

文法最小化を目指した日本語プログラミング言語 「敷島」の構想

大岩 元^{1,a)} 中鉢欣秀^{2,b)}

概要：プログラミング言語は、巨大なプログラムを書く上での困難を解決する目的で発展を遂げてきた。この結果、最終的に書かれるプログラム自体の表現水準があがり、巨大なプログラムを書く必要がなくなったことから、今後は問題をかかえた人達が直接自分でプログラムを書く時代が来ることが予想される。その際には、プログラム作成より問題自体の本質的な困難の解決が重要になり、言語自体が理解し易く、学習が容易になることが求められる。日本語は少数の語彙と簡潔な文法で複雑で深いことを表わすことができるので、この性質をプログラミング言語で追求し、だれでも簡単に学習できる日本語プログラミング言語「敷島」を構想した。当面の目標として、小学生がプログラミングの本質を学べる「しきしま1号」を開発する。

キーワード：プログラミング言語，日本語プログラミング，敷島

1. はじめに

プログラミングに関して日本では、専門家が使えればよいだけで、一般人が学ぶ必要はないと考えられてきた。この考えは、パソコンソフトが提供されるようになり、Basicでプログラムを書かなくてもパソコンが使えるようになってから一般化した。この点は欧米でも同じであったが、大学進学者については、プログラミングを中心とする情報教育が行なわれてきた。このことは、ユネスコが主として途上国に対して行った情報教育に関する提言の120ページに記載されている [1]。

英国や米国では、大学で情報専門学科に進学する学生に対しては、日本の大学の専門学科で教え

られる内容乃至、それ以上の高度な内容が、中等教育段階で教えられる仕組がすでに1990年代から確立している。例えば、米国では大学の初年次の授業を高校で受講して、その単位が大学入学後に認められるAdvanced Placement(AP)の制度が広く行なわれているが、プログラミングに関しては、オブジェクト指向のプログラミングの方法論(methodology)が教えられている。Wikipediaによると [2]、かつてこの科目はもっとformalなもので、アルゴリズムとデータ構造、データ抽象まで含むものであったが、受講する学生が少ないために2008-9年に難しい内容が廃止された。例えば、ソートについてマージソートは教えられるが、クイックソートやヒープソートはあつかわれなかったことになった。しかし、プログラムの正当性について、Precondition and postcondition, Assertionsは今でも扱われており、廃止されたのはInvariants

¹ 慶応義塾大学

² 産業技術大学院大学

a) ohiwa@sfc.keio.ac.jp

b) yc@aiit.ac.jp

であるというように、日本の企業内教育ではほとんど教えられない内容を高校生が学べるようになってきている。要するに、問題解決のためのオブジェクト指向概念が教えられているのであり、問題の分析と設計に重点が置かれている。

英国式の教育を行っているオーストラリアでは、専門学科に進学する学生は、プログラム設計に関しては構造化設計を中等教育で習得済として、オブジェクト指向設計から教育が始まるそうだ。専門学科でも Java の記述法しか教えられていない日本の大学や IT 企業の大部分の教育内容とは大きな違いがある。

しかし、それにもかかわらずアマゾン、グーグル、フェイスブックなどの先端企業では、高度な IT 技術者の獲得が社運を決める状況にあり、IT 技術者の能力が国の経済を左右する状況が続いている。また、高度に発達したソフトウェア製品を使いこなすには、利用者側にも論理思考が要求され、その育成が問題となる。最近の OECD による生徒や成人に対する学力評価 (PISA と PIAAC) が求める問題解決能力は、こうした状況を反映したものである。

こうした中で、英・米両国では国の方針として、小学生からプログラミング教育を行なうことが決定された。これを受けて、日本でも、2014 年 5 月の閣議決定で初等・中等教育でプログラミング教育を行なうことが明示された。

