初等中等教育における情報活用と プログラミング教育の系統的カリキュラムの設計とその展望

小澤　雄生・原田　信一
（京都教育大学附属京都小中学校・京都教育大学）

The Design and Prospects of a Systematic Curriculum for Information Utilization and Programming Education in Primary and Secondary Education

Yu OZAWA・Shinichi HARADA

2024 年 9 月 30 日受理

抄録：本研究は，2019 年度から初等教育（小学校 3 年生）から中等教育（中学校 3 年生）において実施してきた系統的なプログラミング教育の実践をカリキュラムとして統合し，その展望を考察するものである。小学校では担任教員が，中学校では技術科の教員がプログラミング教育を担当し，ビジュアルプログラミングから簡易言語，さらには本格的なプログラミング言語を段階的に学習している。また，教科横断的な実践も行い，プログラミング的思考や情報活用能力の育成に役立つ実践を目指している。教育方法としてはペア学習やプロジェクトベース学習などを採用し，年間 20 時間程度の授業を実施している。これらの実践を通じて，情報リテラシーやプログラミングスキルの向上が確認された。本論文は，先進的な海外の取り組みを参考にしながら，これまでの実践を基にカリキュラムの部分を系統的にまとめ，それらがプログラミング教育を行う学校における指標となることを目指している。

キーワード：初等中等教育，教育課程，プログラミング教育

Ⅰ．はじめに

小・中学校でのプログラミング教育は，子どもたちの論理的思考力や問題解決能力を養うことが期待されている。プログラミング教育は，導入が進んでいるものの具体的なカリキュラムの設計や実施方法に関するガイドラインは整備が十分とは言いきれない。そのため，多くの自治体や学校が独自の試行錯誤を繰り返しており，教育の質や効果にばらつきがあると推察される。この問題を解決するためには，体系的かつ効果的な教育課程の設計が必要不可欠である。本稿では，実際に実践してきたプログラミング教育を基に，カリキュラムの具体的な構成とその評価をまとめ，今後の知見としていくことを目的としている。

現在，京都教育大学附属京都小中学校（以下，附属京都小中学校）において，系統的なプログラミング教育を実践している。それらで得た知見は，高等学校への接続と他附属，さらには他の小・中学校などに方法や実践が汎用できるように拡大していくことが重要である。現在，京都教育大学教育研究交流会議技術分科会（以下，技術教育分科会）の構成員で，小・中・高等学校の系統的なプログラミング教育推進の共同研究を進めている。そのため，附属京都小中学校の実践などから経たカリキュラムを拡充できるようにこれまでの取り組みを整理し，今後汎用性のある実践・方法へと繋げられるようにカリキュラムをまとめ，現時点で生徒にどれだけ学習効果が表れているかを調査した。それらから，今後のプログラミング教育の方向性などを定めるための基となるカリキュラムの設計と基礎的なデータを得ることができた。また，日本のプログラミング教育は，現行の学習指導要領から本格的にスタートしたため，先行研究が少ないことから海外の実践的・体系的な取り組みを参考にした。

Ⅱ．プログラミング教育の理論と実践

1．プログラミング教育の歴史と現状

プログラミング教育は 20 世紀中頃，米国の大学や研究機関で始まったが，1970 年代には初等教育にも導入され，専門的な技術から一般教育の一部へと広がった。1960 年代に Papert が考案したロゴプログラミングは，子どもたちが直感的にプログラムを書くための環境を提供し，教育の基盤を築いた。1980 年代から 90 年代にかけては，コンピュータの普及とともに BASIC などの初心者向けプログラミングが教育現場に導入され，子どもたちがプログラミングに触れる機会が増加した。2000 年代にはインターネットの普及により，Scratch などのツールが開発され，プログラミング教育は新たな段階に進化した。

2．世界のプログラミングカリキュラムの比較

現在，世界の多くの国がプログラミング教育を導入しており，その内容や方法は各国で異なる。本節では，エストニア，イギリス，アメリカ，フィンランドの 4 か国におけるカリキュラムを比較し，各国の特徴と教育効果を考察する。また，選定理由は，この 4 か国が様々な取り組みや内容を公開しているためである。

