

教科等での学びをより確実なものにするための プログラミング教育に関する研究

— 小学校算数科における Scratch の活用を通して —

2020/3

四日市市教育委員会 教育支援課

はじめに

本市では、新学習指導要領の本格実施を間近に控え、子どもたちが自らの人生を拓き、生き抜く力を持つことができるよう「四日市市新教育プログラム」を構想しています。その柱の1つである「論理的な思考で道筋くっきりプログラム」には、プログラミング体験等を通してプログラミング的思考を育むなど、これからの時代に求められる論理的思考力を育成する内容が盛り込まれています。

また、「問題解決能力向上のための授業づくりガイドブック2」では、解決して得られたことや解決方法の過程を振り返り、その正しさを確認したり、次の問題に活用したりする活動を含んだ第5プロセスを授業づくりの重要なポイントとしています。

一方、不登校の課題に対しては、改修整備及びセラピストの配置体制の充実が図られた「登校サポートセンター」を核とした取組を進めています。

そうした本市の現況を鑑み、本年度は、3つの課題研究に取り組みました。1つ目は、乗法の筆算の学習において、Scratch を用いてプログラミング的思考を意識した授業を構成することが、論理的思考力及び計算の技能をより高めることに有効であるかを検証しました。2つ目は、算数科の授業で、学習内容と解決過程について記述させる振り返り活動が、児童の学習意欲に及ぼす効果について明らかにする研究を行いました。また、3つ目は、登校サポートセンターの「がんばりチェックシート」を学校での別室運営に取り入れることで、生徒に合った支援の手だてを考えるとともに、それを教職員間で共有することで、より見通しをもって支援にあたれるような校内体制づくりにつながるかを検証する調査・研究を進めてきました。

その成果を調査研究報告書として、ここにまとめました。これらの研究成果が、学校・園の日々の教育実践に活用されることを期待します。

末尾になりましたが、本課の研究調査を進めるにあたって、御指導・御助言いただいた国立教育政策研究所初等中等教育研究部総括研究官の山森 光陽様をはじめ、研究協力員並びに調査・実践面で御協力いただきました学校等の関係者の皆様に心から感謝の意を表します。

令和2年3月

四日市市教育委員会教育支援課
参事兼課長 中村 隆志

— 目 次 —

1 問題	1
2 目的	9
3 方法	9
4 結果	18
5 考察	23
[引用文献]	27
[資料]	28

教科等での学びをより確実なものにするためのプログラミング教育に関する研究 —小学校算数科における Scratch の活用を通して—

1 問題

1.1 小学校学習指導要領（平成 29 年告示）におけるプログラミング教育

1.1.1 プログラミング教育のねらい

文部科学省（2018）は、「コンピュータを理解し上手に活用していく力を身に付けることは、あらゆる活動においてコンピュータ等を活用することが求められるこれからの社会を生きていく子供たちにとって、将来どのような職業に就くとしても、極めて重要」であるとして、2020 年度から小学校においてもプログラミング教育を導入することとなった。

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、③各教科等の内容を指導する中で実施する場合には、各教科等での学びをより確実なものとする」との3つである。小学校段階でプログラミングを体験させるねらいは「プログラミングの技能を習得することではなく、体験を通して論理的思考力を身に付けさせることである」と小学校プログラミング教育の手引に明記されている。

1.1.2 プログラミング的思考とは

プログラミング的思考について、文部科学省（2018）は「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」と定義しており、図で表すと Figure1 になる。「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、論理的に考えることが大切」とあり、その繰り返しが論理的思考力の向上へとつながるといえる。

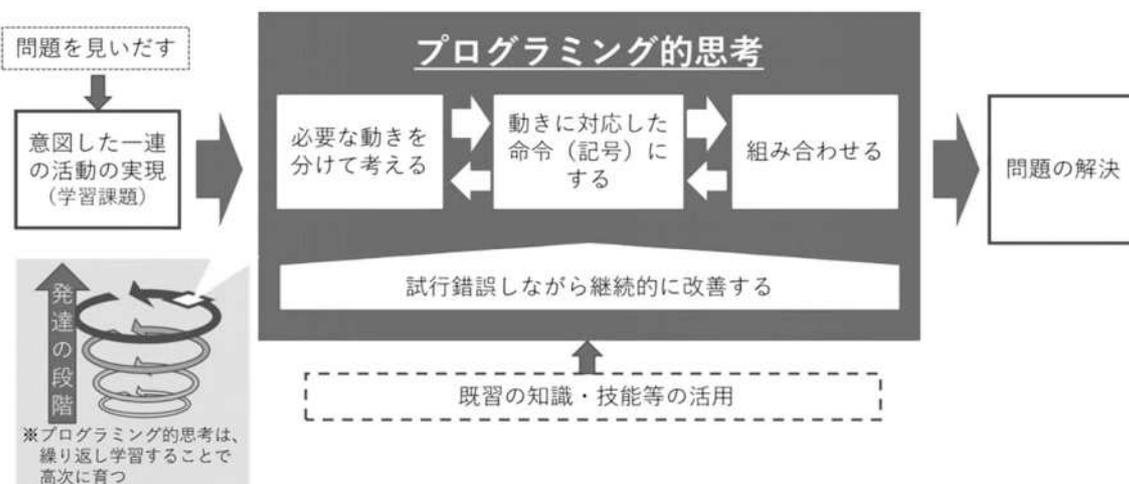


Figure 1 プログラミング的思考（文部科学省 2018）

石嶋 (2018) はプログラミング的思考とは、『『こうしたい』という結果を実現するために、『何を、どのような順番で組み合わせればいいのか』を考える力』であり、Figure1 のようなプロセスで学習することにより、「物事を論理的に考えたり、目的を達成するための手順を明確に描いたり、知識や情報を整理したりする力が身に付く」と述べている。

また、赤堀 (2018) はプログラミング的思考とは教科を横断する論理的な能力だとしている。赤堀は大学生を対象にして、プログラム作成に対応する問題として流れ図を作成する問題とプログラム設計に対応する問題として切符の自動販売機の画面の設計図の問題を取り上げ、他教科との相関を調べた。その結果、流れ図の問題は、数学・代数や理科・天体の問題との相関があり、設計図の問題は国語・読解や社会・考察の問題と相関があったことから、プログラム作成とプログラム設計は、共通性があるが、異なった論理的思考を必要としており、「プログラミング的思考は、他教科・領域の論理的思考と関連するような総合的な論理的思考」としている。

高橋・三井 (2019) は「思考力」「論理的思考力」「プログラミング的思考」の関係について、「論理的思考力は、思考力の一部を構成し、プログラミング的思考より大きく、その一部を取り込む概念である」としている。また、プログラミング的思考について、算数と理科に限らず「他の教科等とも関係していると考えられる」と述べている。

これらから、プログラミング的思考は、自分が求める結果を得るために、どのようなこと（命令）をどのような順番で組み合わせたらよいかを修正や改善を行いながら考える力のことだといえる。また、教科を横断する総合的な論理的思考でもあるため、様々な教科や日常生活の学習場面でプログラミング的思考を意識して指導することが、これからの予測困難な社会を生きていく子どもたちに必要な力を育むことにつながると考える。

1.1.3 プログラミング的思考で育む論理的思考力

米田 (2016) は論理的思考力（ロジカルシンキング）とは、「物事を、道筋を立ててきちんと考える力」であり、「プログラミングの基礎を学ぶことによって、論理的思考力（ロジカルシンキング）の強化や問題解決能力の向上、基本的な IT リテラシーの習得といった、多くの面で良い結果が得られる」としている。また、「全社会人に必須のビジネス基礎力の1つである論理的思考力（ロジカルシンキング）は、一般的には、頭の中で考えていることを紙に書いたり、相手の立場になって物事を考えたりすることによって鍛えられる」と述べており、「この力はプログラミングを学習することで必然的に身につく」としている。その理由として、「プログラミングとは、実作業においては『小さい命令をいくつも組み合わせ、全体で何らかの目的を達成する作業』であり、『どのように指示を出すのか』『どのように表現すればよいのか』を考えながらアイデアを形にしていくことが求められるため、プログラミングを学習する過程で論理的思考力や表現力が高まる」と述べている。

波頭 (2019) は「思考」とは情報と知識を照らし合わせたり繋ぎ合わせたりして何らかの意味合いを得るプロセスであり、ある論理が「論理的である」と認められるためには、その論理展開（推論、あるいは理由づけの思考の道筋）が客観的妥当性（大多数の人によって正しいと判断・承認され得る

こと)を有していて、論理展開の文脈が受け手に理解され、納得される必要があるとしている。これらを総合して「論理的思考」とは、「情報と知識を組み合わせて、客観的妥当性を有する思考の道筋によって、既呈命題から次段階の命題を導く、あるいは結論を根拠によって支える形態の命題構造を作り出すこと」と述べている。また、論理的思考によって正しい結論及び思考成果を得るための主要なスキルとして以下の3つをあげている。

- ① 思考している意味内容を的確に言葉で表すための「適切な言語化」スキル
- ② 事象を構成要素に「分ける」スキルとある事象と別の事象を意味やイメージの連関性によって「繋げる」スキル（「分ける」「繋げる」は論理的思考における情報処理の中核を成すスキル）
- ③ 論理展開に現実的有効性を与えるための「定量的な判断」スキル（「確率」と「統計」によって担保される）

これらの中で、プログラミング的思考で育むことができる論理的思考力は「分ける」「繋げる」及び「適切な言語化」スキルだと考える。プログラミング的思考の定義にあるように、一連の活動を一つ一つの動きに対応した記号まで分けて、それらをどのような順序に繋げるかを修正・改善を繰り返して学習を進めるため、「分ける」「繋げる」スキルを養うことができると考える。また、学習の過程を説明・表現する際に文章で表したり、会話の中で相手に伝えたりする場面が想定されるが、その際にどのように表現すれば相手により伝わるのかを考えることになる。この言葉での表現を意識させることで「適切な言語化」スキルについても育むことができると考える。

総務省（2015）は教育事業者、学識経験者、民間企業の実態を把握し、その結果に基づき分析・整理を行った。プログラミングに関する教育がもたらす効果の中で、特に課題解決能力（論理的思考力、計画性等）の向上については、「フローチャートや仕様書の作成とそれらに基づきプログラミングするという過程で、俯瞰的に考えたり、時系列で考えたり、しくみを考えるなどの合理的、論理的思考が必要となるため、論理的思考力や課題解決能力が向上するとの意見が得られた」と示している。

しかしその一方で、「簡単な論理を必要とする教育は、9才以前に開始することも考えられる」としつつも、「現時点では、効果に関する評価モデルは十分には得られていないため、本格的な論理構成を必要とする教育は、9才頃以降とし、今後、教育の実践と評価を通じて開始時期を調整する必要がある」とも示している。さらに、NPO法人CANVAS理事長の石戸はプログラミング的思考によって育まれる論理的思考力について、「論理的な思考力が育ったかどうかを客観的に評価するのは難しい」という見解を述べている（滝，2016）。このように論理的思考力を身に付けさせる時期やその評価についてはまだ検証事例が少ないことが分かる。

1.2 日本のプログラミング教育の遷移

学習指導要領で初めてプログラミングに関する内容が記述されたのは1989年である。1989年以降に公示された学習指導要領に記載されているプログラミングについての項目の変化をTable1にまとめた（松村・山脇・小野・大森，2018）。

Table1 1989年以降の学習指導要領におけるプログラミング教育の変遷（松村他，2018を編集）

学校	1989年	1998年	2008年	2020年以降
高等学校	数学（選択）	数学（選択） 情報（選択）	情報（選択）	情報Ⅰ（必修） 情報Ⅱ（選択）
中学校	技術・家庭（選択）	技術・家庭（選択）	技術・家庭（必修）	技術・家庭（必修）
小学校	なし	なし	なし	各科目での実践（必修）

1989年の中学校の技術・家庭科で学習する内容は、地域や学校の実態及び生徒の特性等に応じて11の領域のうちから7以上の領域を履修することになっており、プログラミングを学ぶ「情報基礎」は選択領域だった。当時の中学校技術・家庭科で実施されていたのは、プログラムによって絵を描く授業や、技術分野の他領域と関連する「ものづくり」を意識した計測・制御など、幅広いもので、教育環境として用いられたプログラミング言語はBASICやLOGOなどであった。

1998年から中学校の技術・家庭科においては「情報とコンピュータ」が必修となり、全ての中学生が情報について学ぶことになったが、科目内での学習項目の中でプログラミングは選択項目であったため、一部の生徒が学ぶという状況はこれまでと大差はなかった。プログラミング環境も、Microsoft Visual Basicなどが使われるようになったが、教育用に開発されたものではないため、生徒が使ううえで問題があった。

2008年以降の中学校の技術・家庭科では、プログラムによる計測・制御がねらいであることから、LEGO マインドストームなどの教材を導入した授業が行われている。プログラミング言語はScratchベースのブロックプログラミング言語、フローチャート型プログラミング言語、C言語などのテキスト型言語など、多様なプログラミング言語が使われている。

