

特集論文

小学校プログラミング授業の推進における実践上の課題

Report on considerations for promoting programming class at elementary school

豊田 充崇

TOYODA Michitaka

(和歌山大学大学院教育学研究科教職開発専攻)

受理日 平成 30 年 1 月 27 日

抄録：小学校におけるプログラミング教育が平成 32 年から全面実施されることとなり、次年度から移行期間となる。しかしながら、小学校教員による指導上の不安や抵抗感から、その普及・推進が懸念されている。そこで、平成 29 年度に先行的に実施された小学校でのプログラミングの事例をまとめ、プログラミング授業の実態を探った。各種授業実践をその形態別に分類した所、現状でも様々なバリエーションの授業が試行的に進められていた。ただ、児童らが扱うプログラム処理の方式や達成レベル等が考慮されずに、授業の系統性に問題が多いことも明らかになった。さらに、各授業者への聞き取りによって、環境整備・時間数・指導力・評価方法等にも課題がみられた。また、「プログラミング教育」が独立した新たな分野として特別視される傾向にあり、情報活用能力の視点から捉え直す必要性なども見出させたといえる。

キーワード：プログラミング教育、プログラミング的思考、情報活用能力、学習指導要領

1. はじめに

次期学習指導要領において、小学校でのプログラミング授業の必修化が明記された。この起点は、平成 25 年（2013 年）6 月の閣議決定により出された「世界最先端 IT 国家創造宣言」¹⁾の中の「国際的にも通用・リードする実践的な高度な IT 人材の育成」についての項目に、「初等・中等教育段階からプログラミング等の IT 教育を」との記述にあるといえる。

国策として推進されることとなったプログラミング教育は、平成 28 年 6 月「日本再興戦略 2016 - 第 4 次産業革命に向けて」²⁾では更に「人材の強化」として、「日本の若者が第 4 次産業革命時代を生き抜き、主導できるよう、プログラミング教育を必修化する」と語気を強めている。

しかしながら、プログラミング教育自体は新しいものではない。当然ながら、現在の中学校（技術・家庭科）・高等学校の情報科では実際におこなわれているし、小学校においても既に実践されている。昨今では、2013 年に「プログラミン」という児童向けのプログラム学習サイトを文部科学省のアドレス下 ([http://](http://www.mext.go.jp/programin/)

www.mext.go.jp/programin/) で提供したことによって、多くの小学校で使われ実践されてきた。

よって、次期学習指導要領において注目されているのは、小学校でのプログラミング教育の“必修化”である。全ての小学校においてプログラミング教育を実施することとなり、当然ながらその指導者は担任教員に委ねられることとなる。もちろん、専科教員が担当したり、ICT 支援員のサポートも考えられるが、原則としては小学校教員にプログラミング授業を実践できる指導力を備えることが求められたといえよう。

しかしながら、実際の教育現場の反応は、冷ややかという印象を受けざるをえない。一般の教育現場では、ようやく教室内に大型提示装置やデジタル教科書の導入が進み、タブレット情報端末の普及も進みだした状況である。ただ、インターネットをはじめ各種 ICT を学びのツールとして積極的に活用したいと思う反面、一方でスマートフォンや SNS 利用によるトラブルに悩まされることが多いこともあり、その使い方のルールやモラル指導に苦慮する面もある。このような状況下において、小学校でのプログラミング教育の必修化までまだ意識が向かないのも理解できる。期待感

よりも抵抗感が強く、「人材育成」を第一に掲げた産業界からの要請に嫌悪感さえ抱いている教員も多い。

そこで、小学校プログラミング教育の必修化に向けての課題を整理することや、既に実践されている先進事例から普及・推進のためのモデル的な授業をピックアップし、その授業イメージを共有しておく必要があると考えた。

2. 本研究の目的と方法

2.1. 目的

本研究の目的は、小学校でのプログラミング授業を実践する際に、どういった指導力を要するか、授業を設計する上で何に留意すべきか、そして全学級への普及を目指す際の課題は何かを、先行する実践事例から探ることにある。但し、本年度末に文部科学省より「小学校プログラミング教育指針（仮称）」が出されるために、ここでの総括は無駄になる可能性も高い。しかしながら、学習指導要領の移行期間前に実施された黎明期の事例をまとめ、現状における課題を一旦明示しておくことは、今後継続的に実施されるプログラミング教育の発端の記録として価値があるといえる。ここで明示された課題は永続的なものか、創成時特有のものかが後に判明するであろう。

