

授業実践に基づく小学校プログラミング教育「評価規準」の提案

- 授業における評価規準の必要性を踏まえて -

小林未歩*1・宇都宮 晃*2・宮澤豪臣*3・福島健介*4

Email: miiisuke626@gmail.com, hukusima@ehusi.org

*1: 町田市立小山中央小学校

*2: 八王子市立宮上小学校

*3: 株式会社ワイズインテグレーション

*4: 帝京大学教育学部

◎Key Words プログラミング教育, 授業実践, 評価規準

1. はじめに

2018年3月、文部科学省は「小学校プログラミング教育の手引（第一版）」（以下、手引き）を発表した。手引きは「参照していただくことによって、プログラミング教育のねらいやどのような授業が期待されているのかをイメージしていただける」（p.2）と述べているように、2020年度から実施される小学校プログラミング教育の「指針」となることが予想される。

したがって、その内容を精査検討し、必要な提言をしていくことは関係学会、研究団体にとって喫緊の責務となるであろう。

本稿では、特に **Progression Pathways** としての評価規準を取り上げ、その必要性について論じ、併せて試案としての評価規準を提案したい。

2. 小学校プログラミング教育のねらいと評価

2.1 小学校プログラミング教育のねらい

「手引き」はプログラミング教育のねらいについて新指導要領3観点を基に以下のように論じている。

表1 プログラミング教育のねらい（手引きより）

【知識及び技能】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。
【思考力、判断力、表現力等】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。
【学びに向かう力、人間性等】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。

さらに「手引き」では「「プログラミング的思考」を育成することは、小学校におけるプログラミング教育の中核（p.11）と述べ、そのねらいを定めている。

2.2 プログラミング的思考と育成すべき「能力」

ここで議論の中心となる内容は「プログラミング的思考」とは何か、という点であろう。

「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）では、プログラミング的思考とは「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と記載されている。

Computational Thinking を教育の分野で取り入れて

いるのがUK（England）であることから、プログラミング的思考についてもUKの概念を参照した内容であると考えられる。「The national curriculum in England Key stages 1 and 2 framework document」では、「A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world.」とその目的を述べている。さらに学ぶべき学習内容を小学校教員向けに解説した「Computing in the national curriculum A guide for primary teachers」の中では具体的な目標が記されており、

Abstraction[AB], Decomposition[DE], Algorithmic Thinking[AL], Evaluation[EV], Generalization[GE] という5つの能力を示している。

これらはそれぞれ「抽象化」「分解」「アルゴリズム的思考」「評価」「一般化」と日本語化されており、こうした定義をもとに、安藤（2017）は、プログラミング的思考を表2のように示している。

表2 プログラミング的思考の内容（安藤2017より）

自分が意図する一連の活動を実現	抽象化
どのような動きの組み合わせが必要	分解
どのように組み合わせたらいいのか	アルゴリズム的思考・一般化
記号の組み合わせをどのように改善	修正
より意図した活動に近づくのか	評価

以上からも明らかなように、Computational Thinking とプログラミング的思考は異なる概念⁴⁾である。

ともあれ「議論の取りまとめ」で示されたプログラミング的思考を分節化すると、表2のような能力を育成することが目的であることが理解できる。

3. 問題の所在

どのような教科でも、授業を実施する上で必須となるのが、「学習目標」である。実際の授業においては「授業のねらい」「本時の目標」等、表記は様々であるが、その授業において児童に身に付けさせたい内容、換言すれば「その授業終了後に児童が到達している能力」を授業者は意識している。

一時間の（教科の）授業の目標は、学年の目標→単元の目標→本時の目標という関係性を持つ。すなわち、その学年において、「どのような能力」を「どの程度」

育成するのか、が明示されていないければ現実には授業は行えない構造となっているのである。

前者を通常は「観点」、後者を「到達目標」という。

そこで課題となるのが、プログラミング的思考の育成において、「どの程度」が明示されていない点であろう。「アルゴリズム的思考」と言っても、1年生と6年生では当然到達目標は異なる。さらに、学校教育である以上、順序性や系統性の確保も必要である。

「手引き」では「プログラミングを実施した際の評価については、あくまでも、プログラミングを学習活動として実施した教科等において、それぞれの教科等の評価規準により評価するのが基本」(p.18)と述べている。

先に挙げた England では、Computing は教科であるため、評価/評定のための Progression Pathways が明示されている(表3)。

表3 Progression Pathways(sandymoorschool の事例)

K1	I know what an algorithm is and I can express simple algorithms using symbols.
K2	I know that algorithms are implemented on digital devices as programs.
K5	I know that iteration is the repetition of a process such as a loop.

