

博士論文

小学校音楽科における  
プログラミング教育のための学習材の開発  
— 〈ライブ・コーディング〉を手がかりとして—

Developing the Web Application of ‘Live Coding’  
as Learning Material:

Programming Education in Elementary School Music Courses

広島大学大学院教育学研究科  
教育学習科学専攻 学習開発学分野  
カリキュラム開発領域

Hiroshima University  
Graduate School of Education  
Program in Education and Learning Science  
Learning and Curriculum Development  
Curriculum and Instruction Development

D172335 長山 弘  
NAGAYAMA, Hiroshi

2023年3月

## 目次

【序論】 .....	1
本研究の概要 .....	1
本論文における語の定義 .....	1
研究倫理を踏まえた配慮 .....	3
【第一部】 理論編 .....	4
1. 小学校段階におけるプログラミング教育 .....	4
1.1. 学校教育への導入をめぐる議論 .....	4
1.2. 「議論の取りまとめ」以前の動向 .....	9
1.3. 「有識者会議」による「議論の取りまとめ」（2016年6月） .....	15
1.3.1. プログラミング教育を実施する背景 .....	15
1.3.2. 小学校段階におけるプログラミング教育の目的 .....	16
1.3.3. 各教科等におけるプログラミング教育の実践案 .....	16
1.4. 「議論の取りまとめ」以降の動向 .....	16
1.5. 小学校段階におけるプログラミング教育の概要 .....	24
1.5.1. 小学校段階のプログラミング教育において育むべき資質・能力 .....	24
1.5.2. 小学校段階におけるプログラミング教育の評価 .....	28
1.5.3. 各教科等におけるプログラミング教育の実践事例 .....	28
1.5.4. プログラミング教育のためのハードウェアやアプリケーション .....	30
1.6. 必修化以降の動向—GIGA スクール構想（2019年12月～）を中心に— .....	30
1.6.1. GIGA スクール構想の内容と変遷 .....	30
1.6.2. プログラミング教育と GIGA スクール構想 .....	31
1.6.3. GIGA スクール構想の課題 .....	32
1.7. プログラミング教育に関する研究の動向 .....	32
1.7.1. 国内の動向の調査 .....	32
1.7.2. 海外の動向の調査 .....	33
1.7.3. 〈プログラミング的思考〉に関わる研究 .....	34
1.7.4. プログラミング教育のための教材・学習材の開発 .....	35
1.7.5. プログラミング教育の指導・学習法の検討 .....	36
2. 小学校音楽科において〈プログラミング的思考〉を育む活動に対する問題提起 .....	39
2.1. 小学校音楽科とプログラミング教育との親和性に関わる先行研究 .....	39
2.2. 小学校音楽科におけるプログラミング教育の授業実践事例 .....	41
2.2.1. 実践事例収集の対象の範囲と期間 .....	42
2.2.2. 実践事例収集の抽出方法 .....	43

2.2.3. 事例の一覧 .....	43
2.3. 小学校音楽科におけるプログラミング教育の授業実践事例の傾向.....	45
2.4. 「コンピュータを動作させるための手順（例）」に基づいた学習の過程に着目した事例検討.....	46
2.4.1. 「コンピュータを動作させるための手順（例）」 .....	46
2.4.2. 「すべての教科に共通する」児童の学習の過程.....	47
2.4.3. 「音楽科の授業事例を踏まえた」児童の学習の過程 .....	48
2.5. プログラミング教育に反映させるべき音楽科の特質 .....	52
2.6. 小学校音楽科の特質に応じた〈プログラミング的思考〉と新たな問題提起 .....	55
<b>3. 試行錯誤による意図の明確化—〈ティンカリング〉—</b> .....	<b>57</b>
3.1. 〈ティンカリング〉の定義とプログラミング教育に関わる先行研究.....	57
3.1.1. 〈ティンカリング〉の定義.....	57
3.1.2. プログラミング教育における〈ティンカリング〉の意義 .....	58
3.2. プログラミング活動のためのコンピュータ操作の習得過程 .....	60
3.3. 〈ティンカリング〉の特質.....	61
3.3.1. ウィルキンソン・ペトリッチが示した理念.....	61
3.3.2. 〈状況的行為〉と〈プランニング・モデル〉 .....	62
3.3.3. 〈道具主義的方法〉 .....	63
3.4. アプリケーションを用いた音楽づくり活動における〈ティンカリング〉の特質..	64
3.4.1. 道具と材料 .....	64
3.4.2. 目に見えない材料を用いた創作過程の特質 .....	65
3.4.3. 〈ティンカリング〉の道具としてのアプリケーションの意義.....	65
3.5. 〈ティンカリング〉と〈プログラミング的思考〉との関わり.....	67
3.6. 学習材開発における〈ティンカリング〉の強みを意識した位置付け .....	68
<b>4. 〈ティンカリング〉のパフォーマンス化—〈ライブ・コーディング〉—</b> .....	<b>70</b>
4.1. 〈ライブ・コーディング〉による表現の具体.....	70
4.2. 本研究における「〈ライブ・コーディング〉による音楽表現」の定義 .....	72
4.3. 〈アルゴリズム作曲〉の概要と特質 .....	73
4.3.1. コンピュータを用いない〈アルゴリズム作曲〉 .....	73
4.3.2. 〈アルゴリズム・コンピューター音楽〉 .....	74
4.3.3. コンピュータを活用した「音響合成」 .....	75
4.4. 〈ライブ・コーディング〉の即興性.....	76
4.5. パフォーマンスとしてのプログラミング .....	78
4.5.1. 「いま、ここ」への対応力.....	78
4.5.2. プログラムを構築する過程.....	80

4.6. 〈ライブ・コーディング〉に期待される教育的意義 .....	83
<b>【第二部】実践編 .....</b>	<b>86</b>
5. 小学校音楽科におけるプログラミング教育のための学習材の開発 .....	86
5.1. 開発する学習材がめざすプログラミング教育 .....	86
5.2. プログラミング教育を達成するための方策 .....	87
5.3. 〈ライブ・コーディング〉の特質を取り入れたプログラミング教育の実践事例 .....	88
5.4. 開発の方針と工夫の具体 .....	89
5.5. 《テクミュ (Tec-Mu)》の具体 .....	91
5.5.1. ハードウェアの構成 .....	92
5.5.2. Web アプリケーションの具体 .....	92
5.5.3. 《テクミュ》の概要 .....	96
5.5.4. 具体的な操作 .....	96
5.6. 開発において達成された方針と工夫 .....	97
6. 学習材《テクミュ (Tec-Mu)》を用いた授業実践と結果の検証 .....	99
6.1. 授業実践の概要 .....	99
6.1.1. 授業実践の対象と日時 .....	99
6.1.2. 児童の実態 .....	99
6.1.3. 授業のねらい .....	100
6.1.4. 授業実践の環境 .....	100
6.1.5. 授業実践の流れと児童の様子 .....	101
6.2. 授業実践の結果を踏まえた《テクミュ》の検証 .....	102
6.2.1. 「音楽を特徴付けている要素」の理解を深めることに関する観点の検証 .....	102
6.2.2. 直観的な操作性についての観点の検証 .....	103
6.2.3. 児童のつくる音楽の多様性の保障に関わる観点の検証 .....	104
6.2.4. 開発時にめざしたプログラミング教育の検証 .....	105
6.3. 再検証すべき「開発の方針と工夫」の観点 .....	105
7. 《テクミュ》の改良と実践—命令と実行の結果との即時性及び操作性に焦点を当てて— .....	107
7.1. 改良にあたっての手立てと具体 .....	107
7.1.1. 「音楽を特徴付けている要素」の理解を深めるための改良 .....	108
7.1.2. 直観的な操作性を高めるための改良 .....	109
7.1.3. その他の改良 .....	113
7.1.4. 改良によって生じた変更点 .....	113
7.2. 「二版」による授業実践と結果 .....	114

7.2.1. 授業実践の対象と日時 .....	114
7.2.2. 活動の環境と児童の ICT 活用の経験.....	115
7.2.3. 授業実践の流れ .....	115
7.2.4. 個別活動における指導と児童の姿 .....	116
7.2.5. 発表活動における児童の姿と音楽をつくる過程の具体.....	116
7.3. 授業実践と質問紙調査の結果を踏まえた考察.....	117
7.3.1. 『音楽を特徴付けている要素』の理解を深めること」についての検証.....	118
7.3.2. 「直観的な操作性について」の検証.....	119
7.3.3. 《テクミュ》とプログラミング教育との関わり.....	120
7.4. 《テクミュ》における音楽づくり活動の過程と意義.....	123
<b>【総括】</b> .....	124
謝辞.....	128
<b>【引用・参考文献】</b> .....	130

## 図表一覧

図 1	小学校段階におけるプログラミング教育の導入をめぐる議論の経緯を示したタイムライン.....	8
図 2	「プログラミングに関する教育の推進」（文部科学省生涯学習政策局情報教育課 2015a, p. 30）.....	13
図 3	「小学校プログラミング教育のねらいと位置付けについて」（文部科学省 2020a, p. 6）.....	25
図 4	「コンピュータを動作させるための手順（例）」（文部科学省 2020a, p. 14）.....	26
図 5	「手引」における「正三角形」を描くプログラム例（文部科学省 2020a, p. 15）.....	27
図 6	すべての教科に共通する児童の学習の過程.....	48
図 7	事例 3（ベネッセ 2019）における児童の学習の過程.....	50
図 8	事例 10（ベネッセ 2018d）における児童の学習の過程.....	51
図 9	事例 12（小林・兼宗 2017, pp. 34-37）における児童の学習の過程.....	52
図 10	小学校音楽科の表現領域におけるプログラミング教育を取り入れた学習の過程..	55
図 11	「ものづくりにはステップがある」（金 2022, p. 21）.....	59
図 12	〈ライブ・コーディング〉の様子（細谷 2017）.....	71
図 13	〈ライブ・コーディング〉におけるプログラム（アルゴリズム）と音楽との関係.....	77
図 14	《テクミュ》のハードウェア構成図.....	92
図 15	《テクミュ》のアクセス画面.....	92
図 16	《テクミュ》のソフトウェア構成図.....	93
図 17	《テクミュ》における〈ライブ・コーディング〉性をもたせた発音処理の流れ..	93
図 18	《テクミュ》のメイン画面.....	94
図 19	《テクミュ》の画面構成・遷移図.....	95
図 20	《テクミュ》の「音の高さ」変更画面.....	95
図 21	《テクミュ》の「音の長さ」選択画面.....	97
図 22	《テクミュ》の「音の大きさ」選択画面.....	97
図 23	授業で用いたスプリッター（分配器）.....	100
図 24	児童とコンピュータ等との位置.....	101
図 25	「二版」のアクセス画面.....	107
図 26	「初版」の発音処理の流れ.....	108
図 27	「二版」の発音処理の流れ.....	109
図 28	「二版」における画面構成・遷移図.....	110
図 29	「共通概念」を取り入れた「二版」の操作画面.....	111
図 30	「一貫性」を取り入れた「二版」の操作画面.....	112
図 31	「二版」における「音の高さ」入力用の鍵盤画面.....	112

図 32	「二版」で「打楽器群」を選択した場合の「音の高さ」の入力画面	113
図 33	「二版」で表示する「ヒント」	113
図 34	「二版」のハードウェア及びソフトウェア構成図	114
図 35	Q1 の共起ネットワーク	118
図 36	Q2 の共起ネットワーク	121
図 37	《テクミュ》における音楽づくり活動の過程	125
表 1	「手引」(文部科学省 2020a)における語の定義をもとに筆者作成	2
表 2	小学校段階におけるプログラミング教育の変遷の区分と内容(山本・堀田(2021, pp. 38-39)をもとに筆者作成)	5
表 3	中等教育段階におけるプログラミング教育の変遷(阪東ら(2017, pp. 173-175)をもとに筆者作成)	6
表 4	「我が国の初等教育にプログラミング教育が導入されるようになった経過」(阪東ほか 2017, p. 175)	7
表 5	2022 年 3 月時点における「成長戦略」一覧(首相官邸(2020b)をもとに筆者作成)	10
表 6	情報ワーキンググループの議事開催日等の一覧(文部科学省(2016a)をもとに筆者作成)	12
表 7	「手引」における小学校段階におけるプログラミング教育の学習活動の分類(文部科学省(2020a, p. 23)をもとに筆者作成)	18
表 8	「手引」における活動の分類と内容の例示数の変遷	22
表 9	「手引」の A・B 分類における各教科の活動事例	23
表 10	プログラミング教育の授業実践事例を扱った書籍	29
表 11	音楽科における指導内容と留意点(「議論の取りまとめ」(「有識者会議」2016)をもとに筆者作成)	40
表 12	小学校音楽科におけるプログラミング教育の事例一覧(長山(2019, p. 58)の表 1をもとに加筆・修正)	44
表 13	「コンピュータを動作させるための手順(例)」に即した児童の学習の過程	47
表 14	「コンピュータを動作させるための手順(例)」と音楽科の表現領域において音楽表現を成立させるための手順	49
表 15	児童の ICT 端末に対する親しみ	99
表 16	児童のプログラミングに対する親しみ(複数回答可)	99
表 17	児童の即興表現に対する自信	100
表 18	授業の流れ	101
表 19	児童による《テクミュ》を用いた活動の楽しさに対する評価	102
表 20	児童による《テクミュ》の操作性に対する評価	103

表 21	ICT 端末を用いた音楽づくり活動への親しみ.....	115
表 22	「二版」を用いた授業実践の流れ.....	115
表 23	児童による「二版」の操作性に対する評価.....	120



## 【序論】

2020年度より，小学校段階においてプログラミング教育が必修化された。

本研究の目的は，小学校音楽科においてプログラミング教育を実施するための要点を踏まえた学習材として，Webアプリケーションを開発することである。

序論では，本研究の概要と，研究に関わる語の定義を示す。

## 本研究の概要

本論文は，【第一部】理論編と【第二部】実践編によって構成されている。

【第一部】理論編では，小学校段階にプログラミング教育が必修化された背景をとりまとめ，それを踏まえて，小学校音楽科の表現活動にプログラミング教育を取り入れた学習の過程を検討し，活動を実現するための手立てを検討する。具体的には，次の3つの研究内容から成る。

- ① 小学校段階におけるプログラミング教育について，その導入に至った経緯，教育のねらい等を整理して示すとともに，この分野の研究動向を概説する。
- ② 小学校音楽科におけるプログラミング教育の現状を批判的に検討し，表現領域一とりわけ音楽づくり活動一の特質に応じた，プログラミング教育のあり方を考察する。
- ③ ②の成果に基づき，音楽づくり活動とプログラミング活動を関わらせた活動のひとつとして，〈ライブ・コーディング〉と呼ばれる，プログラムを作成する過程をパフォーマンスとして行う表現手法を取り入れることを提案する。

【第二部】実践編では，理論編を踏まえた学習材の開発と，それをを用いた授業実践，改善，考察を行う。具体的には，次の2つの研究内容から成る。

- ④ 小学校音楽科で〈ライブ・コーディング〉を実施するためのアプリケーション《テクミュ (Tec-Mu)》を開発する。また，その開発過程および特質を述べる。
- ⑤ 《テクミュ》を用いた授業を小学校で実践し，その結果を踏まえて，児童に育まれる〈プログラミング的思考〉を検討し，学習材としての意義を考察する。

## 本論文における語の定義

ここでは，プログラミング教育に関わって「プログラム (Program)」，「プログラミング (Programming)」，「アルゴリズム (Algorithm)」，さらに，学習材開発に関わって「コーディング (Coding)」，「ソフトウェア (Software)」，「アプリケーション (Application)」という語の定義を確認する。

現在、プログラミング教育に関わる「プログラム」、「プログラミング」、「アルゴリズム」という 3 つの語は、プログラミング教育に関わる様々な書籍、論文、実践報告等で用いられているものの、定義が統一されておらず、曖昧に使い分けられている<sup>1</sup>。

そこで、本論文では、これらの語を、「小学校プログラミング教育の手引（第三版）」（以下、「手引」）（文部科学省 2020a, p. 1）の定義にしたがって用いることとしたい。「手引」での定義を表 1 に記した。

表 1 「手引」(文部科学省 2020a) における語の定義をもとに筆者作成

「アルゴリズム」	「問題を解決する手順を表したもの」(p. 12)
「プログラム」	コンピュータが動作するための命令 (p. 1)
「プログラミング」	コンピュータに「命令を与えること」(p. 1)

すなわち「問題を解決する手順」を何らかの形で表したものが「アルゴリズム」であり、アルゴリズムをコンピュータが動作する形で表したものが「プログラム」、そして、プログラムを構築する行為が「プログラミング」である。「アルゴリズム」と「プログラム」のいずれも、「問題を解決する手順」を何らかの形で表したものであることには違いはないが、両者は、「コンピュータ」が関わるかどうかで区別している。すなわち「コンピュータ」が動作するための方法（プログラミング言語による命令）を用いて手順を表したものが「プログラム」、他方それ以外の方法—例えば文章で表した手順や紙の上で作成するフローチャートなど—を用いて手順を表したものが「アルゴリズム」ということになる。

次に、学習材開発に関わって、「コーディング」、「ソフトウェア」、「アプリケーション」の語を『学研 現代新国語辞典』（金田一春彦・金田一秀穂 2017）より参照する。

「コーディング」について、金田一春彦・金田一秀穂（2017, p. 501）は「コンピュータで、プログラミング言語を用いてプログラムを作ること」と定義しており、本研究では「プログラミング」と同じ意味を指す語として用いる。

「ソフトウェア」は「機械を操作する・技術（プログラム）。ソフト」であり「ハードウェア」の対義語として（p. 847）、「アプリケーション」は「アプリケーションソフト」の略語であり、「コンピュータで、文書作成・通信など、特定の作業をするためのソフトウェア。アプリケーション。アプリ」と示されている（p. 42）。このように、「ソフトウェア」、「アプリケーション」は、いずれもコンピュータなどのハードウェア上でなんらかの作業をするものを指すと捉えられるが、本論文では、金田一春彦・金田一秀穂の定義を踏まえ、「ソフトウェア」よりも範囲を狭めて示される「アプリケーション」を用いることにする。なお、引用において「ソフトウェア」や「ツール」などとして示されている場合

<sup>1</sup> 例えば、「小学校プログラミング教育の手引（第三版）」（文部科学省 2020a, p. 1）では、コンピュータが動作するための命令を「プログラム」、そして、命令を与えることを「プログラミング」としているが、コンピュータ以外の方法（例えば紙上に描かれる図など）を用いて記された問題解決の手順も「プログラム」と呼び、それを構築する行為を「プログラミング」と呼んでいる例もある（堀田 2017, 寺内 2022）。

には、「アプリケーション」ではなく，引用元の語に従って記す。

### 研究倫理を踏まえた配慮

---

本研究には，実際の小学校における授業実践が含まれる。そのため，授業実践にあたっては，研究倫理を踏まえ，次の点に配慮した。まず，本研究のために授業実践を行うことについて，対象児童の保護者より事前に許諾を得た。次に，児童のプライバシー保護のために，研究成果のとりまとめの際，個人の特典ができないように，実名を A 児や B 児と置き換えて整理した。

## 【第一部】理論編

### 1. 小学校段階におけるプログラミング教育

本章では、小学校段階におけるプログラミング教育について、導入に関わる議論から2022年3月までの動向（1.1.，1.2.，1.3.，1.4.，1.6.）と、文部科学省の示すプログラミング教育のねらい等（1.5.）を整理する。加えて、プログラミング教育に関わる研究の動向も整理する（1.7.）。

#### 1.1. 学校教育への導入をめぐる議論

本節では、小学校段階にプログラミング教育が必修化された動向を、これまでに文部科学省等によって公刊された資料、及び複数の先行研究を参照して検討し、プログラミング教育必修化までのタイムラインを図示する。

##### ① 「手引」（文部科学省 2020a）における経緯

小学校段階にプログラミング教育が導入されるまでの経緯について、「手引」（文部科学省 2020a, p. 8）には「小学校プログラミング教育導入の経緯」として、次の4資料とともに、その流れが簡略に示されている。

- (1) 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（以下、「有識者会議」）<sup>2</sup>（2016）による「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）」（以下、「議論の取りまとめ」）
- (2) 中央教育審議会（2016a）による「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）」（以下、「答申」）
- (3) 文部科学省（2018a）による『小学校学習指導要領（平成29年3月31日告示）』
- (4) 文部科学省（2018b）による『小学校学習指導要領解説 総則編（平成29年6月21日公表）』

このように、「手引」は「議論の取りまとめ」を、小学校段階におけるプログラミング教育導入の契機として位置付けている。

<sup>2</sup> 「有識者会議」は、2016年4月19日に、「小学校段階で育成すべき資質・能力と効果的なプログラミング教育の在り方」や「効果的なプログラミング教育を実現するために必要な条件整備等」などを検討するために設置された（文部科学省初等中等教育局教育課程課 2016a）。

## ② 「教育の情報化に関する手引」及び「情報教育に関連する資料」における経緯

プログラミング教育導入の契機を「議論の取りまとめ」（「有識者会議」 2016）とする資料は、「手引」（文部科学省 2020a）のほかにもみられる。

文部科学省は 2019 年 12 月に、2020 年度からの新学習指導要領の実施に備え、どのように情報活用能力を育むか等を解説した「教育の情報化に関する手引」（文部科学省 2019, pp. 55-56）を公表した<sup>3</sup>。そこでは、小学校段階におけるプログラミング教育の必修化にあたって、「答申」（中央教育審議会 2016a）と、その土台となる「議論の取りまとめ」（「有識者会議」 2016）を踏まえて、学習指導要領が改訂されたことが示されている。

他方、「手引」（文部科学省 2020a）には示されていない事項を取り上げたものもみられる。例えば、学習指導要領改訂にあたって 2015 年 8 月に設立された「中央教育審議会初等中等教育分科会教育課程部会情報ワーキンググループ」（以下、「情報ワーキンググループ」）は、議論にあたって「情報教育に関連する資料」（文部科学省生涯学習政策局情報教育課 2015a）を提示している。それによると、プログラミングの重要性が指摘されるものとして、「日本再興戦略 —JAPAN is BACK—（平成 25 年 6 月 14 日閣議決定）」、「世界再先端 IT 国家創造宣言（平成 25 年 6 月 14 日閣議決定）」、「教育再生実行会議第七次提言（平成 27 年 5 月 14 日）」が取り上げられている。これらは、いずれも「議論の取りまとめ」公表以前の議論である（文部科学省生涯学習政策局情報教育課 2015a, p. 15）。

## ③ プログラミング教育導入の経緯に関わる先行研究

次に、プログラミング教育導入の経緯についての先行研究を参照する。

山本・堀田（2021, pp. 38-39）は、学習に用いられるプログラミング言語や学習指導要領の動向を踏まえ、小学校段階におけるプログラミング教育の変遷を示している（表 2）。

表 2 小学校段階におけるプログラミング教育の変遷の区分と内容  
（山本・堀田（2021, pp. 38-39）をもとに筆者作成）

(1) 黎明期 1980 年～ 1999 年	子ども向けに開発されたプログラミング言語（例えば、LOGO や BASIC）による実践が多くみられる。一方で、課題点も多く、模索段階と位置付けられる。
(2) 過渡期 2000 年～ 2015 年	テキスト型プログラミング言語を用いた実践から、ビジュアル型プログラミング言語（例えば、Scratch）を用いた実践がみられるようになる。ほかにも、ロボット等のハードウェアを用いたプログラミングの実践がみられる。一方、学習指導要領においてはプログラミング教育の位置付けが明確ではなく、多くが教科外における試行的な実践となる。
(3) 試行期 2016 年～ 2019 年	小学校の学習指導要領の内容にプログラミング教育が明記され、「手引」も公表される。2020 年度からの学習指導要領の実施に向けて、「手引」等を踏まえた実践が多く実施される。
(4) 定着期 2020 年～	学習指導要領が実施され、小学校段階におけるプログラミング教育が必修化される。様々な教科等や教科外において実践が行われ、それらの実践報告などが公開される。

<sup>3</sup> 「教育の情報化に関する手引」について、2020 年 6 月に「教育の情報化に関する手引—追補版—（令和 2 年 6 月）」（文部科学省初等中等教育局学校デジタル化プロジェクトチーム 2020）が公表された。

阪東ら（2017, pp. 174-175）は、小学校段階におけるプログラミング教育の必修化までの経緯を示すために、まず、中等教育段階における必修化まで遡って、学習指導要領を踏まえながら、どのようなプログラミング教育が実施されたかを整理している（表 3）。

表 3 中等教育段階におけるプログラミング教育の変遷  
（阪東ら（2017, pp. 173-175）をもとに筆者作成）

1970年	『高等学校学習指導要領』（文部省 1970）	工業、商業、理数において、プログラミング教育に関する学習内容が示される。1970年に設けられた情報科はコンピュータ・サイエンス（CS <sup>4</sup> ）を専門とするものであり、その学習内容の中心は「プログラミングを通して、コンピュータの構造の理解を図ったり、ソフトウェアを開発したりすること」が目的のプログラミング教育であった。
1978年	『高等学校学習指導要領（昭和53年8月告示）』（文部省 1978）	工業、商業にはプログラミングについての記述があるが、理数では、プログラミングの文言が削除され、「CSの一領域にプログラミング教育が位置づけられるようになった」。
1989年	『中学校学習指導要領（平成元年3月告示）』（文部省 1989）	技術・家庭科に「コンピュータの操作等を通して、その役割と機能について理解させ、情報を適切に活用する基礎的な能力を養う」ための領域、情報基礎が新設された（ただし、必修ではない）。
1998年	『中学校学習指導要領（平成10年12月告示）』（文部省 1998）	技術・家庭科の技術分野の内容が「技術とものづくり」と「情報とコンピュータ」に分けられ、後者の内容において、プログラミング教育が実施された。
1999年	『高等学校学習指導要領（平成11年3月告示）』（文部省 1999）	情報A、情報B、情報Cから成る情報科が新設された。このうち情報Bにおいて、プログラミング教育が実施された。
2008年	『中学校学習指導要領（平成20年3月告示）』（文部科学省 2008）	技術・家庭科の技術分野の「D情報に関する技術」においてプログラミング教育が実施された。
2009年	『高等学校学習指導要領（平成21年3月告示）』（文部科学省 2009）	情報科の内容が、情報A、情報B、情報Cから、「社会と情報」と「情報の科学」の2つへと改編された。プログラミング教育は、「情報の科学」の「問題解決とコンピュータの活用」において実施された。

阪東らは、この学習指導要領におけるプログラミング教育の変遷を踏まえて、小学校段階にプログラミング教育が導入されるまでの経緯を、「我が国の初等教育にプログラミング教育が導入されるようになった経過」として整理した（阪東ほか 2017, p. 175）（表 4）。

<sup>4</sup> CSは「Computer Science」（計算機科学）を指す。

表 4 「我が国の初等教育にプログラミング教育が導入されるようになった経過」  
(阪東ほか 2017, p. 175)

年	月	文書・資料等	プログラミング教育に関連する主な内容
2013	4	第 6 回産業競争力会議	義務教育段階でのプログラミング教育推進の方針
	6	日本再興戦略 —JAPAN is BACK—	
2016	4	第 26 回産業競争力会議	小学校における体験的に学習する機会の確保, 中学校におけるコンテンツに関するプログラミング学習, 高等学校における情報科の共通必修修科目化といった, 発達の段階に則した必修化の方針
	6	日本再興戦略 2016 —第 4 次産業革命に向けて—	
	6	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, 有識者会議における議論の取りまとめ	
	12	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)	小学校におけるプログラミング教育の導入, 中学校におけるプログラミングに関する内容の倍増, 高等学校における共通必修修科目「情報 I」の新設によるプログラミングの必修化など, 小中高を通じた体系的なプログラミング教育の方針
2017	3	小学校学習指導要領 (平成 29 年公示)	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を実施すること, 「算数」, 「理科」, 「総合的な学習の時間」におけるプログラミング教育の扱い方
	3	中学校学習指導要領 (平成 29 年公示)	双方向性のあるコンテンツのプログラミングと計測・制御におけるプログラミングについて実生活や社会との関わりを意識した学習活動を実施すること
2018		高等学校学習指導要領 (平成 30 年公示) 予定	共通教科「情報」の「情報 I」を共通必修修科目とし, プログラミング及びモデル化やシミュレーション等を扱うこと (予定)

阪東ら (2017, p. 175) が, プログラミング教育導入の経緯に「議論の取りまとめ」や「答申」を含めている点は, 「手引」の示す「小学校プログラミング教育導入の経緯」(文部科学省 2020a, p. 8) と同様である。しかし, 小学校段階におけるプログラミング教育導入の契機を「第 6 回産業競争力会議」(首相官邸 2013a) に位置付けている点で, 「手引」や「教育の情報化に関する手引」(文部科学省 2019) と異なる<sup>5</sup>。さらに, 「情報教育に関連する資料」(文部科学省生涯学習政策局情報教育課 2015a) と同様に「日本再興戦略」を経緯に含めている。阪東らはそれらの内容については詳述しておらず, また阪東らの研究が「手引」が公開される 2018 年以前だったため, 「手引」の内容についても言及されていない。

<sup>5</sup> 「産業競争力会議」をプログラミング教育必修化の契機として位置付けているものとして, ほかにも, つくば市教育局総合教育研究所 (2019, p. 2) や東京学芸大学プログラミング教育研究会 (2019, pp. 28-35) がみられる。ただし, つくば市教育局総合教育研究所は「第 26 回産業競争力会議」(首相官邸 2016a) を, 東京学芸大学プログラミング教育研究会は「第 27 回産業競争力会議」(首相官邸 2016b) を契機としている点で異なる。

以上が、プログラミング教育の学校教育への導入をめぐる議論に関わる資料や先行研究である。ここでは、これらを踏まえ、「産業競争力会議」を開催した「日本経済再生本部」の設置を変遷の契機として位置付け、小学校段階におけるプログラミング教育の導入をめぐる議論の経緯を図1にタイムラインに沿って示す。

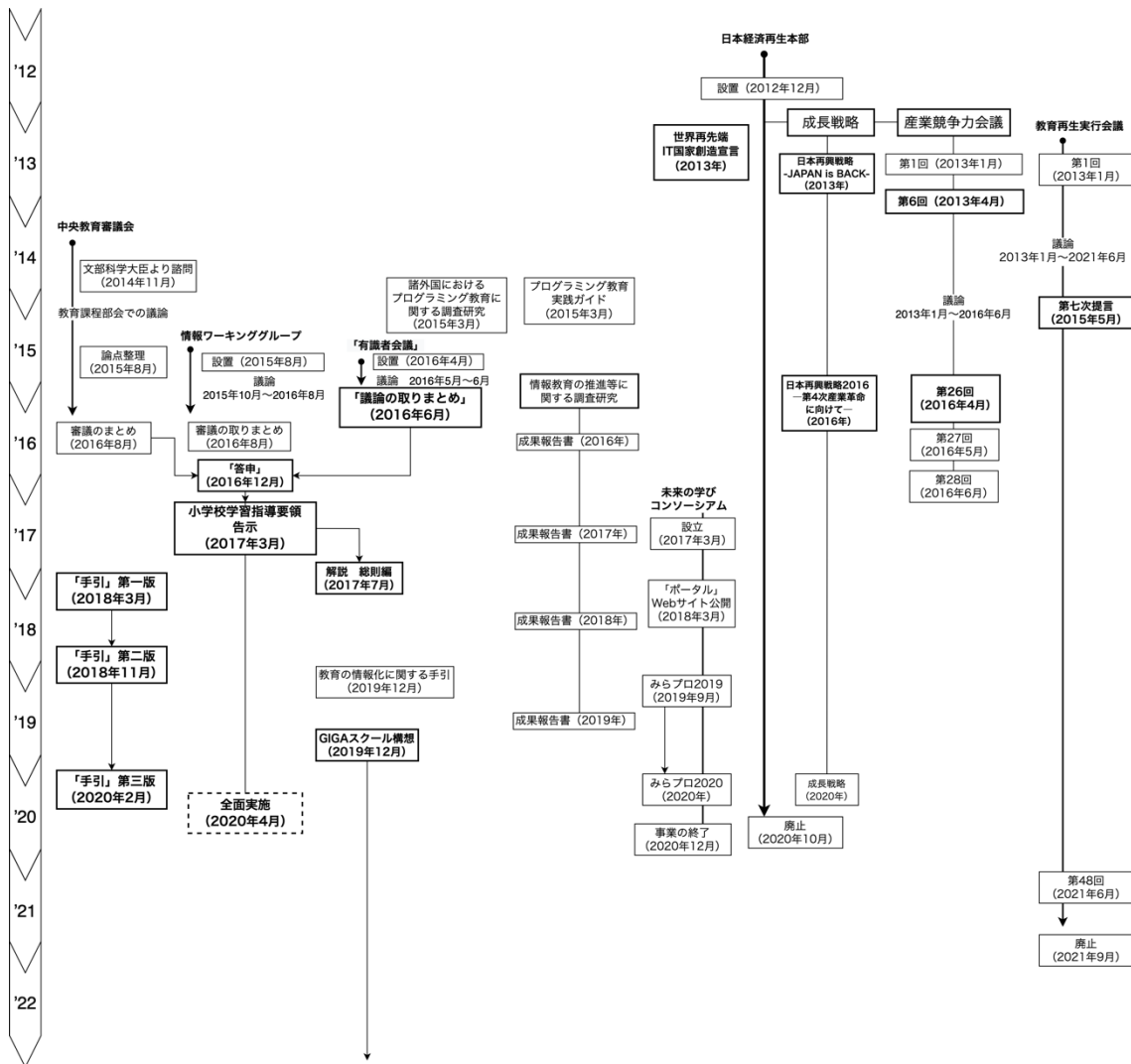


図1 小学校段階におけるプログラミング教育の導入をめぐる議論の経緯を示したタイムライン

タイムラインの左側には文部科学省による動向を示し、タイムラインの右側に行くにつれ産業・経済界の動きを示した。文部科学省・経済産業省・総務省が一体となって運営された「未来の学びコンソーシアム」は、文部科学省と政・経済界の動きの間に位置付けた。また、太字等で強調した出来事や公開資料は、プログラミング教育の必修化や実施に、とりわけ大きな関わりをもつものとして示している。さらに、そのなかでも、最も大きな出



来事である、小学校段階におけるプログラミング教育の必修化を定めた「小学校学習指導要領（2017年3月告示）」の告示は、他の動きよりも大きく図示している。

このように、文部科学省、産業・経済界を視点としてタイムラインを整理すると、プログラミング教育に関わる議論等の動きは、「議論の取りまとめ」を転換点として、産業・経済界主導から文部科学省主導へと移り変わっている様子が窺える。そこで、次節からは「議論の取りまとめ」を中心に取り上げ、その公表以前（1.2.）、その内容（1.3.）、その公表以降（1.4.）の3期に分類し、プログラミング教育の必修化にあたって要点となる議論等を、タイムラインに沿って詳述する。

## 1.2. 「議論の取りまとめ」以前の動向

本節では、「議論の取りまとめ」（2016年6月）以前にみられる小学校段階におけるプログラミング教育の必修化に関わる動きのうち、次の8点を取り上げる。

- ① 「日本経済再生本部」の設置（2012年12月）
- ② 「第6回産業競争力会議」（2013年4月）及び「日本再興戦略 —JAPAN is BACK—」（2013年6月）
- ③ 「世界最先端IT国家創造宣言」（2013年6月）
- ④ 「諸外国におけるプログラミング教育に関する調査研究」（2015年3月）
- ⑤ 「教育再生実行会議」による「教育再生実行会議第七次提言」（2015年5月）
- ⑥ 「情報ワーキンググループ」による議論（2015年10月～2016年5月）
- ⑦ 「プログラミングに関する教育の推進」（2015年10月）
- ⑧ 「第26回産業競争力会議」（2016年4月）

次に、それぞれの内容を示す。

### ① 「日本経済再生本部」の設置（2012年12月）

まず、小学校プログラミング教育導入の契機として位置付けた「日本経済再生本部」と、それによる「成長戦略」と「産業競争力会議」について整理する。

「日本経済再生本部」とは、2012年12月に、日本の経済成長のために首相官邸が設置したものである<sup>6</sup>。「日本経済再生本部」によって、「産業競争力会議」の開催と「成長戦略」の閣議決定が行われる。

「産業競争力会議」とは、日本の「産業の競争力強化や国際展開に向けた成長戦略の具

---

<sup>6</sup> 「日本経済再生本部」は、本部長を内閣総理大臣とし、副総理（本部長代理）、経済再生担当大臣兼内閣府特命担当大臣（経済財政政策）・内閣官房長官（副本部長）、そして、他の全ての国务大臣（本部員）らによって構成される。また、必要に応じて、関係者の出席が認められる（首相官邸 2012）。2020年10月16日の閣議決定において廃止された（首相官邸 2020a）。

現化と推進について調査審議するため」に、「日本経済再生本部」によって開催された（首相官邸 2013b）。

「成長戦略」は、2013年6月に閣議決定された「日本再興戦略 —JAPAN is BACK—」から始まった。成長戦略の一覧を表5に示す。

表5 2022年3月時点における「成長戦略」一覧  
（首相官邸（2020b）をもとに筆者作成）

2013年	日本再興戦略 —JAPAN is BACK—
2014年	日本再興戦略改訂 2014 —未来への挑戦—
2015年	日本再興戦略改訂 2015 —未来への投資・生産性革命—
2016年	日本再興戦略 2016 —第4次産業革命に向けて—
2017年	未来投資戦略 2017 —Society 5.0の実現に向けた改革—
2018年	未来投資戦略 2018 —「Society 5.0」, 「データ駆動型社会」 への変革—
2019年	成長戦略 (2019年)
2020年	成長戦略 (2020年)

## ② 「第6回産業競争力会議」(2013年4月)及び「日本再興戦略 —JAPAN is BACK—」(2013年6月)

小学校段階にプログラミング教育が導入される契機として、阪東ら（2017, p. 175）は「産業競争力会議」の1つである「第6回産業競争力会議」と、それに続けて「成長戦略」の1つである「日本再興戦略 —JAPAN is BACK—」を示している。

「第6回産業競争力会議」は2013年4月17日に開催された。議事録によると、三木谷（当時、楽天株式会社代表取締役会長兼社長）が、日本の今後の経済成長の鍵として、優秀なエンジニアを養成すること、そのためにプログラミング教育を取り入れる必要があることを述べている。

「日本再興戦略 —JAPAN is BACK—」（首相官邸 2013c）は「第6回産業競争力会議」を受けて、2013年6月14日に閣議決定された。「日本再興戦略 —JAPAN is BACK—」は、「日本産業再興プラン」の1つとして「世界最高水準のIT社会の実現」を示し（p. 42）、そのために「産業競争力の源泉となるハイレベルなIT人材の育成・確保」が必要だとしている。プログラミング教育は、「ITを活用した21世紀型スキルの修得」において、その手立ての1つとして示されている（p. 46）。なお、「ITを活用した21世紀型スキルの修得」では、義務教育段階にプログラミング教育を取り入れるだけでなく、1人1台のICT端末の整備といった、後のGIGAスクール構想（文部科学省 2022a）へとつながる方針も定めている。

GIGAスクール構想については、児童それぞれにICT端末を用意したり、学校のネットワーク環境を整備したりするという点でプログラミング教育との関わりが指摘できるが、その設置は「議論の取りまとめ」の公表以後の2019年となる（文部科学省初等中等教育局初等中等教育企画課 2019）。よって、GIGAスクール構想はプログラミング教育の必修化に直接関連はしていないため、その内容はプログラミング必修化後の動向として、1.6.

で整理する。

### ③ 「世界最先端 IT 国家創造宣言」(2013 年 6 月)

「日本再興戦略—JAPAN is BACK—」が公表された同 2013 年、「世界最先端 IT 国家創造宣言」(首相官邸 2013d)が公表された。「世界最先端 IT 国家創造宣言」では、日本の経済再生にあたり、「あらゆる領域に活用される万能ツールとして、イノベーションを誘発する力」を有す情報通信技術 (IT) を発展させること (p. 1), そのために IT を扱うことができる人材の育成の重要性が述べられている (p. 20)。IT が扱える人材の育成・教育にあたっては、「①教育環境自体の IT 化<sup>7)</sup>」, 「②国民全体の IT リテラシーの向上」, 「③国際的にも通用・リードする実践的な高度な IT 人材の育成」の 3 点が示されており、このうちの「③国際的にも通用・リードする実践的な高度な IT 人材の育成」では、プログラミング教育が触れられており、実践の中で技術を習得させること、そのために初等・中等段階からプログラミング教育に取り組むことが必要とされている (pp. 21-22)。

このように、小学校段階におけるプログラミング教育の必修化の背景には、IT 人材を確保するという、政界や経済界からの要請があったことがわかる。

### ④ 「諸外国におけるプログラミング教育に関する調査研究」(2015 年 3 月)

文部科学省は、国内でのプログラミング教育の必修化に先立って、「諸外国におけるプログラミング教育に関する調査研究」(以下、「調査研究」)(文部科学省 2015a)を行っている。そこでは、今後の日本の小学校段階において、プログラミング教育をどのように導入するかを検討するための手がかりとして (p. 7), 23 の国・地域<sup>8)</sup>を対象に実態を取りまとめている。

「調査研究」では、学習者の発達段階を日本における小学校・中学校・高等学校にあたる公立学校に設定している。そして、それぞれの国・地域について、13 の調査項目<sup>9)</sup>を設け、その結果を示している。太田ら (2016, p. 198) は、「調査研究」(文部科学省 2015a)をもとに、各国のプログラミング教育の現状を「諸外国のプログラミング教育を含む情報教育の状況」として整理した。それによると、小学校段階からプログラミング教育を必修化している国は、英国 (イングランド)、オーストラリア、フィンランド、ロシア、インドの 5 ヶ国であり、それぞれ、“Computing” (英国) などといった教科名で実施されてい

---

<sup>7</sup> 1 人 1 台の端末の整備が求められていることから、GIGA スクール構想との関わりがみられる。

<sup>8</sup> 対象の国・地域は、英国 (イングランド)、エストニア、フランス、ドイツ、フィンランド、イタリア、スウェーデン、ハンガリー、ポルトガル、ロシア、アメリカ合衆国 (カリフォルニア州)、カナダ (オンタリオ州)、アルゼンチン、韓国、シンガポール、上海、香港、台湾、インド、イスラエル、オーストラリア、ニュージーランド、南アフリカである。

<sup>9</sup> 調査項目は、①背景、②目的、③位置付け、④目標、⑤指導内容・プログラミング言語、⑥教材、⑦指導者、⑧評価方法、⑨試験科目として課されているか、⑩基礎的な ICT リテラシー教育、⑪大学・NPO・民間、⑫成果、⑬解決すべき問題点から成る。

る。

#### ⑤ 「教育再生実行会議」による「教育再生実行会議第七次提言」（2015年5月）

首相官邸による政策会議「教育再生実行会議」<sup>10</sup>では、その提言の1つ「教育再生実行会議第七次提言」（首相官邸 2015, p. 8）において、「ICT活用による学びの環境の革新と情報活用能力の育成」にあたり、情報活用能力の育成の1つとしてプログラミング教育の実施が挙げられた<sup>11</sup>。

前述した「日本再興戦略—JAPAN is BACK—」, 「世界再先端 IT 国家創造宣言」が、国の経済再生や発展を目指していたことに対し、「教育再生実行会議第七次提言」は、今後訪れる予測困難な時代でも求められる資質・能力を子どもに育むことに主眼が置かれており、それぞれの性格の違いが顕れている。

#### ⑥ 「情報ワーキンググループ」による議論（2015年10月～2016年5月）

2015年8月、学習指導要領改訂にあたって「情報ワーキンググループ」が設置された<sup>12</sup>（文部科学省 2015b）。設置の目的は、新学習指導要領に向けて、社会のあらゆるところで情報化が進んでいることを背景に、児童の情報活用能力のあり方やその育み方を検討することである（文部科学省生涯学習政策局情報教育課 2015b）。

「情報ワーキンググループ」による議論は、2015年10月から2016年5月までに8回行われ、それら議論を踏まえて2016年8月26日に「教育課程部会 情報ワーキンググループにおける審議の取りまとめについて（報告）」（文部科学省生涯学習政策局情報教育課 2016a）が公表された。その一覧を表6に示す。

表6 情報ワーキンググループの議事開催日等の一覧  
（文部科学省（2016a）をもとに筆者作成）

2015年10月22日	第1回
2015年11月24日	第2回
2015年12月22日	第3回
2016年1月20日	第4回
2016年2月23日	第5回
2016年3月15日	第6回
2016年4月20日	第7回
2016年5月18日	第8回
2016年8月26日	「教育課程部会 情報ワーキンググループにおける審議の取りまとめについて（報告）」の公表

<sup>10</sup> 「教育再生実行会議」は、内閣総理大臣が開催し、内閣総理大臣、内閣官房長官、文部科学大臣兼教育再生担当大臣、そして有識者によって構成される。2013年1月より2021年6月までに、48回開催され（首相官邸 2021a）、2021年9月17日の閣議決定において廃止された（首相官邸 2021b）。

<sup>11</sup> これらを実施するために、1人1台のICT端末の整備等が求められていることを示していることから、GIGAスクール構想との関連も窺える（首相官邸 2015, p. 8）。

<sup>12</sup> 「中央教育審議会教育課程部会」による「教育課程企画特別部会」の下に設置された。

「情報ワーキンググループ」では、プログラミング教育を小学校段階においても実施し、小・中学校段階と高等学校情報科との接続を確かなものにしたり、高等学校の情報科を充実させたりすることが議論されている（文部科学省生涯学習政策局情報教育課 2015c）。これらの議論が、中央教育審議会の「審議のまとめ」へとつながることになる（文部科学省初等中等教育局教育課程課 2016b, p. 1）。

### ⑦ 「プログラミングに関する教育の推進」（2015年10月）

「情報ワーキンググループ」の第1回（2015年10月22日）の配付資料「情報教育に関する資料」では、「プログラミングに関する教育の推進」（図2）が示されている（文部科学省生涯学習政策局情報教育課 2015a, p. 30）。

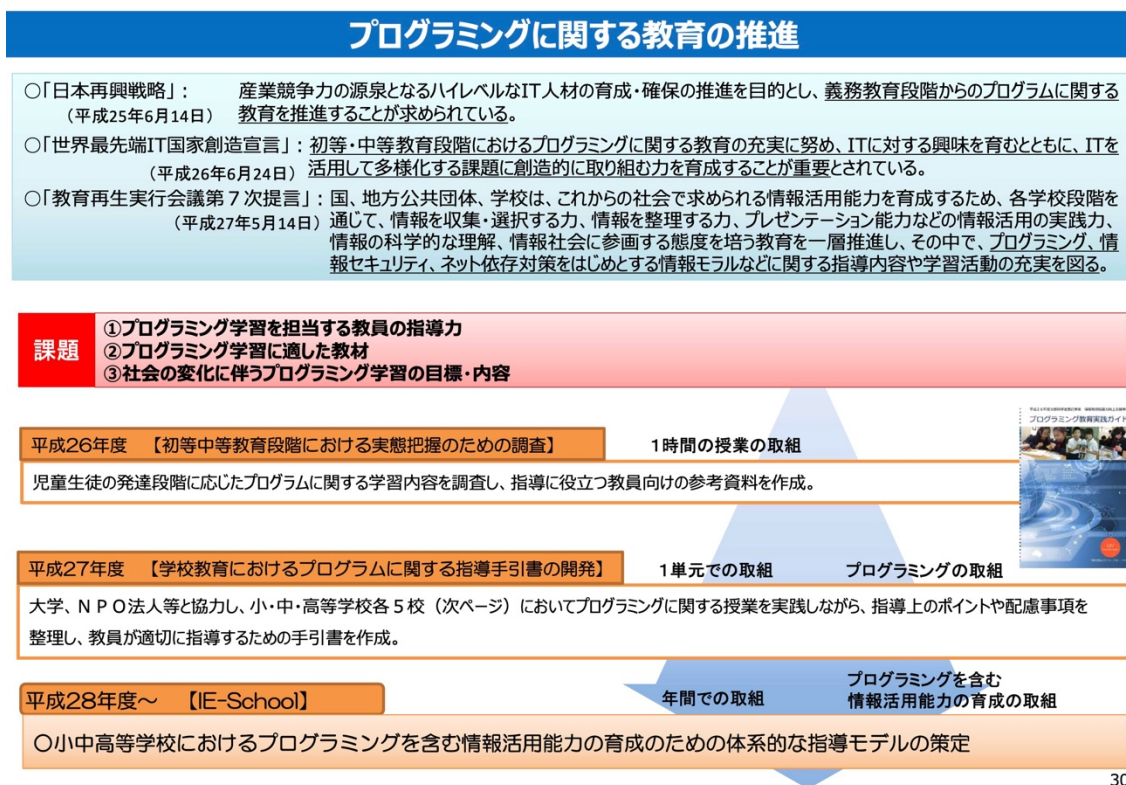


図2 「プログラミングに関する教育の推進」

（文部科学省生涯学習政策局情報教育課 2015a, p. 30）

「プログラミングに関する教育の推進」では、プログラミング教育の実施の前段階として、「日本再興戦略」（2013年）、「世界最先端IT国家創造宣言」（2014年）、「教育再生実行会議第7次提言」（2015年）が位置付けられている。そして、プログラミング教育実施における課題として「①プログラミング学習を担当する教員の指導力」、「②プログラミング学

習に適した教材」, 「③社会の変化に伴うプログラミング学習の目標・内容」の3点が示されている。

そして、これらの課題に対して、2014年度に「初等中等教育段階における実態把握のための調査」、2015年度に「学校教育におけるプログラムに関する指導手引書の開発」、2016年度に「IE-School」（小中高等学校におけるプログラミングを含む情報活用能力の育成のための体系的な指導モデルの策定）といった3つの取り組みを挙げている。「プログラミングに関する教育の推進」では、それらの取り組みの成果は示していないが、筆者が調べられる限り、『プログラミング教育実践ガイド』（文部科学省 2016b）の公開や、「IE-School（情報教育推進校）」による成果報告書（文部科学省初等中等教育局学校デジタル化プロジェクトチーム 2019）が位置付けられると考えられる。

#### ⑧ 「第26回産業競争力会議」（2016年4月）

小学校段階におけるプログラミング教育の必修化について、最初に報道発表されたのは2016年4月19日である（日本経済新聞 2016-04-19, 朝刊）<sup>13</sup>。同日に開催された「第26回産業競争力会議」（首相官邸 2016a）において、安倍内閣総理大臣（当時）が、「進化した人工知能が様々な判断を行ったり、身近な物の働きがインターネット経由で最適化されたりする時代」（「有識者会議」 2016）である「第4次産業革命」の時代を生き抜く力を育むために、初等中等教育からプログラミング教育を必修化することを述べ、小・中・高等学校における発達段階に即したプログラミング教育の必修化の方針が定められた（首相官邸 2016a, p. 12）。

