

# プログラミングが苦手な児童に着目した「プログラミング的思考」の育成

— クラウドを活用した協働的な学びを通して —

和泉 貴裕 (千葉市立金沢小学校)

[研修先 千葉大学教育学部]

## 《研究の概要》

本研究は、クラウドを活用した協働的な学びによって、プログラミングが苦手な児童であってもプログラミング的思考を高められることを明らかにするものである。千葉市が整備した1人1台端末「Chromebook」を使用し、クラウド上で「児童の相互アドバイス」と「情報ライブラリの活用」を取り入れた授業を行った。児童の相互アドバイスでは、デジタルホワイトボードで交流したり他者の学習履歴へコメントを付けたりし、情報ライブラリの活用では、個々の作成したプログラムを共有フォルダに保存し、そのプログラムを引用し発展させることを推奨した。その結果、プログラミングが苦手な児童であっても、プログラミング的思考の要素である「論理的推論」「抽象化」「分析」の高まりが見られた。

## 1 問題の所在

令和2年度から全面実施された小学校学習指導要領(平成29年告示)でプログラミング教育が必修となり、1年が経った。文部科学省は、「小学校プログラミング教育の手引き」や「未来の学びプログラミング教育推進月間」の中で多くの事例を挙げ、プログラミング的思考を育む授業を推進している。千葉市は「令和3年度千葉市学校教育の課題」の中で、「わかる授業」の実現へ向けて、「社会の変化に対応する情報教育の推進とプログラミング的思考の育成」を挙げている。

そのような中、令和2年度に在籍校の6学年の総合的な学習の時間で、自動運転車をつくる学習を通してプログラミング教育を行った。授業では、協働的な学びとしてペアプログラミングを取り入れ、ペアのみでの解決が難しい場合には、コンピューターームのパソコンを利用して情報を共有できるようにした。しかしながら、情報を見られるのがコンピューターームという限られた場所であり、時間も制限されているということから、その情報がうまく共有できたとはいえなかった。また、課題をなかなか解決できずに諦めてしまう児童がおり、ペアの相手に頼りっぱなしになってしまうということが起こってしまった。このような状況では、プログラミングが苦手な児童がプログラミング的思考を高めるのは難しいであろうと考えられる。

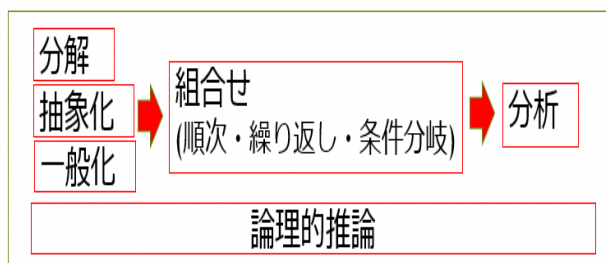
そこで、プログラミングが苦手な児童のプログラミ

ング的思考を高めるため、どのような先行研究があるのかを調べたところ、プログラミングが苦手な児童と得意な児童とを分けてプログラミング的思考の高まりを論じている研究は、見付けることができなかった。そこで、本研究ではプログラミングが苦手な児童に着目して調査を行うこととした。

では、どのようなプログラミング教育をすれば苦手な児童のプログラミング的思考が高まるのだろうか。小学校学習指導要領解説(総合的な学習の時間編)では、「共に学ぶことが個人の学習の質を高め、同時に集団の学習の質も高めていく」という記述があり、協働的に学ぶ意義が書かれている。また、「プログラミング教育の手引き(第3版)」の中では、「他者と協働しながら粘り強くやり抜く態度の育成」が重要と述べている。そこで、1人1台端末で各自がプログラミングをし、自分自身で考える時間を十分に確保した上で、クラウドツールを利用してアドバイスし合う利点を活用した協働的な学びを行えばよいのではないかと考えた。クラウドツールを活用する利点は、他者の悩みや解決方法を容易に参照できることであると考えられる。困ったことを自分から発信しづらい児童であっても、アドバイスの書き込みを見ることで、課題を解決することができるようになると思える。また、クラウドに保存されている他者のプログラムを参照することで、様々な視点での考え方ができるようになると思われる。