（1）エストニアのプログラミング教育

エストニアでは 2012 年に「Proge Tiger」というプログラムが開始され，ICT スキルやデジタルリテラシーの向上を目指している。小学生にはビジュアルプログラミングツールを，中高生には「Python」や「JavaScript」などのテキストベースの言語を用いたプログラム作成や問題解決を学習させている。エストニア政府は，全学生が ICT スキルを習得できるようカリキュラム改革を進めている。

（2）アメリカのプログラミング教育

アメリカでは「K-12」教育でプログラミングを必須とする州が増加している。特に Code.org などの非営利団体や，世界中で行われる「Hour of Code」イベントが教育推進に大きな役割を果たしている。連邦政府もコンピュータサイエンス教育への助成を強化しており，全学生がプログラミングにアクセスできるよう支援している。

（3）イギリスのプログラミング教育

イギリスは 2014 年，世界に先駆けてプログラミング教育を全国カリキュラムに取り入れた。小学生は Scratch などのビジュアルプログラミングを学び，中学生には「Python」や「JavaScript」が導入される。イギリス政府は「Computing at School (CAS)」などの取り組みを通じ，教師支援やリソースの提供にも力を入れている。

（4）フィンランドのプログラミング教育

フィンランドは 2016 年からプログラミング教育を正式にカリキュラムに組み込み，数学や理科と統合した形で実施している。特に探究型学習やプロジェクトベースのアプローチが重視され，初等教育ではビジュアルプログラミング，中等教育では「Python」などのテキストベースの言語が使用される。しかし，フィンランド教育文化省は，2023 年 1 月に Building Review というレポートを発表しており，プログラミングについては書かれていないが，その中で学習成果や教育水準の低下を公表している。

（5）各国のカリキュラムにおける共通点と相違点

共通点として，いずれの国もプログラミング教育を早期に導入し，論理的思考力や問題解決能力の育成を目的としている。また，Scratch などのビジュアルプログラミングツールが初学者の学習に効果的であるとされている。一方，各国のカリキュラムの具体的な内容や進行速度には違いがあり，エストニアでは ICT 教育の一環としてデジタルリテラシーの向上を，フィンランドは探究型学習を重視している。アメリカは非営利団体のリソース活用が盛んであり，イギリスはコンピュータサイエンス全般の教育にも力を入れている。

（6）日本のカリキュラムとの比較

日本では 2020 年度から小学校でプログラミング教育が必修化され，他国と同様，論理的思考力の育成が目的である。特に教科横断的なアプローチにより，算数や理科と連携して学習が進められる。プログラミングが他の教科の学習にも役立つよう配慮されて，学びの一環として問題解決力や情報リテラシーの基礎を養う構成となっている。しかし，日本では伝統的な教科書中心の学習が根強く，教科横断的なアプローチが学校や自治体ごとにばらつきが見られる点も課題である。また，アメリカなどと比較すると，民間や NPO による教材や支援の活用が十分ではなく，プログラミング教育を効果的に進めるためには，さらなる教育インフラの充実が求められる。

3. プログラミング教育の効果と課題

プログラミング教育は，学習者に様々な効果をもたらす一方で，その実施にはいくつかの課題がある。本節では，プログラミング教育が学習者に与える具体的な効果と，その実施における主要な課題について，諸外国の状況を踏まえて述べる。

（1）プログラミング教育が学習者に与える効果

プログラミング教育は，論理的思考力や問題解決能力の向上に効果があるとされている。Wing (2006) は，プログラミングを通じて得られる計算的思考（Computational Thinking）の重要性を強調している。計算的思考とは，問題を解決するための抽象的な思考プロセスであり，プログラミング教育を通じて学習者がこのスキルを身につけることができると述べている。また，Sengupta ら (2013) は，プログラミングが科学教育における学習者の計算的思考を促進し，科学的探究心や問題解決スキルの発達を支援することを示している。