2.2. 方法

文部科学省により公開されている資料を読み解き、再度、小学校プログラミング教育の位置付けを明確にする。また、これまでのプログラミングに関する調査資料等も加えて取り上げ、児童生徒らの実態を捉えたい。その上で、筆者が関わった小学校におけるプログラミング授業の実践事例を取り上げ、その実施形態を概観する。また、各実践者への授業後の聞き取り調査によって、実践上の課題を整理していくこととする。

3. 学習指導要領における位置付け

平成 28 年 6 月に文部科学省より、「小学校段階におけるプログラミング教育の在り方について」（議論の取りまとめ）³⁾が出され、これにて小学校の新しい学習指導要領へのプログラミング教育の必修化やその目指す意図・方向性が示された。

次いで、同年 12 月の中央教育審議会より「幼稚園、小学校、中学校、高等学校及び特別支援学校の学習指導要領等の改善及び必要な方策等について（答申）」が出され、「情報活用能力（情報技術を手段として活用する力を含む）の育成」の頁では、「プログラミング的思考」が育成すべき資質・能力としての情報活用能力に含まれると記された。また、各教科等においても計 10 箇所「プログラミング的思考」に関しての

記述がなされている。さらに、別紙資料にも「議論の取りまとめの図」が再掲されたり、各教科（総合的な学習の時間、理科、算数、音楽、図工、特別活動）でのプログラミング教育の実践事例が示されるなど、一気に小学校でのプログラミングの必修化が確立されたといえる。

これらの答申内容は、当然ながら平成 29 年 3 月に公示された新しい学習指導要領へ受け継がれ、プログラミングを「論理的思考力を身に付けるための学習活動」として明記し、総合的な学習の時間、算数、理科においてもプログラミングを行う際の取り扱いの留意事項が記された。

次に、小学校学習指導要領解説 総則編（平成 29 年 6 月）⁵⁾における、プログラミング指導に関する箇所を項目立てて整理してみたい。

まず、小学校におけるプログラミング教育の前提としては、プログラミング言語を覚えたり、その技能を習得することがねらいではないとの記述があり、「プログラミング的思考」は、将来どのような職業に就くとしても時代を越えて普遍的に求められるとして、プログラマーの養成ではないという点が強調されている。「プログラミング的思考」の定義としては、「自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力」とされており、論理的思考力の一部（もしくは下位項目）として「プログラミング的思考」が位置付けられたといえよう。

もう 1 点前提として押さえておきたいのは、『小学校においては、児童が「プログラミングを体験」しながら、「コンピュータに意図した処理を行わせるために必要な論理的思考力」を身に付けるための学習活動を計画的に実施する。』との記述であり、いわゆるアンプラグドコンピューティング（コンピュータを使わないプログラミング学習）において、仮にプログラミング的思考が育成可能であるとしても、それだけにとどまらず、やはりコンピュータを体験しながら実施することが必須条件であることが明確に示されている。

学習指導要領の解説編における「プログラミング指導のねらい」をまとめると、以下の 4 点が見出だせる。

- (1) 論理的思考力を育む。
- (2) プログラムの働きやよさ、情報社会がコンピュータをはじめとする情報技術によって支えられていることなどを気付く。
- (3) 身近な問題の解決に主体的に取り組む態度やコンピュータ等を上手に活用してよりよい社会を築いていこうとする態度などを育む
- (4) 教科等で学ぶ知識及び技能等をより確実に身に付

けさせる。

これらのねらいを「資質・能力」の「3つの柱」にあてはめて端的に示すと以下ようになる。

【知識・技能】(小学校)身近な生活でコンピュータが活用されていることや、問題の解決には必要な手順があることに気付くこと。(中学校・高校は略)

【思考力・判断力・表現力等】発達の段階に即して、「プログラミング的思考」を育成すること。

【学びに向かう力・人間性等】発達の段階に即して、コンピュータの働きを、よりよい人生や社会づくりに生かそうとする態度を涵養すること。

但し、この解説編にも、小学校において、どの教科で、何時間程度、どういった内容を指導するのかについては踏み込まれておらず、この点については「小学校プログラミング教育指針」(仮称)における公表を待たねばならない。

4. 児童生徒及び指導者の実態

まず、文部科学省による「情報活用能力調査」(平成25年)⁴⁾からプログラミングに関係する結果を取り上げたい。調査結果第3章の「図表 3-2-7a 教員に対する質問調査の結果」において、小学校教員での「ゲームを作ったり、ロボットを動かしたりするなどのプログラミングを行わせること」についての項目では、「ほとんどおこなっていない(12.9%) + 全く行っていない(85.3%)」となっており、わずかに1.7%が「学期に1回以上」と回答している。

