

「プログラミング的思考」を汎用的な力に高めることを目指した教科等横断的な取組

総合支援部小中学校支援課 長期研修員 八木 光隆

1 主題設定の理由

「プログラミング的思考」は、時代を越えて普遍的に求められるものであり、情報技術が飛躍的に進展する未来社会にあっては不可欠な能力であるとされている。

来年度から全面実施される小学校学習指導要領（平成 29 年告示）（以下、新学習指導要領という。）等において、文部科学省は、「学習の基盤となる資質・能力」の一つとして、情報活用能力（「プログラミング的思考」を含む）を取り上げ、教科等横断的な視点で育成していくこととした。続いて、新学習指導要領解説総則編（以下、解説総則編という。）及び「小学校プログラミング教育の手引（第二版）（以下、手引という。）」において、小学校段階でプログラミングに取り組むねらいを示し、教科等の内容を指導する中で実施する場合には、各教科等の特質に応じて「各教科等での学びをより確実にすること」とした。これらのことから、小学校においては、教育課程全体を見渡し、プログラミングを実施する単元を計画的に位置付けていく必要があり、一貫して「プログラミング的思考」を育成することが求められると考える。

しかし、その育成を取り巻く現状においては厳しいものがある。

静岡県教育委員会の調査では、外国語の教科化への対応等の理由から、小学校第 6 学年の週当たりの平均授業時数は、平成 27 年度の週 27.8 時間から令和元年度は週 28.3 時間に増加している。今後、授業時数を更に増加し、予備時数などを用いて、学校独自でプログラミングを体験する時間を確保することは困難な状況である。

また、令和 2 年度から使用される検定教科書から、プログラミングを体験する学習活動を取り出すと、算数科や理科などの一部の教科に著しく偏っており、学習内容は単元とは別にトピックスとして扱われているものがほとんどであった。

こうした現状から、「プログラミング的思考」は、各教科等の適切な学習場面で育成を図ることが必要であり、その中で育まれた「プログラミング的思考」を発揮させることにより、各教科等の学びがより確実になることが期待される。そして、各教科等の学習で育まれた「プログラミング的思考」は汎用化されて、その後のプログラミング体験や日常に生かされる思考になることが大切である。

このように、「プログラミング的思考」を汎用化させるためには、教師が「プログラミング的思考」を育成する上で大切な視点をもつ必要があると考え、それを明らかにすることが重要であると考えた。以上のことから、研究主題を『「プログラミング的思考」を汎用的な力に高めることを目指した教科等横断的な取組』と設定した。

2 研究の目的

児童が発揮した「プログラミング的思考」が、その後のプログラミング体験に生かされ、

日常化されることが大切である。そこで、「プログラミング的思考」を汎用的な力に高めることを目指し、教師が「プログラミング的思考」を育成する上での大切な視点について明らかにする。

3 研究の方法

- (1) 新学習指導要領等や先行研究を基にして、教師が「プログラミング的思考」を育成する上での大切な視点について考える。
 - ・「プログラミング的思考」の観点を解釈し、整理する。
 - ・児童が発揮する「プログラミング的思考」には、どのような共通点があるかを仮定する。
 - ・児童が発揮する「プログラミング的思考」には、各教科等に応じて適用方法の違いが表れることを想定し、適用方法の違いを仮定する。
- (2) 適用方法の違いが表れやすいと考えられる内容を関連付けて、授業を構想する。
- (3) 授業実践を行い、「プログラミング的思考」を発揮する際の、共通点及び適用方法の違いに関するデータを授業中の児童の表れから得る。
- (4) 児童の表れや事後の調査を分析し、教師が「プログラミング的思考」を育成する上での大切な視点について明らかにする。

4 研究の内容

(1) 「プログラミング的思考」の観点を整理

「プログラミング的思考」については、解説総則編において、「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」と説明され、手引で図1が示された。これらを基に、「プログラミング的思考」の観点から児童の表れを捉えるとき、より具体的な解釈の必要性を感じ、表1に整理した。

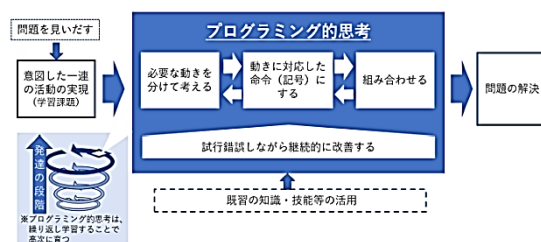


図1 「プログラミング的思考」

『小学校プログラミング教育の手引(第二版)』より

表1 「プログラミング的思考」の観点を解釈(図1と対応)

観点	必要な動きを分けて考える	動きに対応した命令(記号)にする	組み合わせる
解釈	一連の動きを切り分ける。	「必要な動きを分けて考える」で切り分けた手続きを、プログラムへと変換する。	「動きに対応した命令にする」までで整理した断片的なプログラムを組み合わせる。
観点	試行錯誤しながら継続的に改善する		
解釈	目的を実現するために、プログラムを改善する。		

「試行錯誤して継続的に改善する」の観点は、「プログラミング的思考」の全ての観点に働くものと捉えた。これらの観点を解釈を基に、プログラミングを実施する授業を構想したり児童の表れを捉えたりすることを、教師が「プログラミング的思考」を

育成する上での大切な視点の第一として仮定する。

(2) 「プログラミング的思考」を発揮する際の共通点

小学校段階では、教科等横断的な視点で「プログラミング的思考」の育成を図るため、「プログラミング的思考」を一貫して育成するための共通点には、どのようなものがあるかについて調査した。

