

小中学校における〈普通教育としてのプログラミング教育〉の展開と課題

紅林秀治

静岡大学

〈普通教育としてのプログラミング教育〉とは

現在、日本では、中学校技術・家庭科（技術分野）と高等学校情報科においてプログラミング教育が行われている^{1), 2)}。しかし、小学校においては、一部の先進的に取り組んでいる学校以外は、ほとんど行われていない³⁾。一方、諸外国においては、すでにプログラミング教育を含む情報教育の低年齢化が進んでおり、英国では小中学校（5～16歳）でのプログラミング教育を2014年から必修としている⁴⁾。そのため、日本においても、小学校1年生からプログラミング教育を実践できる可能性は十分にある。しかし、プログラミングは、単なる知識ではなく、国民に必要な教養であるとの見方もあり、小中学校のプログラミングの授業に対して、普通教育として行う意義を明確にしなくてはならない。

普通教育を簡潔に理解するためには、その対極にある専門教育の理解が欠かせない。たとえば、小中学校のプログラミングの授業に対して、専門教育的な見方で教育的な価値を見いだそうとすると、作成しているプログラムの内容やロボット教材の動作については、陳腐でまったく役に立たないもののように見えてしまう。なぜなら、専門教育は、学習者が将来その道の専門家になることを想定した教育だからである。そのため、専門教育を受ける学習者は自分が就きたい職種が明確であり、教員はその職に就くために必要な知識や技能、およびそれらの評価規準を把握できている。

それに対して、普通教育は、将来何になり、ど

のような職業に就くのか決めていない学習者を対象に教育をする。そのため、学習後に獲得してほしい知識や技能、およびそれらの評価規準は、職に就くためとか、想定される仕事に役立つためなどという観点からは設定できない。つまり、考え方や問題解決能力、あるいは学習対象の価値に気づくといった汎用的な能力や態度の育成を教育目標として掲げることになる。しかし、プログラミングの考え方や問題解決能力、またその価値に気づくことを目標とすると、普通教育としてのプログラミングの授業ではいったい何を教えるべきなのか検討する必要がある。

私が見てきた〈普通教育としてのプログラミング教育〉の展開

ここで〈普通教育としてのプログラミング教育〉の在り方を考えるために、私が見てきた小中学校での授業を紹介する。

□ 小学校1年生にスクラッチを用いたプログラミングの授業

授業の目標は、「プログラミング学習を通して、『プログラミング的思考』を育成する」である。具体的には、複数のコマンドを使用しないと解決できない課題を児童に与え解決させる流れである。プログラミング経験がまったくないという児童の実態や、1年生という発達段階を踏まえ、まずは、マウス操作やキーボード入力（かな）の練習から授業を始めていた。PCの基本操作を学習した後、スクラッチを用いたプロ

グラミングの学習を行っていた。「スクラッチキャットをステージの左上から右下に動かす」という課題を教師側から提示し、その動きを実現するために、動きの組合せを考えさせていた(図-1)。

□ 小学校 3 年生に教材用のロボットを用いたプログラミングの授業

イモムシ型ロボット「コード・A・ピラー」を用いて、1時間の授業を行っていた。「コード・A・ピラー」とは、赤ちゃん向けのおもちゃを販売するメーカーが、幼児向けの新世代おもちゃとして開発したプログラミング学習ロボットである。イモムシを作る胴パーツには、直進や右折などの命令コードが内蔵されている。ロボットはそれぞれの命令コードが内蔵された胴パーツをつなぐことにより、イモムシが動作する仕組みになっている。ロボットを教室に1台準備し、教員に与えられた課題(障害物をよけてゴールする)を達成するために、命令コードが書かれた胴パーツをどのように接続するか、グループ(4人から5人で構成)ごとに話し合う。各グループには、矢印が書