以上、「日本経済再生本部」設置から「第26回産業競争力会議」までの変遷を整理した。これらを見ると、プログラミング教育の必修化が産業・経済界によって進められていた段階におけるプログラミング教育のねらいと、「手引」が示しているプログラミング教育のねらいとでは、その性質が異なることが分かる。阪東ら（2017, p. 177）が、「産業・経済界の要請が強く、十分に初等中等教育間の接続が熟慮された設計とはいえない」ことを指摘しているように、はじめ、プログラミング教育必修化のねらいとは小学校段階からIT人材を育成することであったが、実施までの議論を経て、身の回りのITを活用できる力を児童に促すことへと移り変わっていったのである。

次節では、「有識者会議」が、小学校段階において、どのようにプログラミング教育を実施するかを検討した「議論の取りまとめ」について取り上げる。

---

<sup>13</sup> 2021年3月16日に、日本経済新聞のデータベース「日経テレコン 21（日本経済新聞デジタルメディア）」にて、「小学校」、「プログラミング教育」、「必修」をキーワードにして、検索を行った。

### 1.3. 「有識者会議」による「議論の取りまとめ」(2016年6月)

「議論の取りまとめ」は、「有識者会議」が3度の議論<sup>14</sup>を通して、「『第4次産業革命』が教育に与える影響」、「小学校段階で育成すべき資質・能力と効果的なプログラミング教育の在り方」、「効果的なプログラミング教育を実現するために必要な条件整備等」などを検討し(文部科学省初等中等教育局教育課程課2016a)、2016年6月16日に公表されたものである。

本節では、「議論の取りまとめ」(「有識者会議」2016)が示す、小学校段階においてプログラミング教育を実施する背景、小学校段階においてプログラミング教育を実施する目的、及び各教科等におけるプログラミング教育の実践案を整理する。

#### 1.3.1. プログラミング教育を実施する背景

「議論の取りまとめ」(「有識者会議」2016)では、プログラミング教育が必修化される背景として、急速に情報技術が発展し、発達する第4次産業革命といった、将来の予測が困難な時代を生きるための資質・能力を育まなければならないことを挙げている。そして、その資質・能力とは、全く新しい力ではなく、従来から重視されている読解力、論理的・創造的思考力、問題解決力、人間性等について、「加速度的に変化する社会の文脈の中での意義」を改めて捉え直したものであり、教育課程全体を通して実現することが「強く求められる」としている。それら資質・能力は次の3点である。

- ① 情報を読み解くこと
- ② 情報技術を手段として使いこなしながら、論理的・創造的に思考して課題を発見・解決し、新たな価値を創造すること
- ③ 感性を働かせながら、よりよい社会や人生の在り方について考え、学んだことを生かそうとすること

(「有識者会議」2016)

なお、資質・能力の②「情報技術を手段として使いこなしながら、論理的・創造的に思考して課題を発見・解決し、新たな価値を創造すること」には、コンピュータを用いた活動が含まれている。

このように、「有識者会議」によるプログラミング教育の実施の背景として、現在、コンピュータなどのICT機器が、生活の至るところにあること、そして、それら情報機器が、今後、社会の移り変わりとともにどのように変化しても、自身のより良い生き方のために用いることができる資質・能力を育むことが求められていることが挙げられる。

<sup>14</sup> 第1回が2016年5月13日(文部科学省初等中等教育局教育課程課2017a)、第2回が2016年5月19日(文部科学省初等中等教育局教育課程課2017b)、第3回が2016年6月3日(文部科学省初等中等教育局教育課程課2017c)に実施された。

### 1.3.2. 小学校段階におけるプログラミング教育の目的

文部科学省は「有識者会議」の設置にあたって、プログラミング教育のねらいを「論理的思考力や創造性、問題解決能力といった資質・能力を育む」ことと、「コンピューターを動かすために必要なコーディング（プログラミング言語を用いた記述方法）を学ぶ」こととに分けて示した（文部科学省初等中等教育局教育課程課 2016a）。その後公表された「議論の取りまとめ」（「有識者会議」 2016）では「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」こと、そして、実施にあたって「コーディングを覚えることが目的ではないことを明確に共有していくことが不可欠」であること、すなわちプログラミングの知識・技能を身に付けさせることがねらいではないことが明示された。

このようにプログラミング教育のねらいは、「有識者会議」による議論を経て、文部科学省の示す2つのねらいのうち「論理的思考力や創造性、問題解決能力といった資質・能力を育む」ことに焦点化された。

### 1.3.3. 各教科等におけるプログラミング教育の実践案

「議論の取りまとめ」（「有識者会議」 2016）は、総合的な学習の時間、理科、算数、音楽、図画工作、特別活動におけるプログラミング教育の実践案を示している。それらの実践案をみると、いずれの教科等においても、どの学習でプログラミング教育を取り上げるかについては示されているが、どのようにプログラミング教育を行うのかといった具体例は示されていない。

1.3. では、「議論の取りまとめ」における、プログラミング教育を実施する背景と目的、そして、各教科等におけるプログラミング教育の実践案を確認した。それらを踏まえ、「議論の取りまとめ」の特質として次の2点が挙げられる。

- ① プログラミング教育のねらいとして、産業界を担う人材を育成することではなく、児童の論理的思考力や創造性、問題解決能力といった資質・能力を育むことに焦点が当てられていること
- ② プログラミング教育の実施にあたって、具体的な活動事例が示されていないこと

### 1.4. 「議論の取りまとめ」以降の動向

本節では、「議論の取りまとめ」（2016年6月）公表後、小学校学習指導要領にプログラミング教育の必修化が明示され、そのあり方が示されるまでの議論等を整理する。具体的には次の事柄を取り上げる。

- ① 「情報ワーキンググループにおける審議の取りまとめ（報告）」及び「次期学習指導要



領等に向けたこれまでの審議のまとめ」(2016年8月)

② 中央教育審議会による学習指導要領の改訂に向けた「答申」(2016年12月)

③ 「未来の学びコンソーシアム」(2017年3月～)

④ 「小学校学習指導要領(平成29年告示)」(2017年3月)

⑤ 「小学校プログラミング教育の手引 第一版」(2018年3月)

① 「情報ワーキンググループにおける審議の取りまとめについて(報告)」及び「次期学習指導要領等に向けたこれまでの審議のまとめ」(2016年8月)

「情報ワーキンググループ」は、2016年8月26日に、「教育課程部会 情報ワーキンググループにおける審議の取りまとめについて(報告)」(文部科学省生涯学習政策局情報教育課 2016a)を公表した。同日、中央教育審議会初等中等教育分科会教育課程部会は、各ワーキンググループの議論を踏まえ、次期学習指導要領の方針を示した「次期学習指導要領等に向けたこれまでの審議のまとめ」(以下、「審議のまとめ」)(文部科学省教育課程部会 2016a)を公表した。プログラミング教育については、「情報活用能力(情報技術を手段として活用する力を含む)の育成」において、児童の〈プログラミング的思考〉を育むことが示されている(文部科学省教育課程部会 2016b, p. 37)。

② 中央教育審議会による学習指導要領の改訂に向けた「答申」(2016年12月)

「手引」(文部科学省 2020a, p. 8)によると、中央教育審議会は「議論の取りまとめ」(「有識者会議」 2016)を土台として学習指導要領改訂に向けた議論を行い、2016年12月に「答申」(中央教育審議会 2016a)を公表した。

「答申」では、「議論の取りまとめ」を踏まえプログラミング教育について詳述し、また、「情報活用能力」の1つとして〈プログラミング的思考〉を取り上げ(中央教育審議会 2016a, pp. 37-38)、プログラミング教育の目的として児童に〈プログラミング的思考〉を身に付けさせること(中央教育審議会 2016b, p. 15)、加えて、「STEM (Science, Technology, Engineering and Mathematics)教育の視点を踏まえ」、理科、算数科、総合的な学習の時間、芸術系教科における指導内容のイメージが示された(中央教育審議会 2016a, p. 91)。芸術系教科における指導内容のイメージとは、小学校音楽科の音楽づくりを通してプログラミングを体験させることである。その留意点として、音楽科の学びに照らし合わせたりすることや、つくった音楽を自ら表現したりすることが挙げられている(p. 165)。

このように、中央教育審議会の「答申」では、プログラミング教育を通して〈プログラミング的思考〉を育むこと、そして、小学校音楽科の音楽づくり活動におけるプログラミング教育の可能性を示唆している。一方で、具体的な活動内容が例示されていない点は、「議論の取りまとめ」と同じである。

### ③ 「未来の学びコンソーシアム」の設立（2017年3月）

小学校段階におけるプログラミング教育の必修化に備え、その円滑な実施をねらって、文部科学省・総務省・経済産業省が協力し「未来の学びコンソーシアム」が設立された（総務省 2017, 文部科学省初等中等教育局情報教育・外国語教育課 2018a）。

「未来の学びコンソーシアム」は、2018年3月に「小学校を中心としたプログラミング教育ポータル」（以下、「ポータル」）Webサイトを公開した（未来の学びコンソーシアム 2018, p. 3）。「ポータル」は、リニューアルや更新を経て、プログラミング教育の「実施事例」を「手引」が示す分類（文部科学省 2020a, p.23）（表 7）に沿って掲載したり、プログラミング教育に活かせる「教材情報」を掲載したりしている。

表 7 「手引」における小学校段階におけるプログラミング教育の学習活動の分類  
（文部科学省（2020a, p. 23）をもとに筆者作成）

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	教育課程内で各教科等とは別に実施するもの <sup>15</sup>
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の学習機会

実践事例は、2018年には26件、続けて2019年には50件追加された（未来の学びコンソーシアム 2020a）。2020年2月に公開された「手引」においても、「ポータル」の活用が記されている（文部科学省 2020a, p. 3）。

加えて、「未来の学びコンソーシアム」は民間企業と連携し、総合的な学習の時間におけるプログラミング教育の指導案等の配信などを行う「みらプロ 2019」、「みらプロ 2020」を公開し、実施事例や指導案を掲載した（未来の学びコンソーシアム 2020b, 2020c）。

その後、「未来の学びコンソーシアム」は2020年12月25日に「全国での小学校プログ

<sup>15</sup> 「第一版」（文部科学省 2018c, p. 19）では「各学校の裁量により実施するもの（A, B及びD以外で、教育課程内で実施するもの）」と示されている。

プログラミング教育の円滑な実施を支援するという役割を達成した<sup>16</sup>。

「ポータル」は文部科学省が一時的に運営を引き継ぎ（文部科学省 2020b）、「ポータル」における「学校における小学校プログラミング教育の実施レポート（公開開始日：2020年11月9日）」（文部科学省 2022b）は、引き続き「小学校プログラミング教育の実施レポート様式」を公開し、実施事例を収集しながら、更新を続けている。「みらプロ」については「協力企業のうち一部企業が独自で同様の取組を実施」しているが、2021年以降の取組について文部科学省は把握していないと述べている（文部科学省 2021）。

以上、必修化以後の官民連携による取組と、その変遷について整理した。

#### ④ 学習指導要領の告示（2017年3月）

中央教育審議会の「答申」（中央教育審議会 2016a）を受けて、2017年3月31日に『小学校学習指導要領』（文部科学省 2018a）が告示された。プログラミング教育は「第1章 総則」の「第3 教育課程の実施と学習評価」（3）情報活用能力を育む事項において、各教科等の特質に応じて計画的に実施する活動に、「イ 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」として示されている（p. 22）。加えて、算数科（p. 92）、理科（p. 110）、総合的な学習の時間（p. 182）の「指導計画の作成と内容の取扱い」においてプログラミング教育の活動案が示されている。

学習指導要領では、小学校段階においてプログラミング教育を実施する背景として、日常生活の中に、スマートフォンやタブレット端末をはじめとする様々な ICT 機器が浸透していること、そして、今後情報技術がどのように発展していくかは予測しがたいが、様々な情報機器やサービス、情報を、あらゆる事態において適切に選択・活用できるようになる必要があることが挙げられている（文部科学省 2018b, p. 84）。そして、それらの力を身に付ける上で肝要なことは、児童が情報を主体的に捉え、それを他者との協働を通して活用しながら、新しい価値の創造を行えるようになることだと述べられている（p. 84）。プログラミング教育はそれらの力、つまり、情報活用能力を身に付けるための手立ての1つとして位置付けられている。そのねらいは、「プログラミング言語を覚えたり、プログラ

---

<sup>16</sup> 「未来の学びコンソーシアム」業務の終了については、2020年12月18日から22日にメール審議で行われた「第8回運営協議会」において議論されたと思われるが、Web上に公開されている「議事概要」（未来の学びコンソーシアム 2020d）では、「未来の学びコンソーシアムの運営に関する事項について（審議）」に対して「計20名の承認があり、本件は承認された」としか記されていないため、議論の内容は明らかではない。『未来の学びコンソーシアム』事務局の活動内容においては、「市町村教育委員会における小学校プログラミング教育に関する取組状況等調査」（未来の学びコンソーシアム 2020a, p. 2）として、1,746校に対してプログラミング教育を実施する環境の悉皆調査（2019年11月）とフォローアップ（2020年1～3月）の結果を示している。それによると、悉皆調査では「整っていない」と回答した114校が、フォローアップでは「整っている」、「令和2年度までには整う予定」と環境が改善されたと回答している。これらのことより、文部科学省が「円滑な実施を支援するという役割を達成した」と判断したと思われる。

ミングの技能を習得したりといったことではなく」、プログラミングを体験することを通して、「論理的思考力を育むとともに、プログラムの働きやよさ、情報社会がコンピュータをはじめとする情報技術によって支えられていることなどに気付き、身近な問題の解決に主体的に取り組む態度やコンピュータ等を上手に活用してよりよい社会を築いていこうとする態度などを育むこと、さらに、教科等で学ぶ知識及び技能等をより確実に身に付けさせること」とされている (p. 85)。

プログラミング教育を実施する教科は、学習指導要領の「指導計画の作成と内容の取扱い」において示された算数科や理科、総合的な学習の時間だけでなく、「例示以外の内容や教科等においても、プログラミングを学習活動として実施することが可能」であり、「プログラミングに取り組むねらいを踏まえつつ、学校の教育目標や児童の実情等に応じて工夫して取り入れていくことが求められる」とされている (文部科学省 2018b, p. 86)。

#### ⑤ 「小学校プログラミング教育の手引」の公開と改訂 (2018年3月～2020年2月)

文部科学省は 2018 年 3 月に「小学校プログラミング教育の手引 (第一版)」(文部科学省 2018c) を公開した。そのねらいは、小学校の教員がプログラミング教育に対して抱く不安を解消し、安心して取り組むことができるように、プログラミング教育についての基本的な考え方などを解説するところにある。「手引」は、「第一版」が公開されて以降 2 度改訂され、2018 年 11 月に「第二版」(文部科学省 2018d)、そして、2020 年 2 月に「第三版」(文部科学省 2020a) が公開された。なお、本研究で「手引」を参照する際には、特に明示していない場合、最新の「第三版」を取り上げる。

「手引」(文部科学省 2020a) の内容は、大きく「プログラミング教育の導入の経緯」(pp. 8-10)、「小学校段階におけるプログラミング教育のねらいと評価」(pp. 11-22)、「小学校段階におけるプログラミング教育の実施案」(pp. 23-57)に分けられる。ほかにも、企業や団体、地域との連携 (pp. 52-57) や、プログラミング教育のための環境 (pp. 58-64) について解説している。次に、これらの内容について概観するが、「プログラミング教育の導入の経緯」については、既に 1.1. の内容と重複するため、ここでの説明は割愛する。

まず、「手引」におけるプログラミング教育のねらいについて確認する。「手引」は、『学習指導要領解説 総則編』(文部科学省 2018b) を踏まえて、次の 3 点を示している。

- 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

(文部科学省 2020a, p. 11)

①の〈プログラミング的思考〉について、「有識者会議」(2016)は次のように定義した。

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

(「有識者会議」 2016)

ねらいの説明にあたって、「手引」(文部科学省 2020a, p. 11)では、プログラミングの技能を習得すること自体がねらいではないことや、小学校 6 年間を通して、ねらいの①～③のいずれかが「全く欠けていた、ということ」は望ましくないということも示されている(文部科学省 2020a, pp. 11-12)<sup>17</sup>。

次に、「手引」が示すプログラミング教育の評価について確認する。「手引」(文部科学省 2020a, p. 21)では、プログラミングの活動にのみ焦点をあてて評価するのではなく、各教科等の評価規準に基づいて評価することとしていることから、ねらい③「各教科等での学びをより確実なものとする」との関わりが大きいことがわかる<sup>18</sup>。さらに、各学校において、プログラミング教育で育みたい力を明らかにすることなど、プログラミング教育のねらいを達成するための学習活動を計画し実施することと示されている(文部科学省 2020a, p. 21)。

以上のことから、「手引」におけるねらいと評価として、次の 2 点が指摘できる。

① プログラミング教育、とりわけ〈プログラミング的思考〉の評価方法が、明確に定められていないこと

<sup>17</sup> 「手引」の「第二版」(文部科学省 2018d, p. 11)及び「第三版」(文部科学省 2020a, p. 11)では、「プログラミング教育の実施にあたっては、1, 2 をねらいとすること、各教科等の内容を指導する中でプログラミング体験を行う場合には、これに加えて 3 をねらいとすることが必要」と、プログラミング教育が各教科等以外でも実施される可能性が示唆されている。それに対し、「第一版」(文部科学省 2018c, p. 9)においては、「プログラミング教育の実施にあたっては、1, 2, 3 の三つをねらいとすることが望まれ」と、各教科等での実施が前提とされていることがわかる。

<sup>18</sup> プログラミング教育の評価にあたって、「第一版」(文部科学省 2018c, p. 18)においては、各教科等の学習を通じて、〈プログラミング的思考〉等を育むとともに、それぞれの教科等の学習をより深いものとするのが重要と、プログラミング教育のねらい(p. 9)と同様に、各教科等での実践が前提となっている。それに対して、「第二版」(文部科学省 2018d, p. 20)及び「第三版」(文部科学省 2020a, p. 21)では、プログラミング教育のねらいと同様に、「プログラミング教育を各教科等の内容を指導する中で実施する場合には、(略)それぞれの教科等の学習をより深いものとするのが重要」と、各教科等以外でのプログラミング教育の実施の可能性も示している。

- ② プログラミング教育で児童に育む力や、そのねらいを達成するための学習活動の計画、及び実施方法については、「手引」を参照として、「各学校」に委ねられていること

続けて、「手引」の示す小学校段階におけるプログラミング教育の学習活動について確認する。「手引」では先に表7で示したように、学習活動を6つに分類しそれぞれの活動事例について紹介している。なお、「手引」が改訂されるごとに活動事例は追加されている。例えば、「第三版」では「第二版」を踏まえて、A分類の総合的な学習の時間における活動について「プログラミングが社会でどう活用されているか」に焦点を当てて、企業と連携しながらプログラミング教育を実施する事例を3例加えている（文部科学省 2020a, pp. 34-41）。それらの事例数を版ごとにまとめて整理したものを表8に示す。

表8 「手引」における活動の分類と内容の例示数の変遷

	「第一版」 (文部科学省 2018c)	「第二版」 (文部科学省 2018d)	「第三版」 (文部科学省 2020a)
A 分類	3 例 (算数科, 理科, 総合的な学習の時間を 1 例ずつ掲載)	「第一版」に 2 例追加 (総合的な学習の時間の事例を 2 例追加)	「第二版」に 3 例追加 (総合的な学習の時間の事例を 3 例追加)
B 分類	2 例 (音楽科, 総合的な学習の時間を 1 例ずつ掲載)	「第一版」に 2 例追加 (社会科, 家庭科の事例を 1 例ずつ追加)	「第二版」と同じ
C 分類	3 例	「第一版」に 1 例追加	「第二版」と同じ
D 分類	2 例	「第一版」と同じ	「第一版」と同じ
E 分類	例示なし	例示なし	例示なし
F 分類	例示なし	例示なし	例示なし

「手引」では、これら A～F 分類のうち、A・B 分類において、教科名と活動内容が「学習の位置付け」と「学習活動とねらい」、また必要に応じて「その他考えられる学習活動の工夫」を示しながら解説されている。それらを表9に示す<sup>19</sup>。

<sup>19</sup> プログラミング教育の実施例とは別に、〈プログラミング的思考〉を解説する章において、算数科においてコンピュータを用いながら『正多角形をかく』場合の活動例が示されている（文部科学省 2020a, pp. 14-15）。

表9 「手引」のA・B分類における各教科の活動事例

教科名	学年	内容	分類
算数科	第5学年	プログラミングを通して、正多角形の意味を基に正多角形をかく場面 (pp. 25-26)	A
理科	第6学年	身の回りには電気の性質や働きを利用した道具があること等をプログラミングを通して学習する場面 (pp. 26-27)	A
総合的な学習の時間	明示なし	「情報化の進展と生活や社会の変化」を探究課題として学習する場面 (pp. 27-29)	A
総合的な学習の時間	明示なし	「まちの魅力と情報技術」を探究課題として学習する場面 (pp. 29-30)	A
総合的な学習の時間	明示なし	「情報技術を生かした生産や人の手によるものづくり」を探究課題として学習する場面 (pp. 30-32)	A
音楽科	第3学年～第6学年	様々なリズム・パターンを組み合わせて音楽をつくることをプログラミングを通して学習する場面 (pp. 42-43)	B
社会科	第4学年	都道府県の特徴を組み合わせて47都道府県を見付けるプログラムの活用を通して、その名称と位置を学習する場面 (pp. 43-44)	B
家庭科	第6学年	自動炊飯器に組み込まれているプログラムを考える活動を通して、炊飯について学習する場面 (pp. 44-45)	B
総合的な学習の時間	明示なし	課題について探究して分かったことなどを発表（プレゼンテーション）する学習場面 (pp. 45-46)	B

教科とプログラミング教育との関わりについては、A・B分類のほかにも、C分類において、「手引」におけるプログラミング教育のねらいの①と②を達成できる可能性が示されている（文部科学省 2020a, p. 47）。

その例として、「C—③—1 各教科等の学習を基に課題を設定し、プログラミングを通して課題の解決に取り組む学習を展開する例」において、「社会科の我が国の工業生産（第5学年）における優れた製品を生産するための様々な工夫や努力の学習と関連付けて、自動追突防止装置のついた自動車のモデルの製作と追突を回避するためのプログラムの作成を行うこと」といった場面（文部科学省 2020a, p. 49）や、「C—③—2 各教科等の学習を基に、プログラミングを通して表現したいものを表現する学習を展開する例」として「国語科において物語を読む学習をした後、学校の裁量で時間を確保し、物語の中から好きな場面を選び、その場面のアニメーションを作成すること」といった場面（p. 50）が、具体的な活動事例とともに示されている。

これらの活動事例は、いずれも、コンピュータを用いたものであるが、「手引」（文部科学省 2020a, p. 20）では、コンピュータを用いない、いわゆる「アンプラグド」なプログラミング教育について、「第2章（4）プログラミング教育のねらいの実現に向けて」の「ア コンピュータを用いずに行う指導の考え方」において言及している。それによると、既存の実践事例より、低学年児童を対象とした活動で〈プログラミング的思考〉を育む活動の可能性が示唆されること、しかしながら、学習指導要領において「児童がプログラミングを体験すること」（文部科学省 2018a, p. 22）が求められていることを踏まえ、プログラミング教育全体において、児童がコンピュータをほとんど用いないことは望ましくない

と述べている。

ここまで、プログラミング教育が小学校段階において必修化されるまでの経緯を整理した。次節では、必修化されたプログラミング教育の概要を示す。

## 1.5. 小学校段階におけるプログラミング教育の概要

本節では、小学校段階におけるプログラミング教育の概要をとりまとめる。

はじめに、これまでに多くの実践事例や先行研究で踏まえられている「手引」（文部科学省 2020a）を手がかりに、小学校段階のプログラミング教育で児童が身に付けるべき資質・能力（1.5.1.）を整理する。次に、「手引」やこれまでの議論を踏まえながら、プログラミング教育の評価方法を検討し（1.5.2.）、既存の実践事例の分析を通して傾向を示し（1.5.3.）、「ポータル」（文部科学省初等中等教育局情報教育・外国語教育課 n.d.）の事例よりプログラミング教育の環境を示す（1.5.4.）。

### 1.5.1. 小学校段階のプログラミング教育において育むべき資質・能力

プログラミング教育の実施にあたっては、「手引」に示された資質・能力を踏まえて、各学校が、プログラミング教育のねらいを達成するための学習活動を計画、実施することとされている（文部科学省 2020a, p. 21）。よって、本研究を進めるために、「手引」に示された資質・能力を確認する。

「手引」では、はじめに、「小学校プログラミング教育のねらいと位置付けについて」において、プログラミング教育の3つのねらいを『情報活用能力』を構成する資質・能力の①「知識及び技能」、②「思考力、判断力、表現力等」、③「学びに向かう力、人間性等」の観点と照らし合わせながら整理している（文部科学省 2020a, p. 6）（図 3）。



**「情報活用能力」を構成する資質・能力**（「情報活用能力」は、各教科等の学びを支える基盤）

**[知識及び技能]**

情報と情報技術を活用した問題の発見・解決等の方法や、情報化の進展が社会の中で果たす役割や影響、情報に関する法・制度やマナー、個人が果たす役割や責任等について、情報の科学的な理解に裏打ちされた形で理解し、情報と情報技術を適切に活用するために必要な技能を身に付けていること。

**[思考力、判断力、表現力等]**

様々な事象を情報とその結び付きの視点から捉え、複数の情報を結び付けて新たな意味を見出す力や、問題の発見・解決等に向けて情報技術を適切かつ効果的に活用する力を身に付けていること。

**[学びに向かう力、人間性等]**

情報や情報技術を適切かつ効果的に活用して情報社会に主体的に参画し、その発展に寄与しようとする態度等を身に付けていること。

児童に、「コンピュータに意図した処理を行うよう指示することができるということ」を各教科等で体験させながら、

**①②「情報活用能力」に含まれる以下の資質・能力を育成すること**

**[知識及び技能]**

②身近な生活でコンピュータが活用されていることや問題の解決には必要な手順があることに気付くこと。

※プログラミング教育を通じて、児童がおのずとプログラミング言語を覚えたり、プログラミングの技能を習得したりすることは考えられるが、それ自体を、ねらいとはしない。

**[思考力、判断力、表現力等]**

**①「プログラミング的思考」**

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。

**[学びに向かう力、人間性等]**

②コンピュータの働きを、よりよい人生や社会づくりに生かそうとする態度。

各教科等の内容を指導する中で実施する場合には、

**③各教科等での学びをより確実なものとする**

**★ 適切なカリキュラム・マネジメントによるプログラミング教育の実施**

各学校は、プログラミング教育を実施する場面を、教育課程全体を見渡しながらか適切に位置付け、必要に応じて外部の支援も得つつ、実施することが必要。

小学校プログラミング教育のねらい(第2章P11参照)

図3 「小学校プログラミング教育のねらいと位置付けについて」

(文部科学省 2020a, p. 6)

加えて「手引」では、これら3つのねらいを、「議論の取りまとめ」(「有識者会議」2016)と児童の発達段階を踏まえながら、「小学校段階のプログラミング教育で育むべき資質・能力」を3つの観点ごとに示している。それらを次に示す。

### ① 「知識及び技能」に関わる資質・能力

「手引」（文部科学省 2020a, p. 12）では、まず、生活のあらゆる場面で様々なコンピュータがある一方、それらコンピュータの動きが「ブラックボックス化」していることを問題として挙げている。そこで、児童に「コンピュータに意図した処理を行うよう指示することができるということを体験させ」ることを通して「コンピュータはプログラムで動いていること」、「プログラムは人が作成していること」、「コンピュータには得意なこととなかなかできないこととがあること」の3点を気付かせることとしている。

さらに、「今後の生活においてコンピュータ等を活用していく上で必要な基盤」として、「コンピュータが日常生活の様々な場面で使われており、生活を便利にしていること」、「コンピュータに意図した処理を行わせるためには必要な手順があることに気付くこと」の2点を挙げている。

### ② 「思考力、判断力、表現力等」に関わる資質・能力

小学校プログラミング教育において育むべき「思考力、判断力、表現力等」とは、1.3.2.で挙げた「有識者会議」によるねらいのうち「① 論理的思考力や創造性、問題解決能力といった資質・能力を育むこと」、すなわち〈プログラミング的思考〉を身に付けさせることにあたる。

「手引」では、この〈プログラミング的思考〉を「小学校におけるプログラミング教育の中核」（文部科学省 2020a, p. 13）とし、「コンピュータを動作させることに即し」、「コンピュータを動作させるための手順（例）」として示している（図4）。

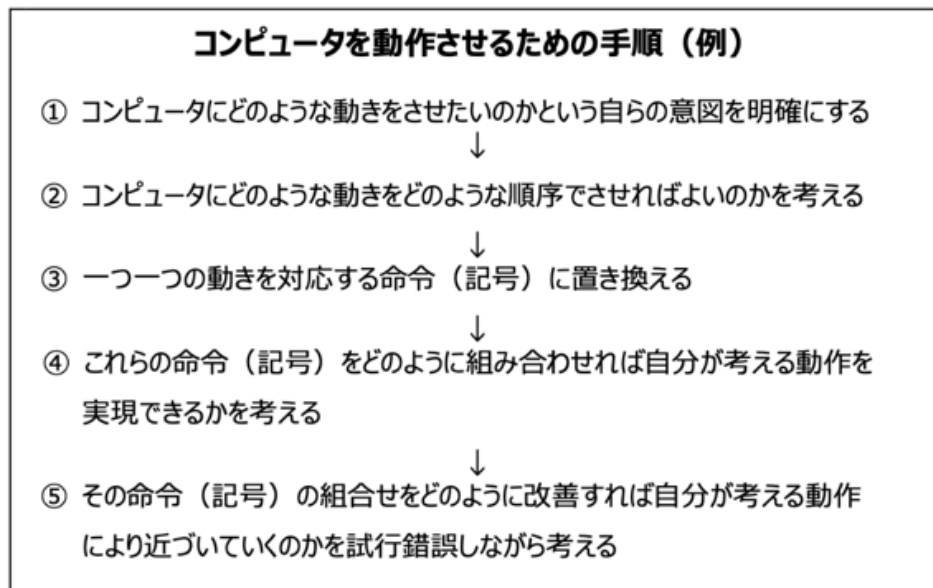


図4 「コンピュータを動作させるための手順（例）」（文部科学省 2020a, p. 14）

「手引」（文部科学省 2020a, p. 14）は、この「コンピュータを動作させるための手順（例）」を踏まえ、『正多角形をかく』場合について考える」活動を示している。具体的には、算数科において「辺の長さが全て等しく、角の大きさが全て等しい」という特質を踏まえて「正三角形」を描く活動の過程である。加えて、「正六角形」や「正八角形」を描くことを考えること、そして、より効率化した過程を、プログラム例（図 5）とともに示している<sup>20</sup>。



図 5 「手引」における「正三角形」を描くプログラム例（文部科学省 2020a, p. 15）

〈プログラミング的思考〉とは、これらの過程を論理的に考える力を指す（文部科学省 2020a, p. 15）。算数科の例を見ると分かる通り、〈プログラミング的思考〉において、児童がめざすべき動きにあたる「意図した一連の活動（学習課題）」は明確である。一方、その具体的な活動過程（動きや、その組合せ）は、試行錯誤を経て明確化される。「手引」の「プログラミング的思考を働かせるイメージ」（p. 16）においても、〈プログラミング的思考〉が「繰り返し学習することで高次に育つ」とあるように、「必要な動きを分けて考える」、「動きに対応した命令（記号）にする」、「組み合わせる」ことを往還する試行錯誤を経て育まれるものとされている。

加えて、「手引」では、留意事項のひとつとして、「意図した一連の活動」を明確化する試行錯誤にあたって、命令は「思い付きや当てずっぽう」で組み立てるのではなく、意図した動きにならなかった場合には、その原因を考えて命令を修正・改善し、再び結果を確かめるといった、論理的に考えられることが求められることが記されている（p. 15）。

<sup>20</sup> 「手引」は図 5 に示された (a) と (b) のプログラム例を比較して、(b) の方が効率的としている（文部科学省 2020a, p. 15）。

### ③ 「学びに向かう力，人間性等」に関わる資質・能力

プログラミング教育における「学びに向かう力，人間性等」に関わる資質・能力とは、「議論の取りまとめ」（「有識者会議」2016）における「発達の段階に即して，コンピュータの働きを，よりよい人生や社会づくりに生かそうとする態度を涵養すること」にあたる。「手引」（文部科学省 2020a, p. 16）では、「児童が，身近な問題の発見・解決に，コンピュータの働きを生かそうとしたり，コンピュータ等を上手に活用してよりよい社会を築いていこうとしたりする，主体的に取り組む態度を涵養すること」といったプログラミングに焦点を当てた態度だけでなく，「他者と協働しながらねばり強くやり抜く態度の育成，著作権等の他者の権利を尊重したり，情報セキュリティの確保に留意したりするといった，情報モラルの育成なども重要」と ICT 端末を用いること自体に対する姿勢・態度を育むことについても示されている。

#### 1.5.2. 小学校段階におけるプログラミング教育の評価

本項では，プログラミング教育における評価について，文部科学省の方針や先行研究を整理する。

1.4. ⑤で整理したように，「手引」（文部科学省 2020a, p. 21）では，プログラミング教育の評価については，各教科等での評価規準に基づいて評価することとされており，児童の〈プログラミング的思考〉の変容をどのように評価するのかについては示されていない。さらに，プログラミング教育で育みたい力や，ねらいを達成するための学習活動を計画，実施することは，「手引」に示されるプログラミング教育で育む資質・能力を踏まえて，各学校の裁量に委ねられている（p. 21）。

#### 1.5.3. 各教科等におけるプログラミング教育の実践事例

プログラミング教育の必修化に先立って，小池（2018）はプログラミング教育の実践研究を学術情報ナビゲータ CiNii を用いて調査した。小池（2018, p. 23）は，まず，「小学」，「プログラミング」をキーワードに論文検索を試みたところ「218 件が一致した」と述べ，続けて，キーワードに各教科等の名称を加えて論文検索を行った結果<sup>21</sup>，得られた実践報告数は，算数科が 8 件，国語科が 5 件，理科が 4 件，図画工作科が 2 件，家庭科が 1 件であり，社会科，生活科，音楽科，体育科は 0 件であったと記している。

筆者は，2022 年 9 月時点において，小池と同様に「小学」，「プログラミング」をキーワードに CiNii の「論文」検索を行った。結果，1,102 件の結果が得られ，そのうち，対象を 2016 年以降に発表された論文に絞ったところ，983 件の結果が得られた。また，対象を小池が調査した 2017 年までに発表された論文とし抽出した結果は 283 件であったことから，プログラミング教育が必修化されて以降，その研究の数が大きく変化したことがわか

<sup>21</sup> 検索語句は，「外国語活動」に「英語」，「特別活動」に「学級活動」を加えるといったように，各教科等に応じて修正されている（小池 2018, p. 31）。

る。そして、それらの研究の傾向は、学習指導要領や「手引」のねらいに沿ったものから、批判的に捉えている事例まで様々である。

以上を踏まえ、本項では、小学校段階におけるプログラミング教育の傾向を示すために、小学校段階での実践や活用を前提とした書籍を抛りどころとして、実践事例を概観し、その傾向を指摘する。表 10 に参照した書籍の一覧を示す。

表 10 プログラミング教育の授業実践事例を扱った書籍

書籍名	著者名
小学校 プログラミング教育がわかる、できる 子どもが夢中になる各教科の実践	教育デザイン研究所 (2017)
教育技術 MOOK 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ア	黒上・堀田 (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	小林・兼宗 (2017)
先生のための小学校プログラミング教育がよくわかる本	利根川・佐藤 (2017)
小学校の「プログラミング授業」実況中継 [教科別] 2020 年から必修のプログラミング教育はこうなる	松田ほか (2017)
プログラミング教育の考え方とすぐに使える教材集	赤堀 (2018)
これならできる 小学校教科でのプログラミング教育	赤堀・久保田 (2018)
小学校の先生のための Why!? プログラミング 授業活用ガイド	阿部ほか (2018)
これで大丈夫! 小学校プログラミングの授業 3+α の授業パターンを意識する [授業実践 39]	小林祐紀ほか (2018)
小学校プログラミング教育の考え方・進め方	蔵満 (2019)
小学校におけるプログラミング教育の理論と実践	東京学芸大学プログラミング教育研究会 (2019)
小・中・高等学校でのプログラミング教育実践 一問題解決を目的とした論理的思考力の育成一	日本産業技術教育学会 (2019)
必須化! 小学校のプログラミング学習一成功する全体計画&授業づくり	磯部 (2020)
授業の腕が上がる新法則シリーズ 「プログラミング」授業の腕が上がる新法則	谷 (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	平井・利根川 (2020)
結局、何を教えればいいかがスッキリわかる! 小学校 はじめてのプログラミング授業	丸岡 (2020)
これで、ICT 活用・プログラミング×『学び合い』は成功する!	水落・齋藤 (2020)
ここがポイント! 小学校プログラミング教育の要点ズバリ! embot で楽しく実践できる授業案特選 15	安藤・額田 (2021)
今こそ知りたい! 学び続ける先生のための基礎と実践から学べる小・中学校プログラミング教育	鳴門教育大学プログラミング教育研究会 (2021)
micro:bit で STEAM 教育 子供の探究する力を伸ばすプログラミング教育	金 (2022)

書籍の例として、各教科等における事例を取り上げたもの（利根川・佐藤 2017，蔵満 2019，谷 2020，平井・利根川 2020），アンプラグドな活動に焦点を当てたもの（小林祐紀ほか 2018），NHK E テレの教育番組「Why!? プログラミング」の内容を活かした阿部ら（2018），特定の学習材を取り上げたもの（安藤・額田 2021，金 2022）が挙げられる。

これら全ての授業実践事例をみたとき、プログラミング教育が様々な教科等において、

全ての学年を対象に実施されていること、それにあたって、活動に相応しい手立てが採られていることがわかる。

#### 1.5.4. プログラミング教育のためのハードウェアやアプリケーション

現在、様々なハードウェアやアプリケーションなどを用いたプログラミング教育の事例がみられる。例えば、「ポータル」（文部科学省初等中等教育局情報教育・外国語教育課 n.d.）では、2022年2月14日時点において、29種類のハードウェアやアプリケーション、書籍などが「実施事例で使用されている教材ツール」<sup>22</sup>として紹介されている。それらを参照したところ、次の4つの傾向が指摘できる。

まず、①ビジュアル言語が多いことである。29個の「教材ツール」のうち、ビジュアル言語が18件（62%）、テキスト言語が13件（45%）、ロボットが6件（21%）あった。次に、②有料の「教材ツール」が多くみられることである。「教材ツール」のうち、15件（52%）が有料、14件（48%）が無料であった。そして、③様々な環境で動く「教材ツール」があることである。「教材ツール」の動作環境を整理したところ、iOSが17件（59%）、Windows、ブラウザが14件（48%）、Androidが13件（45%）であった。また、④低・中・高学年それぞれの段階に対応した教材ツールがみられた。教材ツールの「対象年代」を分析したところ、小学校低学年が20件（69%）、小学校中学年が23件（79%）、小学校高学年が24件（83%）であった。

このように、児童の実態や発達段階に応じて様々な「教材ツール」が様々な用意されている。また、それらの「教材ツール」は多様な環境に向けてつくられていた。

プログラミング教育において、どのようなハードウェアやアプリケーションを用いるかといったことについては、共通の基準が設けられているわけではなく、授業者の判断に任せられている。授業者は児童の実態やICT環境の実態、そして授業のねらいに応じて、ハードウェアやアプリケーションを選定することになる。

### 1.6. 必修化以降の動向 —GIGA スクール構想（2019年12月～）を中心に—

本節では、現在、文部科学省が推し進める GIGA スクール構想の内容を整理する。まず、その内容と変遷を整理（1.6.1.）し、プログラミング教育との関わりを概観する（1.6.2.）。さらに、GIGA スクール構想に指摘される課題についても確認する（1.6.3.）。

#### 1.6.1. GIGA スクール構想の内容と変遷

GIGA スクール構想は、「GIGA スクール実現推進本部の設置について」（文部科学省初等中等教育局初等中等教育企画課 2019）によると、『安心と成長の未来を拓く総合経済対策』（内閣府 2019）を踏まえて次の6点を推進するものとされている。

<sup>22</sup> 文部科学省初等中等教育局情報教育・外国語教育課「教材情報一覧」、インターネット、<https://miraino-manabi.mext.go.jp/teaching>（2022/2/14閲覧）

- (1) 児童生徒 1 人ひとりが端末を持つための予算の適正な執行・管理
- (2) 児童生徒 1 人ひとりが端末を持った際の教科書や教材の在り方検討
- (3) 教師や児童生徒が使いやすい学習コンテンツの調査等や利活用に係る研修
- (4) 地方公共団体が円滑に事業を実施するための国との連携・普及促進
- (5) 新たな教育環境を前提とした教育の企画立案や、情報の収集・利活用の在り方検討
- (6) その他「令和の学校」にふさわしい教育内容を実現するために必要な事項

(文部科学省初等中等教育局初等中等教育企画課 2019)

これらより、GIGA スクール構想とは、児童生徒 1 人ひとりに ICT 端末を整備し、そのために必要な環境を整えることといえる。

1.1. で示したように、児童生徒 1 人がそれぞれの ICT 端末をもつことについては、「日本再興戦略—JAPAN is BACK—」の「IT を活用した 21 世紀型スキルの修得」（首相官邸 2013c, p. 46）において、2010 年代中に整備するという方針がみられる。また、2019 年 12 月 5 日の閣議決定では、学校における高速大容量のネットワーク環境（校内 LAN）の整備を推進すること、それと並行して、2023 年度までに、義務教育段階の児童生徒が 1 人 1 台の ICT 端末をもつことが、『安心と成長の未来を拓く総合経済対策』の「2. Society 5.0 時代を担う人材投資、子育てしやすい生活環境の整備」（内閣府 2019, p. 30）において示されている。

以上、GIGA スクール構想が実施されるまでの変遷を簡潔に整理した。児童への 1 人 1 台の ICT 端末の配備は、経済界からの要請から始まったことがわかる。

## 1.6.2. プログラミング教育と GIGA スクール構想

「手引」において、プログラミング教育を実施するための ICT 環境の整備（例えば、コンピュータの性能や、学内のネットワークについて）のための経費が、GIGA スクール構想の実現に計上されているように（文部科学省 2020a, pp. 58-59）、プログラミング教育と GIGA スクール構想とは関わりがあるといえる。

2020 年には、学習者がどのような ICT 端末を用いるか、どのように学校のネットワークを構築するかについてのモデルを示した「GIGA スクール構想の実現 標準仕様書」（以下、「標準仕様書」）（文部科学省 2020c）が公開された。しかしながら、学習者である児童がどのような ICT 端末を用いるかについては案を示すのみで、その選定は各自治体に委ねられている（p. 1）。

さらに、「標準仕様書」では、ワープロソフトや表計算ソフトといった、ICT 端末に用いる「学習用ツール」とその「想定される利用場面／留意点」が示されており、その中でプログラミング教育についても取り上げられている（文部科学省 2020c, pp. 11-15）。具体的には、「プログラミング教材」として、無償のアプリケーションを使用することや、ICT

端末とプログラミング学習材としてのハードウェアとの互換性について留意することが示されている (pp. 11-12)。

以上を踏まえると、「コンピュータを用いたプログラミング教育の授業モデル」を開発する際には、いずれの環境でも動くような配慮が求められるだろう。

### 1.6.3. GIGA スクール構想の課題

GIGA スクール構想については、各授業における様々な ICT 端末の活用事例がみられる一方で、課題も指摘される。

2021 年の「秋の行政事業レビュー」<sup>23</sup>では、「配付ありきで端末を使う目的が明確ではない」と GIGA スクール構想の目的が曖昧と指摘されている (愛媛新聞 2021-11-09, 朝刊)。ほかにも、GIGA スクール構想で配備された ICT 端末が、プログラミング教育に寄与している一方で、ICT 端末が授業内に有効に位置付けられている割合は少なく、GIGA スクール構想そのものがうまく機能していないこと (日本経済新聞 2022-02-15, 朝刊) や、児童生徒に配付された ICT 端末の故障の多さと修繕やリース更新にかかる費用のこと (読売新聞 2022-10-11) が挙げられている。登本・高橋 (2022) は、2020 年 10 月下旬に全国の初等中等教員を対象に「授業における ICT の活用状況」や「GIGA スクール構想に対する理解」といった調査を行った。その結果、ICT 端末の活用が進んでいない現状や、活用にあたって教員の不安が少なくないことを課題点とし、そのため、ICT 端末を活用するための環境の整備や教員の理解を深める必要があることを指摘している (p. 372)<sup>24</sup>。

以上、GIGA スクール構想の推進によって、2022 年現在、その課題が指摘されながら、児童生徒への端末の整備が進んでいる現状がわかる。

## 1.7. プログラミング教育に関する研究の動向

本節では、学校教育現場で実施されている小学校段階でのプログラミング教育を対象とした先行研究を取り上げ、動向を概観する。

### 1.7.1. 国内の動向の調査

日本国内のプログラミング教育の動向を調査した研究には、①必修化までの動向を整理した研究、②各教科等におけるあり方を検討した研究、③学習材に関わる研究がある。

①の、必修化までの動向を整理したものとしては、阿部 (2021) や磯川ら (2021) が挙げられる。阿部 (2021) は、まず、小学校段階においてプログラミング教育が必修化される前からみられた取り組みを挙げながらプログラミング教育が広まらなかったこと、その

<sup>23</sup> 「行政改革推進会議の下、外部有識者が参加し公開で事業の検証を行うもの」(内閣官房 2021)。

<sup>24</sup> 「手引」(文部科学省 2020a, pp. 52-57)において、企業等と連携したり、プログラミング教育に理解のある ICT 支援員や市民ボランティアを確保したりすることが示されているように、プログラミング教育の実施にあたって、外部と積極的に連携し協力することが大切だといえる。



後、2013年12月のアメリカ合衆国オバマ大統領（当時）のコンピュータ・サイエンスに関わる演説<sup>25</sup>など海外の動きが日本の小学校段階における必修化を後押ししたことを指摘している（pp. 20-21）。続けて、「有識者会議」による「議論の取りまとめ」や2017年告示の学習指導要領、「手引」を取り上げながら、移行期間（2018年～2019年）のアンプラグドな授業例を示し、そこから窺える問題点を指摘している（pp. 21-25）。磯川ら（2021）は、日本国内の6つの学会の論文誌等において2016年6月から2020年3月1日までに報告された研究を収集し、それらを「教育課程内」、「教育課程外」を対象にしたものに分類した上で分析を行い、カリキュラム・マネジメントの方策を検討している。

②の、各教科等におけるあり方を検討した研究としては、「総合的な学習の時間」の実践事例を検討した川崎（2021）やGazzano（2021）が挙げられる。川崎（2021）は、2021年1月に「総合的な学習の時間」におけるプログラミング教育の先行研究を収集し動向を検討している。その結果、収集した研究を「実践研究」、「調査研究（教員）」、「教員養成・教師教育研究」、「実践・研究動向研究」、「システム開発研究」、「意義研究」の6観点に分類し、それら研究の成果や課題を指摘する。Gazzano（2021, p. 11）は、日本の小学校音楽科におけるプログラミング教育の動向を検討し、〈プログラミング的思考〉が創造的な活動を主とする日本の音楽科にそぐわないことを指摘している。

③の、学習材に関わる研究としては、立田（2020）や兼宗（2007）が挙げられる。立田（2020）は、2020年度からのプログラミング教育の必修化を踏まえて、啓林館や東京書籍といった教科書会社によるプログラミング用の教材の特質（pp. 55-56）や、文部科学省の示す指導案例よりプログラミング教育に用いられる環境の動向を検討している（p. 56）。兼宗（2007）は、小学校段階においてプログラミング教育の必修化が明示されるより前に、教育用プログラミング言語を取りまとめ、その動向を示している。

### 1.7.2. 海外の動向の調査

先行研究では、海外でのプログラミング教育の動向を調査したものとして、太田ら（2016）や上松（2016）による各国のプログラミング教育の事例報告や、コールドウェル・スミス（2018）による英国におけるアンプラグドなプログラミング教育の事例の報告がみられる。

とりわけ太田ら（2016）は、1.2. ④で示したように、「調査研究」（文部科学省 2015a）をもとに、各国のプログラミング教育の現状を「諸外国のプログラミング教育を含む情報教育の状況」（太田ほか 2016, p. 198）として整理し、それらを踏まえた上で、初等中等教育でプログラミング教育が必修化されている英国、オーストラリア、米国の一部の州を対象に、プログラミング教育のカリキュラムを調査している。さらに、各国のカリキュラムの内容を検討し、英国とオーストラリアでは〈コンピューターショナル・シンキング

<sup>25</sup> Code.org（2013）「President Obama asks America to learn computer science」インターネット、<https://www.youtube.com/watch?v=6XvmhE1J9PY>（2022年12月24日閲覧）

(Computational Thinking)が情報教育の中核となっていること、また、米国では教科内容と位置付けられていること、さらに、韓国においても選択科目に〈コンピューテーショナル・シンキング〉の考え方が取り入れられていることを示している (p. 201)。

### 1.7.3. 〈プログラミング的思考〉に関わる研究

〈プログラミング的思考〉に関わる研究として、本項では、①〈プログラミング的思考〉の概念に関わる研究、②「有識者会議」(2016)が〈プログラミング的思考〉の定義にあたって参照したと述べる〈コンピューテーショナル・シンキング〉の概念に関わる研究、③〈プログラミング的思考〉と〈コンピューテーショナル・シンキング〉の関わりを論じた研究に分けて整理する。

①の〈プログラミング的思考〉の概念に関わる研究としては、黒上・堀田 (2017) やベネッセ (2018a), 藤原ら (2019), 丸岡 (2020), ロボ団 (2020), 岡崎・大角 (2022) などがみられ、それぞれ、〈プログラミング的思考〉に含まれる要素を指摘している。黒上・堀田 (2017, pp. 7-8) は、〈プログラミング的思考〉の主要素として「①順序 (順次)」、 「②場合分け (分岐)」、 「③繰り返し (反復)」を挙げ、加えて、プログラムを動かす実行の過程と、ある動きが与える影響を考える「相互作用」の過程を提示している。岡崎・大角 (2022, p. 489) は、〈プログラミング的思考〉がもつ要素を示した上で、とりわけ「分解」の過程に着目し、児童がその理解を促すにあたって、「設計図」を考えながらプログラミングに取り組むことが有効であったことを指摘している。

