

## プログラミング教育における評価基準の検討 Evaluation Criteria in Programming Education

米川 雅士<sup>†</sup> 中村 健二<sup>†</sup>  
Masashi Yonekawa Kenji Nakamura

### 1. はじめに

2020年4月から小学校では学習指導要領の改訂により、「総合的な学習の時間」においてプログラミング教育が必修化された。今までにない科目のため2018年より文部科学省では様々な取り組みとして推進事業を行い、特に総務省、経済産業省と共同で「小学校を中心としたプログラミング教育ポータル」[1]を開設して、授業で実施可能な事例の紹介と運用、授業外での学び、他教科との横断的な学び方など教員向けに様々な情報提供を行っている。

小学校で学ぶプログラミング教育は学習指導要領の中では大学で学ぶような変数、関数などを駆使し、テキストを使ってプログラミングを学ぶのではなく、プログラミングを体験しながら論理的思考力を身に付けるための学習活動を実施しながらプログラミング的思考を身に付けることを目的としている。しかし、このプログラミング的思考をどのように身に付けたと判断するか、どのような指標をもって評価をするのかなど評価基準があいまいな部分がある。そこで本研究ではプログラミング的思考とプログラミング能力を比較することで、小学生が将来的にプログラミングを行う際に必要となるプログラミング的思考とはどのようなものなのか考え、現在検討されているプログラミング教育で使われている教材について検討することとした。

### 2. プログラミング的思考

プログラミング的思考とは、図1に示すように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み合わせが必要であり、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、記号の組み合わせをどのように改善していけば、より意図と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[2]と文部科学省では定義している。

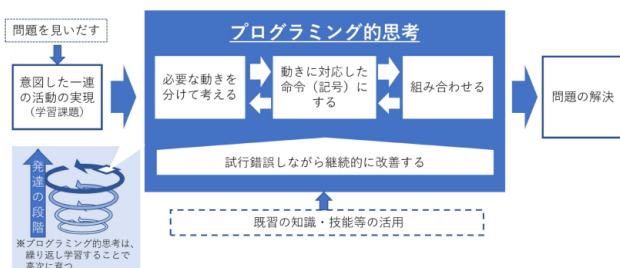


図1 プログラミング的思考

このプログラミング的思考をプログラミング教育教材ではどのように学ばせようとしているのか今回は代表的な以下の3種類について説明する。

<sup>†</sup> 大阪経済大学

### 2.1 アンブラグドプログラミング

コンピュータを思い通りに使えない状態でプログラミング教育を実施した場合、コンピュータとプログラムの両方に苦手意識を持つ、これが小学生に対してなら経験を積んだ大学生以上に強い苦手意識になる恐れがある。そのようなことにならないようにコンピュータを使わずにプログラミング的思考を学ぶ手法がアンブラグドプログラミングである。このアンブラグドプログラミングが有名になったきっかけがフィンランドのプログラマーが書いた書籍「ルビィのぼうけん」シリーズである。

アンブラグドプログラミングでは自分が経験したことを漫然と実施してきた行動に対して論理的思考を身に付けるために細かく手順を考え並べ、コンピュータを使わないことでグループワークとして実施することが向いており、選択する教材によっては学習指導要領で「総合的な学習の時間」に求められている教育用の枠を超えた積極的・総合的な学習と同時に探究的な学習や共同的な学習を実施することも可能である。

### 2.2 ロボット・ブロックプログラミング

世界中の様々な企業がプログラミング的思考を学ぶためにロボットが販売されている。販売されているロボットによって様々な特徴はあるものの基本的には事前に何か目的を設定し、その目的を達成させるために用意されているパーツでロボットを自分で設計・作成し、ロボットに対して命令を与える事で目的を達成させることができる。対象年齢が低いロボット・ブロックプログラミングではロボットを組み立てる仕組みが簡略化や無かったり、ロボットへの命令がカードなどで簡略化されている。

ロボット・ブロックプログラミングでは制作者が目的達成を目指して試行錯誤することが可能である。これは学習指導要領にある探究的な学習を体現しながらプログラミング的思考を学ぶことを可能としている。設定された目的に対し、情報を収集し目的を達成するための分析ののち自分が設計した最適なロボットを作成するという体験ができ、目的達成までトライ&エラーを繰り返すことで最終的に目的を達成する。

### 2.3 ビジュアルプログラミング

Webアプリケーションに近いイメージでブラウザを使って視覚的にプログラミングを実行する。実行できる種類はプログラミング的思考を必要とするアプリケーションが用意されており実行していくタイプか、ビジュアルブロックを組み合わせることで自分が思考したアプリケーションを作成するタイプに大別できる。両タイプとも基本的にはドラック&ドロップのような簡単な操作のみで実行が可能でキャラクターを動かしたり、ゲームを作ったりすることが可能である。また、最大の特徴としてPCやスマートフォ

