

プログラミング的思考を育成する問題発見・解決型の学習指導の工夫 — 思考ツールとコンピュータを相互に活用する学習活動を通して —

竹原市立中通小学校 清水 大助

研究の要約

本研究は、プログラミング的思考を育成する問題発見・解決型の学習指導について考察したものである。文献研究から、プログラミング的思考の育成には、問題発見・解決の流れで学習指導を行うことが有効と考えた。所属校で進めてきたプログラミング教育は知識・技能の伝達に重点をおいていたため、児童が問題を発見し、解決する時間が不十分で、プログラミング的思考も十分育成することができなかつた。そこで、児童にとって身近な場面から問題を見いださせ、機器の操作等の基礎的な技能を習得させた後、自ら解決方法を探る学習過程を考え、これを問題発見・解決型の学習指導とした。また、その中で、思考ツールで考えを整理させ、コンピュータでプログラムを組ませ、実行後の動きから再度思考ツール上で修正させる活動を設定した。児童は自ら問題を発見し、自分ごととして捉え、問題解決の場面では思考ツールとコンピュータを相互に活用しながらプログラミング的思考を働かせることができた。

のことから、問題発見・解決型の学習の中で、思考ツールとコンピュータを相互に活用させることができが児童のプログラミング的思考の育成に有効であることが分かった。

I 主題設定の理由

小学校学習指導要領（平成29年告示、以下「指導要領」とする。）には、情報活用能力育成を図るために学習活動の一つとして「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」¹⁾が示されている。また、小学校プログラミング教育の手引（第二版）（平成30年、以下「手引」とする。）には「児童は試行錯誤を繰り返しながら自分が考える動作の実現を目指しますが、思い付きや当てずっぽうで命令の組合せを変えるのではなく、うまくいかなかった場合には、どこが間違っていたのかを考え、修正や改善を行い、その結果を確かめるなど、論理的に考えさせることが大切です。」²⁾と示されている。黒上晴夫（2017）は「プログラミングと論理的思考力の関係を捉えると、小学校で重要なのは、考え方を可視化できるように、自己の考え方の道筋を客観的に捉える力を育成しておくことだと言えるのではないだろうか。」³⁾と述べている。また、黒上（2012）は、考え方を可視化する思考ツールの一つとしてフローチャートを挙げている⁽¹⁾。

所属校は、平成28年度から総合的な学習の時間に

プログラミング教育を取り入れ、コンピュータを活用して絵を動かして物語を作らせたり、プログラムを組んでLEDを光らせたりする学習活動を行ってきた。多くの児童は意欲的に学び、この学習活動を通して、問題解決のためには必要な手順があることに気付くことができていた。しかし、知識・技能の伝達を目的にした一方向的な講義形式の授業になりがちで、自ら問題を発見、修正し、解決することができた児童は少なかった。

そこで、問題発見・解決型の学習の中で、児童に意図する動きを設定させ、思考ツールによって手順を整理させ、コンピュータによってプログラムを組ませる。そして、プログラム実行後に問題を発見させ、再度思考ツールによって手順を修正させ、プログラムを組ませ解決させることを繰り返す学習活動を行う。このことを通して、児童のプログラミング的思考を育成できると考え、本主題を設定した。

II 研究の基本的な考え方

1 プログラミング的思考について

(1) プログラミング的思考の定義

プログラミング的思考は、「小学校段階における

プログラミング教育の在り方について（議論の取りまとめ）（平成28年、以下「まとめ」とする。）」の中で「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」⁴⁾と定義されている。

「手引」では算数科の授業を例に挙げ、プログラミング的思考を図1のようにまとめている²⁾。

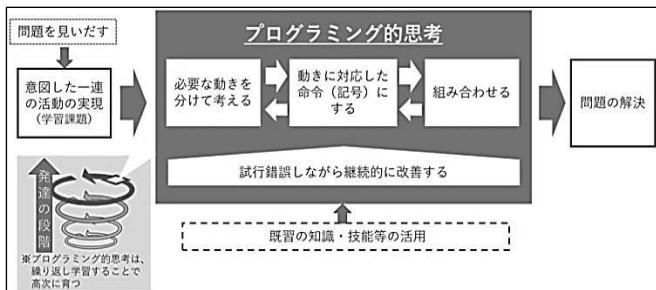


図1 プログラミング的思考

このように、問題を見いだしてから解決に至るまでの流れの中で、プログラミング的思考を働かせて考えていくことが求められている。

(2) プログラミング的思考を育成する流れの一例

株式会社ベネッセコーポレーション（以下「ベネッセ」）では、一般的な問題発見・解決の流れとプログラミング的思考を育む教育の流れを対比して図2のようにまとめ、プログラミング的思考とはコンピュータやプログラミングの概念に基づいた問題解決型の思考であると説明している³⁾。



