

教科学習を横断するプログラミング的思考のパターン

Patterns for Cross-Curricular Competencies through Programming

星 千枝^{*1,*3}・後藤 義雄^{*1,*3}・小田 理代^{*1,*3}・永田 衣代^{*1,*3}・赤堀 侃司^{*2,*3}

ベネッセコーポレーション^{*1}・ICT CONNECT21^{*2}・教育テスト研究センター^{*3}

小学校新学習指導要領でプログラミングが必修化された。我が国では、英国のようにプログラミングの教科が新設されたのではなく、従来の教科学習においてプログラミング的思考を育むことになっている。限られた授業時数の中で、教科のねらいをふまえながら、その学びを深めるために、どのようにプログラミング的思考を導入すればいいのか。教科のねらいとプログラミングのねらいの両方が相乗効果を生み、より教科理解が進むような教材とは何なのか。そのような問題意識のもと、本研究では、教員が従来の教科学習にプログラミング的思考を導入する観点を明らかにするために、文部科学省のプログラミング的思考の定義に合わせて評価規準を作成し、それをもとに、小学校の現職教員とともにプログラミング的思考を導入した教科学習の教材開発を試みた。開発された教材や諸外国の教材事例を分析した結果、教科学習に取り入れやすいプログラミング的思考のパターンとして7つの分類を提案する。

キーワード：教科学習、学習指導要領、プログラミング的思考、資質・能力、教材

1. はじめに

2020年施行の小学校新学習指導要領では、教科の中でプログラミング的思考を育むことが必修化された。本稿では、なぜそのような改訂になったのか、教科の学びを深めるためにプログラミング的思考をどのように導入すればいいのかを考察し、教科学習に適したプログラミング的思考のパターンを整理したものである。

2. 背景

2.1. 社会で求められる力の変化

テクノロジーの進化やグローバル化が進む現代では、マニュアル化された仕事はコンピュータに代替され、人間には、より新しい価値やモノを創り上げていく高次な能力が求められるようになっている（F. Levy 2006）。このような時代の変化をふまえ、2000年初めにOECDは、DeSeCoのキーコンピテンシーを発表し、知識の活用能力や問題解決力を測定する国際調査PISAをスタートさせた（CRET 2010）。我が国でも2007年度から全国学力・学習状況調査が始まり、基礎的・基本的な知識・技能を“活用”できるかどうかをみるB問題が導入されたのは、このような国際動向をふまえてのことである。

2.2. 我が国の新学習指導要領

新学習指導要領では、小学校、中学校、高校を通じて、教科の知識をより深く理解し、未知の課題に適用するのに必要な思考過程を重視し、従来の知識ベースではなく、資質・能力ベースになった。高等学校指導要領の総則にも、「教科等横断的な視点に立った資質・能力の育成」として、「(1) 各学校においては、生徒の発達の段階を考慮し、言語能力、情報活用能力（情報モラルを含む。）、問題発見・解決能力等の学習の基盤となる資質・能力を育成していくことができるよう、各教科・科目等の特質を生かし、教科等横断的な視点から教育課程の編成を図るものとする。(2) 各学校においては、生徒や学校、地域の実態及び生徒の発達の段階を考慮し、豊かな人生の実現や災害等を乗り越えて次代の社会を形成することに向けた現代的な諸課題に対応して求められる資質・能力を、教科等横断的な視点で育成していくことができるよう、各学校の特色を生かした教育課程の編成を図るものとする。」と記載された（文部科学省 2018a）。教科ごとに教員が異なる高校段階においても、教科等横断的な視点から資質・能力の育成が提唱されているのである。1人の担任が複数教科を指導する小学校段階なら、より教科横断的に資質・能力を育成しやすいのではないかと思われる。

未知の問題を解決するには、情報を広く収集し、解

決に必要な情報だけを取り出し、その情報を組み合わせ、適切な言語や数式・記号で手順にする力が必要になる。手順の組み合わせ方は一通りではなく、目的に応じて柔軟に判断することが求められ、コンピュータに代替されない力である。複数の教科で学んだ知識を課題解決に適切に適用できる力が、現代社会に生きる児童・生徒に必要な力なのである。