共通点の一つは、「最適化」である。「最適化」とは、様々な条件に対して、目的や対象に応じて最適な状態に近付けることであると考えられる。松田（2014）の指摘や中学校技術・家庭科の技術分野の内容「D情報の技術」が参考になる。

松田は、高等学校の共通教科「情報」において、「汎用的な問題解決力をどう育てるか」に着目し、情報の表現、記録、計算、通信の特性を考慮した視点とともに、技術教育の考え方である最適化の重要性について述べている。

また、中学校学習指導要領（平成29年告示）解説技術・家庭編（以下、解説技術・家庭編という。）において、情報の「技術の見方・考え方」については、「生活や社会における事象を、情報の技術との関わりの視点で捉え、社会からの要求、使用時の安全性、システム、経済性、情報の倫理やセキュリティ等に着目し、情報の表現、記録、計算、通信の特性等にも配慮し、情報のデジタル化や処理の自動化、システム化による処理の方法等を最適化すること」などが考えられるとされた。

これらのことから、「D情報の技術」の内容では「最適化すること」が大切となる。したがって「D情報の技術」の内容の一部として取り扱われるプログラミングも、小学校プログラミング教育においても、同様に「最適化」を共通点とすることが重要だと考える。

共通点のもう一つは、解説技術・家庭編で示されている「順次、分岐、反復」といったプログラムの構造を支える要素が参考になる。プログラミングの体験には、「順次、分岐、反復」が含まれていることから、これらに着目して学習内容を関連させることが効果的であると考える。

そのため、「最適化」「順次、分岐、反復」という共通点で授業内容を捉えることを、教師が「プログラミング的思考」を育成する上での大切な視点の第二として仮定する。

(3) 「プログラミング的思考」を発揮する際に表れる適用方法の違い

新学習指導要領で示された資質・能力の育成にあたっては、各教科等の特質に応じた見方・考え方が働くものであるとされた。このことから、各教科等での学びをより確実にするプログラミングにも、各教科等の特質に応じた見方・考え方が働くと考えられる。

したがって、教科等の内容を指導する中でプログラミングの体験を行う場合には、各教科等で解決の仕方が異なるため、「プログラミング的思考」を発揮する際にも、その適用方法の違いが表れるのではないかと考えた。

本研究では、適用方法の違いが表れやすいと考えられる教科でプログラミングを体験することができるよう、教科等横断的な視点で授業構想に努めた。

ア 算数科で発揮する「プログラミング的思考」の適用方法の捉え

算数科における数学的な見方・考え方は、「事象を、数量や図形及びそれらの関係などに着目して捉え、根拠を基に筋道を立てて考え、統合的・発展的に考えること」と整理された。算数科の学習活動としてプログラミングを用いて問題を解決する際には、児童は、この数学的な見方・考え方を働かせ、「プログラミング的思考」を適用することが考えられる。

例えば、コンピュータを用いて正多角形の作図を行う学習では、図形を構成する要素に着目し、正多角形の意味を基に、「反復」の命令を組み合わせることでプログラムを簡潔に表現したり、更に一部の数値を変えたりすることで、正 n 角形について統合的・発展的に考察する。

算数科においては、このようなプログラミングを行うことで、児童は問題の解決には必要な手順があること、命令を順序よく組み合わせることが大切であること、正確な繰り返し作業が必要な時にコンピュータを用いるとよいことに気付くことができる。

このようなことから、算数科で制作される「最適化されたプログラム」は、「問題解決のために、筋道を立てて考え、必要な手順を順序よく組み合わせ、簡潔・明瞭・的確な表現に改善でき、統合的・発展的に考えた結果、一般化されたプログラム」という特徴があると考えられる。児童が算数科で発揮する「プログラミング的思考」の適用方法の捉えを、表 2 に示す。

表 2 算数科での適用方法の捉え

「プログラミング的思考」	算数科での適用方法の捉え
必要な動きを分けて考える	・事象を数量や図形及びそれらの関係についての概念等に着目し、その特徴や本質を捉えたことを基にして、必要な動きを切り分ける。
動きに対応した命令（記号）にする	・事象を数量や図形及びそれらの関係についての概念等に着目し、捉えた一つの動きに対応した命令へと変換する。
組み合わせる	・概念や原理に基づき、断片的な命令を、筋道を立てて考え、順序よく組み合わせる。
試行錯誤しながら継続的に改善する	・目的を実現するために、「反復」を用いるなどして、命令を簡潔・明瞭・的確に表現する。 ・意味を拡張し、統合的・発展的に考え、命令を一般化する。

イ 図画工作科で発揮する「プログラミング的思考」の適用方法の捉え

図画工作科における造形的な見方・考え方は、「感性や想像力を働かせ、対象や事象を、形や色などの造形的な視点で捉え、自分のイメージをもちながら意味や価値をつくりだすこと」と整理された。図画工作科の学習活動としてプログラミングを用いて問題を解決する際には、この造形的な見方・考え方を働かせ、「プログラミング的思考」を適用することが考えられる。

例えば、コンピュータを用いて動く模様を作るという表現の活動では、自分が意図した動きのある模様を表現する。そのために、「かいた形を回転させる」「かいた形を移動する」といった命令を組み合わせながら、かいた形が回転しながら移動する「動きのある模様」をつくる活動が考えられる。その際、児童は、どのように改