②の、〈コンピューテーショナル・シンキング〉の概念に関わる研究については、定義が一義ではなく、国や企業、団体によって、様々な解釈がされていることが指摘されているように (吉田・阿部 2017, p. 11), 6つの特徴を挙げた Wing (2006, p. 35), 「抽象化」と「自動化」を取り上げた太田ら (2016, p. 202) や磯辺ら (2016, p. 2), 谷 (2018, p. 3) など、様々な議論がみられる。様々な定義がみられることについて、Hu (2011, p. 223) は、〈コンピューテーショナル・シンキング〉についての共通認識がないこと、そして、定義についての研究と議論が続けられていることを指摘している。同様に、林 (2018, p. 170) も、時代とともに、その捉え方が移り変わっていくことを示している。その原因として、阪東ら (2017, p. 177) は、〈コンピューテーショナル・シンキング〉が、プログラミング教育だけでなく、CS, 数学, 科学といった幅広い分野で取り上げられていることを挙げている。〈コンピューテーショナル・シンキング〉の起源についても、Denning (2009, p. 28) が、1950年代から1960年代に用いられた〈アルゴリズムック・シンキング (Algorithmic Thinking)〉という概念が影響したと述べているように、一律ではない。

③の、〈プログラミング的思考〉と〈コンピューテーショナル・シンキング〉の関わりについて、阪東ら (2017) や谷 (2018), 林 (2018) らの議論がみられる。阪東ら (2017, pp. 180-181) は、海外における〈コンピューテーショナル・シンキング〉を検討し、「有識者会議」の示す〈プログラミング的思考〉が「プログラミング (コーディング) に限定さ

れており」,〈コンピューターショナル・シンキング〉の概念を矮小化したものであること,そして,「情報処理の手順を考える力やコーディングの力に留まってしまう危険性がある」ことを指摘し,〈コンピューターショナル・シンキング〉の概念がもつ「現実世界の問題を解決するためには,どのようなシステムが必要であるか,そのシステムをどのようにして実現したらよいかについて考える力の育成は期待できない」と問題点を挙げている。谷(2018, p. 3)は〈プログラミング的思考〉と〈コンピューターショナル・シンキング〉を比較し,〈プログラミング的思考〉は「どちらかと言えば『プログラミングそのもの』というイメージであり,『コンピュータを動かすためのコマンドをどのように組み合わせるのか』を指すことから,〈コンピューターショナル・シンキング〉の一部と見做している。林(2018, p. 170)は,時代とともに新しい解釈がなされる〈コンピューターショナル・シンキング〉の考え方への理解が日本において不足していることを問題とし,〈コンピューターショナル・シンキング〉を踏まえて定義された〈プログラミング的思考〉が,「果たしてどれだけCTを踏まえたといえるのか,疑問の余地は少なくない」と問題提起している。

#### 1.7.4. プログラミング教育のための教材・学習材の開発

1.5.4. において,「ポータル」(文部科学省初等中等教育局情報教育・外国語教育課 n.d.)をもとに,プログラミング教育で用いられる学習材を収集した。それらの中には,予めプログラミング教育のために開発されたものも含まれる。

その例として,原田が2003年に開発したビスケット(Viscuit)が挙げられる。ビスケットは「議論の取りまとめ」や学習指導要領が小学校段階におけるプログラミング教育の必修化を明示するより前に開発されたビジュアル型プログラミング言語である。ビスケットは,多くのワークショップを通して改善された(原田2009, p. 163)。実践例の1つとして,原田ら(2014)は公立小学校の課外活動において2010年3月よりビスケットを用いた実践に取り組み,その結果を報告している。ビスケットを用いた研究として,小学生を対象としたものだけでなく,幼稚園児を対象にした取り組みもみられる(渡辺ほか2021)。原田(2016, p. 348)は,ビスケットを用いた授業やワークショップによる実践を踏まえて,プログラミングを体験することで理解できることを7点挙げ,とりわけ「単純な命令の組合せで,複雑な動きができています」ことを理解することに有効だと述べている。

「ポータル」にはみられないが,多くの実践事例で用いられる学習材として,兼宗ら(2001)が開発した教育用のプログラミング言語ドリトルが挙げられる。ドリトルは,開発されて以後,様々な実践と改善の取り組みがみられる。2004年には,音楽科との親和性を踏まえ,演奏機能が追加された(辰己ほか2005, pp. 77-78)。その後も改良が続けられ,2019年には「ブロックプログラミング環境」を採り入れたドリトルによる実践が,小学生を対象としたプログラミング教室で行われている(山本ほか2019)。

ほか,アンプラグドな視点から学習材の開発に取り組んだ研究もみられる。例えば,寺内(2020)は,音遊びや即興的表現活動に内在するアルゴリズムに着目し,そのようなア

ルゴリズムを児童自ら構築することを主とした〈演奏行為のアルゴリズムを構築する活動〉を提案している。また、そのための学習材の例として《ひみつのサインをつくれ／みやぶれ》を開発している。2022年には、2016年に開発した音楽づくりのための学習材《ステージ》を、プログラミング活動の教材として用いるための捉え直しを試み、授業実践を通して、プログラミング教育の視点より学習材がもつ意義を考察している（寺内2022）。

### 1.7.5. プログラミング教育の指導・学習法の検討

プログラミング教育の授業実践の指導法や学習法に着目した研究や報告として、①プログラミング教育の評価に関わるもの、②カリキュラムに関わるもの、③アンプラグドな指導法についての研究、④プログラミング教育のための教員養成に関わるものがみられる。

#### ① プログラミング教育の評価に関わる研究

1.5.2. で整理したように、プログラミング教育の評価について、「手引」では児童の〈プログラミング的思考〉の変容の評価方法などは示されていない。その一方で、東京学芸大学プログラミング教育研究会（2019, p. 85）は、実際の小学校においては文部科学省が示すように「プログラミング教育を導入して各教科等の資質・能力がどれほど向上したか」を評価することがほとんどとした上で、「各教科等の指導において、プログラミング教育の概念が正しく授業に取り入れられ、実践されたかどうかを評価することが重要」であると主張している。そして、その手立てとして、ベネッセ（2018b）が公表した「第2版『プログラミングで育成する資質・能力の評価規準（試行版）』」を例に挙げている。

評価規準について研究したものとしては、ベネッセ（2018b, 2018c）や小林未歩ら（2018, p. 259）の研究がみられる。それぞれ、「知識・技能」、「思考力・判断力・表現力等」、「学びに向かう力・人間性等」の観点より、プログラミング教育の到達目標を、学年ごとに設定している。

評価基準について述べたものとしては、川島・菊地（2019）が挙げられる。川島・菊地は、小学校段階におけるプログラミング教育では、正確性と効率性を高めた解法の手順を学ぶアルゴリズム学習が求められ、「順次処理、分岐処理、反復処理を組み合わせることで問題を解決するための学習とそれを評価する評価基準も必要」であること（p. 61）、また、「学習には、その成果を評価することが求められる」ため、「アルゴリズム学習」においても、その結果を評価する必要があるとし（p. 64）、「情報科学の観点からの評価項目と評価基準」、「情報技術の観点からの評価項目と評価基準」を提案している（pp. 64-65）。

ほかにも、パフォーマンス評価については、岡崎・大角（2022, p. 490）による〈プログラミング的思考〉の概念がもつ「分解」に焦点を当てて『『分解』の理解をパフォーマンスで測定・評価』した研究が、ルーブリック評価を用いた事例については、金（2022, p. 19）が挙げられる。また、堀田（2017, p. 15）は、〈プログラミング的思考〉の変容につ

いて、数回の授業のみで育まれるものではなく、設けたカリキュラムの中で様々な活動を通して育まれるものであると指摘している。

## ② カリキュラムに関わる研究

カリキュラムに関わる研究として、つくば市総合教育研究所（2021）や鳴門教育大学プログラミング教育研究会（2021）の例がみられる。

つくば市総合教育研究所（2021）は文部科学省の「手引」を踏まえ、「コアカリキュラム」、「オリジナルカリキュラム」を示している。「コアカリキュラム」では、低・中・高学年の全ての段階を対象に、各教科等におけるプログラミング教育の単元や用いられる教材が例示され、つくば市内の全ての学校が、それらのモデルを参考に共通にプログラミング教育を実践することとされている（pp. 10-11）。

鳴門教育大学プログラミング教育研究会（2021, p. 102）は、「1時間の授業の中で達成すべき目標が明確になるように、プログラミング的思考を育成する段階と、活用する段階に分ける必要がある」とし、それらを踏まえた学習モデルを提案している。続けて、鳴門教育大学附属小学校が策定する「ICTを活用した授業実践に関する『コンピュータを活用する力』カリキュラム」をもとに、「小学校プログラミング教育の内容を反映したカリキュラム・マネジメント」を示している（p. 108）。

## ③ アンプラグドな指導法についての研究

プログラミング教育に、コンピュータを用いない、アンプラグドな指導法を研究したものもみられる。

例えば、小林祐紀ら（2018, pp. 180-181）は「カードなどの教具やパズルを使って『コンピュータの仕組み』を伝える」という「コンピュータサイエンスアンプラグド（CS アンプラグド）」を取り上げ、そのうち、「2進法（点を数える）」、「画像表現（色を数で表す）」、「誤り検出と誤り訂正（カード反転の手品）」という3つの題材を例示している。

一方、阿部ら（2018, p. 85）はアンプラグドな学習を行うだけではプログラミングを体験したことにはならないと考え、「必ず、実際のコンピュータ・プログラミングで試してみる『体験』が必要」と主張している。例として算数科を取り上げ、アンプラグドな学習は具体的操作（具体物操作）であり、内容の理解を深めるための「手段」にあたることと、コンピュータを用いた活動は、念頭操作（思考操作）であり、活動の「目的」にあたることを指摘している。加えて、アンプラグドな学習とコンピュータを用いたプログラミングを体験する活動を往還して繰り返すことでより理解が深まるとしている。また、プログラミングでは意図した結果には様々な解法があり、それらの違いを考えることが深い学びにつながることから、「条件を変えて何度でも即座に試せるコンピュータ」に対し、「人手で確認する」アンプラグドな学習では「正誤の検証が難しく、試行錯誤の回数も増やしにくくな」と指摘している（p. 116）。

#### ④ プログラミング教育のための教員養成に関わる研究

教員養成に関わる研究として、2016年に全国の小学校教員を対象に意識調査を行った黒田・森山（2018）や、2018年に全国の小学校教員を対象に質問紙調査を行った佐藤ら（2020）、ほかにも、学生を対象にした原田ら（2018）や山本ら（2020）の研究が挙げられる。とりわけ黒田・森山（2018, p. 172）は、ほとんどの小学校教員が、プログラミング教育に関する知識・理解について不安をもっていることなどを指摘し、「実効性のある具体的な教員研修の構築」を図ることの必要性を主張している。また、佐藤ら（2020, p. 114）は、プログラミング教育のための教員研修に課題があることを指摘している。

教員向けに小学校段階のプログラミング教育を解説した書籍として、小林ら（2019）による、学校や地域、自治体による教員を対象としたプログラミング教育の研修事例を取り上げた『小学校プログラミング教育の研修ガイドブック』が挙げられる。

本章では小学校段階におけるプログラミング教育必修化に焦点を当てその変遷を辿り、プログラミング教育でのねらい等を整理した。さらに、必修化後の動向として、現状までのGIGAスクール構想について整理し、プログラミング教育に関わる先行研究についてとりまとめた。

それらを概観すると、本研究が対象とする小学校音楽科における実践例としては、表現領域の音楽づくり活動を関連させたものが散見される。確かに、音楽をつくる活動は、何らかの意図した結果をめざすプログラミングとの共通点が少なくなさそうである。しかしながら、筆者は、両者がある程度の共通点をもちながらも、その思考過程については同じではないと考えた。例えば、「手引」の示すリズム・パターンを組み合わせる音楽をつくる活動（文部科学省 2020a, pp. 42-43）では、「このような音楽を、このようにしてつくりたいという自分の考え」をもって活動に取り組み、「更に工夫を重ねて試行錯誤」する過程で〈プログラミング的思考〉が働くと記されている。しかし、音楽づくり活動では、試行錯誤を通して「このような音楽を、このようにしてつくりたいという自分の考え」そのものが変わることも少なくなく、またそのことは避けるべきことでもない。

筆者のこの疑問をより明確にするために、次章では、〈プログラミング的思考〉と、音楽づくり活動における思考との比較を通して、両者の違いを明らかにすることを試みる。

## 2. 小学校音楽科において〈プログラミング的思考〉を育む活動に対する問題提起

本章では、小学校音楽科においてプログラミング教育を実施するにあたって、小学校音楽科の表現領域において、〈プログラミング的思考〉を働かせた児童の学習の過程を考察する。

まず、小学校音楽科におけるプログラミング教育の授業実践事例を収集し、それらに共通する傾向を指摘する(2.1., 2.2., 2.3.)。次に、「手引」の「コンピュータを動作させるための手順(例)」(文部科学省 2020a, p. 14)を、フローチャート化を通して整理し、全教科に共通すると考えられる、〈プログラミング的思考〉を育むための児童の学習の過程を検討する(2.4.)。さらに、検討した過程と、実際の小学校音楽科における授業実践事例を踏まえた学習の過程とを比較、検討することによって、小学校音楽科において児童の〈プログラミング的思考〉を育むにあたっての問題提起を行う(2.5., 2.6.)。

### 2.1. 小学校音楽科とプログラミング教育との親和性に関わる先行研究

これまでに、小学校音楽科の表現領域—とりわけ音楽づくり活動と、プログラミング教育の親和性の高さの指摘がみられる。

小学校段階におけるプログラミング教育の必修化以前においては、辰己(2001, p. 42)が「アルゴリズム・プログラミングの内容は、楽典の知識を持っている音楽の教員ならば、簡単に習得できる」と、音楽科とプログラミング教育との親和性を指摘している。それを踏まえて、辰己ら(2005)はドリトルの演奏機能を拡充した。

学習指導要領の改訂にあたって設置された「情報ワーキンググループ」の議事録では、音楽教育において、曲の構成とそのため段取りを組み立てるという点が、プログラミング教育につながるものだと述べられている(文部科学省生涯学習政策局情報教育課 2016b)。

小学校段階におけるプログラミング教育の必修化の方針を位置付けた「議論の取りまとめ」(「有識者会議」2016)では、音楽科におけるプログラミング教育の指導内容のイメージと留意点を示している。それらを整理したものを表 11 に示す。

表 11 音楽科における指導内容と留意点

(「議論の取りまとめ」(「有識者会議」2016)をもとに筆者作成)

指導内容のイメージ
<ul style="list-style-type: none"> <li>○音楽づくりの活動において、創作用の ICT ツールを活用しながら、与えられた条件を基に、音の長さや音の高さの組合せなどを試行錯誤し、つくる過程を楽しみながら見通しを持ってまとまりのある音楽をつくること</li> <li>○音長、音高、強弱、速度などの指示とプログラムの要素の共通性など、音を音楽へと構成することと〈プログラミング的思考〉の関係に気付くようにすること</li> <li>○デジタルによる演奏と生の演奏から感じる違いなどに気付くようにすること</li> </ul>
留意点
<ul style="list-style-type: none"> <li>○低学年における音遊びなどの経験を基盤として、プログラミングと関連付けた音楽活動が、音楽の学びの本質に照らして適切に位置付けられるようにするとともに、子供1人ひとりに創造的な学びが実現し、つくる学習とそれを実際に音や声で表す学習が一層充実するものとなるように十分配慮すること</li> </ul>

また、「有識者会議」(2016)は、「反復記号なども含めた音楽に関わる用語には、順次、分岐、反復といったプログラムの構造を支える要素と共通する性質がある」と小学校音楽科の内容〔共通事項〕<sup>26</sup>とプログラムの構造との関わりを指摘している。

「議論の取りまとめ」の公表以後にみられる、小学校音楽科とプログラミング教育との親和性の高さについての指摘は、〔共通事項〕「音楽を形づくっている要素」に関わるもの、表現領域に関わるもの、順次、分岐、反復といった「プログラムの構造」に関わるものに分けることができる。

小学校音楽科の〔共通事項〕「音楽を形づくっている要素」とプログラミング教育との関わりを指摘したものとして、新山王(2017)や赤堀・久保田(2018)が挙げられる。

新山王(2017, p. 95)は、「思考を伴った試行錯誤を繰り返しながら、これらの要素を用いて音楽づくりをしたり、演奏表現を工夫する際にこれらの要素を視点として演奏の変化や効果について原因と結果を連動して考えたりすること」がプログラミング教育で求められる論理的な考え方につながることを、そして、「音楽という素材を用いながら、自らの問題解決のために何をしていけばよいのかを冷静に考える活動を児童数多く提供し、論理的な視点から、音楽を向き合おうとする経験を深めていくことが、音楽科におけるプログラミング教育の“要”」ということを述べている。

赤堀・久保田(2018, p. 9)は、「音楽を形づくっている要素」が、プログラミングにおける「順次処理」、「繰り返し」、「場合分け」に相当していることから、音楽科の授業には〈プログラミング的思考〉を促す機会があることを指摘している。

表現領域とプログラミング教育との関わりを指摘したものとしては、堀田(2016)や志民(2017)、中村(2019)、ヤマハ(n.d.)が挙げられる。

<sup>26</sup>〔共通事項〕は、「音楽を形づくっている要素」、及び音符、休符、記号や用語を指す。「音楽を形づくっている要素」は、「音色、リズム、速度、旋律、強弱、音の重なり、和音の響き、音階、調、拍、フレーズなど」の「ア 音楽を特徴付けている要素」と、「反復、呼びかけとこたえ、変化、音楽の縦と横との関係など」の「イ 音楽の仕組み」に分けられる。



堀田 (2016, p. 46) は、音楽の曲作りの活動とプログラミングとの親和性の高さを指摘し、「音符は音の強弱や速さの命令である」、「音符を配置すると音が鳴る」、「同じフレーズを繰り返すことがある」といった点をプログラムの発想だと述べている。

志民 (2017, p. 126) は、音楽科の表現領域が「意図した音楽表現を実現するために工夫していく」活動であることから、「音楽表現活動そのものが、プログラミング的思考に基づいた学習活動」だと指摘している。

中村 (2019, p. 118) は、小学校音楽科の音楽づくり活動が「自分がイメージする表現に近付けるために、試行錯誤しながら様々な要素を組み合わせていく活動」であり、その試行錯誤を通して〈プログラミング的思考〉が促されると述べている。

ヤマハ (n.d.) は音楽づくり活動において歌をつくる過程に、〈プログラミング的思考〉を促す過程が含まれていることを指摘している。

音楽科と、順次、分岐、反復といった「プログラムの構造」との関わりを指摘するものとして、阿部ら (2018) や深見・小梨 (2019) が挙げられる。

阿部ら (2018, p. 139) は「楽譜とプログラム」に類似性があり、音楽科におけるプログラミング活動では、「順次、反復などを使った、打ち込みによる自動演奏」が広く行われていることを挙げている。続けて、「決められたプログラム通りに動くコンピュータでは即興演奏はできないことから、人とコンピュータとの違いと、それぞれの良さにまで考察が進むと、より深い学びにつながる」ことを指摘している。

深見・小梨 (2019, pp. 87-89) は、小学校音楽科とプログラミング教育との間に親和性があるといわれる理由として、「音楽の形式や理論に、順次処理、条件分岐、繰り返しといったプログラムの構造を支える要素と共通する性質があるため、特にコンピュータを使用した音楽制作がプログラミング的思考の育成に有効である」ことを挙げている。しかし、音楽を表現する際に働く論理的思考力が、直接〈プログラミング的思考〉につながるわけではなく、「順次処理、条件分岐、繰り返しなどへの意識付けが乏しい実践は、たとえコンピュータを使用していたとしても、それほどプログラミング的思考の育成にはつながらない」と指導にあたっての留意点を挙げている。さらに、小学校音楽科におけるプログラミング教育は、「音楽科の学びを深めること」と「プログラミング的思考を育むこと」といった 2 つのねらいを達成しなければならないことから、教員はそれらをよく理解する必要があるが、「1 人の教師が準備から実践まで行う」ことは困難であると課題を示している。

ここまで挙げてきたように、小学校音楽科におけるプログラミング教育は、表現領域一とりわけ音楽づくり活動における活動を通して児童の〈プログラミング的思考〉を育むことや、〔共通事項〕「音楽を形づくっている要素」にプログラムの構造と共通する要素が含まれていることへの指摘が多く、その親和性がしばしば言及されてきた。

## 2.2. 小学校音楽科におけるプログラミング教育の授業実践事例

1. で取り上げたように、小学校段階におけるプログラミング教育の必修化について、

どのように実施するか、官民通してさまざまな研究と実践が行われている。

プログラミング教育の実施にあたって公開された「手引」では、各教科等におけるプログラミング教育の実施事例が取り上げられており、小学校音楽科におけるプログラミング教育の実施事例として「様々なリズム・パターンを組み合わせて音楽をつくることをプログラミングを通して学習する場面」（文部科学省 2020a, pp. 42-43）が紹介されている。

このような具体的な実践事例<sup>27</sup>は、小学校現場の教員にとって、とりわけ有用な情報のひとつであり、「手引」の事例に加えて、教師向けの書籍や Web サイトにおいても、実際の教育現場で活用可能な実践事例が数多く紹介されている。

そして、プログラミング教育は「各教科等の特質に応じて」実施されるように示されていることから（文部科学省 2018a, p. 22）、プログラミング教育を実施するにあたっては、プログラミング活動の妥当性（＝プログラミング教育のねらいの①と②）と、各教科等の活動としての妥当性（＝プログラミング教育のねらいの③）とを兼ね備えた授業を行うことが求められている。これまでに実施されてきた小学校音楽科におけるプログラミング教育の授業も、当然ながら、音楽科の活動としての妥当性は十分に意識されてきたことだろう。しかしながら、今後の小学校音楽科のプログラミング教育のあり方を考えるためには、実践事例を個別に参照するだけでなく、多くの実践事例の傾向についても検討し、プログラミング教育や小学校音楽科の基本的な考え方に照らし合わせたうえで、授業づくりの原理になり得る要点を考察しなければならない。

2022年3月時点において、さまざまな学会誌や書籍、Web サイトなどを通して、プログラミング教育の実践事例が紹介されている。筆者は、2018年9月に、実践事例の収集を行った（長山 2019, p. 58）。本節では、それらの結果に最新の書籍や論文に示された実践事例を加筆・修正して示す。次に、実践事例収集にあたっての対象範囲と抽出方法、その結果を示す。

### 2.2.1. 実践事例収集の対象の範囲と期間

実践事例収集の抽出の対象として、学会誌などに掲載された論文、書籍、Web サイトに焦点を当てた。なお、抽出にあたっては、小学校での実践のための具体的な学習内容が明記されたもの<sup>28</sup>を取り上げた。

抽出の対象期間は、小学校段階におけるプログラミング教育の必修化を最初に示した公的な報告書である「議論の取りまとめ」（「有識者会議」 2016）が公表された2016年6月16日以降に出版されたものに絞ることとした。「議論の取りまとめ」公表以前も、プログ

---

<sup>27</sup> 本章では、すでに実践された事例だけでなく、モデルとして提示されている授業も含めて扱うこととする。

<sup>28</sup> 近年、学校外の「習い事」としてもプログラミング教育が盛んになってきた（外岡 2018）。こうした動向もまた興味深いものではあるが、本論文では対象外とする。

プログラミング教育の実践事例はいくつか報告されてはいたが<sup>29</sup>、「議論の取りまとめ」公表以降に発表された事例のほうが、小学校段階での必修化を前提とした性格をもったもの多いと考えられるためである。なお、Web サイトからの事例抽出について、2018年に筆者が事例収集した際には、論文と書籍から得られた実践事例数が合わせて3件のみであったため、Webサイトで公開されている実践事例も対象とした。しかし、2022年2月に再度検索を行った際に、「未来の学びコンソーシアム」（文部科学省初等中等教育局情報教育・外国語教育課 n.d.）に掲載された事例や、「ベネッセ 教育情報サイト」（ベネッセ n.d.）の実践事例数に変化がなかったこと、さらに、書籍や論文の実践事例数が多くみられたことから、Web サイトからの事例抽出は2018年に収集した結果をそのまま用いた。

### 2.2.2. 実践事例収集の抽出方法

論文については、抽出にあたって NII 学術情報ナビゲータ CiNii を用いて、「小学校」、「プログラミング」、「音楽」をキーワードに検索を行い、2022年3月時点で、授業実践について扱っている内容を結果に加えた。ただし、「議論の取りまとめ」の公表（2016年6月16日）より前のものは除外した。なお、小池（2018, p. 23）によると、2018年時点では音楽科における実践研究数は0であった。

書籍に関しては、「議論の取りまとめ」以後、2022年3月までに出版されたもの（電子書籍を含む）を可能な限り収集した。ただし、小学校音楽科の事例が掲載されていないものや、小学校での実践が想定されていないもの（例えば、学校外における個人での学習の手引など）は対象外とした。

Web サイトからの事例抽出にあたっては、まず、検索エンジン Google にて、「プログラミング教育」、「事例」をキーワードとし、抽出した結果の上位100件のWeb サイトから、実践事例を公開しているWeb サイトを取り上げた。また、それらのWeb サイト内で紹介されている別のWeb サイト（実践事例集）も対象に加えた。さらに、地方自治体が運営するWeb サイトのなかには、前述の検索方法では抽出されないものがあったため、「プログラミング教育」、「事例」というキーワードに47都道府県名を加えた検索も行い、それぞれの検索結果上位10件からWeb サイトの抽出を行った<sup>30</sup>。

### 2.2.3. 事例の一覧

論文、書籍、Web サイトから収集したところ、論文について17件の実践事例、書籍について12件（うち、2018年時点の結果は3件）の実践事例、Web サイトからの抽出につ

<sup>29</sup> 例えば、『プログラミング教育実践ガイド』（文部科学省 2016c, p. 1）は、「初等中等教育段階におけるプログラミング教育」の「推進」をねらって公開された。しかし、公開された時点では、小学校段階におけるプログラミング教育の必修化や、そのねらいは示されていなかった。

<sup>30</sup> 検索にあたっては、端末に記録されている閲覧履歴等が検索結果を左右しないように、Web ブラウザ Safari のプライベート・ブラウズモードを用いた。

いては 280 件の実践事例を分類し、そのうち 9 件の実践事例を抽出することができた。これら、あわせて 38 件の小学校音楽科における実践事例を「(実施) 学年」, 「領域」, 「(プログラミングをする) アプリケーション」の視点から整理したものを表 12 に示す (なお、内容が明示されていないものは「—」と記す)。

表 12 小学校音楽科におけるプログラミング教育の事例一覧  
(長山 (2019, p. 58) の表 1 をもとに加筆・修正)

事例	事例名	学年	領域	アプリケーション	引用元
1	—	1, 2	A 表現 (2)	ドリトル, Sonic Pi, Scratch	櫻田 (2018)
2	くりかえしをつかってリズムをつくろう	2	A 表現 (3)	Scratch	文部科学省初等中等教育局情報教育・外国語教育課 (2019)
3	ひょうしをかんにて, リズムうちにしたしもう	2	A 表現 (3)	Scratch	ベネッセ (2019)
4	プログラミングを通じた音楽の授業実践例～小 2 音楽「村まつり」	2	A 表現 (3)	Scratch	ベネッセ (2017)
5	2 年生 音楽「[音のスケッチ] おまつりの音楽をつくろう」リズムを選んで合わせよう (B 分類)	2	A 表現 (3)	Scratch	Type_T・堀田 (2021, pp. 74-77)
6	音楽演奏プログラミング	2	A 表現 (2)	Scratch	奥井ほか (2021)
7	拍子づくり	2, 3	—	—	平田ほか (2020)
8	動物が楽しく踊るリズムループをつくろう	3	A 表現 (3)	LOOPIMAL	文部科学省初等中等教育局情報教育・外国語教育課 (2018b)
9	英語×Scratch で楽器を表す単語を楽しく覚えよう	3	A 表現 (3)	Scratch	みんなのコード (2018)
10	プログラミングを通じた音楽の授業実践例～小 3 音楽「まほうの音楽」	3	A 表現 (3)	Scratch	ベネッセ (2018d)
11	いろいろな音のひびきをかんとろう	3	A 表現 (3)	—	大阪市教育センター (n.d.)
12	はくの流れにのってリズム遊びを楽しもう	3	A 表現 (2)	なし	小林・兼宗 (2017, pp. 34-37)
13	スクラッチで「せんりつづくり」をしよう	3	A 表現 (3)	Scratch	つくば市教育局総合教育研究所 (編) (2018, pp. 42-45)
14	日本の音楽に親しもう～ミソラでお囃子の音楽作り～	3	A 表現 (3)	Pyonkee	森脇 (2018)
15	開いてびっくり!音楽のプレゼント～micro:bit で音楽づくり～	3, 4	A 表現 (3)	micro:bit	北川 (2022)
16	日本の音楽に親しもう～ミソラシドでせんりつづくり～	4	A 表現 (3)	Pyonkee	森脇 (2017)
17	日本の音楽に親しもう～ミソラシドで旋律作り～	4	A 表現 (3)	Pyonkee	森脇 (2018)
18	とんびの鳴き声を音で表そう	4	A 表現 (3)	ボーカロイド教育版	つくば市教育局総合教育研究所 (編) (2018,

					p. 73)
19	日本の音楽に親しもう～ミソラドレで旋律作り～	4	A 表現 (3)	Pyonkee, Viscuit	森脇 (2019)
20	音階から音楽をつくろう	4	A 表現 (3)	ボーカロイド 教育版	中村 (2019, pp. 120-123)
21	プログラムを駆使してリズムから旋律へ	4	A 表現 (3)	embot	安藤・額田 (2021, pp. 48-50)
22	リズムアンサンブルづくり	4	A 表現 (3)	Scratch	安田 (2019)
23	5つの音でふしづくり	4	—	—	平田ほか (2020)
24	音の重なりをプログラムで表現する	4, 5	A 表現 (3)	embot	安藤・額田 (2021, pp. 51-53)
25	リズムアンサンブルをつくろう	5	A 表現 (3)	Scratch	CANVAS (2016)
26	リズムアンサンブルをつくろう	5	A 表現 (3)	なし	黒上・堀田 (2017, pp. 46-47)
27	和音づけ	5	—	—	平田ほか (2020)
28	リズムをつくってアンサンブル	6	A 表現 (3)	Scratch	宮城県総合教育センター (n.d.)
29	和音の音で旋律づくり	6	A 表現 (3)	Scratch	福島ほか (2018)
30	和音を生かして旋律をつくろう	6	A 表現 (3)	ボーカロイド 教育版	悴山 (2019)
31	和音の響きの音でふしづくり	6	—	—	平田ほか (2020)
32	—	6	A 表現 (3)	ボーカロイド 教育版	平田ほか (2020)
33	プログラミングでカノンを編曲しよう	6	A 表現 (3)	Scratch	鳴門教育大学プログラミング教育研究会 (編) (2021, pp. 138-141)
34	ひみつのサインをつくれ！／見やぶれ！	5, 6	—	なし	寺内 (2020)
35	ロボットを用いた創作ダンス＋作曲	—	A 表現 (3)	GarageBand ほか	松田ほか (2017, pp. 74-81)
36	音楽で学ぶプログラミング的思考	—	A 表現 (3)	Music Blocks	瀧川 (2022, pp. 80-81)
37	「Scratch」で音楽のプログラミング	—	A 表現 (3)	Scratch	瀧川 (2022, pp. 82-83)
38	《ネクストステージ》	—	A 表現 (3)	なし	寺内 (2022)

### 2.3. 小学校音楽科におけるプログラミング教育の授業実践事例の傾向

本節では、2.2.3. において抽出した小学校音楽科におけるプログラミング教育の実践事例を分析し、小学校音楽科のプログラミング教育にみられる傾向を5つ指摘する。

#### ① 音楽づくり活動が多いこと

内容が明記された事例33件中、30件を音楽づくり活動が占めていた。

#### ② 鑑賞領域の活動がないこと

音楽づくり活動に先立って、音楽の特徴を感じ取ったり親しみをもったりするために、鑑賞活動が位置付けられたものとして事例16 (森脇 2017) があった。しかし、プログラ

ミング教育を鑑賞活動に位置付けた事例はみられなかった。

### ③ 低学年から高学年にわたる幅広い学年が対象となっていること

第1学年～第6学年のいずれかを対象とした事例、また、いずれの学年でも実施可能とした事例がみられた。

### ④ Scratch を用いた活動が多いこと

コンピュータを用いた活動で、用いたソフトウェアが明記された事例29件のうち15件がScratchを用いた活動であった。なお、Scratch1.4をもとに開発されたiOS上で動作するPyonkeeを用いた活動も4件みられた。それらをScratchを用いた活動と見做すと、29件中19件(約65%)がScratchを用いた活動だったといえる。

### ⑤ コンピュータを用いない活動があること

コンピュータを用いない(いわゆる、アンプラグドな学習)活動として、事例12(小林・兼宗 2017, pp. 34-37), 事例26(黒上・堀田 2017, pp. 46-47), 事例34(寺内 2020), 事例38(寺内 2022)の4件がみられた。それらのほとんどは、表現する内容を実現するための手順を考え、児童みずからが表現する活動であった(黒上・堀田 2017, 小林・兼宗 2017, 寺内 2022)が、寺内(2020)の事例は、ある規則(演奏者の合図)に従って表現したり、聴者がその規則が意味することを考えたりするという点で異なっていた。

## 2.4. 「コンピュータを動作させるための手順(例)」に基づいた学習の過程に着目した事例検討

本節では、2.3.において、小学校音楽科におけるプログラミング教育の実践事例に指摘された5点の傾向を踏まえ、小学校音楽科におけるプログラミング教育のあり方を検討する。そのために、小学校音楽科の特質と、「手引」が示す「コンピュータを動作させるための手順(例)」(文部科学省 2020a, p. 14)に着目する。

### 2.4.1. 「コンピュータを動作させるための手順(例)」

「コンピュータを動作させるための手順(例)」(文部科学省 2020a, p. 14)とは、1.5.1.②でも触れた通り、〈プログラミング的思考〉を「コンピュータを動作させることに即して」示したものである。具体的には、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こと、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こと、③一つ一つの動きを対応する命令(記号)に置き換えること、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するか考えること、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こと、の5つの過程から成る。

このことに関連し、「手引」においては、コンピュータを動作させる命令(記号)の組合せを考えるにあたって、プログラミングを支える基本的な要素にあたる「命令(記号)」を順序立てたり、条件を設定して命令(記号)を分岐させたり、命令(記号)を繰り返さ

せたりすることなど」が考えられると示されており（文部科学省 2020a, p. 13），その一例として，『正多角形をかく』場合について考える」過程が紹介されている（pp. 14-15）。

#### 2.4.2. 「すべての教科に共通する」児童の学習の過程

「コンピュータを動作させるための手順（例）」（文部科学省 2020a, p. 14）は，すべての教科に共通する児童の学習の過程として位置付けられている。ここでは，「コンピュータを動作させるための手順（例）」を基に，児童の学習の過程を整理した。その結果を表 13 に示す。

表 13 「コンピュータを動作させるための手順（例）」に即した児童の学習の過程

「コンピュータを動作させるための手順（例）」	児童の学習の過程
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	①「自らの意図」を明確にすること
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	②③④「命令（記号）の組合せ」（プログラム）を構築すること <sup>31</sup>
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	⑤自分が考える動きを実現するための改善方法を「試行錯誤」（実行と検証）しながら考えること

この過程をフローチャートで表したものが図 6 である。児童は，自らの意図を明確にし，命令（記号）の組合せ（＝プログラム）を構築した後，プログラムの実行と，その結果より，プログラムの妥当性についての検証の過程を経て，自らの意図が実現するまで試行錯誤を繰り返す。

<sup>31</sup> 「コンピュータを動作させるための手順（例）」では，プログラムを構築する手順が②～④の 3 段階に分けて示されているが，ここでは，それらをすべて含むものとして『命令(記号)の組合せ』（＝プログラム）を構築する」とした。

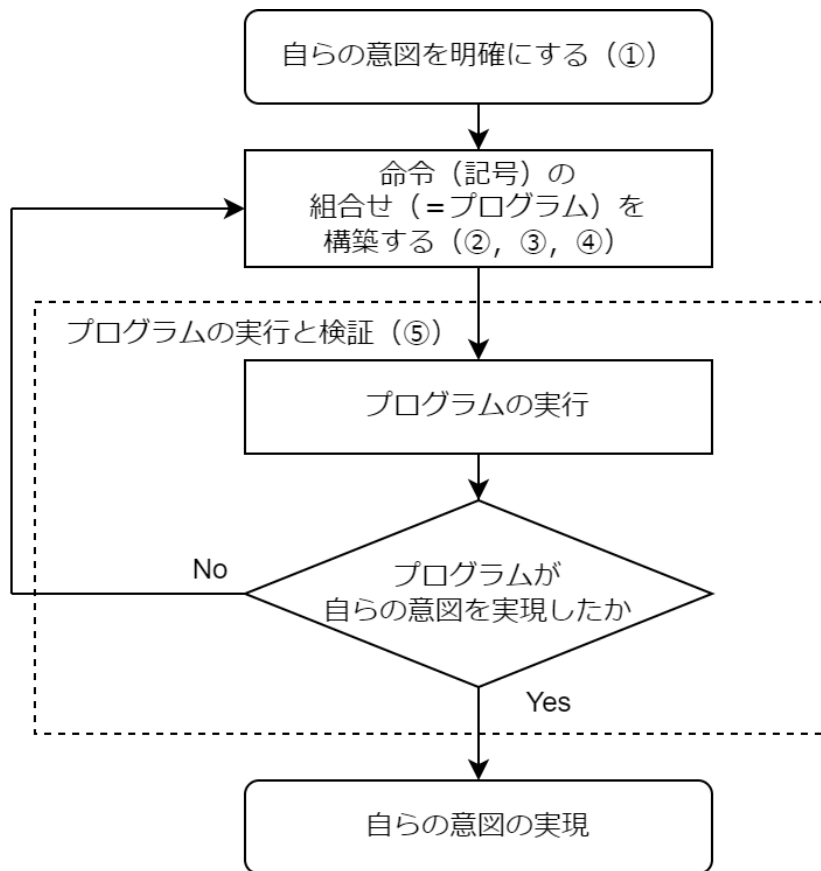


図 6 すべての教科に共通する児童の学習の過程

### 2.4.3. 「音楽科の授業事例を踏まえた」児童の学習の過程

2.4.2. で示した過程を踏まえ、「コンピュータを動作させるための手順（例）」（文部科学省 2020a, p. 14）を基に、音楽科の表現領域にプログラミング教育を取り入れた学習の過程を検討する。その結果を表 14 に示す。



表 14 「コンピュータを動作させるための手順（例）」と  
音楽科の表現領域において音楽表現を成立させるための手順

「コンピュータを動作させるための手順（例）」	音楽科の表現領域において 音楽表現を成立させるための手順
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	①つくりたい音楽，表現したい音楽についての思いや意図を明確にする
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	②どのような表現 <sup>32</sup> をどのような順序で構成す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	③一つ一つの表現を対応する命令（記号やテキスト <sup>33</sup> 等）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	④これらの命令（記号やテキスト等）をどのように組み合わせれば思いや意図に合った表現ができるかを考える
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	⑤その命令（記号やテキスト等）の組合せをどのように改善すれば思いや意図に合った表現により近づいていくのかを試行錯誤しながら考える

2.2.3. で抽出した 30 事例の音楽づくり活動もまた、「音楽科の表現領域において音楽表現を成立させるための手順」①から⑤が含まれている。ここでは、3 事例を取り上げ、その活動過程のなかにこれら①から⑤がどのように含まれているかを示す。

まず、事例 3「ひょうしをかんじて、リズムうちにしたしもう」（ベネッセ 2019）を取り上げる。この事例においては、児童ははじめに「お祭りの音楽」に適したリズムがどのようなものかを、リズムカードを組み合わせながら考え、思いや意図をもつ（①）。次に、コンピュータ（Scratch）でリズムを表現する「命令（記号）の組合せ」（＝プログラム）を考える（②，③，④）。そして、命令の実行によってリズムを演奏した結果をもとに、思いや意図に合ったリズムができたかを検証する（⑤）。思いや意図に合ったリズムが実現できなかった場合は、「命令（記号）の組合せ」（＝プログラム）を考える過程へ戻り、再度検証を行う。この学習の過程をフローチャートで示したものが図 7 となる。

<sup>32</sup> ここでの表現とは、〔共通事項〕「音楽を形づくっている要素」に関連したものを指す。

<sup>33</sup> ここでの「命令」とは、演奏者やコンピュータに対する演奏のための指示である。学習指導要領においては、〔共通事項〕の「音符、休符、記号や用語」として示されているものにあたる。また、コンピュータが演奏を担う場合、用いられる命令は、プログラミング言語やアプリケーションの種類によって異なる。

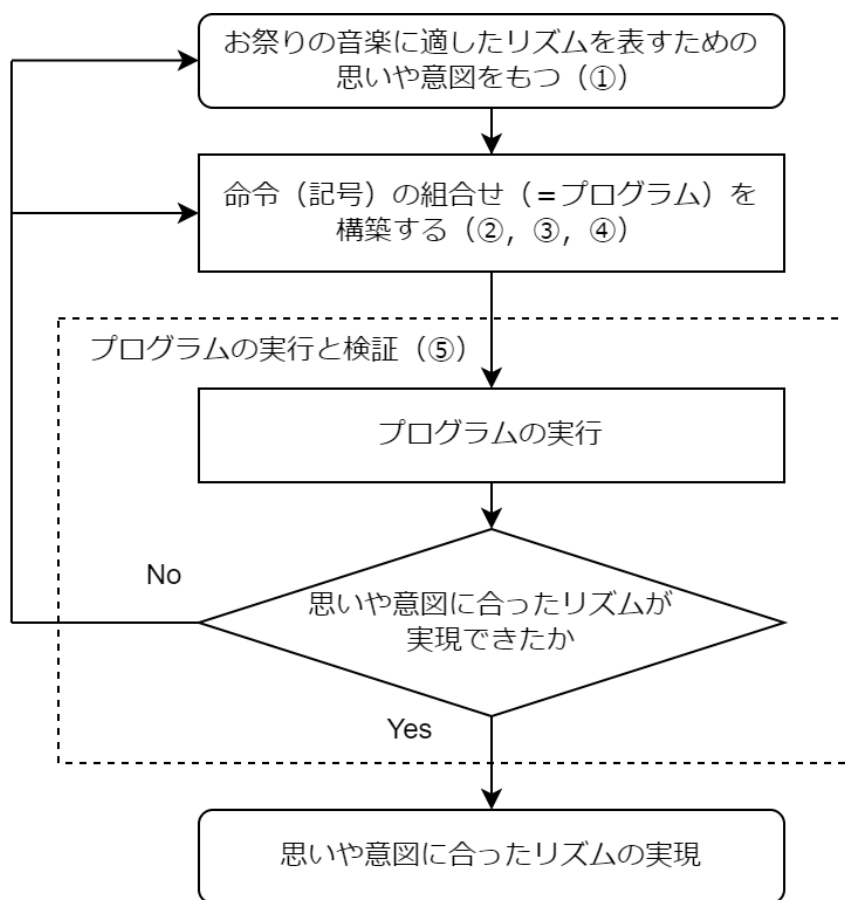


図 7 事例 3 (ベネッセ 2019) における児童の学習の過程

次に、事例 10「プログラミングを通じた音楽の授業実践例～小 3 音楽『まほうの音楽』」(ベネッセ 2018d) を取り上げる。この事例では、まず、自分の思いや意図に合った音色を考える (①)。次に、コンピュータ (Scratch) で思いや意図に合った音色や音価が演奏できるように、「命令 (記号) の組合せ」 (=プログラム) を構築する (②, ③, ④)。そして、命令を実行し演奏した結果より、思いや意図に合った音色が実現できたかを検証する (⑤)。思いや意図に合った音色が実現できなかった場合、「命令 (記号) の組合せ」 (=プログラム) を考える過程に戻り、再度検証を行う。この学習の過程をフローチャートで表したものが図 8 となる。

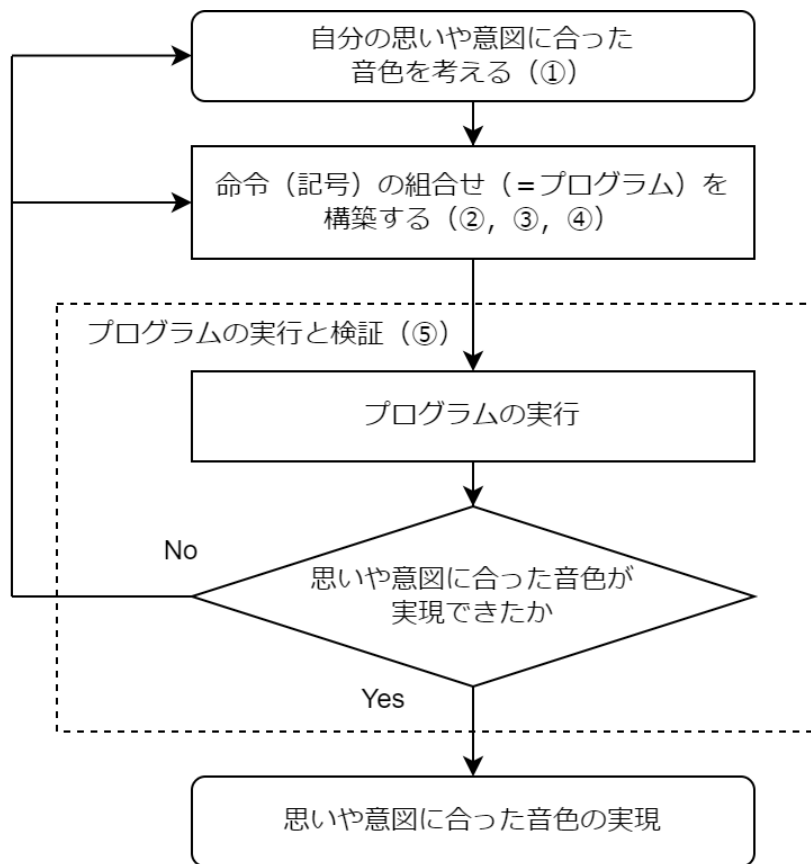


図 8 事例 10 (ベネッセ 2018d) における児童の学習の過程

続いて、器楽活動を取り上げている事例 12「はくの流れにのってリズム遊びを楽しもう」(小林・兼宗 2017, pp. 34-37) を対象に、その学習の過程における「音楽科の表現領域において音楽表現を成立させるための手順」を「コンピュータを動作させるための手順(例)」と照らし合わせながら考える。

まず、「3 拍子のリズムを捉える」(小林・兼宗 2017, p. 35) ために、児童に《陽気なかじや》<sup>34</sup>の範唱に合わせて手遊びをさせ、どのような表現がしたいかという思いや意図をもたせる (①)。次に、それぞれの拍に合った動きや速さを考えさせる (②, ③, ④)。続いて、演奏 (この事例においては手遊び) の結果、動きや速さが思いや意図に合っていたかを検証する (⑤)。思いや意図に合った動きが実現できなかった場合は、拍に合った動きや速さを考える過程へと戻り、実現できるまで繰り返す。このように、事例 12 にも①から⑤の手順が含まれているといえる。これらの過程をフローチャートで表したものが図 9 になる。

<sup>34</sup> 小林・兼宗 (2017) は、《陽気なかじや》の作曲者名を示していない。対象学年が第 3 学年であることから、《陽気なかじや》は『小学音楽 音楽のおくりもの 3』(新実 2016, pp. 38-39) に掲載された《陽気なかじや》(作詞 中地雅之, オーストリア・ドイツ民謡, 佐野真裕編曲) を指していると思われる。

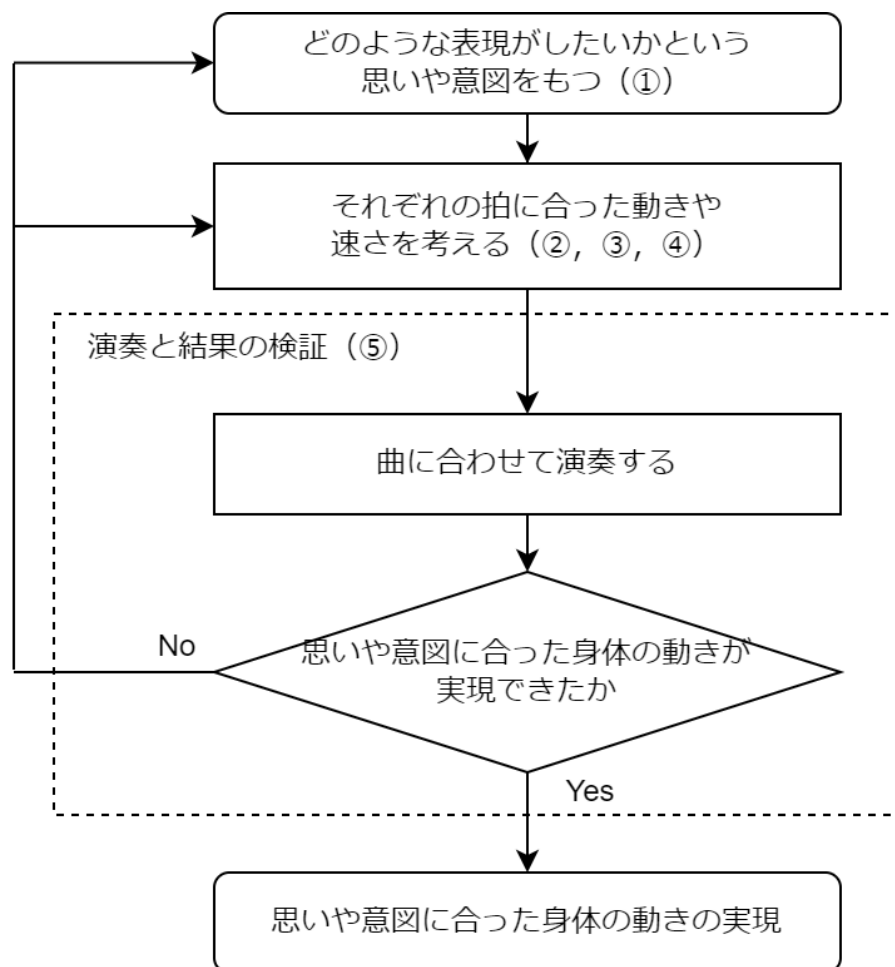


図9 事例12 (小林・兼宗 2017, pp. 34-37) における児童の学習の過程

ここでは、2.2.3. において抽出した実践事例のうち3事例のみを取り上げたが、実際には、すべての実践事例において、「コンピュータを動作させるための手順 (例)」を基にした「音楽科の表現領域において音楽表現を成立させるための手順」の①～⑤の過程が含まれている。

