

プログラミング教育について
—情報リテラシーの涵養か ICT 人材育成か—

本田敏明*・赤間文香**

(2016年11月1日受理)

About programming education
: Education for information literacy or for ICT human resource development

Toshiaki HONDA* and Fumika AKAMA**

(Accepted November 1, 2016)

はじめに

日本政府は、平成28年4月19日に2020年度から初等中等教育段階プログラミング教育を必修化する方針を明らかにした。世界に目を向けると韓国で2015年3月から中学校、2017年3月から小学校の正規教育課程としてプログラミング教育を導入、フィンランドでは2016年から義務教育期間である基礎学校でプログラミング教育が導入された。今日においてプログラミング教育は世界中で注目され、教育の現場に導入され始めている。

現在行われているプログラミング教育は、プログラミングを通して現代テクノロジーそのものよりもその背後にある考え方や思考力を養うことを重点においた教育(以下、「情報リテラシー教育型」とする)と、SEなどITエンジニア系の即戦力となる人材を育成することに重点を置いた教育(以下、「ICT人材育成型」とする)に大きく2つに分けることができる。

上述のように日本もプログラミング教育を導入することが明らかにされ、様々な政策が始まりつつあるが、義務教育段階における日本のプログラミング教育への取り組みは未だ明確になっていない部分も見られる。常に変化していく今日のIT社会に対応していくことができるようになるには、情報リテラシー型とICT人材育成型どちらのプログラミング教育がよいのであろうか。

本論文では、義務教育段階における日本のプログラミング教育は情報リテラシー教育型のプログラミング教育、つまり情報リテラシーを涵養することが好ましいとの立場から、その可能性について考察と提案を行いたい。

*茨城大学教育学部 (〒310-8512 水戸市文京 2-1-1 ; College of Education, Ibaraki University, Mito 310-8512 Japan).

**茨城大学教育学部情報文化課程 (〒310-8512 水戸市文京 2-1-1 ; Course for Information and Culture ,College of Education, Ibaraki University, Mito 310-8512 Japan).

1 プログラミング教育

IT用語辞典 e-Words によると、プログラミングとは、コンピューターに人間が意図した動作を行わせるための指示の集まり（プログラム）を作成すること、狭義には、プログラミング言語を用いて人間に可読な形式で指示をまとめたソースコードを記述する作業（コーディング）のことを指す。広義には、ソースコードをコンピューターで実行可能な形式（オブジェクトコード）に変換したり、意図したとおりに動作するかテストしたり、不具合を見つけて修正したり（デバッグ）といった一連の作業を含む¹⁾。このようにプログラミングとは、コンピューターに人間が意図した動作を行わせるために、コンピューターが理解できる言語で順序立てて指示を出すことである。順序立てた指示を出し、意図した通りに動作するよう問題について考え、不具合を修正していくためには、論理的思考力や問題解決能力が必要になってくるのである。論理的思考力や問題解決能力はプログラミングを行うためだけに必要な能力ではない。情報や知識を整理する力、客観的な根拠に基づいて選択肢を挙げ目的に応じて判断したり実践する力、結果をフィードバックして軌道修正していく柔軟な発想力は何をするにしても重要な能力である。

そのためプログラミング教育では、実際にプログラミングを体験しながらプログラミングを行うために必須の論理的思考力や問題解決能力を涵養していくものであるが、プログラミングに必要なという枠に囚われず、現代テクノロジーの背後にある考え方や思考力を養うための教育であると位置づけることができるだろう。

1.1 プログラミング教材について

上述したようにプログラミング教育とは実際にプログラミングを体験しながら、その背後にある考え方や思考力を養うための教育であるとする。そのため本章では、個別のプログラミング言語を用いてプログラムをテキストで記述していく教材ではなく、視覚的なオブジェクトでプログラミングする「ビジュアルプログラミング言語」を用いた教材として代表的かつインターネット環境があればどこでも入手可能な以下の4つのプログラミング教材について見ていく。

1.1.1 Viscuit（ビスケット）²⁾

Viscuit は原田康徳によって子ども向けプログラミング教育の学習環境として2003年2月に開発された。ブロック型ビジュアル言語³⁾を用いてその組み合わせでプログラムを作るものではなく、変数や配列等の概念も持たない。書き換えの規則に従って、絵の配置を書き換えていく図形書き換え言語としてのビジュアルプログラミング言語である。絵の配置を書き換える規則を与え、対象を書き換えていく様子をアニメーションで見る⁴⁾といった内容で、手書きイラストを用いてプログラムを作るため幼い子どもでもアニメーションやゲームのプログラムを作成することが出来る。

描画ツール内で描かれた絵がViscuitのプログラム内での「部品」となり、部品を書き換え規則が定義された「めがね」に入れることで、書き換え規則が作られ実行画面に表示される。（図1）

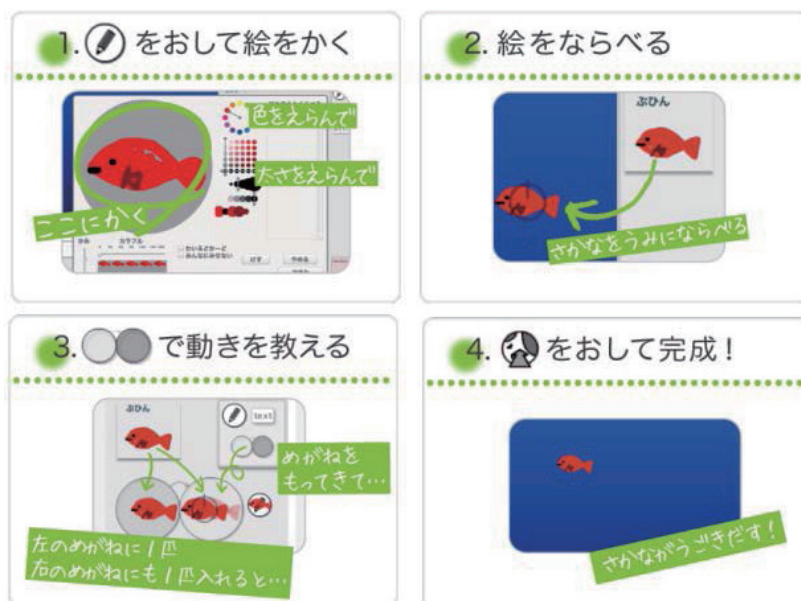


図1 Viscuit の操作方法⁵⁾

Viscuit の目的はプログラミングの技術を習得することではなく、「コンピュータとは何か」ということへの理解を深めていくことにある。したがって Viscuit は上述したようにブロック型のビジュアルプログラミング言語を用いてコードを入力してプログラムを作成していくものではない。ブロック型ビジュアルプログラミング言語の多くは現在主流のテキスト言語を用いたプログラムの入門教材としての学習環境としての一面を備えているが Viscuit においては、自分で描いたイラストを思い通りに動かすために、部品や配置を選択し実行し不具合を調整していくことの繰り返しの中でプログラムの動きやロジックの考え方に触れていくこと、直観的な操作の中でコンピュータを楽しむ「コンピュータに対する理解」を育てていくことを重要な目的として位置付けている。