善すれば自分の意図する表現になるのかを考えながらつくり、試行錯誤してつくりかえることになる。

図画工作科においては、このようなプログラミングを行うことで、児童は、表現しているものを動かしてみるにより、新たな発想や構想を生み出したり、異なる視点からよさや美しさを感じ取ったりすることができる。

以上のことから、図画工作科で制作される「最適化されたプログラム」は、「児童が創造的な学びを実現するために、つくり、つくりかえ、つくる活動を実現するなど、自分が表したい気持ちの意味や価値を表した結果、個別化されたプログラム」という特徴があると考えられる。児童が図画工作科で発揮する「プログラミング的思考」の適用方法の捉えを、表3に示す。

表3 図画工作科での適用方法の捉え

「プログラミング的思考」	図画工作科での適用方法の捉え
必要な動きを分けて考える	・感性や想像力を働かせ、対象や事象について、動きの視点で捉えたことを基に、必要な動きを切り分ける。
動きに対応した命令(記号)にする	・捉えた一つ一つの動きを分け、動きに対応した命令を選択・判断する。
組み合わせる	・自分の夢や願い、経験や体験したこと、伝えたいことなど、児童が表したい、つくりたいと思うことを動きにして、創造的に表現する。
試行錯誤しながら継続的に改善する	・目的を実現するために、動きをつくったり見たりする。動きのよさや美しさなどを感じ取ったり考えたりして、自分の見方や感じ方を深める。

ウ 「各教科等の学習と関連させた具体的な課題を設定する取組」の実施と「プログラミング的思考」の適用方法の捉え

手引によると、「各教科等の学習と関連させた具体的な課題を設定する取組(以下、C分類という。)」は、各教科等の学びを確実にすることをねらいとしないため、小学校プログラミング教育のねらいを達成する取組のみを実施することができると思われる。

算数科及び図画工作科での実施を見据え、今回のC分類の取組では、どのようなプログラミングを体験すれば、育成する「プログラミング的思考」を効果的につなげ、適用方法の違いがより表れやすくなるかを考えた。この取組で想定される「プログラミング的思考」の適用方法の捉えを、表4に示す。

表4 算数科、図画工作科及びC分類での適用方法の捉え

算数科での適用方法	図画工作科での適用方法	今回のC分類での適用方法
<ul style="list-style-type: none"> ・目的を実現するために、「反復」を用いるなどして、命令を簡潔・明瞭・的確に表現する。 ・意味を拡張し、統合的・発展的に考え、命令を一般化する。 	<ul style="list-style-type: none"> ・自分の夢や願い、経験や体験したこと、伝えたいことなど、児童が表したい、つくりたいと思うことを動きにして、創造的に表現する。 ・目的を実現するために、動きをつくったり見たりして、動きのよさや美しさなどを感じ取ったり考えたりし、自分の見方や感じ方を深める。 	<ul style="list-style-type: none"> ・目的を実現するために、「反復」を用いるなどして、命令を簡潔・明瞭・的確に表現することと、各々に現れる様々な条件に応じて、命令を改善することのバランスを考え、目的に近づける。

これらの適用方法の違いを踏まえて、プログラミングを実施する授業を構想したり、児童の表れを捉えたりすることを、教師が「プログラミング的思考」を育成する上での大切な視点の第三として仮定する。

本研究は、「プログラミング的思考」を汎用的な力に高めるためには、①「プログラミング的思考」の観点とその解釈を踏まえること、②プログラムの構造を支える要素である「順次、分岐、反復」に着目し、最適化するという共通点を見いだすこと、③各教科等の特質に応じた解決の仕方によって適用方法の違いが表れることに着目すること、この3点を仮定して取り組み、児童の表れなどを根拠として、これらが大切な視点であることを明らかにするものである。そして、このことは、最適化したプログラム、振り返りでの児童の記述、質問紙調査等に加えて、実践Ⅰと実践Ⅱにより育成された「プログラミング的思考」が、実践Ⅲでどのように発揮されたのかを基に考察する。

(4) 授業実践

(1)～(3)で述べたことを基に、実践ⅠとしてC分類の取組、実践Ⅱとして図画工作科、実践Ⅲとして算数科の授業を行った。

ア 実践Ⅰ C分類の取組の実践

(7) 本時のねらい

社会科の「我が国の食料生産（第5学年）」と関連付け、本時のねらいを「無人トラクターを自動運転するプログラムを制作することを通して、『反復』の要素に着目し、無駄なく水田を耕す動きに近づくように試行錯誤をしながら、与えられた条件を考慮したプログラムに改善することができる。」とする。

(4) 「プログラミング的思考」の具体の想定

「LEGO マインドストーム『EV3』（以下、EV3という。）」を無人トラクターに見立て、長方形に区画された約1㎡の水田模型を効率よく動かすためのプログラミングを行う。児童が主体的に学ぶために、次の手立てを取る。

最初に、水田模型の途中まで動く、間違いを含むサンプルプログラムを提示する。児童は、サンプルプログラムの命令の意味を、実際の機体の動きと関連付けて読み取る。次に、サンプルプログラムの間違いに気付いて修正したり、「反復」の要素を見いだして、プログラムの簡潔性に気付いたりする。しかし、水田模型の枠の形、床面の摩擦、機体のモータの出力の違い等の条件を考慮すると、「反復」を用いて簡略化したプログラムを、更に細かく改良する必要性を感じ、試行錯誤しながら継続的に改善するだろう。

