

研究論文

# 小学校教員養成課程の大学生にさせた プログラミング体験とその結果からの プログラミング教育についての考察

穴 田 恭 輔

## 1. はじめに

2020(令和2)年4月から小学校においてプログラミング教育が必修化された。そこに至るまでには、

- 2016(平成28)年6月に各界の専門家が分野を越えて知見を持ち寄る有識者会議でなされた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の議論の取りまとめ<sup>[1]</sup>
- これを議論の土台として小学校プログラミング教育の導入を検討し、同年12月21日に出された中央教育審議会の答申<sup>[2]</sup>
- これら有識者会議の議論、中央教育審議会の答申を踏まえ、プログラミング学習の必修が示された翌年2017(平成29)年3月31日に告示された小学校学習指導要領<sup>[3]</sup>

という経緯がある。

さらに、文部科学省は、2018(平成30)年3月に「小学校プログラミング教育の手引」を公表し、11月に改訂(第二版)、さらに2020(令和2)年2月に改訂(第三版)を行なって、全国の小学校における円滑なプログラミング教育の実施を推進している<sup>[4]</sup>。

これらの経緯を踏まえ全国の小学校では、必修化される前に先行的にプログラミング教育に取り組むところもある。その一方で、プログラミング教育が必修化されても、それが、プログラミングという教科ではないので、具体的に何を教えるのか、何を目標にするのか、どうやって教えるのかなどの問題に行き詰まり、なかなか取り組めていない現状もある。そして、小学校教員養成課程に在学中の大学生にとっても、プログラミングをあまり体験したことがないのでプログラミング教育に不安を感じている者もいる。

本研究では、小学校教員養成課程に在学する大学生にプログラミング体験をさせ、その結果をもとに小学校におけるプログラミング教育について考えてみたい。

## 2. 本研究の目的とその方法

### 2.1. 問題の設定

『小学校学習指導要領(平成29年告示)解説 総則編』<sup>[5]</sup>によると、小学校段階において学習活動としてプログラミングに取り組むそのねらいは、

- ①「プログラミング的思考」を育むこと
- ② プログラムやコンピュータ等の情報技術を上手に活用する態度を育むこと

③ 各教科等で学ぶ知識及び技能をより確実にすることである。

1つ目にあげられる「プログラミング的思考」の育成は、プログラミング教育によって思考力、判断力、表現力等、つまり、考える力をつけるということである。それは自分が意図する一連の活動を実現するために動きの組み合わせを構成していくプログラミングという形式から、その形式を使いこなすことによって考える力が育成されるという形式陶冶である。形式陶冶ならそれは直接的に示された目標ではなく、極端な言い方をすれば、プログラミングを通して知らず知らずのうちに頭が鍛錬されて頭がよくなると言っているようなものである。もちろんその効果は誰もがなんとなく思い当たるふしもあり一般的に否定はされないが、それを「プログラミング的思考」と表現するならば必ずしも明確なものではない。

2つ目のねらいであるプログラムやコンピュータ等の情報技術を上手に活用する態度を育むことは、そのよさを体験して実感することから始まる。つまり上手に活用する態度は、まず活用していくことから生まれるので活用できる環境を整備する必要がある。

そして3つ目のねらいである各教科等で学ぶ知識及び技能をより確実にすることについては、各教科の学びを確実にすることとプログラミングとの関係において、プログラミングと具体的な学びの内容を関係づけなければならない。これは、「プログラミング的思考」と表現するだけですむ1つ目のねらいよりも難しい課題と考えられる。

## 2. 2. 本研究の目的

そこで、本研究の目的は、

- 小学校で習うプログラミングとはどのようなものなのか
- それによって培われる「プログラミング的思考」とは何か
- そのために児童はプログラミング教育で何をどこまで学ぶ必要があるのか
- 小学校教員のプログラミングスキルは必要なのか、必要ならそれはどこまでか
- 教科の深い学びにつながるプログラミングとはどのようなものであるか（その効用）

について考察し、ある程度の答えを出すこととする。

## 2. 3. 本件研究の方法

本研究では、実際に小学校教員養成課程で学ぶ大学生を対象に行ったプログラミング教育とその成果を考察の材料とする。具体的には、

- (1) 少人数を対象とした初めてのプログラミング体験
- (2) (1) でプログラミングに興味と関心を示した学生へのプログラミング指導とその成果
- (3) 小学校教諭免許状取得見込みの4年次生へのプログラミング指導と意識調査からの考察を行う。

これらの体験・指導に用いたプログラミング言語は、Python, Scratch で、コンピュータを使わ

小学校教員養成課程の大学生にさせたプログラミング体験とその結果からのプログラミング教育についての考察  
ないアンブラグドプログラミングについても体験をした。以下にそれぞれの言語の特徴やその選択を行  
った理由や考えを簡潔に述べる。

- Python

初めてプログラミングを学ぶ初心者にとって最初に習うプログラミング言語は重要である。それは  
その後別の言語も扱うことになる時、最初に修得したプログラミング言語が基礎となるからであ  
る。したがって今回、以下の理由から学習するプログラミング言語に Python を選んだ。

Python はインタプリタ型のプログラミング言語で、作成したソースプログラムをただちに実行で  
きる。初心者にとって、プログラミング作業にかかる負荷はコンパイラ型言語と比較しても小さく、  
それはメリットと考えられる。またシンプルな文法で比較的簡単で始めやすいといわれており、そ  
の後作業効率化のために修得したプログラムスキルを利用することを考えれば、最初に学ぶ言語の有  
候補といえる。そして今話題の人工知能 (AI)・機械学習系の開発をはじめ Web アプリの開発にも  
使われており、プログラムエンジニア求人と言語別に見た場合、Python 修得者は上位にあり需要も  
高い。

- Scratch

Scratch はインタプリタ型のプログラミング言語及びその開発環境である。コーディング方法は、  
テキストだけで記述するのではなく、あらかじめコードがテンプレートで用意されているビジュアル  
コーディングである。テンプレート (ブロック) をはめ込んで記述するので、そのはめ込みができな  
いときは、構文規則を満たしていないということなので、そもそもシンタックスエラーは起こらない。  
その分、プログラミング初習者にとって敷居は低いともいえる。