1.1.2 Scratch (スクラッチ)

Scratch は、2006 年に最初のバージョンが MIT メディアラボの Mitchel Resnick が主導するライフロング・キンダーガーデン・グループが Squeak Etoys (スクイーク)⁶⁾ をベースに開発した、教育用ビジュアルプログラミング開発環境である。8 歳以上が主な対象年齢であり、さらに低年齢 (5 歳～7 歳) 向けに Scratch Jr. も開発されている。ブロック型プログラミング言語を用いることで正しい構文の書き方を知らなくても、視覚的にブロックをくみ上げることでプログラミングを体験することが出来る。Scratch の開発環境のユーザインターフェイスはプログラムが出力されるステージ (背景) とスプライト⁷⁾ のリスト、ブロックパレット、スクリプトエリアと複数の枠に分かれており、ブロックパレットから必要な命令ブロックをスクリプトエリアにドラックし、プログラムを作成していく (図 2)。また命令ブロックが種類ごとに色や形が設定されているため視覚的にプログラムの処理の流れを把握しやすいようになっている。(図 3)

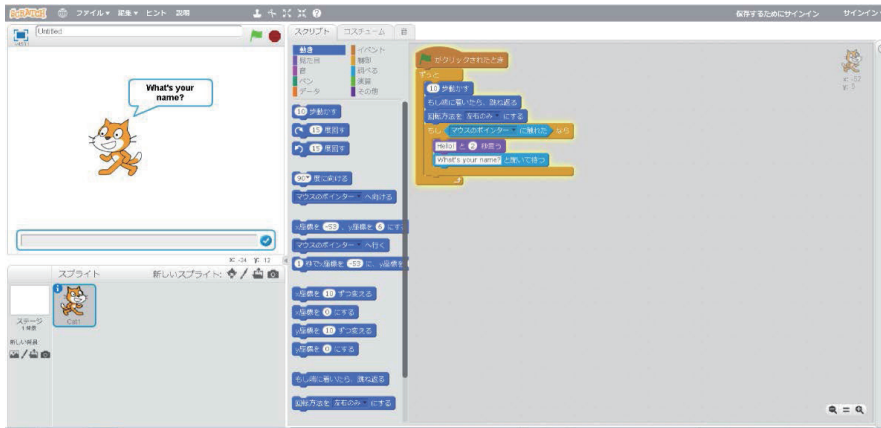


図2 Scratch 開発環境のユーザインターフェイス

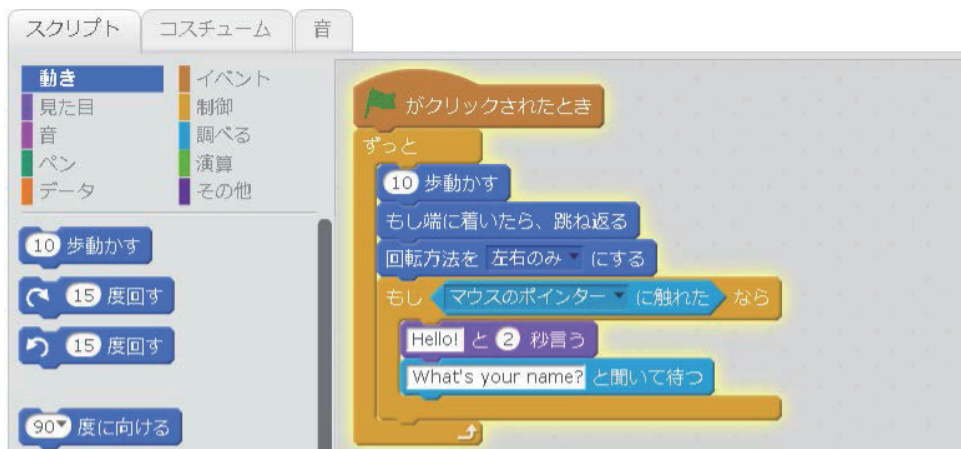


図3 Scratch のブロック型ビジュアルプログラミング言語

Scratch のオンラインコミュニティのスローガンは「Imagine, Program, Share（想像・プログラム・共有）」であり、Scratch を用いたプログラミング体験の背後の共有と創造性の社会的背景のことを指している⁸⁾。そのため Scratch のウェブサイト上には、メンバーが自身のプログラミングしたプロジェクトをアップロードすることができ、興味をもった他のプロジェクトのソースコードをダウンロードできるコミュニティが設置されている。Scratch を用いることでコンピュータ処理の仕組みやプログラミング技術について学ぶだけでなく、学ぶ技術を学ぶ、他者のアイデアから学び、創造性を発揮して新たなプロジェクトに挑戦していくためのサイクル基盤が存在する。

1.1.3 プログラミン

プログラミンは文部科学省がプログラミングの啓蒙コンテンツとして Scratch を参考に開発し公

開した，Web ブラウザで動作するプログラミング環境である。あらかじめ用意されている 28 種類の命令ブロック「プログラミン」を組み合わせてプログラムを作成しアニメーションやゲームを作ることができる。

Scratch のような分割されたユーザインターフェイスではなく，一画面にまとまっている。また命令ブロックを組み上げていくブロック型ビジュアルプログラミング言語ではあるが，プログラミンは命令ブロック「プログラミン」自体がイラスト化されており，プログラムを組み上げていくこと自体がインタラクティブな体験になるように作られている（図4）（図5）。



図4 プログラミンの画面



図5 命令ブロック「プログラミン」(一部)⁹⁾

文部科学省によると¹⁰⁾「プログラミン」は，プログラムを通じて，子どもたちに創ることの楽しさと，方法論を提供することを目的としたものである。プログラミングの基本となる概念に自発的に子どもが触れ，プログラムという表現方法を知り，自らの想像力をもって創作したりそのための

行動を起こしたり、発信する力を育むきっかけになることを想定している。

プログラミングで作成したプログラムは URL を作成しメール、ブログパーツ、Twitter 経由で公開することができるようになっているが、Scratch のような利用者同士の交流を作り出す一覧ページやソーシャルな機能はプログラミング自体には用意されていない。このことに関してユーザー登録や利用規約への同意などの手続きが必要とされるソーシャル機能をあえて付随させず、利用者の利便性を優先したものであると考えられる。また、プログラミング内に交流の場は設置されていないが、有志によるまとめサイト「プログラミング作品ギャラリー」から他者のプログラムを閲覧したり交流を図ることができるようになっている。

1.1.4 CODE.org

CODE.org は Ali Partovi と Hadi Partovi 兄弟によって 2013 年に設立された非営利団体および同名のウェブサイトである。サイト内の Code Studio からユーザーの年齢、習熟度に合ったコースを選択し、プログラミングレッスンを受けることができる。Scratch のような分割されたユーザーインターフェイスであり、実行画面、命令ブロックのパレット、ワークスペースがある。命令ブロックは課題に必要とされるものと複数のダミーブロックだけが用意されているため、その中から組み合わせることで課題を達成するプログラミングを組み上げることになる。また、1つの課題を達成したら次の課題へと進んでいくため、繰り返し学習のドリルのような教材である（図 6）。また、課題達成のためのヒント動画や達成後にコメントが表示されるなど継続して学習できるような学習教材としての面が強く見られる（図 7）。

