

文部科学省におけるプログラミング的思考に関する 議論の過程と内容的特質

荒井陽貴¹⁾・佐藤 守^{2)*}・木下 龍³⁾

¹⁾千葉大学・教育学部・学部生

²⁾東京学芸大学連合学校・教育学研究科・博士課程

³⁾千葉大学・教育学部

Process and Content Characteristics of Discussions on Japanese “Computational Thinking” at the Ministry of Education, Culture, Sports, Science and Technology

ARAI Harutaka¹⁾, SATO Mamoru^{2)*} and KINOSHITA Riew³⁾

¹⁾Undergraduate student, Chiba University, Japan

²⁾The United Graduate School of Education, Tokyo Gakugei University (Doctor Course), Japan

³⁾Faculty of Education, Chiba University, Japan

本研究は、文部科学省におけるプログラミング教育に関する議論が行われた有識者会議および小学校部会での議論の分析を通して、小学校プログラミング教育において育む「プログラミング的思考」が位置づけられるまでの過程とその内容的特質の分析を行った。その結果、「プログラミング的思考」は第2回有識者会議でコンピューショナルシンキングと同義なものとして提言された。しかし、その会議におけるコンピューショナルシンキングの捉え方は、問題解決の手順に重きを置いたものや、コンピュータそのものを理解するものなど統一されたものではなかった。コンピューショナルシンキングと「プログラミング的思考」の相違が第3回有識者会議において指摘されたものの、その両者の関係性は十分に議論されず、各教科で育む方向性が支持される中で、コンピューショナルシンキングの物事を分解するなどの手順のみが強調され、現在の定義に至ったといえる。

キーワード：プログラミング的思考 (Japanese Computational Thinking),
コンピューショナルシンキング (Computational Thinking),
プログラミング教育 (Programming Education), 論理的思考 (logical thinking), コンピュータ (computer)

1. 研究の目的と背景

本研究は、小学校におけるプログラミング教育の必修化へと向けた文部科学省（以下、文科省）における議論の分析を通して、小学校プログラミング教育で育む「プログラミング的思考」が位置づけられるまでの過程とその内容的特質を明らかにすることを目的とする。

今後の社会は、コンピュータの飛躍的な普及や、さらなる情報社会の進展によって、変化を予想することが難しいとされる。そのような時代において、課題を発見し、情報を活用しながら問題を解決していくことが必要であるといわれている。現在でも、自動販売機やロボット掃除機など、身近な生活の中でもコンピュータとプログラミングの働きの恩恵を受けており、これらの便利な機械が「魔法の箱」ではなく、プログラミングを通じて人間の意図した処理を行わせることができるものであることを理解できるようになることは、これからの社会を生きる子どもにとって必要となる。また、こうした社会情勢に応じて、世界的にも、英国や米国をはじめとして、初等教育段階からプログラミング教育を実施している国が増えている。

このような社会を踏まえ、2017年3月に告示された小学校学習指導要領では、これまでの学習指導要領には記述されていなかったプログラミング教育に関する内容が、第1章総則、第2章3節「算数」および4節「理科」、第5章「総合的な学習の時間」において新たに加えられた。総則では、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を身につけるための学習活動を実施することが明記され、各教科においては、プログラミングを体験しながら論理的思考を身につけるための学習活動を行う例が示された。

2018年には『小学校プログラミング教育の手引（第一版）』が公開され、プログラミング教育は各教科指導の中で「プログラミング的思考」の育成を進めていく方針が掲げられた。そこでの「プログラミング的思考」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み合わせが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組み合わせをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力」と定義された。

この「プログラミング的思考」についての定義だけを見ると、「論理的思考」の一部として「プログラミング

*連絡先著者：佐藤 守 m-sato@technik.jp

的思考」が位置づくとも、「論理的思考」と「プログラミング的思考」が同一のものであるとも解釈でき、その関係性は明らかではない。

そこで、「プログラミング的思考」を提言した文科省の「小学校段階における論理的思考力や創造力、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）の議論を取りまとめた『小学校段階におけるプログラミング教育の在り方について（議論のとりまとめ）』を見てみると、「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』の考え方を踏まえつつ、プログラミングと論理的思考との関係を整理しながら提言された定義である」ことや、「各教科で育む思考力を基礎としながら『プログラミング的思考』が生まれ、『プログラミング的思考』の育成により各教科等における思考の論理性も明確となっていく」ことが記載されている。これらの記述から、「プログラミング的思考」が「コンピューショナルシンキング」の考え方を踏まえていること、そしてプログラミングと論理的思考との関係を整理されたことはわかるものの、これらの具体的な関係性や、その関係性を整理した経緯は明らかではない。

2. 先行研究の到達点と課題

「プログラミング的思考」に関する先行研究としては、藤原ら¹⁾や森山²⁾の研究がある。

藤原ら（2019）は、プログラミング教育を意味のあるものにするために、そこで育成する力をより明確にする必要があることや、「プログラミング的思考」を提言した有識者会議の議論の取りまとめでコンピューショナルシンキングの考え方との関係性について十分に言及されていないことから、「プログラミング的思考」そのものについて検討する必要があるとした。そして、「プログラミング的思考」の定義には、(1)分解、(2)アルゴリズム的に考える、(3)抽象化、(4)一般化、といったプログラミングにおける論理的思考の要素が含まれていることから、藤原らは、論理的思考の一部として「プログラミング的思考」を捉えた。

