

# 日本語プログラミングの小学校教育への適用可能性

大岩 元<sup>†1</sup>

プログラミングを日本の小学校で教えようとしても、指導要領の中に入れることは難しい。そこで、算数の授業にプログラミングを教材として導入する可能性について検討した。プログラミング言語として完全な日本語を用いることで、言語を教える手間がかからない。日本語による算数の諸概念を日本語プログラムで記述し、実行することは、一つの教材として有効そうである。

## Applying Japanese Programming to Elementary School Education

Hajime OHIWA<sup>†1</sup>

Because of the difficulty of changing the government regulation for school education, we have developed a plan to take advantage of loopholes of regulation for introducing programming education in elementary schools.

### 1. プログラミングにおける日本語の有効性

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。このことから、米国人が APL, FORTH, Post Script など、日本語の語順の言語を作ってきたが、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[1]。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[2]。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、Mind の発展形の MindSearchII を使って開発が行なわれ、そのシステムは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という。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。

$$g(x) = d/dx f(x)$$

という数式は、"The function g of x is equal to the derivative 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。

### 2. 日本語は非論理的で国際性がないか？

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持って言語を使っているため、こ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こらない。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。

「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が多い。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

<sup>†1</sup> お茶の水女子大学  
Ochanomizu University

ている日本語が論理性に欠けることは考えられない。科学ジャーナリストの松尾義之はノーベル賞講演を行なうまで外国に出たことのない増川俊英教授は日本語で物理学を研究してきたことを例にあげて、日本語が日本における科学研究の進歩に大きな影響を与えていることの主張している [3].

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簡潔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 [4]. 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。

### 3. 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。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、フローチャート作成による設計から始めて作れるようになったことが杉浦らによって報告されている [5]. 彼らの授業では、週 90 分 2 コマの授業を 13 週にわたって行っていたが、その前半に「ことだま on Squeak」 [6] 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、目的とするプログラムを作れたものと推測される。授業の後半は、「ことだま」で教えられた内容を Java で繰り返すことによって、実用言語においても、前半の授業で獲得された論理構築能力が生かされたことが報告されている。

総合政策学部 2 年の男子学生は次のレポートを残した。

=====  
プログラミングには (中略) 論理に基づき筋道だったプログラミングをする以外の方法はない。その意味でプログラミング自体に関する知識や表現の使用を多く求めない「ことだま on Squeak」は、思考訓練の面では大いに有効であったと思うのである。(中略) プログラミングに関する文法や表現を詳しく知らない段階では、「ことだま on Squeak」を使うことで、すでに用意された表現を使用しながら、より本質的な論理思考の訓練に専念できたと感じている。(中略) 読解が出来ない状態で作文など無理であるように、表現や文法を知らない段階で Java によるプログラミングは大変なハードワークであろう。私は初期段階において「ことだま on Squeak」を通して「順次」「分岐」「繰り返し」「変数」などの基礎的な思考方法を重点的に学ぶことができた。そこで得た思考能力が、他者が作成したプログラムを読解し理解する際に活用されたと思う。

=====  
従来 SFC では、授業全体を実用言語の Java (や C) で行ってきたのであるが、90 年代後半以後は、いわゆる学力低下のせいか、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、プログラミングが習得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ようである。

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。最近の日本の教育は正しいことを覚える教育に終始して、試行錯誤して何かを作り出すような教育が行なわれなくなってきたため、文法エラーが頻繁に起こることだけで挫折してしまい、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を目的とした、専門家を対象として言語であるので、実用プログラム作成を効率的に行なう工夫が多く成されている。このため、初心者にとって意味の理解できない作業を多く強いられることになってしまう。

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ろう。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少ない [7].

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。従来のプログラミング授業では、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、それには母語としての日本語が用いられてきた。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、予期したようには動作するとは限らないので、実行

時エラーを退治する必要がある。これを行なおうとすると、慣れない実用言語を正確に解説しなければならない。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維持してきたのである。これは、プログラム作成の本質的な情報を持つ銀行員が、直接プログラムを読まない責任がとれるシステムが構築できないと考えたからであろう。

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」の社会的な地位が日本では低いことがうなずける。

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、テキスト入力が不要となり、タイルを貼ってプログラムを作ることになったため、文法エラーが起らなくなった。これによって、学習者はアルゴリズムの構築体験に集中できるようになったのである。

もう一つ、杉浦らの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。その後で、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て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。日本語プログラマーには、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、このように論理的な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。

かつては数学の教育で、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。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ことの影響は大きい。

#### 4. 現行の指導要領から見た小学校の算数

日本の算数教育は世界的にみて成功しており、日本人の計算能力の高さは万人の認めるところである。しかし、このことは日本語自体が計算に適した構造を持っている所による所が大きいのではないかと考えられる。

日本の算数教育の問題点の一つとして考えられるのは、計算技術は教えているが、その意味についてうまく教えられていないのではないかということである。例えば、高畑勲監督の「おもひでぼろぼろ」(DVD)[8]には、分数の割り算で除数の分子と分母を入れ替えて掛ければよいということが納得いかない少女の話が重要な主題となっている。操作の意味を考えようとしなくて計算手順だけを覚える子供が評価され、意味を考えようとする子供が疎んじられる日本の(学校の)状況は、情報化時代の大きな問題である。理科離れ、数学離れが問題となっているが、その原因の一つはこの点にあるのではないと思われる。

実際、現行の学習指導要領[10]では、小学校の算数においては、その目標が

=====  
算数的活動を通して、数量や図形についての基礎的・基本的な知識及び技能を身に付け、日常の事象について見通しをもち筋道を立てて考え、表現する能力を育てるとともに、算数的活動の楽しさや数理的な処理のよさに気付き、進んで生活や学習に活用しようとする態度を育てる。  
=====