2.3. プログラミング的思考育成の意義

文部科学省が2016年度に設立した「小学校段階における論理的思考力や創造性、問題解決能力の育成とプログラミング教育に関する有識者会議」の名前が示すとおり、プログラミング教育は問題解決能力と明確に関係付けて検討された（太田 2016）。そして、プログラミングの活動で育成されるべき「思考力・判断力・表現力等」として「プログラミング的思考」が提言された。それは、「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」と定義された（文部科学省 2016a）。

小学校におけるプログラミング教育のねらいは、①「プログラミング的思考」を育むこと、②プログラムの働きやよさ、情報社会がコンピュータ等の情報技術によって支えられていることなどに気付くことができるようにするとともに、コンピュータ等を上手に活用して身近な問題を解決したり、よりよい社会を築いたりしようとする態度を育むこと、③各教科等での学びをより確実なものとするものの三つとすることができる（文部科学省 2018b）。

我が国では、思考力は教科内容の習得に付随する形で教授されることが多い。一方アメリカでは独自の思考教授の理論や方法が考えられている。思考教授の教材の特徴は、知識を必要としないこと、オープンエンドで多様な解が出ること、思考技能の要素が明確化されていること、である（樋口 1998）。プログラミング的思考は、後述のとおり、思考教授の特徴に似ていることから、単一教科内容の習得に付随しない思考ととらえていいのではないか。

プログラミング的思考は、まず問題を明確にし、解決のための要素を抽象化し、それらの組合せや実行手順を言葉で表現する【設計段階】と、言葉で表現した手順をコンピュータが動くような命令に翻訳する

ことで、命令通り正しくコンピュータが動くかどうかで命令の正誤を確かめられる【手続き段階】がある。設計したことを手続きにするという一連の活動でコンピュータが動かなければ、思考した命令が伝わらなかったということが、明確になるのである。従来の教科学習では、思考過程を文章や図、記号や式で書いたり、言葉で述べたりすることで思考力があるかないかが評価されている。思考力の程度がそれらの表現の程度に左右されてしまうことも多い。思考できているのに、手が遅かったり、図示が下手だったり表現できないと、思考していないと思われる。

発達段階に応じて、翻訳する方法は適宜変えればいい。例えば、1年生なら命令どおりに人を動かす活動に変えることもできる。人やコンピュータが思った通りに動かないことがわかれば、設計段階のどの命令が不十分だったのかを振り返って修正する。言い換えると、問題発見⇒計画⇒実行⇒振り返りという問題解決の流れが、プログラミング的思考の枠組みであり、それは教科内容によらないものなので、未知の問題や内容に適用しやすく汎用的で、育成の意義が高いといえるのではないか。

また、プログラミング的思考の設計段階は国語や社会と相関が高く、手続き段階は理科や数学と相関が高い（赤堀 2018）という結果から、教科横断で育成される力といえそうである。

2.4. プログラミングで育成する資質能力の評価規準

ベネッセでは、英国で新しいナショナル・カリキュラムに新設された教科「Computing」の中核となるコンピューテーション・シンキングの考え方や、米国CSTAの「Computer Science Standards」、高校情報教科書などを参照し、文部科学省の資質能力の三本柱（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）に合わせて、小学校段階での「プログラミングで育成する資質能力の評価規準＜試行版＞」を開発した（ベネッセ 2017）。評価規準では、次に述べる各項目に対して、小学校低学年、中学年、高学年の各段階に対応した評価規準が設定されている。

2.4.1 知識・技能

文部科学省は、「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」と2つのことを定義している。ベネッセでは、表1のようにブレイクダウンした。

