

プログラミング教育を受ける小学生を対象とした 論理的思考の育成方法

松浦雅子^{†1} 平山秀昭^{†1}

概要 : 2020年に小学校におけるプログラミング教育が必修化される。その狙いはプログラミングの技能の習得にあるのではなく、論理的思考を育むことにあるとされている。小学校でプログラミングに触れ、段階的に論理的思考を育んでいくことを考えた時、その中心的な役割を担うのは当然ながら小学校の教員であるが、それだけでは不十分と考える。我々は小学生以下の子供を持つ保護者等にプログラミング教育を施し、保護者等が補完的に子供に対してプログラミングを教えるという方向を考えている。その際、保護者等に論理的思考の意味は教えるが、子供には論理的思考について説明させたりはせず、子供が小学校教育を受ける長いスパンの中で、論理的思考の意味が子供に伝わる時期や機会を意識して貰い、適切なタイミングでアドバイスしていくようにして貰う。この考えの下で実施した実験教育の結果について報告する。

キーワード : 小学校プログラミング教育, プログラミング教育必修化, 論理的思考

Method for Developing logical Thinking of Primary Schoolchildren who Receive Programming Education

MASAKO MATSUURA^{†1} HIDEAKI HIRAYAMA^{†1}

1. はじめに

2020年に小学校におけるプログラミング教育が必修化される。その狙いはプログラミングの技能の習得にあるのではなく、論理的思考を育むことにあるとされている。プログラミング的思考とは、「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」とされている[1]。

小学校におけるプログラミング教育必修化の狙いにある論理的思考とは何か？ここでは、システム開発の上流工程に位置する要件定義が行える能力と捉えることとする。当然ながら小学生が論理的思考を身に付け、要件定義ができるようになる訳ではなく、将来できるようになるためのきっかけ作りとなることと捉える。要件定義とは、利用者のシステムに対する要求（「利用者はそのシステムで何をしたいのか？」）に基づき、それを実現するために必要な機能等を明確に文書化することである。プログラミング教育を通してコンピュータはプログラムを正確に記述しなければ正しく動作しないことを理解しながら、何かを実現するためには必要な機能を明確化する必要があることを段階的に身に付けていく必要がある。我々は論理的思考をこの様に捉えて議論を進めていく。

2020年を間近に控え、小学校では準備を進めているが、その意図が十分に理解されているかという疑問がある。例えば、実践事例とされる題材[2]には、SCRATCH[3]等のビジュアルプログラミング言語により図形の作図を行うという極めて単純で小さなプログラムが示されている。複雑あるいは規模が大きくなるとプログラムそのものの理解が難しくなるという理由なのだろうが、それで情報技術を専門としない小学校の教員に十分な意図が伝わるのだろうかという疑問が湧く。勿論、ソートやサーチの様な基本アルゴリズムであれば、小さなプログラムであっても論理的思考を育むという意図は通じる。

一般に、単純で小さなプログラムを作成するだけでは、論理的思考というほど大きなものは不要である。プログラムが複雑になったり、大規模になったりする時に論理的思考の必要性が生じる。複雑なプログラムや、大規模なプログラムを開発するには必ず設計が必要になる。論理的思考を発揮するのは設計フェーズである。プログラミングフェーズでは、設計フェーズのアウトプットをプログラミング言語で実装していく、すなわち、プログラミング言語に置き換えていくだけなので、論理的思考を発揮する訳ではない。図形の作図を行うというような極めて単純で小さなプログラムをもって論理的思考について説明しても十分な理解が得られるか疑問がある。

ただし、小学校におけるプログラミング教育必修化の狙いは論理的思考を育むことのみではなく、「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

^{†1} 目白大学メディア学部メディア学科
Department of Media Studies, Faculty of Media Studies, Mejiro University

術によって支えられていることなどに気付く」, 「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」ともされている[1]. よって, 小学校ではプログラミングという情報技術に触れ, 中学校, 高等学校と情報技術の教育を進める中で段階的に論理的思考を育てていくのだと捉えればよいと考える.

小学校でプログラミングに触れ, 段階的に論理的思考を育てていくことを考えた時, その中心的な役割を担うのは当然ながら小学校の教員であるが, それだけでは不十分と考える. 少し前の状況ではあるが平成 27 年に総務省が公表した報告書[4]によると, 学校以外で行われているプログラミングに関する教育を行っているのは非営利目的の組織 (NPO 法人, 法人化されていない組織等) によるものが半数を超えている. 一方, 営利目的の組織 (プログラミング教室等) によるものは3分の1程度である. しかもプログラミング教室の多くは関東, それも都内に集中している. また, プログラミング教室の開校は 2013 年以降に急増していて, 講師はプログラミング経験のない者が多く, 基本的な指導のみを行い, 理系のプログラミング経験のある大学学部生・院生のアルバイト講師がエラー発生時の対応等を行っている. このような講師不足の問題から, シニアのプログラミング経験者の活用や, 家庭の主婦が講師になってちょっと開けるような広がり方を目指す意見も上がっている.

このような状況を鑑み, 我々は小学生以下の子供を持つ保護者等にプログラミング教育を施し, 保護者等が補完的に自分の子供に対してプログラミングを教えるという方向を考えている[14].