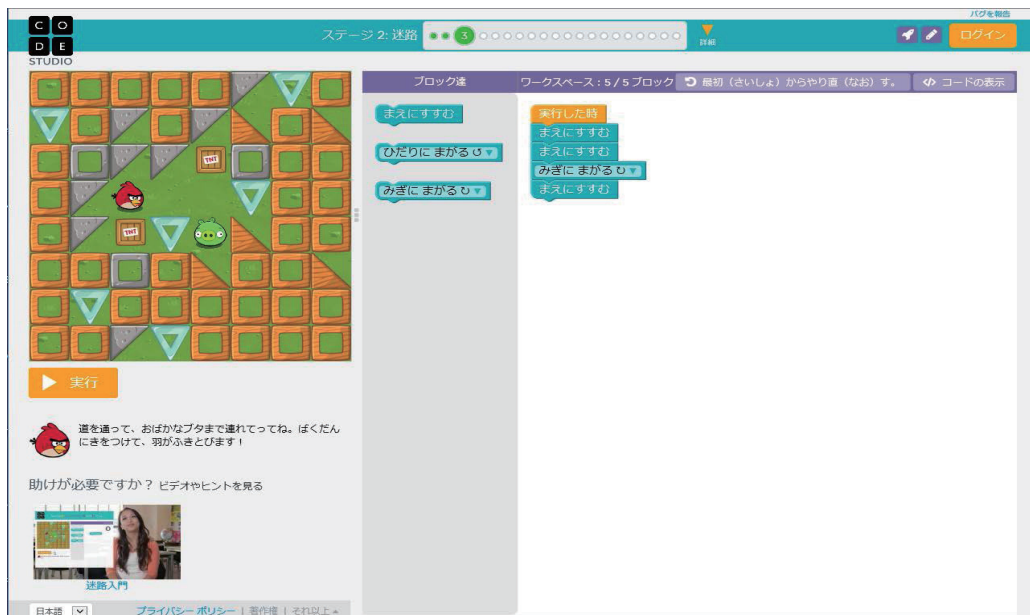


図 6 Code Studio のユーザーインターフェイス

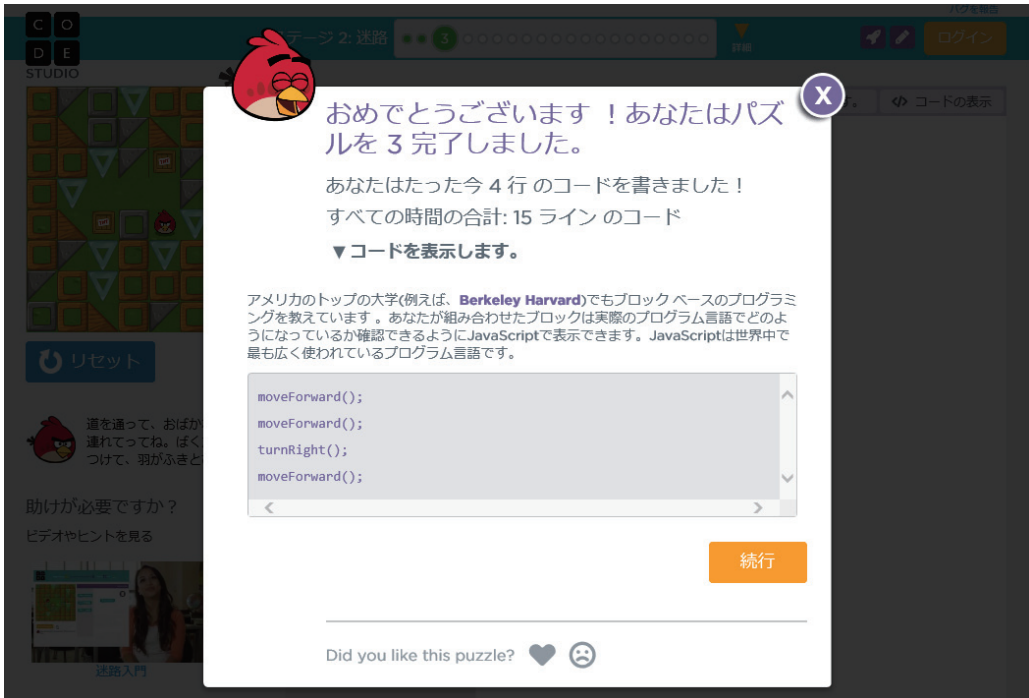


図7 課題達成後のコメントポップアップ

CODE.org はすべての人々がプログラミングできるようになるようにすることを目的として設立された。その中でも、K-12¹¹⁾を対象とした全米の教育カリキュラムにプログラミングを組み込むことをミッションとして挙げている。そのため、サイト内には教育者を対象としたページやサービス¹²⁾も用意されている。CODE Studio 内にはユーザーオリジナルのプログラムを公開できる場が設置されており、そこから他のユーザーがアップロードしたプログラムを見たり、リミックスを作成することができるようになっている。この点において CODE.org の交流環境は Scratch に近いものがあるといえる。

1.2 プログラミング教材についての考察

上述したように4つすべてはビジュアルプログラミング言語であるが、言語のビジュアル化には差があり、そこからプログラミングの自由度も差があると言えるのである。

Viscuit は自らが描いたイラストがプログラムの部品となり、部品の組み合わせによってアニメーションやゲームを作成することができる。そこにコードを組み上げる作業はない。実行した結果から試行錯誤して理想に近付けていく。また最も偶発的な動きが発生しやすく、偶然出来上がった結果から新しい想像を広げていくことができる。教材側から使用する言語の設定や明確なゴールの設置がなされていないため、紹介した4つの教材の中では最も自由度が高い教材であるといえるため、未就学時期や小学校低学年段階¹³⁾において現代テクノロジーに興味をもつ導入として使用し易いと考えられる。

Scratch は命令ブロックを組み合わせてプログラムしていく。プログラミング言語がブロック化されているため組んだプログラムはチャート化する。そのためプログラムの処理がどのように流れていくかがわかりやすくなっている。また言語数が多く用意されているため、使い方を理解したら実行できるプログラムの自由度は高い。ただし、命令ブロックは命令内容を理解するためある程度の読解力が必要とされるため、対象となっている8歳未満の子どもが使用するのには難しいだろう。よって Scratch は義務教育段階に入ってからでの使用が望ましく、プログラミングの中で論理的に思考する感覚や実行結果からのフィードバックする力を育めるだろう。

プログラミンは、命令ブロックを選択する段階でのブロックのビジュアルが記号化されているため実行したい動きに必要なブロックがわかりやすく、説明文を読むのが難しい時期においてもある程度直感的に操作することが可能であるだろう。プログラミンの特徴として、プログラムを下から上に組み上げていく点が挙げられる。ビジュアル・テキスト問わず多くのプログラミング言語は上から下にプログラムを組んでいくため、プログラミンの組み上げ方式のプログラミングは他の教材・プログラムに慣れた者からすると違和感を覚え、プログラミンに慣れ親しんだ者は他とのギャップに驚くのではないだろうか。また、プログラミンに設定された言語数は28種類と比較的少数であり、実行可能なプログラムの自由度は Viscuit や Scratch に比べて狭まるだろうが、初めてプログラミングする段階においては取り組みやすいともいえる。

