

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

松浦雅子^{†1} 平山秀昭^{†2}

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母親にプログラミング教育を施し、母親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。その際、母親に文部科学省が言う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

Programming Lecture for Mothers in the Situation where Programming Education in Elementary Schools is Performed

MASAKO MATSUURA^{†1} HIDEAKI HIRAYAMA^{†2}

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]。

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んで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。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

^{†1} 目白大学
Mejiro University
^{†2} 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

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母親にプログラミング教育を施し、母親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。その際、母親に文部科学省が言う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. 母親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。

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等の普通のプログラムならどのようになるのかを示す。普通のプログラムでは、「{」と「}」のバランスが取れている必要があるのか、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母親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等の普通のプログラムならどの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と改造してみる。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、先に普通のプログラムを見てから、それがSCRATCH だとどのようなプログラムになるのかを見ていく。Java 等の普通のプログラムでの” Hello World”, 変数宣言, 代入文, 演算といった操作が, SCRATCH だとそうなるのかを見る。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から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人はいないだろうが, それで構わない。自分で考え, 自分で調べ, プログラミングするやり方を教える。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, 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]

3. 実験教育の結果

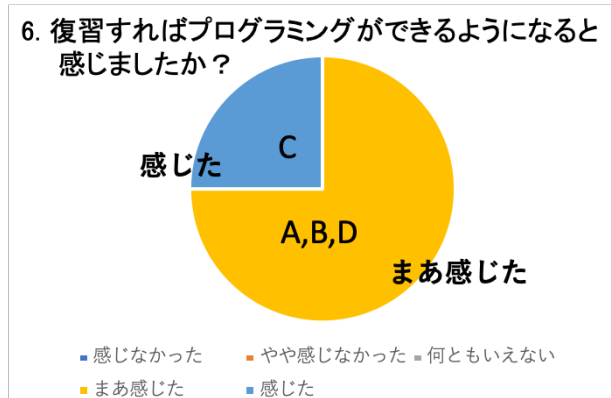
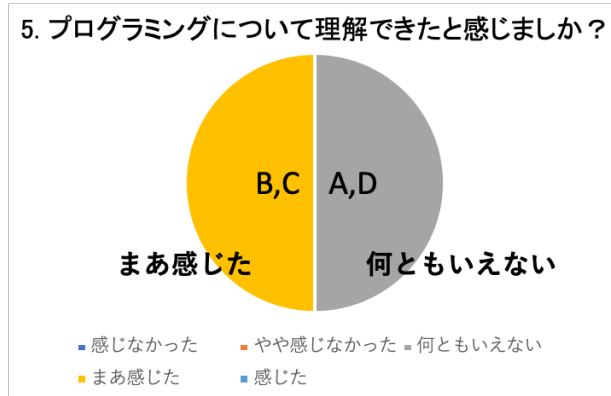
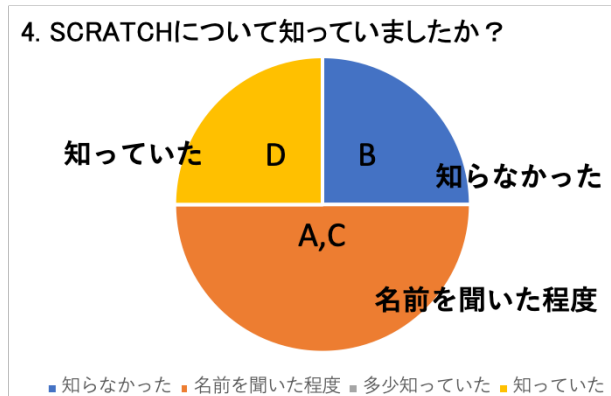
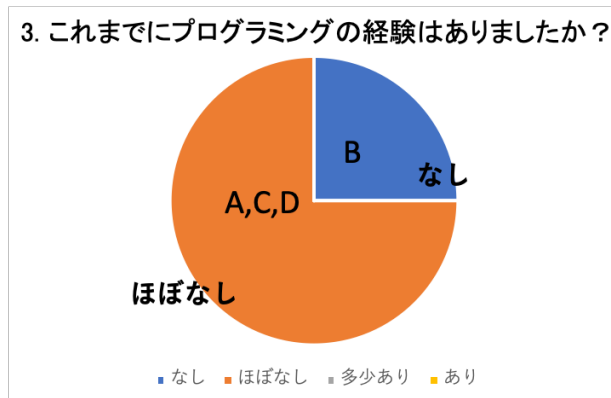
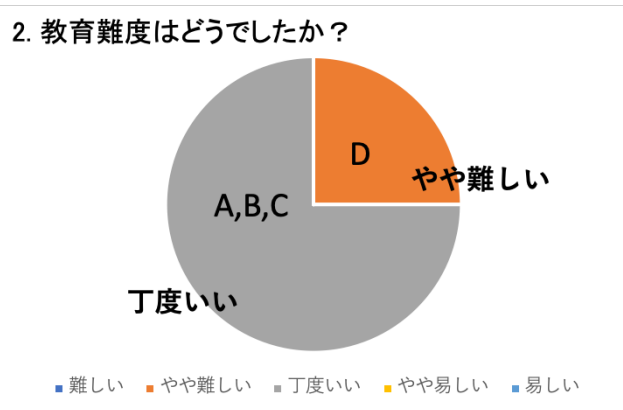
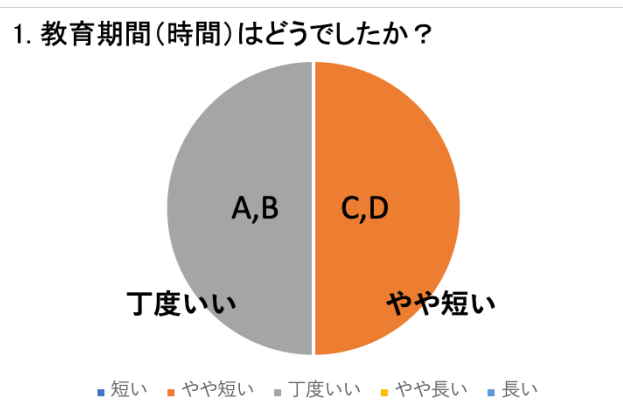
2章に示した内容のプログラミング教育を2019年7月2日, 4日, 9日の3日間で実施した。ただし, 第3回の内容であるシューティングゲームのプログラミングは, 第2回の最後で第3回までの課題として提示し, 第3回において課題プログラムのレビューを行なった。対象となった小学生以下の子供を持つ母親は4名 (A, B, C, D) である。後日 (3週間後), 4名に対してアンケートを実施した。アンケートの内容を図1に示す。