また、小学校がビジュアルプログラミング言語Scratchを使った活動の際、子どもたちが非常に創造的であったという報告や、そもそも「プログラミング的思考」を提言した会議の名称に「創造性の育成」という言葉が含まれていたことから、人はプログラミングをする際に、論理的思考のみを働かせているのではなく、創造的思考も働かせているとした。そのため、「プログラミング的思考」は論理的思考の側面だけではなく、創造的思考の側面もあわせ持っているとした。

一方、森山（2019）は、『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—』において、「プログラミング的思考」の定義に、プログラミングのように「手順を最適化するタイプの問題解決時に働かせる論理的思考」と捉えることができる点の特徴として指摘した。そして、「プログラミング的思考」と、その下地となった諸外国のコンピューショナルシンキングの概念を比較した。その結果、日本

における「プログラミング的思考」は「問題解決における手順の最適化を図る論理的思考」であり、コンピューショナルシンキングの概念よりも狭義の定義であることを明らかにした。そのうえで、「プログラミング的思考」を含むコンピューショナルシンキングを体系的に育む実践の展開が重要であると主張した。

以上のように先行研究では、「プログラミング的思考」が創造的思考を併せ持っているといった特徴や、問題解決における手順の最適化を図る論理的思考といった特徴があること、コンピューショナルシンキングと比較して狭義の概念であるといった指摘がされてきた。しかし、これらの研究は、文科省が掲げる「プログラミング的思考」の概念を前提あるいは再解釈するものであり、「プログラミング的思考」が定義された経緯については検討されていない。

そこで「プログラミング的思考」が定義されるまでの議論を分析し、定義された経緯と、その中のコンピューショナルシンキングとの関係性を明らかにすることで、より明確に「プログラミング的思考」を理解することができる考えた。

3. 研究の方法

そこで本研究では「プログラミング的思考」が定義されるまでの議論に焦点をあてる。プログラミング教育に関する議論は、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）および有識者会議で行われ、一部、中央教育審議会初等中等教育分科会教育課程部会小学校部会（以下、小学校部会）においても議論された。情報ワーキンググループでは、中・高等学校におけるプログラミング教育の内容を中心に議論された。有識者会議では、小学校におけるプログラミング教育の内容を中心に議論された。

そのため「プログラミング的思考」を提言したとされる有識者会議に加え、有識者会議での議論の内容が一部引き継がれた小学校部会も分析の対象とする。この議論の中で「プログラミング的思考」に関する議論に注目し、分析をすることで、「プログラミング的思考」に関する議論の過程とその内容的特質を明らかにする。分析には、各会議の議事録あるいは議事録要旨を用いる。議事録すべてに目を通した上で、以下のように分析を試みる。

第一に、文科省において、小学校プログラミング教育で育む「プログラミング的思考」が定義されるまでの議論の過程を明らかにする。

第二に、有識者会議で指摘された①論理的思考とプログラミングの関係性、②コンピューショナルシンキングとの関係性、という2つの視点からプログラミング的思考の内容的特質を明らかにする。

分析の対象時期は、第1回有識者会議が開かれた2016年5月3日から第3回有識者会議が開かれた2016年6月3日までとする。

4. プログラミング教育に関して議論した会議とその開催日程

プログラミング教育に関して議論した会議とその日程を表1としてまとめた。

プログラミング教育の導入は、中央教育審議会における学習指導要領改訂に向けた議論の中で検討された。中央教育審議会では、情報ワーキンググループにおいて、中・高等学校におけるプログラミング教育の内容について議論がされた。情報ワーキンググループでの主な検討事項は、①小・中・高等学校を通じた情報活用能力の育成について、②高等学校情報科(各学科に共通する教科)の改善について、③学習指導要領の理念を実現するために必要な方策について、の3点であった。2015年10月22日に第1回情報ワーキンググループが開催され、それ以降毎月1回のペースで第8回まで議論を重ねた。ワーキンググループの第7回と第8回の間には第1回有識者会議が開かれた。

7回にわたる情報ワーキンググループの議論の内容が有識者会議において共有された形跡は見当たらない。また、第1回有識者会議での議論の内容も第8回情報ワーキンググループにおいて伝えられたのみであった。本研究では、「プログラミング的思考」の議論の過程と内容的特質に焦点を当てて分析を行うため情報ワーキンググループの分析は対象外とする。

中央教育審議会での議論と並行して、有識者会議において、小学校段階における資質・能力の育成とプログラミング教育の在り方について、より詳細な議論された。有識者会議での主な検討事項は、①いわゆる「第4次産業革命」が教育に与える影響について、②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について、③効果的なプログラミング教育を実施するために必要な条件整備について、④その他、であっ

た。2016年5月13日に第1回有識者会議が開催され、5月19日に第2回有識者会議と議論を重ねた。その後、第1回、第2回有識者会議での議論の内容は、2016年5月27日の第6回小学校部会へと引き継がれ、2016年6月3日の第3回有識者会議において再び議論された。3回にわたる有識者会議の内容は「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)」としてまとめられ、中央教育審議会でのその後の議論の土台となった。その後、2016年12月21日の中央教育審議会答申を経て、2017年3月31日の小学校学習指導要領の告示に至る。

「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)」におけるプログラミング的思考の定義は、その後の『小学校プログラミング教育の手引(第一版)』においても変化はなく、現在でも有識者会議で提言された定義が用いられる。このことからプログラミング的思考の定義は、第1回から第3回までの3回にわたる有識者会議と、その間に開催された第6回小学校部会による議論を経て提言されたといえる。

5. 第1回有識者会議での議論

2016年5月13日に第1回有識者会議が開かれた。議論のはじめに、東京大学大学院特任准教授の松尾豊により、今後の人工知能の技術の変化や社会に対する影響、教育の在り方について説明がなされ、各委員にも情報が共有された。教育面については、人工知能やロボットが活用される社会となった際に、それらを使いこなせる側の人間になるための教育が重要であると説明された。