そして Scratch はプログラミング教育用の開発環境でもあるので、たとえば変数や配列を作るとそ  
の値の表示・非表示が選択できて、必要なら値をトレースしながらプログラムの動作確認ができる。

- アンブラグドプログラミング

コンピュータサイエンスアンブラグド<sup>[6]</sup>と呼ばれるコンピュータを使わずにコンピュータ科学を  
学ぶ教育法が提案されている。コンピュータを使わないプログラミングのことはアンブラグドプロ  
グラミングという。

小学校におけるプログラミング教育はコーディングを覚えることが目的ではない<sup>[1]</sup>とされている  
ので、コンピュータを使わずにアンブラグドで考える教育実践例が多い<sup>[7][8][9]</sup>。それらはフローチャ  
ート等を使って、プログラムの制御構造の3つの基本「順次 (シーケンス)」「反復 (繰り返し)」「分  
岐 (選択)」の考え方が意識されておりプログラミング的思考の学習が可能になる。しかし、アン  
ブラグドはプログラミングに対する壁をなくす効果があるがプログラミングスキルを身につけるため  
のものではないことにも留意したい。

### 3. プログラミング体験とその結果

#### 3. 1. 少人数を対象とした初めてのプログラミング体験

プログラミング経験のない大学生 8 名にプログラミング体験をさせた。これらの学生らは算数教育のゼミに所属して、ふだんのゼミでは算数教育に関する購読や演習を行っている。また、これらの学生たちは毎週、小・中学生へ算数・数学を指導する学習支援活動に参加しているので、算数をはじめ教科指導に対しての意欲は強い。小学校で 2020 年度から必修化されるプログラミングのためにプログラミング体験をしてみようということで行った。

a. 4 年生 3 名, b. 3 年生 5 名 の 2 グループに分けて行なった。

##### a. 4 年生グループのプログラミング体験

【対象者】 大学 4 年生 3 名

【プログラミング言語】 Python 3.7

【体験日と回数】 2019 年 4 月 11, 18 日, 5 月 16 日 (計 3 回, 各回 約 90 分)

【指導内容】 主な内容を表 1 に示す。

表 1 初めてのプログラミング体験と主な指導内容 (4 年生グループ)

回	年	月	日	言 語	形 態	主な指導内容
1	2019	4	11	Python	講義	演算子, データ型, 文字列
2	2019	4	18	Python	講義	変数, 代入文, コメント, プリント文
3	2019	5	16	Python	コーディング	関数 print(), input()

教科書として、邦題『退屈なことは Python にやらせよう』<sup>[10]</sup>を使用した。サブタイトルの一ノンプログラマーにもできる自動化処理プログラマーから、当初は作業効率化のためのコーディングができるようになることも目標とした。

第 1 章は Python 入門であり、プログラミング言語 Python を通して基本的なプログラム構造を学ぶことができる。授業時数計 3 回でこの教科書に沿って、各人にとって初めてのプログラミング体験となった。第 1 回目は、数学演算子、データ型 (整数、浮動小数点、文字列)、文字列についての操作について、第 2 回目は、変数、代入文、コメント、プリント文について、両回ともに基本的な事柄を説明する講義形式となり、あまり興味をわかせることができなかった。第 3 回目は、最初のプログラムを書かせた。それがコーディング初体験であった。

##### 【結果】

第 3 回目に行った初めてのコーディング体験は、サンプルプログラムの動作確認であったが、文字列を画面に表示する print() 関数とキーボードからの入力を返す input() 関数を使ったプログラムで、かなり興味をわかせることができた。言語がインタプリタ型であるためプログラムの記述と動作を適宜都合のよいステップに分けて確認することができたからである。そしてプログラミングの基礎とな

小学校教員養成課程の大学生にさせたプログラミング体験とその結果からのプログラミング教育についての考察  
 変数を扱う操作を知ることができた。もし納得した理解ができたなら、変数を扱う操作である代入と参照の方法は、プログラミングのおもしろさを最初に実感するところである。

残念ながら4年生ゼミでは卒業研究が主課題であるため、プログラミング体験はこの3回で中断し、その後は卒業研究（算数教育）に邁進することになり、自らプログラミングを行うほどの関心を示す者はなかった。しかし、その中にはこれをきっかけとしてソフトウェア開発企業への就職活動を行い、Pythonのプログラミング体験があることから採用の内定を受けた者もいた。このプログラミング体験で実際に修得できたスキルはプリント文を書ける程度のものだったが、Python修得者として見込まれてしまったようである。

#### b. 3年生グループのプログラミング体験

【対象者】 大学3年生 5名

【プログラミング言語】 Python 3.7 および Scratch 3.0

【体験日と回数】 2019年4月25日、12月5日（各回約90分）

【指導内容】 主な内容を表2に示す。

表2 初めてのプログラミング体験と主な指導内容（3年生グループ）

回	年	月	日	言語	形態	主な指導内容
1	2019	4	25	Python	講義	演算子, データ型, 文字列, 変数, 代入文, コメント, プリント文 関数 print(), input(), len(), str(), int(), float()
2	2019	12	5	Scratch3.0	コーディング	ブロック, 順次処理, 反復処理, 多角形の描画

3年生もPythonの教科書<sup>[10]</sup>に沿って、初めてのプログラミング体験となった。a.の4年生グループのプログラミング体験で2回かけて行った講義と同じ内容を1回で済ませた。そして、やはり講義ではあまり興味をわかせることができなかったため、Pythonについての講義はその1回で終了した。

その後半年以上経った12月、ビジュアルプログラミング言語であるScratchで、再びプログラミング体験をさせた。このとき、小学5年生の算数の単元「正多角形」の教育目標を明確にした上で、プログラムによる多角形の作図体験を行った。4月に行ったプログラム体験は座学中心だったので、これが実質初めてのプログラミング（コーディング）体験となり、プログラムとその動作を確認しながら、かなり興味をわかせることができた。