「指導できる」は0.9%、どちらかといえば指導できるの2.6%を加えてもわずかに3.5%にとどまる。逆を言えば、「どちらかといえば指導できない(18.1%) + 指導できない(78.4%)」= 96.5%がプログラミングの指導ができない状況であり、やはり懸念すべき高い割合であることは確かである。

また、同調査においては、中学校2年向けに「フローチャート」に関する出題があった。掃除機モーターの自動制御に関して、フローチャート内の4つの空白を埋める問題である(図1参照)。空白に入る選択肢は4枚のカードとなっているために、惑わせるカードは設定されておらず、それぞれを適した箇所配置すればいい。フローチャート内には、◇(ひしがた)のマークが2箇所あり、いわゆる条件分岐を示しているため、「～か」の末尾で終わるものが入る。そう考えることさえできれば、2択まで絞りきれぬ問題である。しかしながら、正答率はわずかに17.9%となっており、当調査の中でかなり低位の正答率(最下位から3番目)となっている。

これらの結果から、平成25年(2013年)の時点では、プログラミング授業に取り組む小学校は極めて少数であり、指導可能な教員もほぼいない状況であるこ

とがうかがえる。また、生徒の実態として、中学校2年生時点でもフローチャートによるアルゴリズムの理解ができていない生徒が圧倒的に多いといえる。ここ4.5年で小学校でのICT機器の整備が進んだとはいえ、プログラミングに関するこの統計結果が大きく変化したとは考えづらい。よって、この統計結果に変化がないと仮定した場合に、「小学校におけるプログラミング必修化」が2年後に迫ったいま、その実現性が危ぶまれる結果であることは明白であろう。

折しも、小学校の教育現場での関心事は、5・6年生の外国語および3・4年生の外国語活動や「特別の教科 道徳」に向いており、プログラミング教育については懐疑的な見方が多い。外国語(英語)は、教諭に採用される前にも、当然中学校の義務教育から学び、高校・大学受験ともに学習を重ねてきた経緯があるため、その教科内容自体については一定の理解があるといえる。また、道徳の教科化についても、趣旨や指導方法・評価方法が変更されたとはいえ、これまでの道徳のベースがあるために、教材研究を深めることで対応は可能であろう。

しかしながら、プログラミング教育については、大部分の教員は自ら関連の授業を受けたことがない。用語1つとっても、初めて聞く言葉も多く、それを指導者の立場として習得することは非常に困難であることいえる。

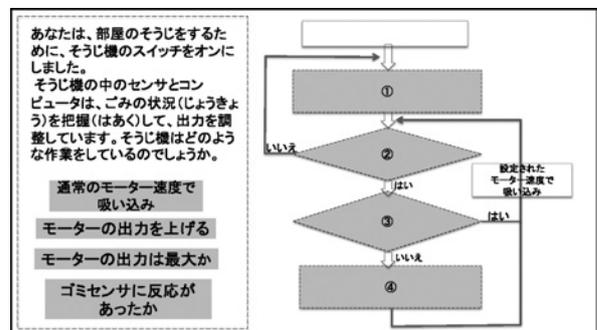


図1 文部科学省・情報活用能力調査におけるフローチャート作成の問題(筆者が調査報告書より再現)

5. 小学校におけるプログラミング授業の実態

以上のような実態・状況の中においても、一部の学校では、創意工夫を凝らして先行的にプログラミング教育の推進をおこなってきた。以下は、2017年度中に先行して実践されてきた小学校でのプログラミング授業の事例であり、形態の異なるものを時系列順に9事例ピックアップした。これらは、実際に筆者が授業づくりからかかわったものや、定例のICT活用授業研究会で実践された事例を集めたものである。また、授業後の協議では各授業者に実践上の留意点や課題についてのヒアリングを実施し、合わせて整理している。

5.1. 小学校でのプログラミング授業実践事例

事例1（大阪市：6年生）：グループで考えた架空のストーリーをプログラムによって創作する。画面上に複数の登場人物（キャラクター）を配置して、各種動作の命令を与えて、ストーリー通りにタイミングを取って動かしていくもの。

- ・使用言語：プログラミン（文部科学省サイト）
- ・扱うプログラム：ストーリー性のあるアニメーションを実行するためのプログラム
- ・扱う処理：主として順次処理

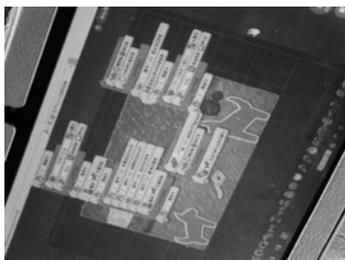


図2 各キャラクターに動作命令を与えている場面

事例2（大阪市：4年生）：総合的な学習の時間の一環として、未来を切り拓く力の獲得を目指すためにプログラミングスキルを育成する授業。基本的には、個人で設計した簡単な当たり判定のあるゲームプログラムの作成等を実施。フローチャートの描画方法も学ぶ。