さらに，プログラミング教育は創造力やチームワークの育成にも役立つとされている。Resnick (2009)らは，「Scratch」を用いたプログラミング教育が，直接的にチームワークを向上すると検証していないが，学習者に他者とプロジェクトを完成させる経験を提供するとしている。

（2）プログラミング教育実施における課題

プログラミング教育の実施には，いくつかの課題が存在する。まず，指導者のスキル不足が大きな障壁となっている。Yadav (2016)らは，プログラミング教育を効果的に実施するためには，指導者自身がプログラミングの知識とスキルを持っている必要があると指摘している。しかし，多くの指導者がプログラミング教育の専門的な訓練を受けていないため，教育現場での実施が困難であると指摘している。

さらに，適切な教材の不足も課題となっている。Grover and Pea (2013) は，プログラミング教育のための教材やカリキュラムの整備が未だ不十分であることが，教育の実施とその質に課題をもたらしていると指摘している。特に，K-12 の学習者に適した教材の開発と提供が重要であると強調している。

Kafai and Burke (2014) は，プログラミング教育における計算的思考の重要性と，その教育の実践方法について論じている。彼らはプログラミング教育の拡充において，指導者の専門性やカリキュラム開発の重要性を強調しており，教育環境の整備と指導者のトレーニングがプログラミング教育を効果的に実施する上で不可欠であることを示している。昨今，教育インフラが整ってきた日本での課題は，指導者のスキル不足や専門的な訓練，プログラミング教育の教材の少なさなどであるが，専門的な指導者の訓練は一朝一夕ではいかない。そのため，現在のプログラミング教育を充実させていくことが解決の糸口になると考えられる。これらの解決策として，様々な実践やカリキュラムを拡大させていくことがますます重要で近道であると考える。

Ⅲ．系統的カリキュラムの提案とその評価

1. カリキュラム提案のフレームワーク

（1）カリキュラムの目的と内容の構成

提案するカリキュラムの目的は，児童生徒にプログラミングスキルを含んだ情報活用能力全般を効果的に習得させることである。本カリキュラムは，学習者が次世代の情報社会において必要とされる能力を身に付けるための包括的な教育プログラムとして設計していきたいと考える。

第一は，プログラミング的思考の育成である。プログラミング的思考は，前章の計算的思考と似ている。文部科学省（2019）では，問題を論理的かつ系統的に解決するための基盤となる思考法であり，プログラミング学習を通して学習者がこの能力を習得することを目指している。これにより，複雑な問題を分解し，アルゴリズムを設計したり，実行可能な解決策を構築する能力を養うことができるとしている。

第二に，問題解決能力の向上である。プログラミング教育を通して学習者が具体的な課題や抽象的な課題に取り組むことで，実践的な問題解決能力を培う。具体的なプロジェクトや課題を通じて，学習者は創造的かつ効率的に問題に対処するスキルを習得し，実社会のあらゆる課題に対するアプローチの仕方を学ぶことができるようになるためである。

第三に，プログラミングの基礎技術の習得である。これには，基本的なプログラミング言語の理解と活用，データ構造やアルゴリズムの基礎知識の習得が含まれる。児童生徒は，これらの技術を学ぶことで，自らのアイディアを具体的なプロジェクトとして具現化できる能力を習得することができる。

さらに，本カリキュラムは，情報リテラシーの向上も目的としている。情報リテラシーとは，情報の収集，評価，活用等の能力を指し，情報社会において適切な判断をするために不可欠である。これは，児童生徒がデジタルツールを効果的に使用し，情報の正確性を評価し，適切に情報を管理・活用する力を養うためである。

以上の目的を達成するために，発達段階ごとに主要なモジュールを構成した。これらのモジュールを通じて，児童生徒は段階的に知識と技能を習得し，実践的な能力を身に付けることができる。

実践よりプログラミングのモジュールは，4 つの段階で区切ることが望ましいと考えられた。小学校から中学校にかけて行うプログラミングは，プログラマーやエンジニアを育てる目的ではない。表 1 で示したように経験して，理解して習得し，目的である思考力やチームワークを養うことが重要だと考えられる。そのため，表 1 のどこまでを行うかをはっきりさせ，次の段階の指導者がそれを理解して継続してできることの方が重要である。