#### 【結果】

Scratchによるプログラミング体験の結果は、正三角形や正方形の作図プログラムから始め、最初に作った三角形の作図プログラムを基に回転角の大きさや反復（繰り返し）の回数を変える試行錯誤によって、四角形、五角形を作図ができた。しかし必ずしも繰り返し回数や回転角の大きさをきちんと考えているようすは伺えなかった。たとえば偶然に正五角形の対角線でできる星形がかけたのでそ

れを楽しんでいるという具合であった。それでも、プログラムのシーケンスや繰り返して作図ができていることはわかっているようだった。

そこで、Scratch のブロック操作から離れて、さらに正  $n$  角形の作図ができるように一般化を考えることにして、回転角の大きさの計算、多角形の外角の和についての考察を促すことで、多角形の性質を深く考察・理解した教材研究をすることができた。これはプログラミングが深い教材研究につながった一例といえよう。これについては、別のところで詳細を述べている<sup>[11]</sup>。

また、その中の 1 人がプログラミングに興味と関心を示し、後期の授業が終了している 2 月から Scratch のプログラミングスキルの修得を目指すことになった。これについては次節 3. 2. で述べる。

ところで、本稿執筆にあたり、2020 年 10 月上旬に当該の 5 人にプログラミング経験についてアンケートをとると、Scratch でプログラミングを行って作図した経験があるのにもかかわらず、5 人中 3 人がプログラミング経験は全くないと答えている。この回答について、改めてヒアリングすると、Scratch のことは頭によぎったがプログラミングをしたという自信がなく、プログラミング経験には入らないと思ったという答えが返ってきた。

### 3. 2. プログラミングスキル修得のための学習と指導

3.1 b からプログラミングに興味と関心を示した学生へのプログラミング指導を行った。

【対象者】大学生 1 名

【プログラミング言語】Scratch 3.0

【学習期間】2020 年 2 月 13 日～7 月 1 日（約 5 ヶ月の間にブランクもあるが、9 週間）

【学習方法】

学習者は Scratch の教科書<sup>[12]</sup>を自習し、そこにある課題プログラムを作っていくことにした。その教科書では、サンプルプログラムの動作を確認しながらサンプルのプログラムを作っていく方法を取り、プログラムへの機能追加が課題となっている。そして、Scratch に関することとプログラミングに関することを並行して学ぶ構成になっている。学習した教科書の構成内容を表 3 に示す。教科書の前半では、Scratch についての操作や仕組みとプログラミングの基礎的なことを学び、後半からはプログラミングに関することが中心となる。

当該学生によると、学習ペースは週に 1～2 回、1 回につき短くても 1 時間、長いときは 5 時間集中して行った。学習記録を表 4 に示す。学習を始めて 3 週間経った頃、課題のプログラムをまとめて進捗報告として筆者に提出してくれたので、それらを確認してバグの指摘とプログラムの改善提案を返した。当該学生はそれを受けてプログラムの修正を行っていくのだが、筆者が提出物の受け取りから最初の改善提案を返すまでに 10 日程経ってしまい、その後の学習は、ややペースダウンしたようすである。課題の内容も探索やソートとなり難しくなったせいか、自習はしていたようだが、作成したプログラムの提出はなかった。

【結果】

教科書の第 4 章あたりの構造化プログラミングに係る課題が始まったところから、バグを含み課題の



小学校教員養成課程の大学生にさせたプログラミング体験とその結果からのプログラミング教育についての考察仕様を満たさないままで提出されているものが増えてきた。その後も学習を続けているがペースダウンしていく。7月1日には、第8章と第9章を1日で終えているが、おそらく、十分に修得できたとはいえないだろう。この日で一旦プログラミングの学習の最終日としている。

学習完了後のヒアリングで、学習を進めるにあたって、困ったことを聞くと、

- (課題の解説がないときには、) 答えのプログラムを見て考えないといけないことが難しかった。
- テキストに手順が丁寧に書いてあるため、どうしても自分で考える前に見てしまう。

とのことだった。一方でその対策として、

- 1回サンプルファイル通りに作ってみて動かしてみる→工程ごとに(ステップごとに)分けて自己解釈する。
- 先にサンプルファイルを動かして組むスクリプトのイメージをもつ。

と答えている。

当該学生に先に述べたプログラム経験を問うアンケートを行っているが、

[経験はまったくない、経験したことはある、ある程度は経験している、できる]

の4つの選択肢のうち、消極的な2つ目の「経験したことはある」を回答している。

とはいえ、当該学生はScratchが得意とするアニメーションで動く簡単なゲームを作成することができるようになっている。また、プログラムとプログラミングの算数教育への応用例を單元ごとに考察することができるようになっている。算数の單元に合わせて、いくつかの算数ゲームプログラムも自作している。そして算数の学習とプログラミングの関係をテーマとした卒論研究に取り組んでいる。

表 3 学習した Scratch3.0 の教科書の構成内容

教科書 <sup>[12]</sup> の章番号と内容		Scratch について	プログラミングに関すること
0	Scratch の準備	画面の見方などの基礎知識	
1	Scratch の基本	基本的操作と用語 スプライト, コスチューム, ブロックとスクリプト	エンタリーポイント, 逐次 (順次) 処理, 繰り返し (反復), 初期化, 乱数, メソッド
2	プログラムの流れ	メッセージの仕組み, 背景, 音の鳴らし方	プログラムの流れ, ブロードキャスト
3	変数と配列	変数やリストの作り方 ユーザによる入力 ブロックのつなげ方 計算式の作り方 条件文の作り方	変数, 配列, 初期化, エラー処理
4	構造化プログラミング	複雑なブロックの組み方	アルゴリズム, 構造化プログラミング, 順次, 分岐, 反復, ネスト, 比較演算・論理演算, 整数と小数の計算
5	関数	ペン, 定義 (関数)	関数, 引数, 再帰処理の計算
6	サーチ (探索)		サーチ (探索), 無限ループ,
7	基本的なソート (並べ替え)		バブルソート, 選択ソート, 挿入ソート, トレース, 評価
8	すすんだソート (並べ替え)		クイックソート, シェルソート, 評価
9	クローン		クローン