ここで、プログラミング的思考について詳しく述べる。プログラミング教育については、小学校学習指導要領で、「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」を行うとしている。また、「プログラミング的思考」は、小学校学習指導要領解説(総則編)で「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」と示されている。しかしながら、藤川(2020)が述べているように、プログラミング教育における「論理的思考力」や「プログラミング的思考」の規定には曖昧さがある。

そこで本研究では、藤川の提言である『論理的思考力』を『論理的推論を行う力』と読み替えるという点を踏まえつつ、ベネッセコーポレーション(2018)が提唱している「主に小学校段階におけるプログラミング的思考」を基に、プログラミング的思考を大きく六つの要素に整理する(〔図1〕)。



〔図1〕プログラミング的思考の六要素

なお、各要素の定義として、「論理的推論」は「一定の事実や前提をもとに緻密に結論を導くこと」、「分解」は「大きな動きを解決可能な小さな動きに分けること」、「抽象化」は「目的に応じて適切な側面・性質だけを取り出し、他の部分を捨てること」、「一般化」は「ものごとの類似性や関係性を見いだすこと」、「組合せ」は「目的に合わせて試行錯誤しながら、明確でよりよい手順を創造すること」、「分析」は「目的に対して、必要十分な評価の観点を考え、実行したことが、意図した活動に近付いているかどうか評価するこ

と」とする。なお、「組合せ」については、下位項目として「順次(順々に処理を実行していく)」と「繰り返し(ある処理を繰り返し実行していく)」「条件分岐(ある条件によって実行する内容を変える)」を設ける。このように整理することにより、児童の変容を容易に見取れるようになるかと考える。プログラミング的思考の見取りには、意識調査として後藤(2021)の「高学年自己評価アンケート」内から「思考力・判断力・表現力」を抜粋したもの、「順次」と「繰り返し」を見取るパフォーマンステストとして、オンライン上のプログラミング体験ゲーム「アルゴロジック1 (<https://algo.jeita.or.jp/prm/1/index.html>)」を扱う。そして、「条件分岐」を見取るものとして、田嶋(2020)の研究を参考にした分岐の文章題を使用する。

## 2 研究の目的と方法

### (1) 研究の目的

クラウドを活用した協働的な学びにより、プログラミングが苦手な児童であってもプログラミング的思考を高められることを明らかにする。

### (2) 研究の方法

#### ①クラウドを活用した協働的な学びの検討

1人1台端末を先行導入している学校の事例を調べ、プログラミング教育の中でどのような協働的な学びができるのかを検討する。

#### ②プログラミング教育の実践

①をもとに、所属校の6学年で総合的な学習の時間を主としたプログラミング教育の単元計画を立て、授業実践を行う。

#### ③プログラミング的思考の分析

実践授業の前後にプログラミング的思考に関する調査を行い、事前と事後の結果を比較する。分析の際には、プログラミングに関する意識と技能の調査による群を作成し、群別に $t$ 検定(平均に有意差があるかどうかを見取る検定)を行う。さらに、苦手群に着目し、協働的な学びの手立ての有用性を確かめる。また、プログラミングが苦手な抽出児の変容を見ていくことで、質的な調査も行う。

### 3 研究の内容

#### (1) クラウドを活用した協働的な学びの検討

##### ① 児童相互のアドバイス

###### ア デジタルホワイトボードでのアドバイス

プログラミングの途中で、一人では解決しづらいことがあった際に、他者に相談できるように、「情報掲示板」としてデジタルホワイトボード(以下「Jamboard」と記す)の付箋機能を使ってコメントを書き込んでいくこととした。1人1台端末上で使えるJamboardは、時間や場所の制約を受けない。プログラミングの途中であっても他者からのアドバイスを受けられることにより、試行錯誤を重ねることができると考えた。

###### イ 学習履歴でのアドバイス

毎時間作成する学習履歴を、一人一人が自分の学習状況を振り返るものとするだけでなく、児童同士でその学習履歴を見合ってアドバイスするのに活用することとした。また、教師が個々の困っていることを把握できるという利点を活用していくようにした。

まずは1人1台端末内の学習支援ツール「SKYMENU」内にある「発表ノート」に、プログラミングの個々の学習履歴を残す。学習履歴には、プログラムの画像と実行結果の動画を残し、自分が現在困っていることや次回への課題などを書き込んでおく。そして、グループワーク機能を使い、児童同士がアドバイスし合う。授業後には、児童の質問事項に対して教師が適切なヒントを与える。