このような社会背景が共有された後、小学校におけるプログラミング教育では普遍的な力を身につけるという観点から、プログラミングの基本的な考え方を理解するために「プログラミング的な考え方」を育てていくこと

表1 プログラミング教育に関して議論した会議の日程

日 程	情報WG	有識者会議	小学校部会
2015. 10. 22	第1回		
2015. 11. 24	第2回		
2015. 12. 22	第3回		
2016. 01. 20	第4回		
2016. 02. 23	第5回		
2016. 03. 15	第6回		
2016. 04. 20	第7回		
2016. 05. 13		第1回 ⁱⁱⁱ	
2016. 05. 18	第8回		
2016. 05. 19		第2回 ^{iv}	
2016. 05. 27			第6回 ^v
2016. 06. 03		第3回 ^{vi}	
2016. 12. 21	中央教育審議会答申		
2017. 03. 31	小学校学習指導要領告示		

の重要性が提言された。「プログラミング的な考え方」を育むことについては、各委員からの賛同を得た。例えば石戸奈々子委員の「(前略)生活、社会、文化、経済、全てをコンピューターが支える社会というのが今の社会で、そして、それらはすべてプログラミングによって生まれるものじゃないかなと思います。よって、その基本的なメカニズムを習得しているということは、国語、算数と同じように、これからどんな人生を送るにしても、必要な基礎能力なのではないかなというふうに考えています。」といった発言や利根川裕太委員の「(前略)ありとあらゆる生活の場面で、コンピューターが活用されているときに、その動作の原理をわかっているというのは、21世紀を生きる社会の一構成員として子供に必要なではないかという風に考えております。(中略)この先、コンピューターによっているんなことができいくという中で、その仕組みを科学的に理解するというのが、公教育、全ての子供に対して必要なではないかというふうに考えております」といった発言があった。これらの発言によって、特定のプログラミングによるコーディングを学習するのではなく、プログラミングにつながる学習をしていくという方向性や「プログラミング的な考え方」を養うことで、コンピューターが支える社会の理解につながることが示された。

「プログラミング的な考え方」について利根川裕太委員は、コンピューターの仕組みを理解し、適切に利用するためには、「コンピューターはコンピューターとして学ぶ」ことが必要であるとした。そして、各教科の内容を深める際にプログラミングの考え方を使って学習していくことで、プログラミング教育を実施することを提案した。

この各教科の内容を深める際に、「プログラミング的な考え方」を使って学習していくことについて隅井淳一委員からは、自身が在籍するヤマハ株式会社が実施した音楽の活動の中に「プログラミング的な考え方」が潜んでいるのではないかという意見が出された。例えば、小学校の図画工作の授業で「おもちゃのPRソングを作ろう」の活動をした際に、活動の中で歌詞を創る、メロディーを創る、その作品の精度や表現力をあげることを繰り返し考えるとといったことはプログラミングと同じような考え方で表現されているとした。このように、音楽では曲の構成やそれに対する段取りをどのように組み立てるのかといったプログラミングに似た要素があり、音楽の学習で「プログラミング的な考え方」を育めるとした。また、清水静海委員からは、「すべてプログラミングでうまくいくとはかぎりませんが、何かうまい場面とか、素材によって、算数の学びの入り口とか、あるいは学んだことを学びなおすきっかけとしていくつかの機会を活かせる可能性がある」と意見が出された。このように、各教科においてもプログラミングにつながっていくような学習活動を位置づけていながら、教科学習で「プログラミング的な考え方」を育むことやプログラミングを学習することで各教科の内容を学びなおすことも考えられるとした。

一方、上野朝大委員は、基礎としてコンピューターはコンピューターとして教える段階が必要であるとした。これは、NPO法人CANVASの石戸奈々子委員によって、あ

る小学校で、全校生徒350人に1人1台Raspberry Piを配布し、全教科科目の中でプログラミングを導入している例が紹介されたことに対する意見であった。この学校では、理解の中でゴムの力を調べるシミュレーションを創ることや、算数で素数を見つけるプログラミングを書く活動や、国語で同音異義語クイズのプログラムを書く活動が実施された。これらの活動は理想ではあるものの、全国各地で普遍的に実施していくことが必要であり、実現可能性を考えた際に基礎としてコンピューターはコンピューターとして教える必要があるとした。

第1回有識者会議では、「プログラミング的思考」という言葉は使われず、「プログラミング的な考え方」を育むことについて議論された。プログラミング教育が育む「プログラミング的な考え方」は、今後の社会の変化を踏まえて、流行り廃りのあるコーディングやプログラミング言語ではなく、ありとあらゆる場面でコンピューターが活用されている社会ではその動作の原理であるプログラミングの基本的なメカニズムを理解することが必要であるとして、提言された。そして、コンピューターに対する理解が不足していると、正しく活用されないといったことや、いきなりプログラミングを用いて問題解決等を行うことは困難であるとして、基礎としてコンピューターはコンピューターとして学ぶ必要性も共有された。また、「プログラミング的な考え方」は、各教科の内容を深める際に使うことや各教科の中で手順の構成やそれに対する段取りをどのように組み立てるかといったプログラミングに似た要素を見つけて、位置づけていながら、教科学習で育むことが目指された。

