

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

A Critical Review on Institutionalization of Programming Education at Elementary and Secondary School Level in Japan

阪東哲也* 黒田昌克** 福井昌則*** 森山潤****
BANDO Tetsuya KURODA Masakatsu FUKUI Masanori MORIYAMA Jun

本研究の目的は、我が国の初等中等教育において普通教育としてのプログラミング教育が導入される経緯を整理し、今後の実践に向けた課題について検討することである。我が国におけるプログラミング教育の位置づけの変遷、海外における Computational Thinking の考え方やカリキュラム改革の動向を整理した上で、技術リテラシー育成の観点からプログラミング教育の課題について検討した。その結果、①我が国のプログラミング教育の中核概念となっているプログラミング的思考の考え方は Computational Thinking の概念に比べて矮小であり、育成できる技術イノベーション力が「手続きを構築する力」に留まってしまう危険性があること、②特に、小学校段階のプログラミング教育では「プログラミングをよりよい人生や社会づくりに生かす」ことの具体として技術ガバナンス力育成の視点が明確でないこと等に課題があることを指摘した。

キーワード：初等中等教育、プログラミング教育、Computational Thinking、技術リテラシー、教育制度

1. 本研究の目的

本研究の目的は、我が国の初等中等教育において普通教育としてのプログラミング教育が導入される経緯を整理し、今後の実践の方向性について検討することである。

2. 背景及び問題

情報技術の進展は私たちのワークスタイル、ライフスタイルを変化させた。情報技術を適切に活用し、私たちの生活をより豊かにするための情報教育の重要性が指摘されて久しい。

我が国の情報教育の萌芽は高等教育における専門教育において見られる。米国の ACM (Association for Computing Machinery) の Curriculum68¹⁾ の影響を受け、1970年には我が国にも新たな学問領域としてコンピュータ・サイエンス (以下、CS) を専門とする情報科が誕生した。Curriculum68に基づく CS の学習内容の中心はプログラミング教育であり、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とされた。しかし、初期の CS では指導法が確立されていなかったために、コンピュータへの苦手意識を高めてしまうことが懸念された。そのためにプログラミング教育から、コンピュータの操作と利活用を中心とした実際の技術習得を目指す教育へ方向転換された²⁾。専門教育として始まった CS も、コンピュータが広く一般家庭に普及されたことに伴い、普通教育としての性質を帯びるようになった。

しかし、近年では、高度情報社会の到来を受け、IT を活用する人材が不足するだろうという予測を受け、コ

ンピュータの操作や利活用を中心とした CS から再転換を図り、プログラミング教育を充実させる CS が注目を集めている。今後の情報社会においては、AI や IoT といった新たな情報技術の活用が見込まれている。更に情報技術が進展すれば、既存の判断の枠組では対応することができない問題も生じる可能性がある。これらのような技術の活用、判断を適切に行えるように、基礎的な資質育成を担うことがプログラミング教育に期待されている。

期待感が高まっているプログラミング教育ではあるが、その実践に向けては2つの重要な課題が残されたままである。1点目は、どのようにプログラミング教育実践を行うかについてである。小学校では情報科の専門的な免許状をもつ教員の配置は考慮されていないために、教員が主体的にプログラミング教育実践を計画することが難しいということが挙げられる。この点については、文部科学省ではプログラミング教育先進校による実践を集めたプログラミング教育実践ガイドを刊行し³⁾、個別のプログラミング教育実践や学習材そのものに対する効果検証が行われているところであるが、初等中等教育の接続を考えたプログラミング教育実践の検討についてはスタート地点に立ったところである。

2点目の課題は、プログラミング教育を通して、どのような力を育成していくかについてである。「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、有識者会議における議論の取りまとめ」(以下、議論の取りまとめ)では、初等中等教育の発達段階に即して、「知識・技能」、「思考力・判断力・表現力等」、「学びに向か

*常葉大学 **兵庫教育大学大学院(専門職学位課程)教育実践高度化専攻授業実践開発コース

平成29年6月28日受理

***兵庫教育大学大学院(修士課程)教科教育実践開発専攻生活・健康・情報系教育コース

****兵庫教育大学大学院教科教育実践開発専攻生活・健康・情報系教育コース, 教育実践高度化専攻授業実践開発コース 教授

う力・人間性等」の3観点に基づく資質・能力を育成することが明示された。その中でも、「思考力・判断力・表現力等」ではプログラミング的思考を育成すると明記された。プログラミング的思考とは、「自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力」と定義されている。さらに、プログラミング的思考は「いわゆる「コンピューショナル・シンキング」の考えを踏まえつつ、プログラミングと論理的思考との関係を整理」したものとされている⁴⁾。しかし、プログラミング的思考に関する記述からは、これまでのプログラミング教育との関連性、Computational Thinkingの考え方をどのように踏襲されたかについては言及されていない。

そこで、次章では、我が国の初等中教育段階における学習指導要領に規定されたプログラミング教育の位置づけの変遷について整理する。

3. 我が国における情報教育の史的展開とプログラミング教育の位置づけ

3.1 中等教育におけるプログラミング教育の黎明期(1970年・1978年 高等学校学習指導要領)

我が国の中等教育におけるプログラミング教育も、高等教育同様、専門教育から始まる。1970年の高等学校学習指導要領では、「工業⁵⁾」, 「商業⁶⁾」, 「理数⁷⁾」において、プログラミング教育に関する学習内容の記述がある。これらの目標は、①プログラミングに関する基礎的な知識と技術を習得させること、②情報を合理的に処理する能力(問題解決、アルゴリズムを含む)を高めること、③コンピュータの構成および機構の概要について理解させることの3点に整理できる。

学習指導要領に基づき、高等学校で学科が設定された。プログラミング教育を実施していた情報技術科の中には、機械系の技術に重点をおくコース、電子系の技術に重点をおくコース、ソフトウェアの技術に重点をおくコースの3コースが考えられており、計算機の仕組み・プログラミング(FORTRAN言語)、問題解決の技術(コボル言語)、ハードウェアとソフトウェアの関係(アセンブラ言語)、計算技術(BASIC)による学習が進められてきたと記録されている⁸⁾。

その後、1978年に改訂された高等学校学習指導要領(昭和53年8月告示)では、工業⁹⁾と商業¹⁰⁾ではプログラミングについての記述されているものの、理数¹¹⁾については、「使用する計算機の機能に応じてプログラムを作成し、…」との記述に留まり、プログラミングの文言は削除されている。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と考えられる。