小学校学習指導要領（平成29年告示）では、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の例として、第5学年算数科（B図形）、第6学年理科（A物質・エネルギー）、総合的な学習の時間で取り扱うことが明記されている。

算数 第3 指導計画の作成と内容の取扱い

2 第2の内容の取扱いについては、次の事項に配慮するものとする。

(2) 数量や図形についての感覚を豊かにしたり、表やグラフを用いて表現する力を高めたりするため、必要な場面においてコンピュータなどを適切に活用すること。また、第1章総則の第3の1の(3)のイに掲げるプログラミングを体験しながら論理的思考力を身に付けるための学習活動を行う場合には、児童の負担に配慮しつつ、例えば第2の各学年の内容の〔第5学年〕の「B図形」の(1)における正多角形の作図を行う学習に関連して、正確な繰り返し作業を行う必要があり、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場面などで取り扱うこと。

理科 第3 指導計画の作成と内容の取扱い

2 第2の内容の取扱いについては、次の事項に配慮するものとする。

(2) 観察、実験などの指導に当たっては、指導内容に応じてコンピュータや情報通信ネットワークなどを適切に活用できるようにすること。また、第1章総則の第3の1の(3)のイに掲げるプログラミングを体験しながら論理的思考力を身に付けるための学習活動を行う場合には、児童の負担に配慮しつつ、例えば第2の各学年の内容の〔第6学年〕の「A物質・エネルギー」の(4)における電気の性質や働きを利用した道具があることを捉える学習など、与えた条件に応じて動作していることを考察し、更に条件を変えることにより、動作が変化することについて考える場面で取り扱うものとする。

総合的な学習の時間 第3 指導計画の作成と内容の取扱い

2 第2の内容の取扱いについては、次の事項に配慮するものとする。

(9) 情報に関する学習を行う際には、探究的な学習に取り組むことを通して、情報を収集・整理・発信したり、情報が日常生活や社会に与える影響を考えたりするなどの学習活動が行われるようにすること。第1章総則の第3の1の(3)のイに掲げるプログラミングを体験しながら論理的思考力を身に付けるための学習活動を行う場合には、プログラミングを体験することが、探究的な学習の過程に適切に位置付くようにすること。

この他にも小学校での外国語・外国語活動の充実などの大改訂が行われ、しかも、授業時数は質的な内容の削減が行われていないため、2002年度から完全実施された週休2日になる前の水準まで授業時数が増加することになる（松村他，2018）。

1.3 プログラミング教育の効果に関する先行研究

1.3.1 コンピュータを教えるプログラミング教育

森本（1989）はプログラミング言語としてLOGO Writerを使用し、小学校4年134人と小学校6年129人を対象に正方形・正三角形・正六角形を描写する課題に取り組ませている。どちらの学年も完成率が高かったのは正方形で4年は77.8%、6年は95.3%だった。また、3つの図形のうち2つ以上の図形を完成させた子どもの割合では4年の約70%に対して6年は90%近くとなった。この差について4年は角度の学習をしていないため当然の結果としつつ、その反対に角度の概念を学習してなくても教材を工夫することで4年以下の子どもにもLOGOを通して角度の概念を学習できる可能性がある」と述べている。

その中で森本（1989）は「この研究では、『コンピュータ活用能力』を、コンピュータについての操作とプログラミングの能力であると考え」と記しているように、当時は情報化社会に適応するためにコンピュータ操作能力が必要だからコンピュータを教えなくてはならないと主張されていた（市川，1994）。BASICはもちろんLOGOも教育用に開発されたプログラミング言語とはいえ、コンピュータ操作とプログラミング技能を教える必要があり、その習得に重きが置かれていたことになる。ところが、ソフトウェアが充実してくるにつれコンピュータをめぐる環境は短期間で変化するため、特定のプログラム言語や特定のソフトの使い方を教えても子どもが大きくなったころには役に立たないと

ということが意識されはじめ、当初のプログラミング必要論は声が小さくなっていった(市川, 1994)。

1.3.2 コンピュータで教えるプログラミング教育

一方で、コンピュータプログラミングを思考や表現のための道具として活用する可能性を述べた研究もあった。

杉野(1988)はより標準カリキュラムに近づけるため学校図形 Logo という自作ソフトを作成し、小中学校の図形領域・関数のグラフの場面での教材開発を行っている。その中で、論理的に考えプログラミングする、あるいは見通しの立ったところまでプログラミングすると発見があり、それによって考えを修正したり、考えをまとめたりしながら、能動的、問題解決的に学習を進めることができるとしている。

宮田・大隅・林(1997)はプログラミングを学習してもプログラミング以外の場面での問題解決能力は伸長しなかったという先行研究から、プログラムの作成を文法や命令の暗記を中心に指導するアプローチと、計画・分析・実行・評価といった問題解決の過程を重視するアプローチの2つによる LOGO のプログラミング学習を中学校3年生に実施し、問題解決能力の伸長の調査研究を行った。その結果、問題解決のプロセスを重視したアプローチでプログラミングを指導した場合、問題解決能力の転移が起りやすいことが限定された状況で実証できたとしている。

このように、指導者がプログラミング技術を育てることではなく、論理的に考える・問題解決能力を育てることを目的としてプログラミングを取り入れた授業を構成することによりそれらの力を育てることができるかと報告されている。

1.3.3 ビジュアルプログラミング言語の普及による授業への活用

プログラミング言語も時代とともにビジュアルプログラミング言語のように複雑な文法や高度な数学の知識を覚える必要がないものなどが開発され、それにより教育に用いられるものも変わってきている。

山本・田賀・新屋・小林(2006)は小学校6年の理科における電磁石と電流(電流のはたらき)の単元で LEGO マインドストームを用いた授業を11単位時間行っている。授業を受けた児童は学習課題に対して興味・関心も高く、全ての児童が最後までプログラムを作成することができ、プログラムに関する興味・関心も有意に増加したとしている。

和田(2019)は小学校全学年に対して Viscuit (ビスケット) を利用したプログラミング教育教材を作成し、各学級4~9単位時間の授業を行い、5, 6年に批判的思考態度を測定するアンケートを実施し、「何でもよく考えないと気がすまない」「物事をよく考える方だ」「何かを決めるときは時間をかけて慎重に考える方だ」等の項目において、実践後にそのポイントが上がっていたと報告している。

このように、小学校でも活用しやすいプログラミング言語が開発され、それに伴い授業で活用される機会も増えてきている。

さらに、現在の教育現場で活用されているのが Scratch である。森・杉澤・張・前迫(2011)は小

学校4年の総合的な学習の時間でScratchを用いて図形を描いたり、個人・グループによる作品作りをしたりするなど、計26単位時間実施する研究を行った。「スプライト」と呼ばれるキャラクターや背景に、動きや制御、繰り返しの命令をさせた作品を作ることができ、条件分岐やキー入力の判別処理といった複雑なプログラミングを含めた作品作りにも8割を超える児童が取り組むことができた」と述べている。

山本・鈴木・岳野・鹿野(2017)は小学校4年の総合的な学習の時間でScratchを応用したアプリであるTickleでSphero(教育用の球型ロボット)を用いて障害物回避の授業を4単位時間実施した。その結果、対象児はプログラミング学習に関する基本的な知識を習得するとともに、それらに関する興味・関心、意欲の認識が向上したことが示唆された。また、学習の難度について26件中、プログラムを難しいと感じただけで終わるのではなく、課題を乗り越えて利点を見いだす回答が19件あった。

このように、2020年度のプログラミング教育の必修化に向けて先進的な研究がなされている。

1.3.4 教科内容の習得に関わる効果

教科学習の中にプログラミング的思考を意識した授業を構成することで教科内容の習得にどのような効果があるかについての研究も出てきている。

菅野(2017)は小学校4年の算数科における「広さを調べよう」の単元で面積の単位の相互関係の理解・定着を図るためにScratchを用いる群と統制群を比較する研究を行った。 $1\text{ m}^2=\square\text{ a}$ の穴埋め問題での正答率比較では、検証授業終了直後は統制群の方が若干高かったものの、1ヶ月後の同じ問題調査では85.7%に対して統制群72.2%と差が出た。また、検証1ヶ月後の $1\text{ a}=100\text{ m}^2$ であることを説明する記述問題では57%に対して統制群22%とさらに差が出たことから、Scratchを活用することで定着度が上がったと述べている。

黒羽・伊藤・川澄・小林(2019)は、学習内容(角度の測り方、角、三角形の書き方等)が順序性を学ぶことに適していること、作図を行う学習活動がありプログラムを扱うことに適していると考え、小学校4年の算数科「角度」の単元においてフローチャート及びScratchを取り扱った。学習内容の理解の程度を把握する評価テストについておおむね満足できるB評価以上は19名(79.2%)であり、平均点を同学校の他学級の結果と比較したところ値の差が小さかったため、学習内容が他学級と同程度習得できたとしている。

斎藤(2019)は、小学校5年の算数科「小数のわり算」の授業において、Scratchを用いて条件分岐のクイズ作りを行った。その振り返りの記述から多くの学習者がプログラミングを通して学習内容を理解していると考察できるとし、授業で扱った問題と同様の評価問題の正答率が87%だったと報告している。

しかし、尾崎・伊藤(2017)は「教科においてプログラミング教育を展開した場合、教科内容の理解とプログラミング的思考の両者が混在し、学習内容に的を絞れないのではないか」という懸念を示しており、「教育方法の一つとしてプログラミング教育を導入することによって、従来の教育方法より教科内容の理解が深まるのかを留意する必要がある」と述べている。

このように、教科学習の中にプログラミング的思考を意識した授業が報告されつつあるが、その効果についての検証はまだ不十分である。

1.4 学習指導要領の遷移と実証研究からみられる課題

小学校学習指導要領（平成 29 年告示）では、プログラミング教育は教科横断的に行われるように示されているが、どの教科・どの単元で行うかは一部（小学校 5 年算数科，6 年理科，総合的な学習の時間）を除いて、各学校のカリキュラムマネジメントに任されている。それによって、プログラミング教育は必修であるにも関わらず、どのような内容を行うべきなのかが教える側には分かりにくく、プログラミング教育に対する不安が広がっている一因だと考える。

さらには授業時数の確保が難しい状況であるため、教科以外でプログラミングについての授業を行うことに困難さも見られる。これまでに示した先行研究においても教科外で行われているものが多く、研究に要した授業時数も多いものもある。

このように、どのような授業を行えばよいのかが明確になっていないこと、各教科の授業時数を確保するためにも無理のないカリキュラムマネジメントを行うことが今後求められていくと考える。

1.5 問題の整理

これまで述べてきたように、日本の学習指導要領で初めてプログラミングに関する内容が記述されたのは 1989 年まで遡る。当時は情報社会に適応するためにコンピュータ操作能力が必要だという考えから「コンピュータを教える」プログラミング教育がされてきた。しかし、特定のプログラミング言語の技術を習得してもそれらは時代の移り変わりとともに廃れていくものであるため、技術を教えるのではなく思考や表現の道具としてコンピュータを活用する「コンピュータで教える」プログラミング教育が注目されていった。また、人間が情報を解析することで価値を生み出していた情報社会から、情報を人間の能力を超えた AI が解析し、その恩恵を受ける新たな社会へと移り変わりつつあるといわれている中で、その予測困難な社会を生きていく子どもたちに、目の前の問題を解決するため「何をどのような順番で組み合わせればいいのか」を考える力、すなわちプログラミング的思考を身に付けさせることが求められている。このプログラミング的思考は先述の「コンピュータで教える」プログラミング教育につながるものであり、論理的に考えることを目的としてプログラミングを授業に取り入れることにより、それらの力を育むことができたという研究結果がある。

しかしながら、プログラミングの体験は総合的な学習の時間などの教科外で行われているものが多く、子どもに体験をたくさんさせたくても授業時数の確保を考えると取り組みづらいこと、さらには、論理的な思考力を客観的に評価することは難しく、思考力及び教科内容の習得への効果に関する検証実例がまだ少ないということが報告されている。

これらから、小学校学習指導要領（平成 29 年告示）で明記されている教科・内容（第 5 学年算数科：図形，第 6 学年理科：物質・エネルギー，総合的な学習の時間）以外において、プログラミングを体験しながら教科の学びをより確実なものにできるかの実証研究があまりなされていないといえる。

そのため、小学校学習指導要領（平成 29 年告示）で示されていない教科・内容でプログラミングを取り入れる授業を検討することは価値のある課題だと考える。

2 目的

小学校学習指導要領（平成 29 年告示）において、プログラミングを体験しながら論理的思考力を身に付けるための学習活動として例にあげられている教科・内容以外についても、プログラミングを取り入れることでプログラミング教育のねらいでもある「教科等での学びをより確実なものとする」ことができるのかについて検証する。

そこで本研究では、小学校 3 年算数科の乗法の筆算の学習において、ビジュアルプログラミング言語である Scratch を用いてプログラミング的思考を意識した授業を構成することが、筆算の仕方について算数用語を適切に用いて説明する論理的思考力、及び計算の技能をより高めることに有効であることを明らかにする。

3 方法

3.1 調査対象

四日市市内の小学校 2 校の 3 年生（各 2 クラス）を調査対象とし、令和元年 9 月から 11 月にかけて調査を行った。検証授業は研修員・研究協力員が行い、記録・分析は研修員が行った。

