

算数教材とプログラミング的思考について

中 込 雄 治¹
黒 木 伸 明²

小学校において児童に一人一台のタブレットPCが配備されたことと、ビジュアルプログラミング言語が普及したことで、プログラミング教育の環境が整ってきた。本研究では算数科におけるプログラミング教育に着目し、教科書の内容をもとにして、児童にどのようなプログラミングの体験をさせることができるかを検討し、プログラムの改善に向けた試行錯誤を繰り返すことによって、プログラミング的思考が深められ、数学的内容が関連付けられながら活用されていくことを明らかにした。

Keywords : 算数教材、プログラミング的思考、ビジュアルプログラミング言語、スクラッチ

1. はじめに

プログラミング教育は2020年度から小学校に導入されている。GIGAスクール構想¹⁾により一人一台のタブレットPCが配備されたことと、ビジュアルプログラミング言語²⁾が登場し普及したことから、児童が自分のペースで手軽にプログラミングを行うことが可能な状況になってきている。

本研究では、ビジュアルプログラミング言語としてスクラッチ³⁾を採用する。算数教科書に掲載されている問題を題材として取り上げ、どのようなプログラミングの体験をさせることができるかを検討し、プログラムの実行と改善を繰り返していく過程で、どのようにプログラミング的思考⁴⁾が育まれ、数学的内容が関連付けられ活用されていくかについて明らかにする⁵⁾。

2. プログラミングの基本的な要素

プログラムを組む作業、つまりプログラミングにおいては、次の3つが基本的な要素となる⁶⁾。これらの要素はプログラミングではとりわけ重要

表2-1 プログラミングの基本的な要素

順次処理	命令を順序立てる
分岐処理	条件によって命令を分岐する(条件分岐)
反復処理	命令を繰り返す(ループ構造)

になるが、日常の学習活動においても意識的に捉え直してみると、これらの活動が取り入れられていることがわかる。実際に計算の問題を解く場合にも、計算の順序を考え、条件分岐や繰返しの活動を行っている。例えば小学校低学年の学習内容に、下記のような条件分岐やループ構造につながる考え方を見て取ることもできる。

[1] 小1「ひきざん」における条件分岐

小1の単元「ひきざん」では、「15-3」や「13-9」のような2位数から1位数を引く減法計算を学ぶ。「15-3」ならば一の位の5から3を引くことができるが、「13-9」のように一の位の3から9を引くことができない場合は繰り下がりのある減法計算になる。「13-9」の計算の仕方については図2-1のような手順が示されている⁷⁾。

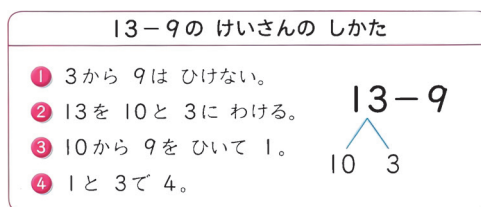


図2-1 繰り下がりのあるひき算

「13-9」のように被減数13の一の位の3から減数9を引くことができない場合は、被減数13を10と3にわけて、10から減数9を減じて1、その

1. 宮城学院女子大学教育学部
2. 上越教育大学名誉教授

1に3を加えて4、というように減加法 ($10-9=1$ 、 $1+3=4$) を用いて答えを算出するわけである。つまり、2位数から1位数を引く減法計算では、被減数の一の位の数より減数が小さいならばそのまま引き、大きいならば減加法を使って計算することになり、ここに条件分岐の考え方を見て取ることができる⁸⁾。

[2] 小2「ひっ算」における条件分岐とループ構造

小2の単元「たし算とひき算のひっ算」では、「76+58」のような2位数と2位数の加法における筆算を学ぶ。「76+58」の筆算の仕方については図2-2のような手順が示されている⁹⁾。

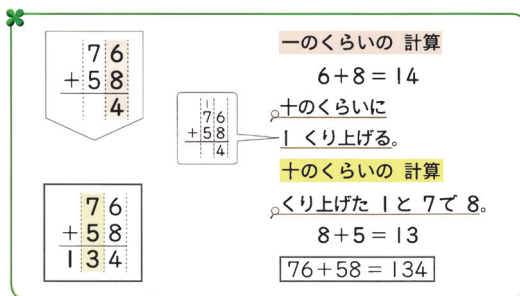


図2-2 たし算の筆算

図2-2にあるように、加法の筆算では被加数と加数を位を揃えて書き、同じ位の数を足していき答えを算出する。このとき、「同じ位の数の足した値が10未満ならばそのまま、10以上の場合は1繰り上げる」という条件分岐を含んだ作業を一の位と十の位それぞれで繰り返して行っている。加法の筆算では桁数が増えても同様の作業を繰り返すことになり、ここにループ構造の考え方を見て取ることができる。プログラムを組む作業を行うにあたっては、こうした基本的な要素を意識的に活用することになる。

3. プログラミング的思考を育成する手順

プログラミング的思考は、コンピュータを動作させることに対応させて考えると、次のような手順を繰り返すことで育成される¹⁰⁾。

[プログラミング的思考を育成する手順]

1) 意図を明確にする。

① 画面の流れをイメージする。

② 必要になる命令を選択する。

③ プログラムを組み実行する。

2) 改善に向けて①②③を繰り返したり、意図を見直したりする。

手順を把握しやすいようにキーワードをもとに図解すると下図のようになる。

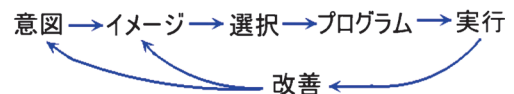


図3-1 プログラミング的思考を育成する手順

意図に基づき、流れをイメージして、命令を選択し、プログラムを作成して実行する。不具合があれば改善に向けて再びイメージ・選択・プログラム・実行を繰り返すことになる。プログラミング的思考の育成においては、この改善していくという行為を繰り返して行うことが重要になる。自分が考えた論理が正しいかどうかは、実際にコンピュータを動作させることによって検証できる。プログラムの修正・追加を行い、改善に向けた試行錯誤を重ねることによって論理的思考がより深められていく。意図が実現できれば、更にバージョンアップを目指して意図の見直しも含めた改善を図っていくことになる。

4. 扱う題材と使用するブロック (命令)