6. 第2回有識者会議での議論

2016年5月19日に第2回有識者会議が開かれた。第2回有識者会議でも、各委員よりプログラミング教育で育む能力についての意見が出された。上野朝大委員は、人工知能が一般的になった社会では、人工知能を作り出せる技術力や使いこなすための理解が必要であるとした。そのためにも、プログラミング教育では、コンピューターへの深い理解やそれらを制御する能力を育成していくことを提案した。中川哲委員は、抽象化や最適化、自動化といった概念を理解することやデータの中に潜む真実を発見するといったコンピューターを理解することをアメリカではコンピューショナルシンキングという考え方で分類されており、このコンピューショナルシンキングを育成することをプログラミング教育で目指すことを提案した。このように、議論の前半では、コンピューターの理解を中心にプログラミング教育を実施していくことが提案された。

議論の後半、磯津政明委員によって、はじめて「プログラミング的思考」という表現が使われた。以下は、磯津政明の発言である。

「プログラミング教育の本質が何であるかというところ、御存じのとおり、コーディングというよりは、プログラミング的思考、先ほどもありましたが、コンピューショナルシンキングと言われているところがプログラミング教育の本質ではないかと考えておまして、この部分を

強めるのが、プログラミング教育のあるべき姿だと感じているところです。そのコンピューショナルシンキングなんですけれども、CTと海外では略されていますけれども、なかなか的確な日本語訳がないので、今日のところはプログラミング的思考ということでお話しさせていただきたいんですが、コンピューショナルシンキング自体は、コンピューターを活用して課題を解決する方法論ですね。決してコンピューターのように考えることではありません。」

磯津政明委員は、「プログラミング的思考」を海外でのコンピューショナルシンキングと同義のものとして捉え、コンピューショナルシンキングについて、コンピューターを活用して問題を解決する方法論であるとした。

続けて磯津政明委員によってコンピューショナルシンキングは①物事を分解する、②問題の法則を発見する、③問題の余計な部分を切り落として抽象化する、④アルゴリズムをデザインする、といった考え方の方法論であり、海外では4つから5つのコンセプトに分類されると説明された。

磯津政明委員は、コンピューショナルシンキングについてコンピューターを活用して問題を解決する方法論であるとする一方で、日本の算数問題を解く際に、物事を分解する、一部いらぬ部分を切り落として抽象化する、最後にアルゴリズムを作って問題を解くという形になっており、算数問題を解く過程は、コンピューショナルシンキング、「プログラミング的思考」の考え方を完全に当てはめているとした。

日本の算数では、コンピューショナルシンキング、「プログラミング的思考」はすでに内包されており、算数自体がプログラミング科目と言ってもよいのではないかといった発言から、問題解決の手順に重きを置いたものとしてコンピューショナルシンキングを捉えていたと推察される。

その後、磯津政明委員の意見に賛同するように、各委員から意見が出された。石戸奈々子委員は、コンピューターを使わないプログラミング教育の例として、イギリスの5歳児がサンドイッチを作りながら手順を考える活動を紹介した。アルゴリズムを考えることで、自らが論理的に文章を構築していきながら読解力を養うことができ、中・高へのプログラミング教育へもつながっていくとした。新井紀子委員は、コンピューターを使わないプログラミング教育の活動を例として、家から学校までの道順を箇条書きで書き出し、どこまで進んだら曲がるか、何メートル進んだら曲がるかを考えていく活動を紹介した。ほかにも、マルバツゲームのルールを書き出す活動やヤギと船頭と狼が一緒にわたることを考えるといった条件・制約の中で手順を考えることもプログラミング教育に含まれるとした。

このように、プログラミング教育で育む能力について、第2回有識者会議では、コンピューターの理解を中心に進めていく方向性とアルゴリズムを考えることで「プログラミング的思考」を育む2つの方向性が示された。

コンピューターの理解については、各委員からその方向性が述べられた。中川哲委員は、各教科の中で「プログラミング的思考」を育む方向性については賛成しつつも、

「教科に溶け込ませたときに、1つだけ溶け込まないのではないかと考えているものがコンピューターそのものの理解というところがある」と意見を出した。兼宗進委員は、「コンピューターのプログラムをすることで学べることはたくさんあると思うのですが、さっきから出ていますように、コンピューターの仕組みですね。どんなものかというのを是非理解できるような形に結び付けられるといいかなと思っております」と意見を出した。しかし、議論の中で利根川裕太委員から、学校現場は非常に多忙であり、単純にプログラミングを増やそうとすると学校では対応することができないため、他の内容と代替して行うことは可能かという疑問が投げられた。そのことに対して、小川雅裕委員は、現在の学習指導要領の中では、プログラミングの基礎・基本を教える時間を取ることは現実的に難しいとした。新井紀子委員は、プログラミングの教育の時間数を工面する方法として、現在実施している学習内容から時代的に必要のないものを削除して、スクラップ・アンド・ビルドをして実施すること、各教科の中で、アルゴリズム的な考え方がある部分を体系的にまとめ挙げ、それをプログラミングの基礎となる教育ということによって位置づけて実施することの2点を提案した。この提案された2つの可能性に対して、有識者会議で配布された事前資料の中の学習指導要領改訂の方向性(案)に、「学習内容の削減は行わない」と明記されていたこともあり、後者が支持された。

清水静海委員は「プログラミング的思考」を働かせて考えると、各教科にプログラミングにつながる要素が散らばっているとして、各教科で育む論理的思考とプログラミングをつなぐものとして「プログラミング的思考」を位置づけることを提案した。「プログラミング的思考」をこのように位置づけることで、小学校の各教科で育む論理的思考が中学校・高等学校でのプログラミングの学習につながっていくとの関係性を示した。

また、奈須正裕委員も論理的思考の育成は、既存の教科等でも行ってきたことではあるものの、プログラミング教育によってそれらのニュアンスや強調点、重要度が変わり、その構造化の筋道が変わってくるとした。そのことによって、結果的に各教科での学習がプログラミングに向かってまとめられていくため、内容を増やすことなく実施することが可能であるとした。