3.2 実験群及び統制群

乗法の筆算の学習において、Scratch を活用した授業を行う学校を実験群とし、Scratch を用いずに従来の授業展開で行う学校を統制群とした。単元に要する授業時数に差があると正しい実験検証結果が得られないと考え、後述する指導計画のとおり授業時数が合うように調整を行った。

3.3 本研究で求める「教科の学び」

小学校学習指導要領（平成 29 年告示）の算数科の目標は、次のとおりである。

数学的な見方・考え方を働かせ、数学的活動を通して、数学的に考える資質・能力を次のとおり育成することを目指す。

- (1) 数量や図形などについての基礎的・基本的な概念や性質などを理解するとともに、日常の事象を数理的に処理する技能を身に付けるようにする。
- (2) 日常の事象を数理的に捉え見通しをもち筋道を立てて考察する力、基礎的・基本的な数量や図形の性質などを見だし統合的・発展的に考察する力、数学的な表現を用いて事象を簡潔・明瞭・的確に表したり目的に応じて柔軟に表したりする力を養う。
- (3) 数学的活動の楽しさや数学のよさに気付き、学習を振り返ってよりよく問題解決しようとする態度、算数で学んできたことを生活や学習に活用しようとする態度を養う。

解説（算数編）では、「算数の知識及び技能やそれを支える概念や原理・法則などには、用語や記号によって表現されるものが多い。用語や記号の意味やよさが分かるようにするとともに適切に用いる

ことができるような配慮が必要」とある。また、「数学的な表現を用いることで、事象をより簡潔、明瞭かつ的確に表現することが可能になり、論理的に考えを進めることができるようになったり、新たな事柄に気付いたりすることができるようになる」とある。よって、算数用語や記号を適切に用いて事象をより簡潔、明瞭かつ的確に表現しようとするのが論理的思考につながると考えた。

本研究で扱う第3学年算数科の「乗法の筆算」では、2位数や3位数に1位数をかける乗法と筆算の形式を学ぶ。本単元の内容は以下のとおりである。

第3学年 算数

A 数の計算

(3) 乗法にかかわる数学的活動を通して、次の事項を身に付けることができるよう指導する。

ア 次のような知識及び技能を身に付けること。

(ア) 2位数や3位数に1位数や2位数をかける乗法の計算が、乗法九九などの基本的な計算を基にしてできることを理解すること。また、その筆算の仕方について理解すること。

(イ) 乗法の計算が確実にでき、それを適切に用いること。

(ウ) 乗法に関して成り立つ性質について理解すること。

イ 次のような思考力、判断力、表現力等を身に付けること。

(ア) 数量の関係に着目し、計算の仕方を考えたり計算に関して成り立つ性質を見いだしたりするとともに、その性質を活用して、計算を工夫したり計算の確かめをしたりすること。

これらから、本研究における「教科の学び」で特に身に付けさせたい内容は、乗法の筆算の仕方について、算数用語を適切に用いて順序立てて説明できるという論理的思考力と、乗法の計算を確実にできるという技能とした。

3.4 論理的思考を育む授業を展開するための手立て

3.4.1 Scratch の活用

Scratch (<http://scratch.mit.edu>) はMIT (マサチューセッツ工科大学) のメディアラボにあるライフロング・キンダーガーテングループが開発・運営しているビジュアルプログラミング言語の学習サービスである。機能ごとに色分けされたブロックを組み合わせることでプログラムを作る。変数、代入、座標といった小学校の算数では習わない概念も扱うが、ブロックを組み立てていくだけで画像処理や音声、計算、条件などを指定できるため、複雑な文法や高度な数学の知識を覚える必要はなく、対象年齢は7歳以上とされている。

ライフロング・キンダーガーテングループの中心であるレズニック (2017) は、子どもが創造的思考者として成長するのを助けるためには、子どもが安心して間違いを犯すことができたり、間違いから学ぶことができたりする環境を作り出す必要があり、プログラムをデバッグするためのアプローチ、すなわち問題を特定して切り分け、問題を回避するための修正を行う方法を学ぶことは、あらゆるタイプの問題解決とデザイン活動に役立つと述べている。Scratch では間違いから回復しやすく、修正も行いやすいため、新たに試すことが可能である。ゆえに、分析・修正・改善を何度も経験することができ、その繰り返しによって論理的思考の向上へとつながると考え、本研究は Scratch を取り入れ

ることとした。

3.4.2 モデル（完成見本）を再現させる

Scratch を用いてモデルの動作を見せて、そのモデルを再現するためにはどのような命令、手順が必要かを分析し、それをもとに検証・修正・改善を繰り返してモデルへとつなげる学習を設定した。

米田（2016）は、旧来のプログラミング教育の最大の難点は、「アイデアを考えたり、それを実現するための仕組みを検討することよりも先に、開発時に必要な『特定のプログラミング言語の文法』を学習させること」にあり、現在のプログラミング教育では、「実際に何かを作る過程を経験しながら（もしくは経験した後で）、プログラミング言語の文法を学んでいく流れが主流」と述べている。さらに、『『何らかのアイデアを形にすること』をゴールに設定して学習することをお勧め』するともある。そこで、まずモデルを見せて自分で作ってみたいと思わせることが大切だと考えた。また、モデルからどのような動きをしているかを分析し、どのような命令（記号）を使い組み合わせるのかを考えさせるのは、Figure1 で示したプログラミング的思考の流れとも合致する。よって、モデルを先に見せてそれに近づけるために試行錯誤を繰り返し行わせる本研究の手法によって、論理的思考を育むことができるかを検証することにした。

3.4.3 プログラミングシート

モデルの動きを見て、どのような命令をどのように組み合わせれば実現できるかを考えさせる際、まずはプリント等に自分の考えを書かせることを大切にした。それは、自分の考えとモデルの動きとの差について、どうしてそうなるのかという分析や、どこをどのように修正すればよいかを考察することが、論理的思考につながると考えたからである。自分の考えをもたずに話し合い活動を行うと、他の子の意見を聞くだけの児童が多くなり、思考をしたとは言いがたい。また、なんとなく変えてみただけでは、同じく思考をしたとは言えず、変えたことによって生じた変化について分析することが大切だと考えた。

そこで、菅野（2017）が作成したプログラミングシートを参考に、モデルを見て自分の考えを書き残すためのワークシートを作成して用いた（資料1～5参照）。これにより、各自が自分の考えをもったうえで話し合いに参加することができ、さらに、前時までの学習と共通する部分と新たに出てきた部分を整理することで、本時の課題をより明確に意識することができると思った。

3.4.4 命令ブロックを並べ替える

プログラミングシートを用いて乗法の筆算の流れを確認した後、未完成のScratchのプログラムを各児童のタブレットへ配付し、先ほど見たモデルのように動作するためにはどの命令をどの順番に並べると実現できるかを考えさせて、実際に各自で命令ブロックを操作してプログラムを完成させた。例としてFigure2に第4時の課題を示す（資料1参照）。

本来であれば児童に筆算のプログラムを全て作らせたいところではあるが、難易度が高く授業時数

も限られているため、プログラムの根幹部分（資料6参照）は指導者が事前に作成したものを用意した。乗法の仕方につながる部分については、命令ブロックを組み合わせたもの（以下命令ブロック群と呼ぶ）を候補として並べ、命令ブロック群をどの順番に並べるとよいかを考えさせるようにした。また、前時から条件や命令が追加されたところを考えさせるようにして、授業を重ねるごとに繰り返り上がりがあるものや、かけられる数が2位数から3位数のものへとプログラムを発展させた。

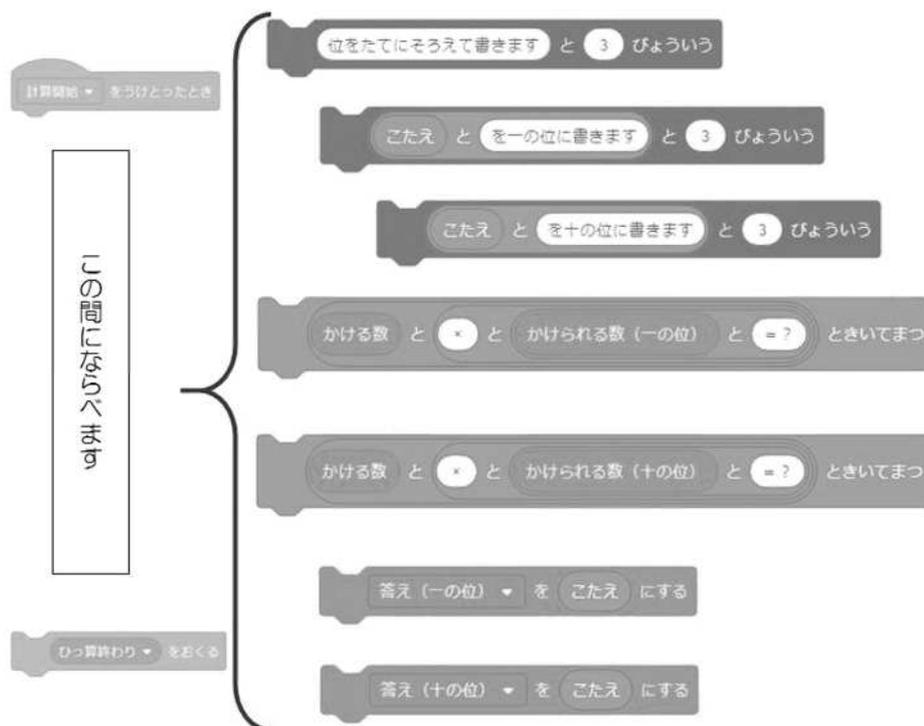


Figure 2 第4時におけるScratchの課題

3.5 計算の技能をより高めるための手立て

3.5.1 適切な命令ブロック群を選択する

児童がScratchのプログラムを組むうえで計算の手順、部分積の処理について理解が深められるように、予め用意した命令ブロック群から適切なものを選択する課題も設定した。課題の例としてFigure3に第9時におけるScratchの課題を示す（資料5参照）。

百の位の答えをどのようにプログラムするかという課題だが、選択肢として、かける数×かけられる数（百の位）の答えのみを書くのか、その答えと百の位にくり上げる数を足した数を書くのかの2つを用意した。児童には選択とともにどのように考えたかを記述させた。このように、命令ブロック群を選択する中で、乗法において大切な部分積の処理を確認させることで計算の技能に関する理解を深めることができると考えた。

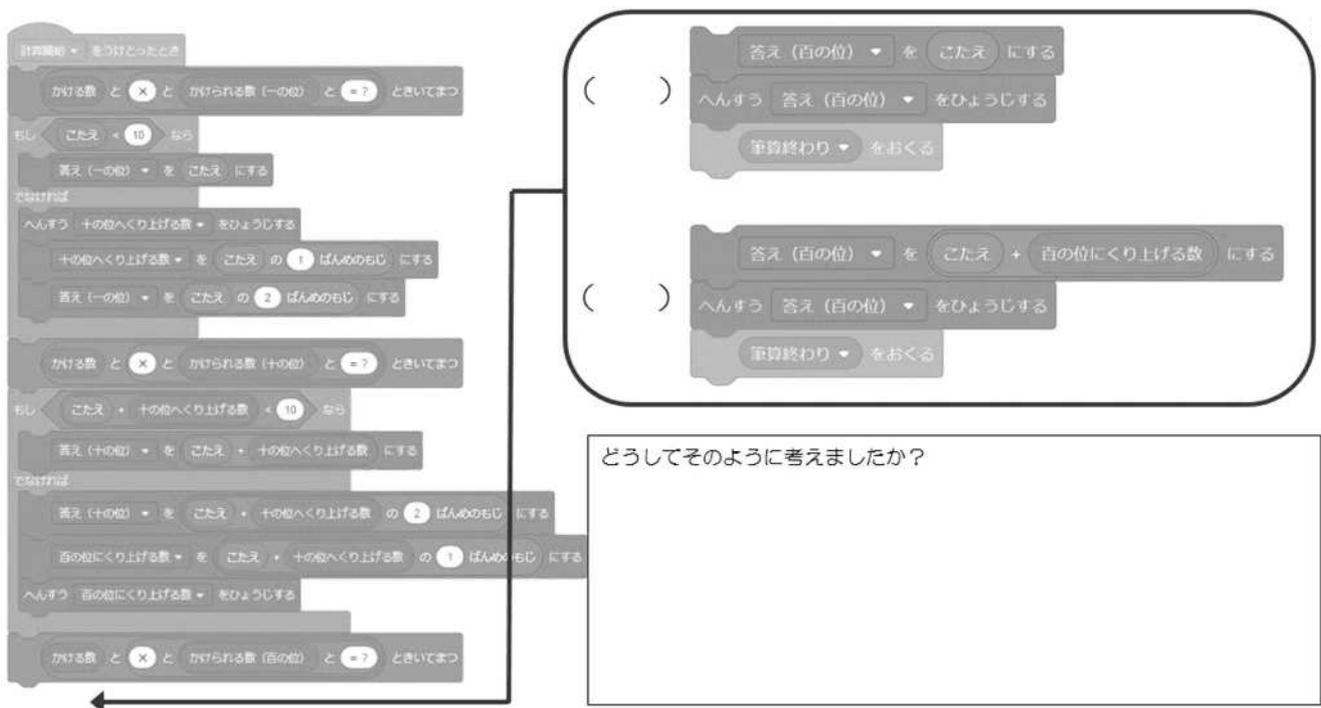


Figure 3 第9時におけるScratchの課題

3.5.2 プログラムを活用した個別学習

実験群，統制群とも筆算の仕方を学んだ後に，ノートを使って適用問題に取り組み，その答え合わせを，実験群は完成させたプログラムを用いて自己採点させた。

完成したScratchの動作を見ながら適用問題を自己採点させることで，答えのチェックだけでなく，答えを導き出すまでの手順についても確認することができ，さらに，個人の能力に合わせた学習を展開することができる考えた。例えば，計算力がある児童は適用問題だけでなく補充問題まで進んで取り組むことができる一方，筆算の手順がまだ定着していない児童は，まずはScratchで適用問題を解いてから自分のノートに書く等，児童に合わせた学習の進め方を自ら判断して取り組むことが可能となるため，乗法の筆算の技能をより確実なものにすることができると考えた。

