

## プログラミング教育の現状についての考察

赤堀 侃司

教育テスト研究センター・ICT CONNECT 21

### 抄録

本小論は、プログラミング教育の現状について、いくつかの文献を引用・参考にしながら、解説した資料である。公式には、小学校からプログラミング教育が必修になった記述は、文部科学省の告示である次期学習指導要領であり、それは2017年3月であるから、最近のことである。したがって、何故小学校からプログラミング教育が必修なのか、どのような活動をするのか、プログラミング言語を覚えることなのか、授業時数はどうなっているのか、機材などのICT環境は十分か、教員は指導できるのか、支援員は必要ないのか、予算は確保されているのか、など難問が山積している。そこで、本小論では、これまでの研究・資料・議論・実践・政策・諸外国の動向などを参照しながら、筆者の考えも加えて、その概要を述べたものである。我が国のプログラミング教育は、まだ途に就いたばかりであり、教育関係者にとって混とんとしていると言っても過言ではないだろう。その意味で、本小論は、関係する文献などを紹介して、資料提供することが目的である。但し、本小論で明らかになったことは、**Computational Thinking** やプログラミング的思考と呼ばれる、コンピュータ科学を基礎とする考え方が、これからの世界では、必要な資質能力として、認識されてきたと言える。

**キーワード**：プログラミング教育、Computational Thinking、プログラミング的思考、コンピュータ科学、カリキュラム、学習指導要領

### 1. 将来の職業の変化とプログラミング教育

プログラミング教育は、何故実施するのかという問いは、いつでも聞かれる。文部科学省（平成28年6月）では、人工知能などの技術の発展により、人の仕事の質が変わることを指摘し、将来どのような職業に就こうとも、その社会で求められる資質能力を身につける必要があることを、述べている。その通りであるが、この「仕事の質が変わる」、「現在の職業の半数は、人間に代わってコンピュータが代用する」という未来予測は、多くの学者が指摘しているが、ここでは、その初期の論文を紹介する。

図1で明らかのように、マニュアルでできる仕事は、コンピュータがとって代わり、創造的な仕事は、人間が行うという予測で、説得力のあるグラフとして、よく引用される。但し、今日議論されていることは、人工知能が、この人間の行う知的な、分析的な、創造的な仕事までも、やってしまうのではないかという、不安と言っても良いだろう。これには、2つの潮流があると思われる。1つは、楽観論・積極論のような議論と、他方は、期待するほどでもないという悲観論・消極論に近い議論であろう。楽観論・積極論は、メディアでも多く取り上げられているので、多くの人に良く知られているので、ここでは省略しよう。また悲観論・消極論の根拠は、これまでに、何度も人工知能が登場して、失敗しているからである。教育の分野でも、人工知能が学習システムに実装されたが、実用までいかなかった。Toy（子どものおもちゃ）の世界で、遊んでいると酷評された歴史がある。

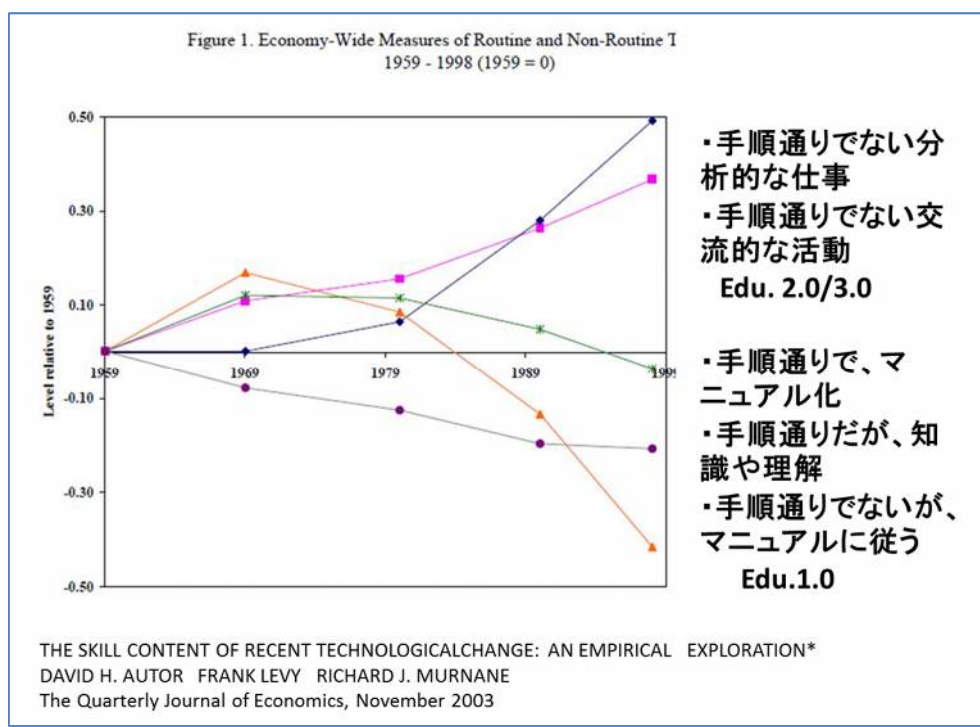


図1 将来の仕事の予測シミュレーション (David H. Autor, (2003) らによる論文の引用)

本小論は、人工知能について述べることは趣旨ではない。但し、人工知能はシンギュラリティという技術的特異点が、レイ・カーツワールによって提唱されたことから、世界の注目を集めるようになった。人間の知能を超える時が、2045年と予測したのである。人間の脳細胞の数を、ニューロコンピュータの素子が超えることは、容易であろう。現在の人工知能は、人間の脳をモデルにしたニューラルネットワークによって、判断したり、推論したりできるので、人間と同じように、学習する機能があり、その学習力を高めれば、人間の知能を超えてもおかしくないのだ。その特異点を、シンギュラリティという。シンギュラリティについては、専門的な論文よりも、多くの解説が出されている（例えば、広口正之、2014）ので、それを参照されたい。

