあまり成功したとはいえない。LispやPrologも当初思われていたような万能なプログラミング言語ではなく、人工知能や自然言語処理などの特定分野で使われるにとどまっている。

プログラミング教育の重要性

最近では、コンピュータ教育を全学的に「必修」として導入する大学が多くなった。プログラミング言語を教えるのではなく、文房具としてのコンピュータの使い方を教える「コンピュータ・リテラシー」教育がなされるようになった。読み書きソロバンに対応する計算機の使い方、すなわち文書処理ソフト、プレゼンツール、表計算ソフト、電子メール、Webページなどについて教えるのである。ここでは、プログラミング言語については教えられないことがない。一般教育としては、これでいいのかもしれないが、将来プロとなる情報系の学生がこれでは情けない。

コンピュータがコンピュータである

ことの原点はプログラムを内蔵することができるところにあり、プログラミング言語は単に何かを記述するための道具ではなく、コンピュータをしてコンピュータたらしめる本質である。大学で情報の学生にプログラミング言語について何を教えるべきかという議論を行ったとき、「言語は道具だから何でもよい」という乱暴な議論を聞き哑然としたが、考えてみるとこれまで日本においてプログラミングは不当に低く扱われてきたのではないだろうか？

もちろん大学が専門学校でないことは存しているが、情報学科を卒業してプログラミングができない学生がいたとするとその学生を採用した企業はだまされたと感じるのではないだろうか。これは、音楽大学のピアノ科を卒業してピアノを弾けない学生がいたらおかしいのと同じである。残念ながら、現在の日本の大学では、情報学科を卒業しても、コンピュータがほとんど使え、まあまあプログラムがかけるのは半分強というところであろうか？

なぜこのような事態が起きるかといえば、原因は実学の軽視とプログラミング言語に関する評価の低さ、考え違いである。日本の大学への愚痴を書いてもきりがないが、日本の大学は小手先の学科の増築や改組を重ねており、比較的歴史の新しい情報関係学科は、そのあおりをうけて大学1年から大学院まで一貫して情報教育を行う体制にない。情報学科とは名ばかりで、内実は他分野の教官が多かったり、さまざまな分野を広くカバーするために基幹となるべきプログラミング言語やソフトウェア分野の内容が貧弱であったりする。また、教育においても教養的な独りよがりの講義やスキルを身につけさせるための手間ヒマを省略してきたのではないだろうか？

産業界もプログラミングに こだわってほしい

産業界ではどうかというと、ここでも実学の軽視が問題を起こしている。日本の会社は終身雇用を理想としており、採用にあたっては即戦的な技能よ

りも大学名を重視する傾向がある。

筆者は、筑波大学に来る前に企業の研究所に勤めていたが、そこで発見したことは、学生の修士時代に良い研究を行い良い論文を書いていた者は入社してから良い仕事をするということである。有名大学の出身者であっても学生時代にぱっとしなかった学生は会社に入っても芽が出ないケースが多い。また、三つ子の魂百までというが、学生時代の専攻により、その後の研究的な興味限定されてしまうケースが多い。たとえば情報系でソフトウェアの研究を行っていた者は会社でも良い仕事をする可能性が高いが、数学、制御などの近隣分野を専攻していた者は、結局、ソフトウェア系の仕事に馴染めないケースも多い。

日本の会社員はプログラムを自ら書かない。実際に情報系の学科でプログラミングを学んだことのない年寄りがプログラムを書けないのはしかたないとしても、若い社員もプログラムを書けない。では、誰がプログラムを書くかという関連会社のプログラマである（これをソフト外注という）。プログラムを書かない／書けなくてどうやってシステム設計ができるのであろうか？ 産業界はもっとプログラミングという実際のスキルにこだわってほしい。筆者は、ソフトウェアを書けない学生は会社の技術部門には採用すべきでないと思っている。また、ソフト外注にはいろいろな理由があることは存しているが、外注してソフトを作ろうとしても外注先にはそのソフトに賭ける熱い意気込みは伝わらず、結果としてロクなソフトにならないことが多い。外注ばかりしているからロクなソフトができないのである。ソフト外注は会社の空洞化を進行させるだけである。人員や開発期間の制約が強い場合には、外注するのではなく、よりプロジェクトを絞り込むことが必要なのである。