3.5.3 間違いをすぐに修正する

自己採点で児童が九九などを間違えたまま計算を終えて誤答を出した場合，もう一度計算をやり直すように促すプログラムを根幹部分に組み込んだ。そのことにより計算ミスがあった場合に，どこが間違っているのか考え直すことになり，覚え間違いをしていた九九や計算の仕方を児童自身で修正するようになる。このように既習内容の復習もできると考えた。

3.6 データの収集と分析

本研究では事前テスト、事後テスト及び遅延テストのデータをそれぞれ2校から収集し、Scratchの活用の有無による論理的思考力及び計算の技能の変化について分析を行った。なお、それぞれのテストを受講できなかった児童は合計から除外した。

3.6.1 事前テスト

実験検証を行う前に、実験群・統制群ともに事前テスト（資料7参照）を実施した。内容は計算問題として第3学年算数科ですでに学習を終えている3位数+3位数の筆算（A(1)～A(3)）と、筆算を用いない乗法（B(1)～B(10)）、本単元で学習する乗法の筆算（C(1)～C(4)）を設定し、記述問題として3位数+3位数の筆算（繰り上がりあり）の仕方を説明するものを設定した。3位数+3位数の筆算、筆算を用いない乗法については、技能としてどれだけ力が定着しているかを把握するために設定した。記述問題は実験検証前にどれだけ論理的に書くことができるかを把握することを目的としている。

記述問題の評価基準と解答例については村上・川崎・妹尾・木村・松浦・植田（2010）が作成したルーブリックを参考にTable2にまとめた。評価基準でいう「適切」とは、できるだけ使う言葉に設定している算数用語の「位」「繰り上げ（繰り上がり）」を以下の条件で使用していること、また、必要以上に使用していないことを基準とする。

【「位」について】

どの位の計算を説明しているかを明示するため、もしくは「(数字)を(位)へ繰り上げる」のように、その繰り上げ先を指定するために各位とも1度は用いている

【「繰り上げ（繰り上がり）」について】

繰り上がりがある度に用いており、その際どの数字を繰り上げるのかも明示している

Table 2 記述問題の評価基準及び解答例（365+472の筆算の仕方）

評価基準		解答例
V	適切に説明をしてお り、数値も正しく正答 を導き出している	まず、 <u>一の位</u> の計算をします。5+2=7なので、答えの <u>一の位</u> に7を書きます。次に、 <u>十の位</u> の計算をします。6+7=13なので、答えの <u>十の位</u> に3を書き、 <u>1</u> を百の位に繰り上げます。最後に、 <u>百の位</u> の計算をします。 <u>1繰り上がっている</u> ので1+3+4=8となり、 <u>百の位</u> に8を書きます。だから答えは837になります。 ----- <u>一の位</u> は5+2=7なので7を書きます。 <u>十の位</u> は6+7=13なので3を書き、 <u>1</u> を繰り上げます。 <u>百の位</u> は1+3+4=8なので8を書きます。だから答えは837になります。
IV	適切に説明はしている が、数値を誤って正答 を導き出していない	一の位は5+2=7なので7を書きます。十の位は <u>6+7=14</u> なので4を書いて1を百の位に繰り上げます。百の位は1+3+4=8なので8を書きます。だから答えは8 <u>4</u> 7になります。

III	適切な説明ではないが 正答を導き出している	【「位」を使っていない】 5 + 2 = 7なので7を書きます。6 + 7 = 13なので3を書き1を繰り上げます。1 + 3 + 4 = 8なので8を書きます。だから答えは837になります。
		【「繰り上げ」を使っていない】 一の位は5 + 2 = 7なので7を書きます。十の位は6 + 7 = 13なので3を書きます。百の位は1 + 3 + 4 = 8なので8を書きます。だから答えは837になります。
II	適切な説明ではなく正答も導き出していない	5 + 2 = 7なので7を書きます。6 + 7 = 13なので3を書きます。3 + 4 = 7なので7を書きます。だから答えは737になります。
I	無解答	無解答

3.6.2 事後テスト

実験検証後に実験群・統制群ともに事後テスト（資料8参照）を実施した。内容は計算問題として筆算を用いない乗法（B(1)～B(10)）、本単元で学習する乗法の筆算（C(1)～C(4)）を設定し、記述問題として3位数×1位数の筆算（繰り上がりあり）の仕方を説明するものを設定した。筆算を用いない乗法及び乗法の筆算については乗法の技能がどれだけ定着しているかを把握するために設定した。記述問題は実験検証後にどれだけ論理的に書くことができるようになったかを把握することを目的としている。

記述問題の評価基準と解答例についてはTable3にまとめた。評価基準でいう「適切」について、できるだけ使う言葉に設定している「位」「繰り上げ（繰り上がり）」の条件は事前テストと同様の基準とする。それに加えて、各位の計算についてまずは乗法を行い、繰り上がりがある場合はその加法を行った後に上位への繰り上がりの有無を明記していることも基準とする。繰り上がりの数を足すことで上位への繰り上がりの数が増える場合があるため、つまずきをなくすために、繰り上がりの数を足してから上位への繰り上がりを書くように理解させることが必要だと考えるからである。

Table 3 記述問題の評価基準及び解答例（237×4の筆算の仕方）

評価基準		解答例
V	適切に説明をしてお り、数値も正しく正答 を導き出している	まず、 <u>一の位</u> の計算をします。4 × 7 = 28なので、 <u>答えの一の位</u> に8を書き、 <u>2を十の位に繰り上げ</u> ます。次に、 <u>十の位</u> の計算をします。4 × 3 = 12だけ <u>繰り上がり</u> の2を足して14なので、 <u>答えの十の位</u> に4を書き、 <u>1を百の位に繰り上げ</u> ます。最後に、 <u>百の位</u> の計算をします。4 × 2 = 8で <u>繰り上げた</u> 1を足して9なので、 <u>百の位</u> に9を書きます。だから答えは948になります。

		一の位は $4 \times 7 = 28$ なので8を書いて2を繰り上げます。十の位は $4 \times 3 = 12$ だけど繰り上がりの2を足すので4を書いて1を繰り上げます。百の位は $4 \times 2 = 8$ だけど繰り上がりの1を足すので9を書きます。だから答えは948になります。
IV	適切に説明はしているが、数値を誤って正答を導き出していない	一の位は $4 \times 7 = 24$ なので4を書いて2を繰り上げます。十の位は $4 \times 3 = 12$ だけど繰り上がりの2を足すので4を書いて1を繰り上げます。百の位は $4 \times 2 = 8$ だけど繰り上がりの1を足すので9を書きます。だから答えは944になります。
III	適切な説明ではないが正答を導き出している	【「位」を使っていない】 $4 \times 7 = 28$ なので8を書き2を繰り上げます。 $4 \times 3 + 2 = 14$ なので4を書き1を繰り上げます。 $4 \times 2 + 1 = 9$ なので9を書きます。だから答えは948になります。
		【「繰り上げ」を使っていない】 一の位は $4 \times 7 = 28$ なので8を書きます。十の位は $4 \times 3 + 2 = 14$ なので4を書きます。百の位は $4 \times 2 + 1 = 9$ なので9を書きます。だから答えは948になります。
		【「繰り上げ」がある場合の計算の手順が違う】 一の位は $4 \times 7 = 28$ なので8を書いて2を繰り上げます。次に $4 \times 3 = 12$ なので1を百の位に繰り上げます。2繰り上がっているので $2 + 2$ をして4を十の位に書きます。百の位は $4 \times 2 = 8$ だけど繰り上がりの1を足すので9を書きます。だから答えは948になります。
II	適切な説明ではなく正答も導き出していない	$4 \times 7 = 28$ $4 \times 3 = 12$ $4 \times 2 = 8$ だから答えは828です。
I	無解答	無解答

3.6.3 遅延テスト

事後テストを行ってから約1ヶ月が経過した頃に、事後テストと全く同じ内容の遅延テストを行った。遅延テストと事後テストの結果を比較し、乗法の筆算の理解度や論理的に説明する力がどれほど定着しているかを調べた。遅延テストは事後テストと全く同じ内容のため、記述問題の評価基準・解答例はTable3に準拠する。

3.7 指導計画

授業計画は次のTable4のとおりである。

Table 4 指導計画（東京書籍 新編 新しい算数 第3学年より）

時数	学習内容	
	実験群（Scratch 活用）	統制群（Scratch 非活用）
事前テストを行う		
第1,2時	何十，何百に1位数をかける乗法計算の仕方を理解し，その計算ができる	
第3時	23×3の計算の仕方をアレイ図や模擬貨幣を使ったり数操作をしたりして考える	
第4時	23×3の筆算の仕方をScratchの動作を見てプログラミングシート①にまとめ，プログラムを完成させて適用問題に取り組む	23×3（部分積がみな1桁）の筆算の仕方を知り，適用問題に取り組む
第5時	1辺16cmの正方形の周囲を求める式を考え，その筆算の仕方をScratchの動作を見てプログラミングシート②にまとめ，プログラムを完成させて適用問題に取り組む	1辺16cmの正方形の周囲を求める式を考えその筆算（一の位の数との部分積が2桁）の仕方を知り，適用問題に取り組む
第6時	42×3，58×3の筆算の仕方をScratchの動作を見てプログラミングシート③にまとめ，プログラムを完成させて適用問題に取り組む	42×3，58×3（十の位の数との部分積が2桁，及び部分積がみな2桁）の筆算の仕方を知り，適用問題に取り組む
第7時	29×4，76×4の筆算の仕方をScratchの動作を見てプログラミングシート④にまとめ，適用問題に取り組む	29×4，76×4（部分積を加えたときに百の位に繰り上がりあり）の筆算の仕方を知り，適用問題に取り組む
第8時	312×3の筆算の仕方をScratchの動作から学び，適用問題に取り組む	312×3（部分積がみな1桁）の筆算の仕方を知り，適用問題に取り組む
第9時	386×2の筆算の仕方をScratchの動作を見てプログラミングシート⑤にまとめ，プログラムを完成させて適用問題に取り組む	386×2（一，十の位の数との部分積が2桁）の筆算の仕方を知り，適用問題に取り組む
第10時	937×4の筆算の仕方をScratchの動作から学び，適用問題に取り組む	937×4（部分積がみな2桁，及び部分積を加えたときに繰り上がりあり）の筆算の仕方を知り，適用問題に取り組む
第11時	3つの数の乗法を1つの式で表して3口の乗法の結合法則をまとめる	
第12時	140cmの3倍の長さを求める計算を考える	
第13時	何倍かを求める計算について考え何倍かを求めるには乗法を使えばよいことをまとめる	
第14時	「力をつける問題」に取り組む	
第15時	「仕上げ」に取り組む	
事後テストを行う		
遅延テストを行う（事後テストを行ってから約1ヶ月後に実施）		

3.8 研究計画

研究計画は次の Table5 のとおりである。

Table 5 研究計画

月	本研究に関する計画	実施する内容・協力校との連携
4	課題研究打ち合わせ会	
5	第1回課題研究会議	・研究協力校への依頼
6	第2回課題研究会議	
7		・研究協力員との打ち合わせ ・調査対象クラスの授業参観
8		
9	第3回課題研究会議	・研究協力員との打ち合わせ ・調査対象クラスの授業参観 ・事前テスト実施 ・検証授業実施
10	第4回課題研究会議	・検証授業実施 ・事後テスト実施
11	第5回課題研究会議	・遅延テスト実施
12	第6回課題研究会議	
1	第7回課題研究会議	
2	第8回課題研究会議	

4 結果

4.1 記述問題における論理的思考力に関する変化

事前テストの記述問題（ $365 + 472$ の筆算の仕方）と事後テストの記述問題（ 237×4 の筆算の仕方）、遅延テストの記述問題（事後テストと同一問題）について Table2 及び Table3 の評価基準にもとづいた人数と割合を表にしたものが Table6 であり、実験群と統制群に分けてグラフに表したものが Figure4 及び Figure5 である。