1. 教育期間 (時間) はどうでしたか?				
短い	やや短い	丁度いい	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	丁度いい	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	ほぼなし		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	名前を聞いた程度		多少知っていた	知っていた
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	まあ感じました	感じました
6. 復習すればプログラミングができるようになって感じましたか?				
感じなかった	やや感じなかった	何ともいえない	まあ感じました	感じました
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	まあ感じました	感じました
8. 教育に参加してみてプログラミングに対する興味・関心は以前より増しましたか?				
増さなかった	やや増さなかった	何ともいえない	まあ増した	増した
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	まあ感じました	感じました
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	思った
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	理解できた
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	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	概ね理解できた	理解できた

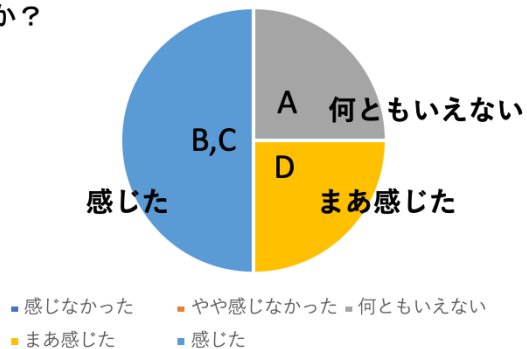
かった	きなかった	ない	きた	
14. 小学校におけるプログラミング教育の必修化に賛成しますか？				
賛成しない	やや賛成し ない	何ともいえ ない	まあ賛成す る	賛成する
15. これからの時代を生き抜くためにプログラミングは重要だと思いますか？				
思わない	やや思わな い	何ともいえ ない	まあ思う	思う
16. プログラミングは小学生にとって楽しく学べる教科になると思いますか？				
思わない	やや思わな い	何ともいえ ない	まあ思う	思う
17. 今後、プログラミングを利用したアクティビティを教科以外の習い事などに取り入れたいと思いますか？				
思わない	やや思わな い	何ともいえ ない	まあ思う	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

