Vol.2016-CE-135 No.8
2016/7/2

初等・中等教育向けプログラミング教育カリキュラムに対応した指導者養成プログラムの提案

大森康正[†1]　伊藤寿晃[†2]　吉田研一[†3]　長瀬大[†4]　山脇智志[†2]　栗林聖樹[†5]

概要：本報告は，我々が開発を行っている初等・中等教育における体系的なプログラミング教育のカリキュラムに対応した指導者養成プログラムに関するカリキュラムと学習環境の提案である．初等・中等教育においてプログラミング教育を実施するには，体系的なカリキュラムと指導者養成が重要であると考えられる．そこで，我々は，小学生以上を対象とした各種プログラミング講座を実施し，その成果を基に公教育および私教育が連携可能な初等・中等教育における体系的なプログラミング教育のカリキュラムの開発を行ってきた．本報告では，プログラミング教育のカリキュラムの体系と，それを実施する指導者に求められる知識，技能を明らかにして，それを習得する養成プログラムと学習環境について述べる．

キーワード：プログラミング教育，指導者教育，カリキュラム，実施環境

PROPOSAL OF AN INSTRUCTOR TRAINING PROGRAM CORRESPONDING TO THE PROGRAMMING EDUCATION CURRICULUM TARGETING ELEMENTARY AND SECONDARY EDUCATION

YASUMASA OOMORI[†1]　TOSHIAKI ITO[†2]　KENICHI YOSHIDA[†3]
DAI NAGASE[†4]　SATOSHI YAMAWAKI[†2]　MASAKI KURIBAYASHI[†5]

Abstract: This report is the proposal of a curriculum and an educational environment with regards to instructor training program that corresponds to the systematic programming education curriculum we develop for elementary and secondary education. In order to implement programming education in elementary and secondary education, a systematic curriculum and the fostering of instructors are important. Thus we have conducted a wide variety of programming workshops targeting elementary school students and up, and have utilized the results as the basis for the development of a systematic programming education curriculum in elementary and secondary education that public and private education can work in tandem with each other. In this report, we identify the system of the programming education curriculum as well as the knowledge and skills required to the instructors who conduct it, and describe the training program and the educational environment to acquire them.

Keywords: Programming Education Curriculum，Instructor Training Program，Educational Environment

1. はじめに

世界最先端 IT 国家創造宣言（平成 25 年 6 月閣議決定）以降，日本においても初等・中等教育および私教育においてプログラミング教育の必要性が浸透し，各種の試みが行われてきた[1]．我々が知る限り，小学生を対象として実践されているプログラミング教育については，私教育における体験的な取り組みが大半であり，それらの実施時間も長くはない．これらの取り組みの意義は十分にあると言えるが，21 世紀型能力の修得から ICT 人材の育成までを連続的にかつ体系的にカバーする取り組みとしての実施は不十分であると考えられる．

また，初等・中等教育におけるプログラミング教育については，一部の意欲的な学校において全学的な実践が行われていたが（例えば参考文献[2]など），2016 年 6 月に文部科学省の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において，“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）（案）”[3]が提示されるなど，2020 年度以降の小学校での必修化に向けて，大きく動き出したといえる．しかしながら，小学校から高等学校におけるプログラミング教育全般において ICT 人材育成へとつながるような体系的なプログラミング教育の実施，教員研修の在り方，公教育と私教育の連携や地域や民間企業との連携などの官民連携については，これからの課題であると考えられる．

このような背景の中，我々は，公教育および私教育の両方をカバーする体系的なプログラミング教育の概略に関す

†1 上越教育大学
Joetsu University of Education
†2 (株)キャスタリア
Castalia Co., Ltd.
†3 ECC コンピュータ専門学校
ECC Computer College
†4 四国大学
Shikoku University
†5 学校法人信学会
Educational Foundation Shingakukai

る試案[4]に基づいた小中学生向けのプログラミング講座を子供プラザ等の児童生徒を対象として実践を行い試案の評価を行ってきた．さらに，それらの結果から得られた知見を基に，初等・中等教育向けの体系的なプログラミング教育カリキュラム[5]について開発を行っている．

さらに我々は，初等・中等教育におけるプログラミング教育の指導者不足への対応として，初等・中等教育における体系的なプログラミング教育カリキュラムに対応した指導者養成プログラムの指導者養成カリキュラムと学習環境の開発を行っている．本報告では，まず，初等・中等教育において児童生徒に修得させたい知識や技能について検討を行い，それに基づくプログラミング教育の体系的なカリキュラムについて述べる．さらに，それを実施する指導者に求められる知識，技能を修得するための指導者養成プログラムと学習環境について提案する．

2. プログラミング教育で修得させたい知識と技能

はじめに，生徒児童に対してプログラミング教育を通して修得させたい知識や技能について検討する．諸外国においては，コンピュータサイエンス（以下，CS と呼ぶ）や Computational Thinking（以下，CT と呼ぶ）などに基づいた情報教育が先行的に実施あるいは今後予定されている[6][7]．我が国においても，小学校段階においてはプログラミング的思考を基礎においた取組が検討されている[3]．なお，参考文献[3]によると，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*』とされている．これは CT とほぼ同等のものと考えられる．