Table 6 記述問題における事前・事後・遅延テストの比較

評価基準	実験群 (n=76)						統制群 (n=71)					
	事前テスト (加法)		事後テスト (乗法)		遅延テスト (事後と同一)		事前テスト (加法)		事後テスト (乗法)		遅延テスト (事後と同一)	
	(人数)	(%)	(人数)	(%)	(人数)	(%)	(人数)	(%)	(人数)	(%)	(人数)	(%)
I	4	5.3	1	1.3	2	2.6	2	2.8	3	4.2	1	1.4
II	10	13.2	6	7.9	9	11.8	6	8.5	10	14.1	8	11.3
III	46	60.5	30	39.5	22	28.9	51	71.8	37	52.1	49	69.0
IV	1	1.3	1	1.3	2	2.6	1	1.4	1	1.4	0	0
V	15	19.7	38	50.0	41	53.9	11	15.5	20	28.2	13	18.3

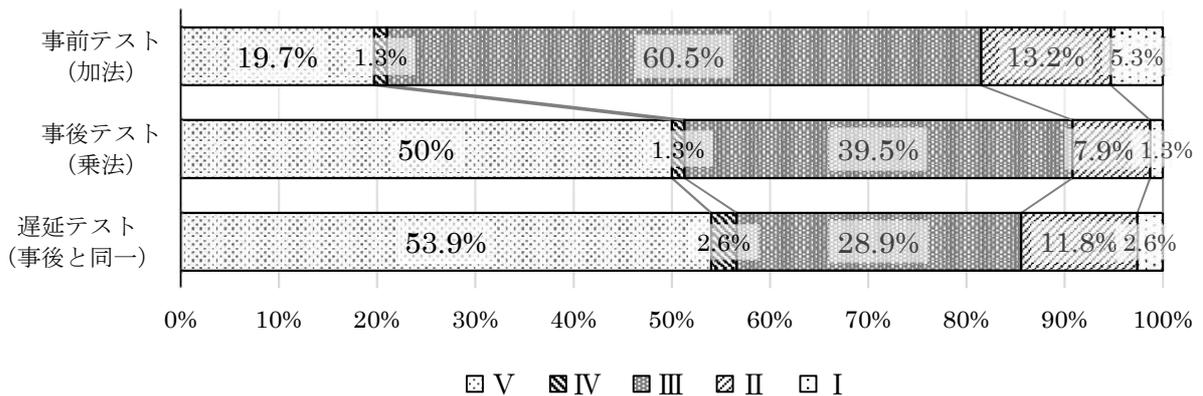


Figure 4 実験群の記述問題における事前・事後・遅延テストの比較

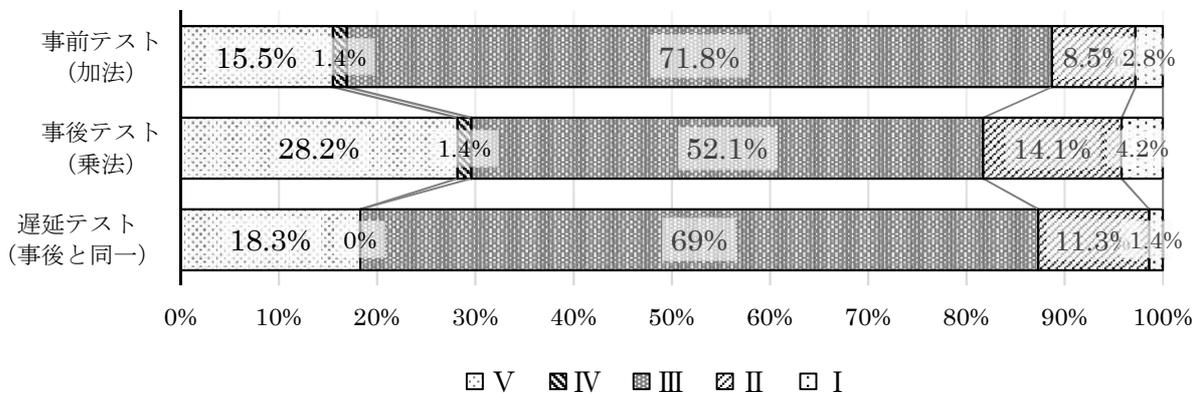


Figure 5 統制群の記述問題における事前・事後・遅延テストの比較

4.1.1 事前テストと事後テストの比較

Table6 より、事後テストにおける評価基準Vの児童の割合は、統制群が 28.2%に対して実験群は 50.0%と実験群の方が 21.8 ポイント高い結果となった。また、事前テストから事後テストにおける評価基準Vの児童の割合の変化について、統制群は 15.5%から 28.2%へと 12.7 ポイント増加し、実験群は 19.7%から 50.0%へと 30.3 ポイント増加している。これらを比較すると、評価基準Vの児童の割合の変化は実験群の方がおよそ 17.6 ポイント高いといえる。

次に、事前テストの記述問題において評価基準ごとに分類した児童が、事後テストの記述問題においてどのように変容したかに関して、実験群については Table7 に、統制群については Table8 に示す。統制群の事前テストで評価基準Ⅲだった児童 51 人のうち、評価基準Vに変わった児童は 11 人 (21.6%) だった。それに対して、実験群の事前テストで評価基準Ⅲだった児童 46 人のうち、評価基準Vに変わった児童は 25 人 (54.3%) であった。これらを比較すると、評価基準Vに変わった児童の割合は実験群の方が 32.7 ポイント高いという結果になった。

**Table 7 実験群の記述問題における
事前テストから事後テストへの変容**

評価基準		事後テスト					
		I	II	III	IV	V	計
事前 テスト	I	1	0	3	0	0	4
	II	0	1	5	1	3	10
	III	0	5	16	0	25	46
	IV	0	0	1	0	0	1
	V	0	0	5	0	10	15
	計	1	6	30	1	38	76

**Table 8 統制群の記述問題における
事前テストから事後テストへの変容**

評価基準		事後テスト					
		I	II	III	IV	V	計
事前 テスト	I	2	0	0	0	0	2
	II	0	0	3	1	2	6
	III	0	9	31	0	11	51
	IV	0	0	0	0	1	1
	V	1	1	3	0	6	11
	計	3	10	37	1	20	71

4.1.2 事後テストと遅延テストの比較

Table6 より、遅延テストにおける評価基準Vの児童の割合は、統制群が 18.3%に対して実験群は 53.9%と、実験群の方が 35.6 ポイント高い結果となった。また、事後テストから遅延テストにおける評価基準Vの児童の割合の変化について、統制群は 28.2%から 18.3%へと 9.9 ポイント減少したのに対して、実験群は 50.0%から 53.9%へと 3.9 ポイント増加している。

次に、事後テストの記述問題において評価基準ごとに分類した児童が、遅延テストの記述問題においてどのように変容したかに関して、実験群については Table9 に、統制群については Table10 に示す。統制群の事後テストで評価基準Vだった児童 20 人のうち、評価基準Vのまま変わらなかった児童は 8 人 (40.0%) だった。それに対して、実験群の事後テストで評価基準Vだった児童 38 人のうち、評価基準Vのまま変わらなかった児童は 30 人 (78.9%) であった。これらを比較すると、評価基準Vのままだった児童の割合は実験群の方が 38.9 ポイント高いという結果になった。

Table 9 実験群の記述問題における
事後テストから遅延テストへの変容

評価基準		遅延テスト					
		I	II	III	IV	V	計
事後テスト	I	1	0	0	0	0	1
	II	0	2	4	0	0	6
	III	0	5	12	2	11	30
	IV	1	0	0	0	0	1
	V	0	2	6	0	30	38
	計	2	9	22	2	41	76

Table 10 統制群の記述問題における
事後テストから遅延テストへの変容

評価基準		遅延テスト					
		I	II	III	IV	V	計
事後テスト	I	1	2	0	0	0	3
	II	0	2	8	0	0	10
	III	0	3	29	0	5	37
	IV	0	0	1	0	0	1
	V	0	1	11	0	8	20
	計	1	8	49	0	13	71

4.2 計算問題における技能に関する変化

事前テストと事後テスト、遅延テストの計算問題 (A(1)~A(3)は事前テストのみの実施のため省略) の正答率及び平均正答数を表したものが Table11 であり、平均正答数の変移を表したものが Figure6 である。また、標準偏差を単位とした平均値差の指標である効果量 d について Table12 にまとめた。

Table 11 計算問題における事前・事後・遅延テストの正答率と平均正答数の比較

設問	実験群 ($n=76$)			統制群 ($n=71$)		
	事前 (%)	事後 (%)	遅延 (%)	事前 (%)	事後 (%)	遅延 (%)
B(1)5×6	100	100	100	97.2	100	100
B(2)3×9	94.7	98.7	97.4	97.2	97.2	95.8
B(3)8×2	94.7	98.7	98.7	94.4	93.0	94.4
B(4)4×4	98.7	97.4	98.7	94.4	94.4	98.6
B(5)9×5	97.4	100	98.7	91.5	97.2	98.6
B(6)7×8	93.4	93.4	100	93.0	94.4	94.4
B(7)10×7	97.4	100	98.7	97.2	100	97.2
B(8)4×10	96.1	100	97.4	94.4	98.6	97.2
B(9)12×3	71.1	97.4	88.2	78.9	93.0	84.5
B(10)14×2	68.4	97.4	86.8	83.1	91.5	87.3
C(1)49×2	42.1	100	96.1	38.0	97.2	88.7
C(2)28×5	21.1	93.4	90.8	35.2	93.0	83.1
C(3)466×2	21.1	92.1	94.7	32.4	94.4	88.7
C(4)394×6	9.2	77.6	80.3	25.4	71.8	69.0
平均正答数 (標準偏差)	10.1 問 (2.0)	13.5 問 (0.9)	13.3 問 (1.5)	10.5 問 (2.5)	13.2 問 (1.2)	12.8 問 (1.8)

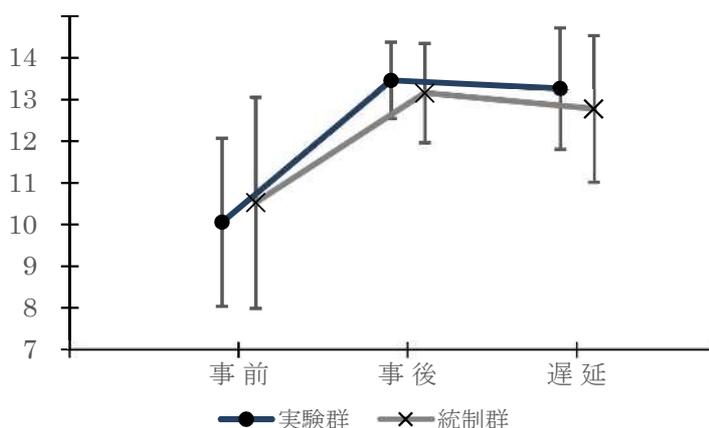


Figure 6 計算問題における平均正答数の比較

Table 12 計算問題における効果量 d の比較

	実験群	統制群
事前→事後	2.16	1.32
事後→遅延	-0.16	-0.25

4.2.1 事前テストと事後テストの比較

まず、乗法の既習内容である B(1)から B(10)の 10 問について述べる。Table11 における事後テストの正答率を比べると 2 問は 100%で同等だが、7 問は実験群の方が統制群より高い結果となった。特に B(9)は統制群が 93.0%に対して実験群は 97.4%，B(10)は統制群が 91.5%に対して実験群は 97.4%であり、それぞれ実験群の方が 4.4 ポイント、5.9 ポイント高いという結果になった。また、B(9)と B(10)において事前テストから事後テストにおける正答率の変化についても差が出た。B(9)では統制群が 78.9%から 93.0%へと 14.1 ポイント増加し、実験群は 71.1%から 97.4%へと 26.3 ポイント増加している。B(10)では統制群が 83.1%から 91.5%へと 8.4 ポイント増加し、実験群は 68.4%から 97.4%へと 29.0 ポイント増加している。

次に、本単元の乗法の筆算問題である C(1)から C(4)の 4 問について述べる。Table11 における事後テストの正答率を比べると、3 問は実験群の方が統制群より高い結果となった。特に C(4)は統制群が 71.8%に対して実験群は 77.6%であり、実験群の方が 5.8 ポイント高いという結果になった。

最後に、平均正答数と効果量 d について述べる。Table11 より、事後テストにおける統制群の平均正答数が 13.2 問であり、標準偏差が 1.2 問であるのに対して、実験群の平均正答数は 13.5 問と統制群よりも 0.3 問多く、標準偏差は 0.9 問と実験群の方が散らばりの度合いが小さいという結果となった。また Table12 より、事前テストから事後テストにおける効果量 d を比較すると、統制群は 1.32 であるのに対して実験群は 2.16 であり、0.84 ポイント高かった。

4.2.2 事後テストと遅延テストの比較

まず、乗法の既習内容である B(1)から B(10)の 10 問について述べる。Table11 における遅延テストの正答率を比べると 1 問は 100%で同等だが、8 問は実験群の方が統制群より高い結果となった。