- ・使用言語：Scratch Jr.（iPad アプリ）
- ・扱うプログラム：主にゲーム系プログラム
- ・扱う処理：順次処理・条件分岐・繰り返し



図3 金魚すくいゲームを作成中の児童の様子

事例3（和歌山市：3年生）：スタート地点にあるボールロボットをゴールまで導くためには、どのような命令（プログラム）を組むのが最適かを考える。

- ・使用言語：Sphero+edu（iPad のアプリ）
- ・扱うプログラム：ボールロボット制御プログラム
- ・扱う処理：順次処理・繰り返し



図4 ボールロボットをコース通りに動かそうとしている様子

事例4（和歌山市：5年生）：児童らが休憩時間やお楽しみ会などで実施している3択クイズは、出題→選択肢（正答と誤答）の提示→解答→正答判断→得点加算→最初へというようなプロセスがあり、順次・条件分岐・繰り返しの処理を含んでいる。この考えをもとに、総合的な学習の時間における地域学習の一環として、地元関連クイズプログラムの作成を実施したものである。

- ・使用言語：Scratch（ダウンロード版）
- ・扱うプログラム：Block形式で書かれた3択クイズの出題・正答判定のプログラム
- ・扱う処理：順次処理・条件分岐・繰り返し
発展レベルでは変数を扱う。



図5 「発展レベル」のクイズプログラムの例

事例5（兵庫県：6年生）：PCとUSBケーブルで接続したキットを使い、「信号」の動きをシミュレートする活動。図6のPCの上部に置かれているのが「信号」を再現したキットである。

- 使用言語：Studuino（Scratchベースの専用ソフト）
- 扱うプログラム：Block形式での信号制御プログラム（出来上がりはほぼ同じものとなる）
- 扱う処理：順次処理・条件分岐・繰り返し

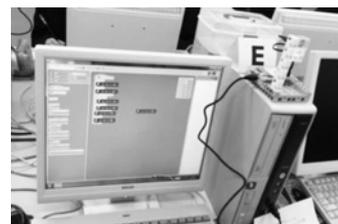


図6 点灯・点滅・消灯等の命令を与える

事例6（大阪府：5年生）：チュートリアル型プログラミング学習サイトの「プログル」を用いて、算数の倍数を求めるプログラムづくりをおこなう事例。5の倍数を求めるプログラムを基本達成レベルとして、発展レベルとして3と5の公倍数を求めるプログラムまでを組むこととしている。正しくプログラムが組まれているかも自動判定される。

- ・使用言語：プログル（ウェブサイトをPCで利用）
- ・扱うプログラム：決められた形のBlock型のプログラムを作成。
- ・扱う処理：順次処理・条件分岐・繰り返し

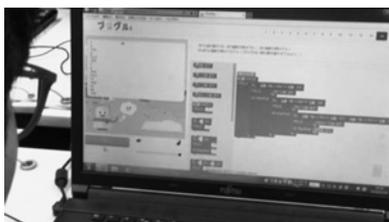


図7 上部に問題文が出題され中央に表示されるブロックを用いてプログラムを作成する

事例7 (大阪市: 5年生): 前半は英語活動で街案内をする際の Go straight や Turn left 等のフレーズの復習。後半は、その命令をプログラミングのコマンドに置き換えて、チュートリアル型のプログラミング学習アプリである Swift Playgrounds に挑戦。

- ・使用言語: Swift Playgrounds (iPad アプリ)
- ・扱うプログラム: 動きの命じる基本コマンドの入力
- ・扱う処理: 順次処理・条件分岐・Function 定義等

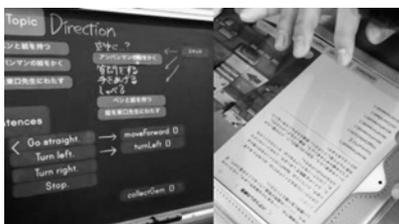


図8 最初は先生に命令を与えるところから始まり、その後英語での街案内からプログラミングへ

事例8 (和歌山市: 5年生): 給食の配膳や掃除の手順などをフローチャート化してその流れを可視化する取り組み。当番かどうかで分岐をつくっている。そのフローチャートの概念を使って、三角形の場合分けをフローチャートを使って考える取り組みに発展させている。