算数教科書では、現在プログラミング教材として「正多角形の作図」など¹¹⁾の題材が扱われているが、学年や単元などに限定することなく、創意工夫を生かしたプログラミング教育の展開が期待されている¹²⁾。本研究では、新たな題材を検討する。小3で一億までの整数、小4で千兆までの整数を学習することに着目し、大きな数と万進法に関する問題を取り上げる。また万進法の活用例として、年数を秒に換算する問題も取り上げた。更に小5で直方体の体積を学習することに着目し、箱の容積の問題に関しても取り上げている。

スクラッチではプログラムの命令に対応した様々なブロックが用意されており、それらのブロックをつなぐように組み合わせてプログラミン

グを行う。ここで取り上げたプログラムにおいては、下表のようなブロック（命令）が使用されている。

表4-1 プログラムで使用するブロック

カテゴリ	ブロック	ブロックの機能
見た目	(こんにちは!)と②秒言う	文字列を○秒表示する
	(こんにちは!)と言う	文字列を表示する
イベント	[スペース▼] キーが押されたとき	指定したキーを押すとプログラムが開始される
制御	⑩回繰り返す	指定した回数だけ繰り返す
	ずっと	ずっと繰り返す
	もし◇なら、でなければ	条件によって行う処理を変える
	◇まで待つ	指定した処理が為されるまで待つ
調べる	(What's your name?)と聞いて待つ	質問を表示して答えの入力を待つ
	[スペース▼] キーが押された	指定したキーが押されたか判断する
演算	○+○	加法
	○-○	減法
	○*○	乗法
	○/○	除法
	①から⑩までの乱数	指定した範囲内で乱数を生成する
	○=⑩	等号
	(apple)と(banana)	文字列と文字列(文字列には数も含む)
	(apple)の①番目の文字	文字列の左から○番目の文字を抽出する
	(apple)の長さ	文字列の長さを抽出する
	○の[絶対値▼]	数に対して数学的処理を施す
変数	[変数▼]を⑩にする	指定した変数に数を代入する
リスト	(なにか)を[リスト▼]に追加する	文字や数を指定したリストに追加する
	[リスト▼]のすべて削除する	指定したリストの内容をすべて削除する
	[リスト▼]の①番目	指定したリストの○番目の内容を抽出する

これらのブロックをもとに、児童にどのようなプログラミングの体験をさせることができるかを具体的に検討し、どのようにプログラミング的思考が育成されていくかを、第3章で示した手順にそって考察していく。

5. 人に役立つプログラムを作る

児童の興味関心を引き出し、プログラミングに意欲的に取り組ませるには、人に役立つプログラムを作るという設定が有効である¹³⁾。ここでは、高学年(5・6年)の児童が、中学年(3・4学年)の児童の学習を応援するための学習支援ドリルをスクラッチで作成するという設定にする。具体的な題材としては、小3の算数教科書から【百万の位の数字を問う】問題、小4の算数教科書から【大きな数を万進法で表す】問題と【年数を秒で表す】問題をプログラミング教材として取り上げ、それ

らのプログラムの作成・実行・改善を通して、どのようにプログラミング的思考が育まれるかを検討していく。バージョンアップを目指した改善の箇所には下線を引いた。

[1] 【百万の位の数字を問うプログラム】

小3算数教科書に下記のような問題がある¹⁴⁾。

13637346の、百万の位の数字は何ですか。

この問題がそのまま提示され、入力した答えの正誤が判断されるようなプログラムの作成を意図とする。

(1) 百万の位の数字を問うプログラム1

① 画面の流れをイメージする。

スペースキーを押してプログラムを開始すると、「13637346の百万の位の数字は？」と表示されて、3と入力すると正解、それ以外の場合は不正解というメッセージが表示される。

画面の流れのイメージとブロックの機能を対応させると下表ようになる。

表5-1 イメージと対応した機能

イメージ	ブロックの機能
i 開始する	指定したキーを押すとプログラムが開始される
ii 問題を出す	質問を表示して答えの入力を待つ
iii 正答の場合	等号が成り立つ
iv 正誤判断	条件によって行う処理を変える
v 評価を表示	文字列を表示する

② 必要になる命令を選択する。

- i 「[スペース▼] キーが押されたとき」
- ii 「(What's your name?) と聞いて待つ」
- iii 「○=⑤」
- iv 「もし◇なら、でなければ、」
- v 「(こんにちは!) と言う」

入力された数字が正答と等しいかどうかを判断するので条件分岐の命令を使う。

③ プログラムを組み実行する



図5-1 百万の位の数字を問うプログラム1

スペースキーを押すと、「13637346の百万の位の数字は?」と表示され、3と入力してエンターキーを押すと「正解!」と表示され、それ以外の数を入力してエンターキーを押すと「残念!」と表示される。

改善：このプログラムだと1問しか出題されず、改良したいという意欲が湧く。そこで様々な問題が出題されるようにすることを意図して改善してみる。

(2) 百万の位の数字を問うプログラム2

① 画面の流れをイメージする。

乱数を使って生成された問題が表示され、答えを入力すると正誤を判断してメッセージが表示される。画面の流れのイメージとブロックの機能を対応させると下表ようになる。

表5-2 イメージと対応した機能

イメージ	ブロックの機能
i 開始する	指定したキーを押すとプログラムが開始される
ii 乱数を使用	指定した範囲内で乱数を生成する
iii 変数に代入	指定した変数に数を代入する
iv 問題を出す	質問を表示して答えの入力を待つ
v 数字の抽出	文字列の左から○番目の文字を抽出する
vi 正答の場合	等号が成り立つ
vii 正誤判断	条件によって行う処理を変える
viii 評価を表示	文字列を表示する

② 必要になる命令を選択する。(追加分)

- ii 「①から⑩までの乱数」,
- iii 「(変数) を①にする」
- v 「(apple) の①番目の文字」

乱数で8桁の数を生成させる。左から2番目が百万の位の数字になるのでそれを抽出して、入力した数字と等しいかどうかを判断する。

③ プログラムを組み実行する。



図5-2 百万の位の数字を問うプログラム2

スペースキーを押すと、例えば「96735428の百万の位の数字は?」と表示されるので、6と入力してエンターキーを押すと「大正解」と表示され、

それ以外の数を入力してエンターキーを押すと「残念」と表示される。

改善：このプログラムだと百万の位の数字しか問われない。そこで千万や十万・一万の位の数字を問う問題も出題されるようにすることを意図して改善してみる。

(3) 百万の位の数字を問うプログラム3

- ① 画面の流れをイメージする。
一万から千万までのどれかの位の数字を問う問題が表示されて、答を入力すると正誤を判断してメッセージが表示される。
- ② 必要になる命令を選択する。(追加分)