第2回有識者会議では、磯津政明委員から、コンピューショナルシンキングと同義のコンピューターを活用して問題を解決する方法論として「プログラミング的思考」が提言された。そして議論の中で、コンピューターの理解を中心にプログラミング教育を実施していく方向性とアルゴリズムを考えることで「プログラミング的思考」を育む2つの方向性が示された。コンピューターの理解を中心にプログラミング教育を実施していく方向性については「プログラミングの基礎・基本を教えることは時間的に難しい」といった意見が出されたことや学習内容の削減は行わずにプログラミング教育を実施するために論理的思考とプログラミングをつなぐ媒介として「プログラミング的思考」が位置づけられたことなど、制度的な問題のため実施が難しいとされた。以上のことから、第2回有識者会議では、アルゴリズムを考えることで「プロ

「プログラミング的思考」を育てていくことができ、各教科で「プログラミング的思考」を育てる方向性が支持されたとみられる。

7. 第6回小学校部会での議論

2016年5月27日に第6回小学校部会が開かれた。第6回小学校部会での議題は、①小学校段階におけるプログラミング教育に関する有識者会議について、②小学校学習指導要領総則の改善イメージについて、③総則・評価特別部会、小学校部会、中学校部会、高等学校部会のとりまとめに向けた論点(案)についての3点であった。第6回小学校部会の委員には、有識者会議の委員も一部出席しており、天笠茂主査や無藤隆委員、清水静海委員が出席した。第6回小学校部会では、大杉教育課程企画室長から第1回、第2回有識者会議での議論の内容が伝えられた。

その際、第1回有識者会議では、音楽での実践には「プログラミング的な考え方」につながる要素があるという発言であったものが、第6回小学校部会では「プログラミング的思考」につながる要素があるとして伝えられた。このことから、第1回有識者会議における「プログラミング的な考え方」は、第2回有識者会議以降では、「プログラミング的思考」として扱われたと推察される。また、第2回有識者会議では、日本マイクロソフト株式会社の中川哲委員から、抽象化や最適化、自動化といった概念を理解することでデータの中に潜む真実を発見するというコンピュータの理解をすることをアメリカではコンピュータシミュレーションという考え方で分類されており、今後の社会で必要とされると発言がなされた。しかし、第6回小学校部会では、コンピュータシミュレーションではなく、「プログラミング的な思考」が必要となると伝えられた。「プログラミング的思考」は第1回有識者会議における「プログラミング的な考え方」や海外で用いられるコンピュータシミュレーションと同義なものとして文部科学省の職員である大杉教育課程企画室長から小学校部会の各委員に伝えられた。「プログラミング的思考」については、①物事を分解して考える、②法則を発見する、③抽象化する、④解放をデザインする、⑤評価する、の5つの要素であると伝えられた。

このように大杉教育課程企画室長から伝えられた第1回、第2回有識者会議の内容について、小学校部会の委員からは賛同を得たけれども、いくつかの意見も出された。

松川禮子委員からは「プログラミング言語を教えるのではなく、社会の中でのプログラミングの役割だとか、それから論理的思考だとかの話になってくると、既存の教科の中、例えば小学校でいえば、算数だとかそういう中での論理的思考を教えることと、どう関わるのか。(中略)それから、小学校ではなくて、中学校・高校での情報教育とどう関わらせていくのかということについても関係を明確にしないと、ちょっとしっくりこないというのが率直な感想です。」と発言があり、各教科で育てる論理的思考との関係性を明確にすること、中学校・高校との学習のつながりを明確にすることが求められた。

渡瀬恵一委員からは、「具体的に5つのプログラミン

グ的思考とはということができていますけれども、このようなことがどれくらい具体的に示されて、それがそれぞれの活動の中でこの部分が伸ばされるんだということがはっきりすることや、それがプログラミングの教科から離れて教科横断的に活用されていくときに、この教科のこういう活動の部分で、この思考の部分がこう役立つんだということが明らかになっていく必要があると思います」と発言があり、5つの「プログラミング的思考」の要素がより具体的に示されて、それぞれの教科活動の中でどの部分が伸ばされるのか明確にさせることが求められた。

このようにプログラミング教育に関する意見が各委員から出され、そのことについて有識者会議にも出席していた無藤隆委員から補足的に説明がされた。無藤隆委員は、小学校におけるプログラミング教育には2つの立場があるとし、1つはプログラミング言語の子ども用に向けたある種のものを取り入れて、何時間かでそれを訓練して使えるようにしていくというもの。もう1つは、特定の言語に限定しない形でどちらかと言えばプログラミング的な思考がわかる活動を用意しようというものとした。無藤隆委員は、後者の「プログラミング的思考」を情報の中で明示化し、明確化していきながら考えていく論理的思考とし、それを育てる活動を各教科において実施していくことを支持した。この活動の具体例として、算数では文章題の論理的な構成を明らかにするために、既知の変数と未知の変数の関係を明確化することがコンピュータシミュレーションの典型であることや、国語において、物語文を理解する際に、登場人物が誰であって、物語の最初にはどの地点にいて、最終的なゴールはどこにあってという関係を図式化していくこともコンピュータシミュレーションの考え方であると説明した。無藤隆委員は、具体例において「コンピュータシミュレーション」と表現しているものの、「プログラミング的思考」とコンピュータシミュレーションを同義のものと捉えていることから、各教科の教材、単元の中で情報の流れを明確にするといった「プログラミング的思考」を育てることを進めていく方針を伝えたといえる。