図-1 小学1年生のスクラッチでの授業の様子



図-2 イモムシ型ロボットで学習している児童の様子

かれたカード(直進・右折・左折)が数枚ずつ分けられている。児童は、そのカードを使ってロボットにさせたい動作を話し合い、グループごとに意見をまとめる。教員は、グループごとに決めた矢印の並びをロボットで順番に実証していった(図-2)。

□ 小学校 6 年生にドリトルでロボットを制御するプログラミングの授業

クローラ型の移動ロボットの動作をプログラムで作成する。定められたコース通りにロボットを動かすという課題で授業が行われた。プログラムは、ドリトルで作成し、ロボットに転送して実行する。小学校6年生くらいになると、キーボード操作にもローマ字入力にも慣れていて、無理なくプログラムを作ることができていた。

□ 中学 2 年生の Pepper を用いたプログラミングの授業

この授業は、Pepperという人型ロボットを動作させるプログラムをChoregraphe(コレグラフ)というソフトウェアを用いて作成するものである。Choregraphe(コレグラフ)では、必要なロボットの動作やセンサを用いるためのプログラムがすでに揃っており、生徒はそれらを選択してつなげる作業を行う。授業では、生徒からの質問をPepperに答えさせるためのプログラムを作ることが課題として与えられていた(図-3)。



図-3 Pepperを用いた授業の様子



□ 中学3年生の制御基板を用いたプログラミングの授業

授業で使用した制御基板は、計測値や出力値等をLCDで確認できるものである。また、その制御基板はブレッドボードを利用して自由に入出力回路を作成できる。さらに、制御プログラムにはドリトルを用いているため、制御プログラムをドリトルの編集画面上で作成した後、USB接続した基板に転送する。基板にはアナログセンサを4個、デジタル出力を6個まで接続できる。計測はPCから独立して行うことができるものであった。LEDの点灯回路やCdsセルを利用したアナログ計測を学習した後、オリジナルなプログラムを作成し発表するという課題を与えられていた(図-4)。

〈普通教育としてのプログラミング教育〉が定着する上での課題

これらの授業では、教員が児童・生徒にサンプルプログラムを示し、そのサンプルプログラムから基本的な知識や技能を獲得する。その後、サンプルプログラムを基に、応用できる課題が教員から出されて取り組むという内容であった。

授業を見学した教員から感想を聞いてみた。さまざまな感想の中から、あえてマイナスな評価の感想の代表例を紹介する。

- (1) 結局これは何を学んだことになるの？ (小学校校長)
- (2) プログラミングって手続きを学ぶことなの？ (小学校教員)



図-4 制御基板を用いた計測している生徒の様子

- (3) 教材を準備するお金と時間がない。(中学校教員)
- (4) これを学んで何かの役に立つの？ (小学校と中学校教員)
- (5) 試行錯誤をくりかえしていたら、何かできてしまうということが本当に教育としてよいのか？ (小学校と中学校教員)

このような教育としての有用性を疑問視する感想は、小学校の英語学習の授業では私が見た限りでは聞いたことはない。私が見た小学校の英語の授業では、基本的に会話を中心に展開されている。ある授業では、児童は英語の言い回しや簡単な単語を覚えてから、教室内の仲間と自己紹介したり、自分の伝えたいと思っていることを教員から紹介された言い回しを用いて話したりしている。授業中の活動を通じて、児童は新しく覚えた英語を使ってみることに喜びを感じている様子が見られた。

プログラミングの授業では新しく覚えたコマンドやプログラムを使って画面上のキャラクターの動作やロボット等の動きを試している。小学校英語の授業も小・中学校プログラミングの授業も基本的に学習の流れはよく似ている。小学生も中学生も新しく覚えた知識を使ってみたい、試したいと思う気持ちは同じである。

しかし、プログラミングの授業では、なぜか有用性を疑問視する声が聞こえてくる。その原因の一端は、学習しているプログラミング言語や学習用ツールが世の中の職業に使われているのか、あるいは社会生活に役立つものなのかという価値基準で、児童・生徒が作ったプログラムを見ていることに原因があると思われる。つまり、普通教育ではなく、専門教育としての規準で評価しているのである。これは、普通教育としての授業が他教科では成立しているが、導入されたばかりのプログラミングの授業では成立していないことを意味しているのではないだろうか。