我々も，これまでの実践経験，文部科学省での議論や諸外国の動向を踏まえて，21 世紀の知識基盤社会において必修の考え方でもある CT および CS を基礎とした初等･中等教育向けプログラミング教育カリキュラムの開発を行っている．ただし，イングランドの「Computing」のように各キーステージにおいて体系的に CS や CT を積み上げていくカリキュラムではない．我々は，図 1 に示すように，小学校では主に CT に関する基礎の修得を行い，中学校では CT のための技術として情報技術（以下, IT と呼ぶ）を修得し，高等学校においては，それまでの CT と IT を修得した体験を基にして，それらを深く理解するための CS を修得する．

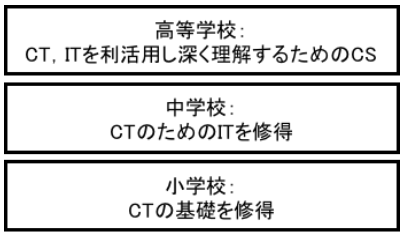

図 1　カリキュラムにおける CT，IT，CS の関係

また，私教育においては，参考文献[1]によると，現在，我が国で行われているプログラミング教育の方向性としては，①問題解決能力，コミュニケーション力など 21 世紀の知識基盤社会で求められる能力である 21 世紀型スキルの習得，②ICT インフラの構築，ソフトウェア開発などによって ICT を支え，新しい ICT を生み出す人材の育成，および③ICT を使いイノベーションを起こす人材の育成に整理される．なお，これら 3 つの方向性は，それぞれ独立してあるのではなく，①の 21 世紀型スキルの習得は，すべての国民が習得すべきものであり，②と③の基盤ともなっている．また，②と③もお互いを補完する関係にあると考えられる．これらは，図 1 で示した関係との対応を考えると，①は小学校段階，②と③は中学校および高等学校段階の方向性に近いといえる．

したがって，小学校段階におけるプログラミング教育で修得させたい知識と技能等は表 1 の様に設定した．特徴的な事は，小学校段階では先の①問題解決能力に最も関係が深いと考えられる CT の基礎概念である，『抽象化（Abstraction）』，『分解（Decomposition）』，『パターン認識（Pattern Recognition）』などを修得させる事，および各学校種に共通した能力育成として問題解決行う際の手法などとしての試行錯誤（トライ＆エラー）を用いる点である．また，問題解決の PDCA サイクルを繰り返し実行するなかで，失敗をしてもそこから学ぶ姿勢を養う．さらに，アルゴリズムの解が複数あっても他者のアルゴリズムに間違いがなければ容認する姿勢を学ぶことが重要であると考えている．

表 1　小学校段階において修得すべき知識や技能など

問題解決を行う際の手法など	試行錯誤（トライ＆エラー）	
	複数解を容認する	
アルゴリズムを設計する際に必要な概念	抽象化	制御構造（逐次，繰り返し，判断）の抽象化 手続き化，モジュール化
		データ構造の抽象化
	パターン認識	
	分解	
	一般化	
	評価	

中学校段階においては，学校生活や家庭においてより良い生活を行うことを目的とした生徒の身近な課題を通して

小学校段階で修得した知識や技能の定着をはかる．そのために必要な IT に関する知識や技能の修得を行う．具体的には，表 2 のように Python や Swift などのプログラミング言語に関する知識および技能と，バブルソート，２分木探索などの基本アルゴリズムに関する知識や技能の修得を行う．プログラミング言語に関する知識や技能は，UX（ユーザーエクスペリエンス）デザインの基礎，Web アプリケーションに関しては情報通信ネットワークの基本原理および情報モラル（情報セキュリティを含む），さらに IoT 関連を題材とした場合は電子・電気回路の基礎知識なども関連して学ぶことが重要である．なお，中学校においては技術・家庭科技術分野にて学習することから，その他に，材料と加工，エネルギー変換，生物育成などの各領域との連携した複合題材も考慮する必要があると考えられる．

表 2　中学校段階で新たに修得すべき知識や技能など
（表 1 の内容の他に修得すべきものを上げている）

プログラミング言語に関する技術	Python や Swift などのテキスト型言語に関する知識と技能
	HTML5+CSS3 など Web アプリケーション開発に必要な言語等に関する知識と技能
アルゴリズムに関する技術	バブルソート，2 分木探索などの基本アルゴリズムに関する知識と技能

高等学校段階においては，実際の地域社会などで起きている課題等について，小学校および中学校で修得した知識および技能等を活用し問題解決を行う．また，その過程において CS に関する基礎的な知識についての理解を深めることで，それまでの学びで得た知識と技能の定着をはかり，その原理原則となる理論について理解を深めるようにする．具体的な CS に関する知識等は，高等学校の教科「情報」の次期学習指導要領の科目である「情報Ⅰ（仮称）」および「情報Ⅱ（仮称）」の内容[8]と同じである．