8. 第3回有識者会議での議論

2016年6月3日に第3回有識者会議が開かれた。第3回有識者会議では、有識者会議(議論のとりまとめ)の案が出され、その内容に関して議論された。大杉教育課程企画室長から伝えられた有識者会議(議論のとりまとめ)案の中で、「プログラミング的思考」の定義は「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み合わせが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組み合わせをどのように改善していけば、より意図した活用に近づくのか、といったことを論理的に考えていく力」とされた。また、有識者会議(議論の取りまとめ)案において「プログラミング的思考」は、プログラミング教育で育てる資質・能力の3つの柱の中の「思考力・判断力・表現力等」に位置づけられた。

第3回有識者会議においても、各委員より「プログラ

ミング的思考」を働かせる例が出された。

磯津政明委員は「プログラミング的思考」がすべての科目に有効な考え方であり、各教科において「プログラミング的思考」を導入することは非常に重要であるとした。それに加えて、日常生活においても「プログラミング的思考」は使うことができるとした。その一例として、学校に行く際に、天気予報を見て降水確率が何パーセントであったら傘を持っていくか、何パーセント以下であったら持って行かないかという考え方ができていれば、その時点でかなり「プログラミング的」な思考ができていたとした。

有識者会議（議論の取りまとめ）案の中でも、「筆算の学習は、計算の手続きを一つ一つのステップに分解し、記憶し反復し、それぞれの過程を確実にこなしていくことであり、これはプログラミングの一つ一つの要素に対応する。つまり筆算の学習は、プログラミング的思考の素地を体験していることであり、プログラミングを用いずに計算を行うことがプログラミング的思考につながっていく」とされた。

こうした議論の流れの中、会議の後半で兼宗進委員から次のような発言があった。「重要なキーワードが個人的に曖昧なままなので、確認させていただきたいのですが、たくさん出てきていますプログラミング的思考というものについて、議論の中で出てきたのが、定義が分からないものですから、簡単に教えていただければということと、海外ではコンピューショナルシンキングというのがかなりキーワード的に使われているんですが、それと何か関連した概念なのか、それとも、全く別に、今回の議論から出てきたものなのかということ、御確認お願いいたします」

この兼宗委員の発言に対して大杉教育課程企画室長は「1回目、2回目の議論の中で、複数の先生方から、コンピューショナルシンキングの重要性を御発言いただいたことを踏まえて、それをプログラミング的思考という形で、しかも、小学校教育ということ踏まえて、少し表現を工夫しながら、御相談しながら置かせていただいているものでございます。」と回答した。また、その回答に補足する形で堀田龍也主査から「専門的な用語を使った方が明確になる一方で、それに親しみのない人には分からなくなるので、どうしても何らかの形でちょっとかみ砕いた表現にする必要があってということになりますので、これは一応努力の成果ということですので、何か御意見がありましたら、また後で付け足していただければと思います」という発言があった。

以上のやり取りから、コンピューショナルシンキングとコンピューショナルシンキングは同義であるものの、専門用語に親しみのない人に向けて表現をかみ砕いた結果、「プログラミング的思考」になったと捉えることができる。

このコンピューショナルシンキングをかみ砕いた表現としての「プログラミング的思考」について、会議の終盤に中川哲委員から次のような指摘があった。「コンピューショナルシンキングというのは、前回の私の発表の中でも用いさせていただいた言葉なんですけれども、参考情報としては、よく日本語では、コンピュータの思

考というような言い方をされます。でも、事務局の皆さんがこの言葉を使わなかったのは、『コンピュータになれと言ってるの?』というような誤解を生むのかなという懸念があるのも容易に想像できます。一方、プログラミング的思考という言葉も、耳当たりはいいんですけども、内容がコンピューショナルシンキングとイコールなのかというとそうでもなく、コンピューショナルシンキングは、アメリカでは比較的頻繁に用いられている言葉ですので、例えば、コンピューショナルシンキングの中では、データの収集・分析・表現というのはものすごく言われるんですけども、そういうところはこの取りまとめの中ではあまり触れられていません。という観点から言うと、プログラミング的思考というのは、コンピューショナルシンキングと比べて、例えば、データに関しては触れていないんですよということなのか、ここは言葉の定義をしっかりとする必要はあるのかなというのは、私も後から見て感じました」

中川哲委員は、誤解を招かないようコンピューショナルシンキングの表現を工夫することに理解は示しながらも、表現を変えたことでその2つの関係性が曖昧になっていることを指摘し、改めて「プログラミング的思考」とコンピューショナルシンキングの関係性および「プログラミング的思考」の用語の定義を明確にする必要性を主張した。しかし、会議の終盤での発言ということもあり、この点についてはこれ以上議論されることなく有識者会議は終了した。

9. 「プログラミング的思考」が定義されるまでの議論の過程

2016年5月13日の第1回有識者会議において、コンピュータが支える社会でプログラミングの基本的なメカニズムを理解するために、「プログラミング的な考え方」を育てていくことの重要性が提言された。この「プログラミング的な考え方」は、各教科の内容を深める際に使うことや、各教科の中で手順の構成やそれに対する段取りをどのように組み立てるかといったプログラミングに似た要素を見つけて、位置づけていきながら、教科学習でそれを育むことが目指された。また、「プログラミング的な考え方」はコンピュータの理解とは別で育む方向性が示され、コンピュータはコンピュータとして学ぶ必要性も共有された。