表1 知識・技能

知識	・プログラムの存在を知る ・コンピュータがプログラムにより様々な動作を知ることを知る。 ・処理の自動実行の意味を知る。 ・プログラムは、順次、繰り返し、条件分岐という処理の組合せで構成されていることを知る。 ・処理の手順を表現する方法を知る（フローチャート等）。 ・様々なプログラミング言語があることを知る。 ・変数の考え方を知る（データ表現とデータ型）。等々
技能	・文字入力ができる。 ・簡単なプログラムを実行できる。 ・プログラムからコンピュータの動きを想像できる。 ・センサーやアクチュエータなどを使って、プログラムによる簡単な計測・制御の実験が行える。 ・並べ替えや整列などのアルゴリズムをプログラミングによりシミュレーションして比較できる。等々

2.4.2 思考力・判断力・表現力等

文部科学省は、「発達の段階に即して、『プログラミング的思考』を育成すること」と定義している。ベネッセでは、2.3.で引用したプログラミング的思考の文部科学省の文言に対応させて、表2のように整理した。

表2 プログラミング的思考

動きに分ける	自分が意図する一連の活動を実現するために、大きな動き（事象）を解決可能な小さな動き（事象）に分割すること。いわゆる分割。
記号にする	分解した動き（事象）の適切な側面・性質だけを取り出して他の部分を捨てること。いわゆる抽象化。
一連の活動にする	記号（動き）の類似の部分特定して、別の場合でも利用できる内容にすること。いわゆる一般化。
組み合わせる	同様の事象に共通して利用できる明確な手順を創造すること。
振り返る	目的に応じて、必要十分な評価の観点を考え、実行したことが、意図した活動に近づいているかどうか評価すること。
論理的に考えを進める	論理的推論と分析を行うこと。

2.4.3 学びに向かう力・人間性等

文部科学省は、「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」と定義している。ベネッセでは、表3のように具体化した。プログラミングに

限らない態度ともいえるが、例えば、挑戦するとは、Tinker（いじくりまわす）を意味しており、新しいツールでも恐れずに試行錯誤して使いこなそうという態度を示している。

表3 学びに向かう力・人間性等

挑戦する	新たなことでも、ひるまず試して触ってみる態度を養う。経験して取り組みの素地を作ろうとする態度を養う。
やり抜く	目標に向かって、粘り強く、寛容な心と強い意志をもってやり抜く態度を養う。
協働する	他者を尊重し、他者と一緒に創造しようとする態度を養う。
創造する	新しいものや価値を創り出そうとする態度を養う。
改善する	目標と合うかどうかを吟味・評価しながら必要な改良を行う態度を養う。

3. プログラミング的思考を導入した教科学習

3.1. 本研究の目的

以上の背景から、本研究の目的は、小学校の教員が従来の教科学習にプログラミング的思考を導入できるように、その観点を示すことである。プログラミング的思考は、内容によらない思考の枠組みなので、その枠組みをビジュアル言語でテンプレート化し、テンプレートの素材を該当の教科内容に変えるだけで、教科横断で同じプログラミング的思考の育成ができることをねらっている。

3.2. 研究方法

日本の小学校での授業実践事例、および、公開されている英国、オーストラリア、米国の教科に関連したコンピューティングの教材事例を参照し、それぞれの教材のプログラミングのねらいごとにまとめ直し、教科学習に導入しやすいプログラミング的思考を帰納的に整理する。その際、ベネッセの評価規準や3か国のコンピューティングの概念の整理も参考にした。3か国の資料を参照したのは、全国レベルで初等教育初年度からプログラミングのカリキュラムを保有しているからである（太田 2016）。