図2 「プログラミング的思考」を育むプログラミング教育の流れ

このように、プログラミング的思考を育成するためには、問題発見・解決の流れをプログラミング教育に取り入れることが有効であると考えた。

2 プログラミング的思考を育成する問題発見・解決型の学習指導について

(1) プログラミング的思考を育成する学習過程

「指導要領」には、学習の基盤となる資質・能力として、言語能力、情報活用能力、問題発見・解決能力の三つが示されている⁴⁾。この中の問題発見・解決能力について小学校学習指導要領（平成29年告示）解説総則編（平成30年）では、物事の中から問題を見いだし、その問題を定義し解決の方向性を決定し、解決方法を探して計画を立て、結果を予測しながら実行し、振り返って次の問題発見・解決につなげていく過程で身に付く力である⁵⁾とされている。そして、この力を身に付ける過程は、プログラミング的思考を育成する際にも有効だと考える。

(2) 問題発見・解決型の学習指導

井上明（2005）は、大学で行われている情報教育が操作訓練実習に終始し、学習者は教わった操作以外取得できないことを指摘し、情報教育をPBL（問題発見解決型学習）で行うことを提案している⁶⁾。



図3 PBL（問題発見解決型学習）による情報教育の概要

この学習では、教師から与えられた知識を記憶するのではなく、学習者自らが解決方法を探して計画を立て、自発的に必要な知識や技術を調べ、実践していくことが求められている。この提案は、社会につながる情報教育として有効であると考えるが、これをそのまま小学生のプログラミング教育に適用すると、高度になり過ぎ混乱を招いてしまうと考える。

そこで、本研究では、表1のような問題発見・解決型の学習指導の段階を考えた。

表1 問題発見・解決型の学習指導の段階

段階	内容	学習形態
ステップ1	問題の発見	グループ
ステップ2	基礎的な技能の習得	教材制作
ステップ3	グループ学習	グループ
ステップ4	ユニットでの技能習得	個人
ステップ5	発表	全体

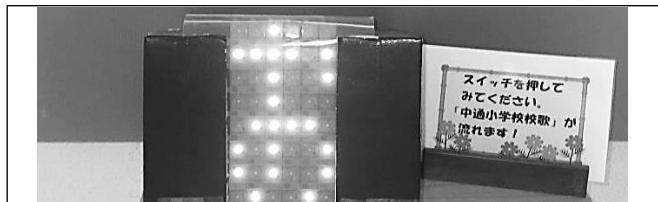
機器の操作やプログラムの組み方等の基礎的な技能の習得は全体で行い、基礎を習得し各自の工夫を加える段階で必要な技能をグループやユニットで調べ、実践させることにする。また、各段階で教材制作を進めさせ、作りながら学ばせるようにする。

以上のことから、プログラミング的思考の育成には、問題発見・解決型の学習が有効であると考える。それを踏まえ、これまで所属校で行ってきたプログラミング教育を振り返ることにする。

3 所属校のプログラミング教育について

(1) 所属校のプログラミング教育の取組

所属校では、平成28年度から高学年を中心にプログラミング教育の取組を進めている。コンピュータ上のキャラクターをプログラミングにより動かし、物語を作る活動や、8×6マスに並べたLEDを光らせ文字にし、電光掲示板を作る活動等を行ってきた。作成した物語を下級生に披露したり、電光掲示板を地元のお祭りで展示したりする等、児童に目的意識をもたせて活動に向かわせてきた。



LEDでの文字づくり（「お」の文字）

(2) 所属校のプログラミング教育の成果と課題

成果としては、児童のプログラミングに対する興味を高め、意図する動きの実現に一連の手順が必要であることに気付かせることができた。

一方、課題は二つ考えられる。一つ目は授業が問題発見・解決型の学習になっていなかったことである。作品の完成を目指すあまり、児童が失敗しないよう必要以上に情報、指示を与えてしまったり、知識、技能の伝達を目的とした一方向的な講義形式の授業になってしまったりした。そのため、児童自ら問題を見いだし、プログラミング的思考を働かせながら解決する機会を十分与えることができなかつた。二つ目は、思い付きや当てずっぽうで活動を進める児童がいたことである。物語づくりの活動では、考えていたストーリーとは異なる動きになってしまい、プログラムを修正することなく、物語の方を変えてしまう児童がいた。事前に作りたいもののイメージを固め、それに向けて活動を進める必要がある。