いずれにしても、コンピュータという機械、機械というよりも、巨大な技術が、まるでSFの世界のように、現実の社会を変え始めた。それは、文字通り、巨大としか表現のしようがない。教育も、その例外ではない。将来を生きる子どもたちにとって、むしろコンピュータを知ること、きわめて大切ではないかと考えることは、不思議ではない。ここで、コンピュータを知ると書いた。これは、世界を動かしている巨大な相手を知るという意味であるが、何を知ればいいのか。どのようにコンピュータを使えばいいのか、どのような仕組みなのか、どのような分野で役立つのか、など多様にあるだろう。端的に言えば、それは、Computational Thinking（計算論的思考）（Jeannette M. Wing（著）2006、（翻訳）中島秀之、2015、）と言ってもよいし、プログラミング的思考（文部科学省、平成28年6月）と言ってもいい。先に述べた、コンピュータの使い方、コンピュータの動作、社会におけるコンピュータの役割などよりも、上記の思考法のほうが重要と考えるのは、何故なのだろうか、それを、以下述べる。

## 2. STEM教育とプログラミング教育

先の Computational Thinking について述べる前に、プログラミング教育の背景でもある STEM 教育について、触れておかなければならない。STEM とは、Science, Technology,

Engineering and Mathematics の頭文字でできていることは、容易に推測できるだろう。但し、Technology, Engineering が入っていることに、日本人は、若干の違和感を覚えるのではないだろうか。理科・数学と言え、理科系の科目という意味を理解できるが、そこに技術・工学と言われると、どこかなじまない気がするの、小中高等学校で、家庭・技術はあっても、工学はなかったし、理科・数学に比べると、重みが違うことに、子どもも大人も納得しているからである。しかし、大学の専攻では、どうだろうか。高等学校が理数系ならば、理学部を専攻するよりも、工学部とか医学・薬学系などを専攻する生徒が多く、文科系ならば、文学部の専攻というよりも、法学部・経済学部などを専攻する生徒が多い。これは、大学の専攻では、学校よりも、社会に比重が高いからである。つまり、社会とか職業とのつながりを考えて、大学の専攻を選ぶのである。

社会とのつながりを考えると、理科・数学に技術・工学を加えることに、違和感はない。日本語では、理数よりも科学技術というほうが、なじみやすい。したがって、STEM 教育は、社会との接続を考えた、学校のカリキュラムと考えるとわかりやすい。それは、理数の教科ができる生徒の育成というよりも、科学技術に優れた人材育成のほうが、なじむ。社会に出て、役立つ能力を身に付けるという意味が、包含されているからである。

アメリカのブッシュ政権とオバマ政権の元で、国際競争力を高めるために、科学技術人材育成として、STEM 教育が、重要な教育政策として取り上げられたと言われる（マルチメディア振興センター、2016年）。アメリカでは、専門誌が発刊されており、教育関係者も多いようだ（Journal of STEM Education）。日本ではまだまだ普及しているとは言い難い。埼玉大学に STEM 教育研究センターなどがあるが、その他は、大学と企業や地域と連携するアウトリーチ活動などである（大島まり、他、2015年）。

この STEM 教育が、何故プログラミング教育と関連するかが、重要な視点であるが、英国の Computer at School Working Group（以下、CAS-WG と略す、2012）は、以下のよう述べている。

「コンピュータ科学は、STEM とまったく同一の学問領域（discipline）を持っている。

- ・理論的な基礎と数学的な土台を持っており、論理的な思考を含む
- ・測定や実験に対して、科学的な方法を用いる
- ・デザイン、構築、テストなどを行う
- ・広い意味での技術を、理解し、評価し、応用する」

このように、コンピュータ科学との共通性を述べている。言うまでもなく、プログラミング教育はコンピュータ科学を学問の基礎としている。同じように、STEM 教育は、科学技術を土台としている。論理的な思考が、どちらも推論のエンジンになっていることは言うまでもない。プログラミング教育では、何かを作る、設計する、テストして修正（デバッグ）をする、ことが基本であるが、STEM 教育でも、同じである。そのためには、カリキュラム・マネジメントが、必要になる。

STEM 教育は、日本ではあまりなじみがないので、筆者がオーストラリアのケアンズ市に訪問した授業を紹介しよう（赤堀侃司、2017年 a）。写真1は、ケアンズ市内にある Edge Hill 小学校の菜園を撮影したものである。なぜ、この菜園を紹介したいかは、紹介していただいた先生が、これは STEM 教育の1つで、キッチン・ガーデン・プロジェクトだと説明したからである。子どもたちは、校庭にある菜園でいろいろな野菜を育てる。いくつかの野菜を栽培し、それで料理する。校庭には、ソーラーシステムがあって、太陽光で発電して、その電気を料理に使う。ゴミが出たら、これを菜園に戻して肥料にするという、循環システムを学習するプロジェクトであった。

栽培するには、生物や化学の知識、ソーラーシステムを理解するには、電気技術の知識、料理を手順良く行うには、プログラムする知識などのように、いくつかの分野の知識が求

められる。そこで、STEM の意味が理解できる。Science, Technology, Engineering and Mathematics で、T や E の中に、プログラムする能力も含まれている。このように考えると、プログラムすることは特別なことではなく、料理のように、目的があり、材料を揃え、手順よく作業し、味見をして、調味料を加減することなどは、Computational Thinking であり、プログラミング的思考と言ってもよい。その学問背景が、コンピュータ科学なのである。この意味で、プログラミング教育と STEM 教育は、同じ背景を持っている。



写真1 オーストラリアのケアンズ市 Edge Hill 小学校の菜園

### 3. Computational Thinking とプログラミング教育

プログラミング教育が、世界中で注目され、小学校から正規授業科目として導入されるきっかけを作ったのは、マイクロソフトのフェローで、カーネギーメロン大学の教授である Jeannette M. Wing (2006) と言われる。中島秀之が、J.M.Wing の”Computational Thinking”を、計算論的思考と訳した(中島秀之、2015年)。そこに、コンピュータ科学の基本的な考え方が整理されていて、興味深い。それは、コンピュータのプログラムを作ることが目的ではなく、コンピュータ科学者のように考える方法だと明記している。この考えは、広い概念であり、例えば、数学を学ぶのは、計算することが目的ではなく、数学的な思考法を習得することだ、とも言えるし、社会科を学ぶのは、歴史の年号を覚えることではなく、社会科的な見方・考え方を身につけることだ、とも言える。この論で言えば、すべての教科・科目の目的と同じはないかと、思われるかもしれないが、彼女は、3Rである、読み・書き・算の他に、Computational Thinking を付け加えるくらい、重要だと述べている。その理由を説明するために、いくつかの思考法の概念を述べているが、紙幅の関係で詳細は省略して、プログラミング教育と実践の中で、触れたい。ただ、3Rは、この世の中を生きていく上で必須のスキルという意味で、リテラシーと呼ばれるが、Computational Thinking は、日常生活を送る上で必須の思考法であると主張しており、その意味では、4つめのリテラシーとも言える。