教える側の論理と産業界の論理

コンピュータサイエンスが他の科学と異なる点は、現実の産業界からフィ

ードバックがある点であると思われる。米国や日本の研究者は幸せである。国内にコンピュータ関連産業があり、それらを通じて研究者のアイデアを実現するチャンスがあるからである。ヨーロッパの場合、必ずしもそうはいかない。米国のコンピュータサイエンス研究を見て愕然とするのは、非常に「実用的」な研究を行っている点である。また研究分野も非常に産業界のニーズに適合している。

プログラミング言語についても、コンピュータサイエンスを実用的な科学ととらえる限り、教える側の論理と産業界の論理を整合することが重要となると思われる。従来、プログラミング言語の選択にあたっては、教える側の論理が優先されていた感がある。教養主義のみに基づいてプログラミング言語の選択を行うのでは必ずしも試みは成功しない。一方、産業界の論理についても、プログラミングをどうとらえ、どのような人材を求めるのが明確にする必要があろう。

今後教えたいプログラミング 言語は

現在、大学で最初のプログラミング言語としてCを教えるのは、それなりに意味があると思われる。しかしながら、今後ずっとCを教え続けていけばいいかとなると、これは明らかにそうではない。

教えるべきプログラミング言語も、今後のコンピュータのプラットフォームや計算機環境の発展、コンピュータサイエンスの発展とともに変化していかなければならない。筆者が今後教えたいプログラミング言語は、時代順にいうと、その1 Java、その2 スクリプト言語、その3 ビジュアルプログラミング言語ということになる。

(1) Java

最近におけるプログラミング言語の最も大きな変化とは、GUIの隆盛、Webブラウザなどのプラットフォーム非依存環境の登場、Javaなどの分散計算環

境の登場，ということになるだろうか。

したがって，ここ数年の範囲で考えた場合，最初に教えるべきプログラミング言語はJavaであろう。理由は，分散環境，Webブラウザとの相性がよい，オブジェクト指向言語としてすっきりとまとまっているなどである。また，最近のプログラミング言語だけあり，リフレクションなどの最近の研究成果もそれなりに取り込まれている。

(2) スクリプト言語

最近，GUI環境が広く使われるようになったが，一方でGUIが既存のプログラミングを難しくしている。マッキントッシュの流行以来，ユーザが自らプログラムを書くことはほとんどなくなった。UNIX環境上でGUIを使うためには，たとえばC言語で記述したプログラムにMotif等でGUI部分を付加する必要があるが，これは初心者にとっては難しい。

そこで筆者が着目しているのがスクリプト言語である。スクリプト言語とは，コマンドやマクロにより手順を記述することにより処理を行うプログラミング言語のことで，たとえばsed, awk, perl, Tclなどが知られている。こうしたスクリプト言語にGUIを結合させたものとしてTcl/Tk, perl/Tkなどが知られており，これらの言語を用いることによりXの壁を乗り越えることができる。現在，さまざまなスクリプト言語が作られており，たとえばVBScriptやJavaScriptなどの言語はWebページに組み込んで使用できるようになっている。

スクリプト言語は通常のプログラミング言語より高機能であり，簡潔にプログラムを記述することができる。それゆえにスクリプト言語はラピッドプロトタイプ用の言語としても優れており，実際に使いでのあるプログラミング言語である。

(3) ビジュアルプログラミング言語

ビジュアルプログラミング (VP) は，従来の文字列や記号に代わって，アイ

コン，図形，アニメーションなどを用いてプログラミングを行う技術である。まだ真の意味で実用的なVP言語は現れていないが，近い将来こうしたVP言語が既存のスクリプト言語にとって代わることと思う。

プログラミングのコミュニティでは図式によるプログラムの表現に違和感を持つ人々も多いかもしれないが，ソフトウェア設計の分野においてはOMTの各種の設計図やデザインパターンなどでプログラムの図式表現がすでに多用されている。

やっぱり

Pictorial Programming?