「(apple) と (banana)」

文字列「千百十一」の左から何番目かの文字を乱数を使って選ぶ。誤答の場合は正答を表示するようにする。

- ③ プログラムを組み実行する。

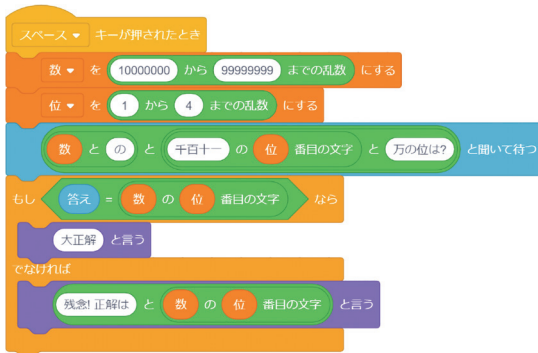


図5-3 百万の位の数字を問うプログラム3

スペースキーを押すと、例えば「56028982の千万の位は？」と表示されるので、5と入力してエンターキーを押すと「大正解」と表示され、それ以外の数を入力してエンターキーを押すと「残念! 正解は5」と表示される。

改善：このプログラムだとスペースキーを押すごとに1題出題されるだけである。連続して出題されて最後に得点が出るようにしてモチベーションアップにつなげることを意図して改善を図ってみる。

(4) 百万の位の数字を問うプログラム4

- ① 画面の流れをイメージする。
問題が連続して5題出題され、正答数がカウントされて最後に得点が表示される。
- ② 必要になる命令を選択する。(追加分)

「⑩回繰り返す」「(こんにちは!) と②秒言う」

問題を出すという動作を繰り返させる。正誤を判断して、正答数をカウントした後、得点計算を行う。

- ③ プログラムを組み実行する。

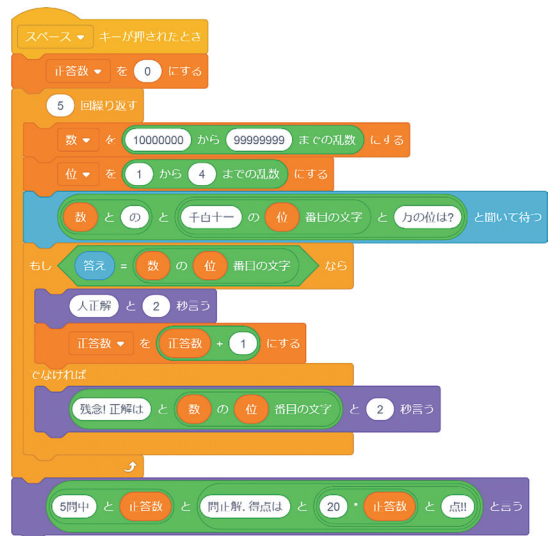


図5-4 百万の位の数字を問うプログラム4

スペースキーを押すと、例えば「29718508の十万の位は？」と表示され、7と入力すると「大正解」と2秒表示され、誤答だと「残念! 正解は7」と2秒表示される。このような問題が5問出題されて解答が終わると、最後に例えば「5問中4問正解. 得点は80点!!」と得点が表示される。

[2] 【大きな数を万進法で表すプログラム】

小4算数教科書に下記のような問題がある¹⁵⁾。

次の①, ②の数を読みましょう。

- ① 101457093570000(円)…日本の国の予算(平成31年度)
- ② 9460000000000(km)…光が1年間に進むきょり

万進法は4桁ごとに区切って読むので、例えば、101|4570|9357|0000と区切って101兆4570億9357万のように万進法表示することができる。ここでは、このように大きな数を入力すると、その数を万進法で表すプログラムの作成を意図とする。

(1) 大きな数を万進法で表すプログラム1

- ① 画面の流れをイメージする。
乱数を使って生成された一兆以上百兆以下の数が表示され、児童がその数を読み上げた後にキーを押すと、その数が万進法表示される。
- ② 必要になる命令を選択する。

「[スペース▼] キーが押されたとき」
 「①から⑩までの乱数」
 「[リスト▼] のすべてを削除する」
 「(変数) を①にする」「○-○」「○*○」「○/○」
 「○の [絶対値▼]」
 「(なにか) を [リスト▼] に追加する」
 「⑩回繰り返す」「(こんにちは!) と言う」
 「(スペース▼) キーが押された」「◇まで待つ」
 「[リスト▼] の①番目」「○と○」

数を4桁区切りにして、万進法に対応させる形でリストに格納する。

- ③ プログラムを組み実行する。

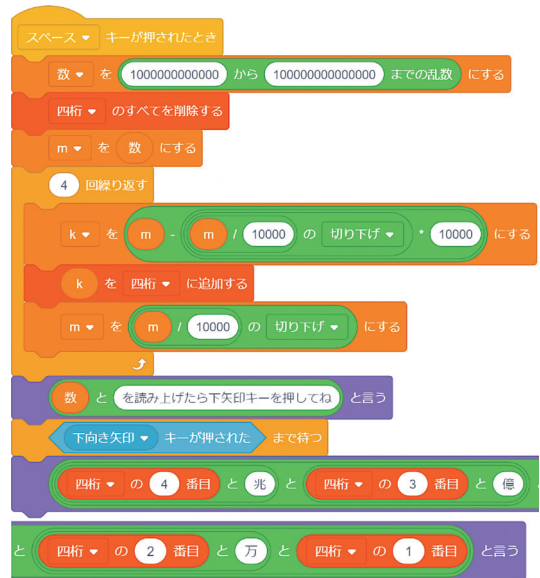


図5-5 大きな数を万進法で表すプログラム1

スペースキーを押すと、例えば「61858461090640」を読み上げたら下矢印キーを押してね」と表示されるので、口頭で読み上げてから下矢印キーを押すと「61兆8584億6109万640」と表示される。
改善：このプログラムだと出題される数は一兆以上百兆以下の限定された範囲の数になるので、読み上げる数がパターン化してしまう。そこで数の範囲を千以上十兆以下にして様々な問題が出題されるようにすることを意図して改善してみる。