以上のことから、コンピュータが一般家庭に普及していない1970年代では中等教育後期の高等学校においてプログラミング教育を中心とした専門教育としてのCSが

展開されてきたことが分かる。

3.2 義務教育段階(中等教育前期にあたる中学校)におけるプログラミング教育の導入(1989年 中学校学習指導要領)

中等教育後期にあたる高等学校ではプログラミング教育が先行して実施されてきたが、中等教育前期にあたる中学校でもプログラミング教育が展開されるようになった。1989年の中学校学習指導要領(平成元年3月告示)では、「技術・家庭¹²⁾」で「情報基礎」の領域が新設された。その目標は、「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こと」としている。「情報基礎」は「技術・家庭」における11の領域のうち、必修ではない領域の1つとしての位置づけであった。

以上のことから、すべての中学生にプログラミング教育の機会が保障されたとはいえないが、中等教育後期の高等学校から、義務教育段階にあたる中学校にもプログラミング教育が盛り込まれるようになったという点は大変意義深い。

3.3 普通教育としてのプログラミング教育の必修化(1998年 中学校学習指導要領・1999年 高等学校学習指導要領)

1990年代後半には、我が国の中等教育において、普通教育として情報教育が展開されるようになった。中等教育前期にあたる中学校では、1998年の中学校学習指導要領(平成10年12月告示)において、「技術・家庭¹³⁾」の「技術分野」(以下、中学校技術・家庭科技術分野)は「技術とものづくり」と「情報とコンピュータ」の2つの内容に分けられた。プログラミング教育の目標は、「情報とコンピュータ」の「プログラムと計測・制御について、プログラムの機能を知り、簡単なプログラムの作成ができること、コンピュータを用いて、簡単な計測・制御ができること」であった。

一方、中等教育後期にあたる高等学校では1999年の高等学校学習指導要領(平成11年3月告示)より普通教育に関する教科「情報¹⁴⁾」が新設された。普通教育に関する教科「情報」は「情報A」, 「情報B」, 「情報C」の3つの科目で構成されており、これらのうち1科目を選択して履修する。この中で、主にプログラミング教育と関連のある内容を取り扱っているのは「情報B」で、その目標は「コンピュータにおける情報の表し方や処理の仕組み、情報社会を支える情報技術の役割や影響を理解させ、問題解決においてコンピュータを効果的に活用するための科学的な考え方や方法を習得させる」とされている。また、生徒同士の相互評価や討議など協働的に問題を解決する活動や、現実的な問題についてモデル化やシミュレーションを活用して解決を目指す活動など、実習や協働性に重点を置くことも示されている。

以上のことから、1990年代後半の学習指導要領の改訂により、原則として中等教育においてはプログラミング

教育が必修科目となった。ただし、内容の取扱いでは、プログラミングの専門的な知識や技能の習得自体を目的としたり深入りしたりしないようにすることが強調されており、コンピュータの操作や利活用を中心に転換してきていることが分かる。

3.4 プログラミング教育の展開期（2008年 中学校学習指導要領・2009年 高等学校学習指導要領）

これまでのプログラミング教育では、コンピュータを利用した計測・制御の基本的な仕組みを知ること、簡単なプログラムが作成できることを目指していたが、2000年代後半には、技術の習得以外の観点も求められるようになった。2008年の中学校学習指導要領技術・家庭科編¹⁵⁾（平成20年3月告示）では、「情報に関する技術」に加えて、「すべての内容において、技術にかかわる倫理観や新しい発想を生み出し活用しようとする態度が育成されるようにするものとする」という内容の取扱いに文言が追記された。

また、高等学校では2009年の高等学校学習指導要領（平成21年3月告示）において、「情報A」、「情報B」、「情報C」という科目の構成から「社会と情報」、「情報の科学」という構成に改編された。プログラミング教育に関する内容として、「情報の科学」の「問題解決とコンピュータの活用」において、問題解決の基本的な考え方、問題の解決と処理手順の自動化、モデル化とシミュレーションについて取り扱うことが明記された¹⁶⁾。問題解決の手段の1つとしてプログラミングを位置づけている点は、ここまでの学習指導要領で終始一貫している。

以上のことから、中等教育におけるプログラミング教育においてはプログラミングの技術を学ぶことに焦点化するのではなく、プログラミング教育を通して、新しい発想を生み出し活用するための態度や問題解決を行うための基本的な考え方の習得を目指してきたことが分かる。

3.5 初等教育へのプログラミング教育の導入の経過

コンピュータの飛躍的普及、更なる情報社会の進展により、今後の社会変化は予想することが難しいとされている。そのような時代においても現代的課題を発見、対応するために、初等教育からの体系的な情報教育の重要性は極めて高い。初等教育からのCS実施に向けて、じっくりと議論されてきた。初等教育にプログラミング教育を充実させるCSが導入されるようになった経過について表1に示す。

初等教育におけるプログラミング教育の導入の議論は2013年4月に開催された第6回産業競争力会議に端を発する。同会議では、ITを活用したビジネスイノベーションを担うエンジニアの質と量を向上させることが提言された¹⁷⁾。それを受けて同年6月に発表された「日本再興戦略¹⁸⁾」では、ITを活用した21世紀型スキルの修得を目指して義務教育段階からのプログラミング教育等のIT教育を推進することが明記された。そして、2016年4月に行われた第26回産業競争力会議では、「第4次産業革

命に向けた人材育成総合イニシアチブ」において、次世代に求められる情報活用能力の育成を目指して「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科目化といった、発達の段階に則した必修化」という明確な姿勢を文部科学省が打ち出した¹⁹⁾。それを受けて同年6月に発表された「日本再興戦略2016²⁰⁾」では、同年5月から6月にかけて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が行われ、同年6月に議論の取りまとめが報告された。さらに同年12月には中央教育審議会の答申でプログラミング教育に関して、小学校、中学校、高等学校の各段階における具体的な目標や内容、枠組等が明示された²¹⁾。そして2017年3月に告示された小学校、中学校の新学習指導要領にプログラミング教育に関する内容が明記された。

以上のことから、CSの充実という教育的配慮の観点からではなく、将来のIT人材の確保に向けて、産業・経済界からの強い要請を受けて、義務教育からプログラミング教育が必修化されるようになったことが分かる。

3.6 新学習指導要領上の位置づけ

3.6.1 小学校におけるプログラミング教育

小学校でのプログラミング教育の目標を資質・能力の3つの柱（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）で説明している。具体的には、知識・技能では「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」が挙げられている。思考力・判断力・表現力等では、発達の段階に即して、プログラミング的思考