表 4 Scratch 3.0 の学習および指導内容

2020	月	日		学習者	筆者	章番号と主な内容	備考
第1週	2	13	～	自習		0章, 1章, 2章	4hほど×1日
第2週	2	20	～			3章, 4章の条件分岐 (スロットマシン) まで	2h～4h×2日
第3週	2	27	～			4章の繰り返し, 条件分岐 (落ちゲーム) ～5章の引数まで	2h～3h×2日
第4週	3	5				5章の再帰処理～6章の線形探索まで	5hほど×1日
		3	6		提出	1～5章で作成したプログラムを送付	
		3	16		返送	バグの指摘と改善提案 (1～3章)	
		3	22			バグの指摘と改善提案 (4～5章)	
第5週	4	7	～	自習		バグの修正 1章～6章の線形探索まで	1hほど×4日
第6週	5	8	～			6章の最小値のサーチ～7章のバブルソートまで	3hほど×2日
第7週	5	15	～			7章の選択ソート	2hほど×1日
第8週	5	22	～			7章の挿入ソート, トレース, 評価	3hほど×1日
第9週	7	1	～			8章, 9章	5hほど×1日

### 3.3. 教職実践演習におけるプログラミング体験

教職実践演習 (オムニバス授業) の1回を使って「算数教育とプログラミング」というテーマで授業を行った。その授業では簡単なプログラミングをアンブラグドで体験させ、小学校で必修化されたプログラミング教育について知ることを目的とした。授業の後、プログラミング教育についての意識調査を行った。

表 5 教職実践演習「算数教育とプログラミング」の内容

No.	主な内容	備考
(1)	ソート (10 個の数を並べ替える)	紙上筆記操作
(2)	プログラムの制御構造の3つの要素を知る	順次, 反復, 分岐
(3)	プログラミング的思考とは何か	
(4)	Scratch 3.0 のテンプレートとソートプログラム	バブルソートの実行
(5)	アンブラグドプログラミング	掃除マニュアル, 料理
(6)	算数の学びとプログラミング	具体例
(7)	プログラミング教育についての意識調査	

【対象者】 小学校教諭免許状取得予定の大学4年生 75名

【プログラミング言語】 アンブラグド および Scratch 3.0

【体験日と回数】 2020年10月15日 (教職実践演習; オムニバス授業の中の1回 約90分)

【指導内容】 主な内容を表5に示す。

2020年から小学校でプログラミング教育が必修化されていることとそのねらいを説明し、次のように授業を展開した。

- (1) ランダムに10個の数が入ったリストから、別のリストに小さい順に並べて入れるように指示し、その手順を考えさせた(図1)。

リスト (before)	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px 10px;">63</td> <td style="padding: 2px 10px;">37</td> <td style="padding: 2px 10px;">27</td> <td style="padding: 2px 10px;">2</td> <td style="padding: 2px 10px;">64</td> <td style="padding: 2px 10px;">72</td> <td style="padding: 2px 10px;">24</td> <td style="padding: 2px 10px;">89</td> <td style="padding: 2px 10px;">45</td> <td style="padding: 2px 10px;">91</td> </tr> </table>	63	37	27	2	64	72	24	89	45	91
63	37	27	2	64	72	24	89	45	91		
(処理) これらの数を小さい順に並べ替える											
リスト (after)	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table>										

図1 ソートの紙上筆記操作

- (2) 実際に行った処理を整理するために図2に示すワークシートに書き込みながら、その手順がプログラムの制御構造の3つの要素「順次、反復、分岐」で表現できることを確認し、同時にプログラムはこれらの3つだけで構築できることが数学的にも証明されていることを知らせた。
- (3) プログラミング教育の取り組みのねらいの1つにあげられている「プログラミング的思考」について、一般に定義として使われている「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み合わせが必要であり、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いのか、記号の組み合わ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」<sup>[1]</sup>を知らせた上で、もっと具体的には、「プログラミング的思考」とは、順次、反復、分岐だけで構成していく力であると説明をした。
- (4) Scratch 3.0で順次、反復、分岐のテンプレート(ブロック)を説明し(図3)、Scratch 3.0であらかじめコーディングしていた並べ替えのプログラム(バブルソート)の解説をした(図4)。そして、1000個の乱数を発生させて、バブルソートを行った。先に述べたようにScratchはプログラミング教育用の開発環境でもあるので、1000個の数が入っている配列の中を表示しながらバブルソートを実行すると、水から泡があがってくるように、配列の最後尾から数が入り替わるようすが目視でき、あらかじめ解説した動きを実際に見ることができた受講生からは感嘆の声が漏れた(図5)。
- (5) コンピュータを用いずにプログラミング的思考するアンプラグドプログラミングがあることを知らせた。
- (6) プログラミング教育のねらいのもう1つにあげられている「各教科等で学ぶ知識及び技能をより確実にする」についてとくに、プログラミングが算数の深い学びとかわる具体例を順次、反復、分岐の3つの要素との関係で示した。
- (7) 授業の最後に、プログラミング教育に対してもっている意識について、大学のLMSであるmanabaを使ってアンケート調査をした(図6)。