（2）教育方法

教育実践の方法は，初期段階では一斉授業による教授型で行っていたが，この方法では授業時数の制約から生徒の理解度や進度，満足度が思うように伸びなかった。そのため，様々な教育方法を試みる必要があった。世界的に見ると，プログラミング教育では多様なアプローチが提案されている。まず，日本でも多くの教育現場で導入されているアクティブラーニングは，児童生徒が主体的に学び，指導者がファシリテーターとしてサポートする教育方法である。ディスカッションやグループワークを通して，生徒同士の協力とコミュニケーションを強化することができる。Smith ら（2005）は，アクティブラーニングによってエンゲージメントが促進され，学習の効果を高めることを示している。

次に，ペアプログラミングは，2 人 1 組でプログラムを作成し，ローテーションやフィードバックセッションを取り入れることで協力的な学習環境を提供する手法である。Williams and Kessler（2000）は，ペアプログラミングが学習者間の協力関係を強化し，互いに知識やスキルを共有しながら学ぶことができる効果的な学習方法であることを示している。また，小学校 3・4 年生ではペア学習を導入することで効果的な学習成果が得られるということは確認している。

さらに，プロジェクトベース学習（PBL）は，学習者が実際のプロジェクトを通じて学習する方法であり，学習者が興味のあるテーマを選び，プロジェクトを企画・実行することで実践的な知識の応用を促進する教育方法である。Blumenfeld ら （1991）は，PBL が学習者の主体的な学びと高いモチベーションを生み出すことを示している。本研究でも中学生主体の PBL を実施してきたが，その有効性を示す段階までは至っていない。

これらの教育方法の導入により，プログラミング教育は従来の一斉授業から，児童生徒が主体的に学ぶための学習環境へと変化し，より効果的な学びを実現することが期待される。

表 1　プログラミングの系統的カリキュラムモジュール

期	モジュール名	学習内容	目標
1	入門モジュール	ビジュアルプログラミングのゲームや遊びを通じて，楽しんで学びながら基本的なロジックとフローを体験・理解する	遊びや教科学習を通してプログラミング的思考の概念を習得
2	基礎概念モジュール	ビジュアルプログラミングや簡易言語を通じて，基本的なプログラミングの技能（変数，条件分岐，ループなど）を学ぶ	教科学習や総合学習と関連させて基本的なプログラミングスキルの理解
3	初歩的プログラミングモジュール	簡易言語やプログラミング言語を通じて，基本的なプログラミング技術を実体化（プログラム）する技能を学ぶ	簡易・プログラミング言語で基本的なプログラミングスキルを習得
4	応用プログラミングモジュール	プログラミング言語を通じて，プログラミング技能の向上と学んだ内容から独自のプロジェクトを実践する。	プログラミングを通して問題解決能力の向上や高度な知識・スキルの習得

2. カリキュラムの体系的構造と内容

本カリキュラムは，小学校 1 年生から中学 3 年生までの生徒を対象とし，段階的かつ体系的にプログラミングスキルと情報活用能力を育成することを目的として設計されている。具体的には，プログラミング的思考の育成，問題解決能力の向上，基礎的なプログラミング技術の習得を主要な目標として掲げ，学年ごとに適切な学習内容を実施しながら改良して設定した。

プログラミング的思考の育成は，問題を論理的に分解し，解決策を設計・実行する力等を含んでいる。これら能力は，特に初期段階においてビジュアルプログラミングツールを用いることで，直感的かつ効果的に児童に伝えることで自然と身に付けることができ，プログラミングを身近に感じさせる効果的な習得方法である。ビジュアルプログラミングは，発達段階が進むにつれて言語的な理解力の向上により，児童生徒にとってはわかりやすいものの，プログラミング的思考を十分に養えない場合があった。それらの実践より，これらの方法は小学 3・4 年生での導入が適切であることを確認している。