(2) 大きな数を万進法で表すプログラム2

- ① 画面の流れをイメージする。
乱数を使って生成された千以上十兆以下の数が表示され、児童がその数を読み上げた後にキーを押すと、その数が万進法表示される。
- ② 必要になる命令を選択する。(追加分)

「(apple) の長さ」

様々な数の読みに対応させるため、万や億・兆を格納したリストを用意する。

- ③ プログラムを組み実行する。



図5-6 大きな数を万進法で表すプログラム2

スペースキーを押すと、例えば「829035443を読み上げたら下向きの矢印キーを押してね」と表示され、口頭で読み上げてから下矢印キーを押すと「8億」「2903万」「5443」とそれぞれ2秒ずつ続けて表示される。

[3] 【年数を秒で表すプログラム】

小4算数教科書に下記のような問題がある¹⁶⁾。

長い時間を、秒の単位で表してみましょう。答えを数字で書いて、その数を読みみましょう。(1年を365日とするよ。)

- ① 1年は何秒ですか。
- ② 人の一生を80年とします。80年は何秒ですか。
- ③ 地球に人類があらわれたのは、今から700万年ぐらい前と考えられています。700万年は何秒ですか。

年数を入力すると秒換算した値が表示されるプログラムの作成を意図する。

(1) 年数を秒で表すプログラム1

- ① 画面の流れをイメージする。

1年は $60 \times 60 \times 24 \times 365$ から31536000秒であること用いて、それに入力した年数をかけた値が表示される。

- ② 必要になる命令を選択する。

「[スペース▼] キーが押されたとき」
 「(What's your name?) と聞いて待つ」「○*○」
 「(変数)を①にする」「○と○」
 「(こんにちは!) と言う」

- ③ プログラムを組み実行する。



図5-7 年数を秒で表すプログラム1

スペースキーを押すと、「何年？」と表示されるので、例えば80と入力すると、「80年は2522880000秒」と表示される。

改善：このプログラムだと秒換算されて表示された数の桁数が多く、どのくらいの大きさなのかわかりにくい。そこでおよそどのくらいの大きさの数のかわかるようにすることを意図して改善を図ってみる。

(2) 年数を秒で表すプログラム2

- ① 画面の流れをイメージする。

年数を秒換算した数が表示された後、その数の頭の方を万進法表示しておおよその大きさが示される。

- ② 必要になる命令を選択する。(追加分)

「[リスト▼] のすべてを削除する」
 「(apple)の長さ」「○/○」「○の[絶対値▼]」
 「⑩回繰り返す」
 「(なにか)を[リスト▼]に追加する」

数を4桁区切りにして格納するリストと、「万」「億」「兆」などを格納したリストを用意する。

③ プログラムを組み実行する。

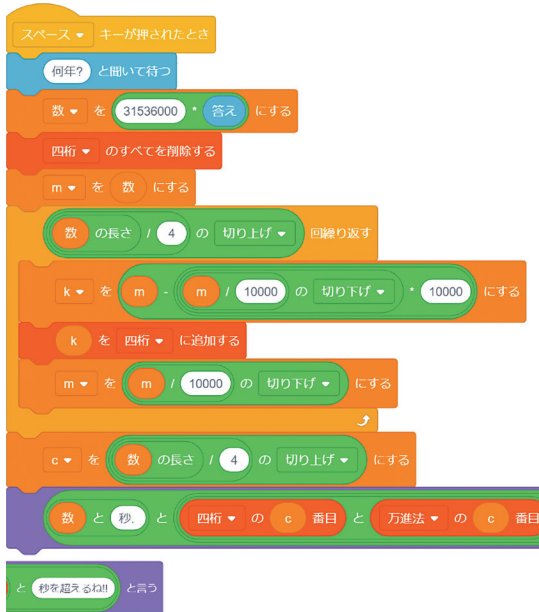


図5-8 年数を秒で表すプログラム2

スペースキーを押すと、「何年？」と表示されるので、例えば7000000（700万）と入力してエンターキーを押すと、「220752000000000秒. 220兆秒を超えるね!!」と表示される。また「四桁」と「万進法」の2つのリストを対応させて見ることによって万進法での表現を読み取ることができる。

6. 探究活動に活かすプログラム

ここでは、高学年（5・6年）の児童自身が、計算結果がどうなるか調べたい、条件を変えたときの変化の様子を調べたいといったような探究活動を行うためにプログラムを作成するという設定にする。

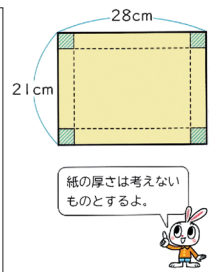
[1] 【箱の容積を求めるプログラム】

小5算数教科書に下記のような問題がある¹⁷⁾。

右の図のような、長方形の厚紙の四すみを切り取って、ふたのない箱を作ります。

① 四すみを3 cmずつ切り取ったとき、できあがる箱の容積を求めましょう。

② 箱の容積が一番大きくなるのは、四すみを何cmずつ切り取ったときか調べましょう。ただし、切り取る長さはcm単位で整数とします。



縦21 cm横28 cmの長方形の四隅から切り抜く正方形の一辺の長さ (cm) を決めると、そのとき箱の容積の値 (cm³) が算出されるプログラムの作成を意図する。

(1) 箱の容積を求めるプログラム1

- ① 画面の流れをイメージする。
切り抜く正方形の一辺の長さを入力すると、そのときの箱の容積の値が表示される。
- ② 必要になる命令を選択する。

「[スペース▼] キーが押されたとき」
 「(What's your name?) と聞いて待つ」
 「(変数) を⑩にする」
 「○と○」「○*○」「○-○」
 「(こんにちは!) と言う」

③ プログラムを組み実行する。



図6-1 箱の容積を求めるプログラム1

スペースキーを押すと、「縦21 cm横28 cmの長方形の四隅から切り抜く正方形の一辺の長さ何cm?」と表示され、例えば3と入力してエンターキーを押すと、「容積は990立方センチメートル」