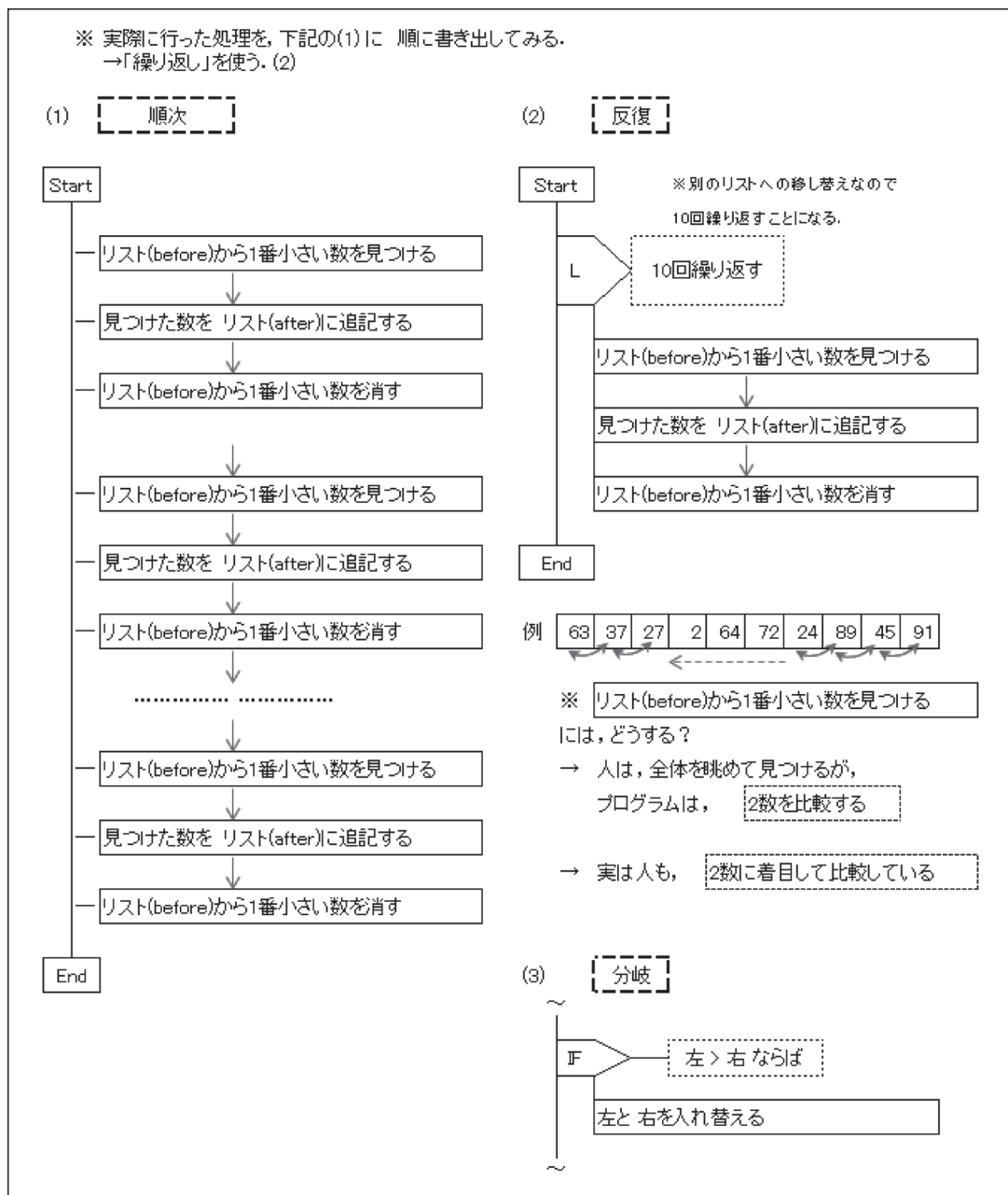


図 2 プログラムの制御構造（順次、反復、分岐）ワークシート（完成例）



図 3 Scratch のテンプレート（ブロック）とスプライト

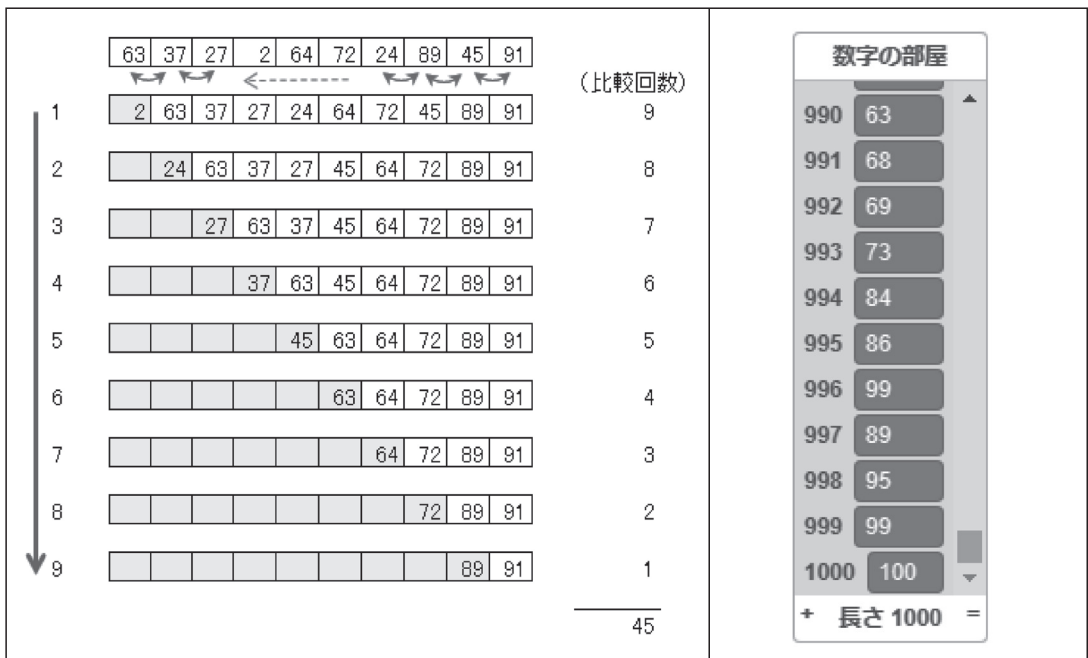


図 4 バブルソートアルゴリズムの解説

図 5 バブルソート中の配列

プログラミング教育についての意識調査	
受付期間	2020-10-15 17:30～2020-10-17 23:55
ポートフォリオ	回答を学生のポートフォリオに追加
学生による再提出の許可	再提出を許可しない