3. 初等中等教育における体系的プログラミング教育カリキュラム

3.1 プログラミング教育カリキュラムの概要

次に，2 で示した知識や技能を修得するための体系的なプログラミング教育カリキュラム（以下，教育カリキュラムと呼ぶ）についての概要を示す．その教育カリキュラムの概要は表 3 の通りである．また，教育カリキュラムの主な特徴は以下の通りである．

- 幼稚園年長組から高等学校までの K-12 を対象としている
- 日本の教育課程の区分およびイングランドのキーステージを参考に，K-12 を５分割し，５才から小学３年生くらいまでをステージ１，小学３年生から４年生をステージ２，４年生くらいから６年生をステージ３，中学生をステージ４，高校生をステージ５として，学ぶ内容を学年固定としないで柔軟に実施できるようにしている．
- 小学校では，特別に教科を定めずに，主に総合的な学習の時間や図画工作などで学ぶことを想定している．中学校では技術・家庭科技術分野，高等学校では教科「情報」で学ぶことを想定している．
- 公教育以外の塾，土曜授業および社会教育などの私教育においても活用できるように，各ステージ内およびステージ間の題材（課題）はスモールステップかつ反復して学べるように設定されている

表 3　各ステージで学ぶ内容と主な教材

Stage	主な内容	主な教材
1	擬似プログラミング言語（コード）を用いて，逐次処理などの基本処理，分解，評価(デバッグ)，トライ＆エラー，複数解の容認などを学ぶ	アンプラグド教材
2	簡易プログラミング言語（コード）を用いて，繰返し処理などの基本処理，分解，パターン認識，抽象化（制御構造の抽象化），評価(デバッグ)，トライ＆エラーなどを学ぶ	studio.code.org
3	簡単なプログラムの制作を通して，問題解決の手法およびアルゴリズムを設計するのに必要な知識や技能を学ぶ	Scratch LEGO EV3
4	学校生活や家庭などの身近な課題を通して問題解決，アルゴリズム設計，プログラミング言語および関連する IT 技術などに関する知識や技能を学ぶ	Python Swift JavaScript
5	実際の地域社会などで起きる課題の解決を通して問題解決，アルゴリズム設計，プログラミング言語および関連する IT 技術および CS などに関する知識や技能を学ぶ	Python Swift JavaScript

3.2 プログラミング教育カリキュラムの内容

小学校を対象としたステージ１から３を例として，具体的教育カリキュラムの内容と考え方について述べる．表 4 にステージ１で学ぶ，目標，指導内容，前提となる知識，到達度確認基準，教具，標準時数，育成すべき知識・技能について示す．ステージ１においては，主にアンプラグド

教材を用いることで遊びを通して学べる工夫をしている．特に表 4 中の 1－4 の題材として，コップタワーやロボットカーなどを取り入れ，図画工作，生活科および総合的な学習の時間などでの取り組みが可能となるように配慮している．

表 4　ステージ 1 の内容（略案）

Stage1（主な対象：年長組から小学校3年生くらい）の内容

Stage1	目的	指導内容	前提知識・技能	到達度確認基準	教具	標準時数	備考	育成すべき知識・技能
1-1	• プログラムとアルゴリズムの関係を理解する • 逐次処理について理解する	1. 逐次処理のアルゴリズムを考える 2. 逐次処理のアルゴリズムを擬似プログラムに変換 3. 逐次処理の擬似プログラムを実行する 4. 逐次処理の擬似プログラムの間違いを見つける	なし	I. 適切なアルゴリズムを考えることができる II. アルゴリズムから適切な擬似プログラムを作成できる III. 擬似プログラムと実行結果の対応が理解できる	アンプラグド：4 ✖ 4のマス目を塗りつぶすプログラムを考えよう（Code.org）	2時間	簡単なアルゴリズム（逐次処理）で行うできるだけ，沢山の例題を解くようにする	・逐次処理 ・分解 ・評価(デバッグ) ・トライ＆エラー ・複数解の容認
					アルゴリズムアプリ(iPad)：LightBotで遊ぶ	1時間	逐次処理に関する内容のみ	
1-2	• 条件分岐について理解する	1. 日常の生活で，条件によって行動が変わる場面を考える 2. 条件分岐のアルゴリズムを擬似プログラムに変換する 3. 条件分岐の擬似プログラムを実行する 4. 条件分岐の擬似プログラムの間違いを見つける	1-1	I. 条件によって行動が変わるアルゴリズムを考えることができる II. アルゴリズムから適切な条件分岐の擬似プログラムを作成（表現）できる III. 擬似プログラムと実行結果の対応が理解できる	アンプラグド：カードゲームを作ろう（引いたカードの数字とマーク（種類と色）に着目して得点を得るゲームを各自（グループ）でルールを決めて遊ぶ。なお，条件分岐のアルゴリズムをプログラムとして書いたもので遊ぶ事）	1時間	条件が複雑にならないようにする	・条件分岐 ・分解（条件と処理） ・評価(デバッグ) ・トライ＆エラー ・複数解の容認
					アルゴリズムアプリ(iPad)：LightBotで遊ぶ	1時間	分岐処理に関する内容のみ	
1-3	• 繰り返しについて理解する	1. 日常の生活で，繰り返し行動がある場面を考える 2. 繰り返しのアルゴリズムを擬似プログラムに変換する 3. 繰り返しの擬似プログラムを実行する 4. 繰り返しの擬似プログラムの間違いを見つける	1-1（1-2）	I. 繰り返し構造を持つアルゴリズムを考えることができる II. アルゴリズムから適切な繰り返しの擬似プログラムを作成（表現）できる III. 擬似プログラムと実行結果の対応が理解できる	アンプラグド：音楽に合わせてダンスを踊ろう！（ダンスの基本動作を組み合わせてダンスの命令書（プログラム）を作成して遊ぶ。同じ処理が繰り返されている事に注目させる。）	1時間	繰り返しが複雑にならないようにする	・繰返し処理 ・分解 ・評価(デバッグ) ・トライ＆エラー ・複数解の容認
					アルゴリズムアプリ(iPad)：LightBotで遊ぶ	1時間	繰り返し処理に関する内容のみ	
1-4	• 簡単なプログラムについて理解する	1. 逐次処理，繰返し，判断を含むアルゴリズムを考える 2. 擬似プログラムに変換 3. 擬似プログラムを実行する 4. 擬似プログラムの間違いを見つける	1-1, 1-2, 1-3	I. 適切なアルゴリズムを考えることができる II. アルゴリズムから適切な擬似プログラムを作成できる III. 擬似プログラムと実行結果の対応が理解できる	アンプラグド：コップタワーを作るロボットアーム	2時間	コップタワーを建てるアルゴリズムを考える。逐次処理を基本として，繰返し構造を見つけ出す。条件に応じて処理内容を変えるよう工夫する（例えば，違った色のコップを用意して，色によって使うコップを変えるなど）	・繰返し処理 ・条件分岐 ・逐次処理 ・分解 ・パターン認識 ・抽象化（制御構造の抽象化） ・評価(デバッグ) ・トライ＆エラー ・複数解の容認
					アンプラグド：ロボットカー（ロボット役とコントローラー役に分かれて，プログラムに従ってコントローラー役がロボット役に指令を出す）	2時間	指示内容に逐次，繰返し，判断を組み入れる。	・繰返し処理 ・条件分岐 ・逐次処理 ・分解 ・パターン認識 ・抽象化（制御構造の抽象化） ・評価(デバッグ) ・トライ＆エラー ・複数解の容認