なお、抽出した実践事例には歌唱活動は含まれていなかったが、歌唱活動も器楽活動 (事例12における児童の学習の過程) と同様に考えることができる。

## 2.5. プログラミング教育に反映させるべき音楽科の特質

これまでに、2.2.3. で挙げた各実践事例には「音楽科の表現領域において音楽表現を成立させるための手順」(2.4.3.) の①～⑤の過程が含まれていることを指摘した。

しかしながら、各事例の学習の過程に目を向けると、「音楽科の表現領域において音楽表現を成立させるための手順」と一致しない過程を辿る場合もみられる。具体的には、学習の過程の前提となる「自らの意図」が変化する場合である。

「すべての教科に共通する児童の学習の過程」(2.4.2.) においては、「①『自らの意図』

を明確にすること」における「自らの意図」が児童にとっての課題となり、「②、③、④」の『命令（記号）の組合せ』（プログラム）を構築することと「⑤自分が考える動きを実現するための改善方法を『試行錯誤』（実行と検証）しながら考えること」は、①の「自らの意図」の実現のために行われる。そして、⑤で検証されることは、「②、③、④」で構築した「命令（記号）の組合せ」（プログラム）が、「自らの意図を実現するに適していたかどうか」という点のみであり、はじめに設定した「自らの意図」そのものが変更されることはない。

それに対し、「音楽科の表現領域において音楽表現を成立させるための手順」においては、⑤の検証の結果、当初設定した「思いや意図」を実現するための「命令（記号）の組合せ」（プログラム）を改善すること（②、③、④の再検討）のみならず、「思いや意図」そのもの（①）を更新する場合がある。

このような違いが生じるのは、音楽科の表現領域の活動の多くにおいて、⑤において「表現方法（プログラム）が意図した結果をもたらすものであったかどうか」という検証のみならず、「その結果が音楽的に納得できるものであるかどうか」という検証についても求められるからである。

学習指導要領では、音楽づくりの活動は「創造性を発揮しながら自分にとって価値のある音や音楽をつくるもの」（文部科学省 2018e, p. 23）とされているが、実際には「自分にとって価値のある音や音楽」である完成形がはじめから明確にイメージできている場合は少ない。学習指導要領解説にて「指導に当たっては、音楽をつくっていく過程で、思いや意図を伝え合うことと、実際に音で試すこととを繰り返しながら、表現を工夫し、思いや意図を膨らませるように促すことが大切である。（中略）また、児童がつくった音楽を互いに聴き合いながら、それぞれの表現のよさを認め合い、思いや意図を明確にしながらくっていく経験を積み重ねることも大切なこととなる」（文部科学省 2018e, p. 103）と示されているように、多くの場合において「自分にとって価値のある音や音楽」は音楽をつくりながら考えていくものである<sup>35</sup>。

2.4. では、音楽づくり活動及び器楽活動の学習の過程を「コンピュータを動作させるための手順（例）」に準えて示したが、前述した音楽科の特質を反映させて考えると、「音楽科の表現領域において音楽表現を成立させるための手順」の①の過程では、多くの場合「思いや意図を明確にする」までには至らず、「思いや意図」は漠然としたものに留まるだろう。そして、⑤の過程において、「試行錯誤しながら考える」ことを通して「思いや意図を明確にしていく」のである。

この⑤の過程は、先に示した実践事例にも含まれている。事例3では、「思いや意図に合

---

<sup>35</sup> 音楽科の創作活動における思考の過程を、ラカンの「自己認識のプロセス」の視点から検討した清水（2017, pp.32-33）は、創作活動では試行錯誤の過程で「音」とともに生成・更新されるイメージの連続から音楽が構成される一すなわちつくられる音楽のイメージが児童の頭の中で予め決まっているわけではないことを指摘している。

ったリズムが実現できなかった場合」に、その結果を踏まえて、再び「プログラムを構築する」場合と、あらためて「お祭りの音楽に適したリズムを表すための思いや意図をもつ」場合とに分岐する。また、事例 10 も同様に、「思いや意図に合った音色が実現できなかった場合」に、再び「プログラムを構築する」場合と、あらためて「自分の思いや意図に合った音色を考える」場合とに分岐する。

さらに、これら 2 つの検証は、その優先順位についても留意する必要がある。ときには、プログラム（表したい音や音楽を実現するための命令）が適切だったとしても、実際に音楽を聴いてみたときに「何か違う（私にとって良いと思える音楽はこれではない）」と感ずることもあるだろう。逆に、プログラムが当初意図していた結果をもたらさない不適切なものだったとしても、そこから生まれる音楽に対して「おもしろい。この結果を採用したい」と評価することもあるに違いない。音楽づくり活動が「創造性を発揮しながら自分にとって価値のある音や音楽をつくるもの」（文部科学省 2018e, p. 23）であるならば、「表現方法（プログラム）が意図した結果をもたらすものであったかどうか」よりも、「その結果が音楽的に納得できるものであるかどうか」ということのほうが優先されなければならぬ。このことは、例えば、正多角形をかく活動（文部科学省 2020a, pp. 25-26）のような、プログラムの実行の結果が児童に予め提示されている事例とは異なる「音楽科の特質」のひとつであり、音楽科でプログラミング教育を行ううえで強く意識しておかなければならない重要な点であると考えられる。そして、この思考の過程は、音楽づくり活動のみならず、歌唱活動や器楽活動など、表現領域全般に通じることでもある<sup>36</sup>。

以上の点を踏まえ、小学校音楽科の表現領域におけるプログラミング教育のあり方を考察すると、図 10 のような学習の過程を意識して授業づくりを行うことが重要であると考えられる。基本的な構造は「すべての教科に共通する児童の学習の過程」と同じだが、検証が二段階になっていることが特徴である。

---

<sup>36</sup> ただし、音楽科においても固定された課題を設定する可能性がないわけではない。例えば、「事例 1」にて、櫻田（2018）が、プログラミング言語「ドリトル」を用いて、《ひのまる》を演奏するプログラムを作成する活動を提案しているように、「コンピュータによって、《〇〇》（曲名）のメロディーを忠実に再現する」といった課題であれば、「すべての教科に共通する児童の学習の過程」をそのまま辿ることができるだろう。

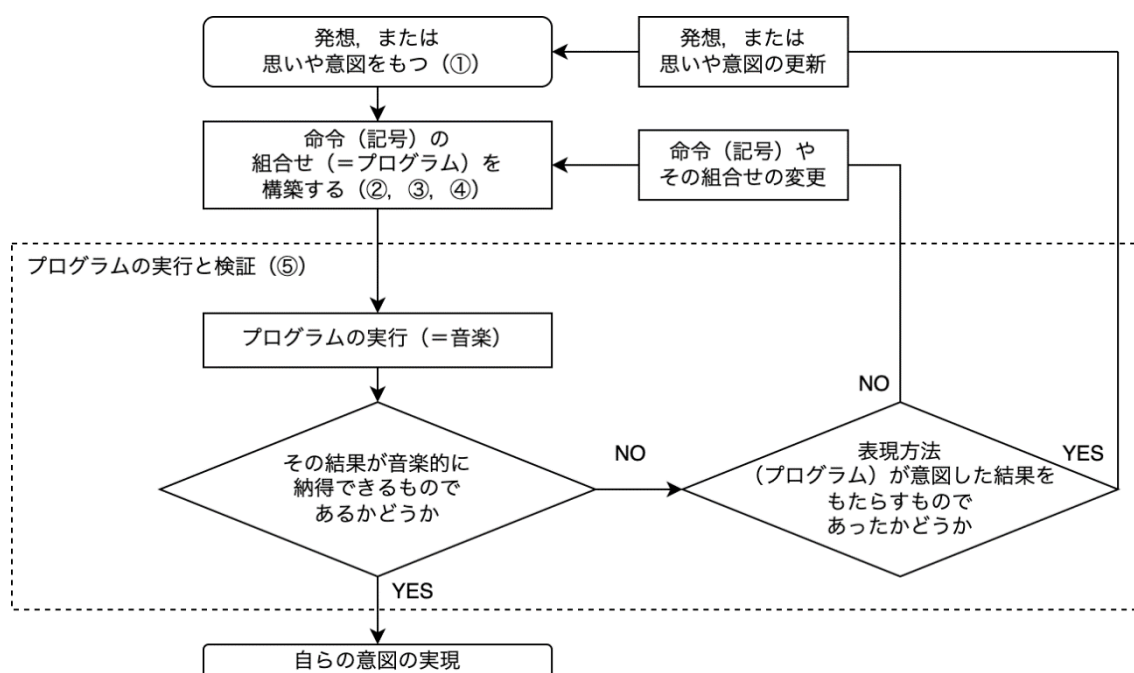


図 10 小学校音楽科の表現領域におけるプログラミング教育を取り入れた学習の過程

## 2.6. 小学校音楽科の特質に応じた〈プログラミング的思考〉と新たな問題提起

本章では、〈プログラミング的思考〉と、音楽科の特質を踏まえた音楽づくり活動の思考過程との違いを検討した。その結果、〈プログラミング的思考〉においては、「自らの意図」が予め明確化されていることが前提であることに対して、音楽づくり活動においては、「自らの意図」にあたる児童の「表現したい音楽」が明確であるとは限らないということが明らかになった。児童の「表現したい音楽」は、図 10 の過程のなかで、次第に明確化していく。ときには途中で大きく転換する場合もあるだろう。

このことから、音楽づくり活動とプログラミング活動を合わせた活動を計画する際には、「小学校音楽科の表現領域におけるプログラミング教育を取り入れた学習の過程」を意識する必要があると考えられる。それは、とりわけ児童 1 人ひとりの「自分にとって価値のある音や音楽」を大切にするためには不可欠な意識だといえる。

本章で示した「小学校音楽科の表現領域におけるプログラミング教育を取り入れた学習の過程」は、小学校音楽科の特質に応じた〈プログラミング的思考〉のモデルのひとつとして位置付けることができよう。しかしながら、この過程にも、ある種の懸念がないわけではない。それは、児童の試行錯誤の速度に関する懸念である。この一連の過程—命令（記号）の組合せを構築し、それを実行（演奏）し、結果となる音楽を聴いて二段階の検証を行い、再び命令（記号）の組合せを変更、あるいは発想または思いや意図を更新する—は、ひとつの楽曲をじっくりと作り上げていくには適しているかもしれないが、即興的俊敏さを発揮して次々と新たな発想を試してみることは不向きだと考えられるからである。場合によっては、実行の結果（音楽）を聴いているうちに、音楽のどの部分の、何に

ついて検証しようとしていたのかがわからなくなってしまうといった状況さえも引き起こすかもしれない。

では、命令と実行の結果（音楽）との因果関係を瞬間的に捉え、その音楽的及びプログラムの妥当性を即時に判断し、次なる命令（記号）をすぐに試してみるといった俊敏さを備えた試行錯誤を促すためには、どのような活動が適しているのだろうか。この問題を考えるために、筆者は、〈プログラミング的思考〉を育むための「コンピュータを動作させるための手順（例）」に基づいた学習の過程ではなく、別のアプローチから小学校音楽科におけるプログラミング教育のあり方を検討することとした。その手法として、次章では〈ティンカリング〉という概念を取り上げ、小学校音楽科のプログラミング教育にもたらず意義を検討する。

\*本章は、筆者が2019年に発表した次の論文を加筆・修正した内容である。  
長山弘（2019）「小学校音楽科におけるプログラミング教育のあり方の検討—授業実践事例を手がかりに—」『初等教育カリキュラム研究』7号，pp. 55-67.



### 3. 試行錯誤による意図の明確化—〈ティンカリング〉—

本章では、2. で指摘した内容をふまえ、試行錯誤を経て「自らの意図」を明確にする〈ティンカリング (Tinkering)〉という概念を取り上げ、小学校音楽科の一とりわけ ICT 端末上のアプリケーションを用いた音楽づくり活動における位置付けを検討する。

はじめに、〈ティンカリング〉の定義と、プログラミング教育との関わりを、先行研究を参照して整理する (3.1.)。次に、アプリケーションの操作の習得にあたって〈ティンカリング〉に期待される働きを示す (3.2.)。そして、〈ティンカリング〉のもつ特質と、アプリケーションを用いた音楽づくり活動に焦点を当てたときの特質を検討する (3.3., 3.4.)。それらを踏まえて、〈ティンカリング〉と〈プログラミング的思考〉との接続について検討し (3.5.)、小学校音楽科の音楽づくり活動における〈ティンカリング〉の位置付けを考察する (3.6.)。

#### 3.1. 〈ティンカリング〉の定義とプログラミング教育に関わる先行研究

本節では、先行研究より、〈ティンカリング〉の定義と、プログラミング教育との関わりを整理する。

##### 3.1.1. 〈ティンカリング〉の定義

〈ティンカリング〉という語の定義を検討したものとして、金井 (2015) やウィルキンソン・ペトリッチ (2015)、柚木・片平 (2015)、レズニックら (2018)、阿部ら (2018)、金 (2022) が挙げられる。そして、その定義として「いじくりまわす (試行錯誤) 中で新しいものを創造する」ことを意味する論が多くみられる。

〈ティンカリング〉を日本語に翻訳するにあたって、金井 (2015, p. 225) は、ちょうど当てはまる語がないこと、そして、「いじくりまわす」がニュアンスとしては近いが、「うまい名詞」がないため、「ティンカリング」とカタカナ表記にしたことを述べている。

語の起源から示したものとして、ウィルキンソン・ペトリッチ (2015, p. 13) は、1300 年代に「家財道具を修理してまわった流しの修理屋」を示す言葉として現れたと説明している。また、金井 (2015, p. 225) は、近年では 2005 年頃からはじまった「ものづくり」を行うことを指す「メイカームーブメント<sup>37</sup>」以降に使われるようになったと述べている。

「ものづくり」に関わった定義として、レズニックら (2018, p. 231) は、〈ティンカリング〉を「ものづくり」の過程において、創造性を引き出すための有効な手立てとして重

<sup>37</sup> アンダーソン (2012, pp. 31-32) によると、「メイカームーブメント」とは、3D プリンタといったデジタル工作機械と、インターネットを組み合わせて、「ものづくり」を行うことである。また、「メイカームーブメント」の契機として、2005 年に創刊された「メイク (Make) 誌」、2006 年に第 1 回が開催された「メイカーフェア」、そして、2007 年に発売された「史上初のオープンソースの卓上 3D プリンタ」の「レップラップ (RepRap)」を挙げている (p. 30)。

視されているものと位置付けている。金（2022, p. 20）は、直訳が「いじくりまわす」ことを指し、ものづくりにおいては「いじくりまわした先にあるメカニズムや機能に思い至ることが含意される」と述べている。

試行錯誤と関わって、阿部ら（2018, p. 89）は、〈ティンカリング〉を「ありあわせの素材を使って試行錯誤をしながら新しいもの（価値）を作り出していくこと」とし、加えて、〈ティンカリング〉は「雑なやつつけ仕事」と軽視されていたものの、1960年代にレヴィ＝ストロース著『野生の思考』によって哲学的に再評価されたことを指摘している。

プログラミングとの関わりをもとに定義したものとして、原田ら（2017, p. 326）は、プログラミングにおける〈ティンカリング〉を「明確な計画を元にプログラミングをするのではなく、素材や自分の書いたプログラムのフィードバックを利用しながら作り上げていくプロセス」と定義している。

柚木・片平（2015, p. 53）は過去の定義を参照し、〈ティンカリング〉の定義は「一義的には定まらない」とした上で、「明確な目標、計画（設計）が存在しない」、「遊戯性、熱意という心情的な側面を持つ」、「組み合わせる、分解するなどをして、様々な目的に合うように作り変えるという技術的な側面を持つ」という、3つの特徴を持つものだとしている。

原田らによる定義や、柚木・片平の示す「明確な目標、計画（設計）が存在しない」という過程は、2. で検討した音楽科の表現領域における特質と一致する点といえる。

本稿では、これらの定義を踏まえて、〈ティンカリング〉を、柚木・片平の示す3つの特徴をもつ概念として論を進める。

### 3.1.2. プログラミング教育における〈ティンカリング〉の意義

これまでに、プログラミング教育と〈ティンカリング〉との関わりを指摘したものとして、プログラミングにおける〈ティンカリング〉の位置付けを示した藤原ら（2019）や金（2022）、プログラミング教育における働きを示した阿部（2021）、Scratch などのプログラミング言語との関わりを指摘した阿部ら（2018）、音楽科との関わりを指摘した志民（2019）が挙げられる。

プログラミング教育との関わりについて、藤原ら（2019, pp. 23-24）は〈プログラミグ的思考〉のもつ論理的思考について整理した上で、プログラミングでは予め全てを綿密に設計してから行うのではなく試行錯誤の繰り返しの過程が含まれていることから創造的思考の側面ももつこと、そして、そのために〈ティンカリング〉が有用であることを指摘し、〈ティンカリング〉によって「児童生徒は、試行錯誤することの方法（探索する、試す、失敗から次を考える、など）や試行錯誤の良さを学ぶことができる」と述べている。また、金（2022, p. 20）は、子供にとって、壊れて動かなくなったものを自由に分解する〈ティンカリング〉は、ものが何からできているのかを実感でき、新しい発想へとつながるという意義があると述べている。その一方で、「授業では時間的・予算的な制約」があ

り児童 1 人ひとりが体験することは難しいことも指摘している。金 (2022, p. 21) は、「プログラミングのいいところは、失敗が失敗とならず気軽に試行錯誤できる点」とし、試行錯誤を繰り返す過程に〈ティンカリング〉を位置付けている (図 11)。

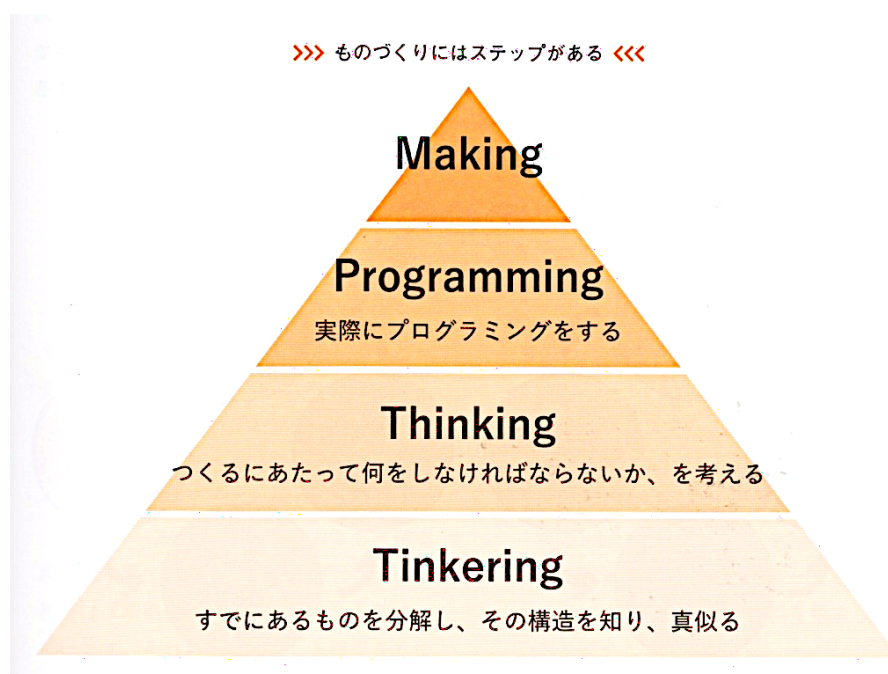


図 11 「ものづくりにはステップがある」(金 2022, p. 21)

プログラミング教育における〈ティンカリング〉の働きについて、阿部 (2021, pp. 24-25) は、必修化に先立って 2014 年から 2016 年にかけて実施したプログラミング教育の事例を踏まえ、プログラミング教育に継続して取り組むことによって児童、さらには授業に対して苦手意識をもっていた教員が変容すること、そのためには「手引」の C 分類にあたる活動が相応しいこと、そして、その活動にあたっては、「教員が授業を統制するのではなく、なるべく児童に自由に弄り回させること (ティンカリング) が求められる」としている。それによって、「児童が、目の前で起こっていることを観察し、仮説を立て、コンピューターや友だちと対話し、試して、失敗し、そのことから学んでいる様子を観察できるだろう」と指摘している。

プログラミング言語との関わりについて、阿部ら (2018, p. 89) は、〈ティンカリング〉の考え方が、Scratch の原型となる教育用プログラミング言語 LOGO を開発したパパートに大きな影響を及ぼしたことを指摘している。そして、Scratch の設計思想には〈ティンカリング〉があり、Scratch でのプログラミングを通して「ティンカリング的思考 (野生の思考)」が育まれることを述べている。

小学校音楽科との関わりとして、志民 (2019, pp. 55-56) は〈ティンカリング〉が音楽づくりや創作の活動、また広く捉えて、音楽を表現すること自体と関わりがあることを指

摘し、とりわけ音楽づくり活動において、学習指導要領の音楽づくり活動の内容にある「即興的に表現すること」と、多くの共通性を見出すことができると述べている。

以上、先行研究における〈ティンカリング〉の語の定義とプログラミング教育における意義について整理した。

次節では、プログラミング活動の過程で求められる知識・技能のうち、とりわけコンピュータの扱いに着目し、〈ティンカリング〉との関係を検討する。

### 3.2. プログラミング活動のためのコンピュータ操作の習得過程

学習指導要領では、プログラミング教育は「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」と示されているように、活動にあたってコンピュータを用いることが求められている（文部科学省 2018a, p. 22）。そこで本節では、実際にどのような環境が用いられているかを確認し、その習得にあたってどのような手立てが考えられるかを検討する。

先に 1.5.4. において、「ポータル」の「実施事例で使用されている教材ツール」（文部科学省初等中等教育局情報教育・外国語教育課 n.d.）を参照し、様々な環境が用いられていることを確認した。2.2.3. で整理した小学校音楽科の事例より、用いられる環境に焦点を当てると、他教科のプログラミング教育でもしばしば用いられる Scratch や、音楽づくり活動に特化したボーカロイド教育版<sup>38</sup>、LOOPIMAL といったアプリケーション、さらに、micro:bit といったハードウェアを用いることによる音楽づくり活動を体験する事例がみられた。このように、目的に応じて様々なハードウェアやアプリケーションを用いられており、さらには、コンピュータを用いない、いわゆるアンプラグドな活動も実施されている<sup>39</sup>。

学習の過程にコンピュータを用いる場合には、児童にそれらを扱うための知識・技能が求められる。例えば、小学校音楽科の場合、旋律をつくる活動のためにボーカロイド教育版を用いるのであれば、コンピュータの画面上で、旋律を書いたり、消したり、再生したりするといった操作の習得が必要となる。また、リズム・パターンを組み合わせるリズムアンサンブルをつくる活動のために Scratch を用いるのであれば、リズム・パターンを作ったり、重ねたり、繰り返したりするといった操作（命令の構築）の習得が必要となる。

こうした操作を児童に理解させるための手立てとして、しばしば採られる方法が、活動に必要な操作をわかりやすく示したマニュアル（説明書）を用意したり、実際に 1 つ 1 つの指示とその結果を、授業者と児童とで確認したりすることである。一方、そういったイ

<sup>38</sup> ボーカロイド教育版は、作成した旋律に歌詞を入力することで、コンピュータに歌わせることができる VOCALOID の機能を学校教育用に最適化したものである（ヤマハ n.d.）。

<sup>39</sup> 例えば、小学校音楽科においては、プログラミングの「反復（ループ）」の考え方を身につけさせるために、拍に合わせて自分たちで考えた身体の動きを繰り返すといった活動がみられた（小林・兼宗 2017, pp. 34-37）。

ンストラクション（教示）に頼らず、児童が自分で勝手に様々な操作を試す、すなわち〈ティンカリング〉を通して、操作を習得していく場合もある。

これら 2 つの方法は、一概にどちらが優れているとは言い難いものであるが、本研究では、後者の方法に焦点を当てる。なぜなら、「いじくりまわす」という〈ティンカリング〉は、単にハードウェアやアプリケーションの操作を習得するだけでなく、創作活動のための発想を生み出す源にもなり得ることが期待されるからである。次に、その根拠として、〈ティンカリング〉の特質を検討する。

### 3.3. 〈ティンカリング〉の特質

〈ティンカリング〉の特質を検討するために、まず、メイカームーブメントにおける〈ティンカリング〉の理念を参照する。次に、〈状況的行為〉と〈プランニング・モデル〉、さらに、〈道具主義的方法〉に着目して議論を進める。

#### 3.3.1. ウィルキンソン・ペトリッチが示した理念

〈ティンカリング〉の特質を考えるうえで、重要な示唆を与えてくれるのが、次に示すウィルキンソン・ペトリッチ（2015, pp. 14-15）による「ティンカリングの理念」である。

- ・ アイデアを何度も見直して作り直そう
- ・ よく知っている素材を知らないことに使う
- ・ 古い技術を見直す（そして新しい技術を発見する）
- ・ すぐに作る
- ・ 科学とアートと技術を融合させる
- ・ 作ることでアイデアを表現しよう
- ・ ユーモアを持とう
- ・ 真面目に取り組む しかし 真面目にならない
- ・ ゴチャゴチャでやかましくて、ときには危ない状況に自分を置く
- ・ 現実世界のお手本はいたるところにある
- ・ 自主性と協力の均衡
- ・ やって困ろう
- ・ 知らなくて当たり前
- ・ 道具を愛せよ
- ・ 消費するより創り出せ

（ウィルキンソン・ペトリッチ 2015, pp. 14-15）

これらの理念からは、なにかを行うという試行の過程が〈ティンカリング〉の中心に位置付けられているとわかる。とくに、「すぐに作る」、「やって困ろう」、「アイデアを何度も

見直して作り直そう」という言葉からは、前述の定義にもあるように、明確な目標や計画をはじめから立てておくのではなく、試行を繰り返す過程で、その目標や計画を明確にしていくという特徴を窺うことができる。実際、レズニックら (2018, pp. 232-233) は、ティンカラー (〈ティンカリング〉する人) が「常に自分たちの目標 (どこに向かうのか) と計画 (どのようにそこに向かうのか) を再評価」し続けることを示している。

ただし、〈ティンカリング〉でははじめの目標が明確ではないものの、「〇〇を使って『何か』を作る」といった広い意味での目的はある程度設定されている。本稿では、このことを「目的」、また、具体的で明確なゴールを「目標」として定義する<sup>40</sup>。

このように、事前に明確な目標や計画をもたず、試行錯誤を通して目標を明確にしていくという〈ティンカリング〉では、そこにある道具や素材が持つ可能性が大きな意味を持つ。何を試すことができるか、どこまで可能かは、そうした〈状況〉や〈環境〉に依存しているからである。つまり〈ティンカリング〉は〈状況的行為 (situated action)〉の1つということができるだろう。そこで、〈状況的行為〉をしばしば対になって論じられることの多い〈プランニング・モデル (planning model)〉と比較しながら、その特質を検討する。

### 3.3.2. 〈状況的行為〉と〈プランニング・モデル〉

〈状況的行為〉とは、「状況に埋め込まれた行為」とも呼ばれる (サッチマン 1999, p. 27)。サッチマン (1999, p. 49) は、〈状況的行為〉では、プランを実現するために行為があるのではなく、プランは〈周辺環境 (circumstances)〉によって構築されることを述べている。また、岡田 (2010, p. 525) は、〈状況的行為〉においては、目的が次の行為を導き出すのではなく、〈状況 (situation)〉<sup>41</sup> が次に採るべき行動を導き、結果として目的が達成されることを述べている。

一方、意図した結果に向けて計画を立て、それに沿って行為することを〈プランニング・モデル〉という (サッチマン 1999, p. 28)。〈プラン (plan)〉とは「行為や行動に先だって用意された計画のこと」である (岡田 2010, p. 526)。〈プランニング・モデル〉の利点とは、目的を達成するまでの効率の良さである。目的のための計画は、事前に綿密に構築されるため、試行錯誤の回数は減り、最適化された過程を経ることができる。その反面、柔軟性は低い。〈プランニング・モデル〉においては、〈状況〉が変化した場合、行為者は再度、計画を作り直す過程に立ち戻る必要がある (サッチマン 1999, p. 29)。

<sup>40</sup> 『大辞泉』(松村 1995, p. 2621)によると、「目的」とは「実現しようとしてめざす事柄。行動のねらい。めあて」である。そして、『目的』は、『目標』に比べ抽象的で長期にわたるめあてであり、内容に重点を置いて使う」とされている。また、「目標」とは、「行動を進めるにあたって、実現・達成をめざす水準」であり、「めざす地点・数値・数量などに重点があり (中略) より具体的」とされる。

<sup>41</sup> 岡田 (2010, p. 526) は、〈状況〉を、次のように説明している。「行為主体を取り囲み、その行為を支えるもの。周囲に存在する物理的な環境、発話や行為の結果など行為者によって生み出されたもの、行為主体を支える人的な環境などを含む」。

〈状況的行為〉は〈プランニング・モデル〉に比べ、効率性で劣るものの、柔軟性の面では優っている。行為者は、1つ1つの〈状況〉の変化に適切に対応することで、目的の達成に近づいていくのである。

しかしながら、ここで1点留意したい。それは、サッチマンや岡田の言説によると、〈状況的行為〉においては、達成されるべき目標は予め明確であり、最終的にその目標が達成されることである。一方、〈ティンカリング〉の場合、ティンカーが試行錯誤の過程で目標と計画を再評価し続けること、また、ウィルキンソン・ペトリッチ (2015, p. 13) が「きちっとした結果につながる手順を着実に踏むわけではなく、「自分自身でも思いもよらなかった素晴らしいものが生まれてくる」可能性がある」と述べていることから、〈ティンカリング〉では予め大まかな目的 (例えば、「ものを作る」、「旋律をつくる」といったこと) は設定されているものの、何をつくるのかといった具体的な目標は、試行錯誤を経て明確になるため、最終的にできあがる形としては様々な可能性が考えられる。この点において、〈ティンカリング〉は、目的は決まっているが、目標が活動の過程で変化するものづくりの活動に適しているといえるだろう。〈ティンカリング〉を扱った書籍には、「安いおもちゃや使われなくなった古い機械をバラバラにして、なにか新しいモノに作りかえる」活動 (バンジ・シロー 2015, p. 7) や、針金を使って何かを作る活動 (ウィルキンソン・ペトリッチ 2015, pp. 108-115) などが例示されているが、音楽づくり活動もまた、このような「目的は決まっているが、目標が活動の過程で変化する活動」として設定することが可能である。

この、目標が活動の過程で変化する特質は、〈状況的行為〉の強みである柔軟性を一層高めることになると考えられる。レズニックら (2018, p. 232) は、〈ティンカリング〉について、「効率を失う代わりに、創造性と俊敏さを得る」と述べているが、こうした創造性、俊敏さを引き出すための〈状況〉として挙げられる重要な点は、「達成されるべき目標の変更が許される」という前提そのものなのである。

### 3.3.3. 〈道具主義的方法〉

〈ティンカリング〉においては、前述の〈状況〉を構成する重要な要素のひとつに、道具や材料がある。

道具や材料は、ティンカーたちの行為にどのような影響を与えるだろうか。このことを考えるために、ここでヴィゴツキーの〈道具主義的方法 (instrumental method)〉といった概念を参照したい。ヴィゴツキー (1987, pp. 51-59) によれば、人の行動は形ある道具だけでなく、形のない心理的道具<sup>42</sup>によっても影響を受ける。動物の行動は〈刺激〉と〈反応〉によって説明されるが、人間の場合にはそれらをつなぐ〈媒介〉がある。このよ

---

<sup>42</sup> 心理的道具の例として、ヴィゴツキー (1987, p. 52) は「言語、記数法や計算のさまざまな形式、記憶術のための諸工夫、代数記号、芸術作品、文字、図式、図表、地図、設計図、そしてあらゆる種類の記号など」を挙げている。

うに、人はある道具や心理的道具によってできることしか、しようとしなない。つまり、はじめに、行為者が用いることのできる道具や材料があり、そこから〈欲求〉が生まれるのである<sup>43</sup>。

つまり、ヴィゴツキーは、〈欲求〉を、道具や材料があることによって引きだされるものとして位置付けているのである。このことは、道具や材料が、新たな〈発想〉を引き出すトリガーになるということでもある。用いることができる道具や材料、それらを用いて「何ができるか／できないか（可能性）」、「どこまでならできるか（限界）」を理解することは、発想に大きく影響するのである。

以上、本節で検討したことを踏まえると、〈ティンカリング〉の特質として、次の3点を挙げることができる。

- ① 〈ティンカリング〉は、〈状況的行為〉のひとつであること
- ② 「達成されるべき目標の変更が許される」という前提が〈状況〉を構成する重要な要素であり、このことによってティンカラーの創造性と俊敏さが引き出されること
- ③ 道具や材料が、行為の（可能性と限界の）理解を促し、〈発想〉を引き出すトリガーとなること

これらの特質を踏まえ、筆者は、〈ティンカリング〉を、試行錯誤が求められるプログラミング活動の過程に取り入れることによって、児童の創造性を育むために効果的な働きとなること、そして、小学校音楽科の表現活動、とりわけ即興的な音楽づくり活動において有用なプロセスになり得ると考えた。

### 3.4. アプリケーションを用いた音楽づくり活動における〈ティンカリング〉の特質

ここまで〈ティンカリング〉の特質を、3.3.1. で示したメイカームーブメントにおける理念を出発点として検討してきた。音楽づくりもまた、何かをつくるという点においてはものづくりと共通する行為であるが、音楽づくり活動における〈ティンカリング〉は、ものづくりにおける〈ティンカリング〉とは異なる点もある。本節では、アプリケーションを用いた活動を取り上げ、音楽づくり活動における〈ティンカリング〉の特質を検討する。

#### 3.4.1. 道具と材料

本項では、音楽づくり活動全般における道具と材料を検討する。

道具については、本節で焦点を当てている「アプリケーション」のほかにも、実際の「楽器」や「音具」、「メモ帳」、「楽譜」など、様々な可能性が考えられる。

<sup>43</sup> その例として、有元（2011, p. 33）は、「『暑いから エアコンをつけよう！』という欲求はわき起こるが、『体温の下がる薬を飲もう』とは欲求しない」こと、「『電車で横浜に行こう』と思うが、『瞬間的に移動したい』とは、夢はしても、欲求はしない」ことを挙げている。



一方、音楽をつくるうえで不可欠な材料のひとつとして、「音」が挙げられる。音は、並べられたり、重ね合わせられたりすることによって、音楽へと構成されていく。この時、音の並べ方や重ね方の様子を認識するための重要な要素や仕組みとなるのが、学習指導要領に〔共通事項〕として位置付けられている「音楽を形づくっている要素」<sup>44</sup>である。

### 3.4.2. 目に見えない材料を用いた創作過程の特質

音楽づくり活動における材料が「音」であるということは、音楽づくりと、ものづくりとの大きな違いを生じさせている。

一般的に、ものづくりでは、目に見える（形のある）材料を使って、目に見える（形のある）何かをつくり出す。一方、音楽づくりは、「音」という目に見えない（形のない）ものを材料として、目に見えない（形のない）音楽をつくる行為であり、その過程で重要となる「音楽を形づくっている要素」もまた、目には見えない。

したがって、音楽づくりの過程においては、創作の経過を確認するために、耳で聴くことが必要になる。だが、音は、形あるものと違い、時間の経過とともに消えてしまうため、それを実際の耳で確認するためには、その都度、実際に鳴らしてみることが必要となる。

### 3.4.3. 〈ティンカリング〉の道具としてのアプリケーションの意義

音楽づくりにおいて、「音」を〈ティンカリング〉する際「音楽を形づくっている要素」をどのように変化させることができるかは、用いる道具の性格に大きく依存している。ここでは、本節で検討対象としているアプリケーションを道具とした場合に焦点を当てて、その意義を検討したい。

#### ①多様な機能を備えていること

まず、アプリケーションの特質として、機能の多様さが挙げられる。

音楽創作アプリケーションは、創作活動のために様々な音素材を備えていることはもちろん、生成される音に対して、リアルタイムに「リバーブ (Reverb)」（残響）や「ディレイ (delay)」（遅延）といった効果を与えるなど、できることも多岐にわたる。

そのため、アプリケーションは、それ自体がひとつの道具というよりも、様々な道具が入った、いわば道具箱だといえるだろう。このことは、児童の創作の発想を多様に引き出す〈状況〉のひとつになり得ると考えられる。

#### ②命令を正確に実行し、再現できること

次に、音楽を正確に演奏することができる特質が挙げられる。

---

<sup>44</sup> 「音楽を形づくっている要素」とは、「音色、リズム、速度、旋律、強弱、音の重なり、和音の響き、音階、調、拍、フレーズなど」の「ア 音楽を特徴付けている要素」と、「反復、呼びかけとこたえ、変化、音楽の縦と横との関係など」の「イ 音楽の仕組み」から成る（文部科学省 2018a, p. 127）。

実際の楽器を道具とした音楽づくり活動においては、児童の演奏技能が、発想の可能性を制限してしまう〈状況〉になりかねない。一方、アプリケーションは、意図した音楽が、高い演奏技能が求められるものであっても、正確に演奏することができる。このこともまた、児童の創作の発想を豊かに引き出す〈状況〉のひとつになり得ると考えられる。

### ③「音楽を特徴付けている要素」がもたらす効果を個別に確認できること

続いて、「音楽を形づくっている要素」の「ア 音楽を特徴付けている要素」の1つ1つに焦点を当てて操作ができることが挙げられる。実際の楽器を道具とした場合には、ある要素を変えた場合に、他の要素も影響を受けることがある。例えば、「音色」を変えることを意図して楽器を別の楽器に変えた場合に、変えることを意図していなかった「音量」まで変わってしまうことがある。これは、実際の楽器では「音楽を特徴付けている要素」に挙げられている諸要素が連動していることが多いからであるが、他方、多くのアプリケーションでは、「音量」のみ、「速度」のみ、というように、1つ1つの「音楽を特徴付けている要素」を個別に、独立して〈ティンカリング〉することができる<sup>45</sup>。このことは、「音楽」を、「音楽を特徴付けている要素」に分解して捉える見方を促すための〈状況〉になると考えられる<sup>46</sup>。

### ④試行錯誤を促すことで、発想、または「思いや意図」に影響を与えること

「コンピュータを動作させるための手順（例）」（文部科学省 2020a, p. 14）を踏まえると、コンピュータは児童が構築した命令を実行する過程を担う。そして、小学校音楽科においては、プログラムの実行とは、発想、または「思いや意図」を基に、音楽を「演奏」（生成）する過程にあたる。堀田（2016, p. 46）が「楽器演奏の技能が十分でない子ども、コンピュータを持ち込むことで、作曲できるようになる」と示しているように、アプリケーションを用いてコンピュータに演奏の過程を担わせることで、児童は演奏のための練習の過程を省くことができる。例えば、2.2.3. に示した事例2では、コンピュータに演奏の過程を担わせることで、「聴きながらリズムを確認したり、より面白い音楽になるように何度もやり直したりすることが容易となる。さらには、次に児童が自分でリズムを演奏する際の、範奏として活用することができる」（文部科学省初等中等教育局情報教育・外国語教育課 2019）という利点が挙げられている。また、事例8では、タブレット端末でリズムの配置を試しながら「リズムループづくりの発想を得て、思いや意図を明確にしていく」（文部科学省初等中等教育局情報教育・外国語教育課 2018b）と述べていることから、コ

<sup>45</sup> 「音楽を特徴付けている要素」の中に「音量」は含まれていないが、ここでは、「音楽を特徴付けている要素」のひとつ「強弱」に影響を与える要素として取り上げた。また、多くの音楽創作アプリケーションでは、「強弱」を変化させるために、「音量」や「ボリューム」、「ベロシティ」と呼ばれる値を操作する。

<sup>46</sup> ここでは「音楽」を「音楽を特徴付けている要素」に分解して捉える見方を促すことに焦点を当てているが、もちろん、実際の楽器を用いた活動にも、複数の要素が連動しているからこそ得られる重要な学びが含まれていることは強調しておきたい。

コンピュータに演奏の過程を担わせることには即時性があること、そして、間違いのない再現性が「思いや意図」にも影響を与えることが挙げられる。

このように、コンピュータに「演奏」の過程を担わせることによって、「プログラムの実行（演奏）のための練習」の過程を省くことができ、試行錯誤を促すことが期待される。

#### ⑤命令と実行の結果の因果関係を考えること

さらに、このような俊敏さを伴う試行錯誤は、児童がコンピュータへの命令とその「実行の結果」を理解することにも寄与すると考えられる。それは、アプリケーションの操作法に関してだけでなく、音楽的な理解にも及ぶ。3.4.2. において、音は、形あるものと違い、時間の経過とともに消えてしまうため、それを実際の耳で確認するためには、その都度実際に鳴らしてみることが必要と述べたが、俊敏さを伴う試行錯誤の経験は、「ここをこうすればおそらくこうなるだろう」のように音楽的結果を想像し、実際に鳴らしてみなくても推測できる力を高めることが期待されるのである。この力は、音楽をつくるうえで大切な力のひとつである。

以上のことから、アプリケーションによって〈ティンカリング〉することの意義として、次の3点が挙げられる。

- ①多様な機能を備えていること、また、高い演奏技能が求められる音楽であっても正確に演奏できるといった特質が、児童から多様な発想を引き出すための〈状況〉になり得ることが期待できること
- ②1つ1つの「音楽を特徴付けている要素」を個別に、独立して〈ティンカリング〉することができるため、「音楽」を、「音楽を特徴付けている要素」に分解して捉える見方を促すと考えられること
- ③命令がもたらす結果をすぐに知ることができるため、俊敏さを伴う試行錯誤を促すことができること。また、そうした経験によって、音楽的結果を推測する力をも高めることが期待できること

### 3.5. 〈ティンカリング〉と〈プログラミング的思考〉との関わり

本節では、プログラミング活動における〈ティンカリング〉とプログラミング教育のねらいの1つである〈プログラミング的思考〉との関わりについて検討する。

〈ティンカリング〉と〈プログラミング的思考〉を比べたとき、その思考の過程の違いに加えて、次のような違いもある。

〈プログラミング的思考〉は、「思い付きや当てずっぽうで命令の組合せを変えるのではなく」と記されているように（文部科学省 2020a, p. 15）、ある程度の明確な根拠をもって試すべき行為を決定する。他方、〈ティンカリング〉では、本章で論じてきたとおり、行為を試してみることが重視されている。たとえある種の不確かさ—いわば「思い付きや

当てずっぽう」—が含まれていたとしても、まずは試し、その結果を即時に省みることが重要なのである。それはまた、「自分にとって価値のある音や音楽」を、試行錯誤の中で偶然見出すことにもつながると考えられる。

この点に目を向ければ、〈ティンカリング〉と〈プログラミング的思考〉とは、一見相容れない思考のようにも見える。

しかしながら、〈ティンカリング〉には、〈プログラミング的思考〉の一部である「一つの動きに対応した記号」を理解するプロセスが含まれている。

例えば、アプリケーションを〈ティンカリング〉する過程には、そのアプリケーションで出すことのできる命令と、その結果との一対一の因果関係を学ぶことが含まれている。この学びを〈プログラミング的思考〉に接続するためには、「どのように組み合わせたらいいのか」といった思考へとつなげていくことが必要となる。行為を試してみることを重視する〈ティンカリング〉では、俊敏さが大きな利点であると同時に、それだけでは論理的思考を働かせる機会が失われる可能性をも孕んでいる。〈ティンカリング〉によって納得できる音楽が生まれたときには、そのことに満足して終わるのではなく、その音楽が「どのような動きの組合せ」によって生じたのかを振り返り、命令と実行の結果との因果関係を論理的に考えることが重要である。

ただ、これら 2 つの因果関係—命令とその結果との一対一の因果関係、および命令の組合せとその実行の結果との因果関係—は、〈ティンカリング〉を行うことで自然と学べることでもある。〈ティンカリング〉と〈プログラミング的思考〉は、違いはあるものの、相容れない思考ではなく、一部に共通する思考を含んだものとして考えることができるだろう。

### 3.6. 学習材開発における〈ティンカリング〉の強みを意識した位置付け

〈ティンカリング〉の意義 (3.4.) と、〈プログラミング的思考〉との関わり (3.5.) を踏まえ、本研究で開発する学習材においても、小学校音楽科においてプログラミング教育を実現するために、この〈ティンカリング〉の特質を取り入れることが肝要になる。ここで留意すべき点は、音楽科の表現活動、とりわけ音楽づくり活動の過程において、どのように〈ティンカリング〉を位置付けるかということである。そこで本節では、ここまで述べてきたことを踏まえ、アプリケーションを用いた音楽づくり活動を通したプログラミング授業における〈ティンカリング〉の位置付けを考察する。

ここまで述べたように、〈ティンカリング〉とは、試行錯誤を繰り返しながら目的を明確にしていく過程であり、その過程で多様な発想が引き出される。例えば、VOCALOID を使って〈ティンカリング〉を行う場合には、既成の旋律を歌わせる、自ら旋律をつくってみる、人間には歌えないような過度に難易度の高い歌唱を実現する、人間には出せないようなユニークな声を出させるなど、様々な行為が引き出される可能性がある。

その一方で、児童が〈ティンカリング〉に専念しすぎるあまり、授業者が想定した活動

の範囲から逸脱した行動が促され、協同的な活動が達成されないことも予想される。こうした逸脱を避ける場合には、道具の使用を制限することが有効だと考えられるが、そのような手立ては、前述した〈ティンカリング〉の強みを失わせることにもなりかねない。

では、小学校音楽科の音楽づくり活動において、〈ティンカリング〉をどのように位置付けるのが適切だろうか。まず考えられるのは、創作のための発想や「思いや意図」を引き出す段階—いわば創作活動の前段階として位置付けることである。このことは、学習指導要領において、音楽づくりの活動は、「音遊びや即興的に表現する」活動と、「音を音楽へと構成する」活動から成るとされ（文部科学省 2018e, p. 24）、それらのつながりについて留意しつつ進めることが示されていることにも重なる位置付けである。多様な発想を引き出すための〈ティンカリング〉と、そこから得られた発想を活かして創作する活動とを分けることによって、〈ティンカリング〉の強みを生かしつつ、逸脱を最小限に抑えることができそうである。また、たとえ引き出された発想が、結果的に採用されなかったとしても、別の機会に活かすこともできるだろう。

他方、多様な発想を引き出すための〈ティンカリング〉と、そこから得られた発想を活かして創作する活動とを分けるのではなく、〈ティンカリング〉を創作活動そのものとして位置付ける可能性もあるのではないだろうか。3.1.2. で述べたように、〈ティンカリング〉は、音楽づくり活動における「即興的に表現すること」とも多くの共通点をもつことが指摘されている（志民 2019, p. 56）。そうであれば、〈ティンカリング〉そのものをパフォーマンス化し、即興的な音楽表現活動として位置付けることによって、多様な発想を得ることと、音楽づくり活動の試行錯誤を促すことを両立することもできると考えられる。筆者は、そのための手立てとして、音楽を生成するプログラムを即興的に生成することによって表現する〈ライブ・コーディング〉と呼ばれる手法に注目した。次章では、〈ティンカリング〉をパフォーマンス化した〈ライブ・コーディング〉を取り入れるために、その特質を明らかにする。

\*本章は、筆者が2019年に発表した次の論文を加筆・修正したものである。

長山弘（2019a）「小学校音楽科におけるプログラミング教育のあり方の検討—授業実践事例を手がかりに—」『初等教育カリキュラム研究』7号, pp. 55-67.

長山弘（2019b）「小学校音楽科のプログラミング授業における〈ティンカリング〉の位置付けの検討—コンピュータ・ソフトウェアを用いた音楽づくり活動に焦点を当てて—」『広島大学大学院教育学研究科紀要. 第一部, 学習開発関連領域』68号, pp. 57-65.