そのため、プログラミンは未就学時期から使用でき、プログラミングとはどのような作業をしていくのかを概念的に知るための導入として取り入れることができると考えられる。

CODE.org は、Scratch と同様な命令ブロックを組み合わせてプログラムしていくが、すでに設定された課題を解決するためにプログラミングしていくためドリル教材の面が強い。ドリル教材的であると述べたが、チュートリアル課題を達成する過程を通してオリジナルのアニメーションやゲーム、デザイン画を作成することも可能である。難易度順に課題が設定されているため、複雑な動きを実行するプログラムの組み方も順序立てて考えやすくなる。明確な課題設定や教育者側へのサービスの充実から今回取り上げた4つのプログラミング教材の中で最も現在の日本の教育現場、授業に導入し易く ICT 人材育成型の教育に用いり易いだろう。

以上の内容を、教材の導入時期について対象年齢が明記されたものは◎、はっきりした明記が無い場合や対象年齢以降を○で表したものが表1であり、4つの教材をまとめたものが表2である。

表1 教材の導入時期

	未就学児	小学校低学年	小学校中学年	小学校高学年	中学校
Viscuit	◎	○	○	○	○
Scratch		◎	○	○	○
プログラミン	○	○	○	○	○
CODE.org	◎	○	○	○	○

表2 4つ教材のまとめ

教材	言語	概要
Viscuit	ビジュアル言語	<ul style="list-style-type: none"> ・手書きイラストがプログラミング言語になる。 ・言語数に制限がなく，創造力次第で様々な部品を作成し使用することができる ・プログラムを自由に配置するためプログラムの処理の流れを理解しにくい ・一方で偶発的な発見からその結果に至った段階を思考し試行錯誤する機会が多い
Scratch	ブロック型 ビジュアル言語	<ul style="list-style-type: none"> ・ブロックの組み合わせによってプログラミングするオブジェクト指向言語 ・ブロックを組み合わせることでプログラムを組んでいくためプログラミングの処理の流れが理解しやすい ・ユーザインターフェイスがスプライトや背景ごとに用意されているため何のプログラミングをしているか把握しやすい ・作成したプログラムを公開や他のプログラムを閲覧し参考にする場が設けられている
プログラミン	ブロック型 ビジュアル言語	<ul style="list-style-type: none"> ・命令を選択する段階のブロックが記号化されているため，命令内容のある程度想像できる ・プログラミングするためのアクションを起こす度に音が鳴りブロックが瞬きするなどの工夫がある ・ユーザインターフェイスが分割されておらず，実行対象の傍にプログラムが表示されるためプログラムと対象の関係把握は容易であるが画面は混雑する ・外部へのプログラムの公開はメール，ブログパーツ，Twitter 経由
CODE.org	ブロック型 ビジュアル言語	<ul style="list-style-type: none"> ・ブロックの組み合わせによってプログラミングするオブジェクト指向言語 ・ドリル形式で課題達成を繰り返していくことで順序立ててプログラミングの考え方や技術を習得していく ・作成したプログラムを公開や他のプログラムを閲覧し参考にする場が設けられている ・明確な課題や指導，指導者用のサービスの用意がある

上記のように4つのプログラミング教材の相違点について述べてきたが，この4つに共通することは，テキスト言語の構文を知らずとも簡単な操作によってプログラムを組み上げることができるよう設定されている点である。これらはコンピュータを筆頭に現代テクノロジーの基礎が専門家

以外には理解できない難解なものではなく、むしろ単純なものであり、そこから発展させていくためには技術よりも自らの創造性や思考力が重要であることを示唆していると言える。また Scratch や CODE.org のコミュニティの設置から、外部に発信する経験や他者の思考に触れ自らにフィードバックし思考することも重要視していることがわかるだろう。

2 各国の取り組み

先述したように、プログラミング教育は世界各国で義務教育に相当する教育段階で導入され始めている。そのため本章では海外のプログラミング教育について、情報リテラシー型と ICT 人材育成型の教育に分けその傾向が強い国の取り組みについて取り上げる。

2.1 情報リテラシー型

情報リテラシー型のプログラミング教育とは、プログラミングを通して現代テクノロジーの背後にある考え方や思考力を養うことを重点においた教育である。即戦力的な ICT 人材になるためにプログラミング言語や技術を習練するのではなく、プログラミングを通してコンピュータやテクノロジーの仕組みや考え方を学び、開発されたテクノロジーを受動的に受け入れる一方的なユーザーになるのではなく、開発の視点に立った問題解決能力を身に着けるための学習と位置付けられ、学習の中に組み込まれていく。また現代テクノロジーの開発側に立った理解のみでなく、ICT 機器やインターネットを利用するために必要なリテラシーの教育に重点を置いている。

2.1.1 イギリス（イングランド）

本項ではイギリス国の4地域（イングランド、ウェールズ、スコットランド、北アイルランド）のうちイングランドにおけるプログラミング教育について取り上げる。

イングランドでは1999年から始まった従来の教科「ICT」を改め、2013年から「Computing」が設置され2014年から義務教育段階において導入されている。従来の「ICT」が、情報活用能力の習得、ICT機器やソフトの「使い方」を教えるICTリテラシーに偏った教科であったのに対して「Computing」はCS（Computer Science）、IT（Information Technology）、DL（Digital Literacy）の3分野で構成されている。その学習目的は文部科学省が公表した「諸外国におけるプログラミング教育に関する調査研究」によると、質の高いコンピュータ教育は、「コンピューショナルシンキング（Computational Thinking）」や、世の中を理解し、変えていくための創造性を、学習者に与える。教科「Computing」は、算数・数学、科学、「Design and Technology」と深く関わっており、自然と人工システムに対する洞察力を与える。教科「Computing」の中心はコンピュータサイエンスであり、その中で学習者は、「information and computation」の原理、デジタルシステムが動く仕組み、またそれらの知識をプログラミングを通して活用する方法を学ぶ。この知識と理解に基づき、学習者はプログラムやシステム、様々なコンテンツの創造に情報技術を活用できるようになる。教科「Computing」はまた、学習者がICTを利用して自己表現し、自らの考えを展開するという、将来の職域に適した、あるいはデジタルワールドへの積極的な参加ができるレベルのデジタルリテラシーを獲得できるようにする¹⁴⁾。

指導内容はナショナルカリキュラムに明記され、アルゴリズムの理解から始まりプログラムの作

成やネットリテラシー、情報リテラシーの涵養、創造的なプロジェクトへの取り組み等の内容・スキル・プロセスの学習や実行、理解の到達段階が学年区分¹⁵⁾ごとに設定されている。指導時間は一般的に初等教育段階で1時間/週(約30時間/年)程度、中等教育段階のKS3では1時間/週(約30時間/年)程度、KS4では少なくとも2時間/週程度である¹⁶⁾。「Computing」の指導者は、基本的に学級担任制の初等教育段階においては学級担任が、基本的に教科担任制の中等教育段階では専任の教員が指導を行っている。