次に問題解決能力の向上は，実際のプログラミング課題を通じて具体的な問題に取り組むことにより，習得した知識・技能を活用し，効率的な解決策を見出す力を育むものである。この過程で，児童生徒は現実世界の問題に対するアプローチを学び，応用力を身につけている。そのため，ビジュアルプログラミングからより高度なプログラミング言語を用いて理解を深める必要があると考える。

また，基礎的なプログラミング技術の習得は，変数や条件分岐，ループ，基本的なアルゴリズムやデータ構造の理解とそれらの応用が含まれる。系統的にプログラミングを学ぶ上では，プログラミング自体の技能を習得させることで既存の情報や経験をもとに，頭の中で新しいイメージや状況を思い描くことができる。これは，物事の解決策を頭の中で構築するプロセスに不可欠であると考える。これらの知識は，プログラミングの基盤を形成し，より高度な技能や知識の習得や新しいものやアイディアを実際に生み出す創造のプロセスに向けた重要なステップである。これらの発達段階は，小学校 5・6 年生で簡易言語の使用により基本的なプログラミングを経験し，中学生段階において，実際のプログラミング言語で確認や知識の落とし込みを行うことで定着した知識と活用できる知識・技能になることが確認できている。

これらを踏まえて資料 1 では，小学校 1 年生から中学 3 年生までの各学年における情報活用能力とプログラミング学習の具体的な学習目標，学習内容，および習得時期などを体系的に整理したものである。この資料 1 は，各学年でのプログラミング教育の進行を一目で把握できるよう設計しており，教育現場における実践を支援するためのものである。プログラミングや情報に関する各種資料や参考書の目標および内容を参考にして作成されている。ただし，情報リテラシーや情報活用能力の指導内容は，使用する PC や OS によって適宜改良が必要である。

Ⅳ. 系統的プログラミング教育の有効性評価に関する検証

1. 情報リテラシー評価の調査

（1）BASIC（The Big6 Information Literacy Skills Self-Assessment）の概要

情報リテラシーは，現代社会において不可欠なスキルであり，情報の検索や評価，活用を効果的に行うための能力を含んでいる。BASIC（The Big6 Information Literacy Skills Self-Assessment）は，Eisenberg and Berkowitz（1990）によって提唱された Big6 スキルをもとに，情報リテラシーの 6 つの主要なプロセスを評価するための自己評価ツールである。Big6 スキルは，教育現場で幅広く採用されており，学生が情報を効果的に活用するための包括的なガイドラインを示している。この BASIC は，情報の問題解決プロセスをモデル化し，そのプロセスを評価するための具体的な自己評価項目が設けられている。これにより，学生がどの段階まで達成しているかを把握し，どこに改善の余地があるかを明確にすることができる。これは，学生向けに作成されているが，段階的に学習している中学生にも理解できる内容であるため採用できると判断した。文部科学省（2019）は，「情報リテラシー」を情報に対する基本的なスキルと定義し，それに基づいて具体的な行動や成果を生み出す力を「情報活用能力」としている。しかし，BASIC の質問項目には両方の要素が含まれているため，本研究では

「情報活用能力」と「情報リテラシー」を同義語として扱う。また，このアンケートは，情報リテラシーを評価するものであるため，以下ではアンケートを指す際に『情報リテラシー』という用語を用いる。

（2）アンケート項目の構成

BASIC のアンケートは，情報リテラシーの各段階に対応する項目で構成されている。まず，Task Definition（課題定義）段階では，生徒が解決すべき問題を明確にし，必要な情報を特定する能力を評価することができる。次に，Information Seeking Strategies（情報収集の方略）段階では，どの情報源を選択し，どのように検索するかを判断する能力を測定することができる。次に，Location and Access（場所とアクセス）段階では，選択した情報源から必要な情報を適切に取得できるかを評価することができる。Use of Information（情報の利用）段階では，取得した情報を理解し，目的に応じて効果的に利用できるかが問われている。さらに，Synthesis（統合）段階では，複数の情報を組み合わせて新たな知見を形成する能力が評価され，最後に，Evaluation（評価）段階では，自分の情報処理のプロセスや成果物の有効性を振り返る能力を測定することができる。このように，BASIC アンケートは，情報リテラシーの各段階を包括的にカバーしており，生徒の情報活用能力を多角的に評価することが可能である。