- ・使用言語: なし (アンプラグド)
- ・制作物: 日常の学校生活をフローチャートで表現したものの。
- ・扱う処理: 順次処理・条件分岐・繰り返し

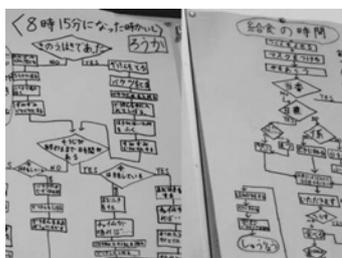


図9 掃除手順や給食の配膳等を示した例

事例9 (和歌山県: 4・5・6年生): 「Pepper 社会貢献プログラム」のひとつ「スクールチャレンジ」の取り組み。Pepper の発声、音声認識、動作、センサー

類などのコマンドをステップごとに学び、最終的には「○○に役立つペッパー」をテーマに、自作のプログラムをグループで制作する。

- ・使用言語: Choregraphe (コレグラフ)
- ・扱うプログラム: ボックスと呼ばれるコマンドを集約したものを順番に配置し、地域紹介や学習クイズ、漫才などのプログラムを作成。
- ・扱う処理: 順次処理・条件分岐・繰り返し計測・制御を含む。



図10 ブロック型のプログラムを解説する児童 (左) と Pepper を制御している様子 (右)

5.2. 小学校プログラミング授業事例を通して

事例 3.5.9 は電子キットやロボットを購入する必要があるが、その他の事例は無料サイトやアプリを利用しているため、現行の校内情報設備環境で実施可能であると考えられる。

懸念事項であったコードの入力については、事例8を除けば、全て Block 形式、Box 形式というものでマウス (タブレットの場合は指) でコマンドアイコンをドラッグすることで操作可能であり、パラメーターを入力する以外はキーボードは不要であり、プログラミングにおけるコード・コマンド入力において困るような場面はほとんどみられなかった。

なお、ここでの事例はすべて「総合的な学習の時間」として実施されている。事例6は算数としての設定でも可能かと思われるが、教科の観点での「評価」がしづらいため、算数の授業としての扱いは慎重にならざるをえない事情がある。

また、事例7に関連しては、プログラミングのコマンドに英語を用いているため、「外国語」とプログラミングの融合的な授業の位置付けとしている。「平成26年度文部科学省委託事業 情報教育指導力向上支援事業 プログラミング教育実践ガイド」での小学校事例における「学びの変容」の項目に、「英語を使ったプログラミングによって、英語とプログラミングがつながるようになった」という成果も記述されており、教科化される「外国語」との関連も検討すべき事案であると考えられる。

5.3. 事例分類表の作成

筆者がかかわった事例1～9をもとに、その授業者らと授業を組み立てる際にリサーチした事例等を加え

て、現状の小学校で実践可能なプログラミング授業の形式を分類した(表1)。これは小学校プログラミング教育の系統性を見出すためでもある。

分類項目としては、まずはPC(タブレットを含む)を使用するかしないかで分けて、そのあとにPC単独利用か、キット等を利用するかで細目を設定している。本来は、授業の成果物として、どういったプログラムを制作し、どのような能力を育成するかの視点での分類が望ましいと考えたが、現時点ではプログラミング授業における評価方法も研究途上であるため、形態別の分類としている。

5.4 実践上の課題

小学校でのプログラミング教育の普及・推進のために、事例1~9の授業から、実践に至るまでと授業時の問題点や課題を抽出した。以下は、授業者から聞き取った課題を5つに集約したものである。

5.4.1 設備環境面の問題

事例3, 6, 9は当然ながら、コンピュータ以外にキットやロボットが必要となる。はじめての指導者にとって、ただでさえ心理的負担感のあるプログラミングの授業において、各キットの準備負担(充電や組み立て、接続設定等)まで課せられることで、実践のハードルがまた1段階あがるという。また、事例3のボー

ルロボットでは使用台数が増えると大きな面積が必要となり、そのコースの準備にも時間がかかる。プログラミング授業においては、プログラムの原理・操作を学ぶ以外に理科の実験準備と同等かそれ以上の設備環境面・場の設定などへの配慮事項が必要とされる。但し、事例6のようなチュートリアル型のプログラミング学習では、その点が軽減されるため、授業の趣旨や意味付けをしっかりとおこなう前提で、導入期に最も進めたい事例である。

5.4.2 時間数の問題

事例1, 2, 9は、児童らがアイデアを発想して創作的なプログラムを作成するため、自ずと授業時間が長くなる。時間不足は、児童らの理解が遅いのではなく、次々と新しい機能を見つけては、プログラムの内容を凝り出すことから生じる。指導者側は意欲的な児童の学習活動の時間を確保したいと考えるため、活動を遮ることができず、予定時数では終わらない状況があるという。そのため、制作作業を家庭学習に委ねるなどして、時間を確保することは今後の大きな検討課題であるといえる。