3.2.1. 日本の小学校での実践事例

ベネッセでは、大阪市、戸田市、柏市の小学校教員と共に、プログラミング的思考を導入した教科学習の授業を考案し、指導案を開発した。また公開されて

いる滋賀県、相模原市、水戸市の小学校の授業案も参考にした。学年別の主な教科と内容は、表4のとおりである。

表4 プログラミング的思考を導入した教科学習

学年	教科と内容
1年	算数「色板でかたちづくり」 体育「たいそうのうごき」
2年	音楽「リズムづくり」 生活「めいれいのじゅんぱん」
3年	音楽「音のひびきをかんとろう」 理科「電気を通すもの、磁石につくもの」 理科「昆虫を調べよう」 算数「かけ算の筆算の仕方」 算数「三角形」 総合「くりかえしのあるめいれい」
4年	社会「土地のようすと人々のくらし」 算数「概数（四捨五入）」 算数「空間内の点の位置の表し方」 理科「水の変化」 総合「思い通りにキャラクターを動かす」 国語「発表名人をめざそう」
5年	国語「物語を台本にしよう」 理科「植物の発芽と成長」 算数「正多角形の性質」 算数「小数のわり算」 総合「そうじロボット」 算数「合同な図形」
6年	家庭科「栄養バランスのとれた献立」 総合「住む町の課題解決をしよう」 理科「センサーを使って省エネしよう」 理科「水溶液」 算数「速さ」 総合「おしゃべりロボット」 家庭科「買い物の達人になろう」 学級活動「そうじの仕方を伝える」

3.2.2. 3か国の教材事例

英国の教科におけるコンピューターサイエンス (Maths and Computer Science, Language and Computer Science, Crafts and Computer Scienceなど) の教材事例(CAS 2018a)を見ると、二進数を用いた手品や、ラスタ形式（ビットマップ描画）やベクタ形式の描画の仕組み、英単語パズル、ソートアルゴリズム体操、編み物の仕組みなどが紹介されていた。

オーストラリアでは、デジタル技術科目 (Digital Technologies) で、英国のようなコンピューテーショナルシンキングを学習する (ACARA 2015a)。就学準備学年 (Foundation) ～2年、3～4年、5～6年、7～8年、9～10年の5つの学年帯別にICTリテラシーとコンピューティングのスタンダードと、教材事例 (Work Sample) が3段階の評価基準別に提示されている。5～6年で二進数を扱っている。

米国では、K～2年、3～5年、6～8年、9～10年、11～12年の5学年帯ごとにスタンダードが決められている (CSTA 2017a)。

上記のそれぞれの教材事例の中から、日本の小学校でプログラミング的思考を導入した教科学習にも適すると思われるものを表5にまとめた。

表5 3か国の教材事例

国	教科と内容
UK	国語「定型あいさつ文のアルゴリズム」 算数「偶数と奇数のカード手品」 算数「座標指定による描画」 家庭科「編み物の仕組み」
AUS	F～2年「ビーチにあるものを数えよう」 F～2年「めいれいしてBee-Botを動かそう」 3～4年「データに基づき、学校をきれいにする」 3～4年「レゴタワーの上にいるキャラを助けよう」 5～6年「かけ算わり算ゲームのボタン操作」 5～6年「簡単なゲームの仕様書（フローチャート）」
USA	K～2年「データの収集と削除」 K～2年「日課の手順と修正、反復」 3～5年「データどうしの関連付け、因果関係」 3～5年「順次、反復、条件制御、修正」 6～8年「ツールによるデータ収集とクリーニング」 6～8年「課題の要素分解と設計、変数の利用」

3.2.3. 3か国の情報教育カリキュラムの概要

文献（太田2016）では、英国(CAS)による教科 Computingでのコンピューテーショナルシンキングの概念5分類とオーストラリア (ACARA) によるデジタル技術科目のキーコンセプト5分類が表に要約されている。米国 (CSTA) のスタンダードは2017年に更新され、コンセプトとプラクティスで整理された。コンセプトは5つで構成され、コンピューティングシステム、ネットワークとインターネット、データと分析、アルゴリズムとプログラミング、コンピュー

ティングの威力である。プラクティスは8つから成り、包括的なコンピューティング文化の育成、コンピューティングに関する協働、コンピューティングの問題発見と特定、抽象化、アプリの創造、アプリの評価と改良、コンピューティングに関わるコミュニケーションである。コンセプトとプラクティスは相互に関わるものなので、一概に、それぞれが内容と思考に対応するわけではないようだ。例えば、英国とオーストラリアは抽象化をコンセプトに分類しているのに対し、米国はプラクティスに入れている。