※アンケート集計シートに表示される問題番号を赤の太字で表示しています(例: **1.1**)。

A. 次の質問1～3は、今日の「算数教育とプログラミング」の  
受講前 のことについてうかがいます。

---

1. 2020年度から小学校でプログラミング教育が必修化されています。  
あなたはこのことを知っていましたか。  
(選択必須)

**1.1**

1.  知っていた
2.  知らなかった

2. あなたはプログラミングというものを知っていましたか。  
(選択必須)

**1.2**

1.  まったく知らなかった
2.  言葉は知っているがよく知らなかった
3.  ある程度は知っていた
4.  よく知っていた

3. あなたのプログラミング経験はどんなものですか。  
(選択必須)

**1.3**

1.  経験はまったくなかった
2.  ほんの少しだったが経験したことはある
3.  ある程度は経験している
4.  プログラミングができる

B. 次からの質問4～8は、今日の「算数教育とプログラミング」の受講後 のことについてうかがいます。

4. 小学校でのプログラミング教育の必要性について、どう思いますか。  
(選択必須)

1.4

- 1.  必要ない
- 2.  必要だ
- 3.  どちらともいえない
- 4.  わからない

5. 上の質問4 で選んだ答えの理由を書いてください。

(入力必須)

1.5

6. 小学校教員になったとき、プログラミングを教えることについてどう思いますか。  
(選択必須)

1.6

- 1.  とても不安
- 2.  少し不安がある
- 3.  あまり不安はない
- 4.  不安はない
- 5.  できる

7. あなた自身はプログラミングを学びたいと思いますが、

(選択必須)

1.7

- 1.  まったくそう思わない
- 2.  どちらかといえば思わない
- 3.  どちらかといえば思う
- 4.  とてもそう思う

8. 上の質問7 で選んだ答えの理由を書いてください。

(入力必須)

1.8

9. 今日の授業「算数教育とプログラミング」の感想を書いてください。

(入力必須)

1.9

図 6 プログラミング教育についての意識調査アンケート



【アンケートの結果】

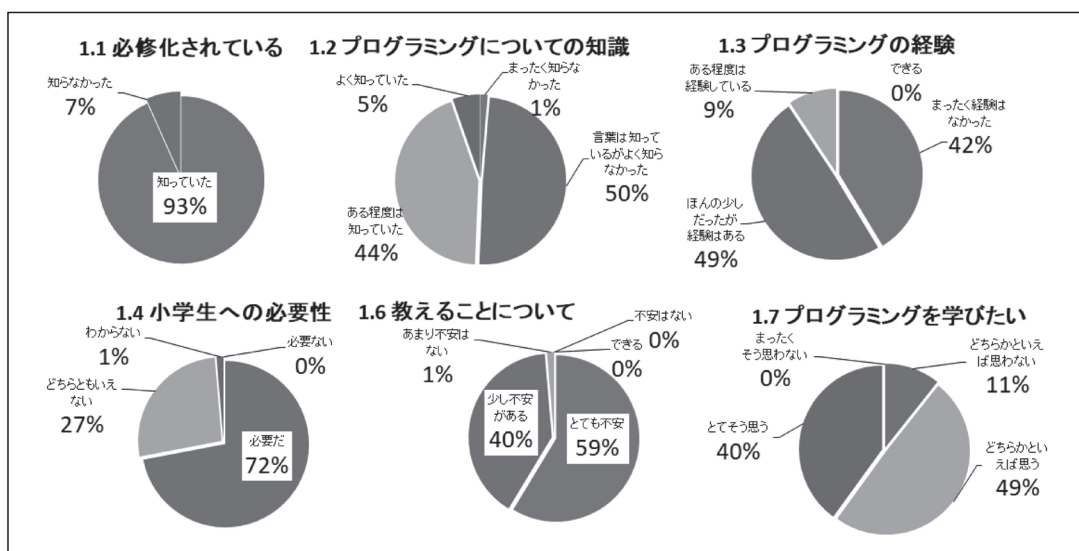


図 7 プログラミング教育についての意識調査結果

図 7 にアンケートの結果を示す。アンケートの 1.1 から 1.3 までは、この授業を受ける前の状況を尋ね、1.4 ～ 1.9 までは、受講してプログラミング教育やプログラミングについてどのような意識をもっているかについて尋ねている。

まず受講前の時点で、小学校では 2020 年 4 月からプログラミング教育が必修化されていることを 93% の受講生が知っていたが、この授業を受けて初めて知ったというのが 75 名中 5 人いた。プログラミングとは何かについて、まったく知らなかったと答えたのは 1 人だった。プログラミングの経験は 42% がまったくなかったが、およそ 6 割が経験したことがあると答えている。

受講後では、小学生へのプログラミング教育の必要性について 72% が必要だと考えており、どちらともいえないが 27% である。必要ないと答えた者はいなかった。必要と答えた理由には「プログラミングの反復や分岐の活動を通して、思考する力がつくと考えるから」「プログラミング的思考はスモールステップと同じようなものだ」と捉えているから」があり、どちらともいえないとした理由には「プログラミング的思考は必要性を感じたが、技能面ではあまり感じなかった」「小学生がどの程度理解できるか」「児童や教師の負担が増える」といった意見があった。

小学校で教えることについては 99% が不安だと感じている。プログラミングを学びたいと思うかについては、89% が学びたいと思っており 11% は思っていない。学びたいと思う理由は「子どもに教えるためには自分自身が理解し、できるようになっていないと教えるのは難しいと思うから。また、プログラミングができれば作業効率が上がると思うから」「今日、先生が scratch を触っているのを見て、私自身もやってみたいと思った」があり、学びたいと思わない理由は「児童のために勉強しようとは思いますが自分のためには勉強しようとするところ思わない」「数学のように論理的に考えることが苦手だから」があった。