日常生活における思考法、例えば、企画をする場合、何かプロジェクトを起こす場合、問題が起きた時に、実現可能な解決策を提案する場合など、この思考法は、すべてに適用できると考えられる。佐藤安紀(2017年)は、この思考法を、上記のような場合に適用して解説しているが、興味深い。この思考法の1つに、ヒューリスティックな推論があるが、コンピュータ科学になじみのない人には、少し意外な内容かもしれない。ヒューリスティックとは、発見的などの意味であるが、次の説明がわかりやすい。「発見法ともいわれる。いつも正解するとは限らないがおおむね正解する、という直感的な思考方法。理詰めで正

しい解を求める方法であるアルゴリズムと対比される概念」(コトバンク、2017年)と解説しているので、意外だという意味が、理解されるであろう。コンピュータでプログラムするとは、必ず正解にたどり着ける手順、つまりアルゴリズムだと思っている人が多いからである。

100個の数字の中から、最も大きな数字を取り出す手順は、と言われれば、その手順は容易に見つかるであろう。しかし、この世の中の問題は、そのような型にはまった問題ばかりではなく、むしろ、容易に解決する手順が見いだせない問題の方が多いことは、誰も納得するだろう。その時は、経験的に知っている方法、その方法は、偶然に発見したとか、多くの経験の積み重ねによって得られた経験則などが有効で、人はその方法、それは必ず成功することは保証できないが、その経験則を適用することが多い。人は、どうやってもうまくいかない時、経験豊かな先輩や、永く生きてきたお年寄りの知恵を借りることがある。それは、ある意味では、その人の経験や直感に信頼を置いているからである。この意味で、**Computational Thinking** が、コンピュータの世界だけでなく、日常生活に必要な思考法だという **Wing** の主張に、意味があることが理解されるであろう。

#### 4. ICT から CS への転換

情報通信技術 (Information Communication Technology; ICT) とプログラミング教育はどのような関係になっているのか、という素朴な疑問がある。これまでは、ICT教育とか ICT 利活用教育などと呼んでいたのが、プログラミング教育と言われると、同じ概念のようでもあるし、異なる概念のようでもある。教育関係者で注目されるのは、英国の2013年のナショナルカリキュラムにおいて、従来の教科「ICT」に代わって 教科「**Computing**」が新設され、2014年9月より実施されていることであろう (大日本印刷、2017年、中沢研也、2017年)。その理由は何であろうか、また、どこが異なるのであろうか。教科「**Computing**」は、我が国のプログラミング教育の独立教科版と言ってもよい。その親学問は、コンピュータ科学 (Computer Science; CS) とも言えるので、英国のナショナルカリキュラムは、ICT から CS への転換と言ってもよいだろう。この違いについて、CAS-WG は、次のように述べている (Computing at School Working Group, 2012年)。

「ICTは、コンピュータという道具の利用者 (user) になるためには、どうしたらいいかを教えるが、CSは、それらの道具の著者 (author) になることを教えることだ」と言う。ICTをいかに有効に使うか、それは、問題解決のために使うので、確かに有能な利用者なるためと言っても良い。一方、プログラミングを実行することは、自分の考えや意図することを、プログラミング言語によって表現することなので、著者になる、つまりプログラムを書く、ということも理解できよう。また、「天文学は、望遠鏡の使い方だけではないと同じように、CSは、コンピュータの使い方だけでない」という **Dijkstra** の言葉を引用して、その違いを述べている。

「ICTは、現実世界に起きている問題解決のために、コンピュータ活用する技術であり、一方、CSは、自然や人工物を理解したり探索したりするための学問 (discipline) だ」とも述べている。少し説明をしておきたい。例えば、私たちは、コンピュータやインターネットを使って、調べることを、日常的に行っている。商品という人工物でも、植物という自然でもいいが、それらを探すには、これをデータベースに登録する必要がある。データとは何だろうか、商品ならば、値段、メーカー、分類、製造日などであろうが、その組み合わせによって、検索することができる。象を撫でて、この動物は、このようなもの、と表すことと同じで、データという形で、対象を理解するのである。そして、探すときは、例えば、ある値段以下なのか、商品は何で、メーカーはどのメーカーなのか、などの手がかりが必要であるが、以下とか、一致するとか、は、記号で書けば、 $<$ 、 $=$ 、などで計算

することである。商品、メーカーなど、すべてコード化されていて、コードは、すべて 2 進数で記録されていることを思えば、商品を、すべて 2 進数で表し、それを演算することで、検索していることになる。それは、対象物を、それが自然であれ、人工物であれ、データで表し、コンピュータ内部では、2 進数で記録し、演算することで、人とやりとりをしている。その仕組みは、計算する、と言ってよい。計算という意味を、数学の計算だけの意味ではなく、対象を、表現し、演算し、出力するすべての動作原理が、「計算する」という意味と解釈してよい。CS は、その基本的な考え方を示した。考え方だけでなく、ICT という技術と結びついて、現実の問題解決に役立って、世界を変えてきたと言っても過言ではない。

このように考えると、CS は、研究 (study)、設計 (design)、実装 (implementation) して、現実の問題解決に関わる学問だという主張もうなずけよう。数学は、世界を記号で表現し、論理的な演算で処理するが、実装まではいかない。文学も、言葉という記号で表現し、科学は、実験や観察などを通して、自然の仕組みを、法則という形で表現したが、実装までは至っていない。その意味では、対象を何かの記号で表し、その背後にある世界の基本を明らかにすることは、学問と名付けていいだろう。CS は、その学問を、工学や技術と結びつけて、現実世界の問題解決をしてきた。その意味では、STEM と関連が深い。

英国のカリキュラムが、ICT から CS (Computing) に移ったことは、上記の意味において、興味深いカリキュラム改革だと言える (マルチメディア振興センター、2016 年、中沢研也、2017 年)。

## 5. プログラミング教育のカリキュラム

諸外国のプログラミング教育のカリキュラムについては、様々な文献があるので、それを参照されたい (大日本印刷、2017 年、マルチメディア振興センター、2016 年、大田剛、他、2016 年、Australian Curriculum in Queensland, 2017 年)。太田剛 (2016 年) が、英国、オーストラリア、米国を表にしてまとめているので、理解しやすい。