その際, 保護者等に論理的思考の意味は教えるが, 子供には論理的思考について説明させたりはせず, 子供が小学校教育を受ける長いスパンの中で, 論理的思考の意味が子供に伝わる時期や機会を意識して貰い, 適切なタイミングでアドバイスしていくようにして貰う. この考えの下で実施した実験教育の結果について報告する.

2. 保護者向けプログラミング教育の内容

プログラミング言語は SCRATCH を選んだ. SCRATCH を選んだ理由は, 以下の通りである.

- SCRATCH は, 小学生にも理解し易いビジュアルプログラミング言語である.
- ビジュアルプログラミング言語は多数あるが, 基本的には, どれも SCRATCH をベースとしている.
- SCRATCH は, ビジュアルプログラミング言語の中で最も普及している.
- SCRATCH はブラウザ上で動作し, 家庭に普通にある

PC でプログラミングができ, 特別な機器を購入する必要がない.

- SCRATCH は, 作ったプログラムをインターネット上で共有する仕組みを提供している.

<実験内容>

実験教育 1

教育期間は 90 分×3 回とした. 体験的な教育は 90 分×1 回であることが多いが, それでは子供に教えられるレベルに到達することは困難であると考えた. 自分がなんとなく理解するレベルではなく, 自分が理解し, 必要に応じて調べながら, 子供に教えられるレベルに教育できる最小時間として 90 分×3 回とした. 3 回の教育の内容は, 以下のようにした.

(1) 第 1 回

まず, 簡単に小学校でプログラミング教育を必修化する狙いが論理的思考を育むことにあることを伝える. ただし, あまり深く考えるのではなく, とにかくプログラミングをやってみることを伝える. そして, 世界で最も普及しているビジュアルプログラミング言語 SCRATCH を使うことを伝える.

次に SCRATCH にユーザ登録する. そして, SCRATCH でスプライト (オブジェクト) をステージで動かすプログラミングを丁寧に教える. SCRATCH の操作に慣れることも含める. ここで作るプログラムは 10 ブロック以内の一塊のプログラムである. できたところで, プログラミングしたというより, 言われた通りに操作して作ったプログラムの意味を相手に説明させる. 説明させることでプログラムの意味を理解させる.

作成した SCRATCH のプログラムが, Java 等の非ブロック型のプログラムならどのようになるのかを示す. 非ブロック型のプログラムでは, 「{」と「}」のバランスが取れている必要があるとか, 条件文は「if」で, 繰り返し文は「while」を使うとか, 事前に文法というものを習う必要があり, 小学生には少し難しいことを説明し, 概略を理解して貰う.

また, コンピュータはプログラミングされた通りに処理を行う. 間違ったプログラミングをすれば間違った処理をする. 正しいプログラミングをすれば正しい処理をする. 人間は曖昧な指示をしても, その意図を解釈してなんとかすることもあるが, コンピュータはそうはいかない. プログラミングした通りにしか動かない. だから正しくプログラミングする, 正しく指示しなければならない. これが論理的思考の基本であることを説明する. わずか 10 ブロック程度であるが, プログラミングを経験した保護者は論理的思考の意味を概略理解できるようになる. ただし, 子供に対しては, その説明は不要であることを伝える.

最後に, これまでとは異なるスプライトとステージを新

たに選んで貰い、これまでやったことと同じプログラムを自力で最初から作成して貰う。これにより、SCRATCH で基本的な動作をさせるプログラムを自力で作れるようになる。[5]

(2) 第2回

第2回では、少し複雑なプログラムに取り組む。10ブロック程度の塊からなるプログラムを2個作る。変数、タイマー、音も使ってみる。スプライトをクリックされた時のイベントも扱う。プログラミングは前回同様に丁寧に教える。できたところで、やはりまだ、プログラミングしたというより、言われた通りに操作して作ったプログラムの意味を相手に説明させる。説明させることでプログラムの意味を理解させる。また、今回も、作成したSCRATCHのプログラムが、Java等の非ブロック型のプログラムならどのようなようになるのかを示す。

第1回、第2回で、言われた通りに操作してプログラムを作れるようになった。しかし、これではプログラミングができるようになったとは言えない。自分で考えて、自分で調べてプログラミングができるようになる必要がある。それができないと、子供に聞かれても解答してあげることができない。それでは、子供にプログラミングを教えることはできない。

そのために、第2回では、作ったプログラムを色々改造してみる。動きを変えたり、速度を変えたり、色々なことを試みる。そして、ブロックパレットに並んでいる様々な機能を見ながら、それを使ってプログラムを改造してみる。これにより、プログラムを自分で改造すること、SCRATCH が持つ機能を自分で調べてみることを経験してみる。[6]

(3) 第3回

最終回では、先に非ブロック型のプログラムを見てから、それがSCRATCH だどのようなプログラムになるのかを見ていく。Java等の非ブロック型のプログラムでの”Hello World”, 変数宣言, 代入文, 演算といった操作が, SCRATCH だどのようなのかを見る。コンピュータが元々は軍事目的の弾道計算のために作られ、ビジネスで使われるようになっても売上計算等のために使われ始めたこと。つまり、コンピュータは、元々は計算を高速に行う機械として開発されたことを理解する。これは、その後、コンピュータが様々な分野で使われるようになったこと、人工知能のようにいわゆる計算目的以外で使われるようになったことを理解し、様々な応用を考える頭を育むきっかけになることを理解して貰う。