我が国の場合、プログラミング教育は教科として独立しているわけでは無いから、評価/評定を目的として評価規準を用いる必要は無い。

しかし、本来、評価規準とは「学習到達すべき目標水準」(国立教育政策研究所,2002)であり、観点によって示された子どもにつけたい力を、より具体的な子どもの成長の姿として文章表記したものである。表3の Progression Pathways はその通りに記されている。

これについて「手引き」は「各学校がプログラミング教育で育みたい力を明らかにし」と、各学校での「学習到達すべき目標水準」開発を要請している。

この点について、問題を2点指摘するならば

1. ICT環境も人的・物的リソースも多様な全国2万の公立小学校または市町村教委がそれぞれ開発した場合、到達目標水準が著しく異なる可能性が高い
2. 現状のICT環境を考えるならば、そもそも開発をし得ない市町村教育委員会、学校の方がむしろ多数派である

つまり、到達すべき目標水準としての「評価規準」を各学校が「明らかに」するためには、何らかのモデル・試案が必要なのである。筆者らはこうした問題意識に基づき、授業実践から帰納的に「評価規準」を作成した。

4. 評価規準の作成過程

4.1 プログラミング的思考の評価の観点

筆者らは、プログラミング教育の評価規準を作成する前に、まずプログラミング的思考を評価する観点の分類を提案し、それを現行の国語と算数の評価規準に当てはめるという作業を行った。評価規準は5項目に

分類し、その内容は以下の通りである。

1. 順序や手順を考えて取り組む力
2. より良い方法を考え、試行錯誤できる力
3. ものごとの組み立てを分解して理解する力
4. 方法を抽象化、一般化、最適化し、他の場合に適用できる力
5. 自分の考えや感性を適切に表現する手段としてプログラミングを用いる力

そして、上記の分類に指導要領解説で示された国語と算数の評価規準を当てはめたものが、表4であり、PCC2017にて提案をしている。

表4 教科の到達目標とプログラミング的思考の関係性(算数)

教科	算数	算数	算数	算数	算数	算数
算数	算数(1) 算数(2) 算数(3) 算数(4) 算数(5)	算数(1) 算数(2) 算数(3) 算数(4) 算数(5)	算数(1) 算数(2) 算数(3) 算数(4) 算数(5)	算数(1) 算数(2) 算数(3) 算数(4) 算数(5)	算数(1) 算数(2) 算数(3) 算数(4) 算数(5)	算数(1) 算数(2) 算数(3) 算数(4) 算数(5)
国語	国語(1) 国語(2) 国語(3) 国語(4) 国語(5)	国語(1) 国語(2) 国語(3) 国語(4) 国語(5)	国語(1) 国語(2) 国語(3) 国語(4) 国語(5)	国語(1) 国語(2) 国語(3) 国語(4) 国語(5)	国語(1) 国語(2) 国語(3) 国語(4) 国語(5)	国語(1) 国語(2) 国語(3) 国語(4) 国語(5)

4.2 観点の改善

先述した5つの観点を、より現場で取り入れやすい評価や分かりやすい表現へと改善する為に検討した結果、以下のように変更した。

1. 順序や手順を考えて取り組む力
2. ものごとの組み立てを分解して理解する力
3. より良い方法を考え、試行錯誤できる力
4. 自分の考えや感性を表現する手段としてプログラミングを用いる力
5. 分かった考え方や方法を生活の他の場面に活かす力

主に改善された点は、4番の表現の修正と、順序の修正の2点である。

1点目は、「4. 方法を抽象化、一般化、最適化し、他の場合に適用できる力」を「4. 自分の考えや感性を表現する手段としてプログラミングを用いる力」という表現へ修正した点である。表現を修正した理由は、抽象化、一般化、最適化の定義が難しく、現場で用いられることを想定するならば、より分かりやすい表現にすべきであると考えたからである。この修正により、現場で取り入れやすい評価規準に近付けたのではないかと考えられる。