いささか手前味噌になるが，筆者はすでに10年以上前からVPに注目し，PP (Pictorial Programming) に関してさまざまな研究を行ってきた。最近ではPPの3次元化を図り，スケーラビリティへの対応，リアリティの追求，3次元レイアウトなどを研究の要素技術として，3D-PPと呼ぶバージョンの開発を開始したばかりである。果たして，我々のPPが世の中に飛ばたく日は来るのだろうか。

(1998.09.20)



プログラミング言語の書き手と読み手

阿草清滋

名古屋大学工学部情報工学科

筑

波大学の田中二郎先生の「はじめてのプログラミング言語」を興味深く読ませていただきました。田中先生と同じように，大学でソフトウェア工学を研究し，プログラミングを教えるものとしてコメントさせていただきます。

大学で最初に教える言語の議論がありますが，これは計算機科学科や情報工学科の学生に対する教育の議論として考えますと，記述の側面のみが論じられているのではないかと思います。実際のプログラミングでは人のプログラムを読まざるを得ないことが多くあり，プログラムの読み方の教育も大切と考えます。これはあまり大学では行われておりません。レビューの教育も行われていません。かなり以前になりますが，当時東京工業大学にみえた木村泉先生が学生にプログラム保守の実験を課したことがプログラム教育に有効であったとお話しされていらっ

いました。計算機科学はある意味で記述の学問であり，記述言語の選択はもちろん重要な要素ですが，書き方教育に偏りすぎ，読み方の教育がおざなりにされてきたと思います。

1980年代の初めにOS360開発者の1人であるWulfはプログラミング言語の持つ機能は，計算機への指示，プログラムの設計支援，そして保守作業とのコミュニケーションであると主張しました。人に読んでもらえるプログラムというのが大切と考えます。この意味ではC言語は書きやすさというが簡便さに触れていて必ずしも読みやすさを考えた言語ではないと思います。

プログラミング言語は背景の計算機ハードウェア，OSなどに影響されます。言語は文化の担い手であり，軍事力・経済力のある国の言語が世界に広まります。同様に計算機社会においてもハードウェアやOSの市場占有率の大きさがプログラミング言語にも影響し

ていると思います。現在、我々の分野で使われる環境はUNIXベースのワークステーションであり、C言語が中心となっています。しかし、PCの台頭も目覚ましいものがあります。どちらにせよ環境はGUIベースのものとなっています。田中先生はGUIの専門家であり、その立場からMotifやMSFCを理解してのプログラミングの敷居の高さからスクリプト言語を推奨されていますが、GUI部と演算処理のロジックをうまく分離した環境も考えられます。GUIが不可避な時代においてはプログラミングの教育にもそれを前提とすべきで、MacでのHyperTalkのような優れたものもあります。ただ、背景となる計算機が問題ではありませんが。

スクリプト言語は人間の計算機操作を記述するものと考え、自分のやっている操作をきっちりと記述するという訓練にはいい言語かと思います。しかし、言語としてきれいなものは少ないように感じます。たとえば何でもありのperlのような言語を最初に教えたいとは思えません。WebのCGIの記述に広く用いられていますが、簡便で強力であればいいというものでもないと思います。

私は、プログラミングの授業では構造化や抽象化の考えを教えています。概念は確かに言語に縛られるものではありません。特に、計算機科学を学ぶ人間にとってはそうであってほしいと思います。しかし、言語は最初に習ったものになりに影響されることも事実です。それゆえ最初に学ぶ言語の選択は重要です。私の書くCのプログラムはFORTRAN的だと学生によく言われました。データ構造として配列しか使わなかったからです。行いたい処理を単に書いただけなら慣れた言語が一番です。それを読まされる人がいることを考えると、個人の趣味や慣れで言語を決めるのではなく、社会の流れをよく見極める必要があります。言語は文化なのですから。

(1998.09.23)

言語はやはり道具でしょう、 教養はやはり必要でしょう

柴山悦哉

東京工業大学 情報理工学研究所

ビジュアルプログラミング言語(VPL)が本当に役に立つと考えているプログラミング言語研究者は少ない。これは、田中先生が指摘しておられるとおりである。私自身、今こそVPLが使い物になると考えているが、昔はおもちゃだと思っていた。したがって、みんなの気持ちもよく分かる。しかし今では、田中先生と私は非常に近い立場であり、我々は、日本におけるVPLエバンジェリストとして活動している(本当かなあ?)。なんだか反論係としてあまり適任でないような気がするが、今回はあえて悪役を引き受けることにした。

誰にとっての「初めて」か?