最後は、3つのスプライト（フットボーラー、ボール、コウモリ）からなるシューティングゲームを自力でプログラミングして貰う。そのためにシューティングゲームの設

計を示す。最初に基本設計を、次に詳細設計を示す。その上でプログラミングを行なって貰う。かなり難しいが、自分で考えながらやって貰う。最後にできたところまでのプログラムの説明をして貰い、わからなかったところ、うまくいかなかったところを皆で考えながら修正する。

最後まで完璧にできるとは限らないが、自分で考え、自分で調べ、プログラミングのやり方を教える。また、いきなり解答（プログラミング例）を見るのと、自分で悪戦苦闘してから解答を見るのでは、効果が全く異なる。あとは、持ち帰って、自分で復習をして貰う。これで、ようやく、子供に聞かれて解答してあげられるようになる。これで初めて、子供にプログラミングを教えられるようになる。

また、あわせて、最初に示した設計にこそ論理的思考があるということを、概略理解して貰う。すっきりとは理解できないだろうが、後々、理解できるようになるだろうと考える。[7]

実験教育2

実験教育2は、実験教育1のフォローアップの位置付けで、90分×2回の教育とした。両回とも、これまで教えていなかった機能を教えた後、課題プログラムの仕様を提示し、自分で考えてプログラミングして貰う。いきなりプログラミングするのではなく、まずどのように作るか設計を考えて貰うこととした。

(1) フォローアップ第1回

最初にリストとメッセージの機能について説明する。続いて、リストとメッセージの機能を使ってプログラミングする課題としてクイズプログラムを提示した。出題者のスプライトとステージのみからなる。最初にステージのプログラムで、準備としてクイズの問題と解答をリストで作成する。それが終わったら、ステージのプログラムが出題者のスプライトにメッセージを送り、問題リストを使って出題を行っていく。ユーザから解答を受けたら、解答リストと比較して正解判定をするものである。

(2) フォローアップ第2回

最初にクローンの機能について説明する。続いて、クローンとメッセージの機能を使ってプログラミングする課題としてブロック崩しを提示した。ブロック、ラケット、ボールのスプライトとステージからなる。ブロックはクローンの機能を使って8個作成する。ブロックにボールが当たったら、ブロックのプログラムはボールに「バウンド」のメッセージを送る。ラケットにボールが当たったら、ラケットのプログラムはボールに「バウンド」のメッセージを送る。ボールは「バウンド」のメッセージを受けたら跳ね返るようにする。

実験教育3

実験教育3では、対象者を変えて、小学生の子を持つ父

親、現役の小学校教員、中学生の子を持つ母親として、実験教育1と同じ内容のプログラミング教育を実施した。

今回は、90分×3回分を1日で実施した、これは、現役の小学校教員、小学生の子を持つ父親のスケジュールを考慮したためである。

3. 実験教育の結果

●実験教育1の結果

2章に示した内容のプログラミング教育を2019年7月2日、4日、9日の3日間で行った。ただし、第3回の内容であるシューティングゲームのプログラミングは、第2回の最後で第3回までの課題として提示し、第3回において課題プログラムのレビューを行なった。対象となった小学生以下の子供を持つ母親は4名(A, B, C, D)である。後日(3週間後)、4名に対してアンケートを実施した。アンケートの内容及び結果を図1に示す。

1. 教育期間(時間)はどうでしたか?				
短い	やや短い C D	丁度いい A B	やや長い	長い
2. 教育難度はどうでしたか?				
難しい	やや難しい D	丁度いい A B C	やや易しい	易しい
3. これまでにプログラミングの経験はありましたか?				
なし B	ほぼなし A C D		多少あり	あり
4. SCRATCHについて知っていましたか?				
知らなかった B	名前を聞いた程度 A C		多少知っていた	知っていた D
5. プログラミングについて理解できたと感じましたか?				
感じなかった	やや感じなかった	何ともいえない A D	まあ感じた B C	感じた
6. 復習すればプログラミングができるようになると感じましたか?				
感じなかった	やや感じなかった	何ともいえない	まあ感じた A B D	感じた C
7. プログラミングをやってみて楽しいと感じましたか?				
感じなかった	やや感じなかった	何ともいえない A	まあ感じた D	感じた B C
8. 教育に参加してみてプログラミングに対する興味・関心は以前より増えましたか?				
増えなかった	やや増えなかった	何ともいえない	まあ増えた A C D	増えた B
9. 子供に教えられそうだと感じましたか?				
感じなかった	やや感じなかった	何ともいえない A C	まあ感じた D	感じた B
10. 今後、子供と一緒にプログラミングをしてみたいと思いましたか?				
思わなかった	やや思わなかった	何ともいえない	まあ思った A C	思った B D
11. プログラミングと設計の違いを理解できましたか?				
理解できなかった	やや理解できなかった	何ともいえない	概ね理解できた C	理解できた A
12. 論理的思考について理解できましたか?				

