

# プログラミング的思考を育成する授業の在り方（第一年次）

## ープログラミング体験を取り入れた数学的活動を通してー

長期研究員 加藤 政 記

### 《研究の要旨》

平成32年4月から全面実施される小学校学習指導要領において、プログラミング体験を通して、論理的思考力を身に付けるための学習活動を計画的に実施することが示され、プログラミング的思考を育成することがプログラミング教育の目的の一つとなっている。そこで、算数科においてプログラミング体験を効果的に数学的活動に位置付け、プログラミング的思考を育成することを目指し、授業実践を行った。

## I 研究の趣旨

新小学校学習指導要領解説総則編では、プログラミング的思考を「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」としている。このプログラミング的思考には、各教科等で育まれる論理的・創造的な思考力が大きく関係している。教育課程全体での位置付けを考える際は、各教科等で育む思考力を基盤としながらプログラミング的思考が生まれ、その育成により各教科等における思考の論理性も明確になっていくという関係を踏まえなければならない。

算数科においては、「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6月）の中で示されたとおり、算数における学びの本質である数学的な活動として適切に位置付けることが求められている。つまり、プログラミング的思考を働かせた数学的活動によって、数学的な概念や意味の深い理解につながるように授業を構成していくことが必要である。

そこで、数学的活動にプログラミング的思考を育成する活動を位置付けることで、児童は問題解決の手順を導き出し、自らの学びを客観的にとらえることができると考えた。このことがプログラミング的思考の育成につながると考え、本主題を設定した。

## II 研究の概要

### 1 研究仮説

「数と計算」「図形」の領域において、試行錯誤を伴った活動や手順をアルゴリズム化していく活動を効果的に数学的活動に位置付ければ、数学的な概念や意味の深い理解につながり、プログラミング的思考が育まれるであ

ろくと仮説を立てた。その実証には、プログラミング体験が必要と考え、プログラミング的思考の育成につながる力を以下の五つとして研究を進めた。

- 物事を抽象化する力
- 物事を分解して理解する力
- 順序立てて考える力
- 最適な方法かを評価、分析する力
- 方法を置き換えて一般化する力

### 2 研究の内容と実際

#### (1) 授業実践における手だてについて

##### ①【手だて1】プログラミング体験

先述の五つの力を高め、プログラミング的思考を育成するために、コンピュータを使ったプログラミングを体験させた。児童は、平面図形をかくアニメーションの制作過程を通して、プログラミングの考え方にふれることができると考えた。児童に体験させたいプログラミングの考え方を以下の三つとし、算数科において問題解決する際に児童に活用させたいと考えた。

- ・ものごとを手順としてとらえて実行すること。（順次）
- ・状況によって次の行動を変えること。（条件分岐）
- ・目標が達成されるまで同じ手順を続けること。（反復）

##### ②【手だて2】アンプラグドプログラミング教育を取り入れた数学的活動

アンプラグド（コンピュータを使わない）プログラミング教育を数学的活動に取り入れ、プログラミングの考え方を活用して問題解決に取り組みさせた。筆算のアルゴリズム化する過程で論理的思考力を育てる授業を展開することにより、プログラミング的思考も育成されていくと考えた。具体的には、筆算の手順をフローチャートで整理させる。そこには、「順次」「条件分岐」「反復」のプログラミングにつながる手順が含まれるため、プログラミング的思考を育むことができると考えた。

#### (2) 授業実践の実際と考察

実践は以下のとおりであるが、この中から第4学年の

実践を紹介する。

対象児童	第3学年(22名), 第4学年(34名) 第5学年(41名)
授業実践 I	総合的な学習の時間(各学年2時間) 「Scratchでプログラミング体験」 算数科 第3学年「あまりのあるわり算」 第4学年「垂直・平行と四角形」 第5学年「小数のわり算」
授業実践 II	算数科 第4学年算数科「わり算の筆算(2)」 第5学年算数科「多角形と円」

①【手だて1】に基づく授業の実際

授業実践 I 第4学年「垂直・平行と四角形」(10/17時)

Scratch<sup>\*1</sup>を使って平行四辺形の作図に取り組ませた。平行四辺形の内角は児童全員が同じ大きさとし、辺の長さは自分で設定して作図させた(図1)。児童は、スプライトと呼ばれるネコが頂点Bを始点とし移動しながら各辺をかいていくプログラミングに取り組んだ。