我が国のプログラミング的思考に関連のある項目のみを抽出して、類似するものを寄せて表6に集約した。

表6 3か国のコンセプト【プラクティス】の集約

コンセプト	概要	国
問題発見と特定【プ】	—	USA
抽象化	問題を単純化するため、重要な部分を残し、不要な詳細は削除する	UK
	詳細を隠し、処理可能な数の特徴を扱う	AUS
抽象化【プ】	—	USA
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	UK
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。	UK
特定、アルゴリズムと実装	問題を特定し、その解決方法を考え、実行する	AUS
アルゴリズムとプログラミング	—	USA
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	UK
アプリの評価と改良【プ】	—	USA
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	UK
データ収集、表現と解釈	データを収集や表現し、成果物の文脈において解釈する	AUS
データと分析	—	USA

(CSTA 2017a) 更新し、文献(太田 2016)を再構成

3.3. 結果

3.2.で述べたことをもとに、教科学習に導入するプログラミング的思考のパターンを表7にまとめた。

表7 プログラミング的思考の7つのパターン

	パターン名	内容	教材例
1	分類	条件にそって分類する	理科「水溶液」 算数「三角形」 理科「電気を通すもの、磁石につくもの」 理科「昆虫を調べよう」 理科「植物の発芽と成長」
2	順次	手順を考える	算数「色板でかたちづくり」 算数「合同な図形」 体育「たいそうのうごき」 生活「めいれいのじゅんばん」 音楽「音のひびきをかんじとろう」 算数「空間内の点の位置の表し方」
3	反復	規則性を見出す	算数「正多角形の性質」 音楽「リズムづくり」 総合「くりかえしのあるめいれい」
4	計算モデル	コンピュータが動く仕組みを考える	算数「偶数と奇数のカード手品」 算数「概数(四捨五入)」 算数「かけ算の筆算の仕方」 算数「小数のわり算」 社会「都市の地理的特徴」
5	評価(テスト)	考え得る全ての可能性を確認する	社会「土地のようすと人々のくらし」 総合「思い通りにキャラクターを動かす」 総合「そうじロボット」 理科「センサーを使って省エネしよう」 家庭科「栄養バランスのとれた献立」
6	構造	全体像を想定して設計する	国語「発表名人をめざそう」 国語「定型あいさつ文のアルゴリズム」 理科「水の変化」 国語「物語を台本にしよう」
7	データ	条件を満たすデータを抽出する	F～2年「ビーチにあるものを数えよう」 3～4年「データに基づき、学校をきれいにする」

表7の教材例の欄を見るとわかる通り、7つのプログラミング的思考のパターンは、内容によらず、教科横断で思考のベースとなるものである。これらの思考パターンを教科横断で明確化することで、思考の枠組みが形成され、教科の深い学びにつながるのではないかと期待される。

次にひとつひとつの事例を紹介する。

3.3.1 分類

ものごとを分解(Decomposition)して抽象化し、条件にそって、分類する思考パターンである。

表8は水溶液の性質ごとの判別方法を表している。

表8 水溶液の性質

	性質1	性質2
	酸性かアルカリ性か中性か	臭いがあるかないか
判別方法	リトマス紙 酸性(青→赤), アルカリ性(赤→青)	臭いをかぐ
食塩水	中性	ない
アンモニア水	アルカリ性	ある
塩酸	酸性	ある

リトマス紙では色の変化で3通りの水溶液を分類することができる。これをテンプレートにプログラミングしたものが図1である。



図1 水溶液の分類

表9 三角形の性質

	性質1	性質2	性質3
	辺の長さは ぜんぶ同じ	角の大きさは 全部同じ	直角がある
正三角形	○	○	○
二等辺三角形	○	×	×
直角三角形	×	×	○

表10 電気と磁石の性質

	性質1	性質2
	電気を通すか	磁石につくか
判別方法	電気	磁石
プラスチック	×	×
アルミ	○	×
鉄	○	○

図1のテンプレートは、性質によって対象を分類する思考パターンを表している。よって、表9や表10の内容も同じテンプレートで分類できるのである。実験でできないことも、プログラミングすればパソコン上でいろいろシミュレーションができるのが利点である。

