小学校教育課程におけるプログラミング教育の考察

－プログラミング教育に最適なプログラミング言語－

Study of programming education in elementary school curriculum

－Optimum programming language to programming education－

丸 山 幸 三

Kozo Maruyama

1. はじめに

小学校において、平成23年4月より全面施行となった新学習指導要領では、「生きる力」を育むという理念のもと、知識や技能の習得とともに思考力・判断力・表現力などの育成を重視する¹⁾としている。

新学習指導要領では、ICTや情報化というキーワードが随所に用いられている。これは文部科学省が、教育の情報化ビジョンにて21世紀に生きる子どもたちに求められる力として、「情報及び情報手段を主体的に選択し活用していくための個人の基礎的な資質が、読み、書き、算盤に並ぶ基礎・基本と位置付けられた」²⁾事によるものと考え。教育の情報化に関しては、子ども達の情報活用能力の育成のみならず、教科指導においても情報通信技術を効果的に活用した分かりやすい授業、さらに、校務の情報化による教職員の校務負担の軽減に至るまで、教育インフラ全ての面において情報通信技術の効果的な活用を目的としている。このような状況の中、2013年6月に、政府は経済政策「アベノミクス」の「3本目の矢」として発表した成長戦略の素案では「義務教育段階からのプログラミング教育等のIT教育を推進する」³⁾としている。すでに中学校では、平成24年度に完全実施となった新学習指導要領でプログラミング教育が技術家庭科に取り込まれおり、小学校においても近い将来情報教育の一環として取り入れられる可能性が高い。

本研究では、小学校教育課程におけるプログラミング教育の役割を明確にし、今後、プログラミング教育が本格的に実施されるようになった場合に重要となるプログラミング言語について、その機能と特徴を整理し、教育資源としての有用性を考察する。

2. プログラムとは

プログラムとは、端的に言うと、コンピュータが人に代わって仕事をするためのアルゴリズムの集

合である。人間が仕事をこなす場合においては、自己の持つ知識や経験を基に課題を解決することが可能であるが、今現在のコンピュータにはそのような知能は持ち合わせていない。従って、課題の解決手順をプログラムという形でコンピュータに人間が指示を与える必要がある。

プログラムデータは通常、プログラミング言語と呼ばれる独自のルールに基づいて書かれたテキストデータの集合である。このプログラミング言語は、人間社会における言語が国によって異なる事と同様に、動作させるコンピュータの種類や作成するソフトウェアの種類に応じて多数存在する。現在主に利用されているプログラム言語には、C、FORTRAN、COBOL、JAVA、BASIC、JavaScriptなどがある。プログラムは人間側から見ると理解しやすい形式になっているが、コンピュータにはそのままでの形では理解させることができない。そのため、プログラムはコンパイルと呼ばれる変換処理を経て機械語に翻訳し、コンピュータに理解させる。この機械語に変換されたものがソフトウェアである。つまり、プログラムを作成するための言語は多数存在するが、コンピュータ上で実行される際には、コンパイル（翻訳）することで全て同一の形に変換されていることになる。このコンパイルと呼ばれる変換処理は、例えるなら外国語を日本語に訳すことと考えれば理解しやすい。

3. 小学校教育課程におけるプログラミング教育の役割

プログラミングとは、人が何らかの目的をもってコンピュータに仕事をさせるための指示書をプログラミング言語を用いて作成する作業のことである。従って、プログラミング教育では、このプログラムを作る方法を学ぶことになる。

プログラムを作成する場合に求められる知識としては、主に次の二つが要求される。一つ目は、コンピュータに指示を与えるための言葉であるプログラミング言語を理解し、プログラムの書き方や指示を与えるための命令を覚える事である。二つ目は、プログラミング言語の中で使われる命令の組合せや順序を思考する能力となる。