(3) 所属校のプログラミング教育の改善案

これら課題から、本研究では二つの改善を行う。

一つ目は前述した通り、問題発見・解決型の学習を取り入れることである。児童自ら問題を見いだせるように、身近なテーマを設定し自分ごととして問題解決にあたらせる。また、与える情報や指示を精選し、児童が問題解決の方法を探る時間を確保する。

二つ目は児童が思い付きで活動を進めないよう、学習活動に工夫を加えることである。そのために、思考を可視化し、プログラミングにつなげる手立てを考えることや、どのようなプログラミングの学習活動がより適切なのか考えることが必要である。

4 思考ツールとコンピュータを相互に活用する学習活動について

(1) 思考ツールを活用する学習活動

思考を可視化し、客観的に捉えるための方法として、思考ツールを用いることが考えられる。黒上

(2012) は思考ツールの一つとしてフローチャートを挙げ、プログラムの流れを論理的に表現するための記述方法であると説明している⁽⁷⁾。大森康正 (2018) は「プログラムでは、フローチャートのようなアルゴリズム等の図式化技法や、UMLなどの仕様記述言語などを同時に活用することも有用です。」⁵⁾と述べ、プログラムを組む際に図で考えることの有効性を指摘している。

そこで、本研究では思考ツールを用いコンピュータにさせたい動きを整理させ、プログラムを構造的に見えるようにまとめさせ、コンピュータ上でのプログラミングをしやすくなるような手立てを行う。

(2) コンピュータを活用する学習活動

ア ビジュアル型プログラミング言語を用いた学習活動の特徴

コンピュータ上でプログラムを組む際に、ビジュアル型プログラミング言語（以下「ビジュアル型」）を使用する方法がある。命令が書かれたブロックを並べ替え、組み合わせることによってプログラムを組むことができるので、文字を打ち込まなくてもよい。利根川裕太・佐藤智 (2017) は「『ビジュアル』タイプは、文法によるエラーなどがなく、『順次』『繰り返し』『分岐』といったプログラミングの考え方の学習に集中しやすいです。」⁶⁾と述べている。

「ビジュアル型」の特徴を、赤堀侃司 (2018) は、「構造化」「可視化」「操作法」の3観点で調査している。「構造化」とは命令の流れがブロックの並びで表されており構造的に見やすい特徴、「可視化」

とはプログラム実行後キャラクター等の動きから即座にプログラムの成否が分かる特徴、「操作法」とは命令ブロックを並べ替えるだけでプログラムを書き換えることができる特徴である。この調査では「可視化」「構造化」「操作法」の順で、認知的効果が高いことが明らかになった⁽⁸⁾。

これらの特徴から、プログラミング導入期である小学生においては「ビジュアル型」を用いてプログラミングを体験することが効果的であると考える。

イ ロボットを用いた学習活動の特徴

作ったプログラムをロボットに転送し制御する学習方法である。利根川ら（2017）は「特徴としては実際に“物”を動かすことで、児童の興味を強く喚起することができるのです。」⁽⁷⁾と述べている。本研究で使用する「アーテックロボ」⁽⁹⁾は「ビジュアル型」でプログラムを組むことができるので小学生でも取り組みやすい。また、車のように走らせたり、LEDを光らせたり、音を鳴らしたりすることもできるので、活動の幅を広げることができる。

（3）思考ツールとコンピュータの特徴を相互に活用する学習活動

前述した「ビジュアル型」の特徴のうち「可視化」しやすいことが欠点になる場合がある。例えば、適当に命令を並べたり、数値を変えたりして偶然動きが実現してしまう場合である。思い付きや当てずっぽうの活動になることは「手引」でも危惧されており⁽¹⁰⁾、そうならないための手立てが必要である。

草野俊彦（2018）は「プログラミングするには、コンピュータで何をどうするのかを考えるプログラム設計という作業が必要です。」⁽⁸⁾と述べている。事前にプログラム設計をすることで、ゴールイメージができ、思い付きや当てずっぽうの活動になりにくくなると考える。また、プログラムの構造や流れが可視化され、修正する際、共有する際、一般化する際にも有効に働くと考える。一般的にプログラム設計はフローチャートやUML（統一モデリング言語）等の図で行われる。本研究においては、これらの図を思考ツールとして取り扱うこととする。

本研究では、事前に思考ツールでプログラム設計させてからコンピュータでプログラムさせる。プログラムは「ビジュアル型」で組ませ、ロボットも活用させる。実行後に不具合があれば思考ツールに戻り問題点を見付けさせ、コンピュータで修正し、実行して動きを確かめさせる。うまくいったプログラムは思考ツール上に残させるようにし、他のプログラムを考える時の参考にさせる。以上のように思考

