プログラミング教育を取り入れた授業実践
- ICT 機器を使わない指導の提案 -

鈴木はるか*1・坂井敦*2・古屋一希*2・牧野豊*3・小澤理*4・原田篤翼*5・福島健介*6
Email: xkqqb347@yahoo.co.jp

*1：東京都練馬区立石神井台小学校
*2：東京都町田市立小山中央小学校
*3：東京都世田谷区立給田小学校
*4：東京都町田市立忠生小学校
*5：東京都八王子市立城山小学校
*6：帝京大学教育学部

◎Key Words プログラミング教育，授業実践，アンプラグド

1. はじめに

新学習指導要領が公示され、2020 年度から小学校段階での「プログラミング教育」の導入が決定された。小学校プログラミング教育のねらいとして、「プログラミング的思考」の育成が定められている。しかしながら、教育現場ではプログラミング教育実施に向けての不安や誤解が生じていると実感している。

「小学校段階における論理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議」においても、「コーディング（プログラミング言語を用いた記述方法）を覚えることがプログラミング教育の目的であるとの誤解が広がりつつあるのではないかとの指摘もある。」と提言されている。

将来的に、コンピュータを用いることも必要であるが、プログラミング的思考は、日常の授業の中においても育むことのできるものであると考えている。福島(2017)による提言として、教科でプログラミング教育を取り扱うために、「プログラミング教育によって可能なまなび」と教科のもつ特性や単元との親和性を見出し、指導計画を作成することが挙げられている。

本研究の目的は、小学校の学習指導の中で「プログラミング的思考」を育むための指導をどうしたらよいか授業実践を通して検討を行うことである。

2. プログラミング的思考を育むアンプラグド

文科省の「小学校プログラミング教育の手引（第一版）」（以下、手引き）によると、小学校におけるプログラミング教育のねらいは、①「プログラミング的思考」を育むこと、②プログラムの働きやよさ、情報社会がコンピュータ等の情報技術によって支えられていることなどに気付くことができるようにするとともに、コンピュータ等を上手に活用して身近な問題を解決したり、よりよい社会を築いたりしようとする態度を育むこと、④各教科等での学びをより確実なものとすることの４つであると示されている。この中でも、①「プログラミング的思考」を育成することは、小学校におけるプログラミング教育の中核であると述べられている。

この「プログラミング的思考」について安藤*１は、表１のように分類しており、プログラミング教育では、このような能力を育成することが目的であるとしている。

表１　プログラム的思考の内容（安藤 2017）

抽象化	自分が意図する一連の活動を実現
分解	どのような動き組み合わせが必要
アルゴリズム的思考・一般化	どのように組み合わせたらいか
修正	記号の組み合わせをどように改善
評価	より意図した活動に近づくのか

2.1 アンプラグドの意義

福島によると「小学校のプログラミング教育では①アンプラグド、②バーチャルプログラミング、③ロボティクスの 3 種類が実施されている。」としている。①アンプラグドでは、ICT 機器を使用しない。②バーチャルプログラミングでは、コンピュータ上のプログラムを利用する。③ロボティクスでは、ロボットなどの機器をプログラミングによって動かすことを体験しながら学習する。

小学校では「プログラミング教育」とは、コンピュータを使い、コンピュータに命令することによって画面上で、あるいは実物に意図した動きをさせることであると考えている教員が多い。すなわち、福島の分類による②③である。

プログラミング教育を行うにあたり、小学校では、専門性を有する教員が指導する中学、高校とは異なり、学級担任が指導することが多いと考えられる。その際、特別なアプリやソフトを使って学習を行うことには消極的な教員も多い。

表２　学習活動の分類
（小学校プログラミング教育の手引き（第一版））

A	学習指導要領に例示されている単元等で実施するもの
B	学習指導要領に例示されてはいないが、学習指導要領に示される各教科等の内容を指導する中で実施するもの
C	各学校の裁量による実施するもの
D	クラブ活動など、特定の児童を対象として、教育課程内で実施するもの
E	学校を会場とするが、教育課程外のもの
F	学校外でのプログラミング学習機会