ンなどインターネットを利用できる環境下であれば無料で利用することが可能である。

ビジュアルプログラミングは本章で説明した 3 種類の中で最もテキストで実施するプログラミングに近く、殆どのビジュアルプログラミングでは日本語での利用が可能のため関数の知識が無くてもプログラミングのように思考して操作することで自然とプログラミング的思考を身に付けることが可能である。また、2.2 節で説明したロボット・ブロックプログラミングのうちロボット作成はできないが、その他のプログラミングの思考を学ぶ手法については再現されている。

### 3. プログラミング的思考とプログラミング能力

プログラミング的思考を身に付けるために様々な育成すべき資質・能力や学習到達目標について研究が行われている。大森氏らの先行研究 [3] によれば、プログラミング教育を実施している小学校 6 事例を対象として研究を実施しており、重要なことは単年ではなく 6 年間のカリキュラム・マネジメントによりスパイラルアップ的に繰り返し体験することがプログラミング的思考を身に付けさせるには重要であると論じている。教員として大学生を指導する場合に私自身も先行研究と同様に期間は少し短い 2 年間のカリキュラム・マネジメントを考え先行研究と同様な考えのもと教育を実施しているが、自分が SE を行っていた時代を考えるとプログラミング的思考を意識して学んでいた記憶が希薄である。よって、最初にプログラミング的思考とプログラミング能力の関係性について調査した。

#### 3.1 プログラミング能力の定義

私はよく学生に「プログラムを作成できるか」と質問をすると多くの学生は「基本的な物なら」、「できません」のどちらかの回答が返ってくる。しかし、実際にプログラミングをさせてみると「基本的な物なら」と答えた学生ではプログラミング能力に大きな開きが、「できません」と答えた学生でも簡単なプログラムなら構築できる学生が何人かいる。このようにアンケートでは学生の意識のずれによる偏りがあると考え、大学生 33 人に表 1 で示した 3 種類のプログラムを作成させ、プログラミング能力のレベル分けを実施した。なお、対象学生 33 人は学習による偏りがないようにするために全員がプログラミングに関する同じ授業を受けている。

表 1 プログラミング能力レベル分けプログラム内容

レベル	人数	内容
なし	3 人	何も作成できなかった。
初級	8 人	キーボードから入力した文字を画面に表示させる。
中級	18 人	1 からキーボードで入力した数字までの合計値を計算して画面に表示する。
上級	4 人	1 から 100 までの数字を使いランダムに足し算計算を学ぶためのアプリケーションを作成する。

#### 3.2 プログラミング的思考の確認

各レベルの学生に対しプログラミング的思考を学んでもらうために Google が無料で提供しているビジュアルプログラミング Google Blockly を使い 8 種類のゲームを実施してもらった。それぞれのタイムと達成度を測定してもらい、その結果を表 2 に記述する。なお、長時間経過した場合にはスキップウィンドウが自動で表示されるため、その場合はスキップしてもよい旨も通知してある。

表 2 プログラミング的思考の学びの結果

レベル	トータル平均時間	達成度
なし	4 時間 40 分	88%
初級	3 時間 54 分	92%
中級	3 時間 30 分	94%
上級	3 時間 28 分	94%

#### 3.3 アンケート

プログラミング能力とプログラミング的思考についての確認が終わった後に今回の検証についてアンケートを行った。アンケートの質問項目は大きく分けて「Google Blockly の難易度と疲労度」「Google Blockly の実施状況」「PC の利用頻度」の 3 項目について 5 件法で実施した。

#### 4. おわりに

今回はプログラミング的思考を学ぶ教材の中でビジュアルプログラミングを選択した。その結果プログラミング能力が高い方が若干理解度が高い結果になったが、基本的には殆どの学生が高い達成度で学んでいることがわかった。また、トータルの平均時間を確認すると、ここではプログラミング能力の差によって最大 1 時間以上の開きがあることが分かった。この結果からプログラミング能力が高い学生はプログラミング的思考が早い傾向がわかった。また、アンケートにおいて Google Blockly 関連の質問に対しては全学生でほとんど差はなかったが、PC の利用頻度については違いが顕著に出た。レベルが「なし・初級」の学生は家に PC はあるが全く使っていないことが分かった。なお、中級・上級の学生はほとんどの作業を PC で実施し、1 日に数時間は PC を利用していることが分かった。これらの結果から、大学生ではプログラミング的思考は日々の生活である程度身につけており、プログラミング的思考の処理の速さがプログラミング能力の高さとつながっている。また、アンケート結果から学習成果の指標として、プログラミングに触れている時間を評価基準として新しい指標に取り入れることが重要だと考える。

#### 参考文献

- [1] 未来の学びコンソーシアム事務局, “小学校を中心としたプログラミング教育ポータル”, <https://miraino-manabi.jp/>.
- [2] 文部科学省, “小学校プログラミング教育の手引き (第三版)” [https://www.mext.go.jp/content/20200218-mxt\\_jogai02-100003171\\_002.pdf](https://www.mext.go.jp/content/20200218-mxt_jogai02-100003171_002.pdf).
- [3] 大森康正, 磯部征尊, 上野朝大, 尾崎裕介, 山崎貞登, “小学校プログラミング教育の発達段階に沿った学習到達目標とカリキュラム・マネジメント”, 上越教育大学研究紀要, 第 37 卷, 第 1 号 (2017).