上述したように、「Computing」はコンピュータサイエンスを中心にICTへの理解を深め、思考のプロセスを学び自己表現に活用することを目的としている。今日のIT社会においてICT機器の理解と活用は、国語や算数とおなじような基盤・必須教科であると位置付けられ、プログラミングは世界を理解するための道具(プラクティカルワーク)と考えているのである。そのため、イングランドのプログラミング教育は、プログラマーの育成のためにコーディング技術を学ぶものではなく、現代テクノロジーの理解やそれらを用いて問題解決や自己表現に取り組むための方法や思考プロセスを学ぶ教育であると言える。

2.1.2 フィンランド

フィンランドでは2016年から2019年にかけて基礎学校9年間の義務教育カリキュラムの中にプログラミングを順次導入することが決定され、同年秋から基礎学校の初等教育段階(1-6年生)でも教育が開始された。

プログラミング教育を導入した理由を2014年当時教育相特別顧問であったEsa Suominenは、フィンランドではITが生活に取り巻いており、全ての子どもがソフトウェアについて基礎的なことを学ぶ機会を与えるため。今やテクノロジーと生活は切っても切り離せず、コンピュータサイエンスに関する知識は世界を正しく知るために必要不可欠。特殊な技能ではなく、市民として一般的な知識になっていくはず。プログラマー育成に力を入れるというより、すべての人に機会を与えるのが目的¹⁷⁾であると述べている。

義務教育に導入されるプログラミングは独立した科目として設置されるのではなく、算数・数学を中心にすべての科目に横断的に取り入れられる。教育内容は、1-2年生ではコンピュータを用いずに正確な指示・伝達を行うための方法や論理的な思考について学習し、3-6年からコンピュータやタブレットの使用の開始、Scratchなどのビジュアルプログラミングを使用してゲームや動画を作る経験をする。7-9年生になるとアルゴリズムについて考え、最低1つ以上のコードを描く必要があるプログラミング言語を学ぶように設定されている¹⁷⁾。プログラミングが科目として新設されたのではないため授業時間が増えることはない。そのため指導者も新たな教員を配置するのではなく、学級担任制の1-6年生に対しては学級担任が、教科担任制の7-9年生には数学の教科担任が指導している¹⁸⁾。教員向けにプログラミングの基礎を学ぶためのサポート体制も整備されているが、教員に求められるのはプログラミングに対する専門知識ではなく、児童・生徒が自由に試行錯誤を繰り返しながら学べる環境を整え、児童・生徒が必要としているときに手を差し伸べ、ガイドすることである¹⁹⁾。

フィンランドではプログラミングとは英語や日本語のような自然言語やピペットのような実験道具あるいは画材のように問題解決や表現の1道具として捉えている。そのためフィンランドのプロ

プログラミング教育は、社会基盤の多くがソフトウェアに依存する現代においてプログラミングは21世紀のリテラシーであるため、その知識や思考方法（問題解決能力や論理的思考力）を学び、将来に必要なスキルを育て、それらを応用し多方面の分野で活躍できる人材の育成する教育であると言える。

2.2 ICT人材育成型

ICT人材育成型のプログラミング教育とは、現代社会の基盤を支えている情報技術を重要視し、専門的知識や熟練したコーディング技術を有し即戦力となりえるICT人材を育成することに重点を置いた教育である。ICT機器やインターネットを利用する際に必要なリテラシーからソフトウェアの開発に必要な知識や技術を段階的に学んだ現代のIT社会の中で成功を取めることができる人材を育成することで自国の経済成長や国際的な競争力を高めることを目的としている。

2.2.1 韓国

韓国で情報教育の導入は1970年代と比較的早いものであった。1987に文教部から「コンピュータ教育強化方案」が発表され、初等中等教育における情報教育が開始された。情報教育の方向性は現行に至るまでに、コンピュータリテラシーの習得を重視したり情報科学の概念や原理の理解を中心にしたリ、ICTリテラシーを重視したりと改定の度に大きく変化してきた。2005年から2007年において再び情報科学やコンピュータサイエンスが重要視され、この時期に韓国において本格的なプログラミング教育が導入された。2010年代に入るとコンピューテーショナルシンキングがあらゆる学問分野において必要な手法とされ、プログラミング教育についても実社会に役立てられることが意図されるようになった²⁰⁾。

2015年3月から前期中等教育段階（中学校）の正規教育課程カリキュラムに「ソフトウェア」が導入され、2017年3月からは初等教育段階（小学校）の正規教育課程に導入されることになっている。2018年には「情報」教科が「ソフトウェア」教科に改定されプログラミングは正規科目となる。

プログラミング教育を導入した理由は、韓国の産業の多くは情報技術によるものといわれているため、情報教育を体系的に行い、安定して専門技術者を養成することの必要性が頻繁に主張されてきた。現在でも国内のソフトウェア関連人材養成が急務とされており、情報教育を学んだ生徒の中から、情報分野の専門家を多く輩出することも、国家的見地からの大きな狙いとなっている。初等中等段階への導入の理由としては、大学教育段階からの養成では遅いこと、早い段階から情報技術に触れる機会を広く提供することで、生徒の可能性を広げるとともに、情報分野へと進む人材を拡大することがあげられている²¹⁾。

小学校に先立って導入された中学校のソフトウェアの教育内容はアルゴリズム原理の理解、ビジュアルプログラミングを用いてゲームを作成するといったプログラミングの活用やテキスト言語を用いた基礎的なコンピュータプログラムを作成し工作機器やロボットなどを作動させることなどが含まれている。2017年からの小学校指導内容では、遊びを中心にコンピュータ思考の理解やソフトウェアツールを活用した問題解決方法の学習、ソフトウェアコーディングの体験活動が設定されている。指導者は派遣された専門の講師が行うことになっているが、ソフトウェア教育が問題な

く運営するため、2018年までに全小学校教師の30%である600万人に職務教育を実施して、このうち6000人に深化研修を実施する。中学校の「情報」科目の教師と「情報・コンピュータ」の資格保有教師1800人以上の全員が深化研修を受けることになるとしている²²⁾

以上から韓国におけるプログラミング教育は、国の主要産業に情報技術は密接に関係していると認識し、小学校段階から体系的にソフトウェア中心にプログラミングを学習する中で論理的思考力や創意的思考力、問題分析能力、ICTリテラシーの育成を土台に専門的なコーディングの知識や技術を習得し、ICT人材として主要産業に貢献できる人材を育成する教育であると言える。

2.2.2 シンガポール

現在の初等教育段階におけるプログラミング教育はカリキュラム外であり、一部の学校で応用科目として独立教科となっているところもあるが、必修ではない。また中等教育でも普通校技術コース (Normal (Technical) Course)²³⁾で独立した必修教科「Computer Applications」が設けられている。2014年にシンガポール国内インターネット産業を担う国家機関、Infocomm Development Authority of Singapore (IDA) が、公立学校にソフトウェアのプログラミング授業を積極的に導入し、授業内で生徒にコンピュータコードを書く機会を与えることを検討している²⁴⁾と発表し、李顯龍(リー・シェンロン)首相の「スマート国家構築宣言」に基づいた、コンピュータ技術を最大限活用し、便利な生活を実現する「スマート国家」計画に沿った措置で2017年から普通校技術コースに限らず中等教育段階でコーディングをカリキュラムの中心とした「コンピューティング」の授業を実施することと高校進学の基準となる全国テスト (CCE O-LEVEL) に科目「プログラミング」を導入することが決定された^{25) 26)}。