表 5　ステージ２の内容（略案）

Stage2（主な対象：小学校3年生から4年生）

Stage2	目的	指導内容	前提知識・技能	到達度確認基準	教具	標準時数	備考	育成すべき知識・技能
2-1	• 逐次処理の簡単なプログラムが書ける	1. 簡単な逐次処理プログラムを書く 2. 簡単な逐次処理プログラムを実行する 3. 簡単な逐次処理プログラムの間違いを見つける	1-1	I. 逐次処理構造を持つアルゴリズムを考えることができる II. アルゴリズムから適切な逐次処理の簡単なプログラムを作成（表現）できる III. 逐次処理の簡単なプログラムと実行結果の対応が理解できる	code.orgの教材 コース2 ステージ3	2時間	https://studio.code.org/s/course2	・逐次処理 ・分解 ・評価(デバッグ) ・トライ＆エラー
2-2	• 条件分岐の簡単なプログラムが書ける	1. 簡単な条件分岐プログラムが書く 2. 条件分岐の簡単なプログラムを実行する 3. 簡単な条件分岐プログラムの間違いを見つける	1-2 (2-1)	I. 条件分岐構造を持つアルゴリズムを考えることができる II. アルゴリズムから適切な条件分岐の簡単なプログラムを作成（表現）できる III. 条件分岐の簡単なプログラムと実行結果の対応が理解できる	code.orgの教材 コース2 ステージ13	2時間	https://studio.code.org/s/course2	・条件分岐 ・分解（条件と処理） ・パターン認識 ・抽象化（制御構造の抽象化） ・評価(デバッグ) ・トライ＆エラー
2-3	• 繰り返し処理の簡単なプログラムが書ける	1. 簡単な繰り返しプログラムが書く 2. 繰り返しの簡単なプログラムを実行する 3. 簡単な繰り返しプログラムの間違いを見つける	1-3 (2-1, 2-2)	I. 繰り返し処理構造を持つアルゴリズムを考えることができる II. アルゴリズムから適切な繰り返し処理の簡単なプログラムを作成（表現）できる III. 繰り返し処理の簡単なプログラムと実行結果の対応が理解できる	code.orgの教材 コース2 ステージ6	2時間	https://studio.code.org/s/course2	・繰返し処理 ・分解 ・パターン認識 ・抽象化（制御構造の抽象化） ・評価(デバッグ) ・トライ＆エラー

ステージ２（表 5 参照）においては，ステージ１で学んだことを，簡易プログラミング言語などを用いて学ぶことで，遊びを通して日常の問題解決の考え方とプログラミングにおける問題解決との類似性について考えることができるよう考慮する．また，ステージ１と同様の知識や技能を反復して学ぶことで,学びを定着させる効果が期待できる．