2. 母語によるプログラミング

慶応大学湘南藤沢キャンパス (SFC) では、1990 年の開学以来、外国語教育、保健体育とともに 3 つだけの必修科目の 1 つとして情報科目が 8 単位、全学生に義務づけられてきた。現在も、環境情報学部は 8 単位必修であるが、外国語科目が軽減されている。総合政策学部は逆に情報科目が軽減され、外国語科目の 8 単位が必修となっている。情報科目の内容はプログラミングを中心とするものであるが、全員が自分の目的を実現できるプログラミング能力を獲得出来たわけではなかった。

そうした状況の中で、プログラミング言語を日本語にすることで、クラス全員が挿入ソートをス

クラッチから作れるようになった [3]。この授業では、週 90 分 2 コマの授業を 13 週にわたって行ったが、その前半に「ことだま on Squeak」[4] による日本語プログラミング教育を行ったことによって達成された。授業の後半は、前半の概要を Java で繰り返すことによって、実用言語においても、日本語プログラミングの授業で獲得された論理構築能力が活かされることが分かった。

従来は、授業全体を Java (や C) で行ってきたために、90 年代後半以後は、自分の目的を実現できるプログラミング能力を獲得する学生が減少して、出来るのは入学前からその能力を持っている学生だけに、ほぼなくなってしまっていた。これは、いわゆる学力低下の反映であろう。

実用言語を使って初心者にはプログラミングを教えると、

- (1) 「新しい言語を理解すること」
- (2) 「それを使って仕組を作ること」

という 2 つの初めての作業を同時に行なわなければならない。殆どの受講者は、新しく学ぶ言語の文法通り、プログラムを書く段階で挫折してしまう。結局、授業は文法通り書くことの訓練で終わってしまう場合が多い。

実用言語は実用目的を達成することが目的の、専門家を対象として言語であって、実用プログラム作成を効率的に行なう工夫が成されている。これらの工夫を避けて初心者が使えるようになっていけばよいのであるが、そうではない。このため、初心者にとって意味のない作業を強いられることになる。

もう一つの問題は、実用言語を将来使う可能性である。一般教育の受講者で、将来プログラマーになる人は少数であり、今後はさらに高度な技術者だけが生き残る時代になってきたために、プログラマー自体の人数が減っていくことが予想される。従って、現在の実用言語を学習者が使う可能性は小さい [5]。

一方で、論理思考が必要とされる人数はこれから増大していくことが予想される。情報技術の応用分野がますます広がるからである。コンピュータの導入で、社会生活で使う情報システムが複雑な

作業を行なうようになり、使い方を丸暗記するのでは対応できなくなって、論理的に考えて使う必要性が高まることが予想される。こうした論理思考能力を育成する手段として、プログラムを作ってみる体験は大変有効である。

論理思考教育を行なおうとすると、プログラミング活動の中で、アルゴリズム構築が一番役に立つ部分である。従来の授業では、ここに到達する前に、文法理解で終わってしまっていたために、社会に出て役に立つ教育が行なわれてこなかったのである。

アルゴリズム構築の教育においては、従来は疑似コードが使われ、それは母語としての日本語が用いられた。疑似コードは実行できないので、それを実用言語に翻訳しなければならない。すると、文法エラーを退治できたとしても、今度は実行時エラーを退治する必要がある。それが終わると、予期したようには動作するとは限らない。これを行なおうとすると、慣れない実用言語を正確に解釈しなければならないのであるが、これが初心者には難しいため、あてずっぽうでプログラムをいじり出して、試行錯誤で動かそうとする。これでは論理思考の教育にはならない。

疑似コードとして日本語が使われてきたのは、教育用に限らない。大手金融機関では日本語の疑似コードを1対1でコボル文に対応するように作ってきた。そして、この対応を維持したまま、巨大な基幹システムを保守してきたのである。疑似コードをコボル文に翻訳する人間を「プログラマー」と呼ぶのが、業界の慣習であるらしい。単なる「コーダー」を「プログラマー」と呼ぶことから、「プログラマー」の社会的な地位が低いこともうなずける。

もう一つ、日本語疑似コードが実用になっているのは、国の行なう研究開発である。請負契約にするために、発注仕様通りのものが作られたかを調べて検取する必要があるのだが、C等で書かれたプログラムを読む作業が高くつくために、書かれたプログラムを日本語に翻訳して、それを日本語なら読める人間に仕様通りできているか確認させることが行なわれてきた。翻訳費用は全開発費の約1/3前後もかかっていたらしい。