また、「手引き」ではプログラミングに関する学習活動を表２のように分類し、A～D について参考事例を挙げている。この中では、A～C が学級での指導になる。このうち「A 学習指導要領に例示されている単元等で実施するもの」については特別なアプリや機器が必要になってくる。また、「C 学校の裁量により実施すること」についても、どんなアプリを使い、どんな指導計画で学習したらいいのか考えなければならず、負担が大きい。

そこで、我々は、プログラミング教育を始めるにあたり、コンピュータを使わない「アンプラグド」の事例を蓄積するべきではないかと考えた。日常の授業を行うようにしながら、教科のねらいに沿いつつプログラミング的思考力を高めていくことで、プログラミング教育に容易に取り組むことができると考えたからである。先ほどの内容に示されている「B 学習指導要領に例示されてはいないが、学習指導要領に示されている各教科等の内容を指導する中で実施するもの」の活動として、プログラミング的な要素が含まれている単元を洗い出し、その指導の中に含まれる形で、プログラミング教育を進めることにした。

3. 算数の四則計算のカード化とアンプラグド

3.1 教科への位置づけ　アルゴリズム的な思考育成

算数の四則計算の筆算は繰り返し練習することで身に付けることができ、多くの児童が単元学習後には問題なくすることができるようになる。しかし、その手順を明確に言語化することができるかといえばそうではなく、なんとなくわかっている、似たような問題をやったことがあるからできる、という状態である。評価のためのワークテストを実施すると、立式や計算より筆算の手順を文章化する問題に児童は苦労することからもそのことが言える。

四則計算の筆算は数多くの処理を行っている。その処理一つ一つを言語化することは、まさにプログラミング的思考で求められる「分解」の活動である。また、分解して言語化した処理一つ一つをどのような順番でどのような条件で行っていくのかを考える活動は、プログラミング的思考で求められる「アルゴリズム的思考」である。

よって、我々は算数の四則計算の筆算の手順を明らかにする活動をプログラミング教育教材として開発した。

２桁の足し算の筆算をフローチャート化すると以下のようになる。なお、フローチャートの図形の使い方は小学校低学年児童でもわかりやすいようにアレンジした。

このようにフローチャート化することで、上から下へ定められた順序で計算を行う「逐次処理」の考え方を身に付けることができる。（図内①）

また、繰り上がりが発生するかしないかはどのような条件で決まってくるのか考えることで、「条件分岐」の考え方を身に付けることができる。（図内②）

さらに、この処理はどこの位の計算なのかを考え、一の位と十の位で行われる処理が同じであることに気付かせることで、３桁や４桁などの大きな数であっても同じことが繰り返されることを明確に言語化できれば、「繰り返し」の考え方を身に付けることができる。（図内③）

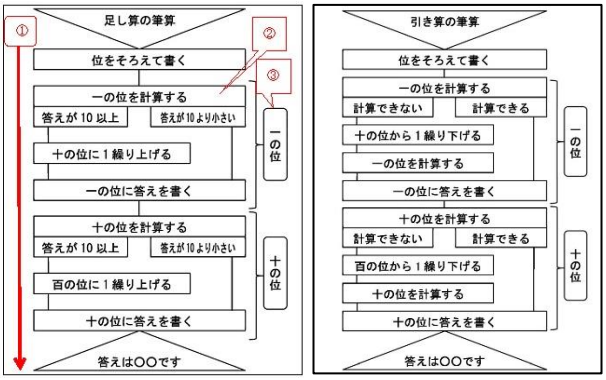

図１ 足し算と引き算のアルゴリズム例

また、引き算でも同じようにフローチャートを作ることで、足し算と引き算の共通性について考えることもできる。

なお、この活動は児童の発達段階・学年、習熟度、確保できる時間数等に応じて難易度を変更することできる。たとえば、難易度を下げるには「アルゴリズム的思考」を育むカードの並び替えに限定すればよく、難易度をあげるには「分解」の思考を育むカード作りからさせたればよい。今回の授業実践は２年生で行ったため、カード作りは省き、カードの並び替えのアルゴリズム的思考を育む活動とした。

4. 授業実践

授業実践は、2018 年 6 月に小学校 2 校、合計 2 学級で行った。

4.1 練馬区立Ｓ小学校

2018 年 6 月 4 日に小学校 2 年生（児童数 33 名）で授業実践を行った。算数の単元「引き算の筆算」の学習を終え、まとめとして時間を設定した。この学級で、アルゴリズムを考える活動は初めてである。そのため、「足し算のアルゴリズムをもとにして、引き算のアルゴリズムを考え、つくる」ことをめあてとした。