ツールとコンピュータを相互に活用する学習を行うことによって、児童のプログラミング的思考を育成することができると考える。

5 本研究における学習指導の工夫について

（1）問題発見・解決型の学習指導の具体的な工夫 ア 身近なテーマの設定

児童が自分ごととして考えたくなる場面設定の工夫をすることで、主体的に問題解決にあたり、粘り強く試行錯誤するようになると考える。具体的には地域への貢献等、児童にとって身近な問題場面とするため、総合的な学習の時間において竹原の魅力を紹介することを課題として学習する場面を設定する。竹原市を訪れる観光客数の現状から問題を見いださせ、解決するための取組を考え、プログラミング的思考を働かせながら問題解決にあたらせたい。

イ 与える情報、指示の精選

基礎的な知識、技能については一斉指導で行うが、児童に与える情報、指示は精選して行う。また、児童自らが工夫を加える場面では、必要な知識、技能を自ら考え、調べられるようにする。教師は手助けやアドバイスにより児童の問題解決を支える。

（2）思考ツールとコンピュータを相互に活用する学習活動の具体的な工夫

学習活動の工夫としては、4の（3）で述べたとおり、思考ツールとコンピュータを相互に活用させながら学習に取り組ませたい。

本研究で取り扱う思考ツールは、児童の実態や活動内容を考慮し、プログラムの流れを構造的に表現する図を考えた（以下「プログラム設計図」とする）。「プログラム設計図」は「ビジュアル型」の「構造化」に近付けるために、プログラムの流れを構造的に見ることができるような図にする。しかし、図に命令を書き込む方法では修正や改善がしにくくなるので、付箋を使い「ビジュアル型」の「操作法」に近付ける。付箋であれば命令の順番を変えることも、命令自体を加除することも簡単に行うことができる。また、今回制作するプログラムは、ボタンや写真、LEDごとに別のプログラムを組まなければならず、各命令が相互に関連しながら動くものを作らなければならない。そこで、「プログラム設計図」では、プログラムの構造やつながりを、付箋や矢印を使って表現できるようにする。

思考ツールとして「プログラム設計図」、コンピュータとして「ビジュアル型」とロボットを相互に活用させることで、児童のプログラミング的思考

を育成することができると考える。
この学習の効果について本研究では検証を行う。

III 研究の仮説及び検証の視点と方法

1 研究の仮説

問題発見・解決型の学習の中で、思考ツールとコンピュータを相互に活用する活動を行えば、プログラミング的思考を育成することができるであろう。

2 検証の視点と方法

(1) 検証の視点と方法の概要

検証の視点と方法について、表2に示す。

表2 検証の視点と方法

検証の視点	検証の方法
思考ツールとコンピュータを相互に活用する学習活動は児童のプログラミング的思考を育成するために有効であったか。	<ul style="list-style-type: none"> ○ 児童への四段階評定尺度法による事前・事後アンケート調査 ○ 授業における児童の行動分析 ○ ワークシート・ノートの分析 ○ 振り返りの分析

(2) アンケートについて

表3は「ベネッセ」が「まとめ」を基に作成した、プログラミングで育成する資質・能力の評価規準(試行版) (2018) から抜粋し、整理したものである⁽¹¹⁾。

表3 「ベネッセ」によるプログラミング的思考の整理

観点	説明
論理的に考えを進める	コンピュータの働きを自らの問題解決で使うために論理的推論を行うこと。
動きに分ける	自分が意図する一連の活動を実現するために、大きな動き(事象)を解決可能な小さな動き(事象)に分割すること。いわゆる分割。
記号にする	分割した動き(事象)の中から適切な側面・性質だけを選び出し、他の部分を除くこと。いわゆる抽象化。
一連の活動にする	記号(動き)の類似の部分を特定して、別の場合でも利用できる内容にすること。いわゆる一般化。
組み合わせる	目的に合わせてよりよい手順を創ること。
振り返る	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、意図した活動に近づいているかどうか評価すること。

表4 事前・事後アンケート

	質問事項	観点
1	どうやつたら「させたい動き」になるか考え、見通しを立てることができる。	論理的に考えを進める
2	「させたい動き」が、どのような動きの集まりによってできたものなのか考えることができる。	動きに分ける
3	「させたい動き」にするために、どんな命令ブロックが必要か選ぶことができる。	記号にする
4	「させたい動き」にするために、選んだ命令ブロックの並べ方や組合せ方を考えることができる。	組み合わせる
5	「させたい動き」がうまくできるまで、プログラムを直すことができる。	振り返る
6	「させたい動き」になったプログラムを活用して、ちがう動きを作ることができる。	一連の活動にする
7	これまでのプログラミングの学習活動について、感想を自由に書いてください。(自由記述)	