2. プログラミング活用能力評価の評価

（1）プログラミング活用能力の重要性

プログラミングを継続的に学び，それを活用する能力は，児童生徒が論理的思考力や問題解決能力を養う基盤であり，デジタル社会における創造的な活動や革新的な取り組みを支える重要なスキルである。プログラミングの継続的な学習は，知識の深化とスキルの向上につながり，学習者が将来的に直面する複雑な課題に対応できる能力を育むことができる。特に，学んだ内容や方法を実際の問題解決に応用し，自らのアイディアを具体的な形にする力は，プログラミング教育の最も重要な利点である。

先行研究では，Lye and Koh（2014）は，K-12 教育における計算的思考の指導に関する文献をレビューし，プログラミングが学生の論理的思考や創造力，問題解決能力を向上させる可能性を指摘している。一方，Grover and Pea（2013）は，K-12 教育における計算的思考教育の現状をレビューし，プログラミング教育このスキルの育成に重要であり，21 世紀の様々な職業で必要とされる可能性があることを示している。

これらの研究は，プログラミング活用能力が一時的なスキルではなく，学習者の将来にわたって重要な役割を果たすものであることを示している。したがって，プログラミング教育においては，単に技術を教えるだけでなく，それをどのように継続的に学び，活用するかを重視したカリキュラム設計が求められる。本研究では，児童生徒がプログラミングを通じて得たスキルの習得状況とその意識を評価することが重要であると考える。

（2）アンケート項目の構成

プログラミングの授業理解度を評価するためのアンケートは，プログラミングの基本概念から応用スキルまでを網羅するように設計されるべきである。しかし，本研究の対象となる生徒は，ゲーム作成や現実の問題解決などの応用スキルまでは学習していないため，アンケート項目は基礎的なスキルに限定した。まず，プログラミングの基本的な概念の理解を判断できる項目を含んだ。これは，変数や条件分岐，ループ構造など，プログラムの基本的な要素が理解できているかを問う質問である。次に，プログラミング言語の文法や構文をどの程度理解しているかを評価する項目である。これらの項目は，生徒が学んだプログラミングを正しく使用し，間違えを判断し修正する力を評価するものである。さらに，プログラムの作成と実行に関する項目では，実際にプログラムを書き，実行できるかどうかを評価する。最後に，生徒のプログラミング学習に対する態度や授業への感想を評価する項目を追加することで，プログラミング教育の有用性や生徒の学習意欲を総合的に測定できるようにした。

3. 調査結果の分析と考察

（1）調査対象と方法

本調査は，2024 年 8 月に 7 年生と 9 年生（中学 1 年生・中学 3 年生）を対象に実施した。7 年生の調査対象

は，91 名であり，9 年生の調査対象は，86 名である。7 年生は，ビジュアルプログラミングから簡易言語を経て「Python」の学習を始めた段階であり，プログラミングの基礎的な概念の理解を深めている段階である。一方，9 年生は，「Python」の基礎を一通り学習しており，よりレベルの高い多くのスキルを持つことが期待される。

本調査は，情報リテラシーの質問 12 項目とプログラミングの授業理解度の質問 14 項目である。いずれも「5：とても当てはめる，4：ある程度当てはまる，3：どちらとも言えない，2：あまり当てはまらない，1：全く当てはまらない」から最も当てはまるものを選択する 5 件法で回答を求めた。本調査の結果は，男女差の分析を目的とせず，全体としての傾向を把握するために実施した。これらの前提の下，7 年生と 9 年生の異なる学習進度が情報リテラシーおよびプログラミング能力の評価にどのように影響を与えるかを検討する。