前述の通り、プログラムを書くための言語は多数存在するが、これらのほぼ全てが英語に近い表記であり、英語の知識が少ない小学生に理解させることは難しいと思われる。従って、学校教育で一つ目の能力である「プログラムの書き方や、命令を覚える」ことを目的とすることは不合理であり、そこから子ども達が得るものは少ないと考えられる。一方、「命令の組合せや順序を思考する能力」については、子ども達が発論理的な思考を得るための教材として有望であると言える。プログラムを記述する際の基本的な概念の一つに「構造化プログラミング」がある。これは「一つの入口と一つの出口を持つプログラムは、順次・選択・反復の3つの論理構造によって記述できる」⁴⁾ というものである。つまり、この3つの構文をどのように組み合わせるかを思考する能力が必要となり、プログラミングを学習することで自然と論理的な思考が身につけられると考える。

4. プログラミング教育向け言語の特徴

一般的にプログラム言語は難解ですぐには理解できないものである。従って、これをそのまま学校教育の現場に持ち込み理解させることは、プログラミング経験のない教員、児童双方の立場からしても難しいと考えられる。しかし、近年その問題を解決するためのプログラム言語として、ビジュアルプログラミング言語と呼ばれるツールを利用してプログラムを組むことのできる言語が開発された。すでに海外では教育分野で実践的に活用されており、日本においても、産学官で様々なビジュアルプログラミング言語を用いての試みがなされるようになってきた。ビジュアルプログラミング言語とは、プログラムを作るためにキーボードから直接文字を入力する作業を極力少なくし、部品として用意されたブロックを組み合わせることでプログラムを組むことのできるソフトウェアである。従って、一般的なプログラミングのようにキーボードからプログラムコードを入力する事もなく、また、ステートメントや関数の使い方などを覚える必要もない。必要となるのは、ブロックイメージの持つ機能を視覚的に認識し、組み合わせを考えブロックを画面上にドラッグし配置するだけである。教育向けビジュアルプログラミングツールで作成するアプリケーションソフトウェアは、主に自分で描いたキャラクターを利用したゲームプログラムやアニメーションである。従って、ツールとして必要な機能としては、「キャラクターが描ける」、「描いたキャラクターを動かすことができる」、「キャラクターの衝突などのイベントを検知、操作できる」などが要求される。今現在のビジュアルプログラミング言語においては、前述の機能は概ね充たしておりプログラムの作成手順や操作方法についても大きく違いはない。ただし、「プログラムを作成する際の操作性」、「完成したプログラムの実行環境」、「ビジュアルツールで作成したプログラムから、JavaScriptなどのソースコードを生成する機能の有無」などで違いがある。

現在国内で用いられている主なビジュアルプログラミング言語の特徴について、インターネットに公開されている情報を元に調査した結果を以下にまとめる。

(1) Scratch (スクラッチ)

2006年にMITメディアラボのミッチェル・レズニックの研究グループが開発したビジュアルプログラミング言語で、子どもたちがプログラミングを学習するための教材として作成されたものである。プログラム教育用言語としては世界中の多くの地域で利用されており評価も高い。動作環境は、ローカルコンピュータにインストールして利用するタイプとWebサービスを利用するタイプが提供されている。日本語化もされており、文部科学省が主導し開発されたビジュアルプログラミング言語であるプログラミンはこのプログラムを参考に作成されたとされている。利用できるコマンド数も標準的に理解しやすいものとなっている。コード生成機能を有していないため、ツールで実現可能な範囲以上の機能は望めない。Webサービスを利用する場合は、事前にユーザー登録が必要となり、サイトを開いた際はログイン認証を要求される。作成したプログラムはWebサーバ上に保管され、プログラムの公開も自由に行うことができる。プログラムの作成は

- ① 画面で動作させるスプライト（キャラクター）を選択する。
- ② 選択したスプライトを動作させるためのスクリプト（プログラム）を予め用意されたブロックを組み合わせて作成する。

簡単なプログラムであれば以上で完成となる。この他、手書きしたオリジナルのキャラクターを登録し、動作させたり効果音を付け加えるなども可能である。

Scratchではさらに、2014年8月に幼児教育向けにScratchJrのサービスも追加されている。このことで、幼児教育から初等教育まで幅広い年齢での活用が期待されている。