理解できなかった	やや理解できなかった	何ともいえない	概ね理解できた A B C D	理解できた
13. 小学校におけるプログラミング教育必修化の意味が理解できましたか?				
理解できなかった	やや理解できなかった	何ともいえない A D	概ね理解できた B C	理解できた
14. 小学校におけるプログラミング教育の必修化に賛成しますか?				
賛成しない	やや賛成しない	何ともいえない A C D	まあ賛成する B	賛成する
15. これからの時代を生き抜くためにプログラミングは重要だと思いますか?				
思わない	やや思わない	何ともいえない A D	まあ思う B C	思う
16. プログラミングは小学生にとって楽しく学べる教科になると思いますか?				
思わない	やや思わない	何ともいえない A	まあ思う B C D	思う
17. 今後、プログラミングを利用したアクティビティを教科以外の習い事などに取り入れたいと思いますか?				
思わない	やや思わない	何ともいえない A C	まあ思う B D	思う
18. その他、ご意見をお聞かせ下さい。				

図1 アンケートの内容及び計画

アンケート18の自由記述の回答は、図2のようになった。

回答者A
(1) 日頃の先生の忙しさを鑑みると、現場がさらに疲弊するのではないかという気がする。講義において、当初のプログラミング教育導入の目的をお伺いして理解はできたものの、実際の学校の状況では大きなギャップがあるように感じられる。
(2) 忙しさの余り、「何のために行うのか」という本来の目的はさておかれて、「消化試合」のようになってしまったら本末転倒。教師に任せるといふより、教師をフォローする人もしくは教師に代わってプログラミングの教育に長けた(外部の?)人材が必要のように思う。(そこまで予算化できるかどうかについても疑問。)
(3) 一方で教育に限らず、子供の頃の経験は大人になるまでに何かがどう影響するのか、とても計り知れない部分が大いにあると思う。ゆえに、「何かしらした」という経験がいつの日か何かをもたらす事があるかもしれない。もちろん、プログラミング教育を受けた直後から開眼してその道に突き進む子も出るかもしれない。そういう「出会い」の機会になるような意味で期待しても良いかもしれない。
(4) また、親の立場として追加するとしたら、普段国語や算数など自宅で勉強を見る事はできるにしても、プログラミングに関して見られる親がどれだけのいるのか、もしくはプログラミングをできる環境があるのか疑問に思う点もある。他方、今でも(どこまで本当の話かはわからないもの)「一日に家で何時間もプログラミングをやっている!」と豪語する子もクラス内にいる。結果的に、クラス内にかなりの落差ができ、辛い思いをする子もいるかもしれないと思う。(というのも、先日学校の授業を見に行ったら、英語の時間中に先生が何を言っているのかもわからず立ち尽くす子が数人おり、とても辛そうにしていたのを目にしたため。そんな経験全てを排除すべきとは思わないが、得るもの・失うもの考えると、現状金銭的・人的制限が多い中で、公教育でどこまでフォローできるのかという気もする。)
回答者B

(1) 初めて目にするばかりで、とても充実した楽しい時間でした。
 (2) まったくの初心者だったので、x座標とy座標が、どっちがどちから記憶がおぼろげな状態でしたので、説明の時に横軸、縦軸という補足があると安心できます。
 (3) 1進むはドット 1 つ分だからほんの少ししか進まないというのを、後から本で読んで知ったのですが、そのような説明があればもっとわかりやすかったです。
 (4) 宿題としてやる時間が子育て中だととれませんので、自分で考えてみる回を 1 回設けていただいたうえで最終発表までとって理解が深まったように思います。
 (5) 巷のプログラミング塾でもこのような教材で教えているならば、自分で教えられるし、一緒に楽しめるなど感じました。
 (6) また今回の講師陣や、今回集ったメンバーたちから、子どもの教育について様々な意見を聞くことができとても有意義な会でした。
 (7) 次回開催（秋冬頃希望）までには、娘と楽しめるようになっていたいと思います。
 (8) 設問 11（「プログラミングと設計の違いを理解できましたか？」）でプログラミングと設計の違いって、説明ありましたっけ？記憶になく、教えていただけたらすっきりします。
 (9) スクラッチ、本を買って満足してしまい(笑)まだ先に進めてないのですが、別の本を見たら、無料でダウンロードするスクラッチもあるそうですね。そのほうがオフラインでもできるから使い勝手がよさそうですが違いなどあるのでしょうか、もしわかるようなら教えてください。

回答者 C

(1) 設問 9（「子供に教えられそうだと感じましたか？」）に関しては、教える・教えられる、という関係性が成り立たないと感じました。補い合うというか、教え合うという関係性になるのではないかと、というのは、同じものを作るのならともかく、作りたい仕様が多岐にわたるので、それぞれの分野で詳しい人が出そうなので。
 (2) そういう意味で、設問 10（「今後、子供と一緒にプログラミングをしてみたいと思いませんか？」）は、「さあ、いままらプログラミングと一緒にやろう」という感じじゃなくて、それぞれが、自分の時間に作って、一緒にシェアし合う、という感じかと。遊びもそうですが、「バラバラで一緒」というのが一番いいと思うし、これからの時代、現実的だと思います。そういう意味での「一緒にやろう」なら回答は「思った」です。
 (3) 最近では、どこにいても「プログラミング」です。この時代の流れで、避けては通れない、と感じています。そんな中で、構造をちょっと垣間見る機会を持てたこと、私にとって「プログラミング」という未知の世界を、「やればいいのか」と思える機会を持てたことに、感謝します。ただ、多大な時間がかかるので、時間を捻出することが一番の課題で、宿題がでても、それをこなす時間がとれず、自分のことながら残念です。逆に、子どもたちは、持っている多大な空き時間を、YouTube などを見る側として消費するよりは、作る側に回れるきっかけになればいいな、と思います。