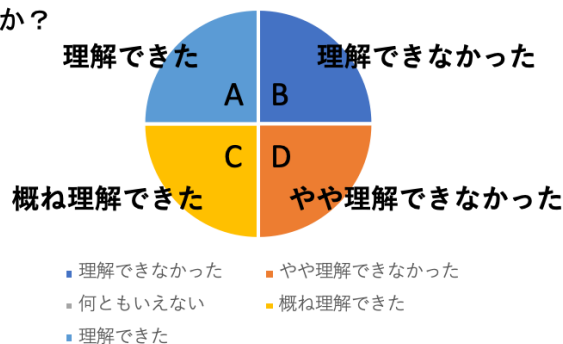
アンケート1から17の回答は、図2のようになった。



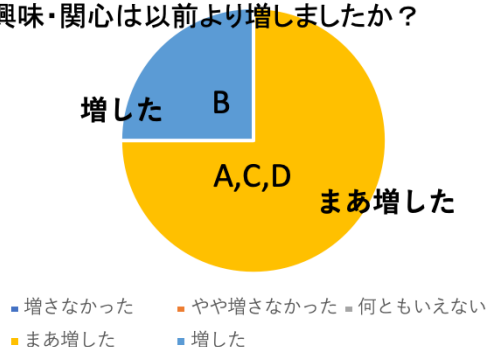
7. プログラミングをやってみて楽しいと感じましたか？



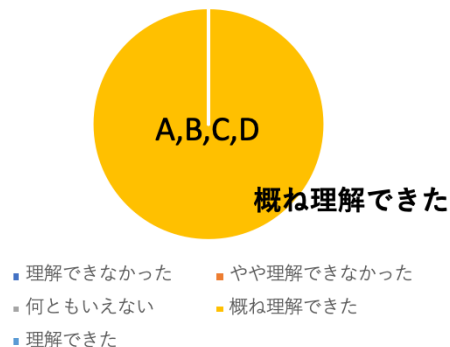
11. プログラミングと設計の違いを理解できましたか？



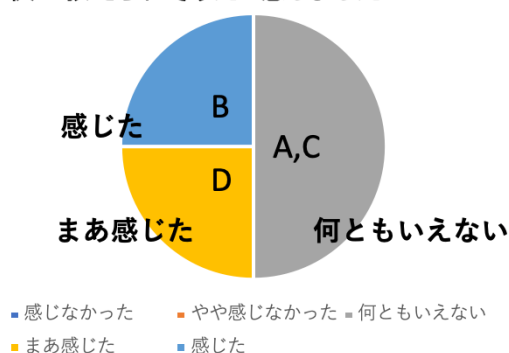
8. 教育に参加してみてプログラミングに対する興味・関心は以前より増しましたか？