疑似コードの日本語プログラムがそのまま実行できれば、これを実用言語に翻訳する必要がなくなる。実際、「ことだま」はそのような目的で開発されたプログラミング言語である。それが科学教育用にアラン・ケイによって開発されたSqueak上に搭載されたことで、テキスト入力が必要となり、タイルを貼ってプログラムを作ることになったため、文法エラーが起こらなくなった。従って、学習者はアルゴリズムの構築に集中できるようになったのである。

もう一つ、授業が成功してクラス全員が挿入ソートのプログラムが作れたのは、最初に選択ソートのアルゴリズムを、手作業で体験したことが大きく影響している。この体験をフローチャートで表現することを学んだ後、それを「ことだま」のプログラムとして組み立てて、実行させた。それから挿入ソートの実行過程を見せて、そのアルゴリズムを理解させた後、それをフローチャートに表現させ、「ことだま」のプログラムとして完成させる、というのが授業の流れであった。

タイルを貼る作業でプログラムを作った文法エラーから解放されても、意図通りには動くとは限らない。自分が書いたプログラムを読んで、そこに書かれたアルゴリズムがなぜ意図通りに動かないかを考えなければならない。この部分が論理思考教育として重要な部分である。ここで、日本語プログラミングが威力を発揮する。書かれたプログラムの意味が日本語で記述されていれば、それを読み解くことでなぜ意図通り動作しないかが理解できる。正確に日本語文を読む能力が要求されるのである。

従来の国語教育では、このように日本語を読み解く訓練はほとんど行なわれていない。文を読むことに関して、コンピュータに較べてはるかにインテリジェントな人間が読むことしか想定していないから、国語の授業ではプログラミングで要求される正確な読解は想定されていないのである。数学の教育で、このような正確な日本語を読み書きする訓練が行なわれてきたが、近年はそのような手間のかかる教育があまり行なわれなくなってきたようである。公理系から始まる幾何の証明が、

数学教育から消えてしまった影響が大きい。

3. UNESCO が勧める情報教育

2002年に発表されたユネスコの勧める情報教育 [1] では、情報教育を次の4つの段階に分けて論じている。

- (1) ICT Literacy
- (2) Application of ICT in Subject Areas
- (3) Infusing ICT across the Curriculum
- (4) ICT Specialization

(1) は、日本でも広く行なわれている、「情報手段」(指導要領の用語)の活用である。(2)の、各科目への「情報手段」の活用は、今始まろうとしている。しかし、外国で行なわれているような、周到的な教師教育と、教育用のシステム開発が行なわれていないために、導入しても負の側面が目立って、こうした教育への現場教員の支持が日本では広く得られていない。高校の校長会からは、「普通教科 情報」は廃止して欲しい」という要望が出る状況である。従って、欧米では現在進行中の、(3)の「カリキュラムの中に溶け込む」ような状況にはない。

特に問題なのが、(4) ICT Specialization である。UNESCO が推奨している教育内容が、情報技術の専門家とともに、高等教育に進学する学生のための内容であるとしていることが、日本では全く認識されていない。実際、欧米の大学卒のビジネスマンはみな、プログラミングができるのが当然で、できなければ大学に進学しなかった人であると見なされるそうである。

ユネスコの提案では、ICT Specialization は日常生活の問題をアルゴリズムとして解く能力を身につけることを目的としている。そして、その最初の内容である Introduction to Programming では、課題を解くアルゴリズムを設計 (Design) すること、その設計をプログラムとして実行可能な表現に変換すること、最後にプログラムを実生活で利用できるようにする所までを体験させるよう、推奨している。このことは、プログラミングの入門段階から、プログラムが使用される環境の中で、何を実現するかという仕様を考え、それを実現する