と表示される。

改善：このプログラムだと長方形の大きさが特定されているので、実際にいろいろな大きさの長方形をもとにして箱を作ろうとするときには役に立たない。そこで任意の大きさの長方形の場合にも対応できるようにし、更に切り抜く正方形の一边の長さを何回も入力して、そのときどきの容積の値を調べることができるようにすることを意図して改善してみる。

(2) 箱の容積を求めるプログラム2

① 画面の流れをイメージする。

長方形の縦と横の長さ、切り抜く正方形の一边の長さを入力すると、そのときの箱の容積の値が表示され、再び正方形の一边の長さの入力が促される。

② 必要になる命令を選択する。(追加分)

「[スペース▼] キーが押された」「◇まで待つ」
「ずっと」

スペースキーを押すごとに切り抜く正方形の一边の長さを問うようにする。

③ プログラムを組み実行する。

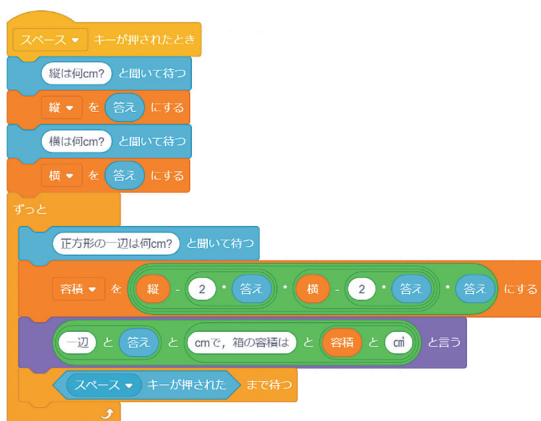


図6-2 箱の容積を求めるプログラム2

スペースキーを押すと、「縦は何cm?」と表示されるので任意の値を入力しエンターキーを押すと、続いて「横は何cm?」と表示される。再び任意の数を入力してエンターキーを押すと、「正方形の

一边の長さは何cm?」と表示されるので、任意の値（縦横の短い方の長さの半分より小さい値）を入力してエンターキーを押すと、その時の箱の容積 (cm³) が表示され、更にスペースキーを押すと、再び正方形の一边の長さの入力が促される。いろいろな長さを入力することによって、容積の変化の様子を調べることができる。

改善：このプログラムだと、箱の容積の変化の様子を見るためには、スペースキーを押して新たに正方形の一边の長さを入力するという作業を何度も繰り返さなくてはならない。そこで容積の変化が一覧できるリストが表示されるようにすることを意図として改善してみる。

(3) 箱の容積を求めるプログラム3

① 画面の流れをイメージする。

長方形の縦と横の長さを入力すると、切り抜く正方形の長さ1cm刻みに対応した箱の容積の値が一覧できるリストが表示される。

② 必要になる命令を選択する。(追加分)

「[リスト▼] のすべてを削除する」「○/○」
「○の [絶対値▼]」「@回繰り返す」
「(なにか) を [リスト▼] に追加する」

切り抜く正方形の一边の長さを1cmずつ変化させたときの容積の値をリストに格納する。

③ プログラムを組み実行する。

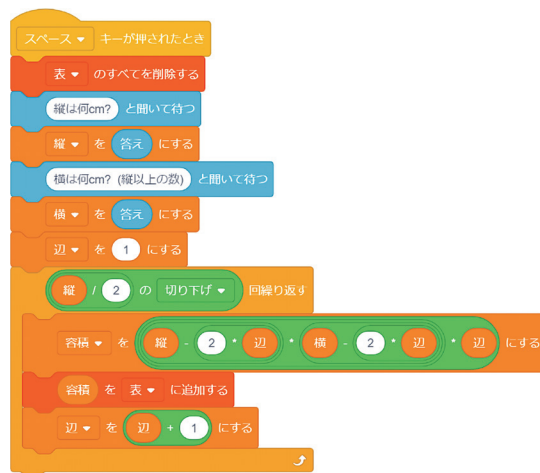


図6-3 箱の容積を求めるプログラム3

スペースキーを押すと、「縦は何cm?」と表示されるので、例えば21と入力してエンターキーを押すと、続けて「横は何cm? (縦以上の数)」と表示されるので、例えば28と入力してエンターキーを押すと、下図のような箱の容積が一览できる「表」(リスト)が表示される。切り抜く正方形の一辺の長さ (cm) がリストの番号に対応している。このリストから容積の変化の様子が把握できる¹⁸⁾。



図6-4 箱の容積を求めるプログラム3のリスト

このような長方形の厚紙から作る箱に関する問題は、小中高を通して扱われている¹⁹⁾。

7. 使用したブロックとプログラムの構造

表7-1は、ここまでのそれぞれのプログラムにおいて使用したブロック(命令)の一覧である。「百1」は「百万の位の数字を問うプログラム1」を、「百2」は「百万の位の数字を問うプログラム2」を表している。また「万1」「秒1」「箱1」はそれぞれ「大きな数を万進法で表すプログラム1」「年数を秒で表すプログラム1」「箱の容積を求めるプログラム1」を表している。11のプログラムにおいて、計23のブロックが使われている。

表7-1 それぞれのプログラムで使用したブロック

カテゴリ	ブロック	百1	百2	百3	百4	万1	万2	秒1	秒2	箱1	箱2	箱3
見た目	(こんにちは!)と②秒言う				●		●					
	(こんにちは!)と言う	●	●	●	●	●	●			●	●	
イベント	[スペース▼] キーが押されたとき	●	●	●	●	●	●	●	●	●	●	●
制御	⑩回繰り返す				●	●	●		●			●
	ずっと										●	
	もし◇なら、でなければ	●	●	●	●							
	◇まで待つ					●	●				●	
調べる	(What's your name?)と聞いて待つ	●	●	●	●			●	●	●	●	●
	[スペース▼] キーが押された					●	●				●	
演算	○+○				●							●
	○-○				●	●	●	●	●	●	●	●
	○*○				●	●	●	●	●	●	●	●
	○/○					●	●		●			●
	①から⑩までの乱数		●	●	●	●	●					
	○=⑤⑩	●	●	●	●							
	(apple)と(banana)		●	●	●	●	●	●	●	●	●	
	(apple)の①番目の文字		●	●	●							
	(apple)の長さ						●		●			
	○の[絶対値▼]					●	●		●			●
変数	[変数▼]を⑩にする		●	●	●	●	●	●	●	●	●	●
	(なにか)を[リスト▼]に追加する					●	●		●			●
	[リスト▼]のすべて削除する					●	●		●			●
	[リスト▼]の①番目					●	●					