ステージ３（表 6 参照）は，ステージ１および２で反復して学んだ基礎的な概念などを Scratch などのブロック型プログラミング言語を用いて学ぶ．ステージ３は，Scratch を使ったプログラミング教育とLEGO EV3を用いたプログラミング教育の２部構成となっている．これらは，必ずしも両方を実施する必要はない．STEM 教育を重視し，動く教材を活用する場合は，LEGO EV3 を用いたプログラミング教育を主として，Scratch はステージ２との繋ぎの導入部分として活用することも考えられる．実際にどのように運用するかは，実施する科目の学習目標あるいは児童生徒の興味関心，地域の特性などを考慮して実施することが望まれる．

ステージ４および５においても同様のカリキュラムを用意しているが，考え方の基本としては生徒の生活に密着した題材，あるいは地域社会での問題に対する課題解決型の題材を通して，ステージ３までに修得してきた知識および技能を基にして学びを深化させていくような課題を設定することが重要である．また，ステージ４と５については，すでにプログラミング教育を取り入れている既存の教科があることから，それらとの整合性を保ちながら学ぶ題材を設定することが不可欠でもある．

今後は，これらを指導する指導者の育成が大きな課題となる．以下，これら教育カリキュラムを実践する指導者に必要な知識と指導者養成プログラムについて提案を行う．

4. 指導者に必要な知識と技能

4.1 公教育と私教育における指導者の違いと連携方法

本報告で言うところの指導者は，公教育において授業などを担当する教師（以下，教師と呼ぶ）と私教育の塾などで教える講師（以下，講師と呼ぶ）を指している．教師は，教員免許状を有しており教育全般について専門的な知識を有しているがプログラミングに関する専門的な知識や技能を有しているとはいえない．また，講師はプログラミングに関する専門的な知識を有していても教育全般についての専門性を有しているとは限らない. 民間企業や NPO 法人な