## 4. 〈ティンカリング〉のパフォーマンス化— 〈ライブ・コーディング〉 —

3. において、試行錯誤によって意図が明確化される〈ティンカリング〉の特質を検討し、その教育的意義を考察した。そこで、本章では、小学校音楽科の音楽づくり活動に、〈ティンカリング〉を取り入れるための手立てとして、〈ライブ・コーディング (Live Coding)〉という表現手法に着目する。「ライブ」(生演奏)と「コーディング」(プログラムを書くこと、つまり、プログラミング)という2つの語が組み合わさったこの言葉は、文字通り、常に流れ続ける音楽を生成するプログラムを生演奏の形式で組み立てる(つまり、構築する)パフォーマンスであり、言い換えれば、〈ティンカリング〉をパフォーマンス化したものということができる。

はじめに、〈ライブ・コーディング〉による表現の具体を示す(4.1.)。次に、本研究における〈ライブ・コーディング〉の定義を、先行研究を踏まえ示す(4.2.)。そして、〈ライブ・コーディング〉におけるアルゴリズムによる音楽創作の特質を検討し(4.3.)、〈ライブ・コーディング〉の即興性とその特質を検討する(4.4.)。さらに、プログラミングの過程もパフォーマンスの1つと位置付けられる〈ライブ・コーディング〉が、演奏者であるライブ・コーダー(Live Coder)や聴衆にどのように評価されているのかをライブ・コーダーや即興演奏家の言説を手がかりに検討する(4.5.)。最後に、学習材に〈ライブ・コーディング〉を取り入れる際に期待される教育的意義を述べる(4.6.)。

### 4.1. 〈ライブ・コーディング〉による表現の具体

クラブミュージックのような音楽が流れる中、バックスクリーンにプログラムコードが表示されている。リアルタイムでコードが書き換えられ、画面が点滅するたびに音楽が変化していることにお気づきだろうか。

(細谷 2017)

これは、細谷(2017)が記した、〈ライブ・コーディング〉の一場面の描写である。細谷が取り上げている演奏風景をみると、ライブ・コーダーは机の上に置いたノートパソコンで音楽を生成するプログラムを組み立てながら、リアルタイムに音楽を演奏している。そして、ライブ・コーダーの背後に設置されたスクリーンには、そのプログラムの内容が映し出されている(図12)。



図 12 〈ライブ・コーディング〉の様子 (細谷 2017)

近年、細谷 (2017) が例示しているような〈ライブ・コーディング〉による音楽表現が増えており<sup>47</sup>、コンサートホールからバルまで幅広い現場で行われている (Roberts & Wakefield 2018, p. 294)。〈ライブ・コーディング〉によって生成される音楽は、特定のジャンルに属するものではないが、音を出しながらリアルタイムでプログラミングをすることによってつくられるため、「ミニマル・ミュージック」、「テクノ」、「エレクトロニカ」など、繰り返されるリズム・パターンが次第に変化していくことを特徴としたジャンルの音楽と親和性が高い<sup>48</sup>。

なお、細谷 (2017) が例示している場面では、プログラムによって音楽が生成されているが、〈ライブ・コーディング〉には、音楽のほか、映像を生成する場合もある。他にも、Web アプリケーションや機械学習による開発などをテーマとした勉強会や講演で、受講者を前に実際にプログラミングとその解説を行うことも〈ライブ・コーディング〉の 1 つとされている (田所 2018, p. 12)。実際、動画共有サイト YouTube にて、「livecoding」という語で検索を行い、上位 100 件を参照したところ、2019 年 9 月 11 日時点で、音楽以外にも様々な表現を生成している〈ライブ・コーディング〉がみられた。本稿では、〈ライブ・コーディング〉のなかでも、とりわけ音楽表現に焦点を当てたい。

そこで、YouTube において検索条件を「livecoding music」とし、音楽表現を含む〈ラ

<sup>47</sup> 例えば、〈ライブ・コーディング〉によるパフォーマンスが多く行われているイベント「Algorave」のホームページには、世界各地の〈ライブ・コーディング〉によるライブ情報がとりまとめられているが、それを参照すると、2016 年には 33 件、2017 年には 45 件、2018 年には 51 件の公演情報が記述されている (Algorave n.d.)。

<sup>48</sup> 細谷 (2017) が「クラブミュージックのような音楽」と記しているのは、これらの音楽を汎称しているものと思われる。

ライブ・コーディング」に絞り込んだところ、2019年9月11日時点では上位50件の結果の内45件について、「コンピュータを用いて音楽を生成するプログラムを作成し、実行しながら同時に編集すること」、「リアルタイムに音楽を生成すること」、「プログラムの内容が視聴者に共有されていること」といった点において、同様の演奏形態を採っていた。つまり、〈ライブ・コーディング〉による音楽表現を味わうという行為には、次の2つの側面があるといえる。

1つはライブ・コーダーが創り出す音楽を聴くという側面である。もう1つは、その音楽を生み出すためのプログラムが即興的に作られていくのを観るという側面である<sup>49</sup>。

#### 4.2. 本研究における「〈ライブ・コーディング〉による音楽表現」の定義

本節では、本研究における「〈ライブ・コーディング〉による音楽表現」について定義する。まず、これまでの〈ライブ・コーディング〉の定義を参照する。

久保田（2005, p. 149）は「プログラムを実行させながら、同時にプログラムを書いたり修正すること」と述べている。また、久保田・畠中（2018, p. 178）は、2000年代から始まった、プログラム言語を直接操作することで行う、リアルタイム・パフォーマンスと定義している。これらの定義をみると「どのように生成するのか」といった手法に焦点が当てられているが、「何を生成するのか」については触れられていない。

他方、Ward et al.（2004）による〈ライブ・コーディング〉の定義によれば、生成されるものとして音楽と映像が挙げられている。

コンピュータを使った生演奏による音楽や映像は、今や、それらを生成するアルゴリズムを実演しながら編集することができるようになった。通常、これらに用いる機器は演奏前にあらかじめ決めておくものの、構築するアルゴリズムについては、実行しながら柔軟に変更や修正を加えることができる。

（Ward et al. 2004, p. 243）

この定義では、音楽や映像を「どのように生成するか」という点について、久保田らの定義よりも具体的に述べられており、そこに「アルゴリズムの構築」という方法が示されている。さらに、Brown（2006）は、そのプログラミングの過程そのものが「パフォーマンス」として位置付くものであるとしている。

以上を踏まえ、本研究では、次の3点を満たす行為を「〈ライブ・コーディング〉による音楽表現」として定義し、また、〈ライブ・コーディング〉という語も「〈ライブ・コーディング〉による音楽表現」と同じ概念として扱うこととする。

<sup>49</sup> 「Algorave」のガイドライン（Algorave 2017）に「演奏／演目はたいてい、コードの記述とアルゴリズムによる音楽や映像の生成が含まれていて、そのコードとアルゴリズムのプロセスが可視化されていないければなりません」と示されているように、中にはこの側面を重視しないライブ・コーダーもいる。



- ①音楽を生成するためのアルゴリズムを、コンピュータでプログラミングすること
- ②そのプログラムは、演奏時にリアルタイムに作られ、また、実行されながら、同時に編集されること
- ③プログラミングは、それ自体がパフォーマンスの一部として位置付くこと

実際、4.1. において、YouTube で「livecoding」という語で検索した際には、上位 100 件の結果の内 38 件が<sup>50</sup>、また、「livecoding music」と音楽表現に絞り込んで検索した際には、上位 50 件の結果の内 45 件が、定義として挙げた行為①～③の条件を兼ね備えたものであった。

### 4.3. 〈アルゴリズム作曲〉の概要と特質

4.2. の〈ライブ・コーディング〉の定義において、①「音楽を生成するためのアルゴリズムを、コンピュータでプログラミングする」ことを挙げたが、アルゴリズムによって音楽を生成すること自体は、〈ライブ・コーディング〉が一般化する前から行われており、〈アルゴリズム作曲 (Algorithmic composition)〉と呼ばれている。

そこで、本節では、定義の①にて〈ライブ・コーディング〉がアルゴリズムによって実現されると示した点に着目し、〈アルゴリズム作曲〉を整理し、その特質を検討する。

#### 4.3.1. コンピュータを用いない〈アルゴリズム作曲〉

〈アルゴリズム作曲〉とは、その語が示す通り、作曲に「アルゴリズムを用いる」ことを指す(木石 2018, p. 45)。つまり作曲家は、五線譜などに音符を記すことで固定された音響結果を考えるのではなく、演奏者がどのように演奏するのかという仕組みや手法(アルゴリズム)を考えることで作曲を行うのである。

その例として、ローズ(2001, p. 679)は、「歴史に残る最も有名なアルゴリズム作曲の作品」としてモーツァルトの《音楽のさいころ遊び (Musikalisches Würfelspiel)》(1787)を挙げている。《音楽のさいころ遊び》では、「数小節の曲の断片をあらかじめいくつか用意しておき、サイコロ遊びをしながらそれらをメヌエットに組み上げ」ていく(ローズ 2001, p. 679)という仕組みそのものが音楽作品になっている。

このように、〈アルゴリズム作曲〉の手法は、音楽表現にコンピュータが登場するはるか以前よりみられるため、〈アルゴリズム作曲〉にコンピュータを介在させた音楽を〈アルゴリズム・コンピューター音楽〉(コープ 2011, p. 296)と呼ぶこともある。

ここからは、〈ライブ・コーディング〉の定義①より、〈アルゴリズム・コンピューター音楽〉に焦点を当てて論を進める。

---

<sup>50</sup> 音楽を生成するもの以外に、映像のみを生成するもの(9件)や、Webアプリケーション、スマートフォンのアプリケーションなどを生成するもの(49件)がみられた。

#### 4.3.2. 〈アルゴリズム・コンピューター音楽〉

コンピュータを用いて最初に〈アルゴリズム作曲〉を行ったのはヒラーである（ローズ 2001, p. 683）。ヒラーとアイザクソンはコンピュータが音楽を生成するプログラムを書き、《弦楽四重奏のためのイリアック組曲（Illiac Suite for String Quartet）》（1956）を発表した。

同時期に推計学的手法に基づいて〈アルゴリズム作曲〉を行ったのがクセナキスである。クセナキスは「自由な推計学音楽（free stochastic music）」というプログラムを作り、その計算の結果として《メタスタシス（Metastasis）》（1954）、《ピソプラクタ（Pithoprakta）》（1956）、《アホリプシス（Achorripsis）》（1957）といった作品を生み出した（コープ 2011, p. 298）。推計学的手法では、ある音楽的な事象が起こる確率が定義された確率表を用いることにより音楽を生成する。ローズ（2001, p. 718）は〈アルゴリズム作曲〉に取り組んでいる作曲家のほとんどが推計学的手法を用いていると述べている<sup>51</sup>。

これらの音楽は、音楽が生成される過程に何らかの不確定な要素が含まれているが、大きく次の2つの場合に分類することができる。

ひとつは、結果として生成された音楽が固定されている場合である。このタイプの作品は、生成された音楽に作品としてのアイデンティティが付与されており、何度演奏しても同じ音楽が生じることになる。先に挙げたヒラーとアイザクソン、クセナキスの作品のほか、ケージの《易の音楽（Music of changes）》（1951）などもその代表例である。

もうひとつは、結果として生成された音楽が固定されていない場合である。このタイプの音楽は、演奏するたびに異なる音響が生じることになる。前述の《音楽のさいころ遊び》などがその代表例である。

いずれも〈アルゴリズム作曲〉には違いないが、本研究で取り上げる〈ライブ・コーディング〉は、生演奏で即興的にプログラミングを行うという性格をもつため、必然的に後者のタイプになる。

これら 1960 年代に始まった「コンピュータに作曲させる研究」は、1970～80 年代にはやや静まったものの、1990 年代に入って再び活発になり始めた（後藤 2016, p. 172）。そして、先に示した、コンピュータが自動的に作曲をする手法に対して、新しく生まれた考え方がコンピュータ・アシステッド・コンポジション（Computer Assisted Composition）と呼ばれる、コンピュータに作曲を支援させる考え方である。そこで用いられる代表的なプログラムの1つが、IRCAM（フランス国立音響音楽研究所）が開発した OpenMusic である。OpenMusic を使うことで、作曲家は複雑なプログラミングをする必要がなく、既に用意された機能を組み合わせたり、編集したりすることによって、音楽を生成することができる（後藤 2016, p. 174）。

---

<sup>51</sup> ローズ（2001, p. 688）は、推計学的手法を「確率的（stochastic, probabilistic）な手法」と呼んでいる。

他方、〈ライブ・コーディング〉の場合、コンピュータは、演奏の過程を担うという点でも重要な役割を果たしている。そこで、ここからは、演奏をするための道具—いわば「楽器」—としてのコンピュータの歴史を簡便に振り返る。

#### 4.3.3. コンピュータを活用した「音響合成」

1960年代、コンピュータは、音を生成するための波形を計算し、出力すること（「音響合成」<sup>52</sup>）が可能となり、「単一音から音楽作品全体までどんな音でも生み出すことができる」ようになった（Howe 1996, p. 343）。Howeは、その草分けの作曲家としてとして、マシューズを挙げている（p. 343）。1968年にマシューズが作成した音響合成プログラム Music Vは、「今日存在する多数のサウンド合成プログラムの基礎」として位置付けられている（後藤 2016, p. 135）。この頃の音響合成について、Kassler & Howe（1994, p. 200）は「1分の曲を作るのに200分はかかった」と述べている。このように多くの時間を費やす必要があったコンピュータは、演奏会等でリアルタイムに活用できるものではなかった。

しかし、1980年代以降、IRCAMが、素早く音響合成を行える音響処理専用のボード ISPW（IRCAM Signal Processing Workstation）を開発したことによって、演奏会等においてリアルタイムで使用する、いわゆる〈ライブ・エレクトロニクス〉が可能になり、1990年代になると、個人のコンピュータでもリアルタイムに音響合成することができるようになった（田所 2018, p. 17）。

〈ライブ・コーディング〉を行うためには、これまでに述べてきたような、作曲と演奏の両方がリアルタイムにできる、ということが必要条件となる。1990年代に入って、個人のコンピュータでもこれらの活動が実現できるほどの性能をもてるようになり、〈ライブ・コーディング〉のための技術的土壌の準備が整ったのである。さらに、Max, Pure Data, SuperCollider といった、個人のコンピュータで作曲と演奏をリアルタイムに行うことができるプログラミング言語が一般に普及したことも<sup>53</sup>、〈ライブ・コーディング〉の興隆に多大な影響を与えた。

ここまで、〈アルゴリズム作曲〉の概要を述べてきた。それを踏まえて、〈ライブ・コーディング〉における〈アルゴリズム作曲〉の特質として次の3点が指摘できる。

- ①生成される音楽だけでなく、音楽を生成するための過程にも重きがおかれること
- ②音楽を生成するための過程に、しばしば何らかの不確定な要素が含まれること
- ③複雑なプログラムを瞬時に計算し、リアルタイムに音楽を生成するために、コンピュー

<sup>52</sup> 「音響合成」とは、「デジタルの信号で音波を作ることによって、音（音楽でも言葉でも何でも）を合成する」ことであり、「そのデジタル信号はデジタル・アナログ変換器を使って音に変換される」とされる（Howe 1996, p. 343）。

<sup>53</sup> プログラミング言語 Max は、音響をリアルタイムに合成したり、一般的な楽器の演奏をリアルタイムに分析したりできるだけでなく、映像や照明効果などを扱ったりすることもできる（コープ 2011, p. 276）点で、ライブ演奏に適している。

タが不可欠な役割を担うこと

#### 4.4. 〈ライブ・コーディング〉の即興性

4.2. の定義②にて「そのプログラムは、演奏時にリアルタイムに作られ、また、実行されながら、同時に編集される」と示したように、〈ライブ・コーディング〉では、事前に用意した通りにアルゴリズムを構築し披露するのではなく、常に「いま、ここ」の状況とのフィードバックのなかで、アルゴリズムを編集し続けることが求められる。

ここでは、そうした即興性を考えるために、まずは〈ライブ・コーディング〉の演奏がどのような過程によって実現されているかを確認する。

〈ライブ・コーディング〉において、演奏は、まず、何らかのプログラムを作成するところから始まる。ライブ・コーダーは、白紙の状態から 1 つずつの命令を組み合わせ、あるいはテンプレートとして予め用意したプログラムを編集することで、音楽を生成するプログラムを作成する。そのプログラムの中に含まれる命令は、次の 2 種類に大別される。1 つは、新たに音を鳴らすための命令であり、もう 1 つは、いま、鳴っている音に変化を与えるための命令である。前者は、音がない、あるいは、すでにいくつかの音が鳴り続けている場面に、新しい音を加える場面に行われる。一方、後者は、すでにある音の強弱やリズム・パターンなどを修正したり、音そのものを止めたりする場面に行われる。

ある程度のプログラムができた段階で、ライブ・コーダーはそのプログラムを実行し、実際の音楽が始まる。すなわち〈ライブ・コーディング〉においては、プログラミングの開始時点演奏開始とするならば、実際に音楽が始まるまでにいくばくかのタイムラグが生じることになるのである。そのため、この時のライブ・コーダーには、初めのプログラムをいかに素早く構築できるかという手際の良さが求められる。

初めのプログラムが実行され、音楽が流れ始めると、次に、ライブ・コーダーは、即興的に新たなプログラムを加えたり、すでに作ったプログラムを編集したりすることによって、音楽に変化をもたらす。ここでも、適切なタイミングで適切な変化をもたらすために手際の良さが求められる。手際よくプログラムを編集するための方策として、ライブ・コーダーたちは独自の工夫をしているようで、例えば、田所（2018, p. 120）の場合、リズム・パターンに関わる命令やローパスフィルタ、ディレイといった音響効果に関わる命令など、「役割ごとにコードを整理しておくことで、パフォーマンスの際に臨機応変にコードを編集しやすくな」という工夫を挙げている。

〈ライブ・コーディング〉の演奏は、流れている演奏をライブ・コーダーが止めることによって終了する。その手法は命令を次第に停止させていくこともあれば、全ての命令を一度に停止するという手段をとることもある。また、プログラムが強制終了してしまったり、エラーによって演奏が中断してしまったりするといったライブ・コーダーの意図に反することも起こりうる。このように〈ライブ・コーディング〉の演奏は、プログラムの作成・実行、編集、終了と、音楽の作曲・演奏、変化、終了という過程が対応して成立して

いるといえるのである。

こうした一連の過程を即興的にすすめていくためには、単にどのような音楽を実現したいかだけを意識するだけでは不十分である。即興演奏家のベイリー（1981, p. 115）が「即興演奏することと、周囲に反応しないこととは、あいられない」と述べているように、作成したプログラムと実際に生成される音楽、それに対する聴衆からの反応、他にも会場の広さや音響などの環境も含めた、あらゆる要素によって構成される「いま、ここ」の雰囲気を感じ取りながら、そこにふさわしいプログラムを瞬時に構築していく必要もあるだろう。実際、〈ライブ・コーディング〉に用いられるプログラミング言語 TidalCycles の開発者でありライブ・コーダーでもある McLean は次のように述べている。

私がパフォーマンスをする間、私の目の前では観客が反応して踊り、エナジーを返してくれます。この時の私はコードの世界にのめり込んでいるのですが、体験は場全体から得ているのです。

（レッドブル・ミュージック・アカデミー n.d.）

このように、ライブ・コーダーはプログラムとその結果（つまり、音楽や聴衆の反応など）との関係に意識を向けることになる。加えて、その結果は〈ライブ・コーディング〉に「プログラムまたはその結果を、もっと〇〇すればよかった」、「プログラムまたはその結果に、△△を加えれば（あるいは、削除すれば）、より面白いプログラムや演奏になる」といった思いをもたらす。そして、その思いをもとに次に続くプログラムを編集する。図 13 は、この過程におけるプログラム（アルゴリズム）と音楽との関係を示したものである。実線が演奏の過程を、点線は影響を与えるものを示している。

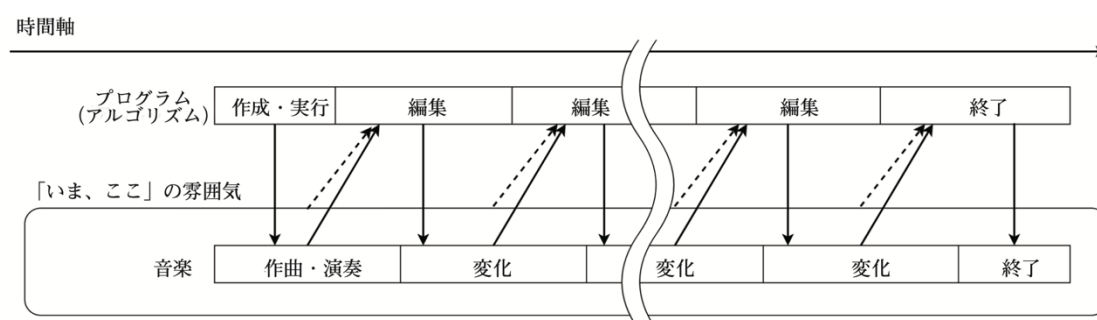


図 13 〈ライブ・コーディング〉におけるプログラム（アルゴリズム）と音楽との関係

すなわち〈ライブ・コーディング〉とは、自らの構築するプログラムによって出力される実際の音楽が「いま、ここ」の状況を生み、その状況は次に構築するプログラムに影響するという、絶えざる循環によって成立している。ライブ・コーダーの即興的な思考力や判断力、そして表現力は、そうした循環のなかで、「いま、ここ」そして「その先」の状

況をいかに作り出すかに向けられる。

このこと自体は、一般的な楽器を用いた即興演奏の多くにも共通することである。しかしながら〈ライブ・コーディング〉においては、演奏者が直接、音そのものを変更する命令（例えば、音程や音価の変更）を出すだけでなく、いま生じている音響をもたらす仕組みそのものを変化させる命令（例えば、音程や音価を決定するための計算の過程の変更）を出すことができる。このことは、わずかなプログラムの変化が、音響結果に大きな変化をもたらす場合があることを意味する。さらに、そこに不確定な要素が含まれている場合には、次にどのような音響をもたらされるかを予想することが困難となる。ほかにも、演奏の過程をコンピュータが担うという特質は、演奏者の予想が難しいコンピュータ・トラブルによって、演奏が中断される危険性も孕んでいる。

以上のことから、〈ライブ・コーディング〉の演奏における特徴として、一般的な楽器を用いた即興演奏に比べて、その先の状況が予測し辛く、全く予測できない状況をもたらされる可能性が高い点が指摘できる。

#### 4.5. パフォーマンスとしてのプログラミング

4.2. の定義③で示したように、〈ライブ・コーディング〉では、音楽だけではなく、プログラミングそれ自体もパフォーマンスとなる。したがって、〈ライブ・コーディング〉に特有の演奏の評価が存在していることが推察できる。では、〈ライブ・コーディング〉における「良い演奏」とは、どのような演奏だろうか。この問いに答えることは簡単ではない。本節では、即興演奏家たち、また、ライブ・コーダーたちの言説を手がかりに、〈ライブ・コーディング〉の演奏に対する評価の観点を検討する。

##### 4.5.1. 「いま、ここ」への対応力

4.4. で検討したように、〈ライブ・コーディング〉とは即興演奏である。即興演奏の場合、単に、聴衆に届けられる音楽そのものの出来栄だけで〈ライブ・コーディング〉を評価することは不十分である。まずは、〈ライブ・コーディング〉が即興演奏であることに焦点を当てて、〈ライブ・コーディング〉の評価の観点を検討する。

「即興演奏することと、周囲に反応しないこととは、あいられない」（ベイリー 1981, p. 115）という言説にみられるように、多くの即興演奏家たちが強調していることのひとつに、演奏者の「いま、ここ」への対応力が挙げられる。それは、演奏をどのように始めるか、どのように変化させるか、どのように終了させるかはもちろんのこと、聴衆や演奏会場の雰囲気や予期せぬトラブルやアクシデントなどにもまた、「いま、ここ」への対応力が反映される。

「いま、ここ」に適切に対応するための技能として、1つには4.4. で述べた手際の良さが挙げられる。加えて、生じる状況に対していかに対応するかを予め想定しておく力も求められるだろう。こうした力を育むことにより、演奏に安定が生まれ、次の過程で何が起

きるかわからないといった不安定さを避けることができる。しかしながら、演奏の安定は、即興演奏においては必ずしも求められることとは限らないことを示す意見もある。即興演奏家でもあり、即興演奏の研究者でもあるナハマノヴィッチは、次のように述べている。

すばらしいジャズ演奏家なら誰でも、フレーズにつまったときに復調させる技を数知れず持っています。しかしインプロヴァイザーであるためには、そういった技を捨て、危険を冒し、リスクを取ります。ときには大失敗に陥ることもあります。しかし、実をいいますと、観客が一番好きなことは、あなたが何かを試みて、しくじることです。そしてどのように立ち直り、世界を復元させるかを見ることなのです。

(ナハマノヴィッチ 2014, pp. 43-44)

すなわち安定よりも、不安定のなかにあるスリルや予期せぬ出来事への対処もまた、即興演奏の魅力成す重要な要素であるといえるのである。

このような不安定は、とりわけ〈ライブ・コーディング〉においては、プログラミングにおけるエラーやシステムエラー、あるいはプログラムが自分の思い通りの結果にならないことなどのアクシデントによってしばしば引き起こされる。久保田 (2005, p. 154) が、「ライブ・コーディングのパフォーマーは、常に（あえて）リスクを背負っている」と述べているように、ライブ・コーダーは、リスクとアクシデントへの対処を、演奏の魅力的な側面として意識している。例えば、ライブ・コーダーであり、ライブ・コンピュータ音楽の研究者でもある Knotts は次のように述べている。

私は、リスクがある状況や、高い認知負荷がかかる状況で演奏をするのが好きです。また、意図しないアルゴリズムの結果が、どのような音楽構造をもたらすかに興味があります。そのため私は、エラーのリスクを減らし過ぎないようにして、ライブ・コーディングをします。

(Roberts & Wakefield 2018, p. 295)

また、McLean はまだ十分に習得していない技術をあえて使うことによってもたらされる新鮮さとスリルについて、同様の見解を述べている。

即興のライブコーディングでは、「練習しすぎ」になりがちです。洗練されすぎたパフォーマンスをしてしまっただけでは、あまり楽しくありません。これに対処する方法は、パフォーマンスの直前に新しい機能やテクニックを学習することです。パフォーマンス中に、新しく学んだことを試したり、そこから学ぶことで、新鮮さを感じることができま

(田所 2018, p. 34)

ここでは、経験を積むことによって熟練していく手際の良さや「いま、ここ」への対応力だけでなく、あえて新しい（不慣れな）方法を取り入れることによってもたらされる不安定さの重要性が強調されている。このことは、予め楽譜に記された音楽を、いかに魅力的に表現するかに力を注ぐことの多い非即興演奏家にはみられない発想だといえるだろう。すなわち「いま、ここ」への対応力は、〈ライブ・コーディング〉の演奏に対する評価を左右する観点となり得る可能性があると同時に、そうした対応力を引き出すための重要な要素が、演奏中に起こる予想外のアクシデントだといえる。

このことから、〈ライブ・コーディング〉では、手際のよさによる安定した演奏と、それを壊す予想外のアクシデント、さらにそれへの緊急対応力、これらが絶妙に混在することによって魅力が引き出されると考えられていることがわかる。

#### 4.5.2. プログラムを構築する過程

〈ライブ・コーディング〉では、その多くにおいてプログラムを構築する過程が聴衆に示される。それは、「今、演奏者が何をやっているのか」を聴衆に示すことでもある。久保田・畠中（2018）は、〈ライブ・コーディング〉の演奏風景の一例を次のように示している。

プログラミングが好きな NERD [コンピュータおたく] たちが、四つ打ちレイブ [等間隔に打ちならされるリズムを用いたダンス・ミュージック] のようなステレオタイプの既成音楽のなかで、単なるプログラミング自慢合戦を繰り広げている。（[] 内は筆者による補足）

（久保田・畠中 2018, p. 179）

久保田・畠中が「プログラミング自慢合戦」という言葉を用いているように、〈ライブ・コーディング〉ではプログラミングがいかに優れているかについても評価の要素となる。ここで、久保田・畠中が、「四つ打ちレイブのようなステレオタイプの既成音楽」、「単なるプログラミング自慢合戦」と批判的に述べていることから、少なくとも久保田・畠中には、プログラミングの過程のみに着目することを良しとしない態度が窺える。しかしながら、この指摘は、プログラミングの過程に重きが置かれがちであるという、〈ライブ・コーディング〉の現場の実態の一端を表しているともいえよう。

本項では、プログラムを構築する過程がいかに優れているかという点に着目してその評価の観点の検討を行う。そのために、まずは〈ライブ・コーディング〉においてプログラムを構築する過程が示されるようになった経緯を整理し、次にプログラムそのものに対する評価のあり方を検討する。



#### 4.5.2.1. プログラムを構築する過程が聴衆と共有されるようになった経緯

コンピュータを用いた音楽表現は、声楽や器楽による音楽表現に比べ、演奏者やコンピュータがどのようなことをしているのかが聴衆に伝わりにくい。ときには、聴衆に「何もしていない」と思われてしまうこともあったという (Collins 2011, p. 209)。久保田 (2005, p. 148) や田所 (2018, p. 18) は、90年代以降、ライブ演奏にノートパソコンを取り入れるようになった演奏形式が、このような問題を抱えていたと指摘している。音楽を生成するアルゴリズムを構築する過程をスクリーンに投影するという方法は、その問題への解決法の1つとして登場したのである。

田所 (2018, pp. 19-20) によれば、ステージ上のスクリーンにプログラムを投影し、プログラムを編集しながら演奏するという形式で、はじめに〈ライブ・コーディング〉を実践したのは、McLean や Ward, Griffiths らによって結成されたバンド slub である。slub の最初の演奏は2000年7月15日にロンドンの Hoxton Foundry で行われた (Slub n.d.)。彼らは、演奏中にアルゴリズムを構築、編集できるプログラミング言語を自作し用いていた。Collins et al. (2003, pp. 323-326) による演奏例では、命令が記述されたブロック同士を繋ぎ合わせるプログラミング言語 Map と、プログラムを1行ずつ記述したり、画面上のスライダーを動かしたりできるプログラミング言語 Pure Events が紹介されている。

プログラムを構築する過程を明らかにすることは、演奏者がその思考を聴衆と共有することへとつながる。聴衆は「なぜ、このような命令を書くのか」といったことを考えながら、その結果生成される音楽を聴くことを通して「だから、このような命令になったのか」、「あそこに表示されている数字は音色に結びついているのだな」といった因果関係をも体験することができる。

slub が様々なプログラミング言語を用いたように、今日のライブ・コーダーたちもまた、多種多様なプログラミング言語を用いている。伝統的な演奏家が演奏会の性格に応じて用いる楽器を使い分けるように、ライブ・コーダーもまた、自らの表現に適したプログラミング言語を選ぶのである。プログラミング言語が異なれば、どのようなアルゴリズムが実現しやすいかも変わる。さらに、プログラミングの過程が可視化される〈ライブ・コーディング〉においては、どのようなプログラミング言語を採用するかを考えることは、聴衆に届けられる視覚的情報がどのようなものになるかを考えることでもある。

#### 4.5.2.2. プログラムそのものに対する評価のあり方

次に、4.5.2.1. の経緯を踏まえて、〈ライブ・コーディング〉におけるプログラムそのものに対する評価のあり方を検討する。このことを考えるために、まず、いったん〈ライブ・コーディング〉から離れ、一般的なプログラミングにおける価値観を確認したい。

プログラムを記述する際に注意する点は書き手によって様々である (池本 2016, p. 17)。とはいえ一般的に、プログラムには意図したとおりの結果を生成することのみならず、書き手の意図が読み手に伝わるということが重要だとされている (高橋・谷口 2017, p. 11)。実際、

Roberts & Wakefield (2018, p. 300) によれば、〈ライブ・コーディング〉に用いられるプログラミング言語も、聴衆が理解しやすい文法になるように設計されている。その上、こうした要素は時として人に感動すら与える場合がある。プログラマーのボズウェル・フォシェ (2012, p. x) は「美しいコードを見ると感動する。優れたコードは見た瞬間に何をしているかが伝わってくる」と述べ、意図の伝わりやすさの重要性を強調している。ここで特筆すべきは、彼らが単に「意図のわかりやすいプログラム」を、合理性や利便性の高さだけでなく、「美しい」、「感動する」といった美的感覚に基づいて評価している点にある。この種の感動は、〈ライブ・コーディング〉においてももたらされる可能性のあるものである。久保田・畠中 (2018, p. 179) のいう「プログラミング自慢合戦」の背景には、このことが多分に意識されていることが窺える。

ただし、〈ライブ・コーディング〉においては、この一般的なプログラミングにおける価値観が、必ずしも適用されるわけではない。久保田 (2005) は次のように述べている。

これまでプログラムコードは、きちんと確実に動くこと、シンプルで効率良く動くこと、高速に動くことがまず求められてきた。それはコードがコードであるための前提であり、ゴールであり、そしてコードの美のルーツでもあった。しかしそうしたコードの美は、ノイズとしての人間が介入するパフォーマンスの実践とはあまり関係がない。

(久保田 2005, p. 153)

生演奏である〈ライブ・コーディング〉では、プログラミングにかけられる時間に制約があるという性格上、十分に推敲された最適なプログラムを構築することは困難でもある。加えて、構築されるプログラムは不完全なこともありうる。筆者が McLean に「プログラムを書く時に気をつけていること」について訊ねたところ、彼から以下のような回答を得た<sup>54</sup>。

演奏中は読みやすさに気をつけるよりも、音楽を聴くことと、それをどのように変化させるかということに集中している。(中略) スクリーンに投影することは、私の音楽の作り方を素直に見せるためだけにある。

つまり、彼にとって、スクリーンにはありのままの思考過程が投影されているのである。ありのままの思考過程を聴衆が見たとき、それを理解しやすいと感じるかどうかは、ライブ・コーダーが用いるプログラミング言語に左右される。①で示したように、ライブ・コーダーは多くのプログラミング言語の中から演奏に用いるプログラミング言語を選択したり、自作したりする。聴衆によっては、プログラミング言語 Max などによる、命令が記されたブロックを組み合わせることで構築されるプログラムが、視覚的に理解しやすいと感

---

<sup>54</sup> 2019年5月4日～5日に、Twitterのダイレクト・メッセージによって連絡をとった。

じるものもいるだろう。

どのようなプログラミング言語を選択するかということは、①で述べた因果関係を聴衆が体験することとも関連する。聴衆の背景として、プログラミングを経験したことがある人、また全く関心がない人と様々なものがあると考えられるが、ここでは、「演奏者の用いるプログラミング言語を知っている場合」、「プログラミングの経験をもつ場合」、「プログラミングに全く関心がない場合」に分けて検討したい。

#### ○聴衆が、演奏者の用いるプログラミング言語を知っている場合

演奏者の用いるプログラミング言語を知っている場合、聴衆は、演奏者がプログラムを構築する過程を見ることで、コンピュータが結果を出力する前にプログラムの計算結果を先読みしたり、生成された音楽をもとにプログラムを評価し、自分がライブ・コーダーの立場ならばどのようにプログラミングするかを考えながら聴いたりすることができる。

#### ○聴衆が、プログラミングの経験をもつ場合

聴衆が演奏者の用いるプログラミング言語を知らなくとも、プログラミングの経験が持っているならば、スクリーンに映し出された数値や記号が、結果として出力される音楽にどのように影響しているのか、またはしていないのか、実際に出力された結果を手がかりに、それを生成するプログラムを分析しながら聴くことができるだろう。

さらに、視覚的に理解しやすいプログラミング言語であれば、プログラミングに全く触れたことのない聴衆にとっても、画面上に示されたプログラムと実際に鳴り響いている音楽との因果関係を、ある程度推察することもできると考えられる。

#### ○聴衆が、プログラミングに全く関心がない場合

画面上に表示されるプログラムに関心を持たない聴衆は、当然のことながらプログラムを構築する過程を評価の対象に含まず、生成される音楽のみを手がかりに演奏を評価することになる。

このように、〈ライブ・コーディング〉において提示されるプログラムが「誰が見ても理解できるもの」が目指されるのかどうかはライブ・コーダーのコンセプト次第であり、聴衆の評価の観点についても、聴衆 1 人ひとりの知識や態度のあり方次第ということになる<sup>55</sup>。

## 4.6. 〈ライブ・コーディング〉に期待される教育的意義

本章では、ここまで、〈ライブ・コーディング〉による音楽表現がどのような特質をもった行為なのかという考察を、実践や構造、様々な言説を頼りに試みた。その結果、〈ラ

<sup>55</sup> これまでに〈ライブ・コーディング〉を競技化したイベントもみられる。例えば、2005年5月21日にベルギーのヘントで行われた「Coding Bull」(Toplap 2010)、2005年9月7日にスペインのバルセロナで行われた「Raging code」(Toplap 2009)、2007年6月28日にオランダのアムステルダムで行われた「the Max Live Coding contest」(Paulus n.d.)など。このうち、「the Max Live Coding contest」では、Maxを用いて6分間の〈ライブ・コーディング〉を行い、聴衆から最も大きな拍手を受けた参加者が勝者となるというルールで行われた (Paulus n.d.)。

ライブ・コーディング)における〈アルゴリズム作曲〉の特質、〈ライブ・コーディング〉の独自の即興性、評価の実態の一端について、それぞれ示すことができた。本節では、それらを踏まえた上で、〈ライブ・コーディング〉を小学校という学校現場に取り入れることを前提に、そこにどのような教育的意義が期待できるかを検討する。

〈ライブ・コーディング〉の表現過程において、コンピュータを用いてその場に流れ続けている音楽を即興的に変化させることは、児童が「コンピュータをいじくりまわす」行為であると同時に「音楽そのものをいじくりまわす」行為でもある。すなわち〈ティンカリング〉を二重の意味で含む過程といえる。

ここで「即興的に表現すること」について学習指導要領における位置付けを確認する。小学校音楽科の音楽づくり活動は、「音遊びや即興的に表現する」活動と「音を音楽へと構成する」活動に分けられるが、「即興的に表現すること」は、これらの前者に含まれる(文部科学省 2018e, p. 23)。ここでの「音遊び」とは「設定した条件に基づいてその場で音を選んだりつなげたりして表現することによって、音楽づくりへの様々な可能性を広げていくもの」、「即興的な表現」とは「いろいろな音の響きを基に、設定した条件に基づいて、即興的に選択したり組み合わせたりして表現すること」とそれぞれ定義されている(文部科学省 2018e, p. 133)。

3.3.3. において、筆者は、〈ティンカリング〉が、学習者の発想を引き出すために「道具」(アプリケーション)や「材料」(音)が、学びにつながる重要な役割を果たすことを述べた。また、3.6. では、「まずは試してみる」という精神が様々な発想を得るために必要であり、プログラミング教育でも大切となることを述べた。この〈ティンカリング〉における実験的な—いわば、「思い付きや当てずっぽう」な—試行過程は、〈ティンカリング〉を続ける過程で、次第に意図を帯びた命令へと変わっていく。このことは、「音楽を特徴付けている要素」とその働きの理解を、即興的な活動を通して促進することを意味する。

さらに、4.2. の〈ライブ・コーディング〉の特質で示したように、演奏にあたっては、プログラムの結果として生じた音楽や映像だけでなく、プログラムを構築する過程自体も、聴衆が見えるよう配置したスクリーン等にリアルタイムに映し出して表現の一部となる。演奏者が音楽をつくる過程を聴衆と共有すること—言い換えれば「聴衆の存在を意識した〈ティンカリング〉」—は、聴衆役の児童にとっても、「音楽を特徴付けている要素」に関わる命令がどのような働きをするのかを視覚と聴覚によって確認できることへとつながる。つまり、〈ライブ・コーディング〉において、コンピュータを用いて音楽そのものをいじくりまわし、意図を明確にしていく活動は、演奏者となる児童個人の体験に留まるだけでなく、聴衆役となる他の児童と共有されることによって、双方の理解が促されることが期待される。

次章からは、〈ライブ・コーディング〉による教育的意義を踏まえ、小学校音楽科の音楽づくり活動においてプログラミング教育を実施するための学習材の設計と開発を行う。

\*本章は、筆者が2019年に発表した研究報告と、2022年に発表した論文を加筆・修正したものである。  
長山弘（2019）「〈ライブ・コーディング〉による音楽表現の特質に関する一考察」『音楽表現学』Vol.17,  
pp. 63-72.  
長山弘（2022）「〈ライブ・コーディング〉の特質を取り入れた音楽づくり活動のための Web アプリケー  
ションの開発」『日本教育工学会論文誌』46 卷 3 号, pp. 567-578.

## 【第二部】実践編

### 5. 小学校音楽科におけるプログラミング教育のための学習材の開発

【第二部】では、【第一部】の議論を踏まえて、小学校音楽科におけるプログラミング教育のための学習材として、〈ライブ・コーディング〉の特質を取り入れたアプリケーションの開発に取り組む。具体的には、次の過程で進める。

- ① 学習材開発のための方針を整理し、学習材となるアプリケーションを実際に開発すること（5.）
- ② 児童を対象に、開発した学習材を用いた授業実践を行い、開発の方針が実現できたかどうかを検証すること（6.）
- ③ ②の結果を踏まえ、学習材を改善すること（7.）

本章では、学習材開発のための方針を整理する。そのために、まず、本学習材がめざすプログラミング教育を確認する（5.1.）。次に、そのねらいを達成するための手立てを、これまでの議論を踏まえて検討する（5.2.）。さらに、講じた学習材を用いた既存の授業事例を概観し、そこでの問題点を検討する（5.3.）。検討した点をもとに、開発する学習材の開発の方針と工夫（5.4.）、そして、開発した学習材の具体を示す（5.5.）。最後に、設けた方針と工夫について、開発を通して達成された点と授業実践を通して検証する点を確認する（5.6.）。

#### 5.1. 開発する学習材がめざすプログラミング教育

【第一部】では、文部科学省の示す〈プログラミング的思考〉を働かせる過程と、小学校音楽科の音楽づくり活動において働く思考とは、必ずしも一致しないことを指摘した。そこで、本節では、本研究で開発する学習材がめざすプログラミング教育を、文部科学省（2020a）が「手引」で示すねらいを踏まえて整理する。

まず、プログラミング教育のねらいの①に関連して、児童の〈プログラミング的思考〉を育むことをめざすこととする。2.6. で「コンピュータを動作させるための手順（例）」（文部科学省 2020a, p. 14）を基に学習の過程を検討したところ、小学校音楽科の音楽づくり活動では、算数科における「正多角形をかく」活動という一般的なプログラミング活動と異なり、「意図した一連の活動」にあたる児童の「表現したい音楽」が明確でないことと、検証過程に「その結果が音楽的に納得できるものであるかどうか」が含まれることを挙げた。そこで、本研究では〈ティンカリング〉の過程で、コンピュータへの命令（及び命令の組合せ）と実行の結果との因果関係を考えることを通して、児童の〈プログラミング的思考〉を育むこと（3.5.）をめざす。

次に、ねらいの②に関連することとして、とりわけ「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」(p. 11) とのつながりを意識し、コンピュータの特徴への気付きを促すことをめざすこととする。とくに、速いフレーズやリズム・パターンを正確に反復する(またそれを重ねる)といった、コンピュータの得意とする音楽表現ができること、そして、それによって音楽が多様に表現できることに気付かせることに焦点を当てる。

そして、ねらいの③に関連することとして、とりわけ「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」(p. 11) とのつながりを意識し、コンピュータを用いた音楽表現の過程で、リズム、旋律、音色といった音楽的諸要素をいじくりまわしながら音楽を即興的に変化させる活動を通して、音楽科の内容、とくに、[共通事項]「音楽を形づくっている要素」のうち「ア 音楽を特徴付けている要素」の働きの理解を深めることをめざすこととする。

## 5.2. プログラミング教育を達成するための方策

5.1. で取り上げたプログラミング教育を達成するために、筆者は、「プログラムを作成」し、「それを実行して鳴り響く音楽を検証」し、再び「プログラムを編集」し…といった試行錯誤の活動ではなく、音楽が「常に流れ続けている」中で、その音楽を生成するプログラムを「リアルタイムに編集する」活動—すなわち〈ライブ・コーディング〉—を取り入れた学習材を開発する。

流れ続けている音楽をリアルタイムに修正することは、発した命令の結果を、即時に音楽として確認できることより、その因果関係を考えることへつながることが期待される。

流れ続けている音楽をリアルタイムに修正すること自体は、コンピュータを用いない即興演奏にもみられる。その例として、トンプソンによって考案された「サウンド・ペインティング (Sound Painting)」や、モリスによって考案された「コンダクション (Conduction®)」のような、指揮者が演奏者にハンドサインで即興的に指示する方法が挙げられる<sup>56</sup>。しかし、ある声部に対して「音色」、「リズム」、「速度」、「旋律」、「強弱」、「音階」、「拍」などの諸要素にかかわる指示を、即時かつ正確に命令し、音楽として反映させることはコンピュータでなければ難しい。さらに、それらを複数の声部に対して行い、結果を確認することは、「コンピュータの特徴への気付きを促すこと」へとつながることが期待される。

さらに、4.6. で示したように、〈ライブ・コーディング〉においては、実験的な試行過

<sup>56</sup> このような方法は、学校現場で取り上げられることも少なくない。例えば、小枝(2020)は、中学校段階における知的障害児と健常児を対象に「身ぶりに合わせた即興演奏」を行った(p. 48)。小枝は、授業を通して、身ぶりに合わせた即興演奏が「障害の有無を架橋し音楽科のねらいを達成できる交流及び共同学習の題材」になることを指摘している(p. 52)。

程が、〈ティンカリング〉を続ける過程で、次第に意図を帯びた命令へと変わっていくことより、「音楽を特徴付けている要素」とその働きへの理解が、即興的な活動を通して促されることが期待される。

### 5.3. 〈ライブ・コーディング〉の特質を取り入れたプログラミング教育の実践事例

〈ライブ・コーディング〉の特質を取り入れたプログラミング教育の授業実践事例について、国内においては、管見の限り見当たらない。他方、海外においては、Burnard et al. (2014) や Ruthmann et al. (2010) の事例がみられる。Burnard et al. は、11歳から14歳までの生徒を対象にした実践において、プログラミング言語 Sonic Pi を用いた実践を行い、生徒らがクリエイターの意識をもつことに有効だったと述べている (p. 25)。Ruthmann et al. は、〈コンピューティショナル・シンキング〉や音楽を専攻する大学生を対象に、Scratch を用いて、音楽を演奏するプログラムを作成・実行しながら、音楽に関わる命令を修正することで〈ライブ・コーディング〉を実践している。しかしながら、これらの実践で用いられた Sonic Pi や Scratch というプログラミング言語は、日本の小学校音楽科で〈ライブ・コーディング〉を実践するうえで2つの問題点を孕んでいる。

1つ目は、使いやすさについての問題点である。Burnard et al. (2014) の実践で用いられた Sonic Pi では、プログラムの作成にあたって、英語での文字入力が求められる上に、そのプログラミング言語に関する知識・技能をもつことが前提となっている。〈ライブ・コーディング〉に適したプログラミング言語として、Sonic Pi のほかにも Max, Pure Data, Super Collider などいくつか挙げられるが、Sonic Pi と同様の (あるいは類する) 理由で、小学校での〈ライブ・コーディング〉の実践に相応しいとはいえない。

2つ目は、実現される音楽の可能性と、聴衆にとっての理解のしやすさとのバランスに関することである。Ruthmann et al. (2010) の実践で用いられている Scratch は、多くのプログラミング教育の実践事例で活用されている、児童にも使いやすいプログラミング言語といえるだろう。しかしながら、Ruthman et al. の実践のように、予め用意されたプログラムに対して、命令内の各パラメータにのみ変更を加えていく形では、実現できる音楽の可能性が狭まり、音楽づくりの自由度も低くなってしまふ。反して、自由度を高めようとするれば、多くの命令を複雑に組み合わせた結果プログラムの見た目も分かりづらくなり、表示されるプログラムを介して音楽づくりの過程を共有することができず、聴衆が「いま、ライブ・コーダーがどのように音楽を変化させようとしているのか」を考えることも困難となる。

以上、小学校音楽科において〈ライブ・コーディング〉の特質を取り入れた学習材を開発するための課題点を示した。次節では、これらの課題を踏まえて、開発の方針を検討する。



## 5.4. 開発の方針と工夫の具体

本節では、5.3. で示した課題を踏まえて、学習材の「開発の方針と工夫」を7つの観点から示す。

### ①〈ライブ・コーディング〉の過程に着目した観点

4.4. において、「ライブ・コーディング」におけるプログラム（アルゴリズム）と音楽との関係」を図示した。この関係は、〈ライブ・コーディング〉において、ライブ・コーダーが辿る過程といえる。よって、開発される学習材においても、児童による命令が即時に音楽として鳴り響き、それを聴きながら編集できるようにしなければならない。

具体的な工夫として、児童が設けた「音」の命令ごとに、繰り返しその「音」を鳴らす処理を作成する。そして、処理の実行時に、都度、「音の高さ」や「音色」といった演奏に関わる値を取得し鳴らすことで、児童による命令が即座に結果へと影響する仕組みとする。また、スピード感をもって命令を次々と作成できるように、命令が反映されるまでに待ち時間が生じないようにしたり、すでに発した命令の内容をあらためて確認する—「〇〇を削除しても良いですか」に「はい（または、いいえ）」と答えるような一手順を省くことによって、演奏者の手数を減らしたりする。

### ②「音楽を特徴付けている要素」の理解を深めることに関する観点

開発する学習材のねらいの1つとして、「音楽を特徴付けている要素」の働きの理解を深めることを挙げた。そして、その手立てのひとつとして、命令と実行の結果との因果関係を即座に理解するために、3.4.3. でアプリケーションを道具として〈ティンカリング〉する際に、命令が演奏結果へと即座に（リアルタイムで）反映される仕組みが有効であると述べた。

本開発では、そのような仕組みを取り入れるとともに、命令と実行の結果との因果関係をより分かりやすくするために、見た目の工夫も取り入れる。具体的には、「オクターブ」の値を変更するボタンに「↑」、「↓」といったアイコンを用いたり、「パターン」の追加には「+」のアイコンを、「削除」ボタンには「ゴミ箱」のアイコンを用いたりするといった、ある命令を与えるボタンに、その効果に相応しいアイコンを併せて示す。また、「音楽を特徴付けている要素」の働きについては、聴衆役の児童にもわかりやすく示される必要がある。その工夫として、編集画面の上部に、「『音の大きさ』を変える」、「『速度』を変える」のような、命令がどの「音楽を特徴付けている要素」に関わっているのかを表示する。

### ③直観的な操作性についての観点

活発な〈ティンカリング〉を促すためには、直観的に操作できる画面設計が肝要である。

そのために、画面上のボタンをマウスでクリックすること<sup>57</sup>によって命令を操作できることを基本とする。例えば、音の長さの入力にあたっては音符のアイコンをクリックしたり、音の高さを入力する場合には鍵盤楽器画面に表示される鍵盤をクリックして値を入力したりできるようにする。また、入力された音の高さや長さは、五線譜ではなく、数値で表示する。これらの工夫により、キーボードによる文字入力が不慣れな児童でも操作しやすく、五線譜を読むのが苦手な児童にとっても、音の高さや長さを表示される数値の大小で認識できる。

次に、音色の表示画面では、数多くの音色から直観的に選択できるよう、次の3点の手立てを講じた。1点目は、**General MIDI (GM)** (音楽電子事業協会 1998) で定義されている音色名をそのまま使うのではなく、児童にとって馴染みのある名称にする。例えば、「シンセ・ストリングス1」は、小学校音楽科の教科書『音楽のおくりもの』(新実 2020) 及び『小学生の音楽』(小原ほか 2020) の記載を参考に「弦楽器群」とする。ほかにも、児童が分かりやすいように「矩形波」を「ピコピコ音1」とした一方で、「鳥の声」はそのまま示すようにする。2点目は、音色表示の並びを『ニューグローヴ世界音楽大事典』の「楽器分類法」(Wachsmann et al. 1994) の項を参考に、種別ごとに分類する。3点目は、楽器の名称についての知識がなくとも、楽器がイメージしやすくなるようにイラスト付きのアイコンを添える。

#### ④児童のつくる音楽の多様性の保障に関わる観点

坪能 (2010, p. 27) は、音楽づくりにおいて、音が出るものは全て音素材になる一方で、あらゆるものを自由に使うことによって生まれるものはカオスであり、音素材を限定したり、音階などの音組織を用いたりして枠組を設けることが音楽づくりには重要だと指摘している。他方、Terauchi & Myodo (2021, p. 220) は音楽を構成する諸要素についての制約が、児童の自由な創造性をも制限し、どの児童も似通った音楽をつくってしまう結果にもつながりかねないことを指摘している。両者の主張を踏まえ、本学習材では、音色、速度、音階、拍などに一定の枠組を設けつつも、つくられる音楽の多様性を保障するために、それらの要素をできるだけ幅広く設定できるようにする。

#### ⑤小学校音楽科の学習内容を考慮した観点

命令に関わる語として、「音楽を特徴付けている要素」や、教科書に掲載されている言葉といった、小学校の授業内で多く用いられる言葉を取り入れる。具体的には、音楽創作アプリケーションで広く用いられている「BPM」や「Beat」といった語は小学校音楽科においては馴染みが薄いため、これらをそれぞれ「速度」、「拍」と示す。ほかにも、児童が演奏に用いることができる「音色」については、GMで定義されるGMサウンド・セットより、教科書掲載の楽器を中心に抽出する。