5.4.3 指導者のプログラミング能力と指導力

事例1, 2, 9のプログラム創作型においては、「こういうことがしたい」という児童の高度な要求・質問

表1 小学校におけるプログラミング授業の分類例(※表中下線太字は事例1~9を示す)

	システム	型式	具体的な名称	主な創作物・学習活動	補足説明	
PCや情報端末を利用したプログラミング学習	ソフトウェアのみ	汎用的なプログラミング言語(完成プログラムを単独動作可)	コードを入力・配置形式	・Moonblock ・Swift	ゲームからアニメーション、実用プログラムまであらゆるものが作成可能。	MoonblockはJavaScriptへ変換でき、Swiftはアプリとして単独実行が可能となるため、汎用性がある。
		プログラミングを学ぶための教育向け	チュートリアル型(出される課題に対して適切なコード・コマンドの配置をおこなうもの)	・Hour of code ・アルゴリズム ・ プログル ・CodeMonkey ・ Swift Playgrounds	ドリル形式のように、簡単な問題から順番に出題され、解答となるコードを配置(入力)していけばステージをクリアしていき、完成するプログラムはほぼ同じものとなる。	CodeMonkeyはコード入力もできるがボタンでの入力も可能。他はマウスドラッグのみで操作可能。正答の自動判定機能がついているため児童だけで進行可能。
			コード・コマンド配置型	・ Scratch ・ プログラミングゼミ ・ Scratch Jr. ・Viscuit	基本的には、キャラクターを配置して動作を与える、制御することが主要な使い方となるが、ゲームや学習、実用プログラム等も作成可能。なお、Scratch以外の4つは低学年から利用できることを特徴としている。	Scratchはイントール版もあり。プログラミングゼミはチュートリアルも内蔵している。Scratch Jr.とViscuitは「記号」のみでプログラムを組める。
	電子キットとの併用	キット・ボード系	ワンボードマイコン	・Raspberry Pi	これ自体がシングルボードコンピュータであり単独で利用可能。PCの原理やそれぞれのインターフェイスの特徴等を学ぶことができる。	Scratchやその他のプログラミング言語を標準で搭載している。
			マイコンボード	・Micro:bit	ボード上に光・温度・加速度等のセンサーを搭載しており、それらを使って計測・制御でき、25個のLEDで各種表示が可能。	PC上で作成したプログラムをボードに転送して実行する形式。
			電子タグ・電子キット	・MESH ・ アーテック(キット)	MESHはブロック形状の電子タグを利用して、IoTを仮想的に体験・学習することができる(制御はタブレットの専用アプリにて)。アーテックのキットは、様々なセンサーやモーターを組み合わせて、信号機や自動ドアなどに見立てたものを制御できる。	アーテックのキットは組み立てるところから電子工作の学習となっている。専用のソフトウェアはScratchの改良版=StuduinoをPCやタブレットにインストールして利用。
	ロボット系	ロボット系	自走式(センサー制御機能なし)	・ Sphero シリーズ ・Airblock	タブレット等から動作プログラムを送り込むことで自動で動く(リモコンモードもあり)。ブロック形式の命令を与える専用アプリで動作。	Spheroはボール型で回転しながら自走する。Airblockはいわゆるドローンタイプの組み立てキット。
			自走式(センサー制御機能有り)	・マインドストーム EV3 ・mBot ・Ozobot ・プロロボ	動作命令を与えて動かすことに加えて、各種センサー類がついており、計測・制御によって条件付きの動作をさせることができる。	Ozobotは主にライトレース型。プロロボは接触センサーを備える。mBotやマインドストームEVは多数のセンサーを組み込める。
			人型	・ Pepper ・アーテックロボアドバンス	各種センサーを搭載し、人間的な動作ができる。特に、Pepperには音声認識機能やタッチセンサーもあり、人間の双方向のやりとりができる(クイズや地域紹介等は容易に作成可能)。	アーテックロボは組み立てキットで各種のセンサーを組み込める。PepperはChoregraphe(コレグラフ)という専用ソフトウェアで命令を与える。
	アンブラ	・ フローチャート作成 ・コンピュータの理解 ・動作化	例えば、掃除や給食配膳などのフローチャートで示し、効率的に実施するにはどうすればいいのかといったことを検討するといった学校生活を改善する取り組みや、フローチャートを算数の場合分けにも応用している事例もある。また、ドット絵を使って2進数の理解をしたり、文字コードを使って暗号電文を送るといった例もある。なお、ダンスのパートを1つのコマンドに見立てて実際に踊るといった事例もあり、比較的、児童らにも指導者にも親和性が高い取り組みが多いといえる。			