学習履歴にアドバイスがあることにより、次時に見直しをもってプログラミングに取り組めると同時に、自分では気付かなかった視点による「論理的に推論する力」も高まるだろうと考えた。

##### ② 情報ライブラリの活用

各自が作成したプログラムを共有ドライブ上に集めた「情報ライブラリ」を作成し、他者のプログラムを引用し発展させていくことを推奨していった。

情報ライブラリを扱った実践は、桂元(2018)が協働的思考への転換を促すために行っている。しかし本研究は、プログラミングが苦手な児童でもプログラミング的思考を高めることを主としているため、全員が同

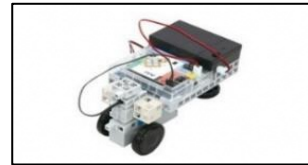
じゴールを目指す共通課題の段階では用いない。なぜならば、正しいプログラムをそのまま引用して終わりにしてしまう可能性があるからである。

それぞれが課題を設定する個別課題の段階で活用することにより、他者のプログラムを参照しつつも、自分なりの考えを加える必要が出てくる。そのことにより、プログラミングが苦手な児童であっても思考する機会が多くなるであろうと考えた。

#### (2) プログラミング教育の実践

##### ① 単元について

所属校の令和2年度の6学年総合的な学習の時間「Society5.0の社会に向けて」の指導計画を基に、ロボットカーを扱う小単元を作成した([資料1]、[表1])。

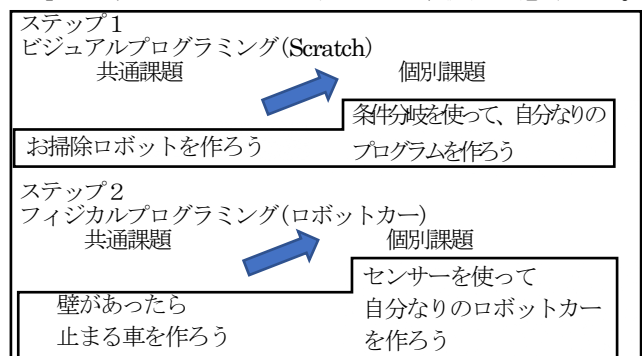


【資料1】ロボットカー(株式会社アーテック)

【表1】単元計画

時数	学習内容
1	○単元の見直しをもつ。
2~4	○Scratch(コンピュータで行うビジュアルプログラミングソフト)でお掃除ロボットを作る。(ビジュアルプログラミングによる共通課題)
5~7	○Scratchで条件分岐を使って自分なりのプログラムを作る。(ビジュアルプログラミングによる個別課題)
8、9	○節電のことを考え、暗くなったら光るライトを作る。
10、11	○ロボットカーを組み立てる。 ○ロボットカーが直進するようにプログラムをする。 ○センサーの役割を知る。
12~17	○壁があつたら止まる衝突回避車を作る。(フィジカルプログラミングによる共通課題)
18~23	○センサーを使った自分なりのロボットカーを作る。(フィジカルプログラミングによる個別課題)

なお、単元計画を作成する際には、中村(2018)が「プログラミング教育全体としての学習の系統性を大きく捉える必要がある。」と述べていることを参考に、[図2]のようなスモールステップでの系統性を意識した。



【図2】プログラミングの系統性

## ②授業の実際

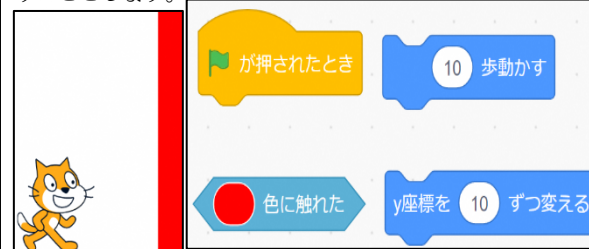
### ア 対象児童のグループ分け

本研究における「プログラミングが苦手」とは、「プログラミングに苦手意識をもっていること及びプログラミングの技能が身に付いていないこと」と定義する。そこで、プログラミングが苦手かどうかを意識面と技能面で見取り、それぞれを考慮したグループに分けた。そして、主に意識面と技能面の両方において苦手のグループに着目しながら研究を進めた。