表1 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

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 ー第4次産業革命に向けてー	
2016	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	小学校におけるプログラミング教育で育成を目指す、知識・技能、思考力・判断力・表現力等、学びに向かう力・人間性等それぞれの観点における目標の明示など
	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科目「情報1」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年告示）	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年告示）	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年告示）予定	共通教科「情報」の「情報1」を共通必修科目とし、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）

の育成が示されている。そして、学びに向かう力・人間性等では、「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」がある。

これらの議論の取りまとめで示されたプログラミング教育の目標を踏まえ、2017年3月に告示された小学校の新しい学習指導要領では、これまでの学習指導要領には存在しなかったプログラミング教育に関連する内容が、第1章総則、第2章各教科の「算数」及び「理科」、第5章「総合的な学習の時間」において新たに加わった²²⁾。総則では、各教科等の特質に応じて計画的に実施する学習活動として「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」が明記された。「算数」では、第1章総則に掲げるプログラミングを体験しながら論理的思考力を身に付けるための活動として「〔第5学年〕の「B図形」の(1)における正多角形の作図を行う学習に関連して、正確な繰り返し作業を行う必要がある、更に一部を変えることでいろいろな正多角形を同様に考えることができる場面」が例示された。また、「理科」では、「〔第6学年〕の「A物質・エネルギー」の(4)における電気の性質や働きを利用した道具があることをとらえる学習など、与えた条件に応じて動作していることを考察し、更に条件を変えることにより、動作が変化することについて考える場面」を例として挙げている。「総合的な学習の時間」では、「プログラミングを体験することが、探究的な学習の過程に適切に位置づくようにすること」に留意することが付記された。

このように小学校のプログラミング教育では、プログラミングは体験として位置づけられることとなった。

3.6.2 中学校におけるプログラミング教育

2016年12月の中央教育審議会の答申では、急速な発達を遂げている情報の技術に関しては、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、技術・家庭科の技術分野においてプログラミングに関する内容を倍増する方針が示された。

この中央教育審議会の答申を受けて、新しい学習指導要領の技術・家庭科技術分野では内容D「情報の技術」において従来の計測・制御におけるプログラミングに加えてネットワークを利用した双方向性のあるコンテンツのプログラミングを用いて問題を解決する活動が組み込まれた²³⁾。具体的な内容としては、情報通信ネットワークの構成と、情報を利用するための基本的な仕組みを理解し、安全・適切なプログラムの制作、動作の確認及びデバッグ等ができること、問題を見いだして課題を設定し、使用するメディアを複合する方法とその効果的な利用方法等を構想して情報処理の手順を具体化するとともに、制作の過程や結果の評価、改善及び修正について考えることの2点が示されている。

また、従前からの計測・制御におけるプログラミングにおいてもその内容がより具体的に示された。計測・制

御システムの仕組みを理解し、安全・適切なプログラムの制作、動作の確認及びデバッグ等ができること、問題を見いだして課題を設定し、入出力されるデータの流れを元に計測・制御システムを構想して情報処理の手順を具体化するとともに、制作の過程や結果の評価、改善及び修正について考えることが明記されている。新しい学習指導要領では、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のどちらにおいても実生活や社会との関わりを意識した学習活動となることをより重視している点が特徴的だといえる。また、内容D「情報の技術」全体では、2つのタイプのプログラミングを問題解決的な学習活動の中心に位置づけながら、情報の技術に関する科学的な理解、生活や社会に果たす役割と影響の理解、情報の技術の概念、情報の技術の評価、選択、管理・運用の在り方や改良、応用の在り方を考えることなど、体系的な学習内容を構成している点に小学校との違いを見ることができる。

3.6.3 高等学校普通科におけるプログラミング教育

2009年改訂の高等学校学習指導要領において、プログラミング教育に関する内容は、「情報の科学」に含まれている。しかし、必修でないために「情報の科学」を履修しない生徒がいることが、小中高を通じた体系的なプログラミング教育の実施という観点から見ると、重大な課題であった。そこで、2016年12月の中央教育審議会の答申では、情報技術を適切かつ効果的に活用する力を全ての生徒に育み、全ての高校生がプログラミングによりコンピュータを活用する力を身に付けられるようにすることが目指された。

そして、情報科の科目編成（「情報Ⅰ」、「情報Ⅱ」）が議論され、プログラミング教育はすべての生徒に必修の学習内容となった。具体的には、「情報Ⅰ」では「プログラミング及びモデル化とシミュレーション、ネットワーク（関連して情報セキュリティを扱う）とデータベースの基礎といった基本的な情報技術と情報を扱う方法とを扱うとともに、情報コンテンツの制作・発信の基礎となる情報デザインを扱い、さらに、この科目の導入として、情報モラルを身に付けさせ情報社会と人間との関わりについて考えさせることとして、内容を構成すること」と示された。2018年にこの答申を踏まえて高等学校の学習指導要領が改訂される予定であり、我が国の情報教育において、小中高を通じた体系的な情報教育が展開されることとなる。

3.7 我が国のプログラミング教育の特徴

本章では、学習指導要領を中心に、プログラミング教育の変遷について整理した。それらを踏まえた上でこれまでのプログラミング教育の歴史的経緯の特徴を探る。学習指導要領におけるプログラミング教育に関連する主な教科・科目・内容等の変遷を校種別等でまとめたものを表2に示す。

プログラミング教育はアメリカのACMのCurriculum

68の影響を受け、高等教育としてCSが誕生した。その後、1970年には中等教育後期の高等学校の専門教育としてCSが始まり、プログラミング教育が始まった。当初のCSの学習目標はプログラミングを通して、コンピュータの構造理解、ソフトウェアの作成から、アルゴリズムの理解まで、単なるコーディングやプログラミングの教育ではなく、新しい技術としてのコンピュータに対する理解や新しい技術としてプログラミングの考え方を習得させようとするものであった。

専門教育としてのCSから、1990年代から中学校技術・家庭科技術分野や高等学校の「情報」において普通教育としての性質を帯びるものになった。2010年代後半には初等教育へと内容が拡大していく流れは、社会の変容に大きく影響を受けているといえる。コンピュータをはじめとした情報機器やインターネットなどの情報技術が一部の専門的な人々だけのものではなく、急速に広がる情報化社会への対応は学校教育の懸案であったことは想像に難くない。

以上のように、プログラミング教育の特徴を俯瞰的に整理していくと、情報技術の飛躍的な普及に対応するために、プログラミング教育のための制度が整備されてきたものの、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のが現状である。特に、教科内での位置づけや学習内容が短期間で急激に変化しており、教育課程におけるプログラミング教育の位置づけが校種によって統一されていないことは大きな課題といえよう。社会の変化に柔軟に対応できるよう、普遍的に求められる資質・能力の育成を目指すという我が国のプログラミング教育の根幹に基づいて教育活動を発展させていくことが重要であろう。