表 6　ステージ3の内容（略案）

Stage3（主な対象：小学校4年生くらいから6年生）

Stage3	目的	指導内容	前提知識・技能	到達度確認基準	教具	標準時数	備考	育成すべき知識・技能
3-1	• 簡易プログラミング言語とScratchの命令の違いについて理解する	1. プログラムの基本3構造（逐次処理，条件判断，繰り返し）の確認 2. 基本3構造を組み合わせた簡単アルゴリズムを書く 3. 基本3構造を組み合わせた簡単なプログラムの間違いを修正する	Stage2全般	I. 基本3構造に対応したScratchプログラムが理解できる II. 基本3構造に対応したプログラムを作成できる III. 基本3構造を組み合わせたプログラムを理解できる IV. 基本3構造を組み合わせたプログラムの間違いを修正できる	• Scratch2 • Stage2で使ったcode.org教材などとの違いを説明した資料 • 基本3構造に対応した問題（code.orgの教材で書いたプログラムをscatchで書くように指示したものから，サンプルを与えずにアルゴリズムからScratchプログラムを作成するなど，レベルに応じた課題を用意）	2時間	使用するScratchはPC版（ブラウザ版）の方が好ましい。	・繰返し処理 ・条件分岐 ・逐次処理 ・分解 ・パターン認識 ・抽象化（制御構造の抽象化） ・評価(デバッグ) ・トライ＆エラー ・複数解の容認
3-2	・ Scratchで使われる基本アルゴリズムの体験	1. 基礎的なアルゴリズムをScratchを使ったプログラムで体験する 2. 基本的なプログラムの間違いを修正できる	3-1	I. 基礎的なアルゴリズムを理解できる II. 複数の基礎的なプログラム間の関係を理解できる III. 基礎的なプログラムの組み合わせを理解できる IV. 基礎的なプログラム（組み合わせも含む）の間違いを修正できる	• Scratch2 • Scratchカード（基本アルゴリズム） • 簡単な応用課題（基本3構造を組み合わせたもの）	3時間		・繰返し処理 ・条件分岐 ・逐次処理 ・分解 ・パターン認識 ・抽象化（制御構造の抽象化） ・抽象化（データの抽象化） ・評価(デバッグ) ・トライ＆エラー ・複数解の容認
3-3	• 簡単なゲームを作成できる	1. 簡単なゲームプログラムを作成する 2. 自分のアイデアに基づいてプログラムの改良ができる 3. 作成したプログラムを発表する	3-2	I. 簡単なゲームプログラムの動作を理解できる II. 簡単なゲームプログラムの間違いを修正できる III. 自分のアイデアをアルゴリズムとして表現できる IV. 自分のアイデアをプログラムとして作成することができる	• Scratch2 • 簡単なゲーム課題 • プレゼンテーション機器	4時間	Scratchのステージを理解するには，2次元座標のグラフ，負の数の計算が必要（日本では，負の数の演算は中学1年生で学ぶ）	・繰返し処理 ・条件分岐 ・逐次処理 ・分解 ・パターン認識 ・抽象化（制御構造の抽象化） ・抽象化（データの抽象化） ・評価(デバッグ) ・トライ＆エラー ・複数解の容認
3-4	• Scratchを使った応用課題	1. ゲーム等を企画する 2. 企画に基づいてプログラムを作成する 3. 作成したプログラムのプレゼンテーションと競技会を実施する	3-3	I. ゲームプログラム等の企画を立てることができる II. 自分のアイデアをアルゴリズムとして表現できる III. アイデアに基づいてゲームプログラム等を作成できる IV. 自らプログラムの間違いに気づき適切に修正ができる	• Scratch2 • 簡単なゲーム課題 • プレゼンテーション機器	4時間		・繰返し処理 ・条件分岐 ・逐次処理 ・分解 ・パターン認識 ・抽象化（制御構造の抽象化） ・抽象化（データの抽象化） ・評価(デバッグ) ・トライ＆エラー ・複数解の容認
3-5	• 簡単なロボットプログラムの作成	1. ScratchとLEGO EV3プログラムの違いについて学ぶ 2. アクチュエータ（Lモーター，Mモーター）の制御の基本を学ぶ 3. 決められたルートをアクチュエータだけを使い走行するロボットカーを作成する 4. 作成したプログラムのプレゼンテーションと競技会（走行タイムと正確性を競う）を実施する	3-1 (3-2, 3-3)	I. Scratchとの命令の違いについて理解できる II. アクチュエータを動かすプログラムの動作を理解できる III. アクチュエータを動かすプログラムを作成することができる IV. 自分のアイデアをプログラムとして表現できる V. 自らプログラムの間違いに気づき適切に修正ができる	• LEGO EV3 • iPad • コース	2時間	自律型ではないので，センサーを使わない	・繰返し処理 ・条件分岐 ・逐次処理 ・分解 ・パターン認識 ・抽象化（制御構造の抽象化） ・抽象化（データの抽象化） ・評価(デバッグ) ・トライ＆エラー ・複数解の容認
3-6	• 簡単な自律型ロボットプログラムの作成	1. センサー（カラーセンサー，タッチセンサー，距離センサー，ジャイロセンサー）の動きの基本を学ぶ 2. センサーを使ってライントレースなどを行う自律型ロボットを作成する 3. 作成したプログラムのプレゼンテーションと競技会を実施する	3-5	I. センサーの特徴が理解できる II. センサーを使ってアクチュエータを制御するプログラムを作成することができる III. 自分のアイデアをプログラムとして表現できる IV. 自らプログラムの間違いに気づき適切に修正ができる	• LEGO EV3 • iPad • コース	2時間	自律型のロボットを作成する（センサーを使う）	・繰返し処理 ・条件分岐 ・逐次処理 ・分解 ・パターン認識 ・抽象化（制御構造の抽象化） ・抽象化（データの抽象化） ・評価(デバッグ) ・トライ＆エラー ・複数解の容認
3-7	• 自律型ロボットの応用課題（サッカー競技ロボット）	1. センサーとアクチュエータを使う自律型ロボットの課題（RoboCup準拠のサッカーロボット）に基づいてプログラムを作成する 2. 作成したプログラムのプレゼンテーションと競技会を実施する	3-6	I. センサーを使ってアクチュエータを制御するプログラムを作成することができる II. 自分のアイデアをプログラムとして表現できる III. 自らプログラムの間違いに気づき適切に修正ができる	• LEGO EV3 • iPad • コース	2時間	その他の課題の例として，お掃除ロボットなどもある	・繰返し処理 ・条件分岐 ・逐次処理 ・分解 ・パターン認識 ・抽象化（制御構造の抽象化） ・抽象化（データの抽象化） ・評価(デバッグ) ・トライ＆エラー ・複数解の容認

どとの協働的な活動が求められている中（例えば参考文献[1][3]参照）で，プログラミング教育を地域社会と一体的に進めるには大きな課題であると考えられる．

そこで，まず，今後のプログラミング教育を定着させるために必要な教師と講師の役割と協調関係について検討する．一つの方法は，お互いの専門性を補うような協働的な

活動が考えられる．例えば，学校でのプログラミング教育に講師がプログラミングの専門家として参加すること（図 2 の（a））や，地域のプログラミング教室に教師が教育の専門家として子供たちの学びの支援(ファシリテーター役)を行うことなどが考えられる．このような協働的活動を行うには，お互いの専門に対する理解が必要でありお互い対等な立場として活動することが重要である．そうでない場合，主–従の関係が生まれお互いに不満がたまり，よい結果が生まれない可能性が高くなると考えられる．

２つ目の方法としては,従来の学校と塾のような関係(図 2 の（b)）や，学校の部活動における外部コーチのような関係である．この場合は，学校での学びを深化させたい児童生徒の希望を私教育において充足するような活動となる．この場合も，教師と講師がお互いの学びの内容を知ることで児童生徒にとって効果的な学びが期待できると思われる．

いずれにしろ，講師は積極的に学校などの公教育に関わり,学校側も積極的に受け入れる体制が必要である.また，教師と講師共に指導者として,教育の専門的な知識や技能，そしてプログラミングの専門的な知識や技能を学び，対等な立場でプログラミング教育を行うことが重要であると考えられる．