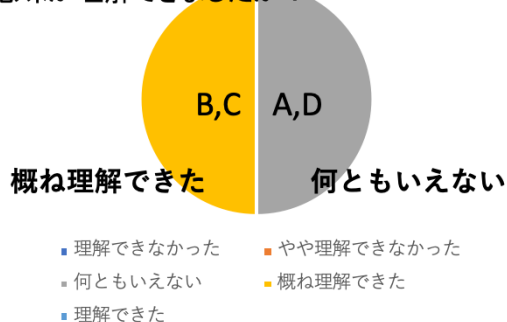
12. 論理的思考について理解できましたか？



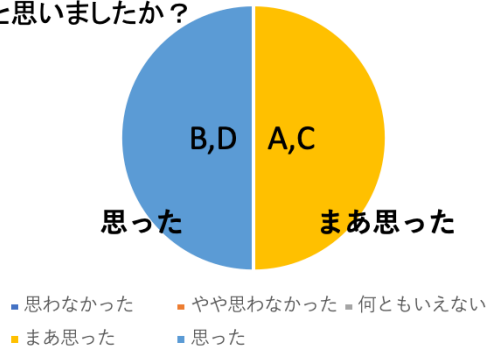
9. 子供に教えられそうだと感じましたか？



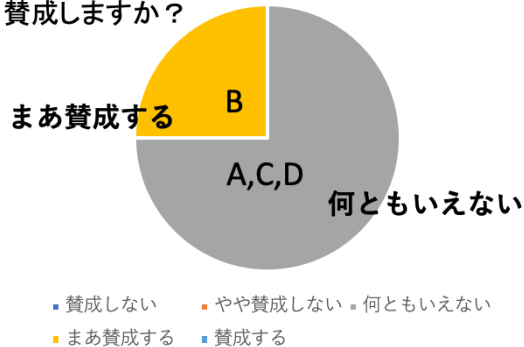
13. 小学校におけるプログラミング教育必修化の意味が理解できましたか？



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？



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？



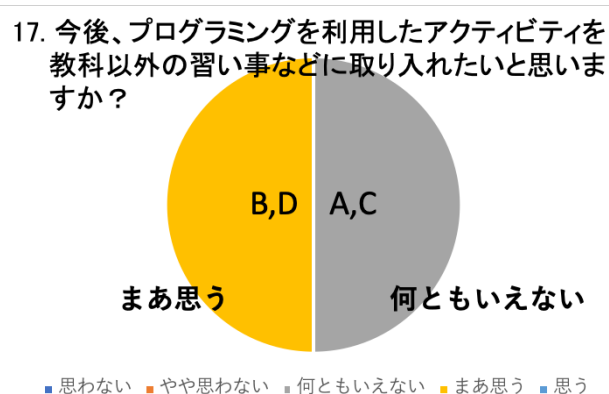
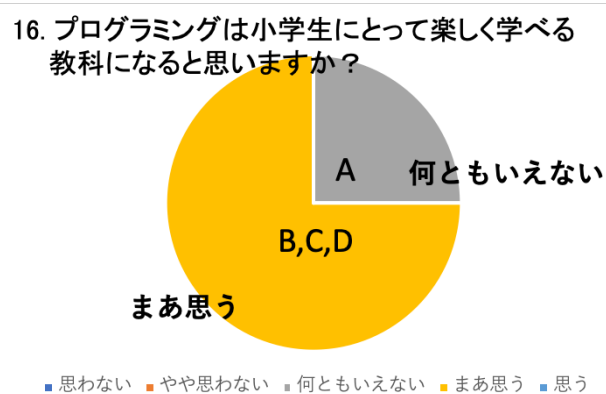
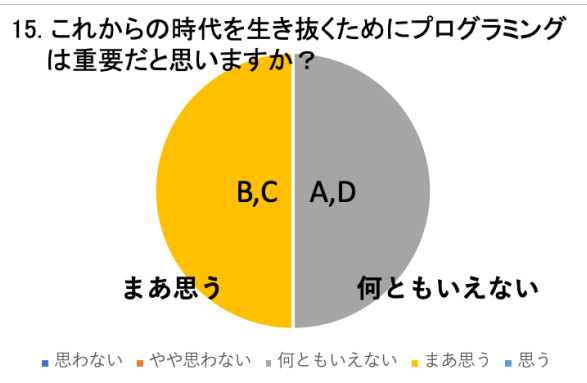


図2 アンケート1-17の回答

アンケート18の自由記述の回答は、図3のようになった。

<p>回答者 A</p> <p>(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。</p> <p>(2) 忙しきの余り、「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。教師に任せるというより、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?)人材が必要なように思う。(そこまで予算化できるかどうかについても疑問。)</p>
--

(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がどう影響するのか、とても計り知れない部分が多いかと思っている。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際、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・失うものを考えると、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) 初めて目にする事ばかりで、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、どっちがどっちかす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で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回発表だともっと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、一緒に楽しめるなと感じました。
- (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?記憶がなく、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ようです。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。

<p>回答者 C</p> <p>(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。</p> <p>(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したか?」) は、「さあ、いまか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。</p> <p>(3) 最近は、どこにいても「プログラミング」です。この時代の流れで、避けては通れないな、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たせたこと、私にとって「プログラミング」という未知の世界を、「やればできるかも」と思える機会を持たせたことに、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。</p>
<p>回答者 D</p> <p>(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。</p> <p>(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。</p> <p>(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もったいないですし、そういった子どもたちにも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。</p> <p>(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ると、子どもたちにとっても有益な場になると思います。</p>

図 3 アンケート 18 の回答

4. 考察

実験教育のアンケート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いのかということ、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。

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れる父兄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てしまう特性の母親が多く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男女共に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. 関連研究

参考文献[8]は、高度情報化が進む現代において「プログラミング教育」は、現代社会の基盤であり、子供がテクノロジーに対する理解を深め、それを受け入れることに繋がると述べている。プログラミングがどのような思考力を育むことができるのかという「プログラミングがもたらす教育的効果」を明らかにすることを目的に、独自に作成した学習指導案をもとに、小学 2 年生に向けて授業実践し、授業前後における児童の論理的思考力・読解力・問題解決力の変化を検証している。3 つのスキルにおける授業前後での大きな向上は見られなかったが、今後の問題作成を調整していく必要が伺えるとともに、批判的思考力等、他のス