4. Computational Thinking に基づくプログラミング教育の展開

4.1 Computational Thinking の考え方

前章では、我が国において、プログラミング教育がCSの普通教育として導入されるようになった背景について整理した。そして、普遍的に求められる資質・能力の育成の観点でプログラミング教育をとらえる必要性を指摘した。前述したように、我が国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の育成を目指している。プログラミング的思考は Computational Thinking の考え方に基づいているが、相互の関係性については明確にされてはいない。そこで、Computational Thinking の考え方について整理する。

Computational Thinking の語が初めて用いられたのは Papert による。Papert は自著 Mindstorms の中で、“Their visions of how to integrate computational thinking into everyday life was insufficiently developed (p. 182)²⁴⁾” と Computational Thinking を基本的な能力の一つとして統合する方法論が確立されていないと指摘した。今日の Computational Thinking は、Wing により提唱されたものである²⁵⁾。その定義や概念については一定のコンセンサスが得られてはいない。その背景には Computational Thinking は読み、書き、算術（そろばん）の3R同様、すべての人にとって必要な基本的スキルの1つと考えられている幅広い概念としてとらえていることにある。そのため、Computational Thinking はプログラミング教育、CSだけではなく、数学や科学などの多くの領域で用いられている。本稿では主にCSの領域での Computational Thinking について取り上げ、整理する。

CSの領域での Computational Thinking の定義として、Wing (2011) による「問題発見・解決の思考プロセスであり、その解決策は情報処理エージェントによって効果的に実行できる形式で表される。」がよく知られている²⁶⁾、²⁷⁾。

Wing による定義の他には、Aho はコンピューショナルステップとアルゴリズムに代表される解決方法といった問題の定式化を含む思考プロセスとしている²⁸⁾。また、Mannila らは K-9カリキュラムにおいて、異なる領域における問題や解決策の定式化を助けるためのCSからの概念や思考プロセスと考えている²⁹⁾。Priami は Computational Thinking の基本的な特徴を抽象化にあるとし、モデルで無視された細部を機械によって実行可能になるような方法で現実を抽象化することととらえている³⁰⁾。また、Riley & Hunt はコンピュータ科学者が考える方法や推論するための仕方といった思考の認知的方略としている³¹⁾。これらの Computational Thinking 研究を分析し、Computational Thinking のフレームワークを導

表2 学習指導要領におけるプログラミング教育に関連する主な教科・科目・内容等の変遷

学習指導要領の改訂 校種等	1970	1978	1989	1998・1999	2008・2009	2017・2018
	プログラミング教育に関連する主な教科・科目・内容等					
小学校						総合的な学習の時間 算数 理科 などの教科等を中心に行う
中学校	技術・家庭 「情報基礎」		技術・家庭 「情報とコンピュータ」		技術・家庭 「情報に関する技術」	技術・家庭 「情報の技術」
高等学校 (共通教科)			情報 「情報B」		情報 「情報の科学」	情報 「情報Ⅰ」(予定) 「情報Ⅱ」(予定)
高等学校 (専門教科)	工業 「プログラミング」 「プログラミング理論」 商業 「プログラミングⅠ」 「プログラミングⅡ」 理数 「計算機数学」	工業 「情報技術Ⅰ」 「情報技術Ⅲ」 商業 「情報処理Ⅰ」 「情報処理Ⅱ」	工業 「情報技術基礎」 「プログラミング技術」等 商業 「情報処理」 「プログラミング」	工業 「情報技術基礎」 「プログラミング技術」等 商業 「ビジネス情報」 「プログラミング」 情報 「情報システムの開発」 「アルゴリズム」等	工業 「情報技術基礎」 「プログラミング技術」等 商業 「ビジネス情報」 「プログラミング」 情報 「アルゴリズムとプログラム」 「情報システム実習」等	教育内容の改善・充実の方向性 〔工業〕技術の高度化や情報技術の発展等への対応に関する学習の充実 〔商業〕プログラミングとシステム開発に関する知識と技術の一体的な習得 〔情報〕問題解決やプログラミングに関する学習の充実

出する試みがなされている^{32),33)}。これらの研究では、①問題解決、問題の理解、問題の定式化に関する定義、②アルゴリズムや前提条件のような能力の改善に関する定義に大別している。

Computational Thinking の定義は多岐に渡るために、その概念も領域に応じて主張が幅広くなされている。

Wing が当初に提唱したものとしては以下の7つに整理される。

- (1) 計算プロセスの能力と限界の上に成立しているもの
 - (2) コンピュータ科学者だけではなく、すべての人にとって基本的な技術
 - (3) 一見難しそうな問題を我々がすでに解き方を知っている問題に変換する
 - (4) 再帰的に考えることであり、並列処理であり、命令をデータとし、データを命令とすること
 - (5) 巨大で複雑なタスクに挑戦したり、巨大で複雑なシステムをデザインしたりするときに、抽象化と分割統治を用いること
 - (6) 予防、防御、そして最悪のシナリオからの復帰という観点を持ち、そのために冗長性、故障封じ込め、誤り訂正などを用いること
 - (7) ヒューリスティックな推論により解を発見すること
- 以上の概念を踏まえて、Wing は以下の6つの具体的な特徴をもつものとしている。

- (1) 概念化であって、プログラミングではない。(様々な抽象化レベルで考えること)
- (2) ルーチンワーク(機械的)でなく、考えるという基礎的な技能
- (3) 人間の創造性や知性、スマートさをコンピュータで拡張できるもの
- (4) 数学的思考と工学的思考を補完するもの
- (5) モノ(ソフトウェアやハードウェア)ではなく、発想や概念
- (6) 万人のためのもの

その後、Wing は Computational Thinking の本質として、抽象化(Abstraction)と自動化(Automation)という二種類の分析的思考を説明している³⁴⁾。この考え方を踏まえて、磯辺らは Computational Thinking を(1)問題を解決するための手順の過程(抽象化)、(2)問題解決のプロセスを定型化する過程(自動化)ととらえている³⁵⁾。

また、アメリカの情報教育への影響が強い ISTE と CSTA では協力して、Computational Thinking に共通の認識を操作的定義として提案している。ISTE と CSTA は Computational Thinking を問題解決プロセスとした上で、以下の特徴をもつ概念としている³⁶⁾。

- ・コンピュータや他のツールを使って問題を解決できるように問題を策定すること
- ・データを論理的にまとめ、分析すること
- ・モデル化やシミュレーションなどの抽象化によるデータ表現