最後にこの授業を受けた感想を次の①～④のカテゴリに分けて示す。

① プログラミング教育についての不安

「今日の授業でプログラミングの入口の部分がようやく分かったように思う」

「難しい印象を受けたが、ロボットにハンカチのたたみ方を説明することや、掃除のマニュアルを作ることは、身近に感じやすく、とっつきにくいなと思っていた部分が少なくなった」

「自分自身がプログラミング教育をしっかり受けてきたことがなかったので、とても不安に感じていました。今日の講義を聞いて、少しプログラミングについての不安が薄まりました」

「教師の負担がまた増えるかもしれないと感じた」

② アンプラグドプログラミング

「今日の授業で、順次、反復、分岐を自分の頭で行なっていたことがプログラミング思考であり、プログラミングにつながっていたことを知ることができた」

「コンピュータを用いずとも、『プログラミング的思考』をすることができるアンプラグド教材はともこれから教育において重要になると感じた。地域によっても情報機器の導入などに差があると思うが、この教材によりどの自治体も一定の基準でプログラミング教育を行うことができる考えた」

「プログラミング的思考は大切だと思う。しかし、アンプラグドでも十分なのかなとも思った」

③ プログラミングのよさ、おもしろさ

「実際にソートを手動でやって手順を考えて書き出してから、Scratch でやってみせることで利便性がよく分かった。パッと一目見て分かるようでも、実は人の思考過程もプログラミングと同じであるという点に面白さを感じた」

④ 算数をはじめ教科の学びとプログラミングの結びつき

「プログラミングはプログラミング と考えていましたが、算数と結びつけることができることを知りました」

「スクラッチのことは知っていたが、プログラミングが図形の分類とつながっていることは知らなかった。中学生が図形の特徴を判別することが難しいと言っていたのを耳にしたことがあったので、分類表を作るなどしたら、その子達にもわかりやすく説明することができるのかなと思った」

「算数教育だけでなく、他の教科にもこのプログラミング教育は活用できるのではないかと、コロッケの作り方を見て感じることも出来た」

#### 4. まとめと考察

小学校におけるプログラミング教育については「プログラミング的思考」などを育むことであり、コーディングを覚えることが目的ではない。そして教育実践例の調査から、実際の小学校で行われているプログラミング教育では、プログラムの制御構造の3つの要素である順次、反復、分岐を意識したアンプラグドプログラミングが多く実践されていることがわかった。

本研究では小学校教員養成課程で学ぶ大学生を対象にプログラミング教育およびアンケート調査を

小学校教員養成課程の大学生にさせたプログラミング体験とその結果からのプログラミング教育についての考察を行った。それぞれ、

- (1) 少人数を対象とした初めてのプログラミング体験
- (2) (1)でプログラミングに興味と関心を示した学生へのプログラミング指導
- (3) 小学校教諭免許状取得見込みの4年次生へのプログラミング指導と意識調査

である。

(1)で、テキストで記述するプログラミング言語 Python を使ってプログラミングの基本的な事柄を説明した講義形式では、あまり興味をわかせることができなかつたが、動作確認を伴ったプログラミングではかなり興味をわかせることができた。Python は、インタプリタ型言語なのでプログラムの動作確認が適宜都合のよいタイミングでできる。それは初心者のプログラミング学習にとっても都合がよい。同様にビジュアルプログラミング言語の Scratch でもすぐに動作確認ができることで興味をわかせることができた。最初、プログラミングに対して尻込みをしていた学生も、プログラミングとそれによる動作が連動するとかなり興味がわいてくるようであった。

(2)で、Scratch の修得にチャレンジした学生からは、自身でプログラミング言語を学んだことについて、「プログラミングを用いた授業のイメージをもてるようになり、授業にスムーズに導入できると自己評価している。」そしてプログラミングを使用した算数の授業を考えたときには「この授業で押さえるべき点はどこなのか、真に児童に理解してもらいたい事柄を見つけることができるようになり、学習指導案を作成することが今までより苦手だと感じるものが減った。それは、プログラムを作成するときと同じで、要件とその優先順位を考えているからだと思われる。」と報告している<sup>[13]</sup>。プログラミングスキルを修得した結果、これまでと授業づくりに対しての見方が変わったようである。この例からもプログラミングスキルの修得は教師にとっても有効に働くのではないかと考える。

(3)の小学校教諭免許状取得を目指す4年次生を対象として行ったアンケート調査からは、ほとんど全員がプログラミングを教えることについて不安に思っていたが、プログラミング教育は必ずしもコンピュータを使うものではなく、「プログラミング的思考」育成のためのアンブラグドプログラミングを知ると少し不安が解消されたようすが伺えた。そして小学校における各実践例から見ても、およそどの教育現場も同じような状況ではないかと考えられる。つまり、まずプログラミング教育として何から取り組んでいけばよいかという不安や迷いがある、アンブラグドプログラミングにたどり着くのである。

したがって、小学校で習うプログラミングとはどのようなものなのかという点、まずはアンブラグドで「プログラミング的思考」を育むことであり、それによって培われる「プログラミング的思考」は、プログラムの制御構造の3つの要素である順次、反復、分岐を組み立てて自分が意図する一連の活動を実現する力を用いて、アンブラグドプログラミングの指導の際、指導者は「順次、反復、分岐」を意識しておくことが肝要であると考えられる。

その一方で、内閣府は2019年12月5日「安心と成長の未来を拓く総合経済対策」を閣議決定し、「義務教育段階において、令和5年度までに、全学年の児童生徒一人一人がそれぞれ端末を持ち、十分に活用できる環境の実現を目指す」こととなり、そこに年が明けて起こった新型コロナウイルスの