2016年5月19日の第2回有識者会議においては、磯津政明委員によって「プログラミング的思考」という表現が使われた。そこでの「プログラミング的思考」は、海外でのコンピューショナルシンキングと同義であるとされ、コンピュータを活用して問題を解決する方法論とされた。例えば、算数問題を解く際に、「プログラミング的思考」、コンピューショナルシンキングの考えを当てはめているとした。これに加えて、コンピュータを用いなくとも、アルゴリズムを考えることで「プログラミング的思考」を働かせている例が出され、各教科で「プログラミング的思考」を育む可能性が示された。ここから、問題を解く際に、物事を分解する、問題の法則を発見するなど、いくつかの要素に当てはめて表現すること

で「プログラミング的思考」を育むことができるという認識が共有された。

その後、第1回、第2回有識者会議での内容は2016年5月27日の第6回小学校部会にて報告され、「プログラミング的思考」は、①物事を分解して考える、②法則を発見する、③抽象化する、④解法をデザインする、⑤評価することであると伝えられた。

小学校部会においてもこのことに対して賛同を得たけれども、「プログラミング的思考」の5つの要素をより具体化することが求められた。

2016年6月3日の第3回有識者会議で配布された「有識者会議（議論のとりまとめ）」案において「プログラミング的思考」は、現在の定義となった。第3回有識者会議においても、プログラミング的思考が日常生活でも使えるということが強調され、順次や反復といった情報の流れを意識し、明確にすることに重きを置いた定義として提言された。

議論の終盤に、「プログラミング的思考」とコンピューショナルシンキングの関係性が不明確であるため、改めて用語の定義づけが必要であるとの意見が出されたものの、その点について十分に議論されることはなく会議は終了した。

合計3回にわたる有識者会議での結果まとめられた「有識者会議（議論の取りまとめ）」を元に中央教育審議会へと議論は引き継がれたものの、コンピューショナルシンキングとの関係性に関して議論した形跡は見当たらず、第3回に提示された「有識者会議（議論の取りまとめ）案」における定義は変更されず、現在の「プログラミング的思考」へと継承されていった。

10. 「プログラミング的思考」の内容的特質

「プログラミング的思考」が定義されるまでの議論の過程をふまえ、その内容的特質として次の2点が明らかとなった。1点目は論理的思考とプログラミング的思考の関係性、2点目はコンピューショナルシンキングとの関係性である。

(1) 論理的思考とプログラミング的思考の関係性

2016年5月19日に開催された第2回有識者会議のなかで、プログラミング教育の実施時間数をどのように工夫するかについて、新井紀子委員から発言があった。新井紀子委員は、実施可能な方法について2つの可能性を提案した。1つ目は、現在実施している学習内容から時代的に必要のないものを削除して、スクラップ・アンド・ビルドして実施すること。2つ目は、各教科の中で、アルゴリズム的な考え方がある部分を体系的にまとめ上げ、それをプログラミングの基礎となる教育ということで位置づけて実施することであった。

この提案された2つの可能性に対して、有識者会議で配布された事前資料の中の学習指導要領改訂の方向性（案）に、「学習内容の削減は行わない」と明記されていたこともあり、清水静海委員や那須正裕委員からは、「プログラミング的思考」を各教科で育むものとして位置づけることが提案された。

第2回有識者会議でのこうした議論を経て、これまでに各教科で行われてきた論理的思考の育成を、「プログラミング的思考」にあてはめて考えることで、よりその論理的思考が明示化され、プログラミングへつながるといった関係性が示された。小学校におけるプログラミング教育は、実施する時間数の関係上、各教科で実施せざるを得なかったこともあり、小学校での各教科で育む論理的思考を中・高等学校でのプログラミングへつなげる媒介として位置づけられたと推察される。

(2) コンピューショナルシンキングとの関係性

2016年5月19日に開催された第2回有識者会議において「プログラミング的思考」は、コンピューショナルシンキングと同義のものとして磯津政明委員によって提言された。磯津政明委員は、コンピューショナルシンキングについて、コンピュータを活用して問題を解決する方法論であるとした。そして、①物事を分解する、②問題の法則を発見する、③問題の余計な部分を切り落として抽象化する、④アルゴリズムをデザインする、といった考え方の方法論であり、海外では4つか5つのコンセプトに分類されると説明した。他方で、磯津政明委員は、日本の算数問題を解く際の思考の例をあげ、日本の算数問題を解く際に、コンピューショナルシンキングの考え方を当てはめているとした。そして、コンピューショナルシンキング、「プログラミング的思考」は日本の算数では、すでに内包されており、算数自体がプログラミング科目と言ってもよいのではないかと提案した。

コンピューショナルシンキングは、コンピュータを活用して問題を解決する方法論だと捉えたものの、算数問題にコンピューショナルシンキングは内包されているとの発言から、「プログラミング的思考」は問題解決の手順に重きを置いたものと捉えていることが推察される。

それに対して、中川哲委員は、抽象化や最適化、自動化といった概念を理解することやデータの中に潜む真実を発見するといったコンピュータを理解することをアメリカではコンピューショナルシンキングという考え方で分類されていると説明した。

磯津政明委員は、コンピューショナルシンキングについて、問題を解決する際に働かせる思考としたのに対して、中川哲委員は、コンピューショナルシンキングをより広く捉えており、コンピュータそのものを理解するものとした。この点から、第2回有識者会議においてもコンピューショナルシンキングが統一して捉えられていなかったことがわかる。その状況の中、小学校におけるプログラミング教育を、各教科と結びつけて実施する必要性から、各教科においてアルゴリズムを考える例が出され、コンピューショナルシンキングのなかでもアルゴリズムに重きを置いたものになったとみられる。