太田は、①コンピューテーショナルシンキング、②テストと評価、から⑩情報社会まで、11 項目について、比較表にまとめている。中でもコンピューテーショナルシンキングは、我が国のプログラミング教育に最も近い概念なので、解説したほうがよいが、ここでは、原文 (Computing At School、2015) の代わりに、太田 (2016 年) の翻訳を引用して、以下示す。

表1 英国の教科 Computing での Computational Thinking の概念

(太田剛、他、2016 年より引用)

概念	概要
抽象化	問題を単純化するため、重要な部分は残し、不要な詳細は削除する。
デコンポジション	問題や事象をいくつかの部分に、理解や解決できるように分解する。
アルゴリズム的思考	問題を解決するための明確な手順で、同様の問題に共通して利用できるものである。
評価	アルゴリズム、システムや手順などの解決方法が正しいか、確認する過程である。
一般化	類似性からパターンを見つけて、それを予測、規則の作成、問題解決に使用する。

なお、筆者はオーストラリアの Queensland 州の教育省を訪問して調査をした（赤堀侃司、2017年 a）が、太田の文献にまとめられているので、ここでは、詳細は省略したい（Australian Curriculum in Queensland、2017年）。

### 6. 我が国のプログラミング教育

我が国のプログラミング教育については、次期学習指導要領が告示されたので、その内容を示すことになるが、その前に、いくつかの動向を簡単にまとめておきたい。

教育課程は、正規の授業としての教科・科目の他に、課外活動として、例えば部活動やクラブ活動などがあるが、教育課程に位置づけられた内容については、学習指導要領に明記され、課題活動については、簡単に触れられているが、厳密な拘束力はない。そこで、教育課程に位置づけられた内容は、文部科学省が有識者会議を組織して決め、その実施についても、文部科学省が責任を担っている。また、課外活動のプログラミング教育においては、総務省が中心に進めている（総務省、2017年）。

プログラミング教育は、小学校から必修として実施することが決められている。教育課程における必修という位置づけは、重い。機材や ICT 環境や時間がないなどの理由で、実施しないということは、許されない。学習指導要領は、法的な拘束力を持つからである。そのためには、文部科学省だけでなく、民間の協力も必要で、ICT 環境の整備や課題活動のプログラミング教育との連携という観点からは、総務省の協力も必要であり、冒頭で述べた、将来の職業の変化という観点では、経済産業省の協力も必要になる。かくして、官民の連携で、このプログラミング教育を実施する方向が閣議決定されたことは、歓迎すべきことである。その官民の連携組織として、未来の学びコンソーシアムが設立された（未来の学びコンソーシアム、2017年）。その考え方などについては、佐藤安紀（2017年）が解説している。

ここでは、次期学習指導要領（文部科学省、2017年 a、2017年 b）を、現行の学習指導要領（文部科学省、2008年 a、2008年 b）との対比の上で述べて、プログラミング教育の位置づけを明確にしておきたい。その前に、現行の学習指導要領における、ICT や情報活用能力における位置づけを確認しておく。

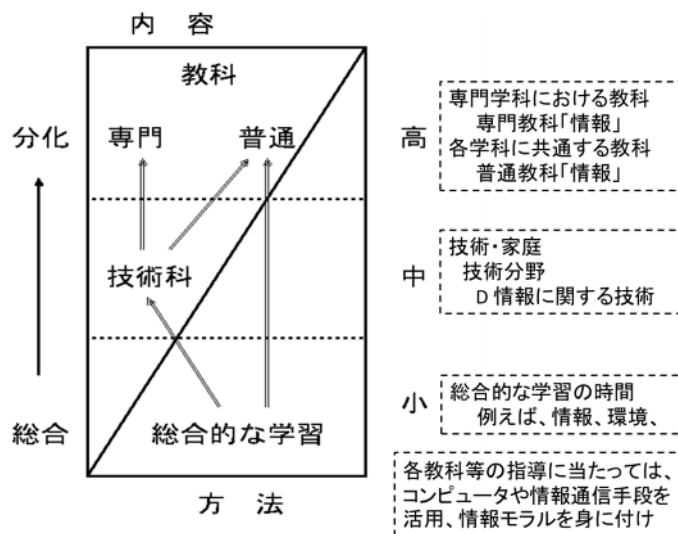


図2 現行の情報に関するカリキュラム

その枠組みを図2に示す。教育課程は、当然ながら、総合から分化に向かう。小学校低学年では、生活科であるが、3年生から理科と社会科に分かれ、中学校では、理科は、第

1分野と第2分野に分かれ、高等学校では、物理・化学・生物・地学などの科目に分かれ、より教科専門に近づいてくる。これが、図2における教育内容として示しているが、「情報」に関する内容は、小学校校では、総合的な学習の時間で、その内容は、国際理解、情報、環境などと示され、中学校では、技術・家庭科の中の技術分野の「D情報に関する技術」の中で、高等学校では、普通高校・専門高校に関係なく学ぶ普通教科「情報」と、専門高校で学ぶ専門教科「情報」に分かれる。なお、総合的な学習の時間は、小中高等学校とも学ぶので共通である。

また、総則において、「情報モラルを身に付け、各教科において、コンピュータや情報通信手段を活用し」と記されているので、これは、教育方法としてのICT利活用である。

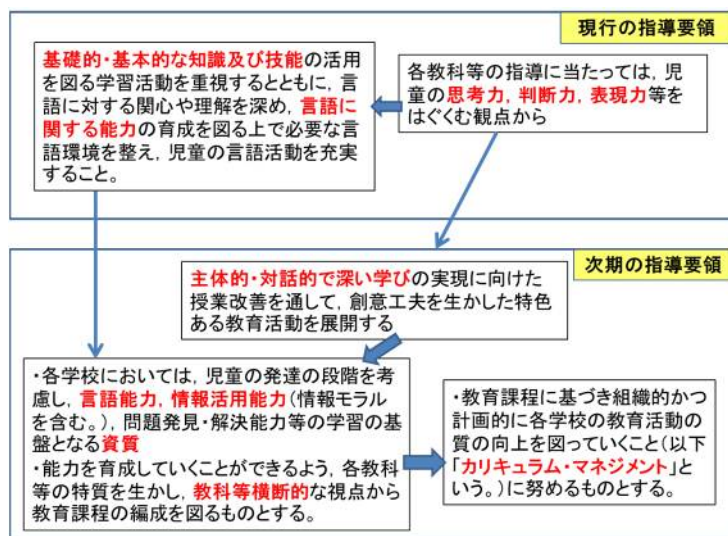


図3 現行（2008年）と次期（2017年）の学習指導要領の比較

この現行の学習指導要領（2008年告示）と、次期の学習指導要領（2017年告示）との概要の比較を、図3に示す。模式的に示すように、「思考力・判断力・表現力」が、「主体的・対話的で深い学び」に、「言語能力の育成」が、「言語能力と情報活用能力などを資質能力として」に、教科横断的な観点から、「カリキュラム・マネジメント」が、総則に明記された。