意識面では、「プログラミングが得意か苦手か」という質問を4件法(4:得意、3:少し得意、2:少し苦手、1:苦手)を用いて行い、2と1を選択した児童を「プログラミングに苦手意識をもっている児童」とした。

技能面では、[資料2]のような課題を出し、制限時間の5分以内にプログラミングができるかどうかを確かめた。

**プログラミング課題**  
旗を押したら、ねこが赤いブロックまで右に歩いていき、ぶつかったらそのまま上に歩いていくプログラムを作りましょう(制限時間5分)。ただし、以下の四つのブロックは使うこととします。



[資料2] 技能面を問う課題

児童の実態を考慮し、正解だけではなく条件分岐を使おうとしている児童も「得意」とし、条件分岐も使わずに不正解の児童を「プログラミングの技能が身に付いていない児童」とした。その結果、[表2]のようになった。

[表2] プログラミングが得意か苦手かの児童数(対象児童数127名)

技能意識	苦手	得意
苦手	53名(42%)	4名(3%)
得意	49名(39%)	21名(17%)

実践の前後でのプログラミング的思考の変容を見取る際には、この分類グループを基に考察していく。以降[意識×技能]が[苦手×苦手]を苦手群、[得意×苦

手]及び[苦手×得意]グループを中間群、[得意×得意]を得意群として表記する。

### イ Jamboard を使った情報掲示板の活用

苦手群の児童は、最初こそ書き込みを躊躇する様子も見られたが、授業が進むにつれて、悩んでいることを Jamboard 上に表すようになっていった。ただし、状況だけを書き込むと、原因が分からないため、[資料3]のやりとりのようにプログラムや実行結果の画像を貼り付けることで、自分の思いを伝えようとしていた。最終的には、得意群の児童からのアドバイスを基に、プログラムを修正して自分の思いに沿った動きを実現できるようになった。



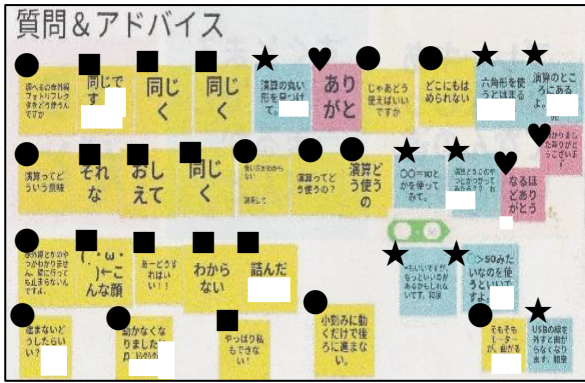
[資料3] 情報掲示板でのやり取り

得意群は、自分の課題に取り組みつつ、投稿された質問に対しアドバイスをしようという姿が見られた。少ない情報からも、質問者がどのようなことに困っているのかを推測しながらアドバイスを書き込んでいた。また、Jamboard 上にヒント集というものを作成し、質問がなくても上手くいったことを書きこめるようにしたところ、「『もし〜でなければ』を使うといちいち演算を使わなくて済むので楽です。」といったアドバイスを積極的に書くようになっていった。

第5時から、学年統一の情報掲示板を作り、クラス内のやり取りだけではなく、他クラスへのアドバイスができるようにした。

すると、中間群からも共感の言葉の書き込みが見られ、[資料4]のように書き込み数が増えていった。さらに、教師がアドバイスやヒントを書き込むことにより、困った時に情報掲示板を見て考える児童が増えた。それにより、情報掲示板の活用が広がり、結果的に苦手群へのアドバイスの量が多くなっていった。





●：質問、■：共感、★：アドバイス、♥：感謝

〔資料4〕学年統一での情報掲示板

### ウ 発表ノートを使った学習履歴の活用

授業が終わる毎に、学習履歴を作る時間を設けた。内容は、プログラムの画像と実行結果の動画、感想である。各自が作成し終わったら、グループワーク機能を使って児童同士がコメントし合うようにした。学習履歴は、単元を通して〔資料5〕のような形式で統一した。