アルゴリズムを構築した上で、それをプログラムとして表現して、実際に使用して評価する所まで体験させること意味する。このように、プログラミング全体を体験させるような教育は行なわれていない。大学の専門教育でも、企業の技術者教育でも、プログラム作成の作業手順を教えるだけで、自分がしていることの意味を考える教育を行っていない。

4. 日本語プログラミング

日本語の語順は目的語の後に動詞が来るため、複雑な処理を記述する場合に、記述者は目的語のスタックだけを脳内に用意しておけば自分が実行したい動作をイメージできる、逆の語順の英語の場合は、動詞のスタックも必要となり、脳の負荷が大きくなる。このため、米国人が APL, FORTH, Post Script など、日本語の語順の言語を作っているが、母語の英語と語順が異なるため普及しなかった。日本語はコンピュータとのインターフェースがすぐれた言語なのである。

FORTH を輸入販売していた片桐 明氏が、FORTH を日本語化するだけで日本語プログラミング言語になったことから、これを本格化的な日本語プログラミング言語 Mind として商品化した [6]。この言語は、まず教育用として使われた所、Logo に較べて格段に学習効果が上がることが確認された [7]。Mind は単に教育用言語であるだけでなく、ソフトウェア開発言語として発展し、例えば「ぐるなび」の全文検索エンジンは、発展形の MindSearchII を使って開発が行なわれ、そのシステムが2004年5月から6年間使われ続けた。

Mind の特長として、修得が C 言語の約3分の1の時間で実務アプリケーションが組めるようになることがあげられる。また、可読性が高く、再利用がし易いので、開発期間が短縮できるようだ。

実は、数学も我々は英語の語順で記述されている。

$$g(x) = \frac{d}{dx} f(x)$$

という数式は、“The function g of x is equal to

the derivatives of the function f of x .” という英語の文を略記したものである。「 x の関数 g は x の関数 f の導関数に等しい。」という日本語表現とは全く違う語順であることが分る。数学の記述を変えることは今では不可能であるが、プログラミングは、母語を元にしたものの方が使い易い。日本人のために、日本語の語順のプログラミング言語を作る必要がある。

5. 日本語の特徴

日本語でプログラムを作ることを提案すると、国際化時代に逆行すると言われる。しかし、この問題は、必要なら作ったプログラムを読者が読める言語に翻訳することにすれば解決できる。人工知能研究として機械翻訳が研究されたが、人間は膨大な常識を前提として言語を使っているため、常識の扱いが難しくて実用化に至らなかった。しかし、形式的に意味が規定されているプログラミング言語の場合にはこの問題が起らない。従って機械翻訳の研究成果を利用すれば、プログラミング言語間の翻訳は十分可能であることが予想される。実用言語で書かれたプログラムを自然言語に翻訳することは、プログラミング教育にとどまらず、使われる可能性がある。ソフトウェア開発の検収に日本語への翻訳が必要であったことは、その可能性を示している。

「日本語は論理的でない」ということから、プログラムのような論理的な事象の記述には適さないと考える人も多い。しかし、これも思いこみに過ぎない。どんな言語を使っても、使用者が論理的でなければ、表現された文章は論理的でない。高等教育に至るまで日本語だけで国を維持している日本語が論理性に欠けることは考えられない。

最近米国人のロジャー・パルパーズが、日本語は少数の語彙と単純な文法で豊かな表現ができることから、英語より世界語としての可能性が高いという指摘をしている [8]。少数の語彙と単純な文法はプログラミング言語の特徴であり、日本語は、プログラミング言語と同じような特性を持っていることになる。

6. 算譜言語「敷島」の設計方針

パルパーズの日本語に関する指摘を受けて、日本語プログラミング言語も、文法を可能な限り単純化することを試みることにした。プログラミング言語として必要最小限なものは、処理対象の記述機能とその処理機能、それにこれら2つ機能に対する抽象化機能である。また、用語はできる限り日本語、特に大和ことばを使うことにする。かつてプログラミング用語の日本語化が検討されたが、現在使われているのは、「処理系」位であろう。この復活を試みる。手はじめに、「プログラム」には「算譜」を使うことにする。