以上の比較を、「情報」という観点から、まとめた模式図を、図4に示す。なお、図2～図4は、筆者の考えによる模式図であり、文部科学省が公表している図ではないことを、お断りしておきたい。ポイントだけを可視化した図であり、全体の把握に役立つと思われるからである。



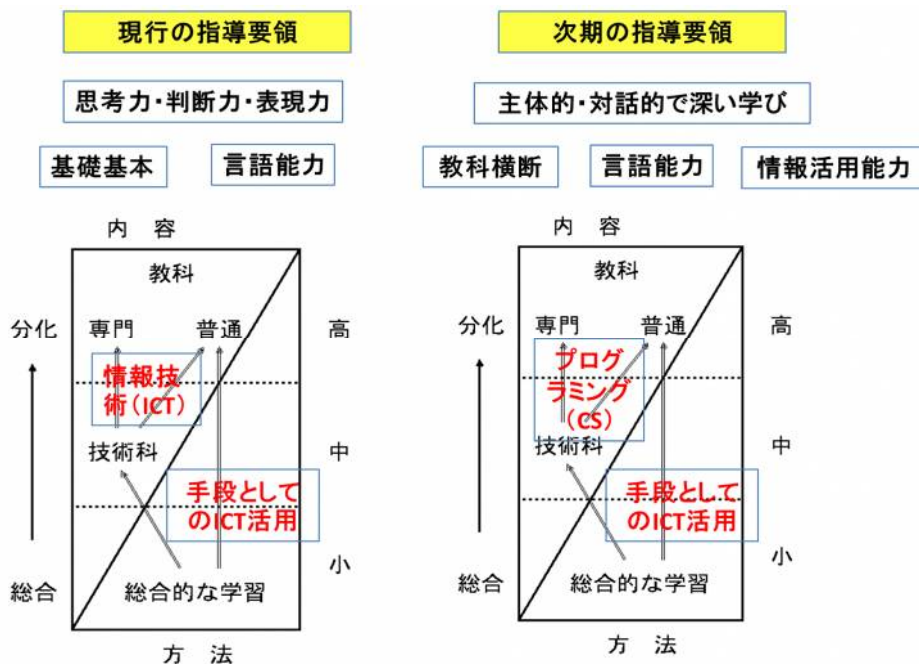


図4 現行（2008年）と次期（2017年）の「情報」に関する比較

図4において、教育方法としてICTの活用は、現行も次期も変わらない。教育内容としての情報の扱いは、これまでICT中心からプログラミングへ、つまりCS（コンピュータ科学）へ比重が移ると言える。英国のように、独立教科として、ICTからComputingへという改革ではないが、小学校からのプログラミング教育は、学習指導要領では、以下のように記述されている。

- ①「情報活用能力の育成を図るため、各学校において、コンピュータや情報通信ネットワークなどの情報手段を活用する」（小学校学習指導要領・総則）  
これは、情報活用能力（資質能力）を育成する（目的）ために、ICTという手段（教育方法）を活用するという記述である。
- ②「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」（小学校学習指導要領・総則）  
これは、プログラミング教育の記述であり、その論理的思考力の中心は、プログラミング的思考であるが、これは、次の教育実践の中で、解説したい。このプログラミングは、手段ではなく、学習活動そのものであり、プログラミング的思考、つまりその背景となっているCS（コンピュータ科学）の思考ともいえる。したがって、手段ではなく、教育内容である。
- ③「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」（小学校学習指導要領・算数・理科・総合的な学習の時間）  
特に、算数・理科・総合的な学習の時間では、必ず実施することが、明記されている。以上の概要を、図4にまとめて図示した。

### 7. プログラミング教育の教材と実践

我が国のプログラミング教育のカリキュラムは、これから開発され編成されるが、単元ごとの教材開発や実践が期待される。まだ試行段階であるが、少し紹介したい。

米国にはナショナルカリキュラムはないが、CSTA(the Computer Science Teachers

Association)は、K-12 までの CS (コンピュータ科学) のカリキュラムと評価基準を提案しており、興味深い (CSTA、2016)。教材のヒントになる内容もあるので、少し触れたい。大森康正らは、Computational Thinking を元にした情報技術教育のカリキュラムと評価基準を提案しているが、これらも参考にしながら、米国の CSTA の教育目標と教材について、参考までに紹介する (大森康正、他、2016、pp.276-277 より部分的引用)。

例えば、表 1 における、Computational Thinking 中のデコンポジション (問題の分解) では、PK-2 学年では、「学校への道順を要素化して、わかりやすい道順図を作成すること」、3-5 学年では、「学校緑化計画を立てる際、リサイクルペーパーや空き缶、電気使用量の減少などを、切り離すこと」、6-8 学年では、「ニュースレターの計画として、プロジェクトを遂行する必要な役割・義務・予定などを確認する」、9-12 学年では、「ロックスターになるには、何が必要か、という大規模問題を、スモールステップに分けること」などである。

同じく、表 1 のアルゴリズム的思考では、3-5 学年では、「ゲームをデザインして、その説明図を作り、試行すること」、6-8 学年では、「迷路を抜ける手順を考えて、ロボットに動作させる」、9-12 学年では、「大学を選ぶ意思決定プロセスを手順化して実装する」などである。

同じく、表 1 の抽象化では、3-5 学年では、「物語を聞いて、本質となる用語を反映した、タイトルを決定すること」、6-8 学年では、「歴史を勉強して、その時代を代表する、象徴・テーマ・行事・価値観などを確認すること」など、興味深い教材事例が示されている。

次に、日本における実践を紹介しよう (赤堀侃司、2017 年 b)。写真 2 をご覧いただきたい。東京都足立区西新井小学校古谷陽平教諭の実践である。この実践は、プログラミング教育を想定した授業ではない。タブレット PC を活用した総合的な学習の時間での 1 コマである。この時のテーマは、「大地震から命を守るために、できることを考えよう」であった。グループで話し合い、その考えを、付箋紙に貼って、写真のように、大地震前、ゆれている間、避難する場合、避難所に着くまで、などの場合に依じて、タブレット PC に書き込み、グループ発表した。