情報通信産業を国の基幹産業と位置付けているため、教育方面においても1997年からITインフラを整備しICT設備を充実させている。文部科学省の諸外国におけるプログラミング教育に関する調査研究²⁷⁾によれば、現行の初等教育カリキュラムにプログラミング教育は含まれていないが、一部先進モデル校などの教科「Applied Learning」の中でプロジェクトワーク、汎用的な学習、情報モラル、ICTスキルの授業を行っている。中等教育の「Computer Applications」では情報通信技術やコンピューティングと通信機器の基礎知識や文書処理や表計算、プレゼンテーションソフトを使ったマルチメディアコミュニケーション、ビジュアルプログラミングを用いたメディアコンピューティングを教えている。2017年から中等教育で導入される「コンピューティング」では、テキスト言語「Python (パイソン)」を用いたプログラミング、アルゴリズムの理解、データ管理、コンピュータキテクチャー (基本設計概念) が教育内容の中心となる。指導者は初等・中等教育とも教科担任制であるため、「コンピューティング」に関しても同様に専門の教員が受け持つと考えられる。

プログラミング教育を導入する背景には、シンガポールが狭小な国土、それに依存する資源が乏しいことから、「人材こそ最大の資源」であるという国家観に基づいた戦略的な情報通信技術に関する教育の促進があり、自国の国際競争力の強化や経済力の維持する目的があると考えられる。また、自国内にICT人材が不足しているといった課題から技術者の認知度を上げ、技術者が非常に価値ある職業であると認識させるねらいもあると言える。

シンガポールの教育は、「人材を自国最大の資源である」という認識のもと国際競争力を強化す

るための人材の育成に尽力し続けている。その中でプログラミング教育は、自国内の国際競争力の強化や経済力の維持するためのIT技術を進化させる手段として戦略的にICT人材の育成を促していくものであると言える。

2.3 情報リテラシー型とICT人材育成型のプログラミング教育についてのまとめ

2.2ではそれぞれ2カ国ずつ情報リテラシー型とICT人材育成型のプログラミング教育についてみてきた。上述した内容を導入する時期、目的、位置付け、教育内容、指導員の観点でまとめたものが表3になる。

表3 4カ国2つのプログラミング教育形態の比較

	情報リテラシー型		ICT人材育成型	
	イギリス	フィンランド	韓国	シンガポール
導入期	初等教育段階	初等教育段階	前期中等教育段階 (2017年から初等教育段階)	一部中等教育段階 (2017年から中等教育段階)
目的	ICTの理解と思考プロセスの学習	21世紀型の教養の涵養	国の主要産業に貢献する人材の育成	国内のIT技術を進化させる担い手の育成
位置	独立教科 「Computing」	算数・数学を中心に科目を横断する	独立教科 「実科」 「ソフトウェア」	独立教科 「Applied Learning」 「ComputerApplications」
教育内容	<ul style="list-style-type: none"> ・アルゴリズムの理解 ・プログラムの作成 ・ネットリテラシー、情報リテラシーの涵養 ・創造的なプロジェクトへの取り組み 	<ul style="list-style-type: none"> ・正確な指示・伝達方法や論理的な思考の学習 ・コーディング(ビジュアルプログラミング言語) ・アルゴリズムの理解 ・コーディング(テキスト言語) 	<ul style="list-style-type: none"> 初等教育段階 ・コンピュータ思考の理解 ・ソフトウェアコーディング ・ICTリテラシーの涵養 前期中等教育段階 ・アルゴリズム原理の理解 ・コーディング(ビジュアルプログラミング言語) ・基礎的なコンピュータプログラムの作成(テキスト言語) 	<ul style="list-style-type: none"> 初等教育段階 ・情報モラルの理解 ・ICTスキルの学習 中等教育段階 ・コンピュータの基礎知識の理解 ・ソフトウェアを用いたメディア技術の習得 ・文書処理、表計算 ・メディアコンピューティング(ビジュアルプログラミング言語) ・コーディング(テキスト言語)

指導員	初) 学級担任	初) 学級担任	初) 学級担任	初) 教科担任 (Applied Learning)
	中) 教科担任 (Computing)	中) 教科担任 (数学)	中) 専門講師	中) 教科担任 (Computer Applications)

表3からわかるように、情報リテラシー型と ICT 人材育成型のプログラミング教育の大きな違いは教育目的の内容である。情報リテラシー型のプログラミング教育を行う国はプログラミングを 21 世紀における教養であると認識し、今日の IT 社会のテクノロジーの背景や考え方を理解したうえで、必要に応じてその知識や思考法を用いた充実した人間生活を送る人材を育成するための教育としてプログラミングを位置付けているのである。一方で ICT 人材育成型のプログラミング教育を行う国はプログラミングを 21 世紀における必須技能であると認識し、国際競争を勝ち抜いていくための戦力と位置付けている。したがってプログラミング教育のゴールを自国の主幹産業に関わる人材の育成、国の経済力を支えることができる人材の育成とし、そのために行っている戦略の中にプログラミング教育が組み込まれているのだ。

3 日本の取り組み

日本政府は平成 28 年 4 月 19 日に 2020 年度から初等中等教育からプログラミング教育を必修化する方針を明らかにした。それに伴って教育内容の検討や時期学習指導要領の審議が進められている。本章では、プログラミング教育に関する日本の政策と現状について取り上げ、第 2 章で記述した諸外国の取り組みと比較していく。

3.1 日本の政策と現状

日本の初等中等教育段階におけるプログラミング教育の必修化に向けた動きとして、平成 20 年 3 月の学習指導要領の中学校の技術・家庭教科内の技術分野の項目「プログラムによる計測・制御」の必修化、平成 25 年 6 月に発表された政府の成長戦略（素案）、世界最先端 IT 国家創造宣言の中に初等中等教育段階におけるプログラミング教育に関する内容が盛り込まれた。

そして平成 28 年 4 月 19 日、初等中等教育段階からプログラミング教育を必修化することが発表された。同日開催された第 26 回産業競争力会議において安倍晋三総理が明言したほか、文部科学省作成の「第 4 次産業革命に向けた人材育成総合イニシアチブ～未来社会を創造する AI/IoT/ビッグデータ等を牽引する人材育成総合プログラム～」関連資料にも明記され、「日本再興戦略 2016—第 4 次産業革命に向けて—」にもプログラミング教育必修化の内容が見られる。

政府の方針を受け、文部科学省は「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」を設置し 3 回にわたる会合の結果、小学校でのプログラミング教育は、総合的な学習の時間においてプログラミングを体験しながら社会における役割を理解し、それを軸としながら、各教科等における多様なプログラミング教育につなげていくという位置付けを示しつつ、次期学習指導要領の総則にプログラミング教育を行うことを定めるにとどめ、具体的な実施の在り方は各学校に委ねる²⁸⁾とした。また、中学校におけるプログラミン