次に、本単元の乗法の筆算問題である C(1)から C(4)の 4 問について述べる。Table11 における遅延テストの正答率を比べると、4 問とも実験群の方が統制群より高い結果となった。いずれも 6 ポイント以上差があり、特に C(4)は統制群が 69.0%に対して実験群は 80.3%であり、実験群の方が 11.3 ポイント

イント高いという結果になった。

最後に、平均正答数と効果量 d について述べる。Table11 より、遅延テストにおける統制群の平均正答数が 12.8 問であり、標準偏差が 1.8 問であるのに対して、実験群の平均正答数は 13.3 問と統制群よりも 0.5 問多く、標準偏差は 1.5 問と実験群の方が散らばりの度合いが小さいという結果となった。また Table12 より、事後テストから遅延テストにおける効果量 d を比較すると、統制群は-0.25 であるのに対して、実験群は-0.16 であり、下がり幅は実験群の方が 0.09 ポイント少なかった。

5 考察

5.1 本研究の成果

5.1.1 論理的に考える力を育むための工夫

本研究では、論理的に考える力を育むために Scratch を用いてプログラミング的思考を意識した授業を構成した。Table6 で示した記述問題における事前テストと事後テストの比較では、実験群の方が筆算の仕方を適切に説明している児童の割合及びその割合の増加率が高かった。また、Table7 と Table8 で示した記述問題における事前テストと事後テストの児童の変容の比較においても、実験群の方が適切に説明できるようになった児童数及びその割合が高く、論理的に考える力は統制群より実験群の方が身に付いたといえる。さらに、Table6 で示した記述問題における事後テストと遅延テストの比較でも、実験群の方が筆算の仕方を適切に説明している児童数及びその割合が高く、割合の変化率をみると統制群は減少しているのに対して実験群は増加している。また、Table9 と Table10 で示した記述問題における事後テストと遅延テストの児童の変容の比較においても、事後テストで適切に説明できた児童のうち、遅延テストにおいても適切に説明できた児童数及びその割合について、実験群の方が高く、論理的に考える力は統制群より実験群の方がより定着したといえる。

この結果が得られたのは Scratch を用いることで以下の2点が特に効果的だったためと考える。まず1つ目は、Scratch は命令ブロックを上から下へ順番に実行していくプログラムであり、それが順序立てて考えることに大きく関わっているという点である。論理的にものごとを説明するためには順序立てて説明する必要がある。今回の乗法の筆算においては、一の位の計算から説明を行い、繰り上がりが発生した場合はその都度どのような処理を行えばよいかを述べる必要がある。この順序が可視化される Scratch 上では、上から下へと読み進めることとつながり、児童には理解がしやすかったと考える。また、Scratch は一つ一つの命令を順番に組む必要があり、そのため繰り上がりを先に書くのか、位の数字を書くのが先なのかなど、細かな順序まで考えなければならない。その細かな順序まで意思決定することを繰り返し体験していたことにより、筆算の仕方の流れを説明する記述問題でも順序立てて書けるようになったと考える。

2つ目は、Scratch では算数用語を扱う機会を意図的に増やすことができるという点である。教科書でも「位」や「繰り上げ」という算数用語は筆算の仕方を説明するうえで出てきている。しかしながら、授業1時間の中で教科書を使って筆算の仕方を繰り返し読み返すことはあまり行われな

れに対して Scratch では、命令ブロックを組み合わせる際に、その順序を決定する要素として「位」や「繰り上げ」の算数用語に着目する必要がある。そのため、何度もそれらの算数用語を意識することとなり、印象として強く残りやすい。さらには「みため」に分類されている吹き出しを出す命令ブロックを使って算数用語を表示するようにすれば、意図的にその回数を増やすこともできる。このように Scratch を用いることで算数用語が記憶に残りやすくなり、それによって事後テストだけでなく遅延テストにおいても「位」「繰り上げ」を正しく使うことができたと考えられる。

これらのことから、論理的思考を育み定着させるうえで Scratch を活用することは有効であり、プログラミング的思考を意識した授業を構成することにより論理的に考える力を育てることができるといえる。

5.1.2 技能の習得をより確実なものにするための工夫

Table11 及び Table12 で示した、計算問題における事前テストと事後テストの比較では、統制群より実験群の方が正答率の高い問題が多く、平均正答数及び効果量 d についても実験群の方が高かったことから、計算の技能においても実験群の方が身に付いたといえる。さらに、計算問題における事後テストと遅延テストの比較についても同様に、統制群より実験群の方が正答率の高い問題が多く、平均正答数も実験群の方が多い。また、効果量 d についてはどちらも下がっているものの実験群の方が下がり幅は小さかったことから、計算の技能においても実験群の方がより定着したといえる。

この結果が得られた理由として、筆算の技能を高められるように課題を工夫したことがあげられる。乗法の筆算では、「繰り上げ」の処理がつまずきやすい。そこで本研究では、前項で述べたように「繰り上げ」を扱う機会を意図的に増やすこととともに、「繰り上げ」に着目させるために Scratch を用いて適切な命令ブロック群を選択する課題を設定した。それにより、「繰り上げ」の処理を正しく理解し、計算においても「繰り上げ」の処理に注意を払うようになったと考えられる。

また、児童が完成させた筆算のプログラムを使って個別学習を行う中で、数字の答え合わせだけでなく、計算の手順まで繰り返し確認させたことも有効だったと考えられる。従来の答え合わせは答えを読み上げて行う、指導者が採点を行う、板書によって確認させるなどがある。答えを読み上げるだけでは数字の確認しかできず、指導者による採点では数字と筆算の見た目しか確認できない。答えを板書している様子を見せるのであれば筆算の手順を確認することもできるが、全ての問題について手順を確認しながら答え合わせをしていると多くの時間を費やしてしまう。それに対して Scratch の筆算のプログラムを使って自己採点をすることで、筆算の手順を画面上で確認しながら個人の能力に合わせた学習をすることが可能となる。さらに、自己採点で答えが違った場合、間違いを見つけるために計算の手順をはじめから振り返ることになる。手順を追いながら九九や繰り上がりを考え直すため、自己採点をしながら自分のつまずきやすいところ気づくことができたと考えられる。

このように、つまずきやすい計算処理に着目させるように課題を工夫すること、計算の手順を追いながら答えを確認する学習を繰り返し行うことで、論理的に説明する力だけでなく計算の技能についてもより深く理解させることができるといえる。

5.1.3 小学校学習指導要領で示されていない教科・内容での授業実施についての考察

本研究では、プログラミングの体験を小学校学習指導要領（平成 29 年告示）で示されている教科・内容ではなく、第 3 学年算数科の乗法の筆算の単元で検証授業を行った。Scratch を取り入れることの有効性を調査するために、単元に要する総授業時数を実験群、統制群とも合わせたうえで今回の結果を得ることができた。これは Scratch のプログラムで自己採点させることで、答え合わせに有する時間を短縮したことにより、Scratch を含めた思考につながる学習に時間を増やすことができたのが大きな要因になったと考える。

このように、児童が生成したプログラムを活用する授業展開を構成することで、小学校学習指導要領（平成 29 年告示）に示されていない教科・内容でも授業時数を増やすことなくプログラミング体験を効果的にさせることができるといえる。

5.2 本研究の課題

本研究では Scratch に慣れていない児童を対象に行ったため、いくつかの命令ブロック群の中から実際にプログラムに組み込むものを選択させたり、命令ブロック群をどのような順番で組めばよいかを考えさせたりするなど、予め命令ブロックを組み合わせた命令ブロック群を使って学習を進めた。しかし、基本操作が身に付いておらず、命令ブロック群を移動するだけでもブロックを崩してしまい、それを修復できない児童も多くいた。本来 Scratch は間違いから回復しやすく修正も行いやすいため、何度も簡単に試行できることが利点としてあげられるが、その良さを生かすことができなかった。操作に戸惑うことがない程度に Scratch に慣れていれば、命令ブロック群をもう少し崩して、順序や組み合わせを試行する機会を増やすことができたと考えられる。このように、命令ブロック群の試行操作を増やすことで、さらに論理的思考力を育むことができるのか検証していきたい。

どのような力がつけば論理的思考力が高まったとするか、その捉え方についても議論を重ねる必要があると考える。本研究では、「算数用語を適切に用いて順序立てて説明する力」を論理的思考力と捉えて検証を行った。しかしながら、赤堀（2018）が「プログラミング的思考は、他教科・領域の論理的思考と関連するような総合的な論理的思考」と述べているように、論理的思考を構成する能力は幅が広く、本研究で育てることができた力もその中の一部分であると考えられる。本研究で目指した論理的思考の能力以外についても、どの教科・内容でどのような授業を行うことが有効なのかを今後も考えていきたい。

また、本研究は乗法の筆算の流れとプログラミングに親和性が高いと考え検証授業を行った。第 3 学年算数科の「乗法の筆算」においては今回のような結果が得られたが、異なる教科・内容でも同様の結果を得ることができるのか検証する余地が残っている。プログラミングとの親和性を考えたうえで、小学校学習指導要領（平成 29 年告示）に例としてあげられているもの以外で「教科等での学びをより確実なものとする」ことができる教科・内容がないか探していきたい。

最後に、本研究では児童一人一人にタブレットを配付し、作成したプログラムを使った個別学習の良さについても述べた。今後さらにタブレットなど ICT 機器を使った授業モデルが提案されていくと

思われる。その中で、デジタル教材を用いて自己採点させる学習形態など、今後「個別最適化された授業」を行ううえで、どのような授業モデルが有効なのか検討することが必要だと考える。

引用文献

- 赤堀 侃司 (2018). プログラミング教育における論理的な思考とは何か. 学習情報研究論文誌, 261(4), 56-61.
- 石嶋 洋平 (2018). 子どもの才能を引き出す最高の学びプログラミング. あさ出版
- 市川 伸一 (1994). コンピュータを教育に活かす—「触れ, 慣れ, 親しむ」を超えて— 勁草書房
- 尾崎 光・伊藤 陽介 (2017). 小学校におけるプログラミング教育実践上の課題. 鳴門教育大学情報教育ジャーナル, 15(1), 31-35.
- 黒羽 諒・伊藤 崇・川澄 陽子・小林 祐紀 (2019). 第4学年算数科「角度」の単元におけるプログラミング学習の提案. 日本デジタル教科書学会発表予稿集, 8, 59-60.
- 斎藤 亮 (2019). 算数科におけるプログラミング学習の研究—ビジュアルプログラミングの有効性— 日本デジタル教科書学会発表予稿集, 8, 37-38.
- 菅野 朋和 (2017). 論理的思考力を育む学習指導の工夫—第4学年算数科「面積」におけるプログラミングを通して— 沖縄県立総合教育センター研究収録, 61, 1-11.
- 杉野 裕子 (1988). 数学教育におけるプログラミングの利用—“学校図形 Logo”を通して—, 教育情報研究, 5(1), 79-88.
- 総務省 (2015). プログラミング人材育成の在り方に関する調査研究報告書
- 高橋 純・三井 寿哉 (2019). これが知りたかった! すぐにできるプログラミング授業実践 小学校理科. 東洋館出版社
- 滝 順一 (2016). プログラミング教育 小学校に必要なか. 日本経済新聞 2016年7月17日朝刊
- 波頭 亮 (2019). 論理的思考のコアスキル. 筑摩書房
- 松村 太郎・山脇 智志・小野 哲生・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
- 宮田 仁・大隅 紀和・林 徳治 (1997). プログラミングの教育方法と問題解決能力育成との関連—Process-oriented Approach と Content-oriented Approach との比較を通して— 教育情報研究, 12(4), 3-13.
- 村上 良太・川崎 正盛・妹尾 進一・木村 恵子・松浦 武人・植田 敦三 (2010). 論理的な図形認識を促す算数・数学科カリキュラムの開発(1)—小学校第5学年における移行を促す算数での実践的研究— 全国数学教育学会誌, 16(1), 73-85.
- 森 秀樹・杉澤 学・張 海・前迫 孝憲 (2011). Scratch を用いた小学校プログラミング授業の実践—小学生を対象としたプログラミング教育の再考— 日本教育工学会論文誌, 34(4), 387-394.
- 森本 哲郎 (1989). 子どもの「コンピュータ活用能力」の実態について—LOGO による学習を通して— 教育情報研究, 5(2), 28-38.
- 文部科学省 (2017). 小学校学習指導要領 (平成 29 年告示)
- 文部科学省 (2017). 小学校学習指導要領 (平成 29 年告示) 解説算数編, 日本文教出版
- 文部科学省 (2018). 小学校プログラミング教育の手引 (第二版)
- 山本 利一・田賀 秀子・新屋 智絵・小林 靖英 (2006). 共同学習を取り入れたプログラミング学習の課題の提案—カーリングゲームを取り入れたプログラミング指導— 教育情報研究, 22(3), 11-18.
- 山本 利一・鈴木 航平・岳野 公人・鹿野 利春 (2017). 初等教育におけるタブレットを活用したプログラミング学習の提案. 教育情報研究, 33(1), 41-48.
- 米田 昌悟 (2016). プログラミング入門講座. SBクリエイティブ
- Resnick, M., & Robinson, K. (2017). Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play. Cambridge, MA: The MIT Press. (レズニック, M. & ロビンソン, K. 酒匂 寛 (訳) (2018). ライフロング・キンダーガーデン—創造的思考力を育む 4 つの原則— 日経 BP 社)
- 和田 重雄 (2019). 探求技能の習得に通じるプログラミング教育教材の開発と実践. 日本科学教育学会研究会研究報告, 33(4), 33-38.