キルにおいても向上の兆しがみえていと述べている。

参考文献[9]では、小学生向けにプログラミング的思考を促す目的で、対話的/スクリプト的実行、順次・分岐・反復を盛り込んだアンブレグド授業を実践している。ハンバーガー・ロボのアイデアの子供向けコース、それとリンクする大人向けコースを提案している。ハンバーガー・ロボへの指示と実行は、単純化されたプログラム・コードとその実行を可視化したモデルと考えることができるとしている。これにより、厳密で曖昧性のない指示、対話モード/スクリプトによる実行、順次・反復・分岐、ブロックといったプログラミングの重要な概念が理解可能となっている。教師・保護者にとっては、子供が実際に利用する教材とリンクして学習することで、教育に際して助けとなることが期待される。大人向けコースは、保護者・教師の理解を深めることを目指し、同時に教師が型や変数への代入、言語の仕様といった、より高度なテーマを学ぶことを射程に入れている。

参考文献[10]では、暗記型のプログラミング教育では、小学生がプログラミング的思考を習得することは困難とされているため、小学生にとって身近な日常の活動を題材としたプログラミング的思考育成ツールを開発している。身近な題材をアプリケーションとすることで、より効果のある学習支援ができるのではないかと考え、「必要な手順があることに気付く」に着目し、「順次処理」、「繰り返し処理」、「条件分岐処理」を実装するツールの開発を行っている。小学校 2 校で利用実験と評価を行ったところ、プログラミング的思考の向上の効果は異なったが、参加者の約 8 割の子供から楽しく学習でき、ツールは使いやすかったと回答があり、開発したツールは小学生が利用するのに適していると述べている。しかし約 3 割が難しいと感じており、子供の理解に合わせた動きや事象と問題のレベルの設定については、再度検討する余地があるとも述べている。

総務省では、2016 年より地域の人材を活用する「若年層に対するプログラミング教育の普及推進」を実施しているが、参考文献[11]では、そのため近隣地域の学生を活用したプログラミング指導者（メンター）の育成と、大阪「ものづくり」DNA を継承する人材育成のきっかけとするための効果的・効率的な小学生向けロボットプログラミング教育を実施し、「地域完結型プログラミング教育モデル」の実証を進めている。小学校 5 年生の 2 クラス 60 名を対象にプログラミング講座を行い、指導者育成の状況および評価を報告している。Ozobot Bit は、準備と組み立てが不要である点、ブラウザ上でプログラミングができる点などが、公教育での実習授業に非常に適しているロボットであると述べられている。また、子供の発達段階に適したよりよいプログラミング教材を開発する際には、知識獲得から実践的な体験を通して改善をくり返すアクティブラーニング的な方法を確立する必要があるとも述べている。

参考文献[12]では、現在において情報系人材が不足していることによってプログラミング教育が促進されている背景から、小・中学生を対象としたプログラミング学習支援のアプリケーション開発を行っている。プログラミング的思考を養成可能なアプリケーションを開発することで、一連の活動を実現するために必要な組み合わせについて、論理的に考える力を養成しようとしている。プログラミング初学者が暗記型学習から思考型に移行する際に課題となるモチベーションの維持のために、ゲーミフィケーションやビジュアルプログラミングの概念を取り入れている。思考型学習への移行を補助することは重要であり、この移行に必要となる能力を支援している。

Scratch は、子供の遊びを観察することから得られた「想像、作成、遊び、共有、振り返り、想像」という螺旋構造を支援するために作られている。参考文献[13]では、デジタルテクノロジーの書き手を育て、豊かな言語能力の育成を目指すためには、パソコンを特別視せず、子供が誰から指図されることもなく、自然に使うような環境や雰囲気を作成する必要があると述べている。これにより、新しいコンテンツを自分たち自身で作出し、機能やテクニックを使ってプログラムする児童も出てきているが、IT人材育成の立場に立つだけでなく、子供の創造性を念頭においた Papert の構築主義や Resnick の言う「表現の新しい形式や学びの新しい文脈」としてのプログラミングも加えるべきだと述べている。

6. 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母親にプログラミング教育を施し、母親が補完的に自分の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認識できた。

ただし、今回の実験教育の対象者である4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、全ての母親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、母親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、母親だけでなく、母親を中心とした

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.