- ・アルゴリズム的思考による解決の自動化(一連の順序づけられたステップ)
- ・ステップとリソースの最も効果的で効果的な組み合わせを達成するという目標を持った、可能なソリューションの特定・分析・実装
- ・問題解決過程の一般化と広い範囲にわたる問題への移行

さらに、これらのスキルを、Computational Thinking の本質的な次元である、多くの性質または態度によってサポートされ、強化されるとし、これらの態度を以下の5点に集約している。

- (1) 複雑さに対処するための自信
- (2) 困難な問題に取り組む際の持続性
- (3) あいまいさに対する許容
- (4) オープンな問題に対処する能力
- (5) 共通の目標または解決策を達成するために他の人とコミュニケーションを行なって仕事をする能力

また、初等教育においてプログラミング教育におけるガイドラインを提供している CAS では、Computational Thinking を、論理的推論を含む、認知や思考プロセスとした上で、以下の特徴をもつ概念としている³⁷⁾。なお、CAS の Computational Thinking の概念は Selby & Woollard の研究に基づいている³⁸⁾。

- (1) アルゴリズム的に考えることのできる能力
- (2) 構造化の観点で考えることのできる能力
- (3) パターンを見つける、使うなど一般化して考えることのできる能力
- (4) よい表現を選ぶことができるなど抽象化して考えることのできる能力
- (5) 評価の観点で考えることのできる能力

以上のように、Computational Thinking には複数の定義が存在するものの、プログラムを考えるプロセスというプログラミングに限定的な思考プロセスではないという点は一貫している。Syslo & Kwiatkowska は、Computational Thinking はコンピュータプログラミングの結果によるものではなく、コンピューティングの原理に着目すべきだと主張している³⁹⁾。このように、プログラミング教育で育成される Computational Thinking は問題発見や解決の思考プロセスとして広くとらえる必要がある。

なお、近年では、Computational Thinking の概念の測定法についても研究が進んでおり、プログラミング環境 Scratch の学習に合わせて、Computational Thinking を computational concepts (概念)、computational practices (実践)、computational perspectives (見方)の3領域でとらえることの重要性が指摘されている^{40), 41)}。

4.2 海外のプログラミング教育の動向

文部科学省が行っ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報告書⁴²⁾や太田らのプログラミング教育の整理⁴³⁾によると、Computational Thinking の考え方は諸外国のプログラミング教育に大きな影響を与え

ていることが分かる。ここでは、初等教育におけるプログラミング教育が必修化されているイギリスと、教育課程、教員養成制度の在り方等、我が国の教育制度の様々な領域で大きな影響力を持ち、かつ、前述した ACM の A Model Curriculum for K-12 Computer Science 以降、全世界の情報教育を牽引しているアメリカに焦点を当てて、Computational Thinking とプログラミング教育の関連性について簡単に整理する。

4.2.1 イギリスのプログラミング教育の動向

イギリスでは2010年代前半において、情報教育の質の低下が大きな問題となった。当時、教科「ICT」では、CSを指導できる教員の不足、教材開発の不足、また、「専門職能発達」の提供・機会が不十分であったことから、情報教育の改善が必要になった⁴⁴⁾。そこで、2014年より教科「ICT」の内容が大幅に精選された、教科「Computing」が展開されることとなった。

イギリスでは国内の教育水準が一定に保たれるように、教科ごとに学習目的と目標を示す National Curriculum が規定されている。さらに、生活年齢を区分とした Key Stage (Key Stage1から Key Stage4の4段階) ごとに詳細な学習内容と到達目標が設定されている。Computing では以下のように学習目的が設定されている⁴⁵⁾。

「質の高いコンピュータ教育は、“コンピューテ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⁴²⁾。」

さらに、Computing の学習目標として以下の4点が示されている。

- ・抽象概念や論理、アルゴリズム、データ表現を含む、コンピュータサイエンスの原理と概念の基礎を理解し、応用できること。
- ・コンピューテーションに関する用語で課題を分析することができ、またそれらの課題を解決するためにプログラムを作成するという実践的経験を積んでいくこと。
- ・課題解決のため、新たな、あるいは初めて出会う情報技術を評価したり、応用することができること。

- ・責任ある有能で自信を持った創造的な ICT ユーザーとなること。

Computing の学習領域は、CS、Information Technology、Digital Literacy の3分野からなり、Computing の中心はCSである。プログラミング教育は、それ自体の習得が目的ではなく、CSを学習するためのプラクティカルワークであると考えられている。

さらに、教科「ICT」での「専門職能発達」の提供・機会が不十分であったことを鑑み、具体的な学習内容や目標については、CAS と Hodder Education 社の協力のもと作成された Computing Progression Pathways に示されている⁴⁶⁾。Computing Progression Pathways によると、Computing の学習内容として以下の6領域が示されている。

- ① Algorithms (アルゴリズム)
- ② Programming & Development (プログラミングと開発)
- ③ Data & Data Representation (データとデータ表現)
- ④ Hardware & Processing (ハードウェアと処理)
- ⑤ Communication & Networks (コミュニケーションとネットワーク)
- ⑥ Information Technology (情報技術)

これらの内容に対して、KSごとに以下の5つのコンセプトの観点から具体的な到達目標が設定されている。

- ① Abstraction (抽象化)
- ② Decomposition (構造化)
- ③ Algorithmic Thinking (アルゴリズム思考)
- ④ Evaluation (評価)
- ⑤ Generalization (一般化)

このように、イギリスでは National Curriculum の下、初等中等教育における体系的な情報教育が展開されている。我が国では校種ごとに目標や内容が設定されているのに対し、発達段階が考慮された目標や内容が明確に設定されていることが特徴である。特に、Computational Thinking は Computing の中核となる学習目的に記述されており、重要な位置づけにあることが分かる。

以上に示すように、プログラミング教育における Computational Thinking の概念について、様々な操作的定義が挙げられているが、抽象化・構造化、パターン化などの思考、アルゴリズム的思考については共通点も多い。

4.2.2 アメリカのプログラミング教育の動向

アメリカにおいても、我が国同様、CSは高等教育に始まった。World Wide Web の登場に伴い、CSが担う役割が大きく変化してきたことにより、K-12段階でのCSが議論されるようになった。2003年にはACMがA Model Curriculum for K-12 Computer Scienceを公開し、初等中等教育でのCSが展開されるようになった⁴⁷⁾。その後2005年にCSを教える教師集団としてCSTA (Computer Science Teachers Association) が設立されることとなった。アメリカでは教育に関する権限を連邦政府から州に移譲しているため、州法により教育制度が規