※その他、プログラミングを学ぶためのチュートリアル型/コード・コマンド配置型として、Google Blockly (Google for Education)、Smalruby、Squeak eToysなど数多くのサービスが無料で公開されている。

に比べられないもどかしさを感じる人が多いという指導上の課題が出された。ただ、最も重要なことは、使用しているプログラミングソフトウェアで「できること」と「できないこと」の判断がつけられるかであるという。ビジュアルプログラミング言語は「できること」は非常に分かりやすいし、児童らに基本操作を教えれば、あとは児童同士で発見していくことも多い。しかし、できないことをすぐに見極めることは難しいため、その点でも指導者の役割は重要であるともいえる。

一方、チュートリアル型は、もともと児童に分かりやすいように、自主的に進められるように作られている。よって、指導者は取り組む前の目的の伝達やルールの徹底、授業後の振り返り等に力点を置く必要がある。事例5, 6, 7では、なんのためにこれを実施するのか、終えたあとは何を学びとして認識させるのかといった点に注力している様子が見えられた。なお、事例3では、単にゴールを目指すだけではなく、試行錯誤の様子を評価していた。

5.4.4. 授業の系統性

現状での最大の課題ともいえるのが、授業の系統性をどのように校内で計画・策定していけるかによるという。これは、担任裁量に任された場合に、系統性が考慮されずに、指導内容の逆転現象や重複が生じる可能性があるということにもつながる。

例えば、ここでの9事例が同じ学校で実施されたと仮定すると、事例1（6年生）は順次処理のみの扱いである一方で、事例7（5年生）では「関数」までを念頭に置いているし、事例9（4・5・6年生）では計測・制御までを実施している。また、事例8（5年生）ではフローチャートを描いているが、事例3（3年生）はボールロボットを制御していたりする。やはり、扱う処理内容での逆転現象が生じることとなり、当然ながら重複して実施されることも起こりうる。特にチュートリアル型のプログラミング指導を学年を超えて2度実施したり、ボールロボットのゴールへの誘導実践を何度もおこなうということはあまり考えられないため、発達段階を考慮した系統性の確保は最重要課題であるといえるだろう。

5.4.5. 指導内容の到達度の設定

命令を与えるという直接的な動作は理解しやすいが、「変数」と「関数」という概念の理解までいくと指導が困難となることが挙げられる。

特に、関数においては、同じ動作をおこなう機能を1つのFunctionにて定義しておくことになるが、抽象性が増すために指導が困難であるといえる。概念的・言語的発達段階における精査をおこなったわけではないが、各事例の指導者らの実感としては、小学校段階

におけるプログラミングの必須事項として「順次処理・条件分岐（計測・制御を含む）・繰り返し」を定め、変数・関数の扱いは発展的事項として扱うことが無難ではないかとの認識であった。「順次処理・条件分岐・繰り返し」は日常の生活でも置き換えられる場面が多く指導が具体化しやすい。しかしながら、変数や関数といった視覚的に示しづらい内部処理的なことを細かく指導することは困難であると捉えられていた。

5.4.6. 評価の困難さ

チュートリアル型は一定の手順の相違はあるが、ゴールが明確である。しかし、全員一律の活動となるため評価が難しい。達成ステージの進捗状況や、そこから何を気づきとするのかを感想文で書いてもらうといったことで評価をしている場合もあった。しかし、自由記述においても、プログラミング的思考につながる成果が得られるとは限らない。「はじめて取り組んだが面白い。クリアできて嬉しかった。難しかったけどきちんと説明を読んで協力しながら達成した。」など心情的な言葉が並ぶ。自由記述からもプログラミング的思考の獲得につながった箇所を見出すための工夫が必要であるが、現時点では個々の評価方法は研究途上にあるといえる。

但し、「【知識・技能】（小学校）身近な生活でコンピュータが活用されていることや、問題の解決には必要な手順があることに気付くこと」の目的に関しては、自由記述でも随所でその気付きが見られた。これらは各種のプログラミング体験ですぐに表れる成果として把握しておく必要があるといえる。

6. おわりに

最後に、改めてプログラミング授業実践について留意すべき点について述べておきたい。それは、「プログラミング教育」という新しい教育のカテゴリが生まれたわけではなく、「情報活用能力の育成」においてプログラミングの指導が拡充されたことと捉える必要性についてである。