2点目は、分類の順序を「分解して理解する→試行錯誤する→表現する」に修正した点である。順序を修正した理由は、プログラミングの考え方にに基づき、流れを意識すべきであると考えたからである。この修正により、プログラミングについてあまり詳しくない教員であっても、この評価規準を見ながらプログラミングの流れを意識して授業を計画することができると考えられる。

4.3 新指導要領の3観点へ分類

次に、先述した5つの観点の分類から、評価規準をより現場で使いやすく、誰でも理解しやすいものにするため、新学習指導要領の三観点の枠組みに仕分けた。

結果は、次の通りである。

表5 新指導要領観点とプログラミング教育の観点の関係

評価規準 (文部科学省)	資質・能力 (文部科学省)	目標 (文部科学省)	分類 (プログラミング教育開発研究会試案)
知識・技能	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	各教科等で育まれる思考力を基盤としながら基礎的な「プログラミング的思考」を身に付けること	順序や手順を考えて取り組む力
思考力・判断力・表現力等	発達の段階に即して、「プログラミング的思考」が意図する一連の活動を表現するために、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いのか、記号の組合せなどどのように改善していけば、より意図した活動に近づけるのか、といったことを論理的に考えてい力を育成すること	コンピュータに意図した処理を行うよう指示することができるということを体験しながら、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	ものごとの組み立てを分解して理解する力 より良い方法を考え、試行錯誤できる力
学びに向かう力・人間性等	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	コンピュータの働きを自分の生活に生かそうとする態度を身に付けること	自分の考えや感性を表現する手段としてプログラミングを用いる力 分かった考え方や方法を生活の他の場面に活かす力

4.4 評価規準の完成

最後に、3観点に当てはめた表を基に、学年を低学年(1, 2年)、中学年(3, 4年)、高学年(5, 6年)の3つに分け、それぞれの成長に合わせた評価規準を作成した。

また、先述した5つの観点の分類に、新たに「課題を明確に設定する力」を【思考力・判断力・表現力等】の項目に加え、分類の前に(ア)～(カ)のルビを振った(表7)。

追加した理由は、分解や試行錯誤をする段階の前に、課題を設定する能力が必要なのではないかと判断したからである。この追加により、分類の順序が「課題を設定する→分解して理解する→試行錯誤する→表現する」という流れになり、より児童のプログラミング教育における学習のビジョンが明確になると考えられる。さらに、ルビを振ったことにより、指導案に取り入れる際に、明記しやすくなる。

本評価規準では、低学年のプログラミング教育の学びの軸を【知る・気付く】、中学年を【学ぶ】、高学年を【活かす】と設定し、規準を作成している。

本評価基準を、安藤(2017)と対応させたものが以下の表6である。

表6 提案する評価基準とプログラミング的思考の関係性

筆者らの提案	安藤による分類
(ア) 順序や手順を考えて取り組む力	アルゴリズム的思考
(イ) 課題を明確にする力	抽象化
(ウ) ものごとの組み立てを分解して理解する力	分解
(エ) より良い方法を考え、試行錯誤できる力	評価
(オ) 自分の考えや感性を表現する手段としてプログラミングを用いる力(一般化:A)	
(カ) 分かった考え方や方法を生活の他の場面に活かす力(一般化:B)	一般化

ここで、「一般化」を2つに分けたのは、Aが授業で用いたプログラミング技能自体の一般化、Bがプログラミング的思考の一般化としたからである。

コーディングそのものを教えることが小学校での目的ではないが、授業の様子を見てみると、子ども達はプログラミングを様々な表現活動(おはなし作り、作品作り)に応用させようとする。ならば、その態度や志向性も評価の対象とした方が素直ではないか、というのが私達の考え方である。

5. 先行事例との比較

5.1 文部科学省との比較

文部科学省は、プログラミング教育を通じて目指す育成すべき資質・能力を、提案している。比較したとき、本提案にはいくつかの長所があると考ええる。

1点目は、学年ごとの具体的な目標が定まっている点である。文部科学省の評価規準は、3観点の目標は提示されているものの、学年の成長過程に合わせた具体的な目標は明記されていない。対して本評価規準は児童