2016年6月3日に開催された第3回有識者会議においても、コンピューショナルシンキングをかみ砕いて表現したものが「プログラミング的思考」であると、同義である点が強調された。それに対して、表現を変えたことでその両者の関係性が曖昧になっていることが指摘され、改めて「プログラミング的思考」の用語の定義を明確にする必要性が主張された。しかし、その後コンピュ

テーショナルシンキングとの関係性を議論した形跡は見当たらない。

結果として、「プログラミング的思考」は、コンピューターショナルシンキングとの関係が十分に議論されることなく、各教科で育む方向性が明示される中で、コンピューターショナルシンキングの物事を分解するなどの手順のみが強調され、現在の定義に至ったといえる。

11. 結 論

本研究は、プログラミング教育に関する議論が行われた有識者会議および小学校部会での議論の分析を通して、小学校プログラミング教育において育む「プログラミング的思考」が位置づけられるまでの過程とその内容的特質について分析を行ってきた。その結果、以下のことが明らかになった。

2016年5月13日に開催された第1回有識者会議で、現代社会で求められるプログラミングの基本的メカニズムを理解するために、「プログラミング的な考え方」を育むことが必要であるとして議論された。全国各地で普遍的に実施していくことが必要であり、実現可能性の観点から「プログラミング的な考え方」は、各教科の中で手順の構成やそれに対する段取りをどのように組み立てるかといったプログラミングに似た要素を位置づけていきながら、教科学習でそれを育むことが目指された。また、コンピュータに対する理解が不足していると、正しく活用できないことや、いきなりプログラミングを用いて問題解決などを行うことは困難であるとして、基礎としてコンピュータはコンピュータとして学ぶ必要性も共有された。

2016年5月19日に開催された第2回有識者会議では、磯津政明委員によって「プログラミング的思考」が、コンピューターショナルシンキングと同義のものとして提言された。その際、コンピューターショナルシンキングは、コンピュータを活用して問題を解決する方法論であるとされた。また、「プログラミング的思考」は各教科でアルゴリズムを教えることで育むことができるとされた。こうした議論の中、コンピュータの理解を中心にプログラミング教育を実施する方向性とアルゴリズムを考えることで「プログラミング的思考」を育む2つの方向性が示された。前者に対して、プログラミングの基礎・基本を教える時間を確保することは難しいといった意見や、制度的な問題からコンピュータの理解を中心にプログラミング教育を実施することは難しいとする意見が出てきたことで、アルゴリズムを考えることで「プログラミング的思考」を育む方向性が支持された。

2016年5月27日に開催された第6回小学校部会では、「プログラミング的思考」について、①物事を分解する、②法則を発見する、③抽象化する、④解法をデザインする、⑤評価する、の5つの要素であると伝えられた。小学校部会において、このことに対して賛同を得たけれども、「プログラミング的思考」の5つの要素をより具体化することが求められた。

2016年6月3日に開催された第3回有識者会議において、「プログラミング的思考」は現在の定義となった。

この会議においても、「プログラミング的思考」が日常生活でも使えるということが強調され、順次や反復といった情報の流れを意識することに重きをおいた定義として提言された。

論理的思考とプログラミングの関係性については、第2回有識者会議において整理がなされた。清水静海委員、奈須正裕委員は、「プログラミング的思考」を論理的思考とプログラミングをつなぐ媒介として位置づけた。論理的思考の育成はこれまでの教科等でも行ってきたものの、①物事を分解する、②問題の法則を発見する、③問題の余計な部分を切り落として抽象化する、④アルゴリズムをデザインするといった「プログラミング的思考」の要素に当てはめて考えることで、よりその論理的思考が明示化され、プログラミングへとつながるとの関係性が示された。

コンピューターショナルシンキングと「プログラミング的思考」の関係性については、まず第2回有識者会議において、その2つは同義なものとして提言された。ただし、第2回有識者会議においてコンピューターショナルシンキングは、問題解決の手順に重きをおいたものという捉え方やコンピュータそのものを理解するものという捉え方など、統一されたものではなかった。小学校におけるプログラミング教育は、各教科と結びつけて実施する必要性から、アルゴリズムに重きをおいたものとなつたとみられる。第3回有識者会議においても、「プログラミング的思考」はコンピューターショナルシンキングをかみ砕いて表現した同義のものであるが強調された。それに対して、表現を変えたことでその両者の関係性が曖昧になっていることが指摘され、改めて「プログラミング的思考」の用語の定義を明確にする必要性が主張された。しかし、その後コンピューターショナルシンキングとの関係性を議論した形跡は見当たらない。

結果として、「プログラミング的思考」は、コンピューターショナルシンキングとの関係が十分に議論されることなく、各教科で育む方向性が明示される中で、コンピューターショナルシンキングの物事を分解するなどの手順のみが強調され、現在の定義に至ったといえる。

脚 注

- i 藤原伸彦、阪東哲也、曾根直人、長野仁志、山田哲也、伊藤陽介「ティンカリングとしてのプログラミング」、鳴門教育大学情報教育ジャーナル、No. 116, pp. 21-26, 2019
- ii 一般社団法人日本産業技術教育学会『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—』九州大学出版社、2019
- iii 文部科学省(2017)「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」https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/gijiroku/1382201.htm(最終閲覧日:2021年10月29日)
- iv 文部科学省(2017)「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」<https://>

- www.mext.go.jp/b_menu/shingi/chousa/shotou/122/gijiroku/1382200.htm (最終閲覧日：2021年10月29日)
- v 文部科学省 (2017)「教育課程部会 小学校部会 (第6回) 議事録」 https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/074/siryu/1382091.htm (最終閲覧日：2021年10月29日)
- vi 文部科学省 (2017)「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」 https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/gijiroku/1382219.htm (最終閲覧日：2021年10月29日)