(2) プログラミン

平成22年8月に文部科学省が、子どもたちでも簡単にプログラミングが体験できるウェブサイトとして公開したビジュアルプログラミングツールである。Flashベースであるため、ユーザインターフェースに優れており小学生でも直観的に分かりやすい。実行環境はWebサービスとして提供されているにも関わらず、他のツールに比べ動きも軽快である。Web版Scratchのようなユーザー認証は不要であるが、作成したプログラムはローカルコンピュータ上に保管される。作成したプログラムをサーバへアップロードする仕組みも用意されており、互いの作品を見せ合い、評価することも可能である。Scratchを参考に開発した経緯もあり、その他の機能については、概ねScratchに準拠している。

(3) MOONBlock（ムーンブロック）

秋葉原リサーチセンターにて開発された国産のビジュアルプログラミング言語である。他の言語と同様に教育用プログラミング言語として提供されており、Scratchやプログラミンと同様にブロックの組み合わせでプログラムを作成する。利用できる部品は、ゲーム制作に適したコマンドが豊富であり、各ブロックには日本語によるガイダンスが表記されている。また、日本語化されたビジュアルプログラミング言語の中では唯一JavaScript言語への変換機能を有している。操作は他の言語と比較して、やや直感的に操作しづらいユーザインターフェースであるように感じる。また、機能面でも手書きのキャラクターが利用できないため、パペットと呼ばれる予め用意されたものから選択するのみとなる。小学校での利用を検討する場合は、ある程度の言葉の意味を理解する必要があるため、低学年での導入は難しいが、ビジュアルプログラミングでは物足りなくなった高学年には次のステップとしてJavaScript言語でのプログラミングへの移行が容易に行えるメリットがある。

(4) Squeak Etoys（スクイーク イートイズ）

パーソナルコンピュータの父と呼ばれている米国のアラン・ケイ博士らが中心となって開発した教育向けビジュアルプログラミング言語である。Scratchの元となったとされ、ビジュアルプログラミング言語としてはパイオニア的な存在である。現在主流であるブロックを組み合わせで作り上げる形式ではなく、画面上に配置したキャラクターに対し、属性や振る舞いを設定していく事で組み上げる仕組みとしている。他の言語と比較し、抽象的な表現や専門的な用語が多く、やや難易度

が高い印象となっているが、ビジネス分野で現在主流となっているオブジェクト指向プログラミングに近い形であり、JavaやC言語へのステップアップする場合の最初の一步としては、相応しいと考える。また、世界聴診器と呼ばれるセンサー機器との連携で「温度」、「光」、「電圧」などの変化を捉え、そのデータを元にしたプログラムの作成が可能となっており、情報教育以外に理科の授業などでの活用が見込まれる。教育向けビジュアルプログラミング言語としては歴史があり、すでに日本でも多くの教育機関で活用されており、導入事例や検証結果なども多数報告されている。また、指導者向けの講座も定期的に開催されており、実践的に導入を考えている機関にとっては有利である。

(5) ビスケット

2003年にNTTの研究機関で開発された国産のビジュアルプログラミング言語である。プログラムの作成方法は、非常にユニークなものとなっており、メガネと呼ばれる二つの円でできた動作を定義する器に自分で描いたキャラクターをセットし作成する。メガネ状の二つの円は、左側の円がキャラクターの最初の位置または最初の状態を表し、右側の円に移動する方向や移動後の状態を表すキャラクターをセットする。プログラム作成はマウス操作のみで行う事が可能で、プログラミングと言うよりもアニメーションを作成する感覚で作ることができる。開発者のコンセプトにも「ビスケットが伝えたいメッセージは、プログラミング入門ではありません。プログラミング入門のツールだと、現在主流のプログラミング言語に近いことを簡単にできるようにする方法がとられるでしょう。これがScratchやプログラミンといったブロック型のビジュアルプログラミング言語です。」
5) と述べられているように、プログラミング能力そのものを学ぶのではなく、想像力、発想力、表現力などを養う事に主眼がおかれている。非営利団体であるデジタルポケットによるワークショップも盛んに実施されており、各地での取組の様子や実際に実施されているカリキュラムなどの公開もされている。