さて、今回のテーマであるが、編集担当殿からは「初めてのプログラミング言語は〇〇がよい」であると連絡を受けた。一瞬、〇〇の部分がOO(オブジェクト指向)に見えてしまった私であったが、まあそれはどうでもよい。ともかく、最初の反論は、このテーマの解釈に向けたい。

高校でプログラミングを教えようというこの時代に、「初めてのプログラミング言語」といえば、遅くとも高校、へたすると小学校が幼稚園の問題であろう。正直なところ、幼児教育のことはさっぱり分からないし、幼稚園児や小学生向け言語の話になるとマズイなあ…と内心恐れていた。Cocoa, ToonTalk, Visual AgenTalkあたりのビジ

ジュアル言語を取り上げて煙に巻くという作戦も、相手がビジュアル系の田中先生では通用しない可能性が高いと思ったのだ。

しかし、これはまったくの杞憂であった。まさか、大学で初めて学ぶ言語の話になろうとは…。今の時代に論点がズレてやしませんか? 大学以前の教育内容を吟味することなく、大学で最初に教える言語について語ってよいものですかね? これが第一印象である。

まあ、この点に関しては、編集者側を責めるべきかもしれない。元のテーマがあまりにも漠然としており、誰にとつての初めての言語なのか定かでなかったのだ。しかも、反論を頼まれた段階では、テーマと最初の著者しか決まっておらず、原稿の論旨は何も分からないという仕掛けになっていた。そのくせ、反論らしい反論を書くように要求された。

しょうがないから、第1の反論は会誌の編集方針に向けることにしたい。だいたい、自分は安全地帯にいて、人々に無益な争いを要求するという態度がそもそも気に入らない。卑怯者め! ウーム、この程度の反論では、満足してもらえないかな?

Javaはたしかに悪くない、
しかし…

さて、田中説に真面目に反論するためには、同じ土俵にのぼらなければならぬ。10歩くらい譲って、大学の情

報系で最初に学ぶ言語について考えてみることにする。以降、「最初の言語」とは「情報系の大学で最初に教える／学ぶ言語」という意味に限定して考えることにする。

田中説によると、「時代順にいて、その1 Java, その2 スクリプト言語, その3 ビジュアルプログラミング言語」が良いらしい。一見もっともらしい取り合わせだが、「時代順」ということは、現在の時点で望ましい順番ではない。では、時代とともにこの順番で変わっていくということか? 「ここ数年の範囲で考えた場合、最初に教えるべきプログラミング言語としてJavaを挙げておく」と書いてあるので、現在は「その1」の段階のように読める。しかし、Tel/TkはJavaより古いし、現在、すでに多数のスクリプト言語が実用的に使われている。なぜ、ここ数年はスクリプト言語よりJavaが良いのだろうか? また、ここ数年が過ぎると、いったいスクリプト言語がどのように変化して、Javaより望ましい言語となるのだろうか? 私にはさっぱり分からないのであった。

しょうがないので、まずは、最初の言語としてのJavaについて少し見解を述べよう。実は、私が演習科目を担当している東京工業大学理学部情報科学科では、学科所属直後の2年前期にJavaも教えている。「Javaを」ではなく「Javaも」と書いたのは、Schemeも教えているからである。Javaは使える言語として教えており、Schemeは教育用の言語として教えている。よく言えば、実学と教養のバランスに配慮しているわけである。

たしかにJavaは悪くない。学生は新しいものが好きだし、絵が動くに興味を持ってくれる。動機づけは非常に重要であり、これは大きなポイントとなる。しかし、Javaのような主流言語には重大な落とし穴がある。最初にJavaを学んでしまうと、他のマイナーな言語をわざわざ学ぼうという気力が失せてしまう恐れがある。学ぶ側には学ぶ側の論理があって、多くの学生は怠け者であり、なるべく楽をしたいと考えている。自然言語との類推で考えても、