〔資料5〕発表ノートの形式

次時の最初に、全員での振り返りの時間を設けていたのだが、苦手群の児童は、その時間以外にも何度も学習履歴を見返して、友達や教師からのコメントを参考に考えている姿が見られた。衝突回避車を作る課題の際には、演算のプログラムブロックの使い方をアドバイスしてもらったことを参考に、不等号を使ってセンサーの制御ができるようになった児童もいた。

感想の欄に質問があった場合、苦手群だけでなく得意群に対しても、教師がアドバイスを書き込んだ。得意群は、その書き込みを見て徐々にアドバイスの仕方を分かかっていき、『ずっと』のブロックを使ってみるとよいと思います。」「サーボモーターの角度を変えるといいです。」などのコメントを書き込む姿が見られるようになった。

### エ 情報ライブラリの活用

個別課題の際に、他者のプログラムを参考にできるように、クラスの共有ドライブ内に、各自のプログラムをコピーして共有した。

苦手群の中には、自分の力だけでプログラムを作成する児童もいたが、積極的に活用しようという児童も多かった。ある児童は、情報ライブラリに保存されている「壁があつたら曲がる車」のプログラムを参考にしながら、最終的には「壁があつたら曲がりながら光ってブザーが鳴る車」を作り上げた。他者のプログラムを参照することで、「繰り返し」のプログラムブロックを使う場所に気付いたり、思い通りの曲がり方をするために、曲げる角度をどのくらいにすればよいのかを把握したりすることができていた。その児童は、最後に「自分がこんなにできたんだと思うと嬉しい。」という感想を残している。

得意群の中には、積極的に情報ライブラリに自分のプログラムを入れている児童もいた。ある児童は、他の児童のために役立ててほしいという思いから、「後で修正したものを(情報ライブラリに)入れておきます。」と発言し、授業外での作成意欲を見せていた。

### (3) プログラミング的思考の分析

#### ①意識調査

本単元の最初と最後に、後藤(2021)の「高学年用自己評価アンケート」内から「思考力・判断力・表現力」を抜粋したものを6段階のリッカート尺度(6:とてもそう思う、5:そう思う、4:少しそう思う、3:あまりそう思わない、2:そう思わない、1:全くそう思わない)で実施した(〔表3〕)。

〔表3〕意識調査項目

	No	質問項目
論理的推論	1	めあてを達成するために、何をすればよいのか見通しを立てられる
	2	ものごとの「しくみ」や「きまり」を見つけ、説明することができる
分解	3	大きな動きを、いくつかの小さな動きに分けることができる
抽象化	4	めあてを達成するために、どのような手立てが必要なのか考えられる
一般化	5	ものごとの「しくみ」や「きまり」を見つけ、他のものごとにあてはめて使うことができる
組合せ	6	めあてを達成するために、手立てを順番に組合せることができる
分析	7	考えた手順にどんな問題があるのかを考え、改善することができる

そして、群ごとに事前・事後の結果を  $t$  検定を用いて比較した。苦手群の結果は、[表4]のようになった。

[表4] 苦手群の意識調査結果

	事前	事後	$t$ 値
論理的推論①	4.19 (0.90)	4.47 (1.09)	$t(52)=2.22$ *
論理的推論②	3.45 (1.10)	3.94 (1.08)	$t(52)=2.97$ **
分解	3.81 (0.98)	4.13 (1.18)	$t(52)=1.82$ <i>n.s.</i>
抽象化	4.11 (0.97)	4.51 (1.03)	$t(52)=1.82$ *
一般化	3.85 (1.20)	4.01 (0.94)	$t(52)=1.04$ <i>n.s.</i>
組合せ	3.94 (1.03)	4.38 (1.06)	$t(52)=2.14$ *
分析	3.98 (1.10)	4.43 (1.08)	$t(52)=2.46$ *