同じ対象でも、分類条件によって分類の仕方は多様である。もれなくだぶりなく分類できるような条件を見出すことは、教科知識を使って、人間が考えなければならないことである。

3.3.2 順次

命令を順番に並べて手順を考えることである。1年算数の色板で形づくりをする授業で、「ずらす」「まわす」「うらがえす」という3つの命令を組み合わせる色板の位置の特徴を把握した。このことは、5年算数の合同な図形に通じる思考パターンである。

がわかれば、命令をより単純化できるという点で、同じ思考パターンになる。

3.3.4 計算モデル

コンピュータが動く仕組みを考える思考パターンである。二進数、演算の手順、四捨五入の仕組みなどが該当する。



図7 四捨五入

図7は、Scratchにある四捨五入のブロックを使って、数値シミュレーションするテンプレートである。小数点第一位で四捨五入されている。一の位や十の位で四捨五入するには、どのようなプログラムにすればいいかを考えさせる。



図8 たし算

たし算のシミュレーションをするテンプレートもこの思考パターンに該当する。たし算のブロックをうまく使えば、たし算電卓を作ることでもできるのである。かけ算やわり算の筆算の途中式を確認するプログラムも、同様に考えることができるだろう。

「1000より小さい自然数では、3の倍数または5の倍数の和は何か」という問題に対して、算数では、3の倍数の和と5の倍数の和を求めて、15の倍数の和を差し引くという考えで解答することが多い。しかし、コンピュータに命令するなら、1から1000の値を一つずつ判定して、3の倍数または5の倍数の条件にあてはまる数をどんどん足し上げるアルゴリズムの方が命令がわかりやすく、一般化しやすい。

また、社会で都市の特徴を示したいとき、例えば、その都市が海に面しているかどうかという特徴を、

どのようにコンピュータに理解させればいいのかを考える必要がある。人間なら言葉でわかることも、コンピュータは数値化しなければ通じない。そこで、海に面しているなら1、面していないなら0というデータを与えれば、コンピュータへの命令になる。

これらが、コンピュータが動く仕組みを考えた思考パターンの例である。

3.3.5 評価（テスト）

考え得る全ての可能性を確認する思考パターンである。全体を場合分けして、それぞれの場合について、もれなく条件を定めていく必要がある。「もしAならばB、そうでなければC」という条件制御を綿密に考える。条件が抜けてしまうと、コンピュータが行き場を失って止まってしまう。



図9 明るさセンサーによる省エネ

6年理科では、明るさの程度によって、必要なときだけ電気をつける条件を考えた。まず、「暗くなったら電気をつける、そうでなければ電気を消す」という条件を言葉で考える。次に、「暗くなったら」というあいまいな言葉を、数値で定義する必要がある。図9の例では、「明るさが100より小さくなったら」という条件で数値化している。100の値が明るさの閾値である。値を大きくすれば閾値が甘くなる。



図10 栄養バランスのいい献立づくり

図10は、栄養バランスが正しい献立かどうかを見分けるクイズである。明るさと同様、「栄養バランスが正しいならば正解、そうでなければ不正解」という条件制御である。「栄養バランスが正しい」という条件をコンピュータがわかるように翻訳する必要がある。この授業では、「赤、ピンク、オレンジ、黄、緑の5色の栄養素が献立の中に全部あれば」、栄養バランスが正しいことにした。図10で、左上のメニューについての栄養素の色が表示されている。この単品メニューは5色そろっていない。他のメニューで不足分が補えるかどうかを考えさせるクイズづくりである。

3.3.6 構造

全体像を想定して設計する思考パターンである。例えば、作文で、はじめ、なか、おわりという三部構成がある。言いたい意見はたくさんあるのに、この構成にあてはめるのが苦手な児童がいる。そこで、図11のように、この三部構成に意見がはまるようにナビゲーションするテンプレートを開発した。作文の型に慣れることで、自力でも書けるようになる。