表4は、「ベネッセ」が整理したプログラミング的思考の評価の観点を参考に作成したアンケートの質問事項である。アンケートは研究授業の事前・事後に、四段階評定尺度法を用いて実施する。

IV 研究授業

1 研究授業の内容

○ 期間 平成30年11月26日～平成30年12月14日
○ 対象 所属校第6学年(1学級20人)
○ 単元名 大好き！私たちの町竹原
○ 目標 竹原の抱える課題について考え、地域のよさについて改めて調べ、まとめたものを発信することができる。

2 研究授業の概要

(1) 単元指導計画

表5は、総合的な学習の時間の「大好き！私たちの町竹原」の単元指導計画である。

表5 総合的な学習の時間の単元指導計画(全6時間)

時	活動	評価規準(評価の観点)	PBLの段階
1	竹原の観光客の推移から、観光名所を紹介するための方法を考えることができる。	観光名所を紹介する方法を考えることができる。(一連の活動にする)	問題の発見
2	ボタンを押したら写真が表示されるプログラムの流れを考え、プログラム設計をすることができる。	させたい動きにするためにどんなプログラムを組めばよいか見通しを立てることができる。(論理的に考えを進める)	基礎技能取得 教材の制作
	させたい動きになるようなプログラムを作ることができる。(動きに分ける、記号にする、組み合わせる)	させたい動きになるようなプログラムを作ることができる。(動きに分ける、記号にする、組み合わせる)	
3	もどるボタンを押したら最初の画面に戻るプログラムの流れを考え、プログラム設計をすることができる。	実行後の動きからプログラム設計の修正をすることができる。(振り返る)	基礎技能取得 教材の制作
	観光名所の写真に説明を加える方法を考え、プログラム設計をすることができる。	させたい動きになるようなプログラムを作ることができる。(動きに分ける、記号にする、組み合わせる)	
4	実行後の動きからプログラム設計の修正をすることができる。(振り返る)	させたい動きになるようなプログラムを作ることができる。(動きに分ける、記号にする、組み合わせる)	グループ学習 教材の制作
	させたい動きになるようなプログラムを作ることができる。(振り返る)	させたい動きになるようなプログラムを作ることができる。(動きに分ける、記号にする、組み合わせる)	
5	観光名所の場所を示す方法を考え、プログラム設計をすることができる。	実行後の動きからプログラム設計の修正をすることができる。(振り返る)	グループ学習 教材の制作
	させたい動きになるようなプログラムを作ることができる。(動きに分ける、記号にする、組み合わせる)	させたい動きになるようなプログラムを作ることができる。(動きに分ける、記号にする、組み合わせる)	
6	加えたい動きを考える時に、これまで作ってきたプログラムを参考にしながら、プログラム設計を考えることができる。	これまで作ったプログラムを参考にしながら、プログラム設計を考えることができる。(一連の活動にする)	ユニットでの技能取得 教材の制作
	これまで作ったプログラムを参考にしながら、プログラム設計を考えることができる。(一連の活動にする)	これまで作ったプログラムを参考にしながら、プログラム設計を考えることができる。(一連の活動にする)	

(2) 研究授業の様子

ア 問題を見いだし解決法を考える場面（第1時）

はじめに竹原市を訪れる観光客数の推移をグラフにしたものを見せて、児童は一部観光地を除いて減少傾向にあるという問題を見いだしていた。次に解決策として、竹原の観光名所を広く紹介したいという案が出て、その方法としてパンフレットや、パソコンを使って紹介したいという考えが出た。その後観光案内所の様子を調べた結果、タッチパネル式の観光案内表示（以下「案内表示」）というものがあることを知り、自分たちも同じようなものを作りたいという意見が出た。その際、これまでプログラミングの授業で学んできた、LEDを光らせる技術も使ってみたいという意見も出た。

1. 竹原の観光名所を伝える方法にはどんな方法があるでしょう。
インターネット You Tube フェイスブック パンフレット
・パソコンに、竹原のいい所をトーネとのせる。
PR プログラミング パンフレット