グ教育については次期学習指導要領に向けたこれまでの審議のまとめの中に、小学校におけるプログラミング教育の成果を生かし、発展させるという視点から、従前からの計測・制御に加えて、動的コンテンツに関するプログラミングや、ネットワークやデータを活用して処理するプログラミングも題材として扱うことが考えられる²⁹⁾と示したが、具体的な指導内容までは触れていない。

現在プログラミングに関する教育として設定されている初等中等教育段階の公教育は、「プログラムによる計測・制御」のみであると言える。しかし、計測・制御のためのプログラムの作成を通して、コンピュータを用いた計測・制御の基本的な仕組みを知り、簡単なプログラムの作成ができるようにするとともに、情報処理の手順を工夫する能力を育成し、計測・制御に関する技術が社会や産業に果たす役割と影響について理解を深め、それらを適切に評価し活用する能力と態度を育成することをねらい³⁰⁾とした本項目は、学習内容の専門性の高さに対する授業時間の不足、教員の研修や指導経験の不足、教材開発の不十分さなどから学習定着力が低いという現状に置かれている³¹⁾。

公教育におけるプログラミングに関する教育は上述のような現状であるため、現在プログラミングに関する教育の場として主流なのは、民間事業者やボランティア、NPO法人等が主催するワークショップやスクールである。公教育のプログラミングに関する教育が進んでいない要因に教育現場のICT環境や教員のICT活用能力の育成の不十分さが挙げられ、教育を行う環境の整備が急務である。

政府の成長戦略内にプログラミング教育の必修化が盛り込まれ、時期学習指導要領にも各教科を横断した教育を行う方向性が示されたが、プログラミング教育の目的が統一されていないことを指摘したい。

文部科学省が各教科等で育まれる論理的・創造的な思考力を基盤とし、コンピュータの働きを理解しながら、それが自らの問題解決にどのように活用できるかをイメージし、意図する処理がどのようにすればコンピュータに伝えられるか、さらに、コンピュータを介してどのように現実世界に働きかけることができるのかを考えることプログラミング的思考を育むことであり、コーディングを覚えることが目的ではない³²⁾とする一方、政府の戦略・宣言の中では、産業競争力の源泉となるハイレベルなIT人材を継続的に育成・確保するためのIT教育の一環として位置付け³³⁾、突出したIT人材の発掘や、マッチング、継続したイベント等の実施によるハイレベルIT人材の発掘・支援及び表彰を行う³⁴⁾ことも含まれている。

上述のように日本のプログラミング教育は、最終的にICT人材の育成を目指しながらも、義務教育段階ではICT人材の育成ではなくコンピュータ的思考を育むものであると言えるだろう。

3.2 日本の取り組みの諸外国との比較

3.1で述べたように日本の公教育におけるプログラミング的思考を育成する教育は、教科を横断してプログラミング的思考を学ぶものと位置付けられた点においてフィンランドのプログラミング教育に近いといえるだろう。しかし、フィンランドのカリキュラムは学年ごとに明確な学習内容が決められているのに対し、日本の次期学習指導要領には学習内容や学習する学年の決定がない。各学校の現状に合わせて柔軟に実施できるのは学校の自主性を尊重したものであるが、ある程度の内容や学習の到達段階を設ける必要があるのではないだろうか。学習内容の到達段階が設定されなかった場合には、小学校卒業時における児童のプログラミング的思考の能力に学校間で格差が生ま

れたり、その後の教育への影響などが懸念される。

また、ICT人材育成の育成や発掘のために関連するイベント等の実施への積極的な取り組みは、シンガポールにも見られる。産業人材育成を進める情報中通信開発庁（Infocomm Development Authority of Singapore, 以下、IDA）が主催する National Infocomm Competition, 教育省やIDAが後援している National Software Competition など、中学からジュニアカレッジの学生向けのICT各分野の競技会がある³⁵⁾。シンガポールは国土の狭さや天然資源が乏しいことから人材を国家の重要資源と位置付け国家主体で国際競争力の主戦力としての人材を育成している。日本でも天然資源や食料の多くは海外輸入に頼り、国土も大きくないことから人材を国家の資源としてどのように扱っていくのかを参考にできるのではないだろうか。

以上のように、第2章で取り上げた国と日本のプログラミング教育について類似する点も見られるため、プログラミング教育を実施していくうえで参考できる点があるだろう。

一方、日本が各国と大きく異なるのは学習環境におけるICT環境の充実度とICTリテラシーの高さだ。七邊³⁶⁾は、日本では娯楽目的のモバイル端末の利用が盛んであるのに対し、学内外のICT端末の普及、ICTリテラシーとICT学習利用、義務教育家庭でのICT教育のいずれもがOECD加盟国に比べ低いことを示唆している。

日本の特に小学校教育におけるプログラミング教育がプログラミング的思考を養うものとしてアナログ学習を中心に進めていくとしても、パソコンをはじめとするICT機器を学習面で十分に利用できるように、学校のICT環境の整備は不可欠であることを理解し、急務の課題として解決が必要である。

4 プログラミング教育の在り方についての考察

これまで見てきたように、プログラミング教育は国によってその位置付けや内容が大きく異なる。これはプログラミング教育がICTと切っても切り離せない関係にあるからで、ICTに関する教育、日本の教科として言うなら「技術・家庭」や「情報」が常に国や地域の人材育成の教育方針が反映されやすい教科であることが挙げられるだろう。各々の教育方針を反映した結果、情報リテラシー型とICT人材育成型のプログラミング教育というような大別して2種類の特徴的な教育が見られるようになったのである。

そもそもプログラミング教育とは、ICTの存在を知り、ICT機器の使い方を知ったうえで実際に触れてみる。そうしてその背景にある考え方や実際に動かすための技術や知識を学び実践に移っていくという一連の流れの中でICTについて学ぶものであり、根本的な在り方は情報リテラシー型の教育とICT人材育成型の教育で違いは無いのだ。そのため、この2種類の特徴的な教育の在り方は、実質陶冶と形式陶冶のどちらを重視した教育であるかの対比になる。知識・技術の習得や育成を重視した情報リテラシー型の教育は実質陶冶的側面が強く、応用可能性や転移的学力を重視して思考力の育成や人格形成を目指した情報リテラシー型の教育は形式陶冶的側面が強い教育だと言えるだろう。

科学技術の進歩によってICT機器は今や日々の生活で欠かせない存在になった。今後これらを見捨て無視して現代社会の中で生きていくことは不可能に近い。ICTに関する知識は21世紀において必須教養となりつつある。そのため、義務教育段階からICTに関することを学ぶのは非常に歓迎さ

れる取り組みである。しかし、義務教育段階の教育は個人の人格形成の基盤をなしていることを忘れてはならず、プログラミング教育も例外ではない。そのため義務教育段階でのプログラミング教育はやはり情報リテラシー型の教育であることが望ましいと言える。ICT人材の育成も含めて、将来の様々な能力形成のためにも情報リテラシー型の教育に基づく思考力やリテラシーは必須教養として望まれるのではないだろうか。