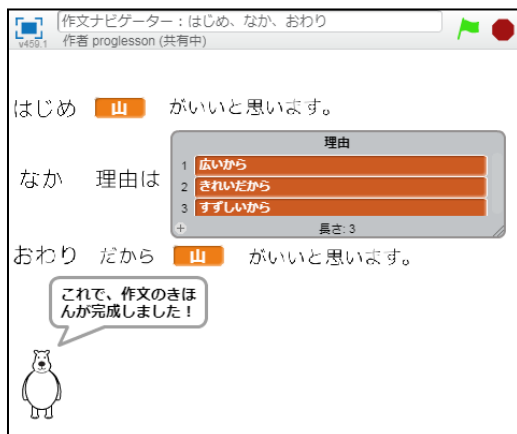


図11 作文ナビゲーター

同様の観点で、図12はUKの教材事例である。クリスマスカードや年賀状などの定型文で、個人による部分を変数にして、構造を示したものである。この教材を説明するのに、アルゴリズムという言葉が使われていた。定式化された構造だからだろうか。

```

"I" + verb + "your" + adjective + " " + noun + "."
"You are my" + noun + "."
"I want" + verb + "you forever."
"I" + verb + "your" + adjective + " " + noun + "."
"My" + noun1 + " " + adverb + " " + verb + "your" +
adjective + " " + noun2 + "."
"Yours," + adverb + " Q"

```

図12 定型あいさつ文のアルゴリズム

3.3.7 データ

条件を満たすデータを抽出する思考パターンである。とり出したいデータは、どういう条件を満たすものなのかを正確に設定する必要がある。3.3.1.の分類と似ているかもしれない。

データは、オーストラリアと米国のカリキュラムでは中核である。小学校段階では、身近な対象の個数を数えて統計資料にし、問題解決に生かす程度の活動から始まる。上学年になっても、プログラミング的思考がなければ、ビッグデータの扱いができないことになってしまうので、小学校段階からデータの収集・抽出には慣れておく方がいい。

4. 考察

これまで見てきたとおり、プログラミング的思考のパターンは教科学習に潜んでいるのだが、それらは、単一教科内容の習得に付随する思考とは別のもので、教科内容によらない思考であり、コンピュータの特徴をふまえた思考も含まれる。このことを模式図で示したのが、図13である。

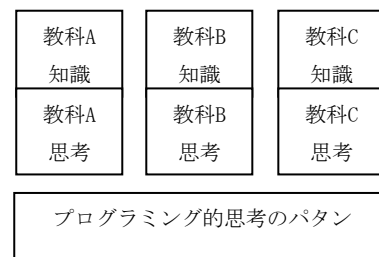


図13 プログラミング的思考のパターンと教科学習

2.3で述べたように、プログラミング的思考のパターンは教科によらないものであること、また、その思考の枠組みの中で設計部分と手続き部分とで関連のある教科が異なること、3.3の教材事例で示したとおり、複数教科で同じ思考パターンがあることから、教科横断で育成しやすいのではないかと示唆される。

育成には、プログラミング的思考の枠組みを配慮したテンプレートを使うことで、学習者の記憶に定着しやすいものになるだろう。

5. 今後にむけて

表7にまとめられたプログラミング的思考の7つのボタンが複数教科で意図的に指導されたとき、本当に知識理解に役立つような深い教科の学びになるのかどうか、確かめる評価手法の開発が必要である。

評価手法としては、児童の態度変容を見るための教員による行動観察のほか、より客観的に評価できるように、例えば、目的に合わせて与えられたプログラムをデバッグしたり、改良させたりするなど、指導と評価が相互的に作用するようなアセスメントを開発し、教材を改良していく必要がある。

また、3.3.4に述べたとおり、コンピュータに適したアルゴリズムが必ずしも人間が解く場合と同じとは限らない場合、教科学習の中でそのようなアルゴリズムも指導した方がいいのかどうか、議論し考察していきたい。

謝辞

本研究を行うにあたり、授業づくりや授業実践に際し、大阪市立茨田東小学校、柏メディア教育研究会、戸田市立喜沢小学校、戸田市立戸田第二小学校、戸田市立美女木小学校、戸田市立戸田南小学校の先生方、鈴木久氏、谷内正裕氏、嶋田宗孝氏に多大なご協力をいただきました。深く感謝いたします。