最初に英語を学んでしまうと、他の外国語を学ぶ気力が失せるだろう。これと似た症状と言えよう。

逆に言うと、最初にマイナーな言語を学んでしまうと、主流言語はどうせ後から覚えざるを得ないので、複数の言語をちゃんと学ぶ確率が高くなる。以上のような理由から、Javaを学ぶのは少し後にした方がよいように思う。しかし、これを実践するのはなかなか難しい。情報系の学生に、主流言語の存在を隠し通せるものではない。マルチリンガルの重要性を言い含めながら、慎重に授業を進める必要がある。

ここで余談を少々。学生時代の友人で文学部言語学科に進学した人がいた。伝聞にすぎないが、学科所属の直後に8カ国語を学んでいたそうである。大学の普通の学科なら、外国語として英語を学べば十分な場合が多い。しかし、言語学のプロを目指す人々なら、そんな態度は許されないわけである。言語学の場合、さすがに学生にも自覚があるのだろう。一方、情報系の場合、そのディシプリンが学生に十分に浸透しているとはいいがたい。そのため、万能言語を1つ覚えればそれで済むという幻想を持ってしまう場合が多い。仮にも情報系の大学教育を受けた人物なら、「プログラミング言語はJavaしか使えません」という発言がいかにか恥ずかしいものであるかを、気づいてほしいものである。

自然言語でも人工言語でも、複数の言語を学ぶことにより、その言語（さらには言語の背景にある文化）を相対化してとらえる視点を獲得することができる。だから、Javaを教える／学ぶことに異論はないが、Javaだけしか教えない／学ばないのは問題外と言えよう。問題に応じて適切な言語を使い分ける能力は重要である。

言語はやはり道具である

さて、本文中でどうにも納得しかねるのが、「プログラミング言語は単に何かを記述するための道具ではなく、コンピュータをコンピュータたらしめる本質である」というくだりである。プ

ログラミング言語が道具ではなく本質だという主張は変だと思う。実学の軽視を問題にするなら、「本質」なんて言ってないで、プログラミング言語という道具の軽視を問題にして欲しかった。仮に、プログラミング言語がコンピュータをコンピュータたらしめる本質であるとしても、どうせコンピュータだって道具にすぎない。

それに本文を読んでいて思うのだが、Javaやスクリプト言語を重視する理由は、要するに道具として便利だからではないのですか? たしかにGUIを手軽に記述するのにスクリプト言語は便利だが、スクリプト言語のどのあたりに、コンピュータをコンピュータたらしめる本質があるのだろうか? しかも、スクリプト言語を使うことにより、他の言語より効果的に教えられる本質って何?

たぶん、私だったらこう言うだろう、「プログラミング言語は単に何かを記述するための道具である。しかし、それは、人間をプログラマたらしめる本質的な道具である」と。高級なプログラミング言語は、どちらかという人間のために設計されたものである。もし本質的な何かがあるとすれば、それは人間との関係で語るべきものであろう。

人間は、図形的イメージを含めた広い意味での言語・記号を使わずに、高度な思考などできない生き物である。自然言語は何かを表現する道具に過ぎないが、人間を人間たらしめている本質的な道具でもある。同様に、プログラミング言語はプログラマが思考・表現するために必須の道具なのである。プログラミング言語を学ぶということは、単に、言語の文法や意味を学ぶだけではすまない。イディオムやパターンを使いこなす、思考・表現ができるようになるための技術を学ばねばならない。

また、プログラミング言語は人間同士のコミュニケーションのメディアでもある。そうでなければ、プログラミング言語がコメントをサポートする必要などないし、プログラマが分かりやすい識別子を選ぶ必要もない。また、