定されている。国家レベルでの基準が明確に設定されていないために、学校、地域による格差が問題となっていた。前述した ACM の A Model Curriculum for K-12 Computer Science の内容をアップデートし、2012年に CSTA K-12 Computer Science Standards (以下、CSTA Standards) を公開するに至った⁴⁸⁾。そこで、本節では CSTA Standards に基づき、アメリカのプログラミング教育の動向を整理する。なお、CSTA Standards では、学齢に応じたレベルとして Level1 (Grade K-3), Level2 (Grade 6-9), Level3 (Grade 9-12) が設定されている。

この CSTA Standards によると、学習内容として、以下の5領域が示されている。

- ①Computational Thinking
- ②Collaboration (協働)
- ③Computing Practice and Programming (コンピューティング実践とプログラミング)
- ④Computers and Communications Devices (コンピュータと通信機器)
- ⑤Community, Global and Ethical Impacts (コミュニティ、グローバルと倫理的影響)

Computational Thinking は、適切な方策を選択し、活用することによって、複雑な問題を概念化し、分析し、解決できるようになることを目指している。Computational Thinking が K-12の全学習者にとって重要であるのは、Computational Thinking の Abstraction (抽象化)、Automation (自動化)、Analysis (分析) といった思考法が CS の中核となる要素と考えられているからである。

また、コンピューティング実践とプログラミングのうち、コンピューティング実践はすべてのレベルにおいて CS 教育の基本と考えられている。プログラミングについては、コンピューティングが単にプログラミングを意味するのではなく、プログラミングを通して問題解決できるようになることが求められている。さらに、コンピューティング実践とプログラミングで培った CS の知識があらゆる領域で多大な影響力を持っていることを理解させることも重要な目的の1つである。

以上のことから、アメリカのプログラミング教育においては、相互に関連した内容になっているとはいえ、Computational Thinking とプログラミング教育が独立した領域となっている点は特筆すべきであろう。

4.3 Computational Thinking とカリキュラムの関連性の比較

以上のようにイギリスにおいて、Computing の目的として Computational Thinking を位置づけている。Computational Thinking の能力は単にコンピュータを使った問題解決だけではなく、人間だけで問題解決する場合にも応用できることを示しており、プログラミング場面に限らず、問題解決プロセスとして広く位置づけている。同様に、アメリカでも Computational Thinking はプログラミング教育に限定されたものではなく、CS の領域の

一部として位置づけられている。

翻って、我が国のプログラミング的思考を再考すれば、Computational Thinking に基づくプログラミング的思考はプログラミング (コーディング) に限定されており、矮小化されてとらえられているものと考えられる。今後のプログラミング教育の動向について、我が国の初等中等教育の教育現場では注視しているところである。一方で、一般家庭、民間団体ではプログラミングを教えるためのプログラミング教室に強い関心を持ち、プログラミング教育の必修化に先駆けてプログラミング市場は過熱している。このような社会情勢を受け、我が国のプログラミング教育が単にコーディングの学習として展開する方向性に導かれるのは危惧される場所である。

5. 技術リテラシーの観点に基づくプログラミング教育の検討

5.1 技術リテラシーの概念

前述したように、我が国におけるプログラミング教育の位置づけは高等教育では諸外国の CS に相当し、新たな技術を理解、活用するための基本的な資質を育成するようにカリキュラムが構成されてきた。一方で、初等中等教育においては、高等教育に倣ったものであったが、CS が普通教育に導入される過程で、指導法が確立されていないことと相まって、コンピュータの基本的操作の習熟を目指した技術活用の教育に方向転換されてきた。また、議論の取りまとめで示されたプログラミング的思考は Computational Thinking の考え方を矮小化してとらえられていることを指摘した。

CS としてのプログラミング教育はテクノロジー理解を図るためにも重要であるが、我が国のプログラミング教育でどのような資質・能力を高めるべきかについては、さらに踏み込んで議論する必要がある。

一般に、テクノロジーを学習内容とする教育は、技術教育の範疇に相当する。技術教育の役割については様々な考え方があがるが、その重要な考え方の1つに技術リテラシーの育成がある。

技術教育分野における世界最大の学会として、ITEEA (International Technology and Engineering Educators Associations) がある。ITEEA が刊行している Standards for Technological Literacy によれば、技術リテラシー (Technological Literacy) は、「技術を理解、活用、評価、管理するための能力 (p. 9)」と定義されている⁴⁹⁾。これは、今後の社会発展と技術進展は切り離すことができないことを踏まえ、市民がテクノロジーの発展に対してよりよい意思決定を行えるようになるために、技術リテラシーの重要性を示したものである。このような世界の技術教育の動向に呼応する形で、我が国においても、日本産業技術教育学会が技術教育における技術リテラシーの重要性を示している。日本産業技術教育学会では技術リテラシーを技術的素養と表現し、「技術と社会の関わりについて理解し、ものづくりを通して、技術に関する知識や技能を活用し、技術的課題を適切に解決する能力、

および技術を構成に評価、活用する能力」と定義している⁵⁰⁾。その上で、同学会は技術的素養を支える要素として、技術ガバナンス力と技術イノベーション力を位置づけている。ここでいう技術ガバナンス力とは、社会を支える技術を評価、選択、活用、管理、運用する力のことを指し、技術イノベーション力とは新しい価値の創造に向けて技術を改良、応用、創造する力を指す。この技術ガバナンス力と技術イノベーション力の二つの能力を相互に関連づけた能力を技術リテラシーとして、これらの習得こそが普通教育における技術教育の役割であると提案している⁵¹⁾。

5.2 技術リテラシーの観点に基づくプログラミング教育の課題

これらの考え方に基づけば、少なくとも普通教育でのプログラミング教育には、専門家を育成する職業教育の観点は含まれず、プログラミングの技術を学ぶことを通じた技術リテラシー育成の重要性が指摘できる。義務教育段階の小中学生や普通科の高校生を対象にプログラミング教育を行うことには、学者者にプログラミングという情報技術を理解させ、活用させるとともに、それを市民として評価、選択、管理・管理する力、自分なりの新たな発想で改良、応用、創造する力を身につけさせることに意味があると考えられる。もとより、我が国において唯一、技術リテラシーを育成する教科として存在する中学校技術・家庭科の技術分野でのプログラミング教育には、当然のことながら技術リテラシーの考え方が適切に反映されている。しかし、新たに必修化された小学校段階のプログラミング教育を技術リテラシー育成の観点から見た場合、少なくとも次の2つの課題があると指摘できる。