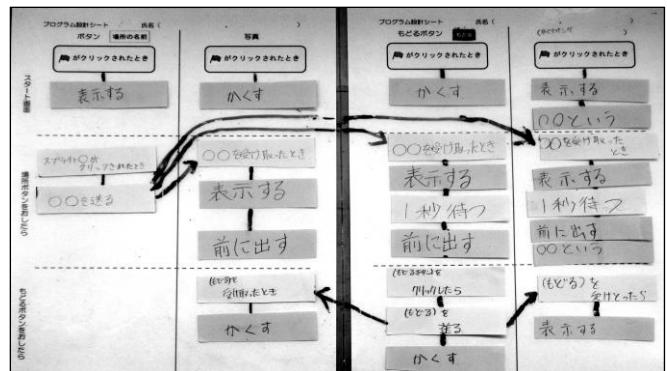
児童が考えた観光名所を伝えるための方法

イ 思考ツールとコンピュータを相互に活用して いる場面（第4時）

「案内表示」の制作を始めて3時間目なので、「プログラム設計図」の作り方やプログラムを組むことに慣れてきた場面である。前回まで、場所名が書かれた画面上のボタンを押したらその場所の写真が提示され、戻るボタンを押したら最初の画面に戻るというところまでのプログラムを組むことができていた。本時ではキャラクターに観光名所の紹介をさせたいという意見が出て、そのためにどんなプログラムを組んだらよいか考え、「プログラム設計図」上で表現していた。その際、前回作ったプログラムを参考にしながら考えることができて、パソコン上でプログラムを組むときにも、「プログラム設計図」を見ながら活動を進めることができていた。



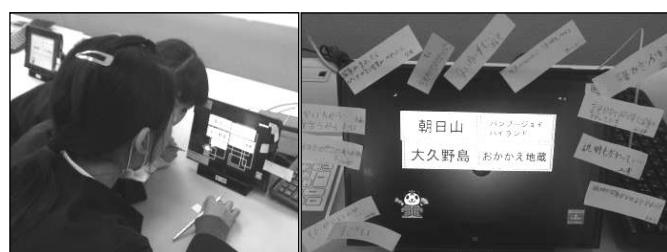
チームで「案内表示」のプログラムを組む様子



第4時に児童が作成した「プログラム設計図」

ウ 各自で加えたい動きを考え解決法を調べ実践 している場面（第6時）

各グループが制作した「案内表示」を見合い、感想を伝え合せた後、各自で個別に「案内表示」の制作を行った。観光名所の説明を増やしたり、音が鳴るようにしたりする等、各自が加えたい工夫を考え、独創性のある作品に仕上げていった。操作やプログラミングに関する教師からの一斉指導はせず、させたい動きが似ているメンバー毎に集め一度だけ説明し、その後はメンバー同士で教え合わせるようにした。前回までは消極的だった児童も今回は主体的に活動に参加し、他のメンバーに教える姿も見せていた。このときの振り返りに「一度の説明ですぐ理解することができたのは、これまでのプログラミングの授業で『プログラム設計図』を使って考え発展させてきたからだと思う」という記述があった。



各グループの制作した「案内表示」を見合う様子（左）

感想を付箋に書き画面の周りに貼っている様子（右）

V 研究授業の分析と考察

1 問題発見・解決型の学習の中で思考ツールとコンピュータを相互に活用する活動をしたことで、児童のプログラミング的思考を育成することができたか

(1) アンケートの分析と考察

表6は、質問事項1～6における、研究授業の事前・事後のアンケート結果である。

表6 事前・事後アンケート結果

質問事項（観点）		授業	A	B	C	D	p 値	
1 どうやったら「させたい動き」になるか考え、見通しを立てることができる。（論理的に考えを進める）	前	5	13	2	0		0.046	*
	後	9	11	0	0			
2 「させたい動き」が、どのような動きの集まりによってできたもののか考えることができる。（動きに分ける）	前	3	10	7	0		0.003	**
	後	11	7	2	0			
3 「させたい動き」にするために、どんな命令ブロックが必要か選ぶことができる。（記号にする）	前	10	6	4	0		0.052	
	後	13	7	0	0			
4 「させたい動き」にするために、選んだ命令ブロックの並べ方や組合せ方を考えることができる。（組み合わせる）	前	8	8	4	0		0.049	*
	後	11	9	0	0			
5 「させたい動き」がうまくできるまで、プログラムを直すことができる。（振り返る）	前	11	8	1	0		0.404	
	後	11	7	2	0			
6 「させたい動き」になったプログラムを活用して、ちがう動きを作ることができる。（一連の活動にする）	前	4	13	3	0		0.057	
	後	11	6	3	0			

A : よくあてはまる B : ややあてはまる
C : あまりあてはまらない D : あてはまらない
t 検定の数値は、有意確率（片側）を示す。P < 0.05
**は P < 0.01 *は P < 0.05 (n=20)