表7 筆者らが提案するプログラミング教育の評価規準(試行版)

評価規準 (文部科学省)	資質・能力 (文部科学省)	目標 (文部科学省)	分類 (プログラミング教育開発研究会試案)	低学年(1, 2年)【知る・気付く】	中学年(3, 4年)【学ぶ】	高学年(5, 6年)【活かす】
知識・技能	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	各教科等で育まれる思考力を基盤としながら基礎的な「プログラミング的思考」を身に付けること	(ア)順序や手順を考えて取り組む力	自分の生活の中にコンピュータが活用され、身近なもの(自動販売機など)にプログラミングが活用されていることに気付くこと。	1.プログラミングには、明確な指示が必要であることを知る。 2.プログラミングの思考を用いて、順序良く明確な表現をすること。	1.プログラムには、プログラミング言語が用いられ、プログラミング特有の考え方があることを知る。 2.プログラミングの思考を用いて、最適な表現をすること。
思考力・判断力・表現力等	発達の段階に即して、「プログラミング的思考」が意図する一連の活動を表現するために、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いのか、記号の組合せなどどのように改善していけば、より意図した活動に近づけるのか、といったことを論理的に考えてい力を育成すること	コンピュータに意図した処理を行うよう指示することができるということを体験しながら、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	(イ)課題を明確に設定する力	自分が意図する一連の活動を表現すること。	自分が意図する一連の活動を、わかりやすく、具体的に表現すること。	自分が意図する一連の活動を、最速に実現するためにどうすればよいかを協同的に考えること。
			(ウ)ものごとの組み立てを分解して理解する力	教科で学習したことや日常生活の中には、いくつかのまとまりの組み合わせでできているものがあることに気付くこと。	教科で学習したことや日常生活の中には、いくつかのまとまりの組み合わせでできているものがあることに気付く、分解することができること。	ものごとの複数の要素で構成されていることに気付く、目的に応じてその構造を分解したり、相手に伝えたりすること。
			(エ)より良い方法を考え、試行錯誤できる力	1. 目的に応じた方法を考えることができる。 2. 結果がうまくいかなかった場合、その原因を考えようとする。	1. 目的に応じて、自分なりに最適なものを考えること。 2. 結果が目的と合っているかを確認し、より良いものにするための方法を考えること。	1. 目的に応じて複数の情報を比べ、最適なものを選択すること。 2. 他の意見や作品から改善が必要かを判断し、原因や改善策を考えようとする。
			(オ)自分の考えや感性を表現する手段としてプログラミングを用いる力	情報機器やプログラミングに興味や関心をもち、その良さに気付くこと。	自分の感じたことや考えたことを伝えるために、情報機器やプログラミング的思考を使って伝えることができること。	自分の感じたことや考えたことを伝えるために、進んで情報機器やプログラミング的思考を用いて相手に分かりやすく伝えること。
学びに向かう力・人間性等	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	コンピュータの働きを自分の生活に生かそうとする態度を身に付けること	(カ)分かった考え方や方法を生活の他の場面に活かす力	各教科で新しく学んだ事柄を、生活体験と結びつけて考えること。	各教科で新しく学んだ事柄や日常生活の中に、既習事項やこれまでの経験との関係性や類似性があるかを考えること。	既習事項を他の問題や日常生活に当てはめ、より良く問題解決するために活かすことや類似性があるかを考えること。

の成長過程を低・中・高学年の三学年に分け、具体的な目標を明記している。この点において、本評価規準を用いれば、児童の成長過程に合わせた授業を組み立てることができる。と考える。

2点目は、観点が分類されている点である。文部科学省の評価規準は、3観点到合わせて大まかな資質・能力、目標を設定している。対して本評価規準では、3観点到合わせて目標を6つに分類している。この点において、本評価規準は、1時間の授業でねらう目標を絞りやすく、また、プログラミング教育にあまり詳しくなくても、どのような能力を身に付けさせればいいのかのわかりやすいのではないかと筆者は考える。

5.2 ベネッセと CANVAS の共同研究チームとの比較

ベネッセと CANVAS の共同研究チームは、プログラミングで育成する資質・能力の目標を提案している。

上記の評価規準と比較し、本評価規準の特長を2点挙げる。

1点目は、適度に分類されている点である。ベネッセと CANVAS の共同研究チームによる評価規準は、3観点から目標を3~5つに細分化し、児童の成長過程を低・中・高学年の三学年に分け、具体的な目標を明記している。