5.2.1 技術イノベーション力育成から見た課題

プログラミングという情報技術を理解させ、探究的に問題解決に取り組ませることには、技術イノベーション力育成の観点が相当する。これは、プログラミングによって創造的かつ論理的に問題を解決する力を育成することや情報技術に関する原理・法則の科学的な理解をもたらすものである。しかし、小学校段階のプログラミング教育では、技術リテラシー育成を目標とする教科・領域を設定せずに、いくつかの教科・領域の中に分散した形でプログラミング的思考の育成を図ることが想定されている。「プログラミング」という用語は、情報処理の手順を具体化する作業を指すものである。そのため、固有の教科・領域という枠組みを持たないまま小学校段階でプログラミング教育をその名の通りに進めると、育成できる技術イノベーション力は、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いう危険性がある。これでは、Computational Thinking の概念のような、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。

プログラミング的思考の概念が Computational Thinking の概念に比べて矮小であることの弊害は、プログラミング教育で育成しうる技術イノベーション力のレベルが、より低位な「手続きを構築する力」に留まってしまふことにあると考えられる。これが小学校段階におけるプログラミング教育の一つ目の課題である。

5.2.2 技術ガバナンス力育成から見た課題

前述したように、中学校技術・家庭科の技術分野でのプログラミング教育では、技術イノベーション力育成とともに技術ガバナンス力育成の視点が適切に含まれている。しかし、小学校段階のプログラミング教育では、「発達段階に即して、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」と述べられているものの、「プログラミングをよりよい人生や社会づくりに生かす」ことの具体は明確ではない。また、理科や算数では、それぞれの教科目標の達成というねらいのもと、論理的思考力を身に付けるための活動、電気の性質や働きを利用した道具があることをとらえる学習などにおいて、プログラミングを学習の道具として位置づけ、それを体験することに主眼が置かれている。このような取り扱いでは、生活や社会を支える情報技術の在り方について技術ガバナンスの観点から学ばせることが授業の主眼におかれることはない。小学校段階のプログラミング教育の持つ二つ目の課題は、「プログラミングをよりよい人生や社会づくりに生かす」ことの具体として、技術ガバナンス力育成の視点が明確に位置づけられていない点にあると考えられる。

5.3 今後の展望

我が国におけるプログラミング教育は、小学校、中学校、高校のそれぞれにおいていずれも異なる教科・領域に位置づけられている。そのため、普通教育として児童・生徒にテクノロジーを学ばせるという理念、すなわち技術リテラシー育成という考え方を支柱とした一貫性のあるカリキュラム編成が容易ではない。プログラミングの技術を中心にしつつ、児童・生徒に生活や社会を支える情報技術の仕組み、役割、影響を適切に理解させ、それを自分なりに活用、評価、選択、管理・運用する力を習得させるためには、それ相応の授業時間の確保と、そのこと自体を授業目標として充実に設定できる小中高一貫した教科・領域の設定が本来、必要である。これを技術教育の範囲として実施すべきなのか、CS や Computing のように技術教育とは別の教科・領域を設定すべきなのかについては多様な意見が考えられる。しかし、仮に両者を技術・情報教育という一つの枠組みで捉えることができれば、我が国のプログラミング教育に技術リテラシー育成という支柱と小中高一貫した体系的なカリキュラムを与えることができるのではないかと考えられる。

6. まとめ

以上, 本稿では, 我が国のプログラミング教育の位置づけの変遷, 海外における Computational Thinking の考え方について整理した上で, 技術リテラシー育成の観点から今後のプログラミング教育の課題を検討した。その結果, 我が国の中等教育・高等教育においては CS の考え方を踏まえた情報教育が展開されてきたが, 初等教育においてはテクノロジー理解の側面は軽視されており, その中核概念となっているプログラミング的思考の考え方は Computational Thinking の概念に比べて矮小であることが明らかとなった。その上で, 技術リテラシー育成の観点から今後のプログラミング教育の課題を検討した。その結果, 中学校段階では技術・家庭科技術分野において適切に技術リテラシー育成の観点からプログラミング教育が位置づけられているのに対して, 小学校段階のプログラミング教育には①技術リテラシー育成を主眼とする独立した教科・領域が設定されていないこと, ②育成できる技術イノベーション力が「手続きを構築する力」に留まってしまう危険性があること, ③「プログラミングをよりよい人生や社会づくりに生かす」ことの実現として技術ガバナンス力育成の視点が明確でないこと等の課題があることを指摘した。

今後は, これらの課題を踏まえて, 体系的な技術・情報教育の枠組みの中にプログラミング教育を位置づけた教育制度の在り方を検討する必要がある。また, 特に小学校段階のプログラミング教育では, Computational Thinking の概念を踏まえた技術イノベーション力の育成と, 生活や社会における情報処理の技術の在り方を評価, 選択, 管理・運用する技術ガバナンス力の育成とを適切に位置づけた授業実践の開発と展開が必要であると考えられる。