回答者 D

(1) 今回 3 日にわけて scratch を指導していただき、ひとりで自由にさわっているだけでは、きっとチャレンジしようとはしなかったであろうスキルに触れることができました。3 日目の内容は高度すぎて、ひとりでできるようになるまでにはまだまだ時間を要すると思いますが、これまでより踏み込んで取り組みたいという意欲につながりました。
 (2) 3 日間の講習ととらえたときに、難度の変化が激しすぎる印象なので、1 日目の単元はもっと時間を短縮して適切なスモールステップが踏めるよう、授業設計されるとよいのではないかと思います。
 (3) 初回はすることがきっちり決まっていて、後半になると自由度が増す内容でしたが、子どもたちの中には「決められたこと」をさせられると興味を持ちにくくなる特性を持った子たちが一定数の割合で存在します。スタート時に興味を失ってしまうのはもったいないですし、そういった子どもたちにこそ届けたい授業だとも思いますので、自由度の調整に

ついても考慮いただけると嬉しく思います。
 (4) 多くの母は、子どもたちの行動をコントロールしたくなる性質を持っています。教えた通りにやらないとすぐに正解を教えてしまうことが多いので、子どもたちが試行錯誤しながら論理的に思考する経験を積めるように、そういった注意も合わせてご指導いただければ、子どもたちにとっても有益な場になると思います。

図 2 アンケート 18 の回答

●実験教育 2 の結果

2 章に示したプログラミング教育のフォローアップを、2019年10月10日、10月24日の二日間で実施した。対象者は、実験教育 1 を実施してもらった小学生を子に持つ、母親 4 人（A, B, C, D）のうちの 3 人に参加してもらった。実施後に、アンケートを実施した。アンケートの結果と内容を図 3 に示す。アンケートは、前回と同じ内容となっており、フォローアップ後の変化を見るためである。

5. プログラミングについて理解できたと感じましたか？				
感じなかつた	やや感じなかつた	何とも言えない	まあ感じました	感じました
			A B C	
6. 復習すればプログラミングができるようになると感じましたか？				
感じなかつた	やや感じなかつた	何とも言えない	まあ感じました	感じました
			A B	C
9. 子供に教えられそうだと感じましたか？				
感じなかつた	やや感じなかつた	何とも言えない	まあ感じました	感じました
	A	C	B	
10. 今後、子供と一緒にプログラミングをしてみたいと思いませんか？				
思わなかつた	やや思わなかつた	何とも言えない	まあ思った	思った
		A C		B
11. プログラミングと設計の違いを理解できましたか？				
理解できなかった	やや理解できなかった	何とも言えない	概ね理解できた	理解できた
		B C	A	
12. 論理的思考について理解できましたか？				
理解できなかった	やや理解できなかった	何とも言えない	概ね理解できた	理解できた
		C	A B	
13. 小学校におけるプログラミング教育必修化の意味が理解できましたか？				
理解できなかった	やや理解できなかった	何とも言えない	概ね理解できた	理解できた
	C	A	B	
14. 小学校におけるプログラミング教育の必修化に賛成しますか？				
賛成しない	やや賛成しない	何ともいえない	まあ賛成する	賛成する
	C	A B		
15. これからの時代を生き抜くためにプログラミングは重要だと思いますか？				
思わない	やや思わない	何ともいえない	まあ思う	思う
		A C	B	
18. その他、ご意見をお聞かせ下さい。				

図 3 アンケートの内容及び結果

アンケート 18 の自由記述の回答は、図 4 のようになった。

回答者 B
小一の娘と夏休みにチャレンジしました。小学生高学年（ローマ字入力ができるようになってから）になってからでないかとプログラミング作成を楽しむのは、厳しいと感じました。低学年のうち、ゲーム感覚でお絵かきソフトなどで遊ばせてもいいのかなと思いました。毎回勉強になります。

図 4 アンケート 18 の回答

●実験教育 3 の結果

実験教育 3 では対象者を、小学生の子を持つ父親 (E)、現役の小学校教員 (F)、中学生の子を持つ母親 (G) とした。同じ内容のプログラミング教育を 2019 年 1 月 24 日に、90 分×3 回分を 1 日で実施した。