アンケート結果から 1, 2, 4 の項目において t 検定によって、有意な差が見られた。特に質問事項 2 「動きに分ける」については p 値が 0.01 を下回っており、大きく変容したことが分かる。

また、1, 3, 4 の項目は、授業後に 100% の児童が肯定的な回答をしており、授業を行ったことで各項目に自信をつけた児童が多いことが分かる。

一方、質問事項 5 「振り返る」については t 検定によって有意な差が見られず、肯定的な回答をした児童の割合も 5 % 下がった。考えられる原因の一つ目は、事前から 95% の児童が肯定的評価をしていたことから、これまでの所属校のプログラミング教育で「振り返る」力は身に付いており、今回の研究では大きく変容しなかったのではないかという点である。二つ目は、振り返る場面での思考ツールの活用が不十分だった点である。そこで、改善策として思考ツールのより一層の活用が考えられる。「プログラム設計図」を他チーム同士で見合って問題点を考えさせたり、教え合わせたりすることで、問題の見つけ方を共有し、よりよいプログラムに修正する等の「振り返る」力を育成することができると考える。

質問事項 7、自由記述の項目は、記述内容の分析を行った。「プログラム設計図」や付箋等について書かれていたのは 20 名中 17 名で、その全てが、事前に考えを整理したことでプログラミングがしやすくなった、というものであった。また、難しかったけど楽しかった、という感想を書いている児童が 10 名

いた。児童が活動の難しさを感じたのは、これまで所属校で行ってきたプログラミング教育に比べて、与えられる情報や指示が少なかったことが要因であると考える。だからこそ児童は自ら考え独自の作品を作り、達成感を味わうことができたため、楽しかった、という感想をもつことができたと推測する。

(2) 児童の行動及びワークシートの分析と考察

ア プログラミング的思考にあまり変容が見られなかった児童の様子

表 7 は、上記児童のアンケート結果である。

表7 あまり変容が見られなかった児童のアンケート結果

質問事項		1	2	3	4	5	6
児童 A	事前	A	C	C	C	C	C
	事後	B	B	B	B	C	C
児童 B	事前	C	B	C	C	B	C
	事後	B	C	B	B	B	C
児童 C	事前	C	C	C	B	B	B
	事後	B	C	B	B	C	C

児童 A は、授業前ほぼ全て否定的に回答し、授業後は一部肯定的に変容したものの、全体としては大きな変容が見られなかった。振り返りには、第 3 時「工夫して考える手順がとても難しかった」第 5 時「LED の接続は出来たらすごい」と記述し、授業を進めるごとに内容への興味が高まったことが分かる。しかし最後の振り返りでは「プログラミングの使い方が難しかった」と否定的感想に戻っていた。このチームの「案内表示」は、基本部分はできていたが、任意に音を鳴らしたいと試行錯誤した末できずに終わっており、否定的評価につながったと考えられる。児童 A は、問題を自分ごととして捉え、粘り強く試行錯誤していたので、その過程を評価するような声掛けをする必要があったと考える。

児童 B, C ともに授業前後であまり自己評価が変わらなかった。二人に共通していた特徴が二つあり、プログラミングに苦手意識があり活動に消極的だった点と、休んだ日があり活動が先に進んでしまった点である。その結果メンバーに任せてしまう場面が多く見られた。しかしどちらの児童も、個別でプログラムを組む場面では積極的に活動を進めており、その日の振り返りからも達成感を得ていたことが分かった。一人になったことで、ようやく自分ごととして考えることができたのではないかと推測する。

以上のことから、学習内容や授業の進め方に課題

があったと考える。今回の研究では問題発見・解決型の学習として、基礎的事項を一斉授業で教え、発展段階で各自が工夫を加えることを目指したが、今回は求めた基礎が多すぎたのではないかと考える。その結果、説明場面が増え、ついていけない児童がより消極的になってしまったと推察する。児童の思いを実現させるために、どこまでを基礎として教え、どこからは各自に任せるか、再考する必要がある。

イ 児童全体の様子

問題発見の場面では、全ての児童が解決策を考えることができておらず、問題を自分ごととして捉えさせることができたと考える。その後の活動で、プログラミングへの苦手意識から消極的になってしまった児童はいたが、最終的には各自がオリジナルの作品になるまで試行錯誤していたので、問題発見・解決型の学習を取り入れたことは有効であったと考える。

問題解決の場面で児童は「プログラム設計図」を活用しながら解決法を探ることができていた。振り返りには「『プログラム設計図』を使うことで簡単に作業を進めることができた」との記述や、「もどるボタンのプログラムを考える時に、それまで作ったプログラムを基にして考えるとすぐ作ることができた」との記述が見られた。問題発見・解決型の学習の中で、思考ツールを活用させたことで、プログラミング的思考を育成することができたと考える。