この地震への対応は、現実に防災教育として実施されている内容であるが、このようにいくつかの場合に分けて、どうするか、を考えることは、プログラミング的思考の考え方である。プログラムを書くとき、〇〇の場合は、「もし〇〇ならば」の条件と呼ばれ、「どうする」は、命令とか指示と呼ばれる。

グループ毎の発表があつて、質疑応答が盛んに行われた。「動かない、と言うけれど、図書館で本が棚から落ちてきたらどうするのか」などの鋭い質問が出て、その時は、こうするという修正案が出された。これは、プログラムでは、修正とかデバッグと呼ばれる。

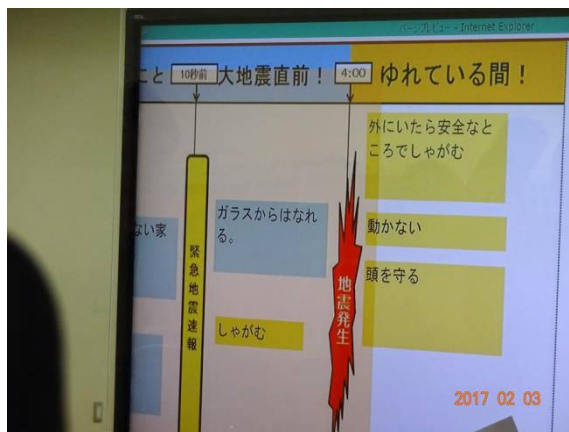


写真2 東京都足立区西新井小学校古谷陽平教諭の実践

つくば市春日学園義務教育学校の佐々木教諭による音楽におけるプログラミング教育の実践は、筆者が初めて教科におけるプログラミングの参観をした授業であった。写真3のように、曲の音符を、黒板に大きく映し出し、その特徴を子どもたちに伝える。曲と言っても、すべて音符が違っているわけではなく、いくつかの小節に分かれている。これは、文章では段落であり、全体を要素に分けること、表1のデコンポジションに相当する。さらに、曲はその小節が、何回か出てくるが、これは、プログラミングでいえば、繰り返しに相当する。例えば、ロボットが動く、と言っても、それは、右足と左足を交互に動かすことを、繰り返しているだけである。プログラムを書く場合は、この繰り返しは、必ず使う方法である。さらに、前の音符が高音に上がっていけば、次は、低音に下がっていき、逆の場合もあり、全体としての調和を目指しているが、これは、プログラミングでは、条件に応じた命令文のまとめ（サブルーチン）に相当するだろう。



写真3 つくば市春日学園義務教育学校の佐々木教諭による実践

実際の授業では、曲のいくつかの特徴を伝えながら、前時で作った曲を、もっと歌いやすいように編曲する活動だった。これは、修正（デバッグ）でもある。タブレットPCにインストールされた音楽アプリを用いて、編曲を行ったが、プログラミングの要素がすべて入っており、今でも、その光景は目に浮かぶ。確かに、プログラミングの考え方、Computational Thinking、プログラミング的思考が、音楽に含まれていることがわかるであろう。

また、課外活動では、写真4に示すように、山口市大殿小学校のロボット制御など多く

の実践がある（総務省、2017年）。この学校では、放課後に、児童の自主的な参加によって、企業や大学との連携によって、実践されたものであり、多くの成果を上げている。オーストラリアのケアンズ市の **Edge Hill** 小学校では、デジタル技術の科目の中で、ドローン制御のプログラムを学習していた。写真5に、その光景を示す。

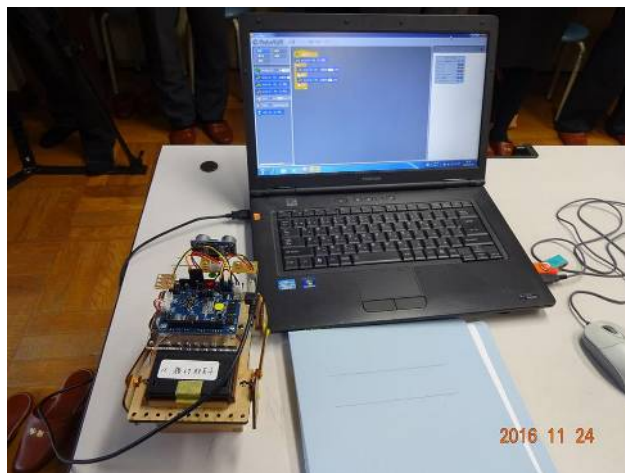


写真4 ロボットの制御（山口市大殿小学校）

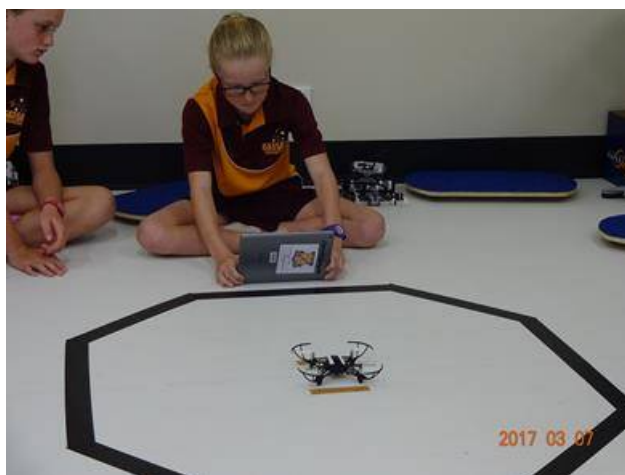


写真5 ケアンズ市 **Edge Hill** 小学校のドローン制御

以上のように、小学校の授業における教科とのクロスカリキュラムとしては、教科の目標とプログラミング教育の2つの目標を達成するために、防災教育や音楽のように教科・領域に比重をおいた内容になりやすく、課外活動や独立教科・科目としてのプログラミング教育は、ロボット制御などのように、プログラミングすることに比重がおきやすい傾向がある。教科とのクロスカリキュラムでは、45分の授業時間の中では、実際にプログラムする時間の確保がきわめて厳しいので、コンピュータを使わない **Unplugged**（電源を入れない）プログラミングの実践も多く見受けられる（中村めぐみ、2017年）。