我々は，そのために必要な知識や技能を整理して指導者養成プログラムのカリキュラムの構築を行っている．

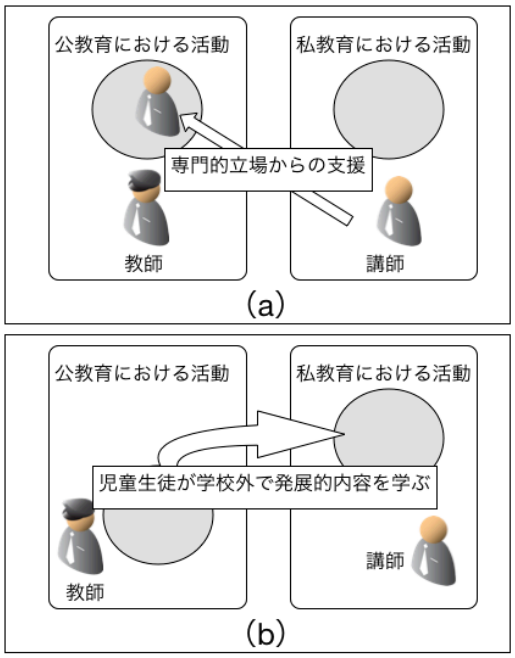

図 2　公教育の私教育の関係例

4.2 指導者に求められる知識と技能

指導者に求められる知識と技能は大きく「プログラミングに関連した知識と技能」および「教育に関連した知識と技能」の２つに分類される．プログラミングに関連した知識と技能は,表 1 および表 2 に示した知識や技能およびその周辺領域と言える．また，教育に関連した知識や技能としては，教育方法，教材開発，授業分析と評価，学習者心理，特別支援などが考えられる．表 7 に指導者に求められる専門的な知識と技能の概略についての一覧を示す．

表 7　指導者に求められる知識と技能（概略）

区分		知識や技能
プログラミングに関連した知識と技能	CT の基礎概念	CT による問題解決
	IT に関する知識	プログラミング言語
		情報通信技術
		情報モラル
	CS に関する知識	コンピュータサイエンス
教育に関連した知識	児童生徒の心身の発達および学習の過程	学習者の心理
		特別支援教育
	教育の方法および技術	教育方法
	指導法	教科の指導法
		教材開発論
	教育実践演習	教育場面分析
		教育実践

特に，2016 年 4 月に施行された「障害を理由とする差別の解消の推進に関する法律」により，合理的配慮を可能な限り提供することを，行政・学校・企業などの事業者に対して求めている．したがって，指導者は，障害学生に対する合理的配慮についての理解を深めておき，障害学生の求めに応じて過度にならない程度の支援を行う必要がある．このように，これからの教育に関しては，特別支援に関する幅広い知識と技能が有することことが重要になってくると考えられる．本指導者養成カリキュラムでも，障害学生への特別支援，引きこもりや不登校などの児童生徒への対応をカリキュラムに反映する予定である．

5. 指導者養成カリキュラム

表 7 の内容を指導者養成カリキュラムとして構築したものが表 8 である．特徴としては，表 7 で示した知識や技能毎に科目を立てるのではなく，可能な限り科目の中でストーリーラインを構成して複数の区分あるいは知識や技能を関連付けるように工夫していることである．例えば，IT や CS に関する知識などは，教材開発の中で教材と関連させるなどである．また，授業分析とワークショップ演習では，ワークショップ演習で行う子供向けのプログラミング講座を授業分析の対象授業とするなども行う．

本カリキュラムの，対象者は教師および講師や社会人で

ある．また，総時間数は 120 時間程度となり 1 年間で学ぶカリキュラムとなっている．また，原則として学習は，e ラーニングとスクーリングを組みあわせたブレンデッド・ラーニングを用いる事を想定している．

表 8　指導者養成プログラムのカリキュラム案（概略）

区分	科目名	チャプター	主なセクション
基礎論	プログラミング教育概論	プログラミング教育の現状とこれから	公教育におけるプログラミング教育の現状
			私教育におけるプログラミング教育の現状
			海外におけるプログラミング教育の現状
			2020年度以降のプログラミング教育
		プログラミング教育の必要性	21世型能力との関係
			コンピュテーショナル・シンキングとの関係
			社会からの要望
		プログラミング教育に必要な知識と技能	アルゴリズムとプログラミング
			問題解決とプログラミング
		プログラミング教育に使われる教材	アンプラグド教材を用いたアルゴリズム教育
			ビジュアル型言語を用いたプログラミング教育
			テキスト型言語を用いたプログラミング教育
		アンプラグド教材を用いたプログラミング	基本3構造のアンプラグド教材
			抽象化，分解，パターン認識
プログラミング技術	Scratchプログラミング入門	Scratchとは	Scratchの特徴
			Scratchで作られた作品
			Scratchの使い方
			Scratchの命令ブロック
		Scratchによるプログラムの基本構造	逐次処理のプログラム
			繰り返し処理のプログラム
			条件判断のプログラム
			基本3構造を組み合わせたプログラム
		シューティングゲームの制作	ゲームの概要
			Scratchによるプログラミング
			シューティングゲームの改良
		ゲームの創作	設計
			プログラミング
	Pythonプログラミング入門	（省略）	
	Swiftプログラミング入門		
	Arduinoプログラミング入門		
教材開発	プログラミング教育における教材開発	体系的なプログラミング教育のカリキュラム	カリキュラムの概要
			各ステージについて
		教材開発の方法	教材開発
	Scratchによるプログラミング教材の開発	簡単な教材作成	風力発電所を作ろう
		教材開発演習	課題研究
	Pythonによるプログラミング教材の開発	簡単な教材作成	IoT教材を作ろう（仮称）
		教材開発演習	課題研究
	Swiftによるプログラミング教材の開発	簡単な教材作成	日記アプリを作ろう（仮称）
		教材開発演習	課題研究
	Arduinoによるプログラミング教材の開発	簡単な教材作成	LEDライトを制御しよう（仮称）
		教材開発演習	課題研究
授業分析・評価	授業分析	授業分析法	
		授業分析演習	
	ワークショップ演習	ワークショップの実施	
心身の発達	学習者心理	学習者の心理状態について	
	特別支援教育	肢体不自由など障害者への対応	