「算譜」を記述する日本語には、新たな文字として「カッコ」が必要になる。かつて、明治維新に際して、日本人は西欧の文書習慣に合わせて、句読点と引用符を導入した。今回、文書がコンピュータによって解読されて、複雑な処理が行なわれることから、それに必要な記号として「カッコ」を導入することにする。何種類かの「カッコ」を用いることで、「算譜」はプログラムとして読める文章になるはずである。

処理対象のオブジェクトは「物」と呼ぶことにする。「物」はスタック上で処理をして、結果をスタックトップに置く。「物」は名前を表わし、「物」を置く操作と「物」をしまう操作で全ての処理を表わす。「基本物」には、数と文字を用意し、これを元にレコード構造によって複雑な「物」を構成していく。また、配列として、「数列」と「文字列」を用意する。「物」の定義と合わせて、その「物」に対する操作を定義できるようにする。

言語名としては「敷島(しきしま)」を用いることとした。当面の開発目標は、小学校における算数と国語の授業支援である。また、この活動を通して、コンピュータが情報を処理する仕組みを理解させたい。このための「しきしま1号」では、実数と整数の区別は設けず、「数」とすることにする。整数の除算には商と剰余からなる「商・余り」という「物」を定義する。文字列に対しては、同一性を調べる演算、長さを調べる演算、繋げたり、分

てたりする演算を設ける。

数式は数値を与える「物」として扱う。また、数式に名前をつけることもできるようにする必要があるかもしれない。この点は、学校教材を作る過程で決めることにする。

命令文は、「{物 (達)} を {操作する}」という形式に統一する。制御構造としては、「逐次構造」、「条件分岐構造」、「繰り返し構造」を用意する。また、「名前」、「物」と付随する「操作」を規定する宣言の機能を設ける。

開発にあたっては、「基本物」として整数を扱う「しきしま0号」をまず開発して、これを用いてその後の開発を行なうことで、言語処理系が日本語プログラミングで可能であることを示したい。

7. おわりに

従来のプログラミング言語は、プログラマーが遭遇した困難を解決するための機能を提供する努力を続けてきた。しかし、今後はプログラミングの専門家でない人達が、自分のしたいことを解決するためにプログラムを書くことが、多数を占めることになるであろう。そこで必要になることは、それぞれのプログラムを書く人が知っていることであって、プログラミングの専門家が知っていることではない。プログラムを書く一般人には、自分のしたいことが何であるかをできるだけ簡潔に記述できることが有難いのであり、便利だと専門外の人間が予想して作った仕組みを苦勞して習得して使うことは望ましくない。プログラミングに求められるものが変わるはずであり、その方向性を追求する第一歩として、算譜言語「敷島」を開発する。

参考文献

- [1] UNESCO, Information and Communication Technology in Education A Curriculum for Schools and Programme of Teacher Development, 2002, <http://unesdoc.unesco.org/images/0012/001295/129538e.pdf>.
- [2] WIKIPEDIA: AP ComputerScience, http://en.wikipedia.org/wiki/AP_Computer_Science#AP_Computer_Science_A
- [3] 杉浦 学, 松澤芳昭, 岡田 健, 大岩 元: アルゴリズム構

築能力育成の導入教育: 実作業による概念理解に基づくアルゴリズム構築体験とその効果, 情報処理学会論文誌, Vol.49, No.10, pp.3409-3427(2008).

- [4] 大岩 元 (監修), 松澤芳昭・杉浦 学 (編著): ことだま on Squeak で学ぶ論理思考とプログラミング, イーテキスト研究所, 2008. http://crew-lab.sfc.keio.ac.jp/lectures/2011s_ironpro/data/Squeak/Text/SqueakText.zip
- [5] ケビン・メイニー: C++も JAVA も子供に教えなくてもいい, Newsweek 日本版, June 24, 2014.
- [6] 片桐 明: 日本語プログラミング言語 Mind, <http://www.scripts-lab.co.jp/mind/whatsmind.html>
- [7] 西之園晴夫: 私信.
- [8] ロジャー・パルパーズ著早川敦子訳: 驚くべき日本語, 集英社インターナショナル, 2014.