となっている。その内容はA 数と計算、B 量と測定、C 図形、D 数量関係 の4つの領域から成っている。これらの領域の育成にあたっては、児童が目的意識をもって主体的に取り組む、算数にかかわりのある様々な活動を「算数的活動」と名づけて、これを指導要領の中で具体的に示している。

現行の算数の指導要領は、A 数と計算 について1、2年次で整数に関する加減乗算を教え、3年次になると整数の除算とともに、従来4年次であつた小数、分数が入ってくる。ここで、小数は、小数部と整数部から成る数の意味である。4年次になると、小数、分数の計算と概数の概念が入り、四捨五入を扱う。5年次では偶数、奇数、約数と倍数、最小公倍数、最大公約数などの整数の概念、

小数の乗除算，分数の整数による除算を扱う．6年次では分数の乗除算を扱ってから，小数，分数の混合計算など，四則演算を完成する．

C 図計については，1年次では前後，左右，上下などの概念を扱い，位置を示せるようにする．また，三角形，四角形，積み木などの図形の特徴を理解させる．2年次では，三角形，四角形，正方形，長方形，直角三角形の概念と，箱形を例にして，頂点，辺，面など立体図形の基礎概念を理解させる．3年次では二等辺三角形，正三角形，角，円，球などを扱う．4年次では直線の平行や垂直，平行四辺形，台形，立方体，直方体を扱い，座標概念による位置の表現などを理解させる．5年次では多角形，正多角形，円周率，角柱，円柱などを扱う．6年次では縮図，量拡大図，図形の対称性について扱う．

指導要領は，これらの領域の内容を B 量と測定，D 数量関係で展開している．例えば，A 数と計算の領域の1年次で2位数，簡単な3位数を扱ってから，B 量と測定の領域の1年次では長さ，面積，体積の比較，2年次で長さ，体積の単位を扱い，D 数量関係の領域の1年次では加法や減法を式や絵，図を用いて数量の概念を通して表現させ，2年次では式による表現で加法と減法の間接関係を理解させ，表やグラフで数量の概念を扱う．また，算数的活動という内容を設けて，1年次にはア 具体物を数える活動，イ 計算の意味や仕方を表わす活動，ウ 量の大きさを比べる活動，エ 形を見つかったり，作ったりする活動，オ 場面を式に表わす活動を行なうことを求めている．

このように，スパイラルに概念の応用を展開することになっていることは，単に概念の意味を理解しただけで終りにするのではなく，具体的な実践活動を通じて，概念を具体的に応用する能力までつけさせようとしている．これが，21世紀の市民に求められる基礎能力である．

## 5. 算譜言語「敷島」の設計方針

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．そこで最も必要になることは，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であり，便利だと専門外の人間が予想して作った仕組を苦労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．

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．

まず，「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」は名前前で表わし，物を記憶する「変数」を用意する．「物」を置く操作と「物」をしまう操作，スタック上の「物」を処理する操作の羅列で全ての処理を表わす．

「基本物」には，「数」と「文字」を用意する．また，配列として，「数列」と「文字列」を用意する．これらを元に「構造物」（構造体，レコード構造）によって複雑な「物」を構成していく．「構造物」の定義と合わせて，その「構造物」に対する操作を定義できるようにする．「変数」の拡張として引数を持つ「関数」も用意する．

言語名としては「敷島」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．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，分け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．

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．

## 6. デジタルの本質を体験する

小学生の論理力を育てる教育として，熊本大学附属小学校では「論理科」カリキュラムを子どもと教師，子ども同志の対話を通じて「ことば＝論理力」の指導カリキュラムを開発し，実践した[9]．この活動の発展として日本語プログラミングを導入することを検討した．

「論理科」では，児童の対話を通じて論理力を育てる試みであったが，日本語プログラミングはコンピュータへの命令を通じて論理力を養おうとする試みである．書かれたプログラムは日本語とし完全であることで，児童は日本語

を正確に読みこなし、書き下す能力が養われる。

分数では $1/3$ という数を表現できるが、これを小数に直すと無限小数となり、どこかで打ち切らざるを得なくなる。我々が日常的に用いている小数表現は、近似表現にすぎないことが分る。しかし、便利であることから、小数で計算を行ない、分数計算を行なうことはない。分数電卓は存在しないのである。

図形を描くタートル・グラフィックスは、制御構造を教える優れた教材であるが、円は多角形で近似する。小数で $1/3$ を表現するのと同じことである。コンピュータ上での表現精度ぎりぎりの1ステップ進み、1度回転する命令を360回繰り返すことで円を表現することは、表現桁数いっぱい $1/3$ を表現するのと同じことになる。

このような例を通じて、コンピュータが現実世界を切り取って表現したものであることを体験させることも、情報教育の重要な内容であろう。

## 参考文献

- [1] 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>
- [2] 西之園晴夫：私信(1995).
- [3] 松尾義之，“日本語の科学が世界を変える”，筑摩書房(2015)
- [4] ロジャー・バ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
- [5] 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, (2008)
- [6] 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\\_ronpro/data/Squeak/Text/SqueakText.zip](http://crew-lab.sfc.keio.ac.jp/lectures/2011s_ronpro/data/Squeak/Text/SqueakText.zip)
- [7] ロジャー・バ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
- [8] 高畑 勲（監督）：おもひでぼろぼろ，スタジオジブリ作品，1991.
- [9] 内田伸子，鹿毛雅治，河野順子，“「対話」で広がる子どももの学び” 明治図書，2012