プログラムは順次処理がなされ、必要に応じて分岐処理や反復処理が組み込まれているが、基本要素をもとにプログラムの構造を分類すると表7-2のようになる。

表7-2 基本要素とプログラム

順次	秒1、箱1
順次+分岐	百1、百2、百3
順次+分岐+反復	百4
順次+反復	万1、万2、秒2、箱2、箱3

8. プログラム作成と数学的内容

ここでは、プログラムの作成過程においてどのような数学的内容が関連付けられ活用されていくかを、特にバージョンアップを目指した改善にかかわる箇所を中心に考察する。

[1] 【百万の位の数字を問うプログラム】

「百1」では、特定の数の場合における百万の位の数字を問い、入力された数字と正答との間に等号が成り立つかどうかをもとにして正誤判断を行っている。

(1) 「百1」から「百2」の改善においては「様々な問題が出題されるようにする」ことを意図している。そのための方法として、乱数を生成させるという数学的手法を用いている。「百2」では、出題範囲を拡張しており、「百2」は「百1」を一般化したものと捉えることができる。

(2) 「百2」から「百3」の改善においては「千万や十万・一万の位の数字を問う問題も出題されるようにする」ことを意図している。「百2」では、乱数で生成した8桁の数の左から2番目の数字を百万の位の数として抽出しているが、「百3」では更に千万や十万・一万の位の数も抽出することができるように抽出範囲を拡張しており、「百3」は「百2」をより一般化したものになっている。位の数の抽出は十進位取り記数法の知識に基づいている。

(3) 「百3」から「百4」の改善においては「連続して出題されて最後に得点が出るようにしてモチベーションアップにつなげる」ことを意図している。「百4」では、正答数をカウント（1ずつ加算）

して、配点に正答数を乗じて得点を算出している。

「百1」～「百4」における主な実行内容とそれに対応した数学の内容を表8-1に、算数教科書にみる主な関連内容を表8-2に示した。

表8-1 百万の位の数字を問うプログラムと数学

実行内容	数学の内容	プログラム
正誤判断	等号・命題の真偽	百1、百2、百3、百4
位の数抽出	十進位取り記数法	百1、百2、百3、百4
変数の利用	文字式・代入	百2、百3、百4
乱数の利用	乱数の生成	百2、百3、百4
カウント	1ずつ加算	百4
得点計算	乗法計算	百4

表8-2 算数教科書における主な関連内容

小3	1億までの数の読み方、書き方
小4	億、兆に及ぶ数の構成 命数法と記数法 十進位取り記数法と十進数の意味

[2] 【大きな数を万進法で表すプログラム】

「万1」では、大きな数を4桁ごとに区切り、万進法で表現している。4桁区切りにして数を抽出するために、10000で割って小数点以下を切り捨てる²⁰⁾という作業を組み込んでいる。

(1) 「万1」から「万2」の改善においては「数の範囲を千以上十兆以下にして様々な問題が出題されるようにする」ことを意図している。「万2」では、乱数で生成される数の範囲を拡張しており、生成された様々な数の大きさに合わせて4桁区切りで数を抽出してリストに格納するために、その数の桁数を4で割って小数点以下を切り上げる²⁰⁾という作業を組み込んでいる。また万・億・兆という万進法の単位も別のリストに格納しておき、2つのリストを対応させるために変数のカウント（1ずつ減算）を行って、リストの対応関係をもとにして万進法表示している。「万2」では、出題範囲を拡張しており、「万2」は「万1」をより一般化したものになっている。

「万1」「万2」における主な実行内容とそれに対応した数学の内容を表8-3に、算数教科書にみる主な関連内容は表8-4に示した。

表8-3 大きな数を万進法で表すプログラムと数学

実行内容	数学の内容	プログラム
位の数抽出	十進位取り記数法	万1、万2
4桁を抽出	命数法・万進法	万1、万2
変数の利用	文字式・代入	万1、万2
乱数の利用	乱数の生成	万1、万2
演算	減法・除法・乗法計算	万1、万2
端数処理	切り捨て・切り上げ	万1、万2
カウント	1ずつ減算	万2
リスト作成	対応関係	万2

表8-4 算数教科書における主な関連内容

小3	1億までの数の読み方、書き方
小4	億、兆に及ぶ数の構成 命数法と記数法 十進位取り記数法と十進数の意味 概数処理の種々の表現方法 2つの数量の対応関係を表に表す

[3] 【年数を秒で表すプログラム】

「秒1」では、1分=60秒²¹⁾をもとに時間の単位換算を行っている。

(1) 「秒1」から「秒2」の改善においては「およそどのくらいの大きさの数なのかわかるようにする」ことを意図している。「秒2」では、算出した数のおよその大きさを把握するために、4桁区切りで抽出した大きい位の部分を万進法表示している。また「秒2」における4桁区切りで抽出してリストに格納するループ構造の部分は、「万2」からの転用であり、「秒2」の作成に「万2」での学びが関連付けられ活かされている。

「秒1」「秒2」における主な実行内容とそれに対応した数学の内容を表8-5に、算数教科書にみる主な関連内容を表8-6に示した。

表8-5 年数を秒で表すプログラムと数学

実行内容	数学の内容	プログラム
単位換算	時間の単位・秒換算	秒1、秒2
演算	減法・除法・乗法計算	秒1、秒2
変数の利用	文字式・代入	秒1、秒2
端数処理	切り捨て・切り上げ	秒2
位の数抽出	十進位取り記数法	秒2
4桁を抽出	命数法・万進法	秒2
大きさ把握	概数	秒2
リスト作成	対応関係	秒2