文献

- 1) Atchison, W. F. et al. (1968). Curriculum68: Recommendations for Academic Programs in Computer Science: a report of the ACM curriculum committee on computer science, *Communications of the ACM*, 11(3), pp.151-197.
- 2) 難波宏司. (2017). フィジカルコンピューティングの教育教材作成の研究. 園田学園女子大学論文集, 51, 71-91.
- 3) 文部科学省. (2014). プログラミング教育実践ガイド. http://jouhouka.mext.go.jp/school/pdf/programing_guide.pdf (最終アクセス日: 2017.6.26).
- 4) 文部科学省. (2016). 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「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)」. http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm (最終アクセス日: 2017.6.26).
- 5) 文部省. (1970). 高等学校学習指導要領(工業). <http://www.nier.go.jp/guideline/s45h/chap2-10.htm> (最終アクセス日: 2017.6.26).
- 6) 文部省. (1970). 高等学校学習指導要領(商業). <http://www.nier.go.jp/guideline/s45h/chap2-11.htm> (最終アクセス日: 2017.6.26).
- 7) 文部省. (1970). 高等学校学習指導要領(理数). <http://www.nier.go.jp/guideline/s45h/chap2-14.htm> (最終アクセス日: 2017.6.26).
- 8) 佐々木享. (1986). 情報技術科, 情報処理科の教育について. 技術教育学研究, 3, 129-159.
- 9) 文部省. (1978). 高等学校学習指導要領(工業). <http://www.nier.go.jp/guideline/s53h/chap2-10.htm> (最終アクセス日: 2017.6.26).
- 10) 文部省. (1978). 高等学校学習指導要領(商業). <http://www.nier.go.jp/guideline/s53h/chap2-11.htm> (最終アクセス日: 2017.6.26).
- 11) 文部省. (1978). 高等学校学習指導要領(理数). <http://www.nier.go.jp/guideline/s53h/chap2-14.htm> (最終アクセス日: 2017.6.26).
- 12) 文部省. (1989). 中学校学習指導要領(技術・家庭). <http://www.nier.go.jp/guideline/h01j/chap2-8.htm> (最終アクセス日: 2017.6.26).
- 13) 文部省. (1998). 中学校学習指導要領(技術・家庭). <http://www.nier.go.jp/guideline/h10j/chap2-8.htm> (最終アクセス日: 2017.6.26).
- 14) 文部省. (1998). 高等学校学習指導要領(情報). <http://www.nier.go.jp/guideline/h10h/chap2-10.htm> (最終アクセス日: 2017.6.26).
- 15) 文部科学省. (2008). 中学校学習指導要領(技術・家庭). <http://www.nier.go.jp/guideline/h19j/chap2-8.htm> (最終アクセス日: 2017.6.26).
- 16) 文部科学省. (2009). 高等学校学習指導要領(情報). <http://www.nier.go.jp/guideline/h20h/chap2-10.htm> (最終アクセス日: 2017.6.26).
- 17) 日本経済再生本部. (2013). 第6回産業競争力会議. <http://www.kantei.go.jp/jp/singi/keizaisaisei/skkkaigi/dai6/siryu.html> (最終アクセス日: 2017.6.26).
- 18) 日本経済再生本部. (2013). 日本再興戦略-JAPAN is BACK-. http://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/saikou_jpn.pdf (最終アクセス日: 2017.6.26).
- 19) 日本経済再生本部. (2016). 第26回産業競争力会議. <http://www.kantei.go.jp/jp/singi/keizaisaisei/skkkaigi/dai6/siryu.html> (最終アクセス日: 2017.6.26).
- 20) 日本経済再生本部. (2016). 日本再興戦略2016-第4次産業革命に向けて-. http://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/zentaihombun_160602.pdf (最終アクセス日: 2017.6.26).
- 21) 中央教育審議会. (2016). 幼稚園, 小学校, 中学校, 高等学校及び特別支援学校の学習指導要領等の改善及び必要な方策等について(答申). http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo0/toushin/_icsFiles/afieldfile/2017/01/10/1380902_0.pdf (最終ア

- クセス日：2017.6.26).
- 22) 文部科学省. (2017). 小学校学習指導要領. http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afiedfile/2017/05/12/1384661_4_2.pdf (最終アクセス日：2017.6.26).
 - 23) 文部科学省. (2017). 中学校学習指導要領. http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afiedfile/2017/06/21/1384661_5.pdf (最終アクセス日：2017.6.26).
 - 24) Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc..
 - 25) Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. (中島秀之(訳) (2015). 計算論的思考. 情報処理, 56(6), 584-587).
 - 26) Cuny, J., Snyder, L., & Wing, J. (2010). Demystifying Computational Thinking for Non-Computer Scientists, *work in progress*.
 - 27) Wing, J. M. (2011). Research Notebook: Computational thinking – What and Why? –. <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why> (最終アクセス日：2017.6.26).
 - 28) Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832-835.
 - 29) Mannila, L. et al. (2014). Computational thinking in K-9 education. In *Proceedings of the working group reports of the 2014 on innovation & technology in computer science education conference*, 1-29.
 - 30) Priami, C. (2007). Computational thinking in biology. *Transactions on computational systems biology VIII*, 63-76.
 - 31) Riley, D. D., & Hunt, K. A. (2014). *Computational thinking for the modern problem solver*. CRC Press.
 - 32) Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43.
 - 33) Kalelioğlu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, 4(3), 583-596.
 - 34) Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences*, 366(1881), 3717-3725.
 - 35) 磯辺秀司, 小泉英介, 静谷啓樹, 早川美徳. (2014). コンピュータショナル・シンキング. 共立出版.
 - 36) International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA). (2011). Operational Definition of Computational Thinking for K-12 Education <http://www.iste.org/docs/Computational-Thinking-documents/computational-thinking-operational-definition-flyer.pdf> (最終アクセス日：2017.6.26).
 - 37) Computing At School. (2015). Computational Thinking A guide for teachers. <http://community.computingatschool.org.uk/files/8550/original.pdf> (最終アクセス日：2017.6.26).
 - 38) Selby, C. & Woollard, J. (2013). Computational thinking: the developing definition. https://eprints.soton.ac.uk/356481/1/Selby_Woollard_bg_soton_eprints.pdf (最終アクセス日：2017.6.26).
 - 39) Sysło, M. M., & Kwiatkowska, A. B. (2013). Informatics for All High School Students. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, 43-56.
 - 40) Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association*, 1-25.
 - 41) Lye, S. Y., Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61.
 - 42) 文部科学省. (2014). 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. http://jouhouka.mext.go.jp/school/pdf/programming_syogaikoku_houkokusyo.pdf (最終アクセス日：2017.6.26).
 - 43) 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. *日本教育工学会論文誌*, 40(3), 197-208.
 - 44) Furber, S. (2012). Shut down or restart? The way forward for computing in UK schools. *The Royal Society, London*.
 - 45) Department for Education. (2013). Statutory guidance National curriculum in England: computing programmes of study. <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study> (最終アクセス日：2017.6.26).
 - 46) Computing At School. (2015). CAS Computing Progression Pathways KS1 (Y1) to KS3 (Y9) by topic. <http://community.computingatschool.org.uk/resources/1692> (最終アクセス日：2017.6.26).
 - 47) Tucker, A. et al. (2003). A model curriculum for K-12 computer science. *Final Report of the ACM K-12 Task Force Curriculum Committee*. https://www.acm.org/education/education/curric_vols/k12final1022.pdf (最終アクセス日：2017.6.26).
 - 48) The CSTA Standards Task Force. (2012). CSTA Computer Science Standards. <http://www.education>

2020.ca/Content/K-12ModelCurrRevEd.pdf (最終アクセス日:2017.6.26).

- 49) ITEEA. (2007). Standards for Technological Literacy. <https://www.iteea.org/File.aspx?id=67767&v=b26b7852> (最終アクセス日:2017.6.26).
- 50) 日本産業技術教育学会. (2012). 21世紀の技術教育(改訂). 日本産業技術教育学会誌, 54(4), 別冊.
- 51) 森山潤, 菊池章, 山崎貞登. (2016). 子どもが小さなエンジニアになる教室ーイノベーション力育成を図る中学校技術科の授業デザイナー. ジアース教育新社.