かけ算の筆算 (1) ①

3年 組 番 名前 ()

課題 23×3の筆算のしかたの動作を見て,どのように計算しているかを書きましょう。

1

プログラミングすること

①

2

プログラミングすること

②

③

3

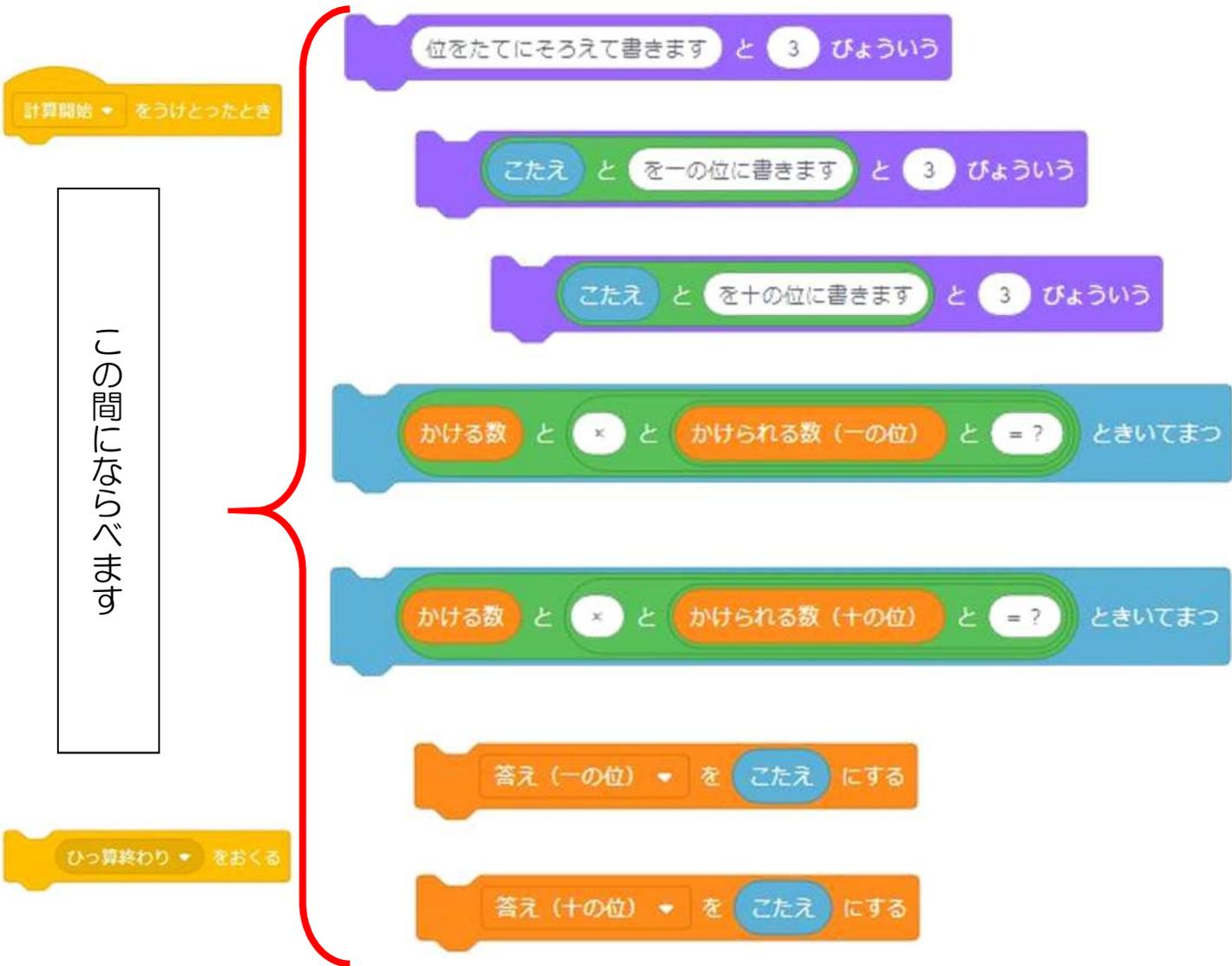
プログラミングすること

④

⑤

じゅ業をふりかえっての感想

筆算の動作をするプログラムを作しましょう。



かけ算の筆算 (1) ②

3年 組 番 名前 ()

課題 16×4の筆算のしかたの動作を見て,どのように計算しているかをくわしく書きましょう。

①

$$\begin{array}{r} 16 \\ \times 4 \\ \hline \end{array}$$

(Note: A red dashed vertical line is between 6 and 4, and a square box is in the units column.)

プログラミングすること

①

②

②

$$\begin{array}{r} 16 \\ \times 4 \\ \hline \end{array}$$

(Note: A red dashed vertical line is between 6 and 4. A small square box is in the tens column, and a larger square box is in the units column.)

プログラミングすること

③

③

$$\begin{array}{r} 16 \\ \times 4 \\ \hline \end{array}$$

(Note: A red dashed vertical line is between 6 and 4. A small square box is in the tens column, and two larger square boxes are in the units and tens columns.)

プログラミングすること

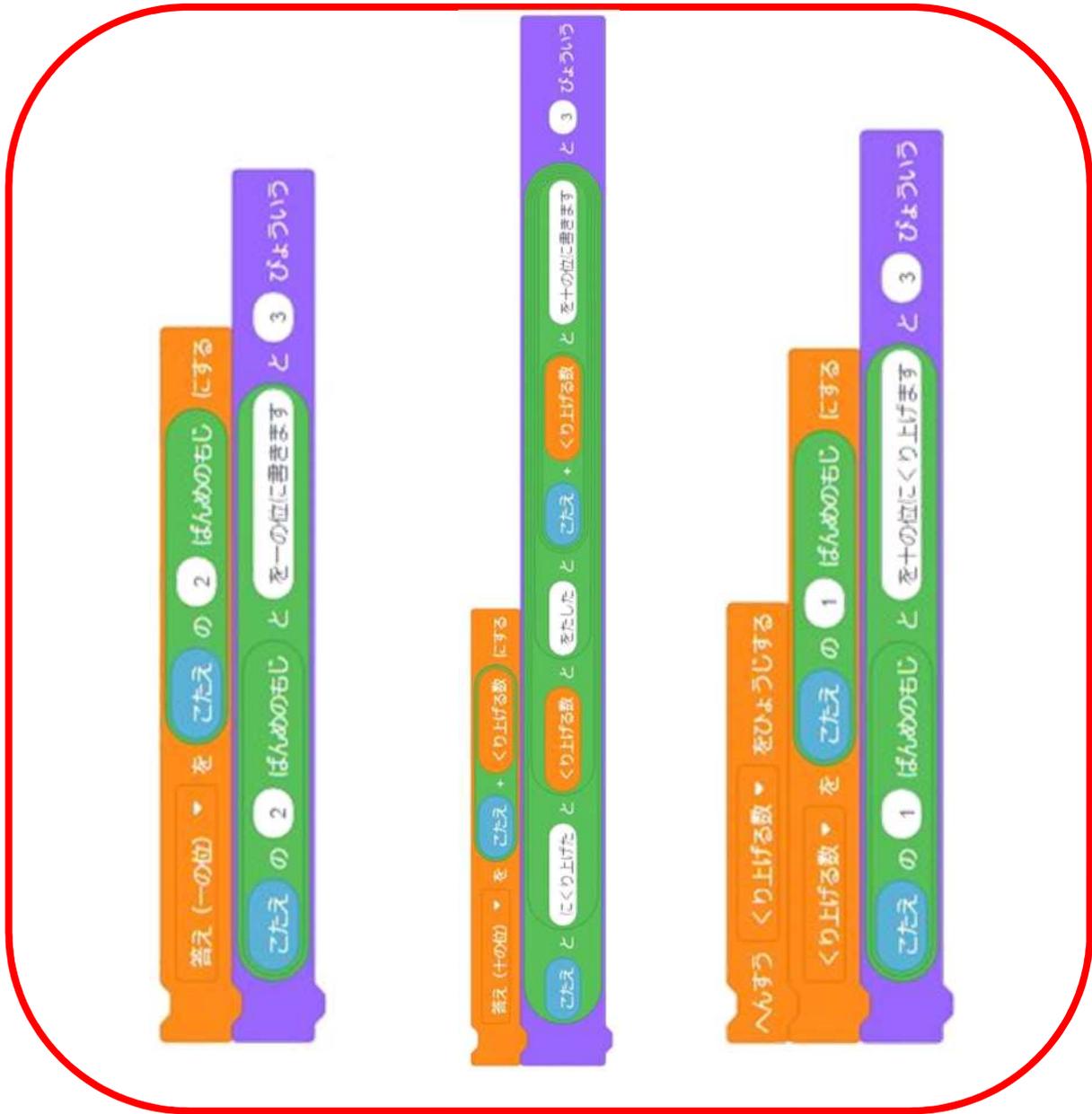
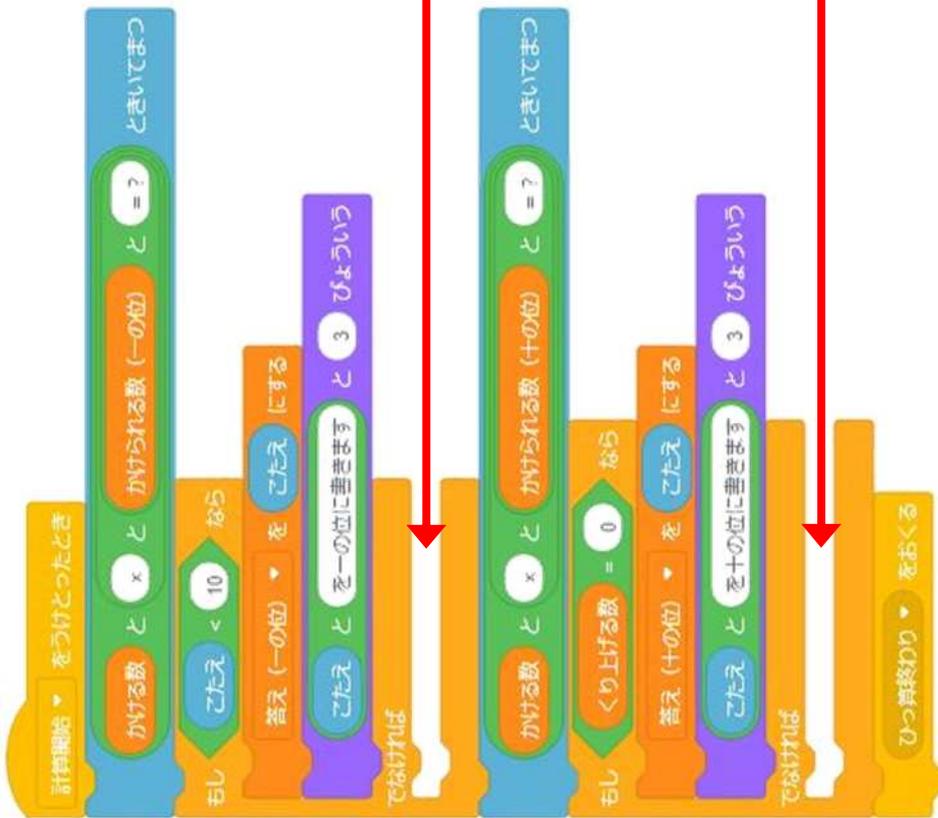
④

⑤

じゅ業をふりかえっての感想

筆算の動作をするプログラムを作りましょう。

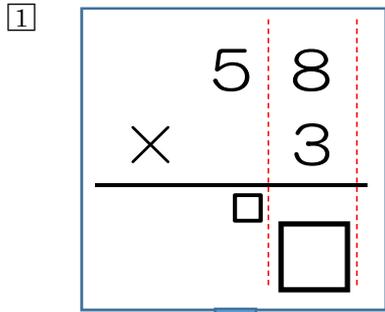
かけ算の筆算 (1) ②



かけ算の筆算 (1) ③

3年 組 番 名前 ()

課題 58×3の筆算のしかたの動作を見て、どのように計算しているかをくわしく書きましょう。

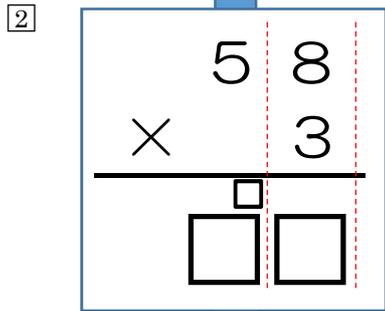


プログラミングすること

①

②

③

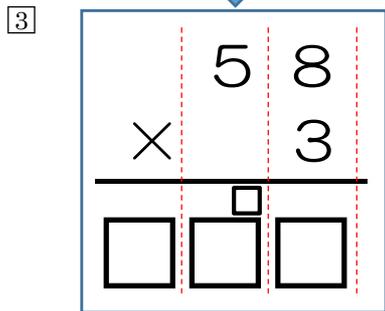


プログラミングすること

④

(どんな計算をするかな?)