表8-6 算数教科書における主な関連内容

小3	時間の単位(秒)と単位関係
小4	億、兆におよぶ数の構成 命数法と記数法 概数の意味 概数処理の種々の表現方法 2つの数量の対応関係を表に表す

[4] 【箱の容積を求めるプログラム】

「箱1」では、長方形の厚紙から容器の展開図を切り抜き、直方体の体積公式²²⁾を用いて容積を立方センチメートル²³⁾の単位で算出している。

(1) 「箱1」から「箱2」の改善においては「任意の大きさの長方形の場合にも対応できるようにし、更に切り抜く正方形の一辺の長さを何度も入力して、そのときどきの容積の値を調べることができるようにする」ことを意図している。「箱1」では特定の長方形であったが「箱2」では任意の長方形となっており、「箱2」は「箱1」を一般化したものになっている。「箱2」では、切り抜く正方形の一辺の長さを独立変数と見て、その定義域における箱の容積(従属変数)の値を算出しており、関数の考え方をもとにその対応関係を調べている。

(2) 「箱2」から「箱3」の改善においては「容積の変化が一覧できるリストが表示されるようにする」ことを意図している。「箱3」では、リストを関数における対応表と見立てており、切り抜く正方形の一辺の長さ1cm刻みに対応した箱の容積の変化の様子をリストから読み取ることができる。

「箱1」～「箱3」における主な実行内容とそれに対応した数学の内容を表8-7に、算数教科書にみる主な関連内容を表8-8に示した。

表8-7 箱の容積を求めるプログラムと数学

実行内容	数学の内容	プログラム
容積算出	直方体の体積	箱1、箱2、箱3
容積表示	体積の単位	箱1、箱2、箱3
演算	減法・乗法計算	箱1、箱2、箱3
変数の利用	文字式・代入	箱1、箱2、箱3
変化の確認	関数、対応関係	箱2、箱3
切り取る	独立変数、定義域	箱3
端数処理	切り捨て	箱3
カウント	1ずつ加算	箱3
リスト表示	対応表、従属変数、値域	箱3

表8-8 算数教科書における主な関連内容

小4	直方体、立方体の概念、特徴、性質 展開図、見取図の意味と書き方 2つの数量の対応関係を表に表す
小5	体積の意味と単位（立方センチメートル） 直方体、立方体の体積公式とその適用

9. まとめ

本研究では、教科書の内容をもとにして、児童にどのようなプログラミングの体験をさせることができるかを検討し、プログラムの改善を通してどのようにプログラミング的思考が育成され、数学的内容が関連付けられ活用されていくかについて探究した。算数教科書の内容を題材にした具体例として次のようなプログラミング教材を示した。

【百万の位の数字を問うプログラム】

【大きな数を万進法で表すプログラム】

【年数を秒で表すプログラム】

【箱の容積を求めるプログラム】

これらのプログラム作成過程で、どのようにプログラミング的思考が育まれていくかを、手順（意図・イメージ・選択・プログラム・実行・改善）にそって具体的に考察し、改善に向けた試行錯誤を繰り返すことよってプログラミング的思考が深められていくことを、それぞれのプログラムにおいて明らかにした。またプログラムの実行内容と数学の内容がどのように関連付けられているかを考察し、どのような数学的知識や数学的手法が活用されているかを、プログラムに即して明らかにした。

今後、様々な学年・教科・単元などで創意工夫を生かしたプログラミング教材の開発がますます期待されてくると予想される。算数のプログラミング教材の開発も、まだ端緒に着いたばかりである。ここで示したように、算数教科書の中から題材を見出すことができ、高々10数行程度のプログラムでも相当のことが実行できるという知見を踏まえ、更にどのような題材がプログラミング教材として開発可能か、どのような数学的知識や数学的手法に関する学びを深めることができるか、探究していきたいと考えている。

註

- 1) 2019年に開始された文部科学省の取り組みで、高速ネットワークを整備し、全国の児童・生徒に一人一台のコンピュータを用意するという構想である。
- 2) 視覚的なオブジェクト（ブロック）の操作で容易にプログラムが組めるプログラミング言語である。
- 3) スクラッチ（Scratch）は、マサチューセッツ工科大学の Lifelong kindergarten グループが開発した教育用ビジュアルプログラミング言語である。ブラウザで Scratch のサイト (<https://scratch.mit.edu/>) に接続して、「作る」をクリックすることで、プログラム作成画面を表示させることができる。本稿では、2019年1月に公開された Scratch 3.0 を使用している。
- 4) 「小学校学習指導要領解説総則編」（文部科学省、2017年）p.85 において「プログラミング的思考」は次のように定義されている。「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」
- 5) 筆者はこれまで、ビジュアルプログラミング言語によって児童にどのようなプログラミングの体験をさせることができるか、また教師がプログラミングによってどのような教材を開発することができるかについて考察してきた（中込・黒木 2022）。帰納的な発見をもとにした探究活動などの場面において、プログラミングが効果的に活用できることを明らかにし、次のような知見を得ている。プログラミングにより①探究活動（帰納的に発見する、変化を調べる、検証する）が可能となる②創作活動が可能となる③算数の学びを深めることができる④コンピュータの捉え方がかわる⑤算数教材を開発することができる。これらを踏まえて、本稿では算数教科書に掲載されている問題を題材として取り上げ、実際にプログラムの作成を通してどのようにプログラミング的思考が生まれ、数学的内容が関連付けられ活用されていくかを探究している。
- 6) 「小学校プログラミング教育の手引き（第三版）」（文部科学省、2020年）p.13 の「*5」参照。

- 7) 算数教科書小1下、東京書籍、H27 発行、p.18
- 8) 繰り下がりのあるひき算では、他に減々法という方法もある。
- 9) 算数教科書小2上、東京書籍、R2 発行、p.93
- 10) 「小学校プログラミング教育の手引（第三版）」（文部科学省、2020年）p.14には「コンピュータを動作させるための手順（例）」として次の手順が示されている。「①コンピュータにどのような動きをさせたいのかという自らの意図を明確にする②コンピュータにどのような動きをどのような順序でさせればよいのかを考える③一つ一つの動きを対応する命令（記号）に置き換える④これらの命令（記号）をどのように組み合わせれば自分が考える動作を実現できるかを考える⑤その命令（記号）の組合せをどのように改善すれば自分が考える動作により近づいていくのかを試行錯誤しながら考える。」
- 11) 例えば東京書籍の算数教科書では、「倍数を求める手順を考えよう」小5上 p.126、「正多角形をかく手順を考えよう」小5下 p.134、「数の並び方を考えよう」小6 pp.242-243 が、啓林館の算数教科書では、「正多角形の作図の仕方」小5 pp.236-237、「条件にあう整数のみつけ方」小6 pp.182-183 などが掲載されている。
- 12) 「小学校プログラミング教育の手引き（第三版）」（文部科学省、2020年）p.2に次のような記述がある。「本手引を参考にして、学習指導要領に例示された教科・学年・単元等に限定することなく、適切なカリキュラム・マネジメントの下で、各学校の創意工夫を生かしたプログラミング教育が展開されることが期待されます。」
- 13) プログラミング言語（BASIC）を用いて高校生に小学生向けの「基礎学力強化用ドリル」を作成させた経験などから、プログラミングに意欲的に取り組ませるには人に役立つプログラムを作るという設定が有効であるという知見を得た（中込 1996）。
- 14) 算数教科書小3上、東京書籍、R2 発行、p.87
13637346 は 2018 年の東京都の人口を表している。
- 15) 算数教科書小4上、東京書籍、R2 発行、p.12
- 16) 算数教科書小4上、東京書籍、R2 発行、p.150
- 17) 算数教科書小5、大日本図書、R2 発行、p.64