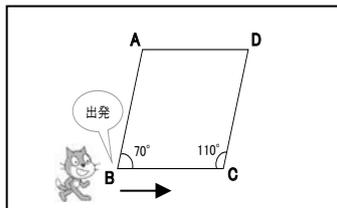


図1 児童が取り組む問題

前時に児童は、三角定規やコンパスで平行四辺形を作図する手順を学習している。そこで、授業の導入で、どんな命令ブロックを使ってプログラムすれば課題の平行四辺形をかくことができるか、各自の予想をノートに書かせた。児童は作図の手順を、命令ブロックによるプログラムの手順に沿って、一つ一つに分解し、ノートにまとめることができた。プログラミング体験により「物事を分解して理解する力」が働いていると考えられる(図2)。その後、

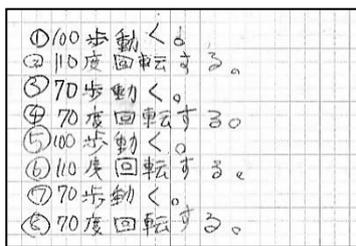


図2 プログラムの予想

予想を基に実際にプログラミングに取り組んだ。すると、同じ組合せの命令ブロックが出てくることに気づき、繰り返しのブロックを活用して、より簡潔にプログラムを作る児童が出てきた。解決方法が複数あることを理解させるために一人一人のプログラムが違うことを確認させた。また、なぜ繰り返しのブロックを使うことができるかを児童に考えさせた。児童は、「対辺の長さや対角の大きさがそれぞれ等しい」という性質から繰り返しのブロックを使うことができることに気付いた(図3)。繰り返しの命令ブロックをどこに入れるかを考えることで、体

験的に図形の性質を理解することができた。作図までの試行錯誤を通して、明らかになった課題を全員で共有し、解決方法を話し合うことで、プログラムをより簡潔なものへ作り上げることができた。これは、「最適な方法かを評価、分析する力」を高めることにもなった。また、内角と外角の関係は小学校で指導する内容ではないが、試行錯誤を通して外角の大きさを入力しなければならないことに気付くこともできた。



図3 児童の作品

プログラミング体験は、プログラムした結果をすぐに試すことができるため、自分の試した結果の評価を基に、自分との対話活動にもつながっていく。自ら問題を発見し、解決する学習過程が連続するため、児童の学習意欲を十分に引き出すことが可能である。児童は、問題解決に向けて主体的に追究するため、教師は先導者ではなく、ファシリテーターとしての役割が求められた。

さらに、児童は、プログラムを完成させると数値を変えるだけで、様々な形の平行四辺形をかくことができることに気付いた。図形を動的に変化させることで、図形についての感覚を豊かにすることができた。

※1 ScratchはMITメディアラボが開発したプログラミング言語である。インターネットで無償配布されており、ブラウザがあれば使用できる。

②【手だて2】に基づく授業の実際

授業実践 II 第4学年「わり算の筆算(2)」(9/13時)

筆算は、計算の手順を可視化し、計算しやすくする方法である。筆算をするときには、その手順の一つずつたどっていく。そこで、単元全体を通してわり算の手順をフローチャートに整理する活動を授業の終わりに5分間設定した。それにより、コンピュータを使わなくとも、教科の中でプログラミングを体験することができる考えた。フローチャートに筆算の手順を整理する活動では、筆算のステップを表す言葉を使いながら、学習した筆算の手順を整理させた。整理する過程で、「条件分岐」や「反復」の考えに触れることができ、「順序立てて考える力」を高めることができると考えた(図4)。

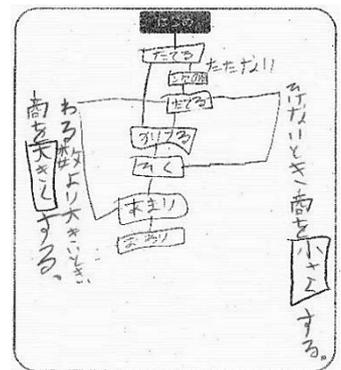


図4 筆算のフローチャート

算数科の学習の面から見ても、「たてる」「かける」「ひく」「お

ろす」という4操作を繰り返して計算を進めていくが、学習が進むにつれて、仮商の修正を行う場合は筆算の手順が変わってくることに気付くことができる。

本時では「わり算の筆算(2)」の単元において毎時間フローチャートにまとめてきた、すべての筆算の手順を整理し、一つにまとめることに取り組んだ。すべてのパターンをまとめることにより、これまでに学んできた筆算の仕方を振り返ることができ、わり算の筆算について理解を深めることができると考えた。