プログラミングには、修正（デバッグ）という活動が伴い、時間がかかるので、放課後の課題活動や総合的な学習の時間などで実施することも、現実には考えなければならない。時間をかけて行った実践では、例えば、小学校6年生は、「このプログラミングの授業が、この1年間で、最も思い出に残る楽しい活動だった」と述べている（渡邊景子、他、2016年）。

このように考えると、プログラムすることは、特別なことではなく、日常生活で実践していることである。入学式のプログラムでは、始めに、開式の辞、校長先生の挨拶、などの順序がある。だからプログラムは、進行表とも呼ばれる。その順序も、来賓の挨拶、保護者代表挨拶などは、この順であって、逆ではおかしい。それは、永い間の人の知恵でもあるが、同時に、来賓と保護者という立場を考えると、そのほうが誰も納得する、つまり論理的なのである。

先の「大地震が起きたら」という場合でも、論理的に考えて対応策を考えるわけで、プログラミング教育は、プログラム言語を覚えることが目的ではなく、プログラムを作る上での基本的な考え方、つまりプログラミング的思考や **Computational Thinking** を習得することが、目的であることは、前に述べた通りである。それは、この世の中を生きる上で、必要な資質能力と言ってもよいからである。但し、プログラミングは、順序的なプログラミングだけでなく、いくつかの対象（オブジェクト）を組み合わせる全体を構成するデザインも重要で、オブジェクト指向プログラミングの考えも、諸外国のカリキュラムでは取り入れられている。

ここで改めて、プログラミング的思考について、その定義を文部科学省の有識者会議から引用する（プログラミング教育に関する有識者会議、2016年）。

「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」

米国の CSTA では、PK・2 学年で、自宅から学校までの道順の図を作る課題では、どのように行ったら安全で短時間で行こうかという意図があり、そこにはいろいろな要素、歩くこと、信号で待つこと、などがあり、これらを組み合わせ、その組み合わせは、無駄な動きがないように、論理的に組み合わせ、道順を図で表すのであるが、出来上がっても、それが完成図ではなく、振り返って、改善することになる。始めに紹介したオーストラリアのケアンズ市 Edge Hill 小学校の STEM 教育としての料理も、東京都足立区西新井小学校の地震への対策の授業も、つくば市春日学園義務教育学校の音楽の授業も、そこに、プログラミングの考え方が活かされている。

このように、プログラミング的思考とは、料理のように、目的があり、食材という要素があり、これを組み合わせ、防災教育のように、条件に応じて、要素を変え、修正をするという思考方法である。上記の定義では、これを、自分が意図する一連の活動（目的）、動きの組合せ（条件や命令）、組み合わせを改善（修正）、などを論理的に考えていく力と述べている。プログラミング的思考は、このように教科を横断する論理的な能力と言える。

## 8. いくつかの課題

以上、いくつかの文献などを参照しながら、我が国のプログラミング教育について、筆者の解説も加えながら、述べた。冒頭に述べたように、何故小学校からプログラミング教育なのだという疑問を持つ人も多いであろう。その背景に、**Computational Thinking** や、プログラミング的思考などの、新しいリテラシーが、現代社会で求められていることを、指摘した。最後にプログラミング教育を実施する上での課題について触れておきたい。

### ① カリキュラムの開発

プログラミング教育の基本が、プログラミング的思考にあることは、文科省によって明記されたが、その目標を下位目標に分解して、それぞれの下位目標を学年別に配列したカリキュラムが必要である。これまで述べてきた、諸外国のカリキュラム（太田剛、他、2016年、大日本印刷、2017年）や、我が国のカリキュラムの試案（大森

康正、他、2016年) などがあるので、参考にして開発するとよい。いくつかの教育機関も、試みている。

② カリキュラム・マネージメントの実施

我が国では、カリキュラム・マネージメントは、あまりなじみがない。それは、学習指導要領に基づいて、教科書会社や教材会社が、学校の代わりに、典型的なカリキュラムや年間指導計画を提供してきたからである。諸外国の **School Based Curriculum** で行ってきた活動とは歴史や文化が異なる。しかし、教科の目標とプログラミング教育の目標の両方を達成するためには、カリキュラム・マネージメントの実施が必要になる。

③ 年間指導計画の作成

先のカリキュラムができれば、教材や単元の目標や指導案なども記述した、年間指導計画を作成する必要がある。学校は、基本的にこの年間指導計画に基づいて、授業を行っているので、必須の活動である。これを、学校だけに任せるのは現実的でない。カリキュラム・マネージメントが、学校単位で求められているが、教員にとって未知の分野では、かなり難しい。教育委員会や教科書会社と連携しながら、作成していただきたい。

④ 指導案の作成

これも、典型的な指導案の作成が求められる。すべての単元に必要ではないが、少なくとも代表的な単元での指導案や、教師用の解説書が必要になるろう。

⑤ ICT 環境の整備

言うまでもなく、プログラミング教育の実施に伴う ICT 環境を整備する必要がある。これは、教育方法としての ICT の活用と連動して整備すればよい。しかし、これは予算を伴い、プログラミング教育に関係なく、難問と言える。

⑥ プログラミング教育の支援員（メンター）の確保

教員が不慣れなプログラミング教育では、専門的な知識や技能を持った支援員（メンター）の存在が大きい。メンターの育成については、総務省のプロジェクトが、精力的に実施しているので、参考になるろう（総務省、2017年）。地域にある大学と連携して、大学生がメンターとして小学校などに出向して、支援活動を行い、大学はそれを単位として認める仕組みが施行されている。総務省の事業では、例えば、九州工業大学が、「サービス・ラーニング」として、地域連携の新しい授業スタイルとして実施している。

⑦ 部活動の質的転換

中高等学校では、部活動がある。パソコンクラブもあるが、活発とは言えない状況が続いている。その理由は、ゲームをする、指導者がいない、甲子園のような夢のある大会がない、などが挙げられる。しかし、プログラミング教育が小学校から必修になったことを契機にして、パソコンクラブの質的転換を図ることが、期待される。このことについては、総務省も検討している。

以上、いくつかの課題を挙げたが、すべて実現可能性のある内容ばかりで、教育機関が連携して、是非解決していただきたい。

最後に、本小論は、NPO 教育テスト研究センターの支援と、科学研究費助成金・基盤研究 C（代表、赤堀侃司、課題番号 15K01034）の支援を受けたことを明記して、厚くお礼申しあげる。