参考文献

- 赤堀侃司 (2018) プログラミング教育における論理的な思考とは何か、学習情報研究論文誌 第261巻第4号
- ベネッセコーポレーション (2017) プログラミングで育成する 資質・能力の評価規準<試行版> Computing At School (CAS) (2015a) CAS Computing Progression Pathways
- Computer Science Teachers Association (CSTA) (2017a), CSTA K-12 Computer Science Standards, Revised 2017
- Computing At School (CAS) (2018a) INTERDISCIPLINARY COMPUTATIONAL THINKING, TEACHING LONDON COMPUTING: A RESOURCE HUB FROM CAS LONDON & CS4FN
- 樋口直宏 (1998) 思考教授プログラムにおける思考技能の構造と教材内容—アメリカ教材の分析を中心に—, カリキュラム研究第7号79-91
- 茨城大学教育学部附属小学校 (2018) 小学校プログ

- ラミング教育授業化プロジェクト 第10回公開授業研究会 実践報告書
- 川崎弘作・寺本貴啓・松浦拓也・角屋重樹 (2009) 科学的思考力の評価問題の開発に関する研究—PISA2006科学的リテラシーの「科学的能力」に着目して—, 日本教科教育学会誌32巻4号
- 久田隆基・萱野貴広他 (2004) 科学的思考力と言語能力を育むための初等中等理科教材の開発研究—中学校理科授業での一実践事例—, 日本科学教育学会年会論文集28
- 黒田昌克・森山潤 (2018) 小学校段階におけるプログラミング教育に対する教員の意識と意義形成要因の検討, 教育メディア研究 Vol.24, No2, 43-54
- 教育テスト研究センターCRET (2010) 21世紀スキルとしての問題解決力と国際的な評価の枠組み
- Levy, F. (2006) How Computerized Work and Globalization Shape Human Skill Demand Department of Urban Studies and Planning, MIT, R. J. Murnane, Graduate School of Education
- 本松裕二・木村捨雄(1998) 非定型的求積問題の問題解決における高次思考と問題解決プロセスに関する研究(1), 日本科学教育学会研究会研究報告13巻1号
- 文部科学省 (2016a) 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)
- 文部科学省 (2018a) 高等学校学習指導要領
- 文部科学省 (2018b) 小学校プログラミング教育の手引 (第一版)
- 文部科学省 (2018c) 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), 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
- 長崎栄三 (2003) 算数・数学の学力と数学的リテラシー, 教育学研究70巻3号302-313
- 中山広文・塩飽修身他 (2018) 科学的思考力を育むために開発したカリキュラムの評価, 日本科学教育学会研究会研究報告25巻4号
- 太田剛・森本容介・加藤浩 (2016) 諸外国のプログラミング教育を含む情報教育カリキュラムに関する調査—英国, オーストラリア, 米国を中心として—, 日本教育工学会論文誌 40(3), 197-208
- 斉藤成海 (2011) 数学的な見方や考え方を基盤とし

た論理的な思考，群馬大学教育実践研究別刷第28号

相模原市立総合学習センター（2017）相模原市のプログラミング教育の実施状況

滋賀県総合教育センター（2016）平成28年度(2016年度) 情報教育に関する研究，小学校におけるプログラミング的思考を育む授業づくりー図画工作科におけるICTを活用した授業実践を通してー

清水欽也(2004)一般成人の教科有用観からみた教科構造ー一般成人調査からの教科再編への提言(1)一，日本教科教育学会誌第27巻第1号

清水美憲(2008) OECD／PISA が提起する数学的リテラシー論の意義，日本科学教育学会第32回年会論文集

田中一磨（2008）科学的思考力を高める単元開発および学習支援に関する実践研究ー第3学年「くらしに役立つ磁石の力を感じよう」からの一考察一，日本科学教育学会第32回年会論文集

<http://benes.se/keyc>

The Australian Curriculum, Assessment and Reporting Authority (ACARA) (2015a) Technologies.