（2）分析結果と考察

情報リテラシーおよびプログラミング活用能力に関する各項目について，7 年生と 9 年生の間で統計的に有意な差があった項目を中心に考察を行う。これらの結果は，2 つの表に分けて示し，各スキルセットにおける学年間の差異を明確にする。

まず，情報リテラシーの各項目について分析した結果，「情報の利用」において 7 年生と 9 年生の間に有意な差が認められた。7 年の平均値が 3.75（SD = 0.74）に対し，9 年では平均値 4.09 でより高く, SD = 0.60 ，t 値は 3.25 であり，p < .001 という極めて有意な差が見られた。また，効果量（Cohen's d）は 0.503 であり，これは中程度の効果を示している。次点として，「統合」においても有意な差が見られた。7 年の平均値 3.62,

表 2　情報リテラシーにおける学年差の統計結果

	学年	平均値 (M)	標準偏差 (SD)	t値 (t)	有意水準 (p)	効果量 (Cohen's d)
1. Task Definition	7年	4.02	0.58			
	9年	4.17	0.45	1.86	$p = .065$	0.29
2. Information Seeking Strategies	7年	3.88	0.74			
	9年	4.15	0.55	2.76	$p < .01$	0.43
3. Location and Access	7年	3.94	0.71			
	9年	4.04	0.64	0.98	$p = .332$	0.15
4. Use of Information	7年	3.75	0.74			
	9年	4.09	0.60	3.25	$p < .001$	0.50
5. Synthesis	7年	3.62	0.85			
	9年	3.98	0.68	3.00	$p < .01$	0.46
6. Evaluation	7年	3.86	0.68			
	9年	4.12	0.59	2.67	$p < .01$	0.41

（SD= 0.85）に対し，9 年では平均値が 3.98（SD = 0.68）であり，t 値は 3.00，p < .01 という結果であった。効果量は 0.463 であり，これも中程度の効果が認められる。これらの結果を表 2 に示す。

さらに，プログラミング活用能力では，プログラミングの理解の項目においてのみ有意な差が見られた。7 年の平均値が 3.45,（SD = 0.79）に対し，9 年では平均値が 4.11（SD = 0.55）であり，t 値は 6.23，p < .001 であった。効果量は 0.959 であり，これは大きな効果を示している。この結果を表 3 に示す。

表 3　プログラミングの授業理解度における学年差の統計結果

	学年	平均値 (M)	標準偏差 (SD)	t値 (t)	有意水準 (p)	効果量 (Cohen's d)
1. 基本的な概念の理解	7年	3.80	0.73			
	9年	3.97	0.77	1.48	$p = .141$	0.229
2. プログラミングの理解	7年	3.45	0.79			
	9年	4.11	0.55	6.23	$p < .001$	0.959
3. プログラムの作成と実行	7年	3.70	0.83			
	9年	3.65	0.89	-0.42	$p < .676$	-0.065
4. 学習態度と授業の感想	7年	3.97	0.81			
	9年	4.08	0.75	0.90	$p = .368$	0.140

これらの結果から，情報リテラシーおよびプログラミング活用能力において，学年が進むにつれてスキルが向上する傾向が確認された。特に「情報の利用」と「プログラミングの理解」における効果量は，教育的介入により 7 年生から 9 年生にかけて実質的な影響を及ぼしていることを示している。今後の教育方針において，これらのスキルをさらに強化するためのアプローチが必要であることが示唆される。

Ⅴ．まとめと今後の課題

情報活用能力とプログラミング能力の重要性は，情報社会の進展と共にますます高まっている。これらの能力を養うことは，単に技術的なスキルとしてだけでなく，論理的思考力や問題解決能力，創造的思考力の育成に貢献することになる。また，データ分析のスキルは，ビジネスの意思決定に不可欠であり，プログラミングのスキルは，新しいソフトウェアやアプリケーションの開発に必須であるかもしれない。しかし，それらの能力を必要とする人の割合は大きくないことは予想できる。しかしながら，それらを学習することによって受ける恩恵は，今後の社会を生きる児童生徒にとって必要な要素である。したがって，それらを体系的かつ系統的なカリキュラムを通じてその能力を育成することは，今後ますます重要になると考えられる。さらに，AI を活用した個別化学習やプロジェクトベースの協働学習が主流となり，グローバルな視点を持つ人材の育成が求められる。これにより，日常生活の質の向上や職業選択の幅が広がり，さらにデジタルスキルを備えた人材がイノベーションを推進し，社会発展に貢献できるようになる。そのため，指導者は，個別支援と協働学習を促進する役割を担うために，多くの知識を持っていなくても情報技術の進展に対応しながら，児童生徒の学習に参加して一緒に考えながらでも継続することが必要である。