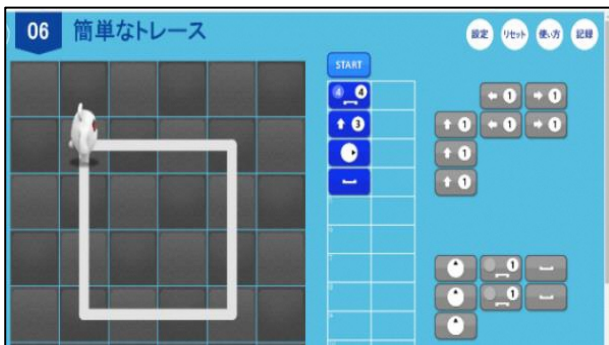
括弧内は標準偏差 \*\*: $p<.01$ , \*: $p<.05$

「論理的推論①」「論理的推論②」「抽象化」「組合せ」「分析」の項目で、事前よりも事後の方が有意に高い結果となった。なお、中間群及び得意群に関しては、事前・事後での有意差は認められなかった。

このことから、本実践の苦手群に対する有効性が示されたのではないかと考える。

## ②パフォーマンステスト(「順次」「繰り返し」)

プログラミング体験ゲーム「アルゴロジック1」を使い、「組合せ」の下位項目である「順次・繰り返し」について見取る。「アルゴロジック」は、[資料6]のように、指定されている道を通ったり決められたマスの中にある旗を取ったりするゲームである。「順次」ができており、手数がかかってもクリアできれば「○」が付く。「繰り返し」ができており、最短手順でクリアできれば「◎」が付く。



[資料6]アルゴロジック1

単元の事前・事後で異なる類似問題を解き、「○」と

「◎」の数で事前・事後の  $t$  検定を行った。

すると、苦手群では、有意差のある項目はなかった。中間群では、「順次」の項目で 0.1% 水準、「繰り返し」の項目で 5% 水準で有意差が認められた ( $t(47)=5.85, p<0.001$ )、( $t(47)=2.01, p<0.05$ )。

得意群では、「順次」の項目で 5% 水準で有意差が認められた ( $t(18)=2.40, p<0.05$ )。

このことから、プログラミングがある程度得意な児童は、本実践により「順次」や「繰り返し」を使って効率的に考えられるようになるが、プログラミングが苦手な児童はあまりできるようにならなかったことが分かる。

## ③記述分析(「条件分岐」)

田嶋(2020)の研究を参考に、プログラミング的思考の「条件分岐」要素を測るために、「天気予報をもとにして、学校に傘を持っていくかどうかを決めるにはどうしますか。」という問いに、文章で回答する調査を行った。「予報が雨だったら傘を持っていく。曇りか晴れだったら持っていない。」などの条件分岐の記述がある回答を1点、条件分岐の記述がなかったり題意と合わなかったりした回答を0点として、群ごとに事前・事後の結果を  $t$  検定を用いて比較した。

すると、苦手群、中間群では有意差が認められなかった。得意群では、5% 水準で有意差が認められた ( $t(18)=2.54, p<0.05$ )。

このことから、プログラミングが得意な児童は条件分岐を意識するようになったが、あまり得意ではない児童は、本実践だけでは条件分岐の思考が身に付かなかったことが分かる。

なお、意識調査の「組合せ」ではプログラミングが苦手な児童の思考が高まったと言えたのだが、下位項目の「順次」「繰り返し」「条件分岐」では高まっていなかったため、本研究での苦手群における「組合せ」の思考の高まりは不十分であったと考えられる。

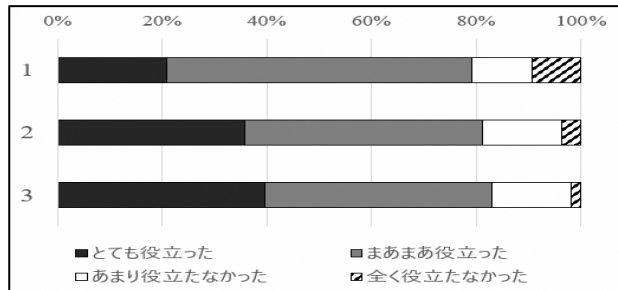
## ④苦手群における手立ての有用性

本実践終了後に、課題解決のために行った手立て [表5] がどのくらい役立ったかを、児童にリッカート尺度(4:とても役立った, 3:まあまあ役立った, 2:あまり役立たなかった, 1:全然役立たなかった)で実施した。