- 箱の容積に関しては他の教科書でも扱われている。
- 算数教科書小5上、東京書籍、R2 発行、p.143
- 算数教科書小5上、日本文教出版、R2 発行、p.135
- 18) 箱の容積を求めるプログラムは、切り抜く正方形の長さを更に詳細に求める次のようなものも考えられる。

```

when green flag clicked
  when space key pressed
    delete from list
    ask "縦は何cm?" and wait for answer
    set height to answer
    ask "横は何cm?(縦以上の数)" and wait for answer
    set width to answer
    set x to 1
    set p to (height / 2) rounded down
    set q to 1

    loop
      ask "何cmあたりを調べる?" and wait for answer
      set x to answer - 0.2
      set p to 5
      set q to 0.1
      delete from list
      loop
        delete all squares of side length x
        delete all volumes of value x
      loop
        set n to 1
        loop p times
          insert x into list n times of square side length
          set volume to (height - 2 * x) * (width - 2 * x) * x
          insert volume into list n times of volume value
          set n to n + 1
          set x to x + q
    loop
  
```

図6-5 箱の容積を求めるプログラム4

スペースキーを押すと、「縦は何 cm?」「横は何 cm?（縦以上の数）」と表示されるので、それぞれに対し

例えば B5 判の長方形を想定して 18.2、25.7 と入力してエンターキーを押すと、正方形の一边の長さ 1 cm 刻みに対応した箱の容積がリストで表示される(図 6-6)。

正方形の一边		容積の値	
1	1	1	383.94
2	2	2	616.28
3	3	3	721.02
4	4	4	722.16
5	5	5	643.6...
6	6	6	509.6...
+ 長さ 9 =		+ 長さ 9 =	

図 6-6 箱の容積を求めるプログラム 4 のリスト

続いて「何 cm あたりを調べる?」と表示されるで、例えば 3.5 と入力すると、正方形の一边の長さを 3.3 cm から 3.7 cm まで 0.1 cm 刻みにしたときの、それぞれに対応した箱の容積がリストで表示される(図 6-7)。

正方形の一边		容積の値	
1	3.3	1	731.148
2	3.4	2	732.5639...
3	3.5	3	733.0399...
4	3.6	4	732.6
5	3.7	5	731.2679...
+ 長さ 5 =		+ 長さ 5 =	

図 6-7 箱の容積を求めるプログラム 4 のリスト

- 19) 小学校算数教科書と高校数学教科書では箱の容積に関する問題が、中学校数学教科書では箱の表面積に関する問題が扱われている。ここで取り上げた箱の容積の問題を高校の範囲で解くと次のようになる。四隅から切り抜く正方形の一边の長さを x とすると、容積 V は 3 次関数として次のように表すことができる。 $V = (21 - 2x)(28 - 2x)x$, $0 < x < 10.5$
 V を x で微分して、 $V' = 0$ となる x の値を α とすると、次のような増減表ができる。

表 6-1 増減表

x	0		α		10.5
V'		+	0	-	
V	0	↗	最大値	↘	0

$\alpha \approx 3.96$ であることから、正方形の一边の長さが約 3.96 cm のときに容積 V は最大となることがわかる。

- 20) 「切り捨て」「切り上げ」は小 4 の学習内容である。スクラッチでは「切り捨て」を「切り下げ」と表す。
 算数教科書小 4 上、東京書籍、R2 発行、p.127
 算数教科書小 4 上、学校図書、R2 発行、p.129
 21) 算数教科書小 3 上、東京書籍、R2 発行、p.32
 22) 算数教科書小 5 上、東京書籍、R2 発行、p.20
 23) 算数教科書小 5 上、東京書籍、R2 発行、p.18

参考文献

- 中込雄治、「小型コンピュータの活用で変化する指導形態について—ポケコンやグラフ電卓が数学の授業に及ぼす効用—」、日本数学教育学会誌第 78 巻第 7 号数学教育 50-4 pp.9-13、1996 年
 驚崎弘宣他、「Scratch でのたのしく学ぶプログラミング的思考」、マイナビ出版、2019 年
 梅田恭子他、「ICT 活用指導力アップ!」、実教出版、2019 年
 中込雄治・黒木伸明・千田真佑子、「算数・数学教育におけるテクノロジーの活用場面について」、宮城学院女子大学発達科学研究 No. 20 pp.7-18、2020 年
 守屋誠司他、「教育現場で役立つ情報リテラシー」、実教出版、2020 年
 文部科学省、「小学校プログラミング教育の手引き(第三版)」、2020 年
 TENTO & できるシリーズ編集部、「できるキッズ子どもと学ぶ Scratch3 プログラミング」、インプレス、2020 年
 アンク、「Scratch の絵本」、翔泳社、2020 年
 石原正雄他、「スクラッチ 3.0 ワークブック」、カットシステム、2021 年
 須曾野仁志・大野恵理・萩野真紀・榎本和能、「小・中学校における STEAM 教育を実現するスクラッチプログラミング学習」、三重大学教育学部研究紀要第 74 巻第 1 号教育実践 pp.151-158、2022 年

中込雄治・黒木伸明、「算数教材とプログラミング教育について」、宮城学院女子大学発達科学研究 No. 22 pp.77-87、2022 年