これらのことから、児童に考えさせたい「最適化されたプログラム」は、「水田模型の枠の形、床面の摩擦、機体のモータの出力の違い等の条件を考慮して、目的の動きに近づくようにしたプログラム」となる。

(4) 児童の表れ

「プログラミング的思考」の適用方法の捉えと児童の表れを、表5に示す。

表5 実践Ⅰ 本時の適用方法の捉えと児童の表れ

「プログラミング的思考」	本時における児童の「プログラミング的思考」の適用方法の捉え	児童の表れ（授業記録、ワークシート等）
必要な動きを分けて考える	<ul style="list-style-type: none"> サンプルプログラムや機体の動きを基に、「直進」「回転を指示する条件」「回転」といった必要な動きを捉える。 	<ul style="list-style-type: none"> 実際に機体を手で押しながら、どちらのタイヤが何回転進むことが必要なのかを探り、サンプルプログラムの数値を改良した。
動きに対応した命令（記号）にする	<ul style="list-style-type: none"> サンプルプログラムを基に、捉えた一つ一つの動きに対応した命令の意味を考察し、改良する。 	<ul style="list-style-type: none"> 一つ一つの命令と実際の機体の動きを比較しながら、プログラムの意味を理解した。 機体の向きを変える命令が間違っていることに気付いた。
組み合わせる	<ul style="list-style-type: none"> 命令を「反復」させて表現し、その簡潔性に気付く。 細かな動きの命令を組み合わせ、目的の動きに近づける。 	<ul style="list-style-type: none"> 長い距離を動かすときは、同じプログラムでは「反復」を用いた方が、プログラムが簡単に表せると気付いた。 「反復」を用いることで長い命令の数を減らせると気付いた。（図2・3実線枠内）
試行錯誤しながら継続的に改善する	<ul style="list-style-type: none"> 水田模型の枠の形、床面の摩擦、モータの出力の違い等、様々な条件に着目し、数値を改善して動きを比較する。 「反復」を用いるか細かく数値を改善するかを判断し、目的の動きに近づける。 	<ul style="list-style-type: none"> 改良したプログラムにもかかわらず、実際の機体の動きと期待していた機体の動きにずれがあることに気付いた。 タイヤを2回転から2.1回転にして、繰り返し試したら効率よく動くことに気付いた。 繰り返しブロックで囲む命令の範囲を変え、繰り返しで命令する部分と細かく命令する部分を分け、個々の条件に対応してそれらのバランスを考えた。（図2・3点線枠内）

(I) 分析と考察

児童は、機体が水田模型全体を効率よく動くようにするため、「反復」の要素に着目し、図2のように「反復」の命令を用いて簡略化した。しかし、様々な条件を考慮した結果、図3のように、作業を細かく分けて数値を微調整することになった。児童は、「反復」を用いた命令の簡略化と、条件を考慮した細分化の二つの考えを、個々の条件に応じてバランスよく組み合わせ、最適化を目指した。

イ 実践Ⅱ 図画工作科

題材「気持ちをもうようにして表そう」の実践

(ア) 本題材及び本時のねらい

本題材及び本時のねらいを「運動会の際の自分の気持ちを模様にして表す活動を通して、動きによって自分の気持ちを表すことができることに気づき、自分の表したい気持ちを表すために、模様の動きの表し方を工夫したり、表現の意図

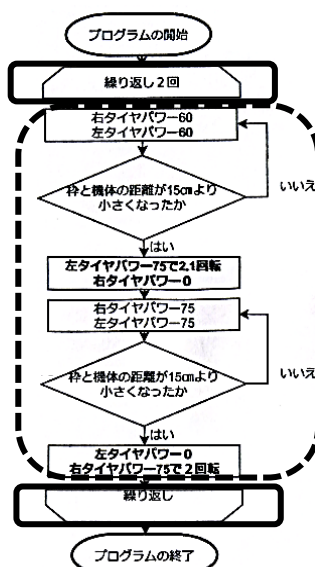


図2 命令の簡略化を重視したプログラム

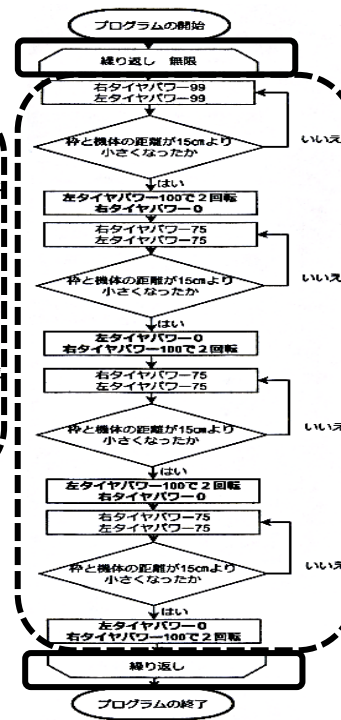


図3 条件を考慮して細かく命令したプログラム

や特徴、表し方の変化などについて自分の見方や感じ方を深めたりすることができる。」とする。

(イ) 「プログラミング的思考」の具体の想定

コンピュータを用いることで、造形的な特徴の中の「動き」に着目した発想や構想を生み出し、自分の表したい気持ちを表すために動きをつくり出すことができる。

そこで、児童が「動き」に着目するための手立てとして、「反復」の命令を用いたサンプルプログラム（図4）を用意する。その「動き」を鑑賞することを通して、「動き」の意図や特徴、表し方の変化などについて、児童が自分なりの見方や感じ方をもつ。それを基に、「動き」を試行錯誤して改善する活動を通して、自分の見方や感じ方を深め、自分の表したい気持ちを表せることに気付く。