対して本評価規準は、児童の成長過程の分け方は同様であるが、目標は全部で6つに分類されている。この点において、本評価規準は細分化されないことで、一時間の授業でねらえる能力を絞られ過ぎず、その自由度が高いといえる。

2点目に、言葉が分かりやすい点である。ベネッセと CANVAS の共同研究チームの評価規準は、「事象の階層構造」「繰り返し処理」「条件分岐処理」というような、プログラミングに特有な知識がないとわかりにくく、一読しただけでは理解することが難しい表現が用いられている。

対して本評価規準は、誰にとってもわかりやすい評価規準を目標としているため、なるべく難しい単語や表現、またプログラミング用語も使わずに作成した。

6. 本評価規準の位置付けと今後の方向性

提案した到達目標としての評価規準は、あくまでも「試案」であり、今後筆者らのチームでもさらに検討を加え、より実践的な内容としていきたい。

すでに、プログラミング教育に先進的に取り組んでいる全国の研究指定校等では独自の「プログラミング教育で育みたい力」を設定する取り組みが始まっている^{*2}。問題は、こうした独自の力ある取り組みが、公開・交流されず、議論されない現状であろう。そのために、いくつかの問題点も生まれている。

第一は、主に民間（企業）主導の、プログラミング習得に特化した学習カリキュラムである。プログラミングさえ学べば、「プログラミング的思考」は自然に身に付くという「自然成長論」は、やはり誤りだろう。

第二は、「思考力全般」の育成に課題を解消しがちな学校・教員の傾向である。特に、アンプラグドの実践にはこうした授業例が散見される。

「プログラミング的思考」は computer science（文科省的には情報活用能力）の中に位置付けるべきテーマである。文科省はこの点について「情報を収集・整理・比較・発信・伝達する等の力をはじめ、情報モラルや情報手段の基本的な操作技能なども含めたトータルな情報活用能力を育成する中に、「プログラミング的思考」の育成を適切に組み入れていく」（手引き p.14）と述べている。この指摘は正しいと考える。

筆者らはこうした「ぶれ」を防ぐためにも、早期に評価規準を設定し、それに基づく実践を行うことが必要である。と考えるものである。

2020年に向けて、全国の多くの学校でプログラミング教育の試行が開始されている。そうした実践に向けて、本規準が一つの参考となることを筆者らは願っている。

参考文献

- (1) 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）”，（2016）
- (2) Department for Education (Government of UK): “The national curriculum in England Key stages 1 and 2 framework document”, pp.178, (May 2015).
- (3) 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（文部科学省平成26年度・情報教育指導力向上支援事業）”，（2014）
- (4) David Morris: “Teaching Computational Thinking and Coding in Primary Schools”, LEARNING MATTERS 社, (2017)
- (5) 安藤明伸: 情報教育対応教員研修全国セミナー資料『どうする？授業で取り組む「プログラミング」』, (2017)
- (6) 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 - 』, (2016)
- (7) SANDYMOORSCHOOL, Curriculum Map - Computing, http://www.sandymoorschool.org.uk/?page_id=3117 2018年6月4日閲覧
- (8) プログラミングで育成する資質・能力の評価規準, (株) ベネッセ コーポレーション <https://beneprog.com/terms/> 2018年6月4日閲覧
- (9) 文部科学省: “小学校プログラミング教育の手引（第一版）”, (2018)
- (10) Simon Humphreys: “Computing in the national curriculum A guide for primary teachers”, STEM Learning Ltd., (2017)
- (11) 及川良紀: “論理的思考力を育むプログラミングの体験の在り方に関する研究 - 小学校算数科・理科の指導を通して - ”, 岩手県総合教育センター, (2018)

*1 及川(2018)は、「コンピューショナル・シンキングは問題解決能力やコミュニケーション能力なども含んだ範囲の広い考え方であるのに対し、プログラミング的思考はその中から限定的な部分を抽出した考え方」と述べている。

*2 例えば立川市立上砂川小学校は、2018年度にむけて「プログラミング教育で育む資質・能力（案）」を2018年2月に発表している。