---

<sup>57</sup> タブレット端末で操作する場合は、指でタップすることを示す。

## ⑥コンピュータの特質を生かした表現に関する観点

⑤において、用いる音色は教科書を手がかりに抽出するとしたが、それらは楽器の音のみであり、コンピュータの特質を活かした音色が含まれていない。そこで、コンピュータ特有の表現を十分に味わうために、「矩形波」や「鳥の声」の音といったコンピュータ独自の音色も加える。以上を踏まえ、児童が選択できる「音色」として、教科書で取り上げられている楽器を中心に 34 種類、コンピュータ独自の音色から 8 種類、計 42 種類を準備する。

## ⑦動作環境に関する観点

現在、GIGA スクール構想（文部科学省初等中等教育局初等中等教育企画課 2019）の下、児童生徒に 1 人 1 台の ICT 端末の整備が進められている。しかしながら、GIGA スクール構想に用いられる ICT 端末について記された「標準仕様書」（文部科学省 2020c）では、その候補を提示するのみで、学校ごとに様々な種類の ICT 端末が運用されている現状がある。また、「標準仕様書」では、学校内の端末に児童がアクセスする「オンプレミス型」ではなく、クラウド上のデータにアクセスする「クラウドコンピューティングを基本とすること」とされている（p. 2）。

よって、どのような ICT 端末でも動くように、加えて、管理者権限でインストールする必要がない、Web ブラウザ上で動くアプリケーションを開発する。なお、「手引」（文部科学省 2020a, p. 59）では、「学習者用コンピュータに搭載している Web ブラウザが対応しておらずプログラミング教材が動作しない」といった ICT 機器の環境に関わる留意点が示されている。それを踏まえ、音声出力に必要な Web Audio API をサポートする Web ブラウザ<sup>58</sup>上で動くように開発する。

## 5.5. 《テクミュ (Tec-Mu)》の具体

本節では、5.4. で示した方針に沿って開発した学習材の具体を示す。同学習材に、筆者は《テクミュ (Tec-Mu)》と名付けた。“Tec”は「テクニック (Technique)」, 「テクノロジー (Technology)」より, “Mu”は「ミュージック (Music)」より採った。

《テクミュ》は現在もバージョンアップをはかっているが、最新バージョンについては「<http://tec-mu.com>」に公開している。なお、本章で開発したバージョン（以下、「初版」）は、後の 7. で取り上げるバージョン（以下、「二版」）が完成した現時点においてはすでに古いものではあるが、本章の関連資料として「<http://v1.tec-mu.com/>」にて公開している。

<sup>58</sup> Edge, FireFox, Google Chrome, Opera, Safari など。

### 5.5.1. ハードウェアの構成

《テクミュ》の運用にあたって、VPS（Virtual Private Server）を用いた。VPSの性能は「仮想CPU2コア」、「メモリ2GB」、「SSD30GB」となる。図14にハードウェア構成図を示す。

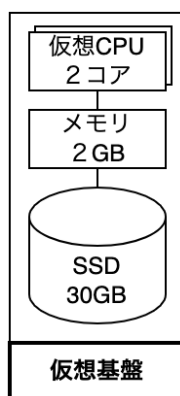


図14 《テクミュ》のハードウェア構成図

### 5.5.2. Webアプリケーションの具体

児童は、ICT端末からWebブラウザによって《テクミュ》にアクセスする。アクセス時の《テクミュ》の画面を図15に示す。

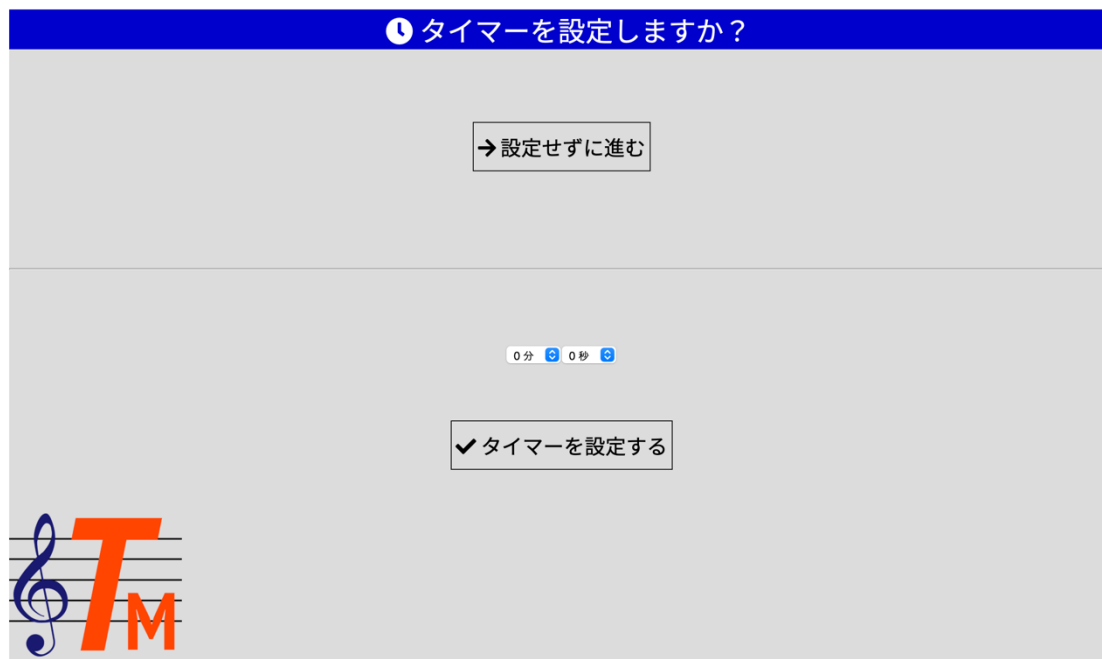


図15 《テクミュ》のアクセス画面

《テクミュ》を運用する OS や Web サーバー，用いたライブラリ，モジュールについては 図 16 のソフトウェア構成図に示した。

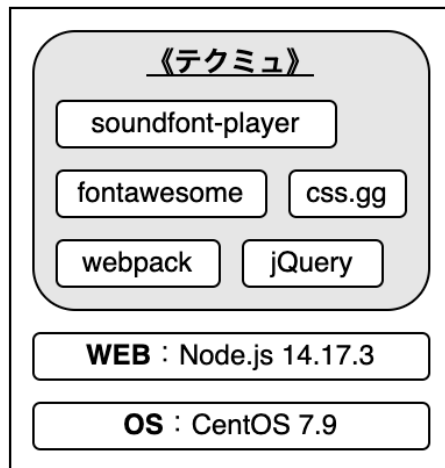


図 16 《テクミュ》のソフトウェア構成図

また，5.4. に示した方針を実現させるための手立ては次の通りである。

まず，演奏者の命令が演奏結果に即座に反映されるよう，JavaScript のタイマー機能を活用し，実行時に発音処理を行う発音タイマーを，演奏者が設定した値（音データ）をもとに，繰り返し生成し，更新する処理を取り入れた。そして，発音タイマーの実行時に，音データを取得することで，演奏者が変更した値を即座に反映するというリアルタイム性を持たせている。この発音処理の流れを図 17 に示す。

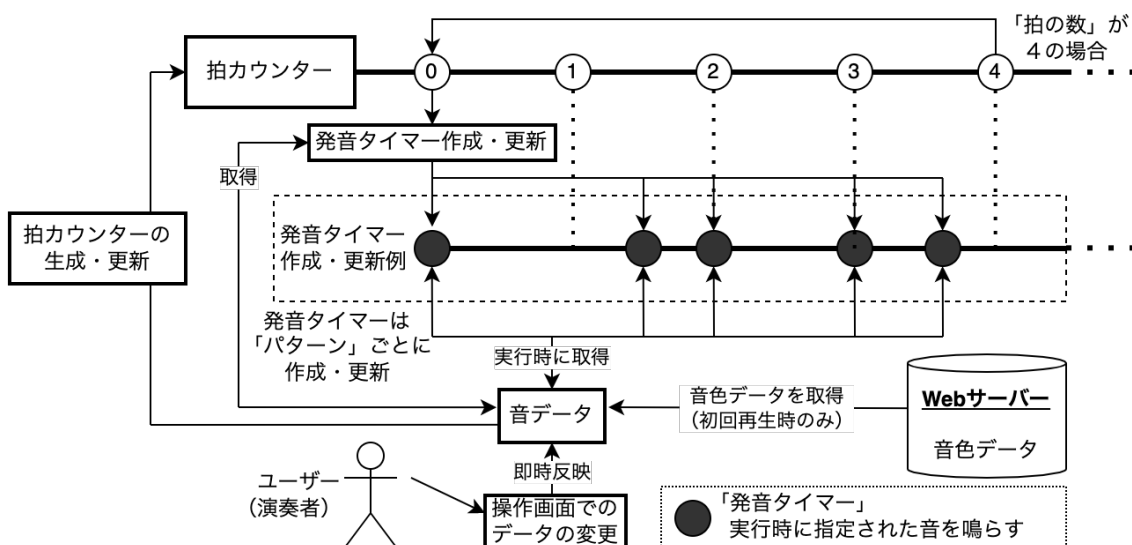


図 17 《テクミュ》における〈ライブ・コーディング〉性をもたせた発音処理の流れ

命令と実行の結果との因果関係を分かりやすく示すために、各画面では、命令の内容を反映したアイコンを示したり、現在演奏している音を強調（ハイライト表示）したりする（図 18）。

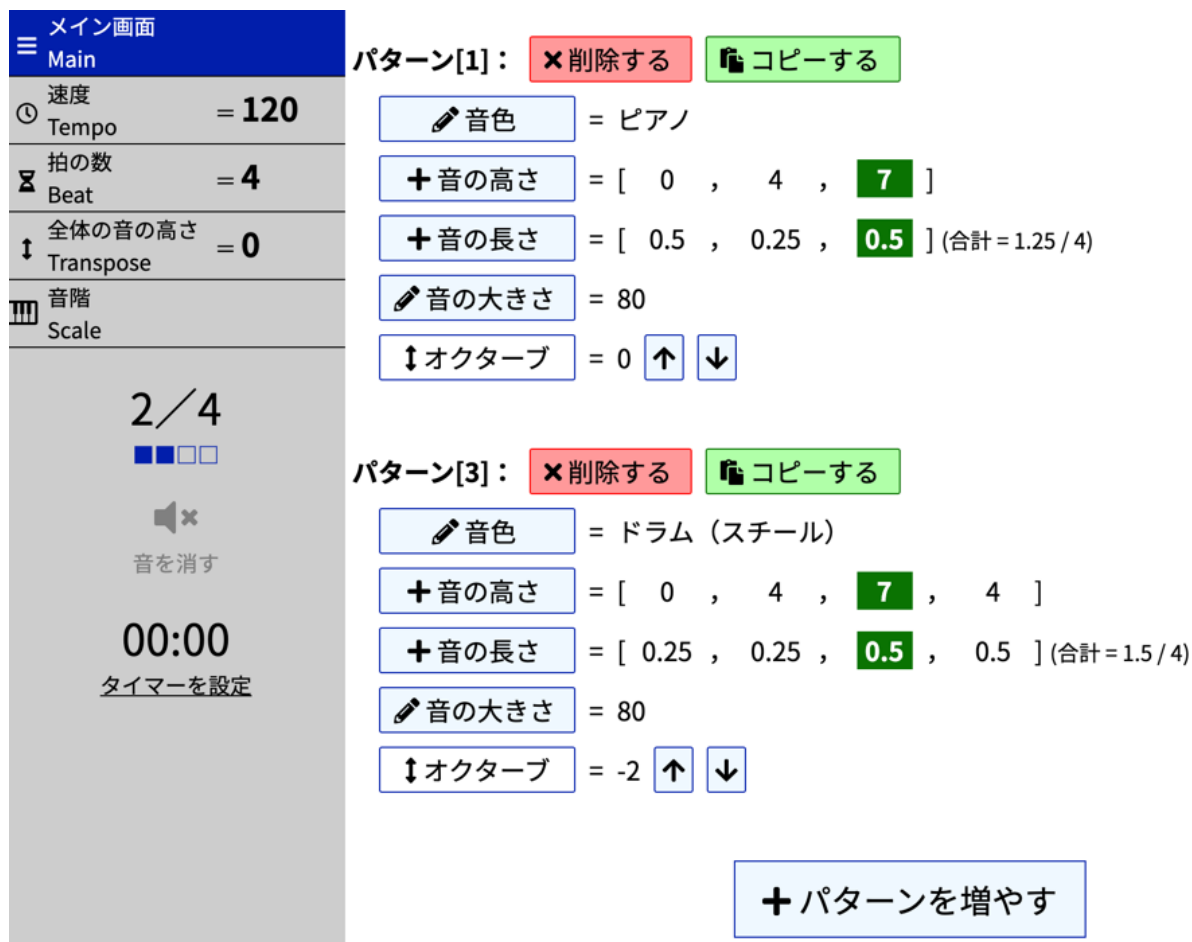


図 18 《テクミュ》のメイン画面

スピード感をもって命令を出せるよう、JavaScript を用いて動的に画面描画を行う SPA (Single Page Application) とし、各画面の遷移時に待ち時間が生じないようにした。画面構成・遷移図を図 19 に示す。

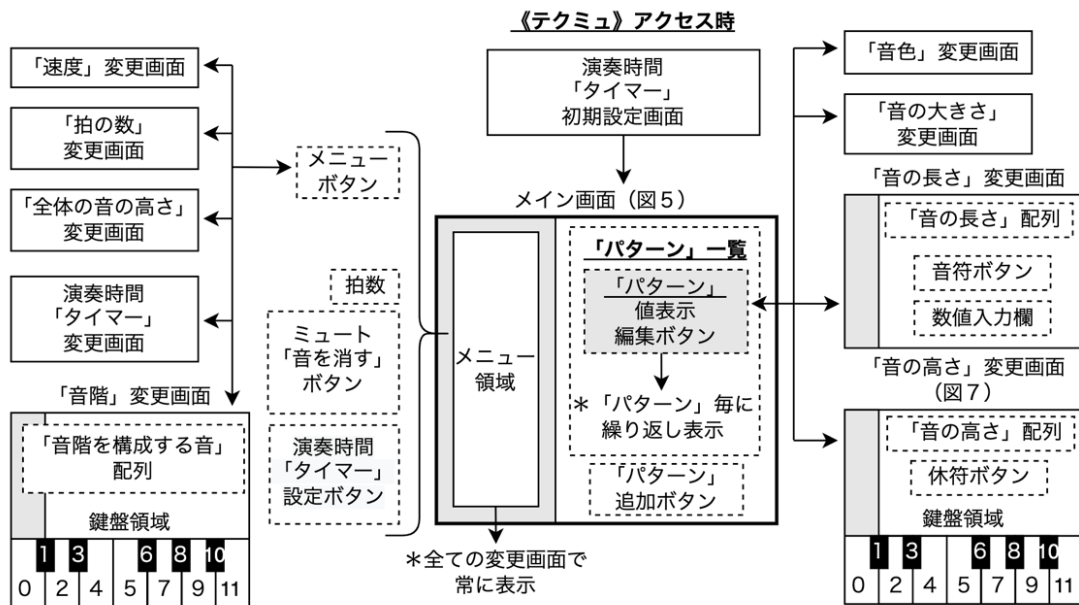


図 19 《テクミュ》の画面構成・遷移図

さらに、直観的な操作性を実現するために、鍵盤画面を JavaScript と CSS で描画し、鍵盤部分をクリックすることで、対応する数値（音の高さ）を入力できるようにした（図 20）。

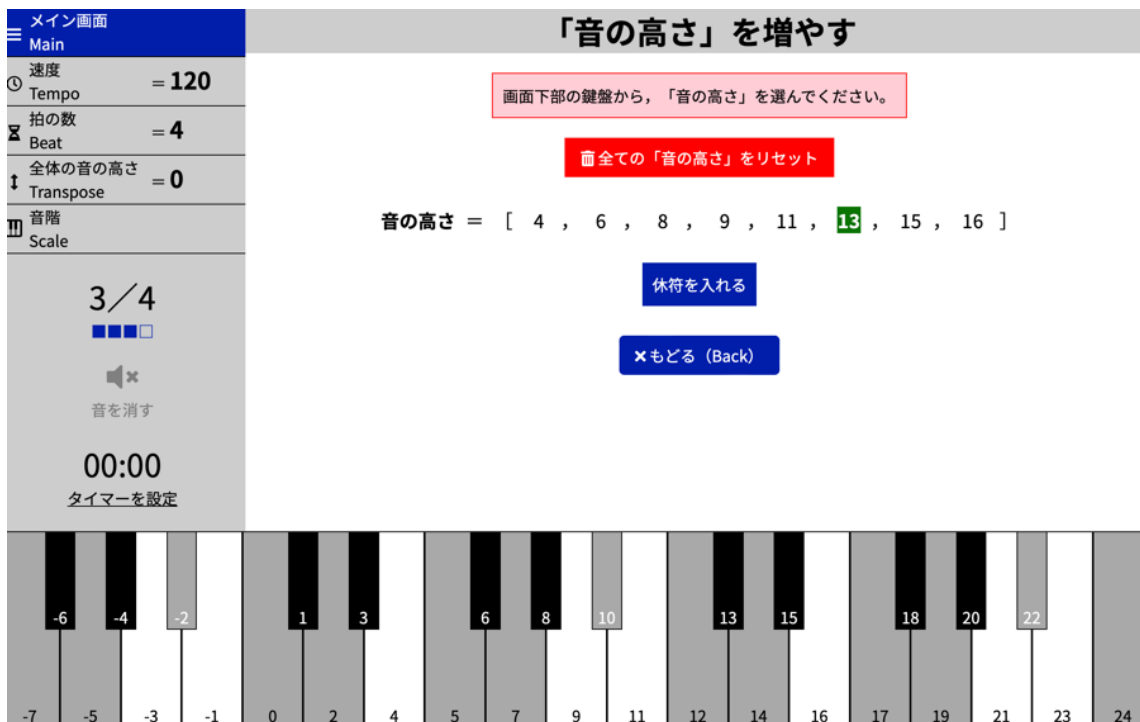


図 20 《テクミュ》の「音の高さ」変更画面

### 5.5.3. 《テクミュ》の概要

《テクミュ》は1拍～16拍から成るパターンの組合せと反復を主とした音楽を生成する<sup>59</sup>。

パターンを組み合わせた反復させたりして音楽をつくる活動は、小学校音楽科の教科書で様々なリズム・パターンを組み合わせて音楽づくりを行う活動といった場面で広く紹介されており、「手引」（文部科学省 2020a, pp. 42-43）においても「様々なリズム・パターンを組み合わせて音楽をつくることをプログラミングを通して学習する場面」として紹介されている。このような、あるパターンを組み合わせ、実際に音を確認しながら、再び組合せを考える活動は、結果を即時に確認できるコンピュータの特徴を活かした活動といえる。

これらの活動と《テクミュ》を用いた音楽づくり活動は、いくつかのリズム・パターンをつくり、それらを組み合わせるという点では同じであるが、その創作過程には大きな違いがある。具体的には、教科書や「手引」の事例では、「音楽をつくる（修正する）」ことと「演奏して検討する」ことを交互に行う試行錯誤が主な過程となることに対し、《テクミュ》においては、反復されるパターンを常に鳴らしている状態で、即興的に次々と命令を編集し、音楽を変化させていく点で異なる。

### 5.5.4. 具体的な操作

《テクミュ》では、生成される音楽全体に関わる命令として「速度」、「拍の数」、「全体の音の高さ」、「音階」が設けられ、それぞれを操作することができる。各パターンは、「音の高さ」、「音の長さ」、「音色」、「音の大きさ」、「オクターブ」の組合せから成る。これらの命令を作成・編集するために、児童は、画面上に表示されたボタンをクリックしたり、スライダーをドラッグして値を編集したりすることによって、「音楽を特徴付けている要素」に対応した様々な命令を組み合わせる（図 21, 図 22）。

---

<sup>59</sup> このような様式の音楽は〈ミニマル・ミュージック (Minimal Music)〉や〈トランス・ミュージック (Trance Music)〉と呼ばれるジャンルの音楽と、高い親和性を持つ。



## 「音の長さ」を増やす

🎵 全ての「音の長さ」をリセット

音の長さ = [ ]



図 21 《テクミュ》の「音の長さ」選択画面

## 「音の大きさ」を変える



図 22 《テクミュ》の「音の大きさ」選択画面

なお、命令の構築の過程においては、命令を取り消す（元に戻す）ことはできない。これは、誤操作等によって不本意な結果になったとしても、その操作を取り消してやり直すのではなく、そこからどのように音楽を展開するかといった判断力を促すことを目的とした、意図的なデザインである。こうした設定には、〈ライブ・コーディング〉の特質である即興性・一回性が反映されている。

### 5.6. 開発において達成された方針と工夫

本章では、5.4. にて示した7つの「開発の方針と工夫」の観点に考慮して《テクミュ》の開発を試みた。

そのうち、開発において「① 〈ライブ・コーディング〉の過程に着目した観点」、「⑤小

学校音楽科の学習内容を考慮した観点」, 「⑥コンピュータの特質を生かした表現に関する観点」, 「⑦動作環境に関する観点」を達成することができた。

他方, 「②『音楽を特徴付けている要素』の理解を深めることに関する観点」, 「③直観的な操作性についての観点」, 「④児童のつくる音楽の多様性の保障に関わる観点」の3つについては, 実際の授業実践を行わなければ検証することができない。次章では, これら3つの点について, 筆者自ら授業者となって小学校での実践を行い, 検証を試みる。

\*本章は, 筆者が2022年に発表した次の論文を, 加筆・修正した内容になる。

長山弘 (2022) 「〈ライブ・コーディング〉の特質を取り入れた音楽づくり活動のための Web アプリケーションの開発」『日本教育工学会論文誌』46巻3号, pp. 567-578.

## 6. 学習材《テクミュ (Tec-Mu)》を用いた授業実践と結果の検証

本章では、学習材《テクミュ (Tec-Mu)》による授業実践を行い、5.4. で挙げた「開発の方針と工夫」の内、授業実践を通して検証が必要とした「②『音楽を特徴付けている要素』の理解を深めることに関する観点」、「③直観的な操作性についての観点」、「④児童のつくる音楽の多様性の保障に関わる観点」について検討し、《テクミュ》がもつ意義を検討する。

はじめに、授業実践の具体を示す (6.1.)。次に、5.6. で授業実践を通して検討すべきとした開発の方針 (3つの観点) を達成することができたかと、プログラミング教育のねらいを達成する学習材を開発することができたかを検討する (6.2.)。それらを踏まえて、本実践で見出された成果と課題を整理する (6.3.)。

### 6.1. 授業実践の概要

本節では、《テクミュ》を用いた授業実践の概要を示す。

#### 6.1.1. 授業実践の対象と日時

対象 A 大学附属 B 小学校 6 年 2 組 (31 名) (内 1 名欠席)

日時 2021 年 7 月 16 日 (金) 3～4 時間目 (10 : 45～12 : 25)

(途中に 10 分休憩を含む 100 分)

#### 6.1.2. 児童の実態

本授業実践の前に、児童が日頃どの程度 ICT 端末やプログラミングに親しんでいるか、また、即興表現への自信についての質問紙調査を行った。結果を表 15～表 17 に示す。

表 15 児童の ICT 端末に対する親しみ

家でパソコンやタブレット (iPad など) をどのくらい使いますか。		
ほとんど毎日	17 名	56.6%
週 4～5 日ぐらい	2 名	6.6%
週 2～3 日ぐらい	7 名	23.3%
週 1～2 日ぐらい	2 名	6.6%
使わない・家がない	2 名	6.6%

表 16 児童のプログラミングに対する親しみ (複数回答可)

「プログラミング」をしたことはありますか。		
小学校でプログラミングをしたことがある (授業・クラブ活動など)	2 名	6.6%
プログラミング教室で習っている, または, 習っていた	4 名	13.3%
自分で調べて, プログラミングをしている, または, したことがある	10 名	33.3%
プログラミングをしたことはない	15 名	50.0%

表 17 児童の即興表現に対する自信

即興表現（音楽やダンスを、その場でつくって 見せること）について自信はありますか。		
4（自信がある）	5名	16.6%
3	2名	6.6%
2	7名	23.3%
1（自信がない）	16名	53.3%

### 6.1.3. 授業のねらい

本授業では、個別活動が始めるにあたって、授業者が《テクミュ》を用いた〈ライブ・コーディング〉を行った後に、「プログラミングで『音楽』をつくろう」というねらいを示した。児童は《テクミュ》によるプログラミングを体験しながら、音楽づくり活動を行う。

### 6.1.4. 授業実践の環境

授業実践時点において、B 小学校では 1 人 1 台のタブレット PC が整備されていなかった。そこで、個別活動においては、児童が 1 人 1 台ずつ使用できるデスクトップ PC<sup>60</sup>が設置されているコンピュータ・ルームで活動し、班活動においては、班ごとに 1 台のタブレット PC<sup>61</sup>を用意し多目的室で活動する。加えて、本実践では、各児童に 1 つずつイヤフォンを用意し、班活動においてはスプリッター（分配器）（図 23）を用いて、1 つのタブレット PC の音声を班全員で共有できるようにする。



図 23 授業で用いたスプリッター（分配器）

発表時には、演奏者用にノート PC を用意し、その画面を複製表示できる電子黒板を用意

<sup>60</sup> コンピュータ・ルームのデスクトップ PC は Windows10 を搭載し、有線 LAN と Google Chrome によって《テクミュ》にアクセスする。

<sup>61</sup> 班活動で用いるタブレット PC には iPad を用意し、無線 LAN と Google Chrome を用いて《テクミュ》にアクセスする。

する。それらを整理したものが図 24 になる。

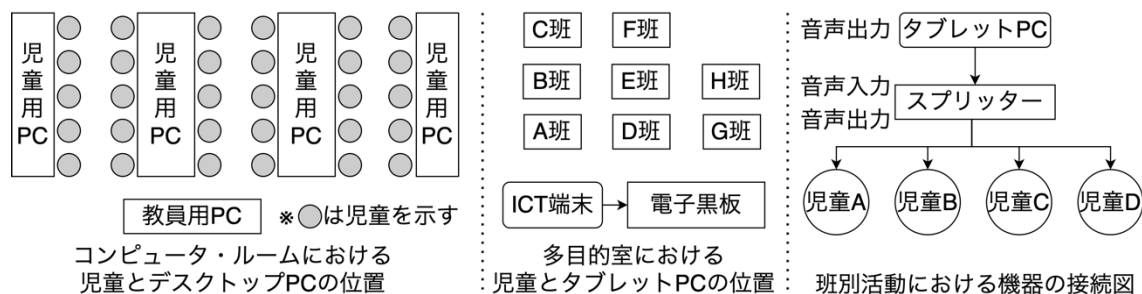


図 24 児童とコンピュータ等との位置

### 6.1.5. 授業実践の流れと児童の様子

本実践は、①1人1台のコンピュータを用いた個別活動による音楽づくり活動と、②3～4名から成る班活動による音楽づくり活動、そして、③〈ライブ・コーディング〉の形を採った発表の3段階に分けて行った。その内容を表 18 に示す。

表 18 授業の流れ

①【個別活動】児童1人1台のデスクトップPCを用いた音楽づくり活動	
(1) 《テクミュ》を理解する (35分)	児童それぞれが、《テクミュ》の〈ティンカリング〉を行うことを通して、《テクミュ》のもつ命令や、その組合せによってどのような音楽をつくることができるかを理解する。
②【班活動】3～4名の班ごとによる1台のタブレットPCを用いた音楽づくり活動	
(2) 協働的な活動を通して、様々な発想を得る (25分)	児童を3～4名の班に分け、班ごとに1台のタブレットPCを用いて、音楽づくり活動を行う。児童は他者の音楽づくり活動を観たり聴いたりすることで様々な発想を得て、自身の音楽づくり活動に活かす。
③【発表】抽出児童及び班による演奏 (音楽づくり活動)	
(3) 〈ライブ・コーディング〉を体験する (20分)	希望する児童や班をライブ・コーダーとして抽出し、他の児童を聴衆役と見立て、〈ライブ・コーディング〉を行う。活動にあたっては、ライブ・コーダーのノートPCの画面を電子黒板に複製することで、プログラムが作られる過程と、それにより生み出される音楽とを共有する。
(4) 振り返りを行う (10分)	ワークシートを用いながら、振り返りを行う。

児童の活動の様子を机間指導で見取ったところ、個別活動では、全員が《テクミュ》にある様々な機能をいじくりまわして(〈ティンカリング〉)、音楽づくり活動に取り組んでいる様子がみられた。班活動においても、「1人1分の時間をもってリレー形式で演奏する」という活動を提示したところ、それぞれの班員が互いの音楽を聴き合いながら、様々な命令を出し合い、音楽を変化させながら取り組んでいる様子を見ることができた。

《テクミュ》を用いた音楽づくりは即興表現としての性質をもつものであるが、事前の質問紙調査(表 17)で即興表現に「1(自信がない)」、「2」と回答した23名の児童も活動できていた。加えて、授業後に実施した質問紙調査より、本活動の楽しさについては、全

での児童から「4（楽しかった）」、「3」という肯定的な反応が得られた（表 19）。

表 19 児童による《テクミュ》を用いた活動の楽しさに対する評価

《テクミュ》のプレイは楽しかったですか。		
4（楽しかった）	23名	76.6%
3	7名	23.3%
2	0名	0.0%
1（楽しくなかった）	0名	0.0%

## 6.2. 授業実践の結果を踏まえた《テクミュ》の検証

本節では、5.4. で示した「開発の方針と工夫」のうち、授業実践を通して検証すべき 3 つの観点（②「音楽を特徴付けている要素」の理解を深めることに関する観点、③直観的な操作性についての観点、④児童のつくる音楽の多様性の保障に関わる観点）と、開発時にめざしたプログラミング教育（5.1.）について、それぞれ達成することができたかを検証する。

### 6.2.1. 「音楽を特徴付けている要素」の理解を深めることに関する観点の検証

観点②の検証を、発表時の児童の姿やワークシートの回答、聴衆役としての児童の姿を手がかりに進める。

#### 6.2.1.1. 「音楽を特徴付けている要素」の働きを意識した活動の様子

発表時の児童の姿として、演奏者役の児童は、新しく取り入れた「音色」に対して聴衆役の児童が関心をもった様子を見ると、「音の大きさ」の値を大きくしてその「音色」を強調した。個別活動や班活動において認識した「強弱」の命令を即興的かつ効果的に用いている様子は、「音楽を特徴付けている要素」とその働きを、リアルタイムな活動を通して学んでいる姿といえる。

また、活動の途中で「低い音での音楽をつくった」、「音色を重視した」、「減らして、こちよいい音にした」、「パターンをふやした」、「最初はメロディーをつくり、あとからわき役をつくった」といった工夫を行ったことが質問紙調査から読み取れた。この結果からは、児童が、実行の結果（音楽）と命令との因果関係—すなわち「音楽を特徴付けている要素」の働きを意識し、意図的に命令を発していることがわかる。

#### 6.2.1.2. 聴衆役としての児童の姿

4.4. の「〈ライブ・コーディング〉におけるプログラム（アルゴリズム）と音楽との関係」で示したように、〈ライブ・コーディング〉の特質として、演奏者の命令（プログラム）が聴衆に明示されること、そして、結果（音楽）や聴衆の反応を踏まえて、演奏者は音楽を変化させることが挙げられる。本実践においても、他者の演奏を聴く（観る）活動において、児童たちは「いま、演奏者がどのような命令を編集しているのか」と、「その

命令が音楽をどのように変化させているのか」を感じ取りながら聴くことができていた。

例えば、演奏中の児童に対して、聴衆役の児童が『合唱の声』に変えたらとても面白い」と反応していたり、ときには「音を小さくした方が良い」と命令について助言したりすることもあった<sup>62</sup>。加えて、それらの反応を受けた演奏者が命令を変更し、音楽が変化した際には、聴衆役の児童に「このように変化するのか」といったさらなる反応がみられた。

これらの児童の姿は「〈ライブ・コーディング〉におけるプログラム（アルゴリズム）と音楽との関係」の過程そのものである。さらに、4.6.において〈ライブ・コーディング〉の教育的意義として挙げた、音楽をつくる過程がスクリーンに表示されることで、聴衆役の児童の理解も促される点を実現できたことが指摘できる。

以上の結果より、《テクミュ》が「開発の方針と工夫」の観点②「音楽を特徴付けている要素」の理解を深める学習材となっていると考えられる。

## 6.2.2. 直観的な操作性についての観点の検証

観点③の検証を、《テクミュ》の操作性と、〈ティンカリング〉の様子を手がかりに進める。

### 6.2.2.1. 《テクミュ》の操作性

授業後、《テクミュ》の操作性についての児童の実感を質問紙で調査した。その結果を表 20 に示す。

表 20 児童による《テクミュ》の操作性に対する評価

《テクミュ》の操作はかんたんでしたか。		
4 (かんたんだった)	14名	46.6%
3	10名	33.3%
2	3名	10.0%
1 (むずかしかった)	3名	10.0%

表 20 より、「4 (かんたんだった)」、「3」と回答した児童が 24 名であったように、ほとんどの児童から肯定的な回答を得ることができた一方で、「1 (むずかしかった)」、「2」と回答した児童も 6 名みられた。このことは、本学習材改善にあたっての課題となった。

しかしながら、授業中の机間指導の様子からは、日頃コンピュータに親しんでいないと回答した児童（表 15）、プログラミングの経験がないと回答した児童（表 16）、《テクミュ》の操作が難しいと回答した児童（表 20）も含めた全員が、様々な命令を組み合わせながら音楽づくり活動に取り組む姿がみられた。具体的には、個別活動においては自分ひとりで、ときには周りの児童の端末の画面を参考に、様々な機能を〈ティンカリング〉しな

<sup>62</sup> これら聴衆役の児童の行為は自発的なもので、授業者が児童に促したものではない。

がら音楽づくり活動に取り組むことが、また、班活動においても、複数人でひとつの音楽を様々に変化させることができていた。

#### 6.2.2.2. 〈ティンカリング〉の様子

活動中、児童は、他の児童にも影響されながら、様々な操作を試行していた。また、速度や拍の値についても、極端に大きな（あるいは小さな）値を試している姿もみられた。そうした様子からは、結果が明確に予想できる命令だけでなく、実験的な命令をも積極的に試行していたことが推察される。さらに、実験的な命令を試行していた児童も、やがて、命令を限定し〈ティンカリング〉を進めていく過程へと変容していった。この姿からは、予め目標となる音楽は曖昧であったものの、次第に、児童の思いや意図が反映された音楽へと移り変わっていったことが指摘できる。活動全体を通して、どの児童も、ほとんど手を止めることなく、様々な操作を試行していた様子は、《テクミュ》が〈ティンカリング〉を促す特質を備えていることの証左である。

以上のことから、「開発の方針と工夫」の観点③直観的な操作性については、操作に困難さを感じた児童が6名みられたという課題が残った（表20）ものの、〈ティンカリング〉を促す直観性をもたせることは、概ね達成できていたといえる。

#### 6.2.3. 児童のつくる音楽の多様性の保障に関わる観点の検証

観点④の検証を、児童が授業を通して生成した音楽を手がかりに進める。

5.4. の「④児童のつくる音楽の多様性の保障に関わる観点」で述べたように、音楽づくり活動においては、自由と制約のバランスを適切に設けることが肝要である。これらを踏まえて、本実践における児童の様子や音楽の特質を検討したところ、次の2点が指摘できた。

1 点目は、筆者の設けた音楽的枠組としての「パターンの組合せと反復を主とした音楽」を保つことができたことである。児童が、数多くのパターンを重ね合わせたり、極端な値の命令を入力したりするといった大胆な試行をした場合でも、生み出される音楽は、筆者の設けた枠組から逸脱することはなかった。

2 点目は、児童のつくった音楽が、筆者の設けた枠組におさまったものであったにも関わらず、似通ったものにならず、それぞれの興味や好みが反映された個性的なものとなったことである。例えば、あるパターンを生成し、その音高や強弱に関わる値、そして音色を編集することを繰り返してブラッシュアップしていく音楽を発表した児童もいれば、数多くのパターンを重ね、様々な要素に関わる命令を矢継ぎ早に繰り返すことによって激しい変化が特徴となる音楽を表現した児童もみられた。ほかにも、「オクターブ」の値を調整することで、それぞれのパターンが別々の音域で演奏される音楽や、演奏時間に伴って、「全体の音の高さ」や「速度」を変化させることにより展開を意識した音楽もみられた。また、児童が自らの興味や好みを反映させることができていたことは、質問紙調査で



「色々な音楽」、「自由な音楽」、「自分の音楽」がくれたことが楽しかったと回答した児童が16名みられたことにもあらわれている。

これらより、《テクミュ》が「開発の方針と工夫」の観点④児童のつくる音楽が多様なものであることを達成できたこと、そして、《テクミュ》に設けられた制約も、音楽の多様性を保障し得る適度なものであったことがいえる。

#### 6.2.4. 開発時にめざしたプログラミング教育の検証

本項では、《テクミュ》が開発時にめざしたプログラミング教育（5.1.）を達成できたか確認する。

まず、児童の〈プログラミング的思考〉を育むことについて、6.2.1.1.において、児童が実行の結果（音楽）と命令との因果関係を意識し、意図的に命令を発していたとしたことや、6.2.2.2. でどの児童も〈ティンカリング〉に取り組むことができたことより、達成することができたといえる。次に、コンピュータの特徴への気付きを促すことについて、6.2.3.において児童がつくった音楽は、いずれもコンピュータなしで（児童自身の生演奏によって）実現することは困難なものばかりであった。このことは、コンピュータを用いることで音楽表現をより多様に行えることの実感（プログラミング教育のねらい②）につながるものであり、達成することができたといえる。そして、「音楽を特徴付けている要素」の働きの理解を深めることについて、6.2.1.において、観点②「音楽を特徴付けている要素」の理解を深めることができたとしたことから、達成できたといえる。

以上のことから、《テクミュ》が「開発の方針と工夫」の各観点と、開発時にめざしたプログラミング教育を達成できる学習材になったことがいえる。

#### 6.3. 再検証すべき「開発の方針と工夫」の観点

本研究では、小学校音楽科の音楽づくり活動においてプログラミング教育を実施するために、プログラミング教育のねらいの①に関連して、小学校音楽科の音楽づくり活動で働く思考にもとづいた〈プログラミング的思考〉を育むこと、ねらいの②に関連して、コンピュータの特徴への気付きを促すことと、ねらいの③に関連して、音楽科の内容の理解を深めることに焦点を当てて、7つの観点から成る「開発の方針と工夫」（5.4.）を立てた。それらを踏まえ学習材《テクミュ》の開発と、それを用いた授業実践に取り組み検証したところ、《テクミュ》が本研究でめざすプログラミング教育（5.1.）にアプローチし得る学習材であることを指摘できた。しかしながら、「開発の方針と工夫」の観点については、いずれも達成できたものの、再検証すべき点が2つ残された。

1つは、「③直観的な操作性についての観点」である。授業後の質問紙調査（表20）では、《テクミュ》の操作性について「1（むずかしかった）」「2」と回答する児童が30名中6名（20%）みられた。児童の〈ティンカリング〉を促すためにも、全ての児童にとって分かりやすい学習材となる改良を行い、直観性をもたせられたかという検証が必要になる。

もう1つは、「②『音楽を特徴付けている要素』の理解を深めることに関する観点」である。6.2.1. より、発表活動において、発表児童に対し聴衆役の児童らが音色や音量について助言をし、それを受けて発表者は命令を修正し、変化した音楽を聴きながら児童らは命令と実行の結果との因果関係を体験していた。質問紙調査からは、音楽を特徴付けている要素を工夫しながら活動に取り組んでいる児童の姿がみられた。以上のことから、観点②を達成することができたといえる。しかしながら、「音楽を特徴付けている要素」の理解を深めることは、《テクミュ》がもたらす学びのなかで、最も期待されていることの1つとなる。そのため、そのような学習効果を一層高めるための改良に取り組み、その再検証を行うことは、《テクミュ》の教育的可能性をより確実にするために重要であると考えられる。

これらを踏まえ、観点②と観点③に焦点をあて、次の改良に取り組む。1つは、命令が即時に結果へと反映されるように発音処理を改良することである。それにより、〈ライブ・コーディング〉性が高められ、「音楽を特徴付けている要素」の理解を深めることにつながることを期待される。もう1つは、画面デザインや画面遷移を見直すことで、誰にとってもそれぞれの命令が意味することを一見して理解できる直観性をもたせることである。

次章では、上の再検証すべき観点を踏まえて改良に取り組み、《テクミュ》の新たなバージョンとなる「二版」を開発する。そして、「二版」を用いた授業実践を通して、プログラミング教育との関わりについての考察へとつなげていく。

\*本章は、筆者が2022年に発表した論文と、2023年に発表予定の論文を、加筆・修正した内容になる。  
長山弘(2022)「〈ライブ・コーディング〉の特質を取り入れた音楽づくり活動のための Web アプリケーションの開発」『日本教育工学会論文誌』46巻3号, pp. 567-578.

長山弘(2023)「〈ライブ・コーディング〉の特質を取り入れた音楽づくり活動のための Web アプリケーションの開発 (2) 一命令と実行の結果との即時性及び画面設計に焦点を当てて一」『初等教育カリキュラム学会誌』11号, 採択決定済み.

## 7. 《テクミュ》の改良と実践—命令と実行の結果との即時性及び操作性に焦点を当てて—

5., 6. では、プログラミング教育のねらいを踏まえ、〈ティンカリング〉をパフォーマンス化した〈ライブ・コーディング〉に焦点をあて、7つの観点から成る「開発の方針と工夫」に沿って学習材《テクミュ》を開発した。そして、《テクミュ》を用いた授業実践を行い、各観点とプログラミング教育のねらいが達成できているかを検証した。検証の結果、再度検証すべき観点として、「②『音楽を特徴付けている要素』の理解を深めることに関する観点」と「③直観的な操作性についての観点」の2つが指摘された。

本章では、それらの結果を踏まえ、《テクミュ》の改良を行う。改良後、あらためて授業実践を行い、前述の2つの課題を再度検討する。さらに、《テクミュ》の、学習材としての特質を明らかにするために、授業後の児童の質問紙調査への記述の分析を通して、児童が活動中にどのような思考を働かせたのかを検討する。

はじめに、6.3. で挙げた再検討すべき観点を踏まえ、学習材の修正の手立てを検討し、実際に改良に取り組む(7.1.)。次に、《テクミュ》を用いた授業実践を行い(7.2.)、その結果を踏まえ、《テクミュ》とプログラミング教育との関わりを考察する(7.3., 7.4.)。

### 7.1. 改良にあたっての手立てと具体

本節では、6.3. で示した2つの観点に対する改良策を検討し、実際に《テクミュ》の改良に取り組む。また《テクミュ》は、初版と同様にVPSサーバー上に設置し「<http://tec-mu.com>」に公開している<sup>63</sup>。アクセスしたときの画面を図25に示す。



図25 「二版」のアクセス画面

<sup>63</sup> 本研究の授業実践で用いた「二版」は「<http://v2.tec-mu.com>」でも公開している。

### 7.1.1. 「音楽を特徴付けている要素」の理解を深めるための改良

まず、観点②にあたって「音楽を特徴付けている要素」の理解を深めるための改良策を検討する。「音楽を特徴付けている要素」の理解を促すためには、児童が自らの出す命令とその実行の結果（つまり、音楽）との因果関係を理解しやすく設計することが肝要となる。つまり、児童が出した命令が、できる限り即時に音楽に反映される仕組みを設ける必要がある。

ここで演奏される音楽の即時性に関わって、《テクミュ》の発音処理の流れを整理する。《テクミュ》では、設定された拍の数（例えば、4や8）を繰り返しながら、音楽が演奏される。

「初版」においては、指定された「速度」（＝テンポ）をもとに「0」、「1」、「2」、「3」・・・と現在<sup>64</sup>の拍<sup>65</sup>を計算する「拍カウントタイマー」（T1）と、音の高さや音色に関わる値など（「音データ」）を反映して音を鳴らす「発音タイマー」（T2）を設けた。その結果、演奏者が出した命令は、次に1拍目となったときに音楽に反映されるため、即時性に欠けるところがあった（図26）。

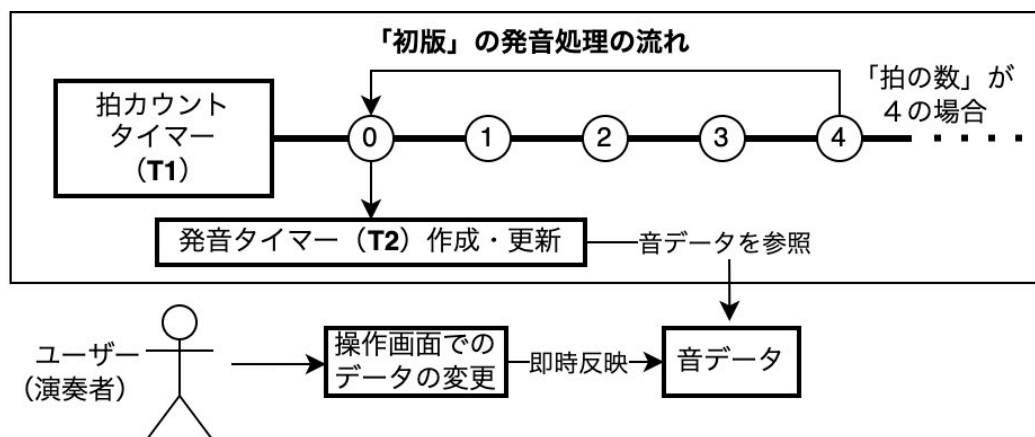


図26 「初版」の発音処理の流れ

よって、「二版」では、「発音する音があるかを確認するタイマー」（T3）によって発音処理を行うようにした。それにより、演奏者は、命令を出すと即時に結果としての音楽を確認することができる（図27）。

<sup>64</sup> ここでの「現在」とは、演奏者が《テクミュ》を操作している「瞬間」を指す。

<sup>65</sup> プログラムの処理では、現在の「拍」の計算は0から開始しているが、画面上ではその値に1加算した数を表示している。つまり、「拍の数」を4と設定した場合、画面上では「1」、「2」、「3」、「4」の値が繰り返し表示される。

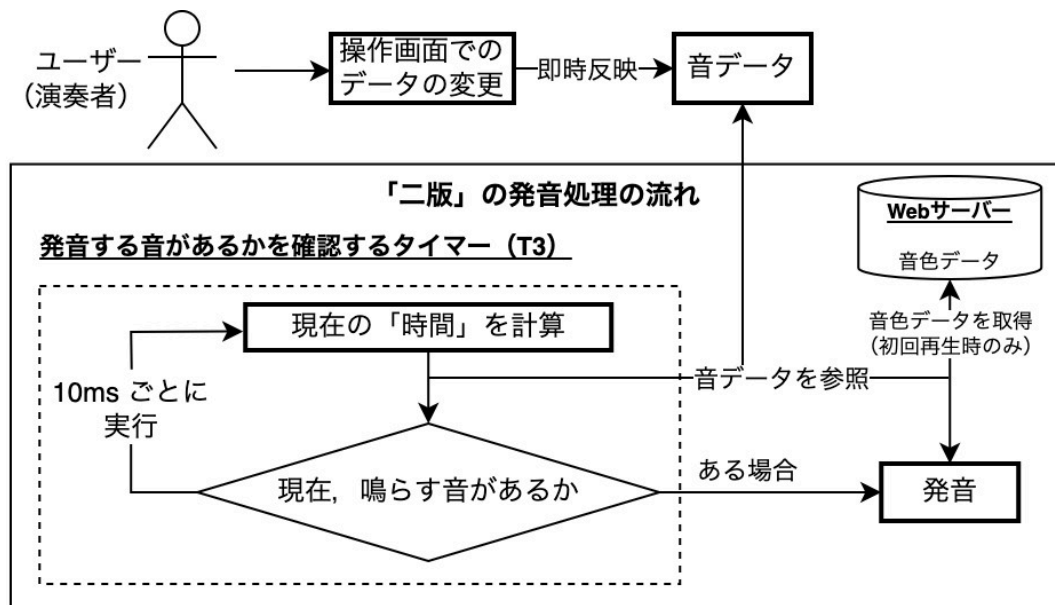


図 27 「二版」の発音処理の流れ

## 7.1.2. 直観的な操作性を高めるための改良

次に、観点③にあたって、直観的な操作性をもたせるための方策を講じる。

### 7.1.2.1. 直観性を高めるための工夫

原田 (2019, pp. 71-75) は「考えることなく見ただけですべての意味を把握でき、かつ操作に手間取ることが何一つない状態」が、利用者にとって精神的・身体的負荷（「インタラクションコスト」）の最も小さい良いデザインだと述べている。続けて、インタラクションコストは、画面の大きさなど、利用者の環境によって異なるため、それぞれに合わせたデザインが必要になるが、いずれにおいても「一貫性」、「シンプルさ」、「共通概念」を併せ持つことが「インタラクションコストを最小化する普遍的手法」だと述べている (p. 75)。

ここで、原田が述べている「一貫性」、「シンプルさ」、「共通概念」の定義を確認する。「一貫性」とは、「デザインの意味や操作をユーザーが予測」しやすいように、ある規則に則って設計をすることである (p. 76)。「シンプルさ」とは、「明快」を指し「何が起きているのかを即座に理解でき、次にどうするか自信を持って決定できる状態」のことである (p. 79)。一方、原田は、手数を減らし即時性を高めるためには「複雑」なデザインも必要であるとし、利用者の習熟度によっては複雑である方が有用であるということも述べている (p. 81)。「共通概念」とは「青文字はリンク」や「星はお気に入りか評価」といった「誰もが知っている標準的なルール」を指す (p. 82)。

これらを踏まえて、原田は、「直感的」とは「ユーザーの予測とその結果が完全に一致」し思った通りに動く状態と定義している (p. 85)。その際、「明快」であることはその操作

が何を意味するのかを理解することとは異なるため、「シンプルさ」と「直感的」とは別の概念であること、また、「一貫性」のあるデザインでも利用者の経験によって直感異なることを指摘している。それらを踏まえた上で、利用者の「それまでの経験に依存」する直感と共通概念に基づきながら、「インタラクションコスト」を小さくした状態が、直感的なデザインだとしている (p. 85)。なお、原田は「直感的」という語を用いているが、本論における「直観的」と同義と考えられる<sup>66</sup>。

以上を踏まえ、「直観的」を生むための工夫を整理すると、全ての利用者にとって一見して理解できるデザインを設計することは、直観が利用者の経験に依ることから困難といえる。しかし、「一貫性」、「シンプルさ」、「共通概念」を工夫することで、多くの人々にとって理解しやすいデザインをねらうことはできる。