表1にビジュアルプログラミング言語の特徴的な機能の一覧を示す。また、機能ごとの要点を以下にまとめる。

(a) プログラムの組み立て

今回調査したビジュアルプログラミング言語の内三つの言語でブロックを組み合わせて作る方法が取られている。これは、構造化プログラミングの基本を学ぶには最適であると考えられる。プログラミング用の部品として用意されたブロックには、構造化プログラミングで必要となる「順次」「選択」「反復」が提供されており、この組合せを思考しプログラミングが行えるようになれば、その思考は次のステップで構造化プログラミング言語でのプログラミングに生かせる。また、Squeak Etoysについては、キャラクターごとの属性を設定する仕組みであることから、オブジェクト指向プログラミングでの思考を身に着けることができると考える。一方ビスケットについては、構築方法が独自で該当する一般的なプログラミング技法は見当たらない。しかし、最も直感的に操作でき

る言語であり、言葉の意味が十分理解できない低学年を対象に、プログラムを作る方法を学ぶのではなく、創造力、協調性、表現力を養うことを目的とした場合には有効であると考ええる。

(b) キャラクターの手書き

MOONBlock以外は全て可能となっている。プログラミングにおいては、キャラクターの手書きが必須ではないが、手書きができることにより、子ども達の想像力が豊かになると共に、プログラミング作業に対するモチベーションが高くなる効果が期待できる。特にプログラミングという概念が理解できていない低学年を対象にプログラミング教育を実施する場合、その概念を理解させる前段階として、まずは興味を抱かせることを目的に授業を進めるためには重要と考える。

(c) ソースコード生成

今回検証した言語の中では、MOONBlockのみが可能であった。MOONBlockでは作成したプログラムをJavaScript言語に変換できる。ビジュアルプログラミング言語で作成されたプログラムを、JavaScriptなどのソースコードに変換する機能を有することで、ブロックの組み合わせが実際のソースコードではどのような展開されるのか照らし合わせる事が可能となり、直接ソースコードによる作成へステップアップする場合に役立てることができる。さらに、ブロックの組み合わせだけでは実現できない機能を直接ソースコードを編集することで可能となり、より深いレベルでの学習が行える。

(d) 作品の公開

MOONBlock以外は全て可能となる。作成したプログラムを公開し互いのプログラムを評価し合うことは、コミュニケーション能力の向上が期待されると共にモチベーションも高まると考えられる。

(e) 動作環境

動作環境では、Squeak Etoys以外はWeb上での開発が可能である。Web接続が行えない環境である場合はSqueak EtoysもしくはScratchを選択する必要があるが、Web接続が可能であれば導入の手間のかからないWebでの動作環境が望ましいと考える。

(f) プログラムの保存先

Web接続が行えない環境を考慮する場合は、ビケットは利用できない。その他の言語については、ローカルコンピュータやリムーバブルディスクにプログラムの保存が可能である。Web接続環境については、今後さらに整備されていくと考えられ、Web上での保存が主流になるとと思われる。

表 1

言語名	プログラムの組み立て	キャラクターの手書き	ソースコード生成	作品の公開	動作環境	プログラムの保存先
Scratch	ブロックを組み合わせる	○	×	○	Web/ ローカル	Web/ ローカル
プログラミン	ブロックを組み合わせる	○	×	○	Web	ローカル
MOONBlock	ブロックを組み合わせる	×	○	×	Web	ローカル
Squeak Etoys	キャラクターごとの属性を設定する	○	×	○	ローカル	ローカル
ビスケット	独自（文字入力が不要）	○	×	○	Web	Web

5. プログラミング教育向に最適な言語

プログラミング言語ごとに機能を整理した結果、小学校教育課程でのプログラミング教育を実施する上で、各プログラミング言語の特徴の違いから、次の2つの点で考慮する必要があると考える。

1点目は、言葉の理解度である。基本的にはビジュアルプログラミング言語は、部品を組み合わせることでプログラミングが可能であるが、今回検証した言語の中で、全く文字を使わずに作成できるのはビスケットのみとなる。従って、小学校低学年（1・2年生）において、ビスケット以外の言語を選択した場合は、言葉の意味を理解させる必要があり、適切な選択ではないと思われる。