【表5】課題解決のための協働的な学びの手立て

	項目
1	情報掲示板
2	学習履歴のみんなからのコメント
3	情報ライブラリの友達のプログラム

「苦手群」で、各項目に対する回答の割合を調べてみると、[図3]のような結果となった。



【図3】苦手群における手立ての役立ちの割合

肯定的な回答が「情報掲示板」では 79.2%、「学習履歴のみんなからのコメント」では 81.1%、「情報ライブラリの友達のプログラム」では 83.0%となり、かなりの児童が役立てたことが伺える。しかし、相対的に見ると情報掲示板の役立ちの割合が低かった。これは、苦手な児童の中には、直接教えてもらう方が効率がよいと感じている児童もいるからであろうと考えられる。もちろん、直接教えてもらえる場合には、それでも構わないであろう。ただし、クラウドを活用したアドバイスは、いつでもどこでも誰でも参考にすることができたり、間接的に教えてもらうことで思考する場面が多くなったりするという利点がある。特に後者の利点は、プログラミング的思考と大きく関わるのではないかと考えられる。

そこで、情報掲示板を含むこれらの手立てが、プログラミング的思考にどう関わるのかを調べるため、「とても役立った・まあまあ役立った」と回答した児童を「役立った群」、「あまり役立たなかった」「全く役立たなかった」と回答した児童を「役立たなかった群」とし、その2群とプログラミング的思考の「事前・事後」の2群を変数とした2要因の分散分析を行い、Holm法による多重比較(どの項目に有意差が認められるのかを確かめる検定)を試みた。その結果、「役立たなかった群」においては、全ての項目で「事前・事後」の有意差は認められなかったが、「役立った群」において、【表6】のような有意差が認められた。

【表6】「役立った」群の有意差が認められた項目(抜粋)

		事前	事後	t 値
情報掲示板	論理的推論①	4.26 (0.14)	4.71 (0.15)	$t(51)=3.36$ **
	論理的推論②	3.60 (0.17)	4.17 (0.15)	$t(51)=3.08$ **
	抽象化	4.21 (0.15)	4.71 (0.15)	$t(51)=2.76$ **
	組合せ	3.98 (0.16)	4.52 (0.16)	$t(51)=2.41$ *
	分析	4.07 (0.17)	4.57 (0.16)	$t(51)=2.40$ *
学習履歴のみんなからのコメント	論理的推論①	4.30 (0.13)	4.70 (0.15)	$t(51)=2.86$ **
	論理的推論②	3.56 (0.17)	4.14 (0.15)	$t(51)=3.18$ **
	分解	3.88 (0.15)	4.30 (0.17)	$t(51)=2.15$ *
	抽象化	4.28 (0.14)	4.67 (0.15)	$t(51)=2.17$ *
	分析	4.19 (0.16)	4.61 (0.16)	$t(51)=2.03$ *
情報ライブラリの友達のプログラム	論理的推論①	4.14 (0.14)	4.55 (0.16)	$t(51)=3.04$ **
	論理的推論②	3.43 (0.17)	4.05 (0.16)	$t(51)=3.44$ **
	抽象化	4.16 (0.15)	4.55 (0.16)	$t(51)=2.15$ *
	組合せ	3.98 (0.16)	4.48 (0.16)	$t(51)=2.24$ *
	分解	3.98 (0.15)	4.61 (0.15)	$t(51)=3.28$ **

括弧内は標準誤差 \*\*: $p<.01$ , \*: $p<.05$

これらのことから、情報掲示板や学習履歴のコメント、情報ライブラリの友達のプログラムを役立てて活用した児童ほど、プログラミング的思考が高まったことが分かる。間接的に教えてもらうことで、ヒントを頼りにしながら自分で考えた結果であると考えられる。

情報掲示板や学習履歴の友達のコメントは、質問が「分析」、アドバイスが「論理的推論」を表現しており、特に手立てを示す「抽象化」のコメントが投稿されていることから、それらの要素が高まったのではないかと考えられる。加えて、情報掲示板では、どのプログラムブロックを使ったらよいかの「組合せ」を示しやすく、学習履歴の友達のコメントでは、細かくプログラムブロックを分けていく「分解」を示しやすかったと考えられる。

情報ライブラリの友達のプログラムは、プログラムを見てどのような仕組みになっているかを「論理的推論」しなければ活用できない。また、自分の課題に沿った部分を取り出す「抽象化」や「組合せ」「分解」を考える必要があるため、それらの要素が高まったと考えられる。