これらのことから、児童にとっての「最適化されたプログラム」は「自分の表したい気持ちを表すために、サンプルプログラムの動きを改良したり必要な命令を追加したりした結果、個別化されたプログラム」となる。児童は、模様の動きを改良することで発展性を感じたり、独自性を出したりするなどの様々な視点から、実践Ⅰとは異なる試行錯誤や改善をするだろう。

(ウ) 児童の表れ

教師も児童もサンプルプログラムの鑑賞を通して、形や色、動きといった造形的な特徴に着目し、表し方の工夫を捉えていた。しかし、授業後に教師が改めて「動き」を焦点化して分析したところ、児童の「プログラミング的思考」の適用方法を顕在化することができた。なお、「プログラミング的思考」の適用方法の捉えと児童の表れを、表6に示す。

表6 実践Ⅱ 本時の適用方法の捉えと児童の表れ

オブジェクト(a)(b)…嬉しい気持ち、(c)…不安な気持ち、(d)…頑張る気持ち、(e)…悲しい気持ち

「プログラミング的思考」	本時における児童の「プログラミング的思考」の適用方法の捉え	サンプルプログラムの鑑賞を通して共通する見方や感じ方を深める児童の表れ	
		児童Aの表れ（図5）	児童Bの表れ（図6）
必要な動きを分けて考える	<ul style="list-style-type: none"> サンプルプログラムを読み取り、動きにはいくつかの命令が「反復」されていることに気付く。 移動させたり回転させたり敷き詰めたりするときに必要な動きを捉える。 	<ul style="list-style-type: none"> 「動き」の視点で捉え、「迷子」「波」「ダンス」「手を繋いでいる」「目が疲れる」などの見方や感じ方をつくりだしたオブジェクト(O。。)の動きについて、上・下・斜め等の動きが繰り返されて、回転する動きになっているのではないかと考えた。 	
動きに対応した命令(記号)にする	<ul style="list-style-type: none"> 自分の表したい気持ちを表すために、捉えた一つ一つの動きに対応した命令を改良して「反復」させる。 	<ul style="list-style-type: none"> 波で嬉しさを表すため、矢印から(a)(b)が「反復」して出現するように命令した。波のうねりで気持ちの変化を表すために、(a)(b)の回転の動きを付けた。 	<ul style="list-style-type: none"> 楽しさと悲しさを表すために、「反復」して矢印から(d)(e)が出現する命令にした。 楽しさと悲しさが消える様子を表すために、(d)(e)が下に移動して徐々に消えていく命令をつくった。

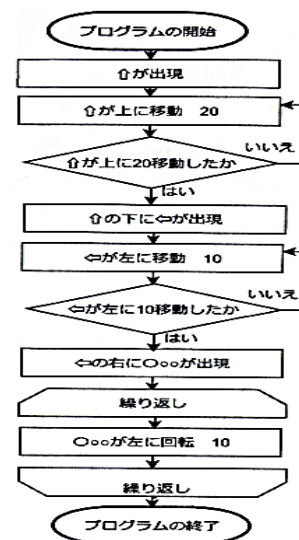


図4 サンプルプログラム

組み合わせる	<ul style="list-style-type: none"> 命令を「反復」させて表現し、その簡潔性に気付く。 自分の表したい気持ちを表すために、「反復」を用いずに別の動きを追加する。 	<ul style="list-style-type: none"> 不安を表すため、その場で回転をしてあまり動かない(c)のプログラムをつくり、嬉しい波を表現した(a)(b)の動きと対比させて表した。(図5点線枠内) 	<ul style="list-style-type: none"> 楽しさと悲しさを交互に表すために、同じ矢印から(d)の命令と(e)の命令それぞれのプログラムをつくらせて組み合わせた。(図6点線枠内)
試行錯誤をしながら継続的に改善する	<ul style="list-style-type: none"> 位置を少しずらしたり、回転を加えたりして動きを試し、自分の表したい気持ちに近づける。 	<ul style="list-style-type: none"> 嬉しさの種類を表すために、(a)(b)の回転の速さを変え、嬉しい気持ちにも違いがあることを表した。(図5実線枠内) 	<ul style="list-style-type: none"> 楽しさが悲しさより多く出現するように、矢印の移動の速さを僅かに変え、(d)が(e)よりも多く出現するようにした。(図6実線枠内)