また、アンケート結果が肯定的評価に変容した児童に共通した行動として、他者に教える姿が見られた。他者に教える体験が、児童のプログラミング的思考を育成し、より深い理解と自信につながることがうかがえる。しかしこれは、教える側にとつては有効に働くが、教わる側は言われた通りやるだけになり、有効に働く場合がある。そこで、両者にとって効果的な活動とするため「プログラム設計図」を使った教え合いの活動を行わせる。それぞれの「プログラム設計図」を持ち寄り見比べさせて、問題点に気付かせる。教える側が修正点を伝える際は「プログラム設計図」上で示しながら理由も併せて伝えさせるようにする。このような教え合いの活動を通して、教える側だけでなく教わる側もプログラミング的思考を育成することができると考える。

VI 研究のまとめ

1 研究の成果

- 問題発見・解決型の学習の中で、思考ツールとコンピュータを相互に活用する活動を行うこと

が、プログラミング的思考の育成に有効であることが分かった。

2 研究の課題

- 「振り返る」力を十分に育成できなかったことや、活動に消極的になってしまった児童がいたことが課題として残った。改善案としては、プログラムの順序や組合せの意味を教え合わせる活動を取り入れることや、教えるべき基礎的事項の精査をすること等が考えられる。今後の実践でより効果的な学習方法を考える必要がある。
- 他教科、他学年における、プログラミング的思考の育成方法について考える必要がある。

【注】

- (1) 黒上晴夫 (2012) : 『シンキングツール～考えることを伝えたい～』 学習創造フォーラム pp. 24-25に詳しい。
- (2) 文部科学省 (平成30年) : 『小学校プログラミング教育の手引 (第二版)』 pp. 14-15 に詳しい。
- (3) 株式会社ベネッセコーポレーション (2018) : 「図で解説『プログラミング的思考』とは」に詳しい。
<https://beneprog.com/2018/07/13/computationalthinking/>
- (4) 文部科学省 (平成29年告示) : 『小学校学習指導要領』 p. 19に詳しい。
- (5) 文部科学省a (平成30年) : 『小学校学習指導要領 (平成29年告示) 解説総則編』 東洋館出版社 p. 51に詳しい。
- (6) 井上明 (2005) : 「PBL (Problem-Based Learning) による問題発見解決型情報教育」に詳しい。
http://www.juce.jp/archives/ronbun_2005/09.pdf
- (7) 黒上晴夫 (2012) : 前掲書 pp. 24-25に詳しい。
- (8) 赤堀侃司 (2018) : 『プログラミング教育の考え方とすぐに使える教材集』 ジャムハウス pp. 48-60に詳しい。
- (9) 株式会社アーテックホームページ、「アーテックロボ」に詳しい。
<https://www.artec-kk.co.jp/artecrobo/ja/>
- (10) 文部科学省 (平成30年) : 前掲書 p. 15に詳しい。
- (11) 株式会社ベネッセコーポレーション (2018) : 「プログラミングで育成する資質・能力の評価規準」に詳しい。
<http://benes.se/keyc>

【引用文献】

- 1) 文部科学省 (平成29年告示) : 前掲書 p. 22
- 2) 文部科学省 (平成30年) : 前掲書 p. 15
- 3) 黒上晴夫、堀田龍也 (2017) : 『黒上晴夫・堀田龍也のプログラミング教育 導入の前に知っておきたい思考のアイディア』 小学館 p. 6
- 4) 文部科学省 (平成28年) : 『小学校段階におけるプログラミング教育の在り方について (議論の取りまとめ)』 に詳しい。
- 5) 大森康正 (2018) : 「どうなる?学校のプログラミング教育 学習指導要領から読み解く、現在・過去・未来」 松村太郎・山脇智志・小野哲夫・大森康正 (2018) 『プログラミング教育が変える子どもの未来』 翔泳社 p. 128
- 6) 利根川裕太・佐藤智 (2017) : 『先生のための小学校プログラミング教育がよくわかる本』 翔泳社 p. 127
- 7) 利根川裕太・佐藤智 (2017) : 前掲書 p. 126
- 8) 草野俊彦 (2018) : 『教養としてのプログラミング的思考 今こそ必要な「問題を論理的に解く」技術』 SBクリエイティブ p. 28

【参考文献】

- 文部科学省 (平成30年) : 『小学校学習指導要領 (平成29年告示) 解説総合的な学習の時間編』 東洋館出版社
石戸奈々子 (2017) : 『プログラミング教育がよくわかる本』 講談社