インデントもいらないだろう。(未来や過去の自分も含め) 他人とのコミュニケーションのためにもプログラミング言語は必須のものである。

さて、だんだん教養主義者の正体が現れ始めてきたような気がする。そろそろ、2年生向けにScheme言語を教えている理由を話そう。別に言語自体を教えたいわけではない。H. AbelsonとG. Sussmanが著したStructure and Interpretation of Computer Programs (MIT Press, 2nd Editionが1996年出版)のように、プログラミングにまつわる諸概念を教えようとして、たまたまSchemeを使っているだけである。手続き抽象、高階手続きに基づく部品化、データ抽象、メタインタプリタなどを中心に、さまざまな抽象化手法、変更に強いプログラミング手法、計算モデルの拡張法などを教えている。ついでにSTk (Tk付きScheme) も教えているので、スクリプト言語の雰囲気も伝えていることになる。

何が一番重要か

結局のところ、田中先生は、即戦力を重視しておられるようだ。また、MITでの教育を「プログラミング教育としてあまり成功したとはいえない」と評価しておられる様子から察するに、Structure and Interpretation of Computer Programsのようなプログラミングに関する教養書をあまり評価しておられないようだ。一方、私は、変化に適応する能力を重視する。我々をとりまく環境の変化は他の業界よりかなり速い。変化への対応能力が、実学の大前提となる。そして、変化に耐えるためにはプログラミングの教養は必須であると考えている。

人間の性として、20歳くらいまでの段階で、ある教養にどっぷりと漬かってしまうと、視野が狭くなり、他の教養を受け付けなくなる傾向がある。これは、宗教や思想の場合に顕著だが、プログラミング言語、OS、エディタなどについても似たようなことが言える。そして、新しい時代への変化が目前に

迫っても、それを無視しようとして対応が遅れる。最悪の場合、変化にまったく対応できない可能性もある。やはり若いうちは視野を広めるように努めるべきであり、どんなに良い言語でも1つだけにこだわってはいけない。一見無駄に見える教養が、実は重要なのである。

さて、編集者からのリクエストに応じて、反論を書きました。いかがでしょうか? でも、書いていて正直むなしくなりました。田中先生は、現

在の日本におけるプログラミング教育に対して強い不満と危惧を持っておられるわけです。それは私も同意見です。一方、プログラミング教育の一担当者として、「俺様はそんな教育などしていません!」と、私はちらっと思ったわけです。ただそれだけのことだと思いません。普段なら反論を書こうなんて思いません。だいたいですね、こんなヤラセの論争で、研究者の本音が引き出せるわけないですよ~だ:P

(1998.09.25)



将来はビジュアル言語?

田中二郎

筑波大学 電子・情報工学系

阿草先生から、プログラミング言語を考えるとき、書き手の都合だけではなく、レビューしやすい(あるいは保守のしやすい)プログラミング言語を考える必要があるとの意見をいただきました。現状のプログラミング言語教育を見た場合、書き方を教えるだけで終わってしまうようです。書きっぱなしでなく、それを添削したり、よいソースコードを読ませる教育が必要であると感じました。また、スクリプト言語について、言語としてきれいなものは少ない、perlのようななんでもありの言語は教える気がしない、と阿草先生は言われています。Perl/Tkについても、言語としてなっていない、とても教える気がしない、などの声をよく聞きますが、こうした意見は当たっているのでしょうか?

また、大野さんは、プログラミング言語に「はまる」ことを重視されているようです(大野さんの意見は下記URLを参照してください)。確かに「は

まる」ことは原点でもありますが、昔と違い学問の蓄積した今では独学では床屋の将棋で、プログラミング言語についても筋の良い教育を効率良く施す必要があります。

たしかにスクリプト言語は、従来のプログラミング言語の感覚からすると違和感を感じるかもしれません。しかしながら、これらの言語を用いる機会は今後ますます増えていくと思われるし、これからさまざまな言語のスクリプト化が進行していくことも予想されます。

柴山先生からは、ビジュアル言語について積極的な支持の言葉を頂戴しました。まだ世の中にはビジュアル言語について懐疑的な方も多いとは思いますが、世の中も変わる時には早く、パラダイムシフトが急速に進行します。阿草先生の「言語とは文化である」とは名言ですが、文化についても既存の価値観にとらわれず新しいものを積極的に採り入れていくべきであると感じています。(1998.10.12)

～ 議論の続きは、次のURLをご覧ください。 <http://www.ipsj.or.jp/magazine/interessay.html> ～