授業の導入では、多く見られたつまづきである過小商の修正をせずに、わり進んでいる誤答を提示した。具体的には、「たてる」「かける」「ひく」「おろす」の4操作の流れを意識しすぎて、「ひく」「おろす」の間に仮商の修正を忘れてしまうミスによるものである。児童に、まず自分で計算し、その結果と比べさせた。どこでまちがったのかを話し合わせると、余りが除数よりも大きいのに、商を修正していないことが原因であると気付いた。そこで、この間違いをしないようにするためにはどうすればよいか考えさせた。すると、前時でまとめた筆算のフローチャートを基に、「余りが除数よりも大きい小さいかを、いつも確認した方が間違えずに計算できる」という結論になった。話し合いを通して明らかにした、筆算でミスをしない方法も含めて、すべての筆算のパターンを各自でまとめさせた。その後、グループでそれぞれのフローチャートを比較・検討し、一つにまとめた(図5)。

グループでの検討により、個人のフローチャートも、より簡潔、明瞭、的確なものへと修正されていった。フローチャートへまとめることに継続的に取り組んだことにより、



図5 グループで一般化する

自己の考えの筋道を客観的にとらえることができるようになったと考えられる。コンピュータを使ったプログラミングでは、自己の思考過程が可視化され、試行錯誤することによってプログラミング的思考が育まれていく。アンプラグドプログラミング教育においても、コンピュータを使ったプログラミングのよさである「思考過程の可視化」「試行錯誤」を数学的活動に位置付けることで、筆算のアルゴリズムを身に付けることができることを確認できた。

### III 研究のまとめ

#### 1 研究の成果

実践後、福島県で実施している「定着確認シート」か

ら授業実践に関わる問題を取り上げ、事後テストを行った。授業実践Iに関わる「平行四辺形を構成することができる辺の組合せを問う問題」の正答率は、県平均を4.6ポイント上回った。また、授業実践IIに関わる「わり算の商の大きさの見積もりについて理解を確かめる問題」では、31.1ポイント上回り、仮商の修正の手順が身に付いていることがうかがえる。数学的活動にプログラミング体験を取り入れることが、数学的な概念や意味の理解に有効であると確認できた。

また、第4学年32名に行った意識調査では、「算数の問題の解き方が分からないときには、あきらめずにいろいろな方法を考えますか」という項目に「あてはまる」と回答した児童が、66.7%から84.4%へ増加した(図6)。問題解決に対する意欲も向上していると思われる。

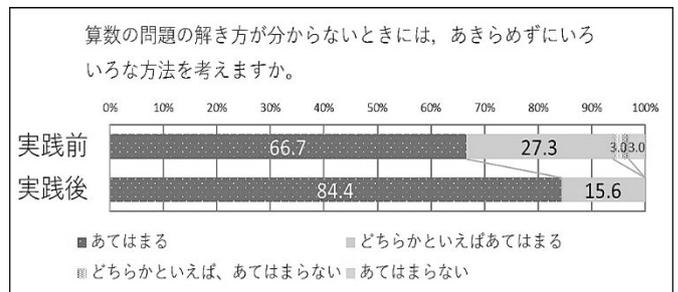


図6 意識調査の容

さらに、児童のコンピュータを使ったプログラムの内容や図4の「筆算のフローチャート」から、「物事を分解して理解する力」や、「順序立てて考える力」「方法を置き換えて一般化する力」が育っていることがうかがえる。これは、プログラミング的思考が育成された姿の一端であると考えられる。コンピュータ使用の有無にかかわらず、算数科においてプログラミング的思考を育成することができた。つまり、「各教科等の中で、問題を解決し、解決方法を言語化できる」ことが、プログラミング的思考の育成につながると考えられる。

#### 2 今後の課題

小学校におけるプログラミング体験のねらいの一つは、教科書等で学ぶ知識及び技能等をより確実に身に付けさせることである。本研究では、プログラミング体験がプログラミング的思考の育成と、数学的な資質・能力の育成の両方に有効であることが確認できた。しかし、算数科でアルゴリズムを習得する単元は数多くあり、発達段階も考えるとフローチャートにまとめることがすべての学年において有効であるとは言い難い。プログラミングの考え方を手だてとして授業のねらいを達成していくことを踏まえ、教育課程全体を見渡し、プログラミング体験に取り組む単元を位置付ける学年や教科等を明らかにしておくことが今後の課題である。