加えて、普通教育としてプログラミングを行う場合、学習対象(ここでは、教育用言語や学習用ツールを使用して行うプログラミングそのものを指す)

に学習者が価値を見いだすことが重要である。それは単に「プログラミングって面白い」という感想を学習者が述べるものでなく、「もっと勉強してみたい」、「もっと作ってみたい」といった、学習対象に向き合っていこうとする能動的な態度や、学習対象に向き合うことで生じてくる問題を自ら考えたり、解決したりする能力に裏付けられた意欲や態度に表れるものである。そのような意欲や態度が、専門的に学んでみたいと思ったり、プログラミングにかかわる仕事に敬意を持ったりすることに繋がる。

その意味で、私が見てきたプログラミングの授業において、先に示したような教員の感想が出てきたもう1つの背景には、サンプルプログラムからの学びから、自ら課題や問題に気づき、能動的にプログラミングに向き合うといった児童・生徒の学習場面がほとんど見られなかったことがあるのではないかと考えている。

〈普通教育としてのプログラミング教育〉の価値と可能性

児童・生徒は、好奇心が旺盛であるため、プログラミングの授業で覚えたコマンドやプログラムを試してみたいと感じる。そのため、プログラミングの授業では、サンプルプログラムを動かしているだけでも、喜んで授業に取り組む。これは、小学生の場合特に顕著である。しかし、サンプルプログラムを学習するのみで授業を終えるとなると、それは専門教育でいうところのトレーニング的な指導とさほど変わらない。このような学習を繰り返していたら、児童・生徒はプログラミングの授業に価値を見いださないであろう。

プログラミングの授業がトレーニング的な指導の形態にならないためには、授業で覚えたコマンドやプログラムを使って、児童・生徒が作りたいと思うものをプログラムで実現する授業が必要である。なぜなら、プログラミングは、作成者の思いや構想を実現する手段であり、何かの役に立つことを学ぶというような目的ではないからである。また、自らが

作りたいと考えるものを、プログラミングで実現したときこそ、プログラミングの面白さや、学ぶ意義や価値を実感できるからである。加えて、その作ったものが他人に評価されたり、喜ばれたりしたならば、その思いはさらに強くなるであろう。このようなことは、クライアントの要求に悩まされたことのあるシステムエンジニア、または実験や研究のためにプログラムを作成した経験のある人なら誰でも分かることだろう。したがって、サンプルを通じて習ったコマンドやプログラムを学習した後、自分が作りたいもの考える機会を与えることが、プログラミングの授業では重要になる。

とはいえ、自分が作りたいもの考えるという課題を児童・生徒に与えることは簡単なことではない。ところが、ある中学校での実践事例の中に、児童・生徒が自ら作りたいもの考え出すヒントがあった。その実践では以下の流れで授業を展開していた。

- (1) 技術科の授業で制御基板を使った回路とプログラムを作成した。
- (2) 多くの生徒は、サンプルを基にLEDの点滅や音が鳴る回路を作成した。
- (3) 教員は、生徒に作品を家に持って帰り、家の人に見せ、家の人に感想を用紙に書かせる宿題を出した。
- (4) 生徒の多くは、家の人に作品を見せ、感想を求めた。家の人からは「よくがんばった」と感心されたが、同時に作品に対する修正要求も出された。
- (5) 生徒たちは、教員にプログラムを修正したいと願いだした。
- (6) 教員は、授業で生徒に修正案を書かせ、プログラムを修正させ、その成果を発表させた。

授業は、大いに盛り上がった。授業が盛り上がった理由は、改善そのものを生徒が習った知識と技能を使って実現し、親の要求に応えたオリジナルな作品を作ることができたからである。そのため、プログラミングの授業に達成感が持てたと考えられる。このように、作りたいものや実現したいものを自覚できたと同時に、プログラミングで実現できるとい