まず、足し算と引き算のアルゴリズムをつくるために 25 枚のカードを用意した。（図２）

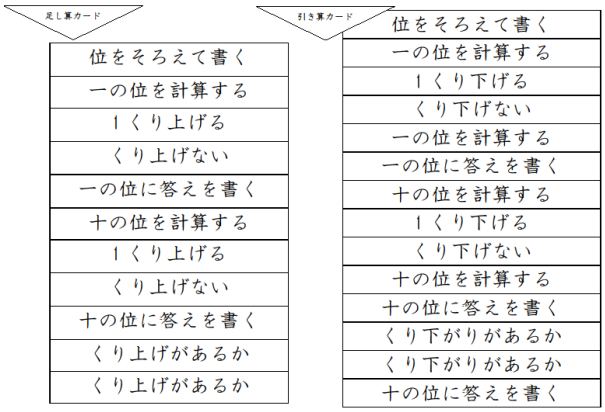

図２ 足し算と引き算のカード

導入では、足し算の筆算のアルゴリズムについてカードを選び、並べながら確認を行った。「位をそろえて書く。」、「一の位の計算をする。」というカードは迷わず選び、並べることができた。しかしその次のカードの選択が３つ

出てきた。①「十の位を計算する。」、②「答えが出る。」、③「一の位に答えを書く。」である。どの選択肢であれ、分岐の存在には全く気付いておらず、手順をとばしてしまうことになった。また、②「答えが出る。」と③「一の位に答えを書く。」のカードの違いがわかっていない様子もうかがえた。ここで、25＋14 と 37＋28 の問題を筆算で解きながら、どのカードがくるのかを確かめた。すると、やっと 2 つの問題の筆算を見て 3 つのカードのどれも当てはまらないことに気付いた。25＋14 には繰り上がりはないが、37＋28 には繰り上がりがあるという違いが筆算を実際に解くことで見つけられたのである。この違いを表すカードは何かと問いかけると、「くり上がりがあるか。」というカードだという意見が出され、そのカードが並べられた。そして、次のカードは躊躇なく「1 くりあがる。」が選ばれ、真下に並べようとした。ここで、分岐の並べ方は始めてだったため、「1 くりあがる。」と「くり上げない」を並列に並べることを教え、カードを置いた。その後は答えまでスムーズにカードを選択し、並べることができた。

次に、引き算のカードを用いたアルゴリズムの作成を行った。2 人組（16 組）に台紙と 25 枚のカードを配り、思考する時間をとった。引き算のアルゴリズムには必要ないカードもあったが、そのカードを選んでしまった組は 1 組もなかった。しかし、引き算の筆算のアルゴリズムを自分たちの力で完璧にすることができたところはなかった。2、3 組は分岐がないまま終わらせてしまい、あとの組は、分岐はあっても「1 くり下げる」のあとに「一の位の計算をする。」を入れず、抜けたアルゴリズム(図 3)になっていた。

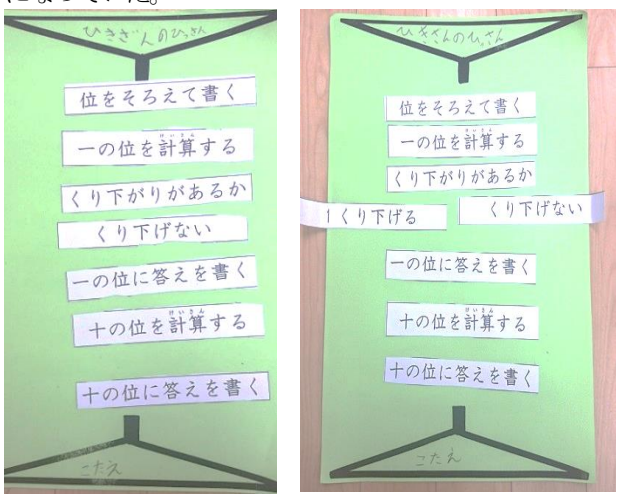

図 3 児童の間違った事例

再度、教師主導で正しいアルゴリズムをつくり、全体で確認した。そして、足し算の時とは違い、引き算では分岐後にも手順があることをおさえた。

最後に、「足し算と引き算の手順の同じところはどこか。」を発問した。すると、2 つのことを児童は発見した。1 つ目は、ほとんど同じカードを使っているということ。つまり、足し算と引き算の筆算の手順はとても似ていること。2 つ目は、1 つのカードだけ疑問になっていること。そして、そのカードで分岐しているということ。