## ⑤抽出児の見取り

本研究では、意識面も技能面も苦手な児童から、担任と相談の上で抽出児を1名決め、抽出児の変容を追っていった。

Scratch の共通課題の際には、プログラミングに興味を持ち、様々なプログラミングブロックをつながながら動きを試していた。最初の学習履歴には、「(お掃除ロボットで)もう少しスムーズに消せたらよかったなと思いました。」と感想を書いていた。このことから、自分で作ったプログラムの課題に気付いていることが伺える。しかし、次時ではプログラムの中身は前時と変わらなかった。この段階では、困った時に近くの友達に直接アドバイスをもらっていたが、行き詰まってしまうことが多かった。

Scratch の個別課題の際には、自分の思ったようにプログラミングができたようで、「今回はすごく上手くいった。」という感想を書いた。この段階では、プログラムの中の数値を少しずつ調整するなど、自分で試行錯誤している時間が多かった。

ロボットカーの共通課題の際には、情報掲示板を見て、分からないことを調べようとする姿が多くなった。学習履歴には、「何が悪かったのだろう。」というように、思い通りに動かなかった原因を追究しようという記述が見られた。このことにより、ロボットカーの動きとプログラムとを関連させて一つ一つのプログラムブロックの意味を考えて取り組んでいたことが分かる。

ロボットカーの個別課題の際には、友達に直接アドバイスをもらうことがほとんどなくなり、自分で情報掲示板を確認したり、動きを確かめたりしながら活動していた。教師がそばに行った時には、困っていることを問いかけてきたが、簡単なヒントだけで、正しいプログラムブロックを見つけていた。そのことにより、既習の知識を生かすことができているのが分かる。単元の最後には、自分の課題であった「壁があったらラ

イトが光って避ける車」を完成させることができた。

これらのことから、単元が進むにつれて「分析」や「論理的推論」の思考ができるようになっていったことが分かる。

## 4 研究のまとめ

### (1) 成果

#### ①プログラミング的思考の育成

児童同士のアドバイスや他者のプログラムを参照することで、プログラミングが苦手な児童に対しても、プログラミング的思考を育めることが分かった。特に、プログラミング的思考の「論理的推論」「抽象化」「分析」の要素が高められることが明らかになった。

#### ②クラウドを活用したプログラミング教育

本研究では、デジタルホワイトボードや学習履歴でのアドバイス、情報ライブラリでのプログラムの参照など、1人1台端末を用いたクラウド上で行った。いずれの手立てもプログラミングが苦手な児童にとって有用性が認められたので、これからのプログラミング教育での活用が期待できる。

### (2) 課題

#### ①活用しやすい手立ての工夫

本研究で活用した手立ては、役立てた児童にはプログラミング的思考を高めることにつながったが、役立てられなかった児童もおり、結果としてその児童のプログラミング的思考を高められなかった。クラウドを通じたアドバイスの良さを実感できるように、単元構成を考えていく必要がある。

#### ②継続的なプログラミング教育

本研究では、苦手群のパフォーマンステスト及び記述分析において、プログラミング的思考の「順次」「繰り返し」「条件分岐」の要素を高めることが難しかった。プログラミング教育を1単元毎に考えるのではなく、教育課程で広く位置付け、継続的に取り組む必要がある。

#### 【主な引用／参考文献等】

- ・千葉市教育委員会「令和3年度千葉市学校教育の課題『21世紀を拓く』」2021年
- ・文部科学省「小学校プログラミング教育の手引 第3版」2020年
- ・藤川大祐「小学校プログラミング教育の鍵概念の検討—実践のための提言—」2020年
- ・桂本憲一「個人志向から協働的志向への転換を促すプログラミング学習の授業づくり」2018年
- ・中村雄司「コンピュータを使用したプログラミング教育の在り方—プログラミング的思考を育成する授業デザイナー—」2018年
- ・後藤 壮史「プログラミング授業による児童の変容に関する一考察—教科横断的・系統的な授業実践を通して—」2021年
- ・田嶋佑基「児童のプログラミング的思考の育成—総合的な学習の時間におけるプログラミング教育の授業開発を通して—」2020年