⑤



プログラミングすること

⑥

じゅ業をふりかえっての感想

「でなければ」の中へどんなじゅん番で入れるとよいか考えよう。

かけ算の筆算 (1) ③

Scratch code blocks for multiplication problems (1) and (2). The blocks are arranged as follows:

- Problem (1):
 - 計算開始 ▶ をうけとったとき
 - かける数 と × と かけられる数 (一の位) と =? と ときいてまつ
 - もし 答え (一の位) < 10 なら
 - 答え (一の位) を 答え (一の位) にする
 - でなければ
 - 答え (一の位) を 答え (一の位) にする
 - 1 ひょうまつ
 - へんすう くり上げる数 ▶ をひようじする
 - くり上げる数 を 答え (一の位) の 1 ばんめのもし にする
 - こたえ の 1 ばんめのもし と を十の位にくり上げる と 4 ひょうまつ
 - かける数 と × と かけられる数 (十の位) と =? と ときいてまつ
 - もし 答え (十の位) < 10 なら
 - 答え (十の位) を 答え (十の位) にする
 - でなければ
 - ひっ算終わり ▶ をおくる
- Problem (2):
 - 答え (百の位) を 答え (百の位) にする
 - へんすう 答え (百の位) をひようじする
 - 1 ひょうまつ
 - 答え (十の位) を 答え (十の位) にする
 - こたえ + くり上げる数 の 2 ばんめのもし にする

どうしてそのように考えましたか？

かけ算の筆算 (1) ④

3年 組 番 名前 ()

課題 29×4 , 76×4 の筆算のしかたの動作を見て、どのように計算しているかをくわしく書きましょう。

①

$$\begin{array}{r} 29 \\ \times 4 \\ \hline \square \square \square \end{array}$$

(十の位の計算から)

_____にくり上げた_____をたして_____だから
十の位に_____を書いて、百の位に_____を書きます。

②

$$\begin{array}{r} 76 \\ \times 4 \\ \hline \square \square \square \end{array}$$

(十の位の計算から)

_____にくり上げた_____をたして_____だから
十の位に_____を書いて、百の位に_____を書きます。

じゅ業をふりかえっての感想

2けたのかけ算の筆算プログラム

```

    計算開始 ▾ を受け取ったとき
    かける数 と × と かけられる数(一の位) と =? と聞いて待つ
    もし 答え < 10 なら
        答え(一の位) ▾ を 答え にする
    でなければ
        答え(一の位) ▾ を 答え の 2 番目の文字 にする
        変数 くり上げる数 ▾ を表示する
        くり上げる数 ▾ を 答え の 1 番目の文字 にする
        答え の 1 番目の文字 と を十の位にくり上げる と 3 秒言う
    かける数 と × と かけられる数(十の位) と =? と聞いて待つ
    答え と にくり上げた と くり上げる数 と をたして と 答え + くり上げる数 と だから... と言う
    もし 答え + くり上げる数 < 10 なら
        答え(十の位) ▾ を 答え + くり上げる数 にする
    でなければ
        4 秒待つ
        答え(十の位) ▾ を 答え + くり上げる数 の 2 番目の文字 にする
        1 秒待つ
        答え(百の位) ▾ を 答え + くり上げる数 の 1 番目の文字 にする
        変数 答え(百の位) ▾ を表示する
    ひっ算終わり ▾ を送る
  
```

かけ算の筆算 (1) ⑤

3年 組 番 名前 ()

課題 386 × 2の筆算のしかたの動作を見て、どのように計算しているかをくわしく書きましょう。

①

プログラミングすること

①

②

②

プログラミングすること

③

(どんな計算をするかな?)

だから

④

③

プログラミングすること

⑤

(どんな計算をするかな?)

だから

⑥

じゅ業をふりかえっての感想

さいごはどちらのプログラムを入れるとよいか考えよう。

かけ算の筆算(1)⑤

どうしてそのように考えましたか？

事前テスト

年 組 番 名前()

問題 1 計算をしましょう。

$$(1) \begin{array}{r} 527 \\ + 193 \\ \hline \end{array} \quad (2) \begin{array}{r} 892 \\ + 765 \\ \hline \end{array} \quad (3) \begin{array}{r} 238 \\ + 664 \\ \hline \end{array}$$

問題 2 計算をしましょう。

- (1) 5×6 (2) 3×9
 (3) 8×2 (4) 4×4
 (5) 9×5 (6) 7×8
 (7) 10×7 (8) 4×10
 (9) 12×3 (10) 14×2

まだ習っていませんがチャレンジしてみよう！

問題 3 計算をしましょう。

$$(1) \begin{array}{r} 49 \\ \times 2 \\ \hline \end{array} \quad (2) \begin{array}{r} 28 \\ \times 5 \\ \hline \end{array} \quad (3) \begin{array}{r} 466 \\ \times 2 \\ \hline \end{array} \quad (4) \begin{array}{r} 394 \\ \times 6 \\ \hline \end{array}$$

問題 4 365 + 472 の筆算のしかたを下の言葉でできるだけ使ってせつ明しましょう。

できるだけ使う言葉

- ・位
- ・くり上げ

ポイント

つなぎ言葉を使うとさらにパワーアップ！

- ・まず～
- ・次に～
- ・～なので
- ・そうすると～
- ・だから～
- ・さい後に～
- ・など

$$\begin{array}{r} 365 \\ + 472 \\ \hline \end{array}$$

だから答えは _____ です。

事後テスト(遅延テスト)

年 組 番 名前()

問題 1 計算をしましょう。

(1) 5×6 (2) 3×9

(3) 8×2 (4) 4×4

(5) 9×5 (6) 7×8

(7) 10×7 (8) 4×10

(9) 12×3 (10) 14×2

問題 2 計算をしましょう。

$$\begin{array}{r} (1) \quad 49 \\ \times \quad 2 \\ \hline \end{array} \quad (2) \quad 28 \quad (3) \quad 466 \quad (4) \quad 394$$

$$\begin{array}{r} \times \quad 5 \\ \hline \end{array} \quad \begin{array}{r} \times \quad 2 \\ \hline \end{array} \quad \begin{array}{r} \times \quad 6 \\ \hline \end{array}$$

問題 3 237×4 の筆算のしかたを下の言葉を使
 できるだけ使ってせつ明しましょう。

できるだけ使う言葉

- ・位
- ・くり上げ

ポイント

つなぎ言葉を使うとさらにパワーアップ！
 ・まず～ ・～なので ・だから～
 ・次に～ ・そうすると～ ・さい後に～ など

$$\begin{array}{r} 237 \\ \times \quad 4 \\ \hline \end{array}$$

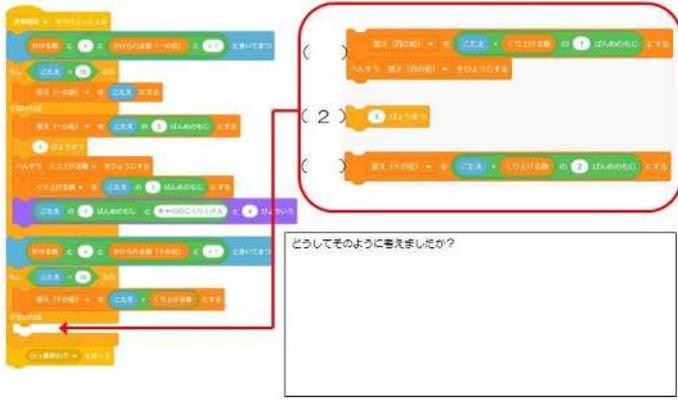
だから答えは _____ です。

小学校第3学年算数科「かけ算の筆算（1）」指導案（第6時）

○ 本時の目標

- ・ 2位数×1位数（一，十の位との部分積が2桁）の筆算の仕方について理解し，その計算ができる。
- ・ Scratch を用いて筆算の手順通りに計算するプログラムを作ることができる。

○ 指導の流れ

問題解決のプロセス&学習活動	指導上の留意点等
<p>1. 筆算の動作を見る。（流れは Scratch 画面を参照）</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>58×3の筆算の仕方を考えよう</p> </div> <ul style="list-style-type: none"> ・ 16×4との違いは何か →百の位が出てくる 	<ul style="list-style-type: none"> ・ 42×3，58×3の筆算の流れを見せる。 ・ 動きを見てノートに筆算を書く。その上で16×4と何が違うのかに気づかせる。 ・ 一の位だけでなく十の位との部分積も2桁について気づかせる。
<div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>58×3の筆算のしかたの動作を見て，どのように計算しているかをくわしく書きましょう。</p> </div> <p>2. プログラミングシートに動作について記述する。</p> <p>①① 3×8=? (24) ② 4を一の位に書く ③ 2を十の位にくり上げる</p> <p>②④ 3×5=? (15) (どんな計算をするかな?) 15+2=17 ⑤ 7を十の位に書く</p> <p>③⑥ 1を百の位に書く</p>	<ul style="list-style-type: none"> ・ プログラミングシートを配付する。 ・ 動作を見せ，「プログラミングすること」に書かせる。 ・ ②⑤⑥は動きを見て左記のような言葉を考えさせる。（コメントは表示されない） ・ (どんな計算をするかな?)の計算式は表示されないが，⑤を見てその前にどんな計算がされているかを考えさせる。
<p>3. 裏面の「でなければ」の中へ入れる命令の順番を考える。</p> 	<ul style="list-style-type: none"> ・ どの位から書くのかをプログラミングシートを見ながら確認させる。 ・ 命令の順番を書かせる（2番目は固定）。 ・ 考えの根拠についても書かせる。
<p>4. 考えを交流する。</p>	<ul style="list-style-type: none"> ・ タブレットで操作する前に意見をまとめる。

5. タブレットを使って 58×3 の筆算の仕方を作る。

6. P.101△△△に取り組む。

- ・ノートを使って問題を解く。

7. △を学級全体で答え合わせをする。

8. 授業の振り返りをする。

- ・各自で筆算のプログラムを組ませる。
- ・ペアで相談しながら作業を進めるように伝える。
- ・できたら動作チェックを行わせる。

- ・ノートを使って問題を解かせる。
- ・Scratch を使って答え合わせを各自でさせる。
- ・早く終わった児童は P. 122 の補充問題④を Scratch を使ってさせる。

- ・筆算の書き方を板書で再度確認する。
- ・「位」「くり上がり」に着目させる。

- ・ワークシートに書かせる。
- ・書き終わったらペアで確認させる。
- ・書いたことを発表させる。

○ 「かけ算の筆算（1）第6時」の動作（Scratch 画面）

①

答え 0

3×8=?

0 0

34

②

答え 0

0 4

③

答え 0

2を十の位にくり上げる

2 0 4

④

答え 0

3×8=?

2 7 4

⑤

答え 0

2 7 4

⑥

答え 174

答えは174パッチリだね！

2 1 7 4

※誤答の場合

答え 174

あれ？答えがちがうよ。もう一度計算してみよう

2 1 7 1

The image shows two Scratch code blocks for calculating multiplication with carry. The first block handles a multiplier with one digit (0-9). It checks if the product is less than 10. If yes, it sets the answer to the product. If no, it carries the tens digit to the next place. The second block handles a multiplier with two digits (0-99). It first calculates the product with the tens digit, carrying the result to the hundreds place, and then calculates the product with the ones digit, carrying the result to the tens place.

○「かけ算の筆算(1)第6時」のプログラム完成形

The image shows the completed Scratch code blocks for multiplication with carry. It combines the logic from the previous blocks into a single sequence. It starts with a 'when green flag clicked' event, followed by a 'when asked for answer' block. It then checks if the multiplier is less than 10. If yes, it calculates the product and carries. If no, it calculates the product with the tens digit, carries the result to the hundreds place, and then calculates the product with the ones digit, carrying the result to the tens place. Finally, it shows the answer and ends with a 'when asked for answer' block.

教科等での学びをより確実なものにするためのプログラミング教育に関する研究
— 小学校算数科における Scratch の活用を通して —

〔研究協力員〕	四日市市立内部小学校	教 諭	柴田 享子
	四日市市立内部小学校	教 諭	舘 英明
	四日市市立内部小学校	教 諭	坂部 久美
	四日市市立富洲原小学校	教 諭	村田 祥子
	四日市市立富洲原小学校	教 諭	坂崎 直弘
〔執筆 者〕	四日市市教育委員会	研 修 員	中塩 英昭
〔指導・助言〕	国立教育政策研究所	総括研究官	山森 光陽

研究調査報告 第410集

教科等での学びをより確実なものにするためのプログラミング教育に関する研究
— 小学校算数科における Scratch の活用を通して —

発 行 令和 2年 3月 6日
発行所 四日市市教育委員会教育支援課
四日市市諏訪町 2 番 2 号
電話 (059) 354-8149
FAX (059) 359-0280