平成32年実施の小学校学習指導要領においては、「第一章 総則 第2 教育課程の編成」の「2 教科等横断的な視点に立った資質・能力の育成」で、「(1) 各学校においては、児童の発達の段階を考慮し、言語能力、情報活用能力（情報モラルを含む）、問題発見・解決能力等の学習の基盤となる資質・能力を育成（以下省略）」とあり、ここではプログラミングという言葉は出てきていない。そして、同指導要領の「第3 教育課程の実施と学習評価」の「1 主体的・対話的で深い学びの実現に向けた授業改善」においては「(3) 第2の2の(1)に示す情報活用能力の育成を図るため、」の記述があり、情報活用能力育成の手立てが記

述された上で、「あわせて」という接続語の次に「各教科等の特質に応じて、次の学習活動を計画的に実施すること。」として、「イ 児童がプログラミングを体験しながら、コンピュータに意図した処理を行わせる」とある。

つまり、「プログラミング教育」という新たなカテゴリが設けられたわけではなく、従来の学習指導要領にも記述されてきた「情報活用能力」の一環としてプログラミングが明記されたに過ぎない。従来の解釈でも「情報活用能力」の「情報の科学的な理解」や「情報化社会に参画する態度」の観点においては、今回のプログラミングに関する意図が内包されてきた。よって、これまでも、小学校において数多くのプログラミングの先行事例が存在しており、それは電子黒板やタブレットの活用よりも遥かに古いといえる。例えば、松田（1985. 3）の「小学校高学年用 LOGO 教育教材の開発 一 基礎編」や岡森（1984）の算数科でのプログラミング授業まで遡ることができる。当時の授業記録をみても、本質的には本稿で取り上げた事例とは大きな差はなく、次期学習指導要領の趣旨にも通じる内容であることが確認できた。特に、岡森論文での実践事例は、当時の 8 ビットパソコンが使われているが、小学校における算数科において、教科指導内容と関連させつつ、コマンドを記述しながら図形を描画したり動かすものであり、現在でも参考にしたい事例といえる。

各種教育雑誌や報道等では「小学校でプログラミング教育がはじまる」とのセンセーショナルな見出しをつけている。確かに学習指導要領にて必修化が明示されたことについての影響は大きいといえるが、改めて考えてみれば、従来の情報教育（情報活用能力の育成）の観点であることに変わりはない。

つまり、プログラミング教育だけが独り歩きすることが無いよう、学習指導要領におけるプログラミング関係の抜粋部分だけを読み解くのではなく、その前後の文脈からも解釈する必要がある。現に、指導要領の解説編では、情報活用の実践力、プログラミング指導、情報モラル指導の順に各 1 ページ程度の分量でバランスをとって記述されている。なぜプログラミング指導を実施するのかという問いに対しては、やはり「情報活用能力」という上位概念からの視点で俯瞰的に捉え

直す必要があると思われる。

先の事例 1～9 は、いずれも ICT 活用の研究校であり、プログラミング以外の基本的な PC 操作スキルや主体的に調べる・まとめる・伝えるという活動もできる児童達である。情報活用能力育成の基盤が無い中で、プログラミングの授業が単独で実施されてもその成果があがらないのではないかという懸念もある。小学校でのプログラミング授業の普及には数々の課題も多いことが挙げられたが、2018, 2019 年度の「移行期間」には、今一度、年間の指導計画を見直して、情報活用能力の体系的な位置付けを探ることからはじめるべきではないだろうか。

引用資料（注）

- 1) 世界最先端 IT 国家創造宣言（平成 25 年）、首相官邸、<https://www.kantei.go.jp/jp/singi/it2/decision.html>
- 2) 日本再興戦略 2016 一 第 4 次産業革命に向けて（平成 28 年）、首相官邸、<https://www.kantei.go.jp/jp/singi/keizaisaisei/kettei.html#saikou2016>
- 3) 小学校段階における論理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議（平成 28 年 6 月）、小学校段階におけるプログラミング教育の在り方について（議論の取りまとめ）
- 4) 文部科学省、情報活用能力調査（平成 25 年）
http://www.mext.go.jp/a_menu/shotou/zyouhou/1356188.htm
- 5) 文部科学省（平成 29 年 6 月）小学校学習指導要領解説 総則編 p.85

参考資料

- ・松田 稔樹、坂元 昂（1985）、小学校高学年用 LOGO 教育教材の開発 一 基礎編、東京工業大学人文論叢（11）、p91-106
- ・岡森博和・柳本朋子・本間 俊宏・西谷 泉・金谷 博史（1984 年 12 月）、数学教育におけるコンピュータ・プログラミングの指導について（Ⅲ）：LOGO 言語を中心として、大阪教育大学紀要 第 V 部門、33（2）、p131-148
- ・丸山幸三、小学校教育課程におけるプログラミング教育の考察 一 プログラミング教育に最適なプログラミング言語 一、豊岡短期大学 紀要 論集第 11 号（平成 26 年）p11-19