6. 指導者養成プログラムの実施環境

本プログラムの受講生の多くは，教師などの社会人であることから，学びの時間は個々で異なり不定期となることが予想される．例えば，休日にまとめて受講，通勤電車の中での受講，さらに勤務の空き時間などに受講することが考えられる．利用する端末もスマートフォンやパソコンなど受講生の都合に合わせて利用できる必要がある．また，スクーリングは長期休暇や週末に実施される．スクーリングの教育効果を上げるために，e ラーニングでの学習履歴を管理して学習状況を把握しておくことも重要である．

よって，本プログラムの実施においては，パソコンのみの利用環境ではなくスマートフォンでの利用が可能であり，学習者の学習履歴を管理が可能なキャスタリア製の Goocus（図 3 参照）を用いる．また，レポート課題などの提出，質疑応答などは，Google Apps for Education のアプリである Classroom などを用いて行う．

図 3　e ラーニングシステムの画面

教員養成大学における教育実習や授業場面分析などの経験を生かし，本プログラムで行うスクーリングは，児童生徒向けのプログラミング教育ワークショップの開催，その分析・評価および実践についてのプレゼンテーションなどを行う．これによって，実践的な能力の育成を図ることが可能となると考えられる．

7. おわりに

我々は，初等・中等教育におけるプログラミング教育の課題の一つの解として，初等・中等教育における体系的なプログラミング教育のカリキュラムに対応した指導者養成プログラムに関するカリキュラムと学習環境の開発について研究を行ってきた．本報告では，初等・中等教育において児童生徒に修得させたい知識や技能について検討を行い，それに基づくプログラミング教育の体系的なカリキュラムについて提案を行っている．さらに，それを実施する指導者に求められる知識，技能を修得するための指導者養成プログラムと学習環境について提案した．

今後，提案した２つのカリキュラム（教育カリキュラムと指導者育成カリキュラム）の評価実験を半年間ほどかけて行う予定である．それにより，2 つのカリキュラムの問題点や実施上の課題について整理を行い，内容の充実を図る予定である．

謝辞　本研究を推進するに当たり，教育カリキュラムの検討・評価のための児童生徒向けのプログラミング教室開催および指導者養成プログラムの検討・評価にご協力頂いた学校法人信学会およびギークラボ長野の皆様に，謹んで感謝の意を表します．また，本研究の一部は JSPS 科研費 JP26350271 の助成を受けたものです．

参考文献

[1] 総務省:“プログラミング人材育成のあり方に関する調査研究報告書”，総務省，(2015).
[2] 品川区立京陽小学校：“「デジタルテクノロジーの書き手を育てる」～豊かな言語能力の育成を目指して～”，平成 27 年度校内研究，http://school.cts.ne.jp/912keiyo/kounaikenkyu/kenkyu.html,（平成 27 年 6 月 6 日閲覧）
[3] 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（議論のとりまとめ）（案）”，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 回配付資料，資料 1，文部科学省，(2016)
[4] 大森康正，山脇智志:“小中高校生を対象とした体系的なプログラミング教育に関する試案”，日本産業技術教育学会第 27 回北陸支部大会，Vol.27，A-4 (2015).
[5] 大森康正，山脇智志，栗林聖樹:“初等・中等教育を対象とした体系的プログラミング教育カリキュラムの開発”，第３１回情報分科会（佐賀）研究発表会，Vol.31，3，pp.5-8(2016)
[6] 久野靖，和田勉，中山泰一:“初等中等段階を通した情報教育の必要性とカリキュラム体系の提案”，情報処理学会論文誌教育とコンピュータ，Vol.1，No.3，pp.48-61(2015)
[7] 大森康正，磯部征尊，寒川達也，山崎貞登：“2014 年実施のイングランドのナショナルカリキュラム 「Design and Technology」 と「Computing」の改訂 に対する STEM 教育運動の影響”，日本産業技術教育学会誌，第 56 巻，第 4 号，pp.239-250 (2014).
[8] 文部科学省:“情報ワーキンググループとりまとめ（たたき台案）”，教育課程部会　情報ワーキンググループ（第 7 回）　配付資料，http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/059/siryo/__icsFiles/afieldfile/2016/05/31/1370666_2.pdf　（平成 28 年 6 月 6 日閲覧）