う見通しを持てたとき、あるいは実際に作成できたとき、生徒はプログラミング学習の面白さに気づき、プログラミング学習の意義や価値を見いだすことをこの授業は示していた。

この授業の中で、生徒に作りたいという思いを抱かせたのは、親から修正要求が出た場面である。それは、PDCA サイクルでいうところの A (アクション：改善) の場面である。生徒いきなり P (プラン) を求めるのではなく、サンプルプログラムを用いた授業から始め、できた作品の C (チェック：評価) を行い、A (アクション：改善) の場面を設定する。それにより、生徒は改善すべき問題を見だし、習ったプログラミングの知識や技能を使ってどうやって解決するか考え始める。生徒は、ここで学習対象に能動的に向き合ったことになる。そして、このとき生徒の P (プラン) の場面が始まる。これは、D (Do) から始め、P (プラン) に続く授業形態である。このような流れの授業は、プログラミングを「目的を達成する手段」として捉え、自分の P (プラン) によりプログラムを作成する学習となる。このときの生徒は、まさに問題解決に向けた取り組みを始め、改善できた暁には、プログラミングの面白さや学習する価値を感じ始めることであろう。このような学習では、扱うプログラミング言語は何でもよい。児童・生徒の発達段階に合わせた学びやすさが保証された言語や学習ツールならば問題なく実施できるであろう。

〈普通教育としてのプログラミング教育〉の今後に向けて

私は、プログラミングを「目的を達成させる手段」と捉えさせることが普通教育としてのプログラミングの授業の中では重要ではないかと考えた。ところが、初めてプログラミングを習う児童・生徒に対して、最初から「プログラミングは手段である」と説明しても授業は成立しない。そこで、教育界でよく使われる PDCA のサイクルを応用した D (Do) から始める授業を提案する。初めてプログラミングを学ぶ場合、サンプルプログラム通りに打ち込むことから

始める学習は大切である。しかし、サンプルプログラムを教える学習にとどまるならば、多くの児童・生徒はプログラミング学習に価値を見いだすことはないだろう。しかし、サンプルプログラムを基に解決するプログラミングの経験を与えることができるならば、児童・生徒が自ら学習対象に向き合う (P (プラン) を立てる) 授業が生まれてくる。サンプルプログラムを使い、まずは教員の指示通りに作業する授業から始めても構わない。しかし、その後児童・生徒がサンプルを通して学んだプログラミングを手段として、達成したい目的に向かって取り組む授業を構成する工夫が大切である。

最近の児童・生徒は自ら問題を発見できないと嘆く教員をよく見かけるが、私が提案する、D (Do) から始めて P (プラン) に至る授業を構成すれば、自ら考え問題を発見し始める児童・生徒は増えるはずである。また、この授業の流れでは、教員の工夫次第でさまざまな応用もできる。さらに、児童・生徒だけでなく、教える教員もプログラミング教育の価値を見いだす取り組みになるのではと期待している。

参考文献

- 1) 文部科学省：中学校学習指導要領解説 技術・家庭編，教育図書，pp.32-37 (2008).
- 2) 文部科学省：高等学校学習指導要領解説 情報編，開隆堂 (2010).
- 3) 森 秀樹，杉澤 学，張 海，前迫孝憲：Scratch を用いた小学校プログラミング授業の実践，日本教育工学会論文誌 34 (4)，pp.387-394 (2011).
- 4) 上松恵理子：小学校にプログラミングがやってきた！，三省堂，pp.91-92 (2016).

(2018年11月15日受付)

紅林秀治 (正会員) kurebayashi.shuji@shizuoka.ac.jp

静岡県内公立中学校，国立大学附属学校，工業高校教諭を経て，2005年より静岡大学教育学部助教授。2009年より教授。専門は技術教育。日本産業技術教育学会理事。博士 (学校教育学)。2017年より静岡大学教育学部附属浜松中学校校長を兼務。