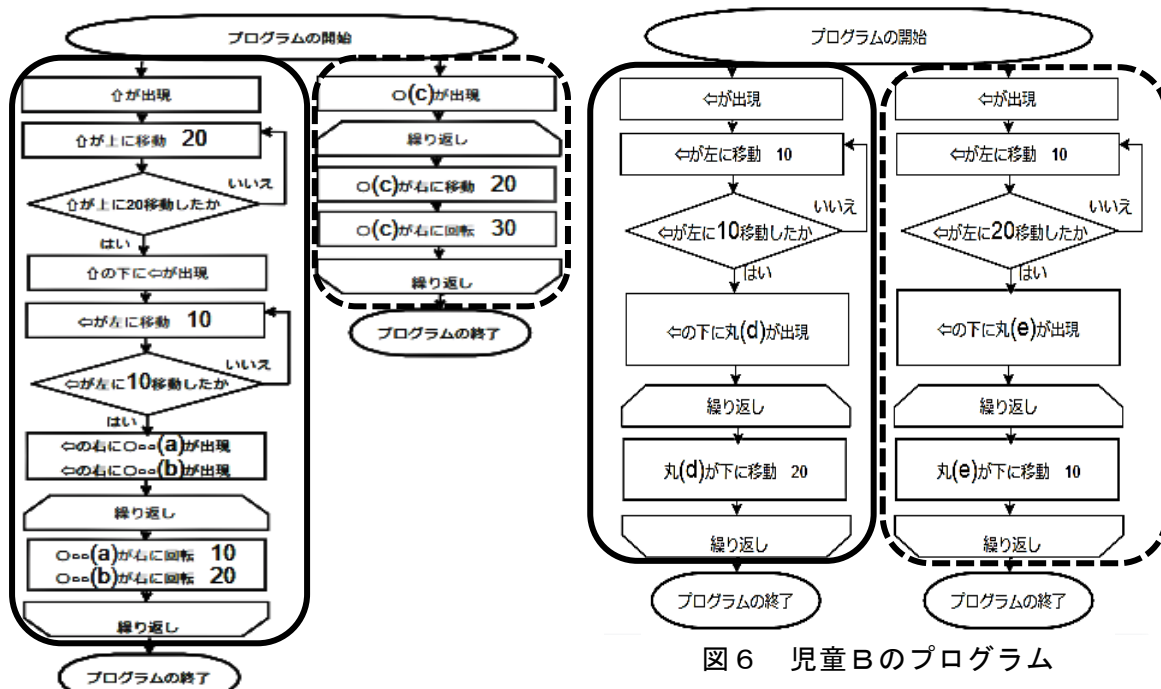


図6 児童Bのプログラム

図5 児童Aのプログラム

(I) 分析と考察

児童は、サンプルプログラムの「動き」を鑑賞することを通して、「動き」に対する自分の見方や感じ方を深め、運動会の際の複雑な心境を表した。具体的には、「嬉しさ」や「楽しさ」といった肯定的な気持ちに対して、速さや方向、間隔、「反復」などの必要な命令を組み合わせた(図5、図6実線枠プログラム)。一方で、肯定的な気持ちの中に隠れる「不安」や「悲しさ」といった否定的な気持ちに対しても、同じ画面上に同時に作動するプログラムを制作し、動きの命令を組み合わせた(図5、図6点線枠プログラム)。このように、児童は、サンプルプログラムの動きを改良したり必要な命令を追加したりして創造的にプログラムを制作し、自分の気持ちの意味や価値を明確に表すために最適化を目指したと言える。

ウ 実践Ⅲ 算数科 単元「正多角形」の実践

(7) 本単元及び本時のねらい

本単元のねらいを「図形を構成する要素に着目し、円と関連させて正多角形の基本的な性質を見だし、その性質や構成の仕方を筋道を立てて考え説明するこ

とができる。」とし、本時のねらいを「辺の長さが全て等しく、角の大きさが全て等しいという正多角形の意味を基に、プログラムを使って正八角形を作図するとともに、更に一部を変えることでいろいろな正多角形を同様に考えることができる。」とする。

(イ) 「プログラミング的思考」の具体の想定

円の性質と関連させた作図に続いて、児童は、正多角形の意味を基に、辺の長さや角の大きさに着目すれば作図ができると考えている。ところが、定規と分度器を使うことでは正確に作図できないことに気付く。児童はこれまでに、一貫して「反復」の要素に着目し、正確に繰り返す作業はコンピュータが得意であることを経験してきたので、コンピュータを用いることによってこの課題を解決できるのではないかと考える。そこで、本単元では実践Ⅰと実践Ⅱのようなサンプルプログラムは使用せず、児童が命令を自ら選択・判断し、最適化を目指して目的の動きをつくることを目指す。

なお、利用するプログラム言語「Scratch」は、基本的には外角で作図をするが、上記のねらいから、内角で命令できるオブジェクトを教師が用意する。

これらのことから、児童に考えさせたい「最適化されたプログラム」は、「正多角形の作図に必要な要素に着目し、『反復』の命令を組み合わせることで簡潔に表現するとともに、更に一部の数値を変えることで、正 n 角形について統合的・発展的に考察して一般化したプログラム」となる。児童は、プログラムの簡潔性の他にも、その一般性、正確性、能率性、発展性等の様々な視点から、実践Ⅰや実践Ⅱのときは異なる試行錯誤や改善をするだろう。

(ウ) 児童の表れ

「プログラミング的思考」の適用方法の捉えと児童の表れを、表7に示す。

表7 実践Ⅲ 本時の適用方法の捉えと児童の表れ

「プログラミング的思考」	本時における児童の「プログラミング的思考」の適用方法の捉え	児童の表れ（授業記録等）
必要な動きを分けて考える	<ul style="list-style-type: none"> 正八角形の意味を基に、正八角形を作図するという一連の動きを「一つの辺の長さ」「一つの角の大きさ」「繰り返しの回数」に切り分ける。 	<ul style="list-style-type: none"> ものさしと分度器を用いて正八角形を作図した経験を基にして、一連の動きを「一つの辺の長さ」「一つの角の大きさ」「8回繰り返す」に切り分けて考えることで作図できると考えた。
動きに対応した命令（記号）にする	<ul style="list-style-type: none"> 正八角形の意味を基に、切り分けた一つ一つの手続きに対応した命令（「○歩動かす」「角度を○にする」「繰り返し○回」等）を選択・判断する。 変換された命令に数値を入力する。 	<ul style="list-style-type: none"> 正八角形の意味と作図をした経験を基にし、幾つかの命令カードの中から、正八角形作図に必要なものとして、切り分けた一つ一つの手続きに対応したカードを、ほとんどの児童が直ちに選択・判断することができた。 辺の長さに着目して「○歩動かす」の命令に20、50、100等の任意の数値を入力したり、角の大きさに着目して「角度を○にする」の命令に前時までの学習を基に135を入力したりした。（図7点線枠内）
組み合わせる	<ul style="list-style-type: none"> 「20歩動かす」「角度を135にする」「20歩動かす」…のように、正八角形の意味を基にして、断片的な命令を順序よく組み合わせる。 	<ul style="list-style-type: none"> 正八角形の意味を基にし、「○歩動かす」「角度を135にする」の順序で命令を組み合わせ、それを8回繰り返そうと試みた。 ところが、プログラムが長くなることに気付き、よりよい方法がないか考え始めた。