### 7.1.2.2. 直観性をもたせるための工夫の具体

「初版」においても、「追加」や「削除」といった命令に相応しいアイコンを併せて表示するなどの工夫を取り入れた (5.4.) が、「二版」では直観性を高めるための工夫 (7.1.2.1.) を踏まえ、児童がより直観的に操作できるような画面設計となるように《テクミュ》を見直した。次にその具体を述べる。

まず、画面遷移の仕組みを見直した。「初版」においては、音の高さ、音の長さ、音の大きさ、音色などの値を設定するために、「メイン画面」から、それぞれの値の編集ボタンを押して編集画面へと遷移していた。そこで「二版」では、「シンプルさ」及び「複雑さ」を踏まえ、同一の画面上で値を編集できるように修正し、利用者が少ない手数で命令を出せることと、現在、何を編集しているのかを分かりやすくすることをねらった。二版における画面構成・遷移図を図 28 に示す。

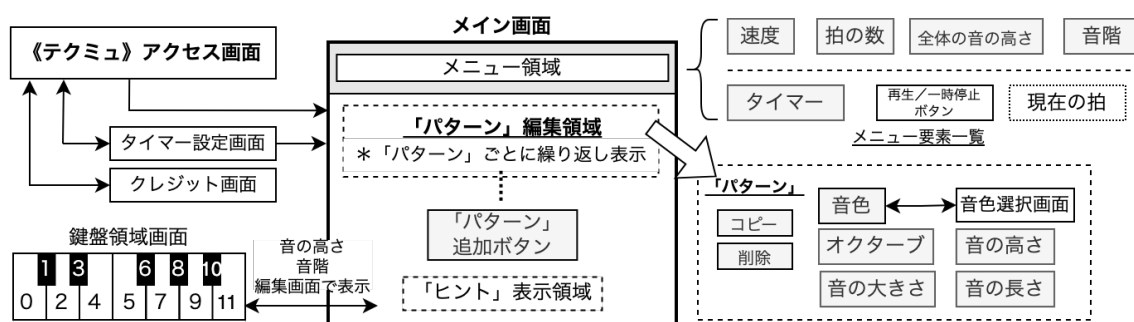


図 28 「二版」における画面構成・遷移図

<sup>66</sup> 『学研 現代新国語辞典』(金田一春彦・金田一秀穂 2017, p. 949) によると、「直観」は「推理や判断などによらずに対象の本質を直接にとらえること。また、とらえた内容」を、「直感」は「[勘などの働きによって] 瞬間的に感じとること」を指す。これらの定義より、それぞれ対象の内容をその場で理解できるという点では同じだが、その根拠は異なることが分かる。そして、原田の「直感的」をみると、その思考は勘ではなく、根拠をもって働いている。このことを、先の定義に照らし合わせると、「直感的」より「直観的」といえる。

その際、値の編集に多く用いられる「スライダー」やカーソルを合わせると背景色が変わる「ボタン」といった「共通概念」を踏まえ、画面上のスライダーを操作したりボタンを押したりすることで値を編集できるデザインとした（図 29）。

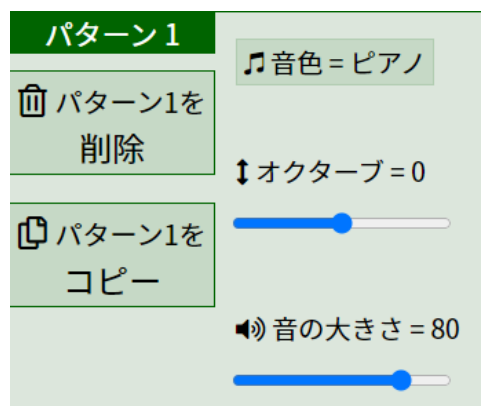


図 29 「共通概念」を取り入れた「二版」の操作画面

次に、音の高さと音の長さの表し方である。「初版」では、それぞれの値は別々の箇所に表示されていたため、どの音の高さが、どの音の長さに対してになっているのかが一見して分かりづらかった。そこで「二版」ではそれらの関係が理解しやすいデザインをねらった。中川（2011, p. 32）は、「複数の要素がある場合、人の目はまず左から右、そして左下へと動くことから、それに沿ったデザインとすることが「理解を深めるレイアウト」には重要だと述べている。筆者では、中川の論と原田の「一貫性」を踏まえ、「二版」では、「音の高さ」、「音の長さ」を行とし、左から右へ列（＝時間軸）をもつ表として示すこととした。加えて、音の長さによって、それぞれの列の横幅を変える工夫も取り入れた（図 30）。

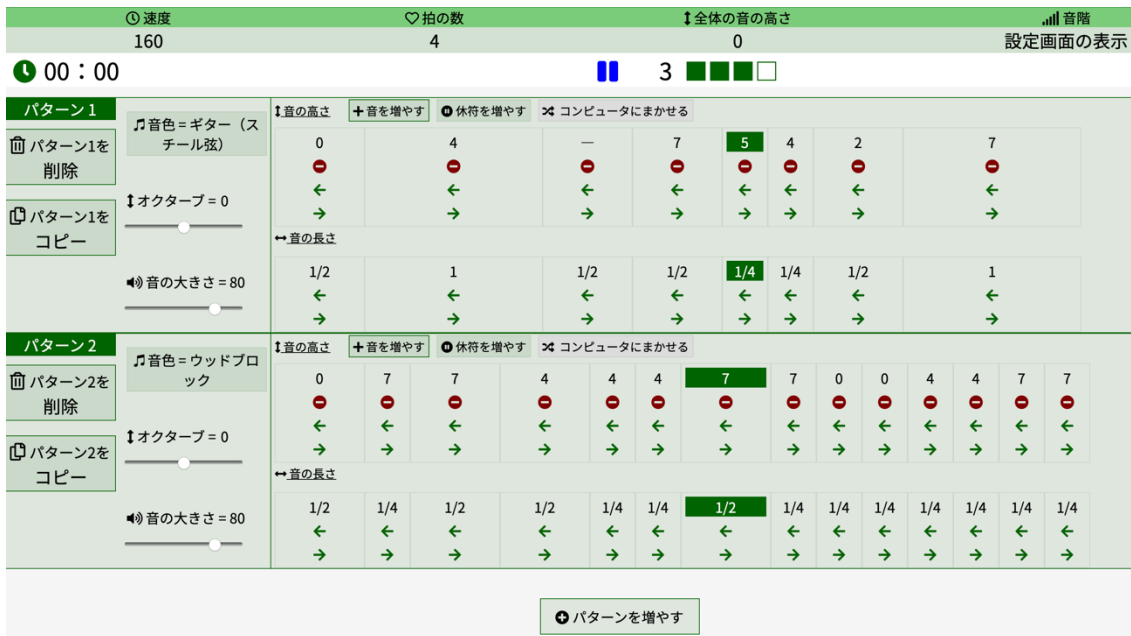


図 30 「一貫性」を取り入れた「二版」の操作画面

### 7.1.2.3. 〈ティンカリング〉を優先した点

プログラミング教育の視点より，児童の〈ティンカリング〉を促すために，あえて修正しない点を設けた。それは，数値で表せる命令（例えば，音の高さ）に関することである。

阿部（2016, p. 350）は，小学校第3学年を対象にScratchを用いたワークショップを行ったが，共通してみられた児童の活動として，変数に「99999...9」といった極端に大きな数値を入力したり，同一のオブジェクトを幾重にも複製したりして，その結果を楽しみ共有したことを報告している。また，西下（2016, p. 1232）も，小学校第1学年を対象にScratchを用いて数値を変えながら画面上の魚を動かす活動を行った際に「やみくもに数値を大きくして目にもとまらぬ速さで魚が動くのを楽しむ子」がいたことを報告している。これらは命令に様々な数値を入力することによって，〈ティンカリング〉が活発に行われている姿といえる。

以上を踏まえ，《テクミュ》においても，幅広い〈ティンカリング〉を促すために，命令の入力や表示に数値を活用する。例えば，音の高さを表す際に，小学校音楽科で学ぶ音名（「ド」，「レ」，「ミ」など）ではなく数値を用いる（図 31）。

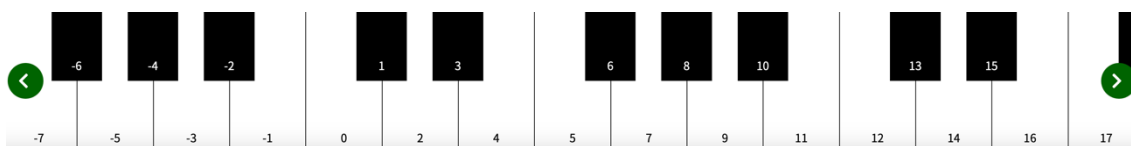


図 31 「二版」における「音の高さ」入力用の鍵盤画面



ただし、「二版」では、音色を「打楽器群」とした場合、「鍵盤」画面に数値と楽器名（「太こ（低）1」、「手拍子」など）を表示した（図 32）。これは、「打楽器群」の音の高低は様々な打楽器と対応しているため、数値の大小と結果とに因果関係がなく、数値を見ても何の楽器を表しているのかが分からないためである。

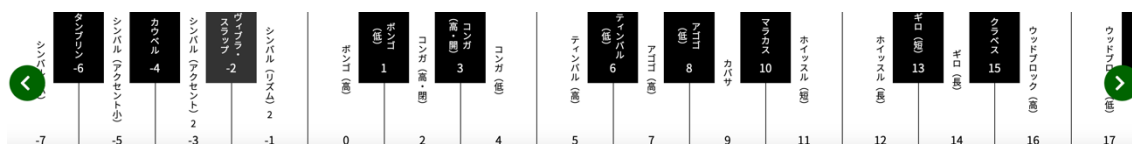


図 32 「二版」で「打楽器群」を選択した場合の「音の高低」の入力画面

### 7.1.3. その他の改良

6.3. で挙げた観点を踏まえた改良に加え、命令を出すための支援となる工夫を取り入れた。その1つが、「音を鳴らすために『⊕パターンをつくる』を選択しましょう」などといった命令の「ヒント」を表示することである（図 33）。

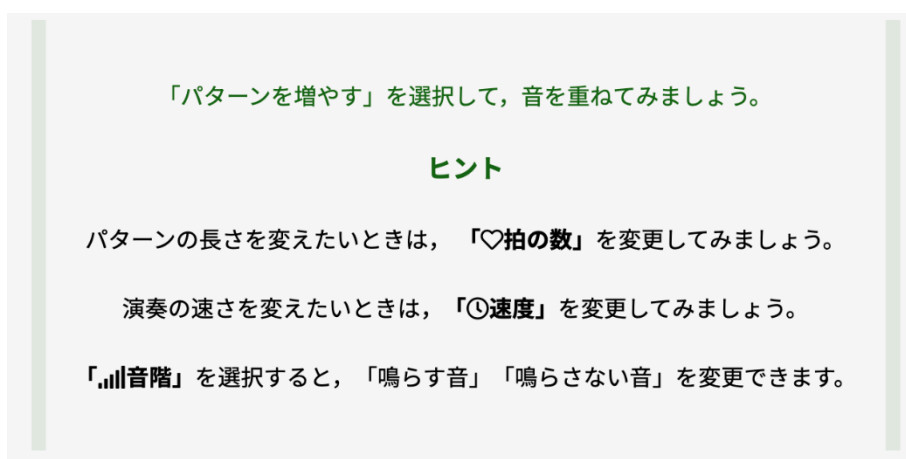


図 33 「二版」で表示する「ヒント」

ほかにも、音の高さと長さをランダムに入力する機能を備えた。それにより、スピード感を伴う〈ティンカリング〉を促すことをねらった。

### 7.1.4. 改良によって生じた変更点

本項では、改良によって生じた変更点を示す。

まず、操作と生成される音楽についてである。「二版」では 1 拍～12 拍から成るパターンの組合せと反復を主として音楽を生成する。演奏者は、音の高さと長さを自ら選ぶか、コンピュータによってランダムに選択することによって旋律を作る。一方、「初版」と同じ箇所は、パターンに関わる命令として、音色、音の大きさ、オクターブが変更できる点、

そして、音楽全体に関わる命令として、速度、拍の数、全体の音の高さ、音階が変更できる点である。

また、「二版」のハードウェア及びソフトウェア構成図<sup>67</sup>を図 34 に示す。

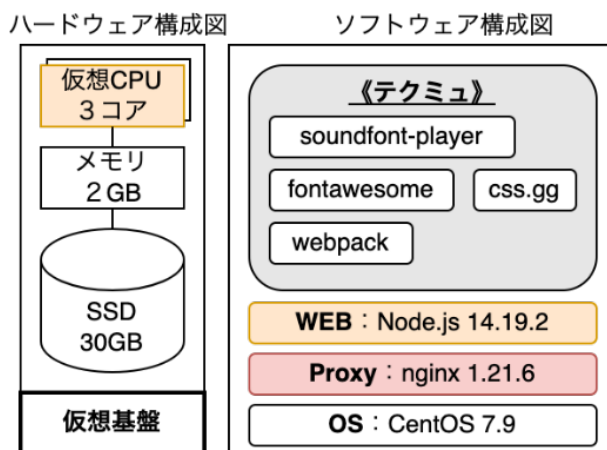


図 34 「二版」のハードウェア及びソフトウェア構成図

次節では「音楽を特徴付けている要素」の理解を深めるための改良や、直観的な操作性を高めるための改良を施した「二版」とプログラミング教育との関わりについて検討するために、「二版」を用いた授業実践に取り組む。

## 7.2. 「二版」による授業実践と結果

本節では、「二版」を用いた授業実践を行い、その結果を児童が音楽をつくった過程の具体などにより示す。児童の活動過程を検討するにあたって、実践における児童の様子をビデオカメラによって記録した。

### 7.2.1. 授業実践の対象と日時

対象 A 大学附属 B 小学校 6 年 1 組 (30 名)

日時 2022 年 7 月 13 日 (水) 4 時間目 (11 : 40 ~ 12 : 25)

対象 A 大学附属 B 小学校 6 年 2 組 (29 名) (実践②は 1 名欠席)

日時 実践① 2022 年 7 月 11 日 (月) 4 時間目 (11 : 40 ~ 12 : 25)

実践② 2022 年 7 月 20 日 (水) 4 時間目 (11 : 40 ~ 12 : 25)

なお、6 年 2 組については、6 年 1 組と同様 1 時間分の授業を予定していたが (実践①),

<sup>67</sup> 開発の過程で、「二版」のハードウェア及びソフトウェア構成図は「初版」とやや異なるものとなった。大きな違いとして、読み込むファイルの軽量化をねらって jQuery ライブラリを使用せずに開発したことで、URL によって行き先を初版、二版とに振り分けるために Proxy サーバーとして nginx を取り入れたことが挙げられる。

授業中に機器トラブルが発生し、活動時間が不足したため、第 2 時となる実践②を設け補った。後述する授業の流れについては、1 組、2 組とも同じである。

### 7.2.2. 活動の環境と児童の ICT 活用の経験

個別活動にあたって、児童に 1 人 1 台ずつ ICT 端末として iPad とイヤフォンを用意し、それぞれの iPad から「二版」にアクセスした。発表活動にあたって、教室の前方に、発表者用の ICT 端末とその端末の画面を映す電子黒板を用意した。また、班活動を行う場合に 1 つの iPad の音を班全員が共有できるようにスプリッター（分配器）を班の数分用意した。また、授業にあたって、ICT 端末を用いた音楽づくり活動についての質問紙調査を児童 59 名（1 組 30 名、2 組 29 名）に行ったところ、表 21 の結果が得られた。

表 21 ICT 端末を用いた音楽づくり活動への親しみ

ICT 端末を用いた『音楽づくり』をどのぐらいしたことがあるか。	
よくしている	1 名 (2%)
たまにする	2 名 (3%)
したことはあるが今は、ほとんどしない	9 名 (15%)
いちども、したことがない	47 名 (80%)

### 7.2.3. 授業実践の流れ

「二版」による授業実践の流れは表 22 の通りである。

表 22 「二版」を用いた授業実践の流れ

【導入】電子黒板を用いた鑑賞	
(1) 〈ライブ・コーディング〉を知る (5 分)	授業者が「二版」による演奏と授業のめあてを提示しする。児童は〈ライブ・コーディング〉の演奏動画を鑑賞する。
【個別活動】1 人 1 台のタブレットを用いた音楽づくり活動	
(2) 「二版」を体験する (15 分～20 分)	児童はそれぞれの iPad より「二版」にアクセスし、個別活動に取り組む。
【発表】発表児童と聴衆役の児童に分かれて「二版」を用いた音楽づくり活動を発表	
(3) 〈ライブ・コーディング〉を体験する (15 分～20 分)	〈ライブ・コーディング〉を実感するために、希望する数名の児童による発表活動を行う。

はじめに、授業者が「二版」を用いた演奏を行う。演奏の様子は電子黒板を用いながら児童に共有する。次に「〈ライブ・コーディング〉に挑戦しよう！」というめあてを示し、〈ライブ・コーディング〉の演奏動画（Web Audio Conf 2018）を 3 分ほど鑑賞する。その後、児童はそれぞれの iPad より「二版」にアクセスし、個別活動に取り組む。なお、6 年 2 組は実践①で 1 度機器の準備などを経験しており、6 年 1 組より多くの活動時間が確保できたため、個別活動の後に 4～5 名が 1 台の iPad を操作する班活動を 10 分ほど取り入れた。最後に、〈ライブ・コーディング〉を実感するために、希望する数名の児童による発表活動を行った。

発表にあたっては、タイマー機能によって演奏時間を設け、電子黒板を用いて発表者の音楽をつくる過程を聴衆役となる児童と共有した。さらに、〈ライブ・コーディング〉が盛んに実践されている場がクラブ・シーンなどであること (Roberts & Wakefield 2018, p. 294) を踏まえ、照明を落とし暗くした教室で発表に取り組んだり、聴衆役の児童には、発表者の演奏中であっても、演奏内容に対して手拍子や発言などで積極的に反応をすることを促したりすることで、その場の雰囲気を発表者に意識させることをねらった。

#### 7.2.4. 個別活動における指導と児童の姿

本授業では、児童が様々な機能を試行錯誤し理解すること (〈ティンカリング〉) を促すために、授業全体を通して、授業者は「〇〇を押したら、△△になる」といった操作の説明を行わず、児童が次の操作に迷ったときは「ヒント」を参照するように指示した。ただし、「二版」ではなく iPad の操作に関する質問 (例えば、音量の調節など) については、それぞれ対応した。

このように、授業者は「はじめに〇〇をしてみましょう」、「次に△△をしてみましょう」といったナビゲートをしなかったが、実際の授業の様子及び活動の記録から、全ての児童が個別活動において音楽づくり活動に取り組むことができていたことが確認できた。

#### 7.2.5. 発表活動における児童の姿と音楽をつくる過程の具体

本項では、発表活動における児童の様子を具体を示す。発表活動に取り組んだ児童は、いずれも様々な命令を出しながら、音楽づくり活動に取り組むことができていた。ここでは、その一端として、その中でも個性の異なる 3 名の児童 (A 児, B 児, C 児) を取り上げ、活動の記録をもとに、「二版」を用いた音楽づくり活動の具体を記す。

まず、矢継ぎ早に様々な命令を出しながらも、聴衆の反応を演奏に反映させていた A 児を取り上げる。A 児は、はじめ、「ピアノ」で 1 オクターブの音域内で跳躍を繰り返す無調の旋律を作った。次に、「鉄琴」で異なる音高を新しく重ね、続けて「木琴」を用いて同じように音を重ねた。そして、「黒電話」による音を重ねたとき、A 児は「気持ち悪い」と発言したが、それに対し聴衆の児童が「いいのいいの」と応えたことで、「黒電話」の音を残したまま、新しい命令を試行し続けた。試行にあたっては、聴衆の反応 (面白いと感じた音楽に対する笑い声、無反応など) に合わせて、音色の命令などを編集していた。演奏を通して様々な音色を試行した A 児の生成する音楽は、全体的に調性が感じられず、複数のパターンが組み合わされることで生じる不協和な響きが特徴的であった。

次に、命令の選択にじっくりと時間をかけて演奏していた B 児を取り上げる。B 児は、まず、「ピアノ」で C 音を繰り返し演奏する命令をつくり、音色を「木琴」へと変えた。次に、「ピアノ」の D 音を複数加え、休符を交えながらリズムを変化させていたが、しばらく試行した後にそれらの音を全て削除した。その後、E 音や A 音を組み合わせながら、音高を試行し続けていたものの、演奏の最後には新しい音色を試していた。B 児の音楽や

命令の特徴として、同じ高さの音を繰り返し演奏していたこと、変化させた音楽を聴いて命令を削除し再び思考する姿がみられたこと、そして、音の高さの命令を色々と試した後、音色を変えるとという別のアプローチを採ったことが挙げられる。1つ1つの命令の選択に時間がかかっていたB児の姿は、納得のいく音楽を探しているようで、そのために適切な命令を何度も試したりしているようであった。

最後に取り上げるのは、生成される音楽がひとときわ个性的であったC児である。C児は、はじめ、ハ長調だと思われる旋律を「ピアノ」で作成した。次に、その旋律に協和する音を同じリズムで「鳥の声」で重ねた。さらに、「ハーブ」によって同様に音を重ねた。途中からは「打楽器群」を使い、短くリズムを刻む命令を加えた。最後には、「ホルン」や「ギター」による実楽器では演奏出来ない高音を加えることで、あたかもシンセサイザーを活用した〈テクノ・ミュージック (Techno Music)〉のような音楽となった。スピード感をもって命令を試行する姿はA児と共通していたが、調性の感じられる音楽となっていた点が特徴的であった。

以上、「二版」を用いた発表活動の具体を、児童3名を取り上げ、演奏の様子や生成された音楽について記した。抽出しなかった発表児童も、「二版」を用いて大胆に、あるいは慎重に命令を試行したり、自身のつくる音楽や聴衆の反応が、新しい命令に影響したりする様子は同様であった。

### 7.3. 授業実践と質問紙調査の結果を踏まえた考察

本節では、「②『音楽を特徴付けている要素』の理解を深めることに関する観点」と「③直観的な操作性についての観点」についての検証を行い、《テクミュ》とプログラミング教育との関わりについて検討する。

これらの検証にあたって、6. では、児童が実際に演奏している様子や聴衆役の児童との関わり、質問紙調査の回答を手がかりとした。本章においても、質問紙調査を実施することとする。その際、6.2.1. において「②『音楽を特徴付けている要素』の理解を深めることに関する観点」の検証のために、演奏者と聴衆役との児童の関わりが手がかかりになったことを踏まえ、他者の演奏に対する自由記述を実施し、命令と実行の結果（音楽）との関わりがみられるかに着目して分析を行う。「③直観的な操作性についての観点」の検証にあたっては、6.2.2. と同様、音楽づくり活動における児童の姿と、操作の分かりやすさについての設問への回答から検討する。

質問紙調査は1組30名、2組28名（1名欠席）による計58名の児童を対象に行い、自由記述の分析にあたっては、記述内容の傾向を客観的に抽出するために、テキストマイニ



### 7.3.1.1. 他者の演奏に対する感想 (Q1) の傾向

図 34 より Q1 の結果を検討すると、児童の回答は、演奏される音楽に関するものと、表現手法（演奏の過程）に関するものとの2つに分類することができる。

前者については、「人」という語が「それぞれ」、「違う」、「色々」、「個性」、「面白い」といった語へつながっている様子から、児童は、発表者それぞれの演奏より、多様な音楽が生まれることを感じとっていると読み取れる。

後者については、「音」という語と「良い」、「演奏」、「楽器」、「変える」とのつながりから、聴衆役の児童は、発表者がどのように音楽を変化させているかに気付きながら評価していることが読み取れる。また、「沢山」、「組み合わせる」や「曲」、「重ねる」といった語のつながりや、「考える」という語と「スピード」、「数」、「時間」、「一つ」とのつながりより、聴衆役の児童は、発表者がどのように命令を組み合わせているかに焦点を当てて聴いている（観ている）と解釈される。

ここからは、観点②の検証のため、後者の、表現手法（演奏の過程）に関する回答に焦点を当て、個別の回答を検討する。

### 7.3.1.2. 命令と実行の結果との関わりへの着目

Q1の結果のうち、表現手法（演奏の過程）に関する個別の回答をみると「思いついたものをぱぱっとうってからあとからいらぬ音などを消したり、たりなかったところをつけくわえたりしているところがいいなと思いました」、「先に音をいれる人や楽器をかえる人、長さをかえる人がいてみんなどこからやるかがちがうと思いました」、「『ランダムで演奏』や『楽器を選択』を自在に操作してすごいなと思った」、「自分のやり方とはちがう操作のやり方を見て僕もまねしてみたいなと思いました」、「H さんの、たくさんの音を組み合わせるのはすごいなと思いました。はげしい音から、くらい音をくみあわせたら、よくなるんだと思いました。音量もだいじですべて 100 にしていたら、ごちゃごちゃして、よい音があんまりでないこともわかりました」といった意見がみられた。

このように、聴衆役の児童は発表者が音楽を変化させる過程、すなわち命令と実行の結果との因果関係に着目することができていた。実際の活動においても、聴衆役の児童は、7.2.5. で述べたように、発表児童の表現に対する様々な反応を示していた。

これらの結果からは、聴衆役の児童にとっても「音楽を特徴付けている要素」への着目を促す学習材となったことが指摘でき、そして、その理解を深めること、すなわち聴衆役の児童の〈プログラミング的思考〉を育むことへとつながることが期待できる。

### 7.3.2. 「直観的な操作性について」の検証

観点③の検証にあたって、質問紙調査の結果と授業における児童の姿を手がかりとする。まず、授業後に実施した質問紙調査では、「《テクミュ》の操作は、わかりやすかったで

すか」という問いを設けた。その結果を、表 23 に示す。

表 23 児童による「二版」の操作性に対する評価

《テクミュ》の操作は、わかりやすかったですか。		
4 (とてもわかりやすかった)	22名	38%
3 (わかりやすかった)	24名	41%
2 (わかりにくかった)	12名	21%
1 (とてもわかりにくかった)	0名	0%

前述したように、「二版」の作成（改良）にあたっては、直観的な操作性を実現するため、操作をよりわかりやすくする工夫を施したが、質問紙調査の結果より、「2（わかりにくかった）」と答えた児童が 21%みられ、「全ての児童にとってわかりやすい」という成果は得られなかった。ただし、「初版」を用いた実践（6.2.）で、「1（むずかしかった）」と答えた児童が 3 名（10%）いたことに対し、「二版」を用いた本実践では「1（とてもわかりにくかった）」と回答した児童が 0 名になったことは、ささやかな成果として挙げる事ができた。

このように、質問紙調査の結果からは直観的な操作性をもたせられたという指摘はできなかったが、授業実践における児童の様子（7.2.4.）をみると、「2（わかりにくかった）」と答えた 12 名を含む全ての児童が授業者のナビゲートなしに音楽づくり活動に取り組み、〈ティンカリング〉することができていた。このことは、「二版」が、「初版」と同様に、〈ティンカリング〉のための適切な直観性を備えられており、観点③が達成できたことを示す事実といえる。

### 7.3.3. 《テクミュ》とプログラミング教育との関わり

7.3.1., 7.3.2. より「二版」が観点②、観点③を達成する学習材であることが検証できた。適切な直観性が児童の〈ティンカリング〉を促し、〈ライブ・コーディング〉性によって、演奏を聴く（観る）児童にとっても、音楽を特徴付けている要素への理解を促し、結果として〈プログラミング的思考〉を育むことが期待される。

本項では、以上に加え、「ほかに感想があれば、自由に書いてください」（Q2）という質問紙調査の結果や記録した児童の活動の様子を踏まえて、《テクミュ》とプログラミング教育との関わりを検討する。

#### 7.3.3.1. 複製された命令への気付き

Q2 の傾向を共起ネットワーク（図 36）より検討する。



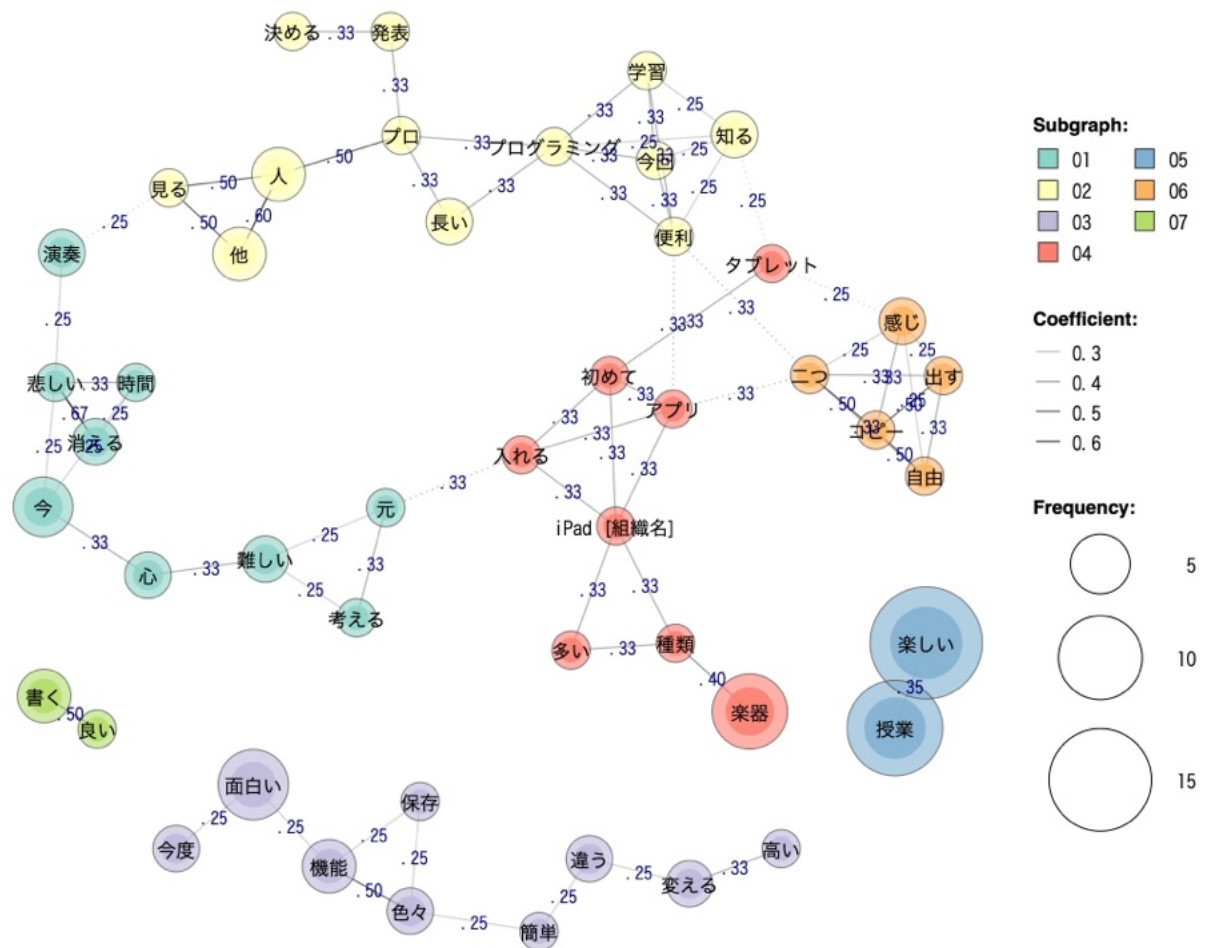


図 36 Q2 の共起ネットワーク

図 36 より Q2 の傾向は 3 つにわけられる。

1 点目は「二版」の機能に関する感想である。「色々」、「簡単」、「機能」、「面白い」、「保存」というつながりは、「二版」の命令に関する感想を表している。なお、「保存」という語は、つくった音楽を保存したいという意見であった。同様に「時間」、「消える」、「悲しい」という語のつながりは、タイマー機能が終了したときに、つくった命令が全て消えてしまう仕様に対する意見である。この仕様は、〈ライブ・コーディング〉の一回性を反映したものであるが、児童にとってはあまり馴染みのないことだった可能性が考えられる。他方、こうした回答からは、児童が指定された時間の中で演奏することへの意識が表れていることも読み取れる。また、「保存」したいという気持ちは、つくった音楽をもう 1 度聴いたり、続きの音楽をつくったりするというモチベーションへつながると考えられる。命令の組合せを「保存」する機能を備えることについては今後の課題としたい。

2 点目は、ICT 端末を用いた活動に対する感想である。「iPad」という語から「種類」、「楽器」、「多い」へとつながっていることは、児童が、ICT 端末を用いることで様々な音

色を試すことができることを感じとっていると考えられる。

3点目はプログラミングに関わる感想である。「プログラミング」、「今回」、「学習」、「知る」、「便利」という語のつながりから、「二版」とプログラミングとのつながりを感じていると解釈できる。

Q2のプログラミングに関わる感想より、「コピー」、「リズム」、「二つ」という語のつながりに注目して自由記述をみると「今回の学習を通して、プログラミングがどんなに便利なものかを知ることができた。具体的には、コピーした後のものを自由に変化させることができ、コピー前のものは変わらないからリズムをずらし、二つの音が重なる感じを出すことができた。このアプリは、二つを重ねるような操作ができるので、すごい便利だと思った」という回答がみられた。

パターンの複製機能は「初版」から取り入れていたものの、「初版」を用いた授業実践(6.)では、それに関わる意見は得られなかった。この背景として、今回、画面構成を見直したことで、複製元のパターン(=命令)と複製後のパターンとの因果関係が理解しやすくなったことが考えられる。

そして、児童のこの回答は、実際のプログラミングに関わる重要な気付きだと捉えられる。複製した命令と元の命令との関わりを知ることは、「値の交換」<sup>70</sup>などのプログラムを考えるために重要な手がかりとなる。この気付きを促せたことは、《テクミュ》とプログラミング教育との関わりの1つといえる。

### 7.3.3.2. 〈プログラミング的思考〉が働く瞬間

記録した〈ライブ・コーディング〉に取り組む児童の様子をみると、新しい命令を瞬時に繰り返すことができない児童の姿もみられた(例えば、7.2.5.において、命令の選択に時間がかかっていたB児)。それは、「命令の出し方がわからない」状態とも解釈できるが、ほかにも、命令と実行の結果(音楽や、聴衆の反応)との因果関係を考える思考が働いていると捉えることもできる。実際、他者の演奏(Q1)に対して「なぜ、そんなに思い通り(というか自分が満足いく)演奏ができるのか不思議に思った」という回答からは、《テクミュ》による音楽づくり活動で「納得できる音楽」を実現しようとする姿勢が窺える。

〈ティンカリング〉の過程では「納得できる音楽が生まれたとき」に命令と実行の結果との因果関係を見直すことで、〈プログラミング的思考〉が働く(3.5.)。しかし、〈ティンカリング〉をパフォーマンス化した〈ライブ・コーディング〉の過程には、実際に生成される音楽だけでなく、それに対する聴衆からの反応といった要素も加わるため、〈プログラミング的思考〉が働く場面が異なることが考えられる。つまり、〈ライブ・コーディング〉では、「自分にとって納得できる音楽」に加え、「聴衆にとって納得できる音楽」を考える場合にも〈プログラミング的思考〉が働くことが考えられる。

---

<sup>70</sup> 「値の交換」とは、複数の変数の値を交換することである(奥村1991, p.1)。

#### 7.4. 《テクミュ》における音楽づくり活動の過程と意義

本章では、学習材《テクミュ》について、開発時に立てた 7 つの「開発の方針と工夫」の観点のうち、再検証が必要とした『音楽を特徴付けている要素』の理解を深めることに関する観点と「直観的な操作性についての観点」に焦点をあてながら《テクミュ》を改良した。そして、授業実践を通して、観点の再検証と《テクミュ》による音楽づくり活動とプログラミング教育との関わりを検討した。その結果、次の 3 点が指摘できた。

1 点目は、「音楽を特徴付けている要素」へ着目を促すことで、聴衆役の児童の〈プログラミング的思考〉を育むことも期待できることである。

2 点目は、直観的な操作性を意識したデザインの改良によって、複製された命令と元の命令との関わりという、実際のプログラミングにつながる気づきを引き出すことができたことである。

3 点目は、〈ライブ・コーディング〉の特質を取り入れることによって、「自分にとって納得できる」だけでなく「聴衆にとって納得できる」音楽を考える場面が生まれ、その過程も、児童の〈プログラミング的思考〉を働かせる活動になりうるということである。

\*本章は、筆者が 2023 年に発表予定の次の論文を、加筆・修正した内容になる。

長山弘 (2023) 「〈ライブ・コーディング〉の特質を取り入れた音楽づくり活動のための Web アプリケーションの開発 (2) —命令と実行の結果との即時性及び画面設計に焦点を当てて—」『初等教育カリキュラム学会誌』11 号, 採択決定済み。

## 【総括】

本研究における学習材開発は、小学校段階のプログラミング教育における〈プログラミング的思考〉が、必ずしも音楽科の表現活動における思考にそのまま当てはめることができないのではないかという問題意識から始まった。

そこで、小学校音楽科においてプログラミング教育を実施するための手立てとして〈ティンカリング〉という概念に着目し、〈ティンカリング〉をパフォーマンス化したものとして〈ライブ・コーディング〉という表現手法を取り上げた。それらを踏まえ、〈ライブ・コーディング〉の表現過程を、小学校音楽科の音楽づくり活動に照らし合わせながら開発した学習材が Web アプリケーション《テクミュ (Tec-Mu)》である。

《テクミュ》は、常に鳴り響く音楽をリアルタイムで変化させるアプリケーションである。そのようなアプリケーションは、5.3. で挙げた海外での事例で用いられているものも含め、これまでに国内外に数多くの優れた前例がみられる。ただ、2000年代初頭には実践され始めた〈ライブ・コーディング〉という営みが、いまだに日本の小学校現場の実践報告にみられないのは、小学校音楽科で活用することを前提とした適切なアプリケーションが作られてこなかったことも一因だったのではないだろうか。学習指導要領の内容や教科書の記述を踏まえた設計や、多様な学校現場や異なる ICT 端末でも動くという点は、いずれも、小学校音楽科の〈学習材〉としての要点である。《テクミュ》の新規性は、そうした点にある。

さらに、本研究では、《テクミュ》を用いた授業実践を通して、実行の結果（音楽）と命令との因果関係を意識しながら音楽づくり活動を行うことで〈プログラミング的思考〉の育成へと結びつくことを示すこともできた。そこで働かせる思考は、「手引」の示す「コンピュータを動作させるための手順（例）」（文部科学省 2020a, p. 14）や「プログラミング的思考を働かせるイメージ」（p. 16）とは、やや異なるものである。具体的には次の3点である。

1 点目は、結果となる音楽や、その源となる発想、思いや意図が、予め明確ではなく、試行錯誤を通して具体化されることである（2.6.）。この思考の過程は音楽科のみならず、創造的な思考が求められる活動において共通して指摘することができよう。

2 点目は、この思考の過程においては、「手引」（文部科学省 2020a, p. 15）では望ましくないとされている、「思い付きや当てずっぽう」な命令も重要な役割をもつことである。なぜなら、それらの結果は、思いや意図が明らかな場合には得られないような、新しい発想や方向性を生む大切な契機となるからである（3.3.）。これまで、多くの音楽作品が、偶然性や不確実性を取り入れてきた。そこには、想定外の結果を積極的に求めていく態度が反映されている。2.5. の学習の過程の検討にあたっては、意図しなかった結果にも、自分にとっての価値を発見する可能性が含まれることを述べた。こうした態度の重要性は、音楽の創作のみにとどまらない。実際、歴史に残る「創造的な発明」にも、想定外の結果が

契機となったものは少なくない。例えば、栞として用いられ付箋紙は、もともと、接着剤を開発する過程で生まれた失敗作であった。すぐに剥がれてしまうその特性は、接着剤としては失敗作だが、その特性を活かせることに気づいたことで付箋紙が発明されたのである（3M 2018）。このような、「失敗」さえも含む想定外の結果に新たな可能性を見出す力は、創造性の一部を成すとても重要な力といえる。

3点目は、次々とあらわれる課題をすぐさま捉えて解決する即興的な思考である（4.4.）。「手引」が示す〈プログラミング的思考〉を育む活動の過程では、試行錯誤を通して与えられた大きな課題を解決する。対して、〈ライブ・コーディング〉による表現活動では、生成した音楽に対して、採用するかどうかの判断を瞬時に下す過程の繰り返しである。そして、生成した音楽が意図した結果と違った、または、聴衆の反応が期待したものにならなかった場合、次にどのような方策をとるかを考えることも、大切な思考過程となるだろう。こうした瞬時の柔軟な判断によって命令を構築し、「いま、ここ」の影響を受けながら、目標を明確化するという活動は、ゲーム的な要素を考慮して活動の過程に位置付けることによって、その効果を高めることができる。具体的には、直観的に操作できること、命令をやり直す（元に戻す）ことができないこと、結果が音楽としてすぐに確認できることが、音楽を生み出す命令を自ら組み合わせるといった活動がもつ「楽しさ」に良い影響をもたらした。ほかにも、演奏できる時間に制約を設けた「タイマー」機能も、演奏の「はじめ」や「おわり」といった展開を考えることに寄与しただけでなく、残り時間を意識しながら演奏に取り組むことで、スリルや緊張感といったゲーム性を高めることができた。

以上を踏まえて、《テクミュ》の表現活動の過程を示したものを図 37 に示す。

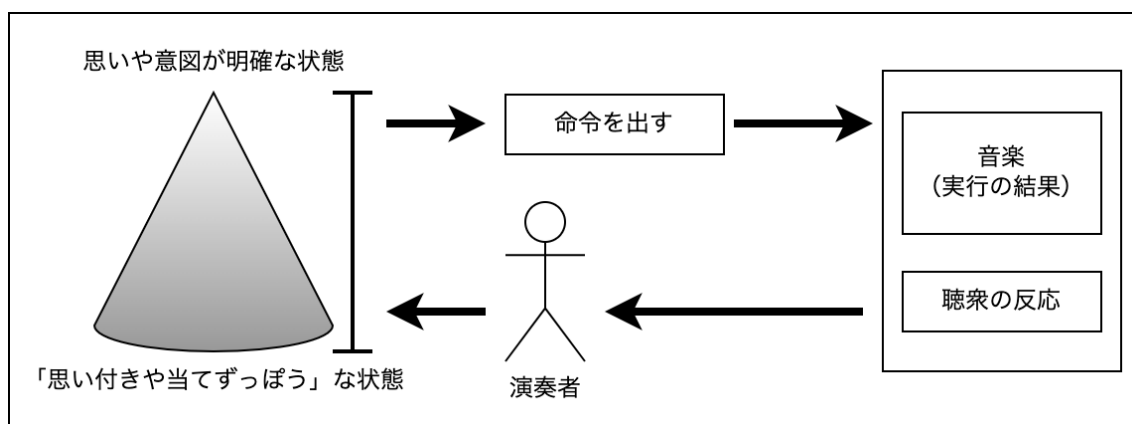


図 37 《テクミュ》における音楽づくり活動の過程

この活動過程では、演奏者は、はじめ、「何らかの命令を出す」という「思い付きや当てずっぽう」な試行からはじまる。そして、〈ティンカリング〉を通して、命令の働き（「音楽を形づくっている要素」との関わり）を理解するにつれ、「思い付きや当てずっぽう

う」な命令は次第に「意図した」命令へと変容する。他方、演奏者が新しいことを試してみようとするとき、命令は再び「思い付きや当てずっぽう」に近いものに戻ってしまう。だが、音楽表現の探究にはそのことも欠かせない。すなわち「思い付きや当てずっぽう」な状態と、思いや意図が明確な状態とを行きつ戻りつしながら活動を行うことになるのである。

命令を組み立て実行しそれらの因果関係を学ぶこと、そして、新しいことを試して再び「思い付きや当てずっぽう」な状態から新たな命令を試すこと、これら両方の過程が含まれる経験は、いわゆる（知識や技能を）「身に付ける」学びと、「探究する」学び、それら両方をもたらすことへとつながるのではないだろうか。

図 37 に示した過程は、文部科学省による「コンピュータを動作させるための手順（例）」や、2.5. で示した「小学校音楽科の表現領域におけるプログラミング教育を取り入れた学習の過程」と比べて、一見単純に見える。しかし、その過程のなかには、自らの思いや意図の瞬間的な省察、音楽を形づくっている要素に関わる学び、幅広い表現活動を生み出す契機、これらすべてが埋め込まれている。

最後に、今後の展望として、本研究からつながる研究的・実践的関心を 3 点述べる。

1 点目は、プログラミング教育—とりわけ〈プログラミング的思考〉—の評価方法の検討である。先述したように、文部科学省は「手引」において〈プログラミング的思考〉の評価方法について明示していない。また、本研究の授業実践（7.）においても、児童の〈プログラミング的思考〉を育むことが「期待できる」という結論に留まった。今後は、他教科における〈プログラミング的思考〉や論理的思考の評価方法も手がかりとしながら、小学校音楽科の特質を踏まえた評価方法を検討したい。

2 点目は、小学校低学年から高学年までの、段階ごとに応じた小学校音楽科におけるプログラミング教育のためのカリキュラムを開発することである。本研究では、第 6 学年を対象とした授業を実践した。しかしながら、低学年を対象とした授業の場合、高学年とは異なる結果が得られたであろう。発達段階に相応しい〈プログラミング的思考〉を検討しつつ、小学校 6 年間を見通した音楽科におけるプログラミング教育のカリキュラムを具体化したい。

3 点目は、《テクミュ》のさらなる改良である。今後は、活動の可能性をより広げるために、今回児童から要望のあった「保存」機能や、複数名が別々の端末を介して共演できる機能を取り入れたい。さらに、1 点目で述べた〈プログラミング的思考〉の評価方法を踏まえ、自身のつくった音楽の変容を見直すことができる機能を取り入れたら、2 点目で挙げたカリキュラムを踏まえ、各学年に応じた使い方ができるような調整をできるようにすることによって、より広く用いられる学習材にしていきたい。

筆者は、これからの「変化が激しく将来の予測が困難な時代」（「有識者会議」2016）を生きる児童にとって、「有識者会議」の示す〈プログラミング的思考〉はもちろん、本論文で議論した即興的な思考も、重要な力だと考えている。本研究で示した学びのあり方が、

プログラミングや音楽表現に留まらず、「自らの人生で直面する、様々な課題を柔軟に乗り越えていくための力」を育む学びへとつながることを願いつつ、筆を擱くこととする。

## 謝辞

最後になりましたが、本研究を進めるにあたり、お世話になった方々へ感謝を申し上げます。

広島大学大学院教育学研究科でご指導いただいた全ての先生、とくに、主任指導教員の寺内大輔先生、副指導教員の権藤敦子先生、植田敦三先生、田中秀幸先生、山崎敬人先生に感謝を申し上げます。権藤敦子先生には、広島大学大学院に入学する前より、研究についてのご相談をさせていただき、入学後も小学校音楽科におけるプログラミング教育の問題の所在やその意義を検討するにあたって、的確なご指導をいただきました。植田敦三先生、田中秀幸先生、山崎敬人先生には、〈プログラミング的思考〉と〈コンピュテーショナル・シンキング〉との関わり、〈ティンカリング〉や〈ライブ・コーディング〉の教育的意義についてなど、本研究で取り上げたそれぞれの理論や実践の結果をまとめていくにあたって、多くのご指導をいただきました。そして、主任指導教員の寺内大輔先生には大学院に入学してから6年間、論文指導や学会発表のための指導はもちろんのこと、研究に関わる書籍や楽曲の紹介、学習材開発のためのご助言など、あらゆる場面で援けていただきました。また、折々の、雑談を交えながらのお話しも、本研究における私の発想をしばしば引き出してくれる、示唆に富むものでした。リサーチ・クエスションの検討から、博士論文の仕上げに至るまで、親身になってご指導くださったことには、感謝の申し上げようもございません。

私は、広島大学教育学研究科博士課程後期の入学と同時に、広島大学附属東雲小学校において、非常勤講師として6年間勤めることになりました。そこでお会いした先生方は、小学校での勤務経験がない私に、児童を指導する姿勢や、児童とどのように楽しく音楽科の授業に取り組むかといった実務上の様々なことをご指導くださいました。このことは、日ごろの授業実践の支えとなったことはもちろんのこと、本研究における学習材開発にも活かすことができました。また、《テクミュ》を用いた研究授業に協力してくれた子どもたちは、真摯にかつ楽しみながら、活動に取り組んでくれました。なかには、授業実践後に、《テクミュ》を「自宅に帰ってから自分で遊んでみた」、「妹に教えながら、一緒に遊んだ」と伝えてくれた児童もいました。開発者として、これほど嬉しいことはありません。私は、そうした子どもたちの姿勢に触れることで、学習材開発のためのモチベーションを高めることができましたと感じています。附属東雲小学校の山崎敬人元校長先生、松浦武人校長先生、谷栄次副校長先生をはじめとする全ての先生に感謝を申し上げます。

大学院のゼミにおいては、吉崎優葵様、福田麟太郎様、住田新太郎様、伊原本木幸馬様、中村恵美子様より、論文や大会発表の内容について、多くのご助言をいただきました。作曲家・即興演奏家の三宅珠穂様からは、研究内容へのご助言をいただいたり、子どもたちを対象とした音楽づくりワークショップの講師をご一緒させていただいたりしました。呉



工業高等専門学校（当時）の富村憲貴様からは、プログラミング教育のセミナーにおいてご助言いただきました。

学習材《テクミュ》の開発にあたっては、元東京電力エンジニアの楠本匡介様に技術面やハードウェア・ソフトウェア構成図の示し方などについてご相談させていただきました。また、日本音楽教育学会大会、日本音楽即興学会大会、初等教育カリキュラム学会大会においては、開発過程に位置付く成果を発表した際、その場にご参加くださった方々より、貴重なご意見を数多くいただきました。

教育学研究科及び附属東雲小学校勤務での同期となる大野木俊文様からは、研究、授業のいずれの面からのご助言をいただきました。学部・修士課程で同期であった木下和彦様には、私が博士課程後期に入学する前から親身になって相談に応じていただき、入学後も論文の内容や就職などについてご助言いただきました。さらに、学部・修士課程時代にお世話になった先生方、とりわけ高澤ひろみ先生、加藤富美子先生、山内雅弘先生、久木山直先生には、卒業・修了後も親身になってご助言いただき、優しい励ましの言葉をいただきました。

私の博士論文は、広島大学大学院に入学した 2017 年度から 6 年のあいだ、上に挙げさせていただいた方々を含む、多くの人のお力をいただくことで、執筆を進めていくことができました。その 6 年間のほとんどが、「これで良いのか」と悩みながら研究を進めていく日々でした。しかし、多くの方々からお力添えをいただき、学会誌への投稿論文がはじめて採択・掲載されたときの喜びはいまでも覚えています。ありがとうございました。

最後に、私をずっと支えてくれた家族、父・長山貴、母・長山和代、祖父・故長山孝典、祖母・長山芳子、祖父・故青木和道、祖母・青木陽子に、心から感謝致します。

```
1 while (true) {  
2   console.log("Thank you very much.");  
3 }
```

2023 年 1 月吉日

長山弘

## 【引用・参考文献】

- 3M (2018) 「<こんなところにも 3M>ポスト・イット® ノート 不屈の魂が生んだ世界のオフィスの必需品」 インターネット, [https://www.3mcompany.jp/3M/ja\\_JP/medical-jp/story/postit/](https://www.3mcompany.jp/3M/ja_JP/medical-jp/story/postit/) (2022/10/17 閲覧)
- Algorave (2017) 「guidelines/README\_ja.md at master Algorave/ guidelines GitHub」 インターネット, [https://github.com/Algorave/guidelines/blob/master/README\\_ja.md](https://github.com/Algorave/guidelines/blob/master/README_ja.md) (2019/9/7 閲覧)
- Algorave (n.d.) 「Home-Algorave | algorithmic dance culture」 インターネット, <https://algorave.com> (2019/5/12 閲覧)
- Brown, A. R. (2006) Code Jamming. *In M/C Journal*, 9(6). <http://journal.media-culture.org.au/0612/03-brown.php> (2019/2/10 閲覧)
- Burnard, P., Brown, N., Florack, F., Major, L., Lavicza, Z. and Blackwell, A. (2014) 「Sonic Pi Live & Coding: A collaborative research project」 インターネット, [https://www.academia.edu/9112375/Sonic\\_Pi\\_Live\\_and\\_Coding\\_-\\_A\\_collaborative\\_research\\_project](https://www.academia.edu/9112375/Sonic_Pi_Live_and_Coding_-_A_collaborative_research_project) (2021/8/3 閲覧)
- CANVAS (2016) 「リズムアンサンブルをつくろう」 インターネット, <http://csforall.jp/case/432/> (2017/6/7 閲覧)
- Collins, N., McLean, A., Rohrhuber, J. and Ward, A. (2003) Live coding in laptop performance. *In Org Sound*, 8(3), pp. 321-330.
- Collins, N. (2011) Live Coding of Consequence. *In Leonardo*, 44(3), pp. 207-211.
- Denning, P. J. (2009) The Profession of IT: Beyond Computational thinking. *In Communications of the ACM*, 52(6), pp. 28-30.
- Gazzano, A. (2021) Japan's programming education: a critical focus on music in elementary schools. *In Arts Education Policy Review*, pp. 1-13.
- Howe, H. S. Jr・嶋津武仁 (1996) 「電子音楽」 神前尚生 (訳), 柴田南雄・遠山一行 (監) 『ニューグローヴ世界音楽大事典』 第 11 巻, 講談社, pp. 341-343.
- Hu, C. (2011) Computational thinking: what it might mean and what we might do about it. *In Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*, pp. 223-227.
- Kassler, M. and Howe, H. S. Jr. (1994) 「コンピュータと音楽」 神前尚生 (訳), 柴田南雄・遠山一行 (監) 『ニューグローヴ世界音楽大事典』 第 7 巻, 講談社, pp. 193-200.
- Paulus, E. (n.d.) 「the works of Edo Paulus」 インターネット, <http://www.eude.nl/projects/max-live-coding/> (2019/3/28 閲覧)
- Roberts, C. and Wakefield, G. (2018) Tensions and Techniques in Live Coding Performance. *In The Oxford Handbook of Algorithmic Music*, McLean, A. and Dean, R. (Eds), Oxford University Press, pp. 293-317.