注

- 1) IT用語辞典 e-Words 「プログラミング【programming】」
<http://e-words.jp/w/%E3%83%97%E3%83%AD%E3%82%B0%E3%83%A9%E3%83%9F%E3%83%B3%E3%82%B0.html> (2016.10.12).
- 2) ヴィジュアルプログラミング言語 Viscuit ビスケット <http://www.viscuit.com/> (2016.10.16).
- 3) ブロックの組み合わせによってプログラミングするオブジェクト指向言語 総務省『プログラミング人材育成の在り方に関する調査研究』(2015) ,p.6より.
- 4) 原田康徳「ビジュアルプログラミング言語ビスケット (Viscuit) の紹介」『コンピュータソフトウェア』32 (2015), pp.18-26.
- 5) 総務省「プログラミング人材育成の在り方に関する調査研究」(2015), p.9.
- 6) Alan Curtis Kay によって作られた教育用オブジェクト指向プログラミング言語.
- 7) ステージ（背景）の上に表示され、プログラムによって動きや大きさ、色などを変化させることができるものこと.
 IT企画研究所 Scratch などの子供のプログラミング教育を考える大人をサポートします How is IT? 「子供向けプログラミング言語 Scratch - エディターの各部名称と役割とは」
<http://www.howisit.jp/2016/03/30/scratch-parts-in-editor/> (2016.10.16).
- 8) Monroy-Hernández, A. and Resnick, M. Empowering kids to create and share programmable Media. *Interactions* ,2 (2008) , 50-53.
<https://arxiv.org/pdf/1507.01282.pdf> (2016.10.17)
- 9) 文部科学省 プログラミン 「プログラミンのやくわり表」
http://www.mext.go.jp/programin/data/help/programin_chart.pdf (2016.10.18).
- 10) 文部科学省 プログラミン 「プログラミンについて」 <http://www.mext.go.jp/programin/about.html> (2016.10.18).
- 11) K-12 (ケースルートトゥエルブ, kay-through-twelve) 「幼稚園 (Kindergarten の K) から始まり高等学校を卒業するまでの 13 年間の教育期間」のこと.
- 12) 教師が生徒の進捗をモニターできるインターフェースがある.
 TechCrunch「Code.org の Code Studio は、子供たちにプログラミングを教えるためのツールとカリキュラム」
<http://jp.techcrunch.com/2014/09/12/20140911code-org-launches-code-studio-a-toolset-and-curriculum-for-teaching-kids-programming/> (2016.10.20).
- 13) 小学校学習指導要領第 3 章道徳の項目から道徳の内容を 2 学年ずつ示していること及び「第 3 指導計画の

作成と内容の取扱い」から、低学年 = 1,2 年生, 中学年 = 3,4 年生, 高学年 = 5,6 年生であると定義した。

- 14) 文部科学省「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), p.22.
http://jouhouka.mext.go.jp/school/pdf/programming_syogaikoku_houkokusyo.pdf (2016.10.20).
- 15) イギリスの初等・中等教育課程における学年の区分けは Key Stage (KS) と呼ばれ, 5 歳から 16 歳までの 11 年間 (11 学年) を KS1 : 5-8 歳 (1-2 学年), KS2 : 8-11 歳 (3-6 学年), KS3 : 11-14 歳 (7-9 学年), KS4 : 14-16 歳 (10-11 学年) と区分されている。
- 16) 文部科学省「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), p.22.
http://jouhouka.mext.go.jp/school/pdf/programming_syogaikoku_houkokusyo.pdf (2016.10.20).
- 17) 同書 p.67.
- 18) 同書 p.68.
- 19) フィンランド大使館, 東京「プログラミングの寵児, リンダ・リウカスが『ルビィ』を語る」
<http://www.finland.or.jp/public/default.aspx?contentid=352707&nodeid=41264&contentlan=23&culture=ja-JP>
 (2016.10.21).
- 20) 文部科学省「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), p.134.
http://jouhouka.mext.go.jp/school/pdf/programming_syogaikoku_houkokusyo.pdf (2016.10.20).
- 21) 同書 p.135.
- 22) 韓国のイマを伝えるもっと! コリア「現在の小4 から, 中学校でソフトウェアが必須科目に」
http://mottokorea.com/mottoKoreaW/KoreaNow_list.do?bbsBasketType=R&seq=24256 (2016.10.21).
- 23) Ministry of Education SINGAPORE 「Education System」
<https://www.moe.gov.sg/education/education-system> (2016.10.22).
- 24) THE BRIDGE 「シンガポールが公立学校にプログラミングの授業を導入, 経済活性化を図る」
<http://thebridge.jp/2014/01/salon-io-japan-launch> (2016.10.22).
- 25) AsiaX ニュース 「コンピュータプログラミングを中等学校で学習, 来年から」
<http://www.asiax.biz/news/36515/> (2016.10.22).
- 26) Biz コンパス 「シンガポールの子供は 4 歳からプログラミングを学ぶ」
http://www.bizcompass.jp/original/re-trend-075-2.html#articleDetailBody_Series (2016.10.22).
- 27) 文部科学省「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), pp.151-153.
http://jouhouka.mext.go.jp/school/pdf/programming_syogaikoku_houkokusyo.pdf (2016.10.20).
- 28) 文部科学省「小学校段階におけるプログラミング教育の在り方について (議論の取りまとめ)」(2016).
http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm (2016.10.23).
- 29) 文部科学省「次期学習指導要領に向けたこれまでの審議のまとめ (素案)」(2016), p.115.
http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/053/siryo/_icsFiles/afieldfile/2016/08/02/1375316_3_2_5.pdf (2016.10.23).
- 30) 文部科学省「中学校学習指導要領 第 2 章 各教科 第 8 節 技術・家庭」
http://www.mext.go.jp/a_menu/shotou/new-cs/youryou/chu/gika.htm (2016.10.23).
- 31) 森石峰一「計測・制御教育のための教材開発とその教材を活用した授業設計及び実践」(2013), pp.8-9.
https://oecu.repo.nii.ac.jp/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=136&item_no=1&page_id=13&block_id=17 (2016.10.23).

- 32) 文部科学省「小学校段階におけるプログラミング教育の在り方について（議論の取りまとめ）」(2016).
http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1371901.htm (2016.10.24).
- 33) 産業競争力会議「成長戦略（素案）」(2013), p.46.
<http://www.kantei.go.jp/jp/singi/keizaisaisei/skkaigi/dai11/siryou1-1.pdf> (2016.10.24).
- 34) 高度情報通信ネットワーク社会推進戦略本部（IT 総合戦略本部）「世界最先端 IT 国家創造宣言工程表」(2016), p.52.
http://www.kantei.go.jp/jp/singi/it2/kettei/pdf/20160520/koteihyo_kaitei.pdf (2016.10.24).
- 35) 文部科学省「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), p.155.
http://jouhouka.mext.go.jp/school/pdf/programming_syogaikoku_houkokusyo.pdf (2016.10.20).
- 36) 七邊 信重「世界の学校内外での生徒の ICT 利用状況と日本の課題—OECD「PISA2012 年調査」の結果から—」(2014), p.5.
https://www.fmmc.or.jp/pdf/report/report_world_20140416.pdf (2016.10.24).