試行錯誤しながら継続的に改善する	<ul style="list-style-type: none"> ・正八角形の意味を基に、「反復」の命令を組み合わせて表現し、その簡潔性に気付く。 ・更に一部の数値を変えることで、正n角形について統合的・発展的に考察し、表現する。 	<ul style="list-style-type: none"> ・「反復」の命令の必要性を教師に求め、それを使ってプログラムを簡潔に表すとともに、そのよさに気付くことができた。(図7・8実線枠内) ・正八角形を作画するプログラムを正九角形や正十角形、正百角形にも活用できるのではないかと考え、円と関連させた正多角形の性質を基にして、一つの角の大きさを求め(正百角形の場合…式$360 \div 100, 180 \div 3.6 = 176.4$)、必要な数値を入力して作画することができた。(図8点線枠内)
------------------	---	--

(I) 分析と考察

児童が制作したプログラムは、正多角形の意味を基に、図形を構成する要素に着目し、必要な動きを分けて考え、「反復」の記号を組み合わせて、必要な要素を過不足なく簡潔・明瞭・的確に表現されたものであった。図8では、児童はプログラムの正確性や能率性に気付き、更に一部の数値を変えることで、正n角形について統合的・発展的に考察して一般化し、最適化を目指したと言える。

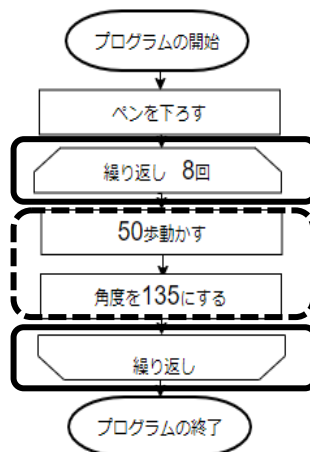


図7 正八角形のプログラム

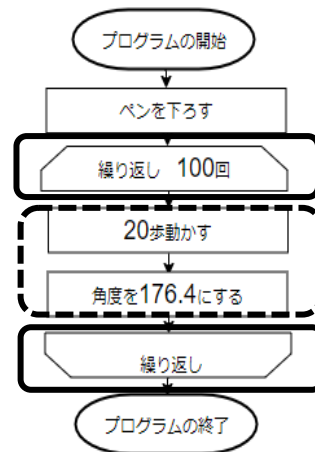


図8 正百角形のプログラム

(5) 考察

ア 最適化されたプログラムの共通点と適用方法の違い

児童が制作した「最適化されたプログラム」のフローチャートを比較し、「プログラミング的思考」を発揮する際の、共通点と適用方法の違いについて考察する。

児童は「必要な動きを分けて考える」「動きに対応した命令にする」などの「プログラミング的思考」の観点に沿って、「反復」といったプログラムの構造を支える要素に着目し、各教科等の特質に応じた見方・考え方を働かせながら最適化を目指していた。このことから、「プログラミング的思考」を発揮する際には、共通点があることが明らかになった。

一方、児童は実践Ⅰで「個々の条件に応じて、命令を簡単にするか細かくするかについて判断し、それをバランスよく組み合わせ最適化した結果、個別化されたプログラム」、実践Ⅱで「創造的に自分の気持ちを表現するために最適化した結果、個別化されたプログラム」、実践Ⅲで「必要な要素を過不足なく簡潔・明瞭・的確に表現し、統合的・発展的に最適化した結果、一般化されたプログラム」を制作したことになる。このことから、「プログラミング的思考」を発揮する際には、適用方法の違いがあることが明らかになった。なお、この適用方法の違いは目的や対象の違いによるものであり、教科等でプログラミングを体験するときには、各教科等の見方・考え方の違いから生まれたと言える。

イ 共通点と適用方法の違いに児童が気付くための振り返り

児童は、共通点と適用方法の違いにどのように気付いているか、実践Ⅰと実践Ⅲを比較させながら振り返ったところ、E V 3と算数科のどちらの動きも「反復」の要素に着目できた。このことから、共通点に気付いたと考えられる。

表8 児童の振り返りからみる、共通点と適用方法の違いへの気付き

教師の質問「なぜ、算数は同じプログラムで表せて、E V 3では違うプログラムになったのか。」 「繰り返しブロックを使ったのに、E V 3と算数のプログラムの長さが違うのはなぜか。」
<ul style="list-style-type: none"> ・ E V 3は、細かいところをやらなければならない。タイヤの回転数や動きにどういう数字を入れるか分からない。算数は、繰り返しが使えて楽。 ・ E V 3は、置いたときに斜めになったり、囲んである場所が違うから、まっすぐに置いたり、まっすぐに進むようにタイヤの回転数を変えたりする。算数は、角度や歩数を揃えれば、いろいろな形をつくれるから楽。 ・ E V 3は複雑。数字や記号の意味やどれくらい進むのか数字では分からないから、微妙にずれる。だから、微調整をしてまっすぐ行くように角度を曲げる。算数は同じ命令で同じように動く。「50歩動かす」「135度回す」それだけでよい。