アンケート内容を図 5 に示す。

5. プログラミングについて理解できたと感じましたか？				
感じなかつた	やや感じなかつた	何ともいえない	まあ感じた	感じた
			E F G	
6. 復習すればプログラミングができるようになると感じましたか？				
感じなかつた	やや感じなかつた	何ともいえない	まあ感じた	感じた
			E F G	
9. 子供に教えられそうだと感じましたか？				
感じなかつた	やや感じなかつた	何ともいえない	まあ感じた	感じた
			E F G	
10. 今後、子供と一緒にプログラミングをしてみたいと思いませんか？				
思わなかつた	やや思わなかつた	何ともいえない	まあ思った	思った
			F	E G
11. プログラミングと設計の違いを理解できましたか？				
理解できなかつた	やや理解できなかつた	何ともいえない	概ね理解できた	理解できた
		G	E F	
12. 論理的思考について理解できましたか？				
理解できなかつた	やや理解できなかつた	何ともいえない	概ね理解できた	理解できた
		G	E F	
13. 小学校におけるプログラミング教育必修化の意味が理解できましたか？				
理解できなかつた	やや理解できなかつた	何ともいえない	概ね理解できた	理解できた
		F		E G
14. 小学校におけるプログラミング教育の必修化に賛成しますか？				
賛成しない	やや賛成しない	何ともいえない	まあ賛成する	賛成する
		F		E G
15. これからの時代を生き抜くためにプログラミングは重要だと思いますか？				
思わない	やや思わない	何ともいえない	まあ思う	思う
			F	E G
18. その他、ご意見をお聞かせ下さい。				

図 5 アンケートの内容及び結果

アンケート 18 の自由記述の回答は、図 6 のようになった。

回答者 E (小学生の子を持つ父親)
小学校でのプログラミング教育が、必須化となる中で、私自身は学んでこなかったため、どのように教えるのか親としてどのように変わっていくのか不安でした。 今回、SCRATCHを通して、小学校での必修化の意味を教えてください、まずは、楽しく概念（論理的思考）を学ぶことが大切だと思います。子供の興味に応じて、後から追加でコードを自分で勉強すると思うので、まずは、プログラミングについて、苦手意識を持たずに子供に楽しいと思わせられればと思います。 スクラッチは、パーツの言葉だったり、組み合わせが複雑なので、難しいと感じている。またパソコン自体に触れる機会も少ないため、キーボードやマウスの操作を楽しく教えるところから入らないと継続は難しいと考えている。 学校ではビスケットを使って授業を行なっているため、まずはパソコン、タブレットの操作に慣れつつ、動かす楽しさ、便利さから教えて行きたいと考えている。 子供とプログラミングをやってみようかについては、小学校での必須化もあり、やってみようと考えている。 プログラミング教育の必須化は当然の流れだと考えている。実際に仕事でプログラマーにならなくても、ユーザー利用は行うので、概念を小学生から身に付けるのは賛成。また、プログラミングを通して、論理的思考を鍛えられることに繋がると考えている。 プログラミングを教えられる先生がいるのが懸念、子供が興味を持つ前に小学校から始めることで、苦手意識を持ってしまったり、食わず嫌いになることだけは避けたいと考えている。
回答者 F(小学校教員)
プログラミングのソフトに初めて触れました。キットをあてはめていだけなので、簡単かと思いましたが、思ったように動いてくれず四苦八苦しました。 これをうまく動かすために、論理的思考が必要なかと思いますが、その思考がどのようなものに生かせるのかを考えていきたいと思えます。 子供にとっては、試行錯誤することが、思考能力の強化に良い結果を生むと思うので、必修化はともかく、プログラミングを学ぶことは良いことだと思います。 お子さんというのが自分の子供で、かつそれが小学生だとすると、今回体験してみて、教えられそうだと感じました、自分の子供に対してであれば、途中、うまく動かせなかったとしても、あれこれ話し合いながら進めることができると思います。失敗して対策を考えることで、よりスキルを向上させられるのではないかと思います。パーツを動かしていくことで絵を動かしたり、ゲームを作ることができれば、子供はとても楽しんで取り組むだろうと思います。こういうゲーム要素の強いものは、たいい子供のほうが上達が早いので、あっという間に親を追い越していくことでしょう。 学校の児童だった場合は、講習後に自分で何度も試してみても、ほぼマスターしてからでないと教えるのは厳しいと思います。わからないことが出てきてしまうと、そこで授業が中断してしまうからです。もしかすると、自分よりも理解している子どもがいて、逆に教えてくれるような場面も出てくるかもしれません。でも、それが何度も続くと、ダメ先生として認定されそうですね。 必修化については、Scratch等の環境を使うことで論理的思考力が身につくのかどうか、私にはまだよくわかりません。でも、論理立ててわかりやすく進めないと目的を叶えることができないことを知るの、子供にとって益であると思います。論理的思考力については、よく言われるように、料理のレシピを書くなど、説明文を書くことも身につけられるだろうと思うのですが、子供の多くはPCを使うことが好きだと思われるので、プログラミングの授業には喜んで取り組むでしょう。必修化については、必修としないと

プログラミングの授業をしない学校も出てきてしまうと思うので、子供の教育の機会均等からみて仕方ないと思います。この必修化のおかげかどうか、先日も子供一人に1台のPCが配布されると決まったようです。ただ、英語や道德の教科化や授業時間増など、現場の先生の負担は増すばかりと思われるので、そのあたりの対策もしっかりとってから進めてもらいたいと思います。

回答者G (中学生の子を持つ母親)

今や、中学生の時から、コンピューターやオフィスソフトを習い、マスターしている生徒が多い中、デジタルデバイド格差が広がらないようにするために、小学生からコンピューターに親しんでプログラミングを学ぶと言うのに賛成です。

タイなどアジア諸国では、小学生がエクセルを学校で習い、使いこなしている時代なので、日本も世界基準にあわせるべきだと思います。

また、日本の子供は、幼い頃よりディスカッションやディベート力を磨く授業が少ないと感じています。一つのプロジェクトを動かすために必要な条件を考えていながら、目標に向かって進めていく過程は、幼い頃から身につける学びは、大変有効であると思いました。