パンデミックにより、2020年4月20日に「新型コロナウイルス感染症緊急経済対策」が閣議決定され、その整備スケジュールはさらに加速し、令和3年4月から、全国のほとんどの義務教育段階の学校において、児童生徒の「1人1台端末」及び「高速大容量の通信環境」の下での新しい学びが本格的にスタートすることになった。したがって、ハードウェア環境が整うならば、アンプラグドプログラミングだけでは、コーディングスキルはつかないことにも留意し、アンプラグドだけに頼らないようにしなければならない<sup>[14]</sup>。アンプラグドプログラミングはプログラミング教育の入口にすぎないので、今後教室にハードウェア環境が整ったときに、Scratchをはじめとするビジュアルコーディング言語や、インタプリタ型言語を使ったプログラミング（コーディング）であれば、児童にとっても全く手が出せないということはない。個人差はあるにしても、児童に関心・意欲があればアンプラグドから発展して実際のプログラミングへと展開していくことも十分可能である。

そこでそのためには、小学校教員もアンプラグドだけにとどまらず、プログラミング言語を一つ修得もしくは、少しコーディングができることが望ましい。したがって、次第に教員養成課程カリキュラムの中にプログラミングの授業を入れることも検討されていくのではないだろうか。

最後に、本稿第2章の冒頭に述べた小学校でプログラミングに取り組むねらいの3つ目にある各教科等で学ぶ知識及び技能をより確実にすることについて、とくに本研究では、算数の学びとプログラミングの関係について述べ、注意喚起をしておきたい。それは算数の学習内容の理解につなげるためのプログラミング教育を考えることであって、プログラミング的思考を通して学習内容の理解を深めるということである。プログラミング的思考（順次、反復、分岐の構造）から算数の深い学びを導くということは、「プログラミング的思考 → 算数の深い学び」であって、この矢印の向きに意味がある。

わり算の筆算を例にあげると、その操作は順次、反復、分岐の構造で操作を行っており、筆算の手順をその構造でとらえることによって、いろいろなタイプのわり算の計算操作をすることができているのである。わり算の学習が進んでいくと余りのあるわり算を扱う。その余りを切り上げる場面、切り捨てる場面を理解することもそこでの重要な教育目標である。

$20 \div 3$  のわり算を例にその意味を考えると、3人掛けの長椅子を用意して20人全員が座ることができるためには余りを切り上げて長椅子の数を考える。また、20個あるものから3個ずつ詰めて1箱の包装をするなら、余った個数では1箱は完成しないので余りを切り捨てる。このような場合分けや意味を理解するためにはプログラミングの構造で考えることが有用であるというのが、プログラミング教育に取り組む3つ目のねらいであると考えられる。

しかし、学習内容からプログラミング的思考をさせるという流れでは、矢印の向きは反対なので、それはプログラミング的思考を育むという1つ目のねらいには当たることがあるかもしれないが、3つ目のねらいには当たらない。この2つを混同してはならない。

なお、正しく3つ目のねらいに沿った算数の授業例が、他にも、

- 位取り記数法で表記された数の大小を比較する（第2学年）

- 既習のグラフ（棒・折れ線・円・帯）の特徴を整理し、目的に応じた適切なグラフの選択（第5学年）

などプログラミング的思考（フローチャート）を活用して、教育目標の達成を目指す授業実践も報告

小学校教員養成課程の大学生にさせたプログラミング体験とその結果からのプログラミング教育についての考察されてきている<sup>[9]</sup>。そのようなプログラミングの構造を用いて学びを深めていくものにどんな教育内容や教材があるのかを具体的に示していかなければならない。そしてそのプログラミングの構造がもたらす効用について考察していかなければならないが、それは今後の課題としたい。

## 引用・参考文献

- [1] 文部科学省 (2016) 「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)」, [https://www.mext.go.jp/b\\_menu/shingi/chousa/shotou/122/attach/1372525.htm](https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm), (最終閲覧日: 2020年10月29日).
- [2] 文部科学省 (2016) 「幼稚園、小学校、中学校、高等学校及び特別支援学校の学習指導要領等の改善及び必要な方策等について (答申) (中教審第197号)」, [https://www.mext.go.jp/b\\_menu/shingi/chukyo/chukyo0/toushin/1380731.htm](https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo0/toushin/1380731.htm), (最終閲覧日: 2020年10月29日).
- [3] 文部科学省 (2017) 「小学校学習指導要領 (平成29年告示)」
- [4] 文部科学省 (2020) 「小学校プログラミング教育の手引 (第三版)」
- [5] 文部科学省 (2018) 『小学校学習指導要領 (平成29年告示) 解説 総則編』, p.85, 東洋館出版社.
- [6] コンピュータサイエンスアンプラグド, <https://www.csunplugged.jp/>, (最終閲覧日: 2020年10月28日).
- [7] 黒上晴夫, 堀田龍也 (2017) 『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』, 小学館.
- [8] 小林祐紀, 兼宗進 (2017) 『コンピューターを使わない小学校プログラミング教育 “ルビィのぼうけん” で育む論理的思考』, 翔泳社.
- [9] 小林祐紀, 兼宗進, 白井詩沙香 (2018) 『これで大丈夫! 小学校プログラミングの授業 3+  $\alpha$  の授業パターンを意識する [授業実践39]』, pp.44-45, pp.96-97, 翔泳社.
- [10] Al Sweigart 著・相川愛三訳 (2017) 『退屈なことは Python にやらせよう—ノンプログラマーにもできる自動化処理プログラミング—』, オライリー・ジャパン.
- [11] 穴田恭輔 (2020) 「プログラミング教育によって導かれる小学校算数の教材研究」2020年春季全国大会 (第36回大会) プログラム集, pp.193-194, 日本教育工学会.
- [12] 中植正剛, 太田和志, 鴨谷真知子 (2019) 『Scratch で学ぶプログラミングとアルゴリズムの基本』, 日経 BP.
- [13] 坪谷論泉那 (2021) 「算数の学びを支援するためのプログラミングの効用とプログラミングを用いた授業づくり」, pp.52-53, 神戸女子大学文学部教育学科卒業論文.
- [14] 平井総一郎, 利根川裕太 (2020) 『なぜ、いま学校でプログラミングを学ぶのか—はじまる「プログラミング教育」必修化』, p.106, 技術評論社.