授業の感想には、以下のような意見が挙げられた。「足し算と引き算が似ているのがわかりました。」、「わたしは、先生が言ってくれたように、引き算や足し算の筆算の手順やルールがわかりました。とても面白かったです。」、「引き算の筆算でも足し算の筆算でも内容はほぼ同じということ。」、「足し算と引き算のくり上げとくり下げ以外がまったく同じことを書いているのをはじめて知りました。」

4.2　町田市立 O 小学校

2018 年 6 月 6 日に小学生 2 年生（児童数 32 名）で授業実践を行った。

算数の単元「たし算のひっ算」学習を終えたところであり、そのまとめの時間として設定した。

この単元では、たし算の筆算ができる『たし算ロボット』をプログラミングすることを一貫して行っていた。プログラミング教育では、基本的にコンピュータを使うことを前提としており、その点を意識し、今回はロボットという題材を設定した。授業の中では、「たし算ロボットが誤答したものを修正するためのプログラミングをする」という活動を毎回行っていたため、たし算の筆算の学習をするのと同時に、人間とコンピュータの思考の違いについて学習することができた。また、たし算の筆算のやり方を既に知っている児童にとっても、プログラムの修正は未知であり、学習への必要感をもって取り組むことができた。

前述のことから、児童はまとめの学習の段階では、ロボットのプログラムを通して、たし算の筆算がどのような手順で行われているかをある程度把握していた。ただし作り終わったロボットのプログラムをすべて通ると、24＋13 のような繰り上がりの必要がない計算においても、繰り上がりが行われるようなプログラムになっていた。

そのため本時では、繰り上がりの無い計算が繰り上がりしてしまうことに気付かせ、「どのようなたし算でも対応できるようなプログラムに修正する」というめあてを設定し、授業をおこなった。

まずたし算ロボットが 48＋87 では間違えないが、24＋13 では間違えることを確認した。最初に 48＋87 の計算で用いるプログラムをカードにし、児童に配った。カードはその計算で用いるもののみを配り、並び替えるだけで完成するようにした。活動はペアで行った。

全てのペアが、「位をそろえて書く」「一の位を計算する」「答えは○○です」の 3 枚は正しい場所に置くことができた。「一の位に答えを書く」「十の位に答えを書く」の 2 枚はどこに入れていいか迷い、多くのペアが間違えていたが、計算後に繰り上がりを行うなど、基本的な計算の手順は間違えずに並べることができた。

カードを並び替える際には、多くのペアが最初に「位をそろえて書く」を一番上に、「答えは○○です」を一番下に配置し、他のカードがその間で並び替えられていた。(図 4) のペアは「十の位で 1 くり上げる」の位置が分からず、相談する様子が見られた。最終的に並べ終わった段階では、(図 5) のようになり、繰り上がりのカードを正しく並べることができた。カード化し、計算の手順を細分化することで、児童がどこの手順が曖昧であるかが、児童の思考過程を確認することができた。

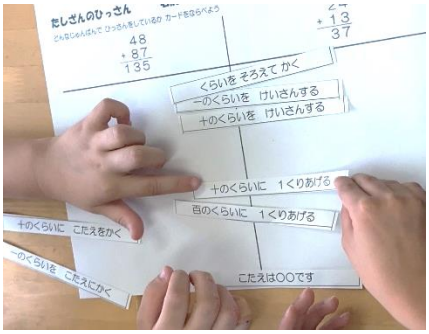

図4 児童の思考過程

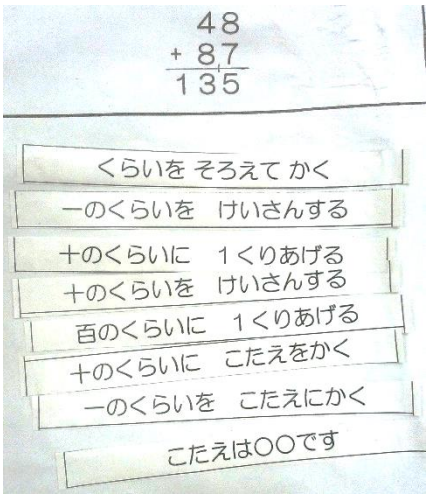

図5 児童の思考結果

授業では、ペアで作ったプログラムを発表し、教員がロボット代わりに計算し、自分たちのプログラムのどこの手順が間違っているかを確認しながら正しいプログラムに修正した。