- Ruthmann, A., Heines, J. M., Greher, G. R., Laidler, P. and Charles S. II (2010) Teaching computational thinking through musical live coding in scratch. *In Proceedings of the 41st ACM technical symposium on computer science education*, pp. 351-355.
- Slub (n.d.) 「slub|smoothishsinusoidalwaveform」 インターネット, <http://slub.org> (2019/3/16 に関連)
- Terauchi, D. and Myodo, H. (2021) Exploring the pedagogical possibilities of the idea of composition based on children's interests and strengths. *In the Proceedings of The 13th Asia-Pacific Symposium for Music Education Research: Exploring Possibilities and Alternatives in a Changing Future*, Morijiri, Y., Imada, T., Ogawa, Y. (ed), Japan Music Education Society, pp. 219-227.
- Toplap (2009) 「Raging code announcement and report」 インターネット, [https://toplap.org/wiki/Raging\\_code\\_announcement\\_and\\_report](https://toplap.org/wiki/Raging_code_announcement_and_report) (2019/10/9 に関連)
- Toplap (2010) 「Coding Bull announcement and report」 インターネット, [https://toplap.org/wiki/Coding\\_Bull\\_announcement\\_and\\_report](https://toplap.org/wiki/Coding_Bull_announcement_and_report) (2019/10/9 に関連)
- Type\_T・堀田龍也 (2021) 『事例と動画でやさしくわかる！小学校プログラミングの授業づくり』学陽書房.
- Wachsmann, K., Hornbostel, E. M. and Sachs, C. (1994) 「楽器の分類」 田島みどり (訳), 柴田南雄・遠山一行 (監) 『ニューグローヴ世界音楽大事典』第4巻, 講談社, pp. 449-454.
- Ward, A., Rohrerhuber, J., Olofsson, F., McLean, A., Griffiths, D., Collins, N., Alexander, A. (2004) Live Algorithm Programming and a Temporary Organisation for its Promotion. *In Read Me: Software Art and Cultures*, pp. 243-261.
- Web Audio Conf (2018) 「Improvisation by Charles Roberts」 インターネット, <https://youtu.be/qi8VX6GawLM> (2022/10/13 に関連)
- Wing, J. M. (2006) Computational thinking. *In Communications of the ACM*, 49(3), pp. 33-35.
- 赤堀侃司 (2018) 『プログラミング教育の考え方とすぐに使える教材集』ジャムハウス.
- 赤堀侃司・久保田善彦 (監) (2018) 『これならできる 小学校教科でのプログラミング教育』東京書籍.
- 阿部和広 (2016) 「子供の創造的活動とプログラミング学習」『情報処理』57巻4号, pp. 349-353.
- 阿部和広・豊福晋平・芳賀高洋 (監) (2018) 『小学校の先生のための Why!? プログラミング 授業活用ガイド』日経 BP 社.
- 阿部和広 (2021) 「2年目に入った小学校プログラミング教育」『コンピュータ&エデュケーション』51巻, pp. 20-26.
- 有元典文 (2011) 「第2章 アーティファクトの心理学」 茂呂雄二・田島充士・城間祥子 (編) 『社会と文化の心理学—ヴィゴツキーに学ぶ』世界思想社, pp. 32-54.
- アンダーソン, クリス (2012) 『MAKERS 21世紀の産業革命が始まる』関美和 (訳), NHK 出版.
- 安藤明伸・額田一利 (2021) 『小学館 eBooks ここがポイント！小学校プログラミング教育の要点ズバリ！ embot で楽しく実践できる指導案 特選 15』小学館.
- 池本公平 (編) (2016) 『Software Design』2016年4月号, 技術評論社.

- 磯川祐地・佐藤和紀・山本朋弘・宮田明子・鈴木広則・清水雅之・堀田龍也（2021）「小学校プログラミング教育に関する先行研究の動向からみたカリキュラム・マネジメントの方策の検討」『上越教育大学研究紀要』40巻2号, pp. 341-350.
- 磯辺秀司・小泉英介・静谷啓樹・早川美德（2016）『コンピューショナル・シンキング』共立出版.
- 磯部征尊（2020）『必須化！小学校のプログラミング学習—成功する全体計画&授業づくり』学芸みらい社.
- ヴィゴツキー, エリ・エス（1987）『心理学の危機』柴田義松（訳）, 明治図書出版.
- ウィルキンソン, カレン・ペトリッチ, マイク（2015）『ティンカリングをはじめよう アート, サイエンス, テクノロジーの交差点で作って遊ぶ』金井哲夫（訳）, オライリー・ジャパン.
- 上松恵理子（編）（2016）『小学校にプログラミングがやってきた！』三省堂.
- 愛媛新聞（2021-11-09）「基金を活用した港湾整備廃止を行政レビュー終了」愛媛新聞 2021年11月9日朝刊.
- 大阪市教育センター（n.d.）「プログラミング指導案・教材・授業づくり」インターネット,  
<http://www.ocec.jp/center/index.cfm/35,17115,279,html>（2018/9/2に閲覧）
- 太田剛・森本容介・加藤浩（2016）「諸外国のプログラミング教育を含む情報教育カリキュラムに関する調査—英国, オーストラリア, 米国を中心として—」『日本教育工学会論文誌』40巻3号, pp. 197-208.
- 岡崎善弘・大角茂之（2022）「設計図を用いた説明がプログラミング的思考の『分解』の理解に与える効果」『日本教育工学会論文誌』46巻3号, pp. 485-491.
- 岡田美智男（2010）「リソースの中に埋め込まれた学び—一次世代ロボット創出プロジェクトの実践から—」佐伯胖（監）（編）『「学び」の認知科学事典』大修館書店.
- 奥井美月・山守一徳・若松如美（2021）「Scratchによる母音キーを使った音楽演奏のプログラミング授業実践」『第83回全国大会講演論文集』2021巻1号, pp. 799-800.
- 奥村晴彦（1991）『ソフトウェアテクノロジー 13 C 言語による最新アルゴリズム事典』技術評論社.
- 小原光一・飯沼信義・浦田健次郎（監）（2020）『小学生の音楽』1-6, 教育芸術社.
- 音楽電子事業協会（1998）「MIDI1.0規格書PDF版」インターネット,  
<https://amei.or.jp/midistandardcommittee/MIDI1.0.pdf>（2021/8/14に閲覧）
- 倅山恵（2019）「プログラミング的思考力を育む音楽づくりの実践—旋律創作を中心に—」『東京学芸大学附属小金井小学校研究紀要』41巻, pp. 119-124.
- 金井哲夫（2015）「訳者あとがき」ウィルキンソン, カレン・ペトリッチ, マイク『ティンカリングをはじめよう アート, サイエンス, テクノロジーの交差点で作って遊ぶ』オライリー・ジャパン, p. 225.
- 兼宗進・中谷多哉子・御手洗理英・福井眞吾・久野靖（2001）「オブジェクト指向言語『ドリトル』を利用した情報教育について」『情報教育シンポジウム2001論文集』2001巻9号, pp. 275-282.
- 兼宗進（2007）「教育用プログラミング言語と授業利用—1. 教育用プログラミング言語の動向—」『情報処理』48巻6号, pp. 589-593.

- 川崎弘作 (2021) 「総合的な学習の時間における小学校プログラミング教育の動向」『岡山大学大学院教育学研究科研究集録』177 巻, pp. 55-63.
- 川島芳昭・菊地章 (2019) 「第 2 章 小学校におけるアルゴリズム学習とその評価」日本産業技術教育学会 (編) 『小・中・高等学校でのプログラミング教育実践—問題解決を目的とした論理的思考力の育成—』九州大学出版会, pp. 61-68.
- 木石岳 (編) (2018) 『はじめての〈脱〉音楽 やさしい現代音楽の作曲法』川島素晴 (監), 自由現代社.
- 北川真里菜 (2022) 「小学校音楽科におけるプログラミングキット (micro:bit) を用いた音楽づくり—音楽をつくる過程における児童の省察や試行錯誤の促進をめざして—」『和歌山大学教職大学院紀要: 学校教育実践研究』6 巻, pp. 117-126.
- 教育デザイン研究所 (編) (2017) 『小学校 プログラミング教育がわかる, できる 子どもが夢中になる 各教科の実践』学事出版.
- 金田一春彦・金田一秀穂 (編) (2017) 『学研 現代新国語辞典 改訂第六版』学研プラス.
- 久保田晃弘 (2005) 「ライブ・コーディングの可能性」『ユリイカ』37 巻 3 号, 青土社, pp. 147-160.
- 久保田晃弘・畠中実 (編) (2018) 『メディア・アート原論 あなたは, いったい何を探しているのか?』フィルムアート社.
- 蔵満逸司 (2019) 『小学校プログラミング教育の考え方・進め方』黎明書房.
- 黒上晴夫・堀田龍也 (2017) 『教育技術 MOOK 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ア』小学館.
- 黒田昌克・森山潤 (2018) 「小学校段階におけるプログラミング教育の実践に向けた教員の課題意識と研修ニーズとの関連性」『日本教育工学会論文誌』41 巻 Suppl., pp. 169-172.
- 小池翔太 (2018) 「小学校第 3 学年の総合的な学習の時間におけるプログラミング教育のカリキュラム開発の試み」『千葉大学大学院人文公共学府研究プロジェクト報告書』324 巻, pp. 23-32.
- 小枝洋平 (2020) 「身体の動きを基盤とした即興演奏の可能性」『音楽教育実践ジャーナル』Vol. 18, pp. 46-55.
- コープ, デイヴィッド (2011) 『現代音楽キーワード事典』石田一志・三橋圭介・瀬尾史穂 (訳), 春秋社.
- コールドウェル, ヘレン・スミス, ニール (2018) 『ある日, クラスメイトがロボットになったら!? イギリスの小学生が夢中になった「コンピュータを使わない」プログラミングの授業』佐分利敏晴・柏木恭典・小倉定枝 (訳), 学芸みらい社.
- 後藤英 (2016) 『Emprise—現代音楽の系譜から, コンピューター・ミュージック, エレクトロニック・ミュージック, ニュー・メディア・アート, 新たなパフォーマンスへの進化』スタイルノート.
- 小林未歩・宇都宮晃・宮澤豪臣・福島健介 (2018) 「授業実践に基づく小学校プログラミング教育「評価規準」の提案—授業における評価規準の必要性を踏まえて—」『2018 PC Conference』インターネット, <https://gakkai.univcoop.or.jp/pcc/2018/papers/pdf/pcc030.pdf> (2022/12/28 に閲覧)
- 小林祐紀・兼宗進 (編) (2017) 『コンピュータ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.

- 小林祐紀・兼宗進・白井詩沙香・白井英成（監）（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]』翔泳社.
- 小林祐紀・兼宗進・中川一史（2019）『小学校プログラミング教育の研修ガイドブック』翔泳社.
- 金洋太（2022）『micro:bit で STEAM 教育 子供の探究する力を伸ばすプログラミング教育』ラトルズ.
- 櫻田安志（2018）「小学校低学年の児童を対象とした音楽プログラミングの導入」『弘前大学教育学部研究紀要クロスロード』22号, pp. 41-50.
- サッチマン, ルーシー・A.（1999）『プランと状況的行為—人間—機械コミュニケーションの可能性—』佐伯胖（訳）, 産業図書.
- 佐藤和紀・石坂諭美・磯川祐地・山本朋弘・堀田龍也（2020）「教員経験, 教員研修, 同僚やメディアからの情報が, 小学校教員のプログラミング教育への期待や課題意識に与える影響の検討」『常葉大学教育学部紀要』40号, pp. 107-115.
- 志民一成（2017）「音楽科における ICT の活用」山下薫子（編）『平成 29 年版 小学校新学習指導要領ポイント総整理 音楽』東洋館出版社, pp. 125-127.
- 志民一成（2019）「音楽科における新たな価値を生み出す豊かな創造性の育成」『季刊 音楽鑑賞教育』36 巻, pp. 54-57.
- 清水稔（2017）「学校教育で音楽をつくることの再認識—音楽は自己のイメージから生成しない—」『音楽教育学』46 巻 2 号, pp. 25-36.
- 首相官邸（2012）「日本経済再生本部の設置について（平成 24 年 12 月 26 日 閣議決定）」インターネット, <https://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/konkyo.pdf>（2022/3/15 に閲覧）
- 首相官邸（2013a）「第 6 回産業競争力会議議事要旨」インターネット, <https://www.kantei.go.jp/jp/singi/keizaisaisei/skkkaigi/dai6/gijiyousi.pdf>（2022/3/2 に閲覧）
- 首相官邸（2013b）「産業競争力会議の開催について」インターネット, <https://www.kantei.go.jp/jp/singi/keizaisaisei/skkkaigi/konkyo.html>（2022/3/2 に閲覧）
- 首相官邸（2013c）「日本再興戦略—JAPAN is BACK—本文」インターネット, [https://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/saikou\\_jpn.pdf](https://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/saikou_jpn.pdf)（2022/3/2 に閲覧）
- 首相官邸（2013d）「世界最先端 IT 国家創造宣言 平成 25 年 6 月 14 日」インターネット, <https://www.kantei.go.jp/jp/singi/it2/kettei/pdf/20130614/siryoushi.pdf>（2022/3/18 に閲覧）
- 首相官邸（2015）「これからの時代に求められる資質・能力と, それを培う教育, 教師の在り方について（第七次提言）（平成 27 年 5 月 14 日）」インターネット, [https://www.kantei.go.jp/jp/singi/kyouikusaisei/pdf/dai7\\_1.pdf](https://www.kantei.go.jp/jp/singi/kyouikusaisei/pdf/dai7_1.pdf)（2022/3/18 に閲覧）
- 首相官邸（2016a）「4/19 第 26 回産業競争力会議議事録」インターネット, <https://www.kantei.go.jp/jp/singi/keizaisaisei/skkkaigi/dai26/gijiroku.pdf>（2021/10/12 に閲覧）
- 首相官邸（2016b）「5/19 第 27 回産業競争力会議議事録」インターネット, <https://www.kantei.go.jp/jp/singi/keizaisaisei/skkkaigi/dai27/gijiroku.pdf>（2022/9/26 に閲覧）
- 首相官邸（2020a）『日本経済再生本部の設置について』の廃止について」インターネット, <https://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/haisi.pdf>（2022/10/16 に閲覧）

- 首相官邸 (2020b) 「これまでの成長戦略について」 インターネット,  
<https://www.kantei.go.jp/jp/singi/keizaisaisei/kettei.html> (2022/3/15 に閲覧)
- 首相官邸 (2021a) 「教育再生実行会議 開催状況」 インターネット,  
<https://www.kantei.go.jp/jp/singi/kyouikusaisei/kaigi.html> (2022/3/18 に閲覧)
- 首相官邸 (2021b) 「教育再生実行会議の廃止等について (令和 3 年 9 月 17 日閣議決定)」 インターネット,  
[https://www.kantei.go.jp/jp/singi/kyouikusaisei/pdf/kyouikusaisei\\_haishi.pdf](https://www.kantei.go.jp/jp/singi/kyouikusaisei/pdf/kyouikusaisei_haishi.pdf) (2022/3/18 に閲覧)
- 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 (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)」 インターネット, [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) (2021/3/15 に閲覧)
- 新山王政和 (2017) 「12 ICT の活用」 宮崎新悟・志民一成 (編) 『平成 29 年版 小学校新学習指導要領の展開 音楽編』 明治図書出版, pp. 94-95.
- 総務省 (2017) 「『未来の学びコンソーシアム』のサイトの公開」 インターネット,  
[https://www.soumu.go.jp/menu\\_kyotsuu/important/kinkyu02\\_000259.html](https://www.soumu.go.jp/menu_kyotsuu/important/kinkyu02_000259.html) (2022/3/15 に閲覧)
- 高橋健一・谷口禎英 (2017) 「良いコードを書く理由」 稲尾尚徳 (編) 『WEB+DB PRESS』 99 巻, 技術評論社, pp. 10-13.
- 瀧川淳 (編) (2022) 『音楽科授業サポート BOOKS 1 人 1 台端末でみんなつながる! 音楽授業の ICT 活用ハンドブック』 明治図書.
- 立田ルミ (2020) 「小学校におけるプログラミング教育の現状: 最近の動向」 『情報学研究』 9 巻, pp. 53-57.
- 辰己丈夫 (2001) 「情報教育の音楽化」 『情報処理学会研究報告コンピュータと教育 (CE)』 2001 巻 101 号, pp. 39-46.
- 辰己丈夫・兼宗進・久野靖 (2005) 「ドリトルと『情報教育の音楽化』」 『情報処理学会研究報告コンピュータと教育 (CE)』 2005 巻 123 号, pp. 77-84.
- 田所淳 (2018) 『演奏するプログラミング, ライブコーディングの思想と実践—Show Us Your Screens』 ビー・エヌ・エヌ新社.
- 谷和樹 (2018) 「先生と子どもたちの役に立つ『プログラミング授業の教科書』」 コールドウェル, ヘレン・スミス, ニール 『ある日, クラスメイトがロボットになったら!? イギリスの小学生が夢中になった「コンピュータを使わない」プログラミングの授業』 学芸みらい社, pp. 2-3.
- 谷和樹 (監) (2020) 『授業の腕が上がる新法則シリーズ「プログラミング」授業の腕が上がる新法則』 学芸みらい社.
- 中央教育審議会 (2016a) 「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)」 インターネット,  
[https://www.mext.go.jp/b\\_menu/shingi/chukyo/chukyo0/toushin/\\_icsFiles/afieldfile/2017/01/10/1380902\\_0.pdf](https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo0/toushin/_icsFiles/afieldfile/2017/01/10/1380902_0.pdf) (2021/3/19 に閲覧)

- 中央教育審議会（2016b）「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）【概要】」インターネット，  
[https://www.mext.go.jp/component/b\\_menu/shingi/toushin/\\_icsFiles/afieldfile/2016/12/27/1380902\\_1.pdf](https://www.mext.go.jp/component/b_menu/shingi/toushin/_icsFiles/afieldfile/2016/12/27/1380902_1.pdf)（2022/2/7 閲覧）
- つくば市教育局総合教育研究所（編）（2018）『これならできる小学校教科でのプログラミング教育』赤堀侃司・久保田善彦（監），東京書籍.
- つくば市教育局総合教育研究所（2019）「つくば市プログラミング学習の手引き第3版」インターネット，  
<https://www.tsukuba.ed.jp/~programming/?wpdmpro=つくば市プログラミング学習の手引き第3版&wpdmml=788&refresh=63314d111d6bd1664175377>（2022/9/26 閲覧）
- つくば市総合教育研究所（2021）「つくば市プログラミング学習の手引き 第4版」インターネット，  
<https://www.tsukuba.ed.jp/~programming/?wpdmpro=つくば市プログラミング学習の手引き第4版&wpdmml=1047&refresh=6331287354be91664166003>（2022/9/26 閲覧）
- 坪能由紀子（2010）『音楽づくり』に見る器楽教育の変容『音楽教育実践ジャーナル』7巻2号，pp. 26-33.
- 寺内大輔（2020）「〈演奏行為のアルゴリズムを構築する活動〉の提案—小学校音楽科におけるプログラミング教育のあり方の検討をとおして—」『音楽教育学』49巻2号，pp. 25-32.
- 寺内大輔（2022）「音楽づくりの学習材《ステージ》の捉え直しと改変—プログラミング教育のための学習材化に向けて—」『初等教育カリキュラム研究』10号，pp. 101-110.
- 東京学芸大学プログラミング教育研究会（編）（2019）『小学校におけるプログラミング教育の理論と実践』学文社.
- 利根川裕太・佐藤智（2017）『先生のための小学校プログラミング教育がよくわかる本』翔泳社.
- 内閣官房（2021）「行政事業レビュー 秋のレビュー（秋の年次公開検証）特設ページ」インターネット，  
<https://www.gyokaku.go.jp/review/aki/>（2022/3/10 閲覧）
- 内閣府（2019）「安心と成長の未来を拓く総合経済対策」インターネット，  
[https://www5.cao.go.jp/keizai1/keizaitaisaku/2019/20191205\\_taisaku.pdf](https://www5.cao.go.jp/keizai1/keizaitaisaku/2019/20191205_taisaku.pdf)（2022/3/1 閲覧）
- 中川聡（2011）『グラフィックデザイナーのためのユニバーサルデザイン実践テクニック 51』ワークスコーポレーション.
- 中村直哲（2019）「第9章 小学校における音の学習を通したアルゴリズム学習」日本産業技術教育学会（編）『小・中・高等学校でのプログラミング教育実践—問題解決を目的とした論理的思考力の育成—』九州大学出版会，pp. 118-123.
- ナハマノヴィッチ，ステイーヴン（2014）『フリープレイヤー人生と芸術におけるインプロヴィゼーション』若尾裕（訳），フィルムアート社.
- 鳴門教育大学プログラミング教育研究会（編）（2021）『今こそ知りたい！学び続ける先生のための基礎と実践から学べる小・中学校プログラミング教育』ジアース教育新社.
- 新実徳英（監）（2016）『小学音楽 音楽のおくりもの』3，教育出版.
- 新実徳英（監）（2020）『小学音楽 音楽のおくりもの』1-6，教育出版.



- 西下義之 (2016) 「小学校図工科におけるプログラミングの活用—1年生の取り組みを通して—」『情報処理』57巻12号, pp. 1232-1234.
- 日本経済新聞 (2016-04-19) 「プログラミング必修化, 政府方針, 小中学校で20年から。」日本経済新聞 2016年4月19日朝刊.
- 日本経済新聞 (2022-02-15) 「学校パソコン, もう返したい, 1人1台ばらまき先行, 教師なお『紙と鉛筆』(InsideOut いまを解き明かす)」日本経済新聞 2022年2月15日朝刊.
- 日本産業技術教育学会 (編) (2019) 『小・中・高等学校でのプログラミング教育実践—問題解決を目的とした論理的思考力の育成—』九州大学出版会.
- 登本洋子・高橋純 (2022) 「初等中等教育における情報端末の整備と活用に関する教員の意識」『日本教育工学会論文誌』45巻3号, pp. 365-373.
- 林向達 (2018) 「Computational Thinking に関する言説の動向」『日本教育工学会研究報告集』18巻2号, pp. 165-172.
- 原田恵理子・渡邊健治・西村明・Kenneth, J. Mackin・朴鍾杰 (2018) 「教員養成におけるプログラミング教育の教育体制に関する研究」『東京情報大学研究論集』21巻2号, pp. 135-143.
- 原田秀司 (2019) 『UI デザインの教科書 [新版] マルチデバイス時代のインターフェイス設計』翔泳社.
- 原田康徳 (2009) 「体験型ワークショップ用ソフトウェアの開発」『第50回プログラミング・シンポジウム予稿集』2009巻, pp. 163-168.
- 原田康徳・勝沼奈緒実・久野靖 (2014) 「公立小学校の課外活動における非専門家によるプログラミング教育」『情報処理学会論文誌』55巻8号, pp. 1765-1777.
- 原田康徳 (2016) 「小学生に分かるコンピュータサイエンスとしてのプログラミング教育—ビスケットを用いて—」『情報処理: 情報処理学会誌』57巻4号, pp. 344-348.
- 原田悠我・荒優・山内祐平 (2017) 「プログラミング学習における Tinkering の支援—自己説明を通じた仕組みの理解を促すシステムの試作—」『日本教育工学会研究報告集』17巻1号, pp. 325-332.
- バンジ, マッシモ・シロー, マイケル (2015) 『Arduino をはじめよう 第3版』船田巧 (訳), オライリー・ジャパン.
- 阪東哲也・黒田昌克・福井昌則・森山潤 (2017) 「我が国の初等中等教育におけるプログラミング教育の制度化に関する批判的検討」『兵庫教育大学学校教育学研究』30巻, pp. 173-184.
- 平井聡一郎・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』技術評論社.
- 平田繁・岡田充弘・木原美樹子・西村敬子・田村孝洋・白石恵里・野上俊一 (2020) 「小学校プログラミング教育の現状と教員養成における課題—B・C分類(国語, 社会, 生活, 音楽, 家庭, 体育, 図画工作, 外国語)—」『中村学園大学・中村学園大学短期大学部研究紀要』52号, pp. 193-203.
- 深見友紀子・小梨貴弘 (2019) 『[音楽指導ブック] 音楽科教育とICT』音楽之友社.
- 福島耕平・勝井まどか・下村勉 (2018) 「小学校音楽科におけるプログラミングソフト—Scratch を活用した旋律づくりの試み—」『コンピュータ&エデュケーション』45巻, pp. 61-66.
- 藤原伸彦・阪東哲也・曽根直人・長野仁志・山田哲也・伊藤陽介 (2019) 「ティンカリングとしてのプロ

- プログラミング』『鳴門教育大学情報教育ジャーナル』16巻, pp. 21-26.
- ベイリー, デレク (1981) 『インプロヴィゼーション』竹田賢一・木幡和枝・斉藤栄一 (訳), 工作舎.
- ベネッセ (2017) 「プログラミングを通じた音楽の授業実践例～小2音楽『村まつり』」インターネット,  
<https://benesse.jp/programming/beneprog/2017/11/02/%e3%83%97%e3%83%ad%e3%82%b0%e3%83%a9%e3%83%9f%e3%83%b3%e3%82%b0%e3%82%92%e9%80%9a%e3%81%98%e3%81%9f%e9%9f%b3%e6%a5%bd%e3%81%ae%e6%8e%88%e6%a5%ad%e5%ae%9f%e8%b7%b5%e4%be%8b%ef%bd%9e%e5%b0%8f2%e9%9f%b3/index.html> (2022/2/22 に閲覧)
- ベネッセ (2018a) 「図で解説『プログラミング的思考』とは」インターネット,  
<https://benesse.jp/programming/beneprog/2018/07/13/computationalthinking/> (2022/9/28 に閲覧)
- ベネッセ (2018b) 「第2版『プログラミングで育成する資質・能力の評価規準 (試行版)』 Ver.2.0.0 (2018/8/31版)」インターネット, <https://benesse.jp/programming/beneprog/wp-content/uploads/2018/08/ver2.0.0.pdf> (2022/9/13 に閲覧)
- ベネッセ (2018c) 「第2版『プログラミングで育成する資質・能力の評価規準 (試行版)』公開」インターネット, <https://benesse.jp/programming/beneprog/2018/08/31/2ndstandard/> (2022/9/13 に閲覧)
- ベネッセ (2018d) 「プログラミングを通じた音楽の授業実践例～小3音楽『まほうの音楽』」インターネット,  
<https://benesse.jp/programming/beneprog/2018/02/15/%e3%83%97%e3%83%ad%e3%82%b0%e3%83%a9%e3%83%9f%e3%83%b3%e3%82%b0%e3%82%92%e9%80%9a%e3%81%98%e3%81%9f%e9%9f%b3%e6%a5%bd%e3%81%ae%e6%8e%88%e6%a5%ad%e5%ae%9f%e8%b7%b5%e4%be%8b%ef%bd%9e%e5%b0%8f%ef%bc%93/index.html> (2022/2/22 に閲覧)
- ベネッセ (2019) 「ひょうしをかんじて, リズムうちにしたしもう | プロアonz」インターネット,  
[https://www.proanz.com/lesson\\_plans/763785ac6d6044a3b50e3df948aed79a/](https://www.proanz.com/lesson_plans/763785ac6d6044a3b50e3df948aed79a/) (2022/2/22 に閲覧)
- ベネッセ (n.d.) 「ベネッセのプログラミング教育情報」インターネット, <https://beneprog.com> (2018/9/2 に閲覧)
- ボズウェル, ダスティン・フォシェ, トレバー (2012) 『リーダブルコードーより良いコードを書くためのシンプルで実践的なテクニック』角征典 (訳), オライリー・ジャパン.
- 細谷滝音 (2017) 「アルゴリズムで踊れ! 君は演奏する“プログラミング” Live Codingを知っているか?」インターネット, <http://ascii.jp/elem/000/001/563/1563265/> (2018/11/12 に閲覧)
- 堀田龍也 (2016) 「なぜ小学校から必要なのか? プログラミング教育が目指すもの」『総合教育技術』71巻10号, pp. 44-47.
- 堀田龍也 (2017) 「小学校におけるプログラミング教育の本質」黒上晴夫・堀田龍也 (編) 『教育技術 MOOK 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』小学館, pp. 11-15.
- 外岡紘代 (2018) 「人気の習い事, プログラミングがダンス抜く」インターネット,  
<https://resemom.jp/article/2018/01/19/42389.html> (2018/9/28 に閲覧)

- 松田孝・吉田潤子・原田廉徳・久木田寛直・赤石先生・利根川裕太・國領二郎・デビドソン，サムエル  
(2017)『小学校の「プログラミング授業」実況中継 [教科別] 2020 年から必修のプログラミング教育はこうなる』技術評論社.
- 松村明 (監) (1995)『大辞泉』小学館.
- 丸岡慎弥 (2020)『結局，何を教えればいいかがスッキリわかる！小学校 はじめてのプログラミング授業』学陽書房.
- 水落芳明・齋藤博 (編) (2020)『これで，ICT 活用・プログラミング×「学び合い」は成功する！』学事出版.
- 宮城県総合教育センター (n.d.)「プログラミング教育スタートパック」インターネット，  
<http://www.edu-c.pref.miyagi.jp/midori/jouhou/pep/index.html> (2018/9/2 に閲覧)
- 未来の学びコンソーシアム (2018)「小学校プログラミング教育必修化に向けて」インターネット，  
[https://miraino-manabi.mext.go.jp/assets/data/info/miraino-manabi\\_leaflet\\_2018.pdf](https://miraino-manabi.mext.go.jp/assets/data/info/miraino-manabi_leaflet_2018.pdf) (2022/3/15 に閲覧)
- 未来の学びコンソーシアム (2020a)「『未来の学びコンソーシアム』事務局の活動内容 2020 年 4 月 16 日 第 7 回運営協議会 資料 4」インターネット，[https://miraino-manabi.mext.go.jp/assets/pdf/200416\\_siryu4.pdf](https://miraino-manabi.mext.go.jp/assets/pdf/200416_siryu4.pdf) (2022/3/16 に閲覧)
- 未来の学びコンソーシアム (2020b)「みらプロ 2019 の実施事例について」インターネット，  
<https://miraino-manabi.mext.go.jp/content/475> (2020/3/16 に閲覧)
- 未来の学びコンソーシアム (2020c)「協力企業と連携した総合的な学習の時間 みらプロ」インターネット，  
<https://mirapro.mext.go.jp> (2022/3/16 に閲覧)
- 未来の学びコンソーシアム (2020d)「未来の学びコンソーシアム 第 8 回運営協議会 議事概要」インターネット，  
<https://miraino-manabi.mext.go.jp/sites/default/files/20201218gaiyou.pdf> (2020/3/16 に閲覧)
- みんなのコード (2018)「【プログラミング教育×英語の指導案】英語×Scratch で楽器を表す単語を楽しく覚えよう」インターネット，  
[https://procurri.jp/2018/01/29/lesson\\_plan\\_english\\_scratch/](https://procurri.jp/2018/01/29/lesson_plan_english_scratch/)  
(2022/2/22 に閲覧)
- 森脇正人 (2017)「小学校音楽科の音楽づくりの内容における，プログラミング教育導入の試み」『情報教育シンポジウム論文集』2017 巻 38 号，pp. 231-238.
- 森脇正人 (2018)「小学校音楽科の，音楽づくりの内容におけるプログラミング教育の実践～日本の音楽に親しむ学習を通して～」『情報教育シンポジウム論文集』2018 巻 4 号，pp. 23-29.
- 森脇正人 (2019)「日本の音楽に親しむ学習の音楽づくりにおける，二つのプログラミング言語の活用と比較」『情報教育シンポジウム論文集』2019 巻，pp. 234-241.
- 文部科学省 (2008)「中学校学習指導要領 平成 20 年 3 月 平成 22 年 11 月一部改正」インターネット，  
<https://erid.nier.go.jp/files/COFS/h19j/index.htm> (2022/3/14 に閲覧)
- 文部科学省 (2009)「高等学校学習指導要領 平成 21 年 3 月 文部科学省」インターネット，  
<https://erid.nier.go.jp/files/COFS/h20h/index.htm> (2022/3/14 に閲覧)

文部科学省（2015a）「情報教育指導力向上支援事業（諸外国におけるプログラミング教育に関する調査研究）報告書」インターネット、  
[https://www.mext.go.jp/a\\_menu/shotou/zyouhou/detail/\\_\\_icsFiles/afieldfile/2018/08/10/programming\\_syogaikoku\\_houkokusyo.pdf](https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/__icsFiles/afieldfile/2018/08/10/programming_syogaikoku_houkokusyo.pdf)（2022/2/9 に閲覧）

文部科学省（2015b）「学校段階等別部会及び教科等別ワーキンググループ等の設置について」インターネット、  
[https://www.mext.go.jp/b\\_menu/shingi/chukyo/chukyo3/056/siryo/\\_\\_icsFiles/afieldfile/2015/10/29/1363262\\_5.pdf](https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/056/siryo/__icsFiles/afieldfile/2015/10/29/1363262_5.pdf)（2022/3/17 に閲覧）

文部科学省（2016a）「教育課程部会 情報ワーキンググループ 議事要旨・議事録・配付資料」インターネット、[https://www.mext.go.jp/b\\_menu/shingi/chukyo/chukyo3/059/giji\\_list/1364461.htm](https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/059/giji_list/1364461.htm)（2022/3/22 に閲覧）

文部科学省（2016b）「プログラミング教育実践ガイド」インターネット、  
[http://jouhouka.mext.go.jp/school/programming\\_zirei/](http://jouhouka.mext.go.jp/school/programming_zirei/)（2016年12月25日に閲覧）

文部科学省（2016c）「プログラミング教育実践ガイド」インターネット、  
[http://jouhouka.mext.go.jp/school/pdf/programing\\_guide.pdf](http://jouhouka.mext.go.jp/school/pdf/programing_guide.pdf)（2017/12/24 に閲覧）

文部科学省（2018a）『小学校学習指導要領（平成29年告示）』東洋館出版社。

文部科学省（2018b）『小学校学習指導要領（平成29年告示）解説 総則編』東洋館出版社。

文部科学省（2018c）「小学校プログラミング教育の手引（第一版）」インターネット、  
[https://www.mext.go.jp/content/20200214-mxt\\_jogai02-000004962\\_004.pdf](https://www.mext.go.jp/content/20200214-mxt_jogai02-000004962_004.pdf)（2022/10/16 に閲覧）

文部科学省（2018d）「小学校プログラミング教育の手引（第二版）」インターネット、  
[http://www.mext.go.jp/a\\_menu/shotou/zyouhou/detail/1403162.htm](http://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1403162.htm)（2018/11/15 に閲覧）

文部科学省（2018e）『小学校学習指導要領（平成29年告示）解説 音楽編』東洋館出版社。

文部科学省（2019）「教育の情報化に関する手引（令和元年12月）第1章～第3章」インターネット、  
[https://www.mext.go.jp/content/20200609-mxt\\_jogai01-000003284\\_002.pdf](https://www.mext.go.jp/content/20200609-mxt_jogai01-000003284_002.pdf)（2022/3/26 に閲覧）

文部科学省（2020a）「小学校プログラミング教育の手引（第三版）」インターネット、  
[https://www.mext.go.jp/content/20200218-mxt\\_jogai02-100003171\\_002.pdf](https://www.mext.go.jp/content/20200218-mxt_jogai02-100003171_002.pdf)（2021/3/15 に閲覧）

文部科学省（2020b）「未来の学びコンソーシアムの業務終了およびポータルサイト移管のお知らせ」インターネット、<https://miraino-manabi.mext.go.jp/content/516>（2022/3/15 に閲覧）

文部科学省（2020c）「GIGA スクール構想の実現 標準仕様書」インターネット、  
[https://www.mext.go.jp/content/20200303-mxt\\_jogai02-000003278\\_407.pdf](https://www.mext.go.jp/content/20200303-mxt_jogai02-000003278_407.pdf)（2022/3/1 に閲覧）

文部科学省（2021）「小学校プログラミング教育における民間企業による指導案等の提供について」インターネット、<https://miraino-manabi.mext.go.jp/content/518>（2022/3/16 に閲覧）

文部科学省（2022a）「GIGA スクール構想の実現について」インターネット、  
[https://www.mext.go.jp/a\\_menu/other/index\\_00001.htm](https://www.mext.go.jp/a_menu/other/index_00001.htm)（2022/3/3 に閲覧）

文部科学省（2022b）「学校における小学校プログラミング教育の実施レポート（公開開始日：2020年11月9日）」インターネット、<https://miraino-manabi.mext.go.jp/content/507>（2022/3/16 に閲覧）

文部科学省教育課程部会（2016a）「次期学習指導要領等に向けたこれまでの審議のまとめについて（報告）」インターネット、  
[http://www.mext.go.jp/b\\_menu/shingi/chukyo/chukyo3/004/gaiyou/1377051.htm](http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/004/gaiyou/1377051.htm)（2016/8/26 閲覧）

文部科学省教育課程部会（2016b）「次期学習指導要領等に向けたこれまでの審議のまとめ（第1部）」インターネット、[https://www.mext.go.jp/content/1377021\\_1\\_1\\_11\\_1.pdf](https://www.mext.go.jp/content/1377021_1_1_11_1.pdf)（2022/3/23 閲覧）

文部科学省生涯学習政策局情報教育課（2015a）「平成27年10月22日教育課程部会情報ワーキンググループ資料8情報教育に関連する資料」インターネット、  
[https://www.mext.go.jp/b\\_menu/shingi/chukyo/chukyo3/059/siryo/\\_icsFiles/afieldfile/2015/11/11/1363276\\_08\\_1.pdf](https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/059/siryo/_icsFiles/afieldfile/2015/11/11/1363276_08_1.pdf)（2022/3/18 閲覧）

文部科学省生涯学習政策局情報教育課（2015b）「教育課程部会 情報ワーキンググループ（第1回）議事録」インターネット、  
[https://www.mext.go.jp/b\\_menu/shingi/chukyo/chukyo3/059/siryo/1382051.htm](https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/059/siryo/1382051.htm)（2021/3/17 閲覧）

文部科学省生涯学習政策局情報教育課（2015c）「教育課程部会 情報ワーキンググループ（第2回）議事録」インターネット、  
[https://www.mext.go.jp/b\\_menu/shingi/chukyo/chukyo3/059/siryo/1382052.htm](https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/059/siryo/1382052.htm)（2021/3/17 閲覧）

文部科学省生涯学習政策局情報教育課（2016a）「情報ワーキンググループにおける審議の取りまとめ」インターネット、  
[https://www.mext.go.jp/b\\_menu/shingi/chukyo/chukyo3/059/sonota/\\_icsFiles/afieldfile/2016/09/12/1377017\\_1.pdf](https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/059/sonota/_icsFiles/afieldfile/2016/09/12/1377017_1.pdf)（2022/12/28 閲覧）

文部科学省生涯学習政策局情報教育課（2016b）「教育課程部会 情報ワーキンググループ（第8回）議事録」インターネット、  
[https://www.mext.go.jp/b\\_menu/shingi/chukyo/chukyo3/059/siryo/1382093.htm](https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/059/siryo/1382093.htm)（2022/3/17 閲覧）

文部科学省初等中等教育局学校デジタル化プロジェクトチーム（2019）「次世代の教育情報化推進事業『情報教育の推進等に関する調査研究』」インターネット、  
[https://www.mext.go.jp/a\\_menu/shotou/zyouhou/detail/1400796.htm](https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1400796.htm)（2022/3/31 閲覧）

文部科学省初等中等教育局学校デジタル化プロジェクトチーム（2020）「教育の情報化に関する手引 一追補版一（令和2年6月）第3章 プログラミング教育の推進」インターネット、  
[https://www.mext.go.jp/content/20200608-mxt\\_jogai01-000003284\\_004.pdf](https://www.mext.go.jp/content/20200608-mxt_jogai01-000003284_004.pdf)（2022/3/18 閲覧）

文部科学省初等中等教育局教育課程課（2016a）「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 平成28年4月19日 初等中等教育局長決定」インターネット、  
[https://www.mext.go.jp/b\\_menu/shingi/chousa/shotou/122/attach/\\_icsFiles/afieldfile/2016/05/06/13](https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/_icsFiles/afieldfile/2016/05/06/13)

- 70404\_1.pdf (2021/3/15 に閲覧)
- 文部科学省初等中等教育局教育課程課 (2016b) 「教育課程の改善に向けた検討状況」 インターネット,  
[https://www.mext.go.jp/b\\_menu/shingi/chousa/shotou/122/shiryo/\\_icsFiles/afiedfile/2016/05/13/1370860\\_004.pdf](https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/shiryo/_icsFiles/afiedfile/2016/05/13/1370860_004.pdf) (2022/3/31 に閲覧)
- 文部科学省初等中等教育局教育課程課 (2017a) 「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 (第 1 回) 議事録」 インターネット,  
[https://www.mext.go.jp/b\\_menu/shingi/chousa/shotou/122/gijiroku/1382201.htm](https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/gijiroku/1382201.htm) (2022/2/4 に閲覧)
- 文部科学省初等中等教育局教育課程課 (2017b) 「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 (第 2 回) 議事録」 インターネット,  
[https://www.mext.go.jp/b\\_menu/shingi/chousa/shotou/122/gijiroku/1382200.htm](https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/gijiroku/1382200.htm) (2022/2/4 に閲覧)
- 文部科学省初等中等教育局教育課程課 (2017c) 「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 (第 3 回) 議事録」 インターネット,  
[https://www.mext.go.jp/b\\_menu/shingi/chousa/shotou/122/gijiroku/1382219.htm](https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/gijiroku/1382219.htm) (2022/2/4 に閲覧)
- 文部科学省初等中等教育局情報教育・外国語教育課 (2018a) 「コンソーシアム組織 | 未来の学びコンソーシアム」 インターネット, <https://miraino-manabi.mext.go.jp/content/139> (2022/2/21 に閲覧)
- 文部科学省初等中等教育局情報教育・外国語教育課 (2018b) 「動物が楽しく踊るリズムループをつくろう | 未来の学びコンソーシアム」 インターネット, <https://miraino-manabi.mext.go.jp/content/264> (2022/2/14 に閲覧)
- 文部科学省初等中等教育局情報教育・外国語教育課 (2019) 「くりかえしをつかってリズムをつくろう | 未来の学びコンソーシアム」 インターネット, <https://miraino-manabi.mext.go.jp/content/265> (2022/2/14 に閲覧)
- 文部科学省初等中等教育局情報教育・外国語教育課 (n.d.) 「小学校を中心としたプログラミング教育ポータル Powered by 未来の学びコンソーシアム」 インターネット, <https://miraino-manabi.mext.go.jp> (2022/2/10 に閲覧)
- 文部科学省初等中等教育局初等中等教育企画課 (2019) 「GIGA スクール実現推進本部の設置について」 インターネット, [https://www.mext.go.jp/content/20191219-mxt\\_syoto01\\_000003363\\_08.pdf](https://www.mext.go.jp/content/20191219-mxt_syoto01_000003363_08.pdf) (2022/3/1 に閲覧)
- 文部省 (1970) 「高等学校 学習指導要領」 インターネット,  
<https://erid.nier.go.jp/files/COFS/s45h/index.htm> (2022/3/14 に閲覧)
- 文部省 (1978) 「高等学校学習指導要領 昭和 53 年 (1978) 改訂版」 インターネット,  
<https://erid.nier.go.jp/files/COFS/s53h/index.htm> (2022/3/14 に閲覧)
- 文部省 (1989) 「中学校学習指導要領 付学校教育法施行規則 (抄)」 インターネット,  
<https://erid.nier.go.jp/files/COFS/h01j/index.htm> (2022/3/14 に閲覧)

- 文部省 (1998) 「中学校学習指導要領 付学校教育法施行規則 (抄)」 インターネット,  
<https://erid.nier.go.jp/files/COFS/h10j/index.htm> (2022/3/14 に閲覧)
- 文部省 (1999) 「高等学校学習指導要領 付学校教育法施行規則 (抄) 中等教育学校等関係法令 (抄)」 インターネット, <https://erid.nier.go.jp/files/COFS/h10h/index.htm> (2022/3/14 に閲覧)
- 安田一平 (2019) 「小学校音楽科におけるプログラミング言語学習環境を活用した授業実践—子どもの声をブロックにして—」 『日本デジタル教科書学会発表予稿集』 Vol. 8, pp. 65-66.
- ヤマハ (n.d.) 「ICT を活用した音楽授業のデジタル教材 | ボーカロイド教育版」 インターネット,  
[https://ses.yamaha.com/products/vocaloid\\_edu/](https://ses.yamaha.com/products/vocaloid_edu/) (2022/2/14 に閲覧)
- 山本大地・本多佑希・島袋舞子・兼宗進 (2019) 「ブロックプログラミング環境を用いたプログラム入力補助機能の提案」 『情報教育シンポジウム論文集』 2019 巻, pp. 309-312.
- 山本利一・中村茉耶・竹澤則乃・軽部禎文 (2020) 「小学校プログラミング教育に関する研修内容の提案と教員の課題意識」 『教育情報研究』 36 巻 2 号, pp. 29-42.
- 山本朋弘・堀田龍也 (2021) 「小学校段階におけるプログラミング教育の変遷と現状」 『中村学園大学発達支援センター研究紀要』 13 号, pp. 36-43.
- 柚木翔一郎・片平克弘 (2015) 「ティンカリングの観点を取り入れた生徒主体の『ものづくり』に関する研究」 『日本科学教育学会研究会研究報告』 30 巻 6 号, pp. 51-54.
- 吉田葵・阿部和広 (2017) 『日本標準ブックレット No. 18 はじめよう！プログラミング教育—新しい時代の基本スキルを育む—』 日本標準.
- 読売新聞 (2022-10-11) 「学習端末『よく机から落ちる』『こんなに壊れるとは』…自治体にのしかかる修理費」 インターネット, <https://www.yomiuri.co.jp/national/20221008-OYT1T50045/> (2022/10/11 に閲覧)
- レズニック, ミッチェル・村井裕実子・阿部和広 (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』 酒匂寛 (訳), 日経 BP 社.
- レッドブル・ミュージック・アカデミー (n.d.) 「ALGORAVE: ライブコーディング+ダンス？」 インターネット, <http://www.redbullmusicacademy.jp/jp/magazine/algorithm-dancing-to-live-coding> (2019/9/9 に閲覧)
- ローズ, カーチス (2001) 『コンピュータ音楽 歴史・テクノロジー・アート』 青柳龍也・小坂直敏・後藤真考・引地考文・平田圭二・平野砂峰旅・堀内靖雄・松島俊明 (訳), 東京電機大学.
- ロボ団 (2020) 「プログラミング的思考を育むには? ~自分の力で考え抜くことが大切~」 インターネット, [https://robo-done.com/blog/2020/05/honbu\\_programming-thinking02/](https://robo-done.com/blog/2020/05/honbu_programming-thinking02/) (2022/9/28 に閲覧)
- 渡辺勇士・中山佑梨子・原田康徳・久野靖 (2021) 「幼稚園児のビスケットプログラムにおける繰り返し続けるプログラムの理解の分析」 『情報処理学会論文誌教育とコンピュータ (TCE)』 7 巻 1 号, pp. 38-49.