2点目は、ビジュアルプログラミング言語から一般的なプログラミング言語へのステップである。ビジュアルプログラミング言語では、論理的な思考やプログラムの構造などは身に付くが、そこからテキストベースのプログラミングへの移行は、部品と構文の関連を理解する上でソースコード生成機能が備わっているMOONBlockが有利だと考える。

以上の結果を踏まえ、学年ごとのプログラミング教育向に最適な言語を表2にまとめる。

表 2

	1年生	2年生	3年生	4年生	5年生	6年生
Scratch	△	△	◎	◎	◎	◎
プログラミン	△	△	◎	◎	◎	◎
MOONBlock	×	×	○	○	◎	◎
Squeak Etoys	△	△	◎	◎	◎	◎
ビスケット	◎	◎	◎	○	○	○

×：不適切、△：指導に工夫が必要、○：適切、◎：最適

6. まとめ

今回の調査で、小学校教育課程における教育用プログラミング言語としては、Java、C、Basicなどの一般的なプログラミング言語は利用されず、ビジュアルプログラミング言語と呼ばれる教育を目的とした専用の言語が用いられている事がわかった。また、プログラミング教育の目的が、「プログラムを作成する能力を身に付ける事」ではなく、21世紀型スキルと呼ばれる「創造力」、「批判的思考力」、「コミュニケーション能力」、「コラボレーション能力」、「問題解決能力」、「自律的に学習する力」を養うための教育用ツールとしての位置づけであることもわかった。従って、教育向けに提供されているビジュアルプログラミング言語は多数存在するが、プログラムの組み方に多少の違いが有るものの、表2で示した通り、年齢ごとにプログラミング言語を使い分けるなどする事で支障なく実施できるのではないかと考える。

今後は、これらのビジュアルプログラミング言語を利用してのプログラミング授業をデザインし、ワークショップなど通じ、その効果の検証を行いたいと考える。

引用文献

- 1) 文部科学省：「新学習指導要領・生きる力」http://www.mext.go.jp/a_menu/shotou/new-cs/ (2014/10/21現在)
- 2) 文部科学省：教育の情報化ビジョン～21世紀にふさわしい学びと学校の創造を目指して～ p46 http://www.mext.go.jp/b_menu/houdou/23/04/_icsFiles/afieldfile/2011/04/28/1305484_01_1.pdf (2014/10/21現在)
- 3) 首相官邸：産業競争力会議 成長戦略（素案） p46 <http://www.kantei.go.jp/jp/singi/keizaisaisei/skkaigi/dai11/siryoul-1.pdf> (2014/10/21現在)
- 4) IT用語辞典 e-Words：構造化プログラミング <http://e-words.jp/w/E6A78BE980A0E58C96E38397E383ADE382B0E383A9E3839FE383B3E382B0.html> (2014/10/21現在)
- 5) ビスケットとは！ VISCUIT（ビスケット）～コンピュータを粘土のように～<http://www.viscuit.com/about/> (2014/10/21現在)

参考文献

- 1) 石戸奈々子：「子どもの創造力スイッチ!」、フィルムアート社、2014
- 2) P.グリフィン、B.マクゴー編、E.ケア／三宅なほみ 監訳、益川弘如、望月俊男 編訳：「21世紀型スキル 学びと評価のあらたなカタチ」、2014
- 3) 関西大学初等部：「関大初等部式 思考力育成法」、さくら社、2012
- 4) 大岩元 監修、松澤芳昭、杉浦学：「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

- 5) Thoru Yamamoto : 「スクイークであそぼう」、翔泳社、2008
- 6) 布留川英一 : 「MOONBlockでつくるゲームプログラミング」、誠文堂新光社、2013
- 7) 阿部和弘 : 「小学生からはじめる わくわくプログラミング」、日経BP社、2014
- 8) 阿部和弘 監修、阿部和弘、石原淳也、塩野禎隆 : 「Raspberry Piではじめるときどきプログラミング」、日経BP社、2014
- 9) 阿部和弘 監修、橋爪香織、谷内正裕、阿部和弘 : 「5才からはじめる すくすくプログラミング」、日経BP社、2014
- 10) デジタルハリウッド大学大学院 : 「スクイークで学ぶゲームプログラミング」、デジタルハリウッド大学大学院、2007