次に 24＋13 の計算も、先ほどのカードを同様に配って考えるようにした。ほとんどのペアが、「繰り上げる」というカードが必要ないことに気付き、正しいプログラムを組むことができた。この活動では、並べ終わったプログラムを筆算にあてはめ、正しい手順になっているか確認するペアも見られた。

最後に 2 つのプログラムを比べ、どの点が異なるかを考えた。児童の考えを聞くと、繰り上がる場合と繰り上がらない場合で分かれるということは理解できていた。ただし、「どういう場合でくり上がり、どういう場合でくり上がらないか」という分岐の条件に関しては、教師が誘導しないと出なかった。

5. 授業実践の考察

まず、カード化することによって、教師は、筆算のアルゴリズムを作成する過程を見られ、児童の思考過程を知ることができた。また、児童自身は、筆算のアルゴリズムを作ることで再度、自分の行っている計算を振り返る機会となった。

そして、この児童の思考過程の観察から、筆算の手順が分解しきれていないということがわかった。授業実践でも述べたが、カードが抜けてしまい、正確なアルゴリズムとならなかった。特に分岐の部分でその様子が見られた。S小学校の事例で言えば、「くりさがりがあるか」、「1 くり下げる」というカードのあとに「一の位の計算をする。」というカードが入らなかったというところが挙げられる。児童の思考過程では、この手順がはっきりとしておらず、計算するときには実行しているはずの手順だが、言語化されなかった。

以上のことから小学校 2 年生では、正確なアルゴリズムを作成することが難しいことがわかった。ここから、自分の行動をメタにとらえる視点が足りないということがうかがえる。ただ 2 つの授業実践のどちらでも、1 回目の活動に比べて、2 回目の活動ではアルゴリズムをスムーズに作成することができた。つまり 2 年生の段階でも、ある程度のメタ認知をすることが可能であることも分かった。町田市立O小学校の実践では、ロボットを題材にしたことで、自分が理解するだけでなく、相手に理解させなくてはならない状況を作り出した。低学年では、客観的に自分をとらえることは難しいが、このような方法をとることでメタ認知を促進させることができることが分かった。また児童は、たし算の筆算を単純な技能の習得だけにならず、理解することへの必要感をもって授業に臨むこともできた。コンピュータを実際に使わないアンプラグドの実践でも、コンピュータを意識させることが重要であることが分かった。

今後、アルゴリズムを自分の力で作成する力が必要であり、このようなカードを使ったアルゴリズムの作成は有効だと考える。さらには、四則計算での発展も期待できるであろう。

6. おわりに

我々は、今回、算数の四則計算の筆算のフローチャートに注目し、プログラミング教育を行おうとした。6 月の 2 年生の授業実践を踏まえ、他学年での授業実践を行っていく予定である。その際、授業実践の考察をもとにもう一度授業展開を考えていく。

参考文献

(1) 文部科学省：“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）”,(2016.6) http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm,2018.6.15 閲覧
(2) 坂巻若菜・福島健介：“授業実践から考える小学校におけるプログラミング教育の課題・方向性”「2017 PC Conference」,151p-154p
(3) 文科省：“小学校プログラミング教育の手引（第一版）」”,(2018.3)
(4) 安藤明伸：“情報教育対応教員研修全国セミナー資料『どうする？授業で取り組む「プログラミング」』”,（2017）
(5) プログラミング教育開発研究会：“小学校プログラミング教育指導案集　教員による授業実践例 16 本”，6 ページ，株式会社ワイズインテグレーション（2018）.
(6) 黒上晴夫,堀田龍也：“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”，11～15 ページ，小学館（2017）.
(7) 小林祐紀，兼宗進：“コンピューターを使わない小学校プログラミング教育”，2～11 ページ，株式会社翔泳社（2017）.