図6 アンケート18の回答

4. 考察

実験教育の結果を分析していく。アンケート1~17は、アンケート3~4を除き選択肢が5個ある。アンケート5~17においては、前半2個の選択肢はDISAGREE (NEGATIVE)、後半2個の選択肢はAGREE (POSITIVE)と捉える。真ん中の選択肢「何ともいえない」は、受講者の講師に対する気遣いを考慮してDISAGREE (NEGATIVE)と捉える。

アンケート1~4の結果を見ると、対象者はプログラミングの経験はほぼ無く、教育の期間はやや短く、難度はやや難しいと回答している。期間がやや短く感じたことに関しては、回答者Bのアンケート18の記述(4)、回答者Cのアンケート18の記述(3)に見られるように、第2回の終了時に、第3回までの課題を提示したが、その課題に取り組む時間の捻出が難しかったことに起因している。課題の提示及び課題への取り組みは、講義時間の中で実施するのが適切だった。また、どのような点をやや難しいと感じたのかということ、回答者Bのアンケート18の記述(2)(3)に見られるように、座標やスプライトの歩数に関する部分の説明が無かったことが一因となっている。一方、4名の中で唯一、SCRATCHを知っていた回答者Dのアンケート18の記述(1)(2)(3)を見ると、SCRATCHを知っている受講者にとっては、第1回の内容は優しすぎるのが分かる。

アンケート5~8の結果を見ると、プログラミングについて理解できたと回答したのは受講者の50%だったが、全ての受講者が復習をすればプログラミングができるようになると回答している。また、受講者の75%がプログラミングを楽しいと回答し、全ての受講者がプログラミングに対する興味・関心が増したと回答している。これらの回答を見ると、もう少し講義を追加し、プログラミング経験を増

やせば、プログラミングができるようになるという感触が確かなものになると感じる。また、実際に受講者からは追加の講義の要望が上がっていた。

また、実験教育2でフォローアップ後のアンケート結果からも、プログラミング授業導入への関心については、理解が深まったと同時に、関心も高まっていることがわかる。アンケート5での、復習すればプログラミングができるようになると思うかの問いに関しては、全員が「まあ感じた」「感じた」と回答しており、やはり回を重ね、体験することで、理解が深まっていることが分かる。現役小学校教員、小学生の子を持つ父親、中学生の子を持つ母親による実験教育3でも、ほぼ同様の結果となっている。

これらの結果により、プログラミングの理解、復習すればできる様になるということはさらに強まった。保護者向けのこの教育方法は適切であると言える。

アンケート9~10の結果を見ると、子供にプログラミングを教えられそうだと回答した受講者は50%だったが、全ての受講者が子供と一緒にプログラミングをしたいと回答している。どのように子供と一緒にプログラミングをしたのかということ、回答者Cのアンケート18の記述(1)(2)に見られるように、一緒に取り組むというより、それぞれの取り組みを教え合うような形だという意見があった。そのような取り組みの中から、論理的思考の意味が伝わる時期や機会が見つかるのではないかとと思われる。

アンケート11~12の結果を見ると、プログラミングと設計の違いを理解できたと回答した受講者は50%だが、全員が論理的思考について概ね理解できたと回答している。プログラミングと設計の違いに関しては、回答者Bのアンケート18の記述(8)にあるように、説明された印象がないという受講者も見られたので確認したところ、「そういえばそう聞いた」という返事だった。プログラミングと設計の違いについては、もう少し丁寧な説明が必要だったと思われる。

アンケート13~14の結果を見ると、概ね50%の受講者がプログラミング教育必修化の意味を理解したと回答しているが、プログラミング教育の必修化に賛成すると回答した受講者は25%だった。プログラミング教育必修化に賛成しない理由は、回答者Aのアンケート18の記述(1)(2)に見られるように、小学校の教員が多忙であるという現実を考慮してのことであると思われる。

また、実験教育3において、現役小学校教員から以下のような意見が得られた。「プログラミング的発想は、子ども達には得意な子が多いことが考えられ、時には大人を凌ぐ知識や発想力がある場合も想定される。ほぼマスターしてからでないで教えるのは厳しい。わからないことが出てしまうと授業が中断するから。また、逆にこどもに教えられる場面が何回かあると、ダメ先生とレッテルを押されてしまう」

アンケート15～17の結果を見ると、これからの時代を生き抜くためにプログラミングは重要だと回答した受講者は概ね50%だが、75%の受講者が、プログラミング教育が楽しく学べる教科になるだろうと回答している。プログラミングを教科以外の習い事に取り入れたいと回答している受講者は50%だった。回答者Aは、アンケート18の記述(3)(4)で、プログラミングに触れる機会に期待するものの、算数や国語といった教科と違い、家庭でプログラミングを見られる保護者がどの程度いるのかということに危惧している。更に記述(2)には、プログラミングの教育に長けた外部の人材の必要性について述べられている。受講者には説明しなかったが、プログラミングの教育に長けた外部の人材に関しては、[1]にも記述されている。また、回答者Dのアンケート18の記述(3)(4)には、子供及び母親の特性に関しても記載されている。それは、決められたことをさせられると興味を失ってしまう特性の子供が一定の割合にいるということである。また、子供の行動をコントロールしようとし、結果が出ないとすぐに解答を教えるという特性の保護者が多くの割合にいるということである。これは、小学校の教員のように教育理論や教育手法を学んでいない一般の人には、このような特性の人が一定の割合にいるだろう。あくまでも小学校の教員の補完的な役割であることを忘れてはならないと考える。