参考文献

- 赤堀侃司、「ケアンズにおける情報教育」、Campus セミナー2017 in 鹿児島、17-24、2017 年 a
- 赤堀侃司、「小学校からのプログラミング教育の必修化とは」、教室の窓（北海道版）、東京書籍、2017 年 b、印刷中
- Australian Curriculum in Queensland, <https://www.qcaa.qld.edu.au/p-10/aciq>（2017 年 5 月現在）
- Computing At School (CAS) (2015), “CAS Computational Thinking - A Guide for teachers”, <http://community.computingatschool.org.uk/resources/2324>（2017 年 5 月現在）
- Computing at School Working Group, “Computer Science: A curriculum for schools”, (2012)  
<http://www.computingatschool.org.uk/data/uploads/ComputingCurric.pdf>（2017 年 5 月現在）
- CSTA(the Computer Science Teachers Association) Standards Task Force、CSTA K-12 Computer Science Standard 2016, (2016)  
[https://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/Docs/Standards/2016StandardsRevision/INTERIM\\_StandardsFINAL\\_07222.pdf](https://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/Docs/Standards/2016StandardsRevision/INTERIM_StandardsFINAL_07222.pdf)
- 大日本印刷、「諸外国におけるプログラミング教育に関する調査研究」（文部科学省平成 26 年度・情報教育指導力向上支援事業）、2017 年 3 月  
[http://jouhouka.mext.go.jp/school/pdf/programming\\_syogaikoku\\_houkokusyo.pdf](http://jouhouka.mext.go.jp/school/pdf/programming_syogaikoku_houkokusyo.pdf)
- David H. Autor, Frank Levy, Richard J. Murnane, “The Skill Content of Recent Technological Change: An Empirical Exploration”, The Quarterly Journal of Economics, November 2003
- 広口正之、「シンギュラリティとはー2045 年問題ー」JNSA Press 第 37 号（2014 年）  
[http://www.jnsa.org/jnsapress/vol37/2\\_kikou.pdf](http://www.jnsa.org/jnsapress/vol37/2_kikou.pdf)
- Jeannette M. Wing（著）（Computational Thinking, Communications of the ACM, Vol.49, No.3, pp.33-35、(Mar. 2006)）、（翻訳中島秀之、Computational Thinking、計算論的思考、情報処理 Vol.56 No.6 June 2015）  
<https://www.cs.cmu.edu/afs/cs/usr/wing/www/ct-japanese.pdf>（2017 年 5 月現在）
- Journal of STEM Education、<http://editlib.org/j/JSTEM/v/17/n/4>（2017 年 5 月現在）
- コトバンク、<https://kotobank.jp/word/ヒューリスティックス-23094>（2017 年 5 月現在）
- マルチメディア振興センター、「次世代 ICT 社会に向けた人材育成策とプログラミング教育の国際動向ー米国、英国、フィンランドにおける将来ビジョンと社会連携ー」、（一社）マルチメディア振興センター、2016 年 10 月
- 未来の学びコンソーシアム、<https://miraino-manabi.jp/>（2017 年 5 月現在）
- 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」（平成 28 年 6 月）  
[http://www.mext.go.jp/b\\_menu/shingi/chousa/shotou/122/attach/1372525.htm](http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm)（2017 年 5 月現在）
- 文部科学省、「平成 26 年度文部科学省委託事業 情報教育指導力向上支援事業、プログラミング教育実践ガイド」、2017 年 3 月  
[http://jouhouka.mext.go.jp/school/pdf/programing\\_guide.pdf](http://jouhouka.mext.go.jp/school/pdf/programing_guide.pdf)（2017 年 5 月現在）
- 文部科学省、小学校学習指導要領、（2008 年 a（平成 20 年）3 月告示）
- 文部科学省、小学校学習指導要領、（2017 年 a（平成 29 年）3 月告示）  
[http://www.mext.go.jp/a\\_menu/shotou/new-cs/\\_icsFiles/afieldfile/2017/04/27/1384661\\_4\\_1.pdf](http://www.mext.go.jp/a_menu/shotou/new-cs/_icsFiles/afieldfile/2017/04/27/1384661_4_1.pdf)
- 文部科学省、中学校学習指導要領、（2008 年 b（平成 20 年）3 月告示）

文部科学省、中学校学習指導要領、(2017年b(平成29年)3月告示)

[http://www.mext.go.jp/a\\_menu/shotou/new-cs/\\_icsFiles/afieldfile/2017/04/26/1384661\\_5\\_1.pdf](http://www.mext.go.jp/a_menu/shotou/new-cs/_icsFiles/afieldfile/2017/04/26/1384661_5_1.pdf)

中村めぐみ、「主体的・協働的な問題解決における論理的思考力の育成—小学校段階におけるプログラミング学習を通して—」、第32回(平成28年度)東書教育賞、(2017年3月)

<https://ten.tokyo-shoseki.co.jp/contest/tkyoiku/no32/nakamura.pdf> (2017年5月現在)

中沢研也、「海外のICT事情—英国の状況について—」、Campusセミナー2017 in 鹿児島、pp.25-44、2017年4月15日

大森康正、磯部征尊、山崎貞登(2016)「STEM教育とComputational Thinking重視の小・中・高等学校を一貫した情報技術教育の基準に関する日イングランド米比較研究」、上越教育大学研究紀要2016, pp.269-283

大島 まり、川越 至桜、石井 和之、「大学と企業の協働によるアウトリーチ活動を基盤としたSTEM教育」、科学教育研究、Vol. 39(2015) No. 2 pp.59-66

太田剛、森本容介、加藤浩、「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、日本教育工学会論文誌 40(3), 197-208, 2016年

プログラミング教育に関する有識者会議(正式には、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)、(2016年)「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」

[http://www.mext.go.jp/b\\_menu/shingi/chukyo/chukyo3/053/siryu/\\_icsFiles/afieldfile/2016/07/08/1373901\\_12.pdf](http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/053/siryu/_icsFiles/afieldfile/2016/07/08/1373901_12.pdf) (2017年5月現在)

佐藤安紀、「社会課題解決のために学ぶ「Computing」—動き始めた「未来の学び」—」、視聴覚教育、2107.5、pp.2-13、2017年

総務省、若年層に対するプログラミング教育の普及推進報告2017、2017年4月

<http://programming.ictconnect21.jp/> (2017年5月現在)

渡邊景子、原田康徳、「ビスケッで音楽演奏—小学校のクラブ活動での実践と今後の展開—」、日本教育工学会日本教育工学会第32回全国大会発表要旨集、pp.297-298, 2016年