また、算数科は必要な動きのみを命令にすることで目的の動きをつくることのできるが、E V 3の取組では、条件に応じて細かく命令しなければ目的の動きに近づかないことが分かったことから、適用方法の違いに気付いていたと考えられる。

ウ 質問紙調査からみる児童の変容

「プログラミング的思考」を發揮する際に、児童が実践の前後でどのように変容したかを調査した。なおB校は、私は授業を実施していないが、外部の協力を得て、児童に対してプログラミング体験を2回行っている。

表9 第1回及び第2回質問紙調査での数値の変容（小数第2位を四捨五入した平均値）

	設問「コンピュータを使って目的の動きをつくるため」に続けて ①～④をそれぞれ質問 6段階で選択（肯定6・5・4・3・2・1否定）	A校5年（8人） 授業実践校		B校5年（10人） 授業未実施校	
		1回目	2回目	1回目	2回目
①	始めから終わりまでの動きの順序を見つけることができる。	2.5	4	1.9	3.1
②	始めから終わりまでの動きの順序を見つけて、動きをつくることができる。	2.3	4	2	4.2
③	動きの中にくり返しがある場合を見つけることができる。	2.9	4.5	2.2	4.3
④	動きの中にくり返しがある場合を見つけて、動きをかんたんにすることができる。	1.6	4.5	2	2.8

両校共に、表9の設問の①から④について、肯定的回答の段階が向上した。児童は、目的の動きをつくるために、「順次」「反復」といったプログラムの構造を支える要素に着目するとよいことに気付いていたと考えられる。

さらに、設問の④に着目すると、A校（授業実践校）の方がB校（授業未実施校）よりも伸びていることが分かる。これは、(5)のイで考察したように、本研究において、児童が動きの中にある「反復」の要素に着目し、最適化を目指しながら、動きを改善するために選択・判断をしてきた成果だと考えることができる。

エ 汎用的な力に高まったと考えられる児童の姿

実践Ⅲでは、8人中7人の児童は、12個の命令カードの中から、3個（辺の長さ、角の大きさ、反復）が作図に必要なだと判断して選択し、残り1人も、「反復」の命令を用いれば、簡潔に表せると気付くことができた。このことから、児童は、正八角

形を作図する際、一連の動きを、「一つの辺の長さ」「一つの角の大きさ」「反復の回数」に切り分ければよいと考え、切り分けた一つ一つの手続きに対応した命令を選択した。つまり、「必要な動きを分けて考える」「動きに対応した命令にする」などの「プログラミング的思考」の観点を踏まえ、正八角形の作図に必要な図形の構成要素を見いだして、プログラムへと変換したのである。また、児童はこれまでに、「反復」に着目して最適化する経験をしてきているので、実践Ⅲでも、同様な考え方を働かせたと考えることができる。

つまり、「プログラミング的思考」の観点を踏まえ、「反復」に着目して最適化するという、児童の「プログラミング的思考」は、実践Ⅰと実践Ⅱを通じて育成され、汎用的な力に高まってきていたので、実践Ⅲで生かすことができたと考えられる。

5 研究のまとめ

(1) 研究の成果

- ア 「プログラミング的思考」を育成する上で、次の(ア)から(ウ)を仮定して取り組み、考察した結果、これらが大切な視点であることが明らかになった。
- (ア) 「プログラミング的思考」の観点を整理し、解釈することで、児童の表れをその観点を一貫して捉えること。
- (イ) 「順次、分岐、反復」のプログラムの構造を支える要素に着目し、最適化するという共通点を見いだすこと。
- (ウ) 各教科等の特質に応じた最適化によって、適用方法の違いが表れることに目を向けること。
- イ 僅か三つの限定的な実践であっても、児童は、育まれた「プログラミング的思考」を、学習の中で生かすことができたと考えられることから、大切な視点に留意して、教科等横断的に授業を行うことで、「プログラミング的思考」を汎用的な力に高めることができると考えられる。

(2) 今後の研究課題

「プログラミング的思考」を汎用的な力に更に高めるために、以下の課題が残された。

- ア 他教科等で実践し、更に分析が必要である。
- イ 「プログラミング的思考」を発揮する際の、共通点と適用方法の違いをより確実なものにするための、質問紙の内容や振り返りのもち方を研究する必要がある。

参考文献

- ・文部科学省（2017）『小学校学習指導要領（平成29年告示）』『小学校学習指導要領（平成29年告示）解説 総則編、算数編、図画工作編』『中学校学習指導要領（平成29年告示）解説 技術・家庭編』
- ・文部科学省（2018）『小学校プログラミング教育の手引（第一版）、同（第二版）』
- ・静岡県教育委員会（2019）「令和元年度教育課程研修会」資料
- ・松田稔樹（2014）「共通教科「情報」で汎用的な問題解決力をどう育てるか～「情報的な見方・考え方」を育てるための視点～」東京工業大学
https://edo.repo.nii.ac.jp/?action=repository_uri&item_id=123&file_id=18&file_no=1（2020年2月28日閲覧）