5. おわりに

小学校でプログラミングに触れ、段階的に論理的思考を育てていくことを考えた時、その中心的な役割を担うのは当然ながら小学校の教員であるが、それだけでは不十分である。我々は小学生以下の子供を持つ保護者等にプログラミング教育を施し、保護者等が補完的に子供にプログラミングを教えるという方向を考えている。子供が小学校教育を受ける長いスパンの中で、論理的思考の意味が子供に伝わる時期や機会を意識して貰い、適切なタイミングでアドバイスしていくようにして貰う。この考えの下で、小学生以下の子供を持つ母親4名及び小学生の子を持つ父親、現役の小学校教員、中学生の子を持つ母親に対して実験教育を行なった。今回の実験教育において、我々の考える方向が適切であることが確認できた。

ただし、今回の実験教育の対象者である7名は、いずれもプログラミングの講義への参加を希望してきた前向きな受講者ばかりである。しかし、全ての保護者等がそのように前向きである訳ではない。あるいは、前向きであっても時間的に困難な場合もある。それを考えると、保護者等だけで我々が考えている方向を実現するのは難しいだろうと推測される。そのため、多忙な小学校の教員が行うプログラミング教育を補完するには、保護者等だけでなく、保護者等を中心とした地域コミュニティが小学校のプログラミン

グ教育を支援し、子供が小学校教育を受ける長いスパンの中で、論理的思考の意味が子供に伝わる時期や機会を意識し、適切なタイミングで子供にアドバイスしていくようにするのが適切ではないかと考える。

参考文献

- [1] “小学校プログラミング教育の趣旨と計画的な準備の必要性について(1)”. 文部科学省初等中等教育局情報教育・外国語教育課情報教育振興室.
http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2019/05/21/1417047_001.pdf.
- [2] “小学校プログラミング教育の趣旨と計画的な準備の必要性について(2)”. 文部科学省初等中等教育局情報教育・外国語教育課情報教育振興室.
http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2019/05/21/1417047_002.pdf.
- [3] John Maloney, Leo Burd, Yasmin Kafai, Natalie Rusk, Brian Silverman, Mitchel Resnick. Scratch: A Sneak Preview. Second International Conference on Creating, Connecting and Collaborating through Computing, Kyoto, Japan, 2004, p. 104-109.
- [4] “プログラミング人材育成の在り方に関する調査研究報告書”. 平成27年6月. 総務省.
http://www.soumu.go.jp/main_content/000361430.pdf.
- [5] “お母さんのためのプログラミング講座(テキスト1)”.
https://hirayama968.web.fc2.com/programmingformom/programmingformom_text1.pdf.
- [6] “お母さんのためのプログラミング講座(テキスト2)”.
https://hirayama968.web.fc2.com/programmingformom/programmingformom_text2.pdf.
- [7] “お母さんのためのプログラミング講座(テキスト3)”.
https://hirayama968.web.fc2.com/programmingformom/programmingformom_text3.pdf.
- [8] 崔仁珠, 上倉隼, 佐藤連, 高塚穂佳, 保土田雪成, 松延奈美, 森田昌樹, 杉本紀子, 中村文宣, 吉田典弘. 小学校プログラミング教育の学習指導案の作成と教育的効果の評価- Programming Of the Kids By the Kids For the Kids.-. 情報処理学会研究報告. 2019, Vol. 2019-CE-148, No. 10, p. 1-17.
- [9] 倉橋農, 越智徹, 尾崎拓郎, 島袋舞子. 子ども向け授業にリンクした保護者・教師向けプログラミングコースの検討. 情報処理学会研究報告. 2019, Vol. 2019-CE-150, No. 9, p. 1-5.
- [10] 内田早紀子, 松村敦. 日常の活動を題材とした小学生向けプログラミング的思考育成ツールの開発. 情報処理学会研究報告. 2018, Vol. 2018-CE-147, No. 9, p. 1-5.
- [11] 吉田研一, 伊藤寿晃, 山脇智志, 大森康正. 小学生を対象としたプログラミング教育指導者育成方法とその評価. 情報処理学会研究報告. 2017, Vol. 2017-CE-138, No. 10, p. 1-4.
- [12] 尾花拓海, 鈴木龍成, 吉村明人, 白田莉菜, 半澤魁士, 佐久間拓也, 川合康央, 池辺正典. 小・中学生を対象にしたプログラミング学習を支援するアプリケーション開発プロジェクト. 情報教育シンポジウム. 2017, p. 202-205.
- [13] 阿部和広. 子どもの創造的活動とICT活用. 情報処理. 2015, Vol. 56, No. 4, p. 350-354.
- [14] 松浦雅子, 平山秀昭. 小学校においてプログラミング教育が実施される状況を鑑みた母親向けプログラミング教育. 情報処理学会研究報告. 2019, Vol. 2019-CE-151, No. 8, p. 1-9.