本稿では，これまでの系統的な実践からカリキュラムをまとめ，情報活用やプログラミングに関わる能力の向上を確認することができた。しかし，これまでの児童生徒の様子から，指導者のスキルによって最終的な理解度や進度に差がみられるため，この差が出ないような具体的な学習プランの確立が必要であると考えられる。また，プログラミングの授業を行う様子から，様々な思考力の向上が明らかに見られた。それらを立証することは，広範に使用できる良い実践の方法や内容，教材など発達段階に応じた提示ができるため，その立証が急務である。また，日本で行われている実践をまとめることができていないため，それらを調査し動向を探る必要がある。

謝辞

本研究は JSPS 科研費 24H02441（研究代表者：小澤雄生）の助成を受けたものです。

参考・引用文献

原田信一・安東茂樹・中峯浩・伊藤伸一・多田知正・赤井淳嗣・山﨑晃平・小澤雄生・森田光大・山田公成（2024）「教員養成課程学生の技術科「D 情報の技術」に関する意識について」，『京都教育大学教職キャリア高度化センター教育実践研究紀要』，6，31-38

竹野英敏 他　（2020）「技術・家庭：技術分野：テクノロジーに希望をのせて」，開隆堂

文部科学省　（2017）「中学校学習指導要領(平成 29 年告示)解説 技術・家庭科編」，開隆堂

文部科学省　（2019）「教育の情報化に関する手引き」，https://www.mext.go.jp/a_menu/shotou/ zyouhou/detail/mext_00724.html　（2024 年 8 月 9 日参照）

文部科学省（2020）「学習の基盤となる資質・能力としての情報活用能力の育成」，https://www.mext.go.jp/content/20201002-mxt_jogai01-100003163_1.pdf（2024 年 8 月 11 日参照）

文部科学省（2016）「プログラミング教育の手引き」，https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1403162.htm（2024 年 8 月 11 日参照）

文部科学省（2020）「第 3 章　プログラミング教育の推進」，https://www.mext.go.jp/content/20200608-mxt_jogai01-000003284_004.pdf（2024 年 8 月 11 日参照）

Papert, S. (1980). Mindstorms: Children, computers, and Powerful Ideas. Basic Books.

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. ACM Computing Surveys, 37(2), 83–137.

Kori, K., Pedaste, M., Leijen, Ä., & Tõnisson, E. (2016). The role of programming experience in ICT students' learning motivation and academic achievement. International Journal of Information and Education Technology, 6(5), 331-337.

Furber, S. (2012). Shut down or restart? The way forward for computing in UK schools. The Royal Society. Retrieved from The Royal Society：https://royalsociety.org/-/media/education/computing-in-schools/2012-01-12-summary.pdf　(2024 年 8 月 11 日参照)

Toikkanen, T., & Leinonen, T. (2017). Creative coding in the Finnish national curriculum. In International Journal of Child-Computer Interaction, 12, 47-54.

Finnish National Agency for Education. (n.d.). National core curriculum for primary and lower secondary basic education. https://www.oph.fi/en/education-system/national-core-curriculum-primary-and-lower-secondary-basic-education　(2024 年 8 月 11 日参照)

Code School Finland. (2019). Coding in the Finnish curriculum. https://www.codeschool.fi/2019/04/finnish-curriculum/　(2024 年 8 月 12 日参照)

Harju, V., & Niemi, H. (2017). Transversal skills in the Finnish curriculum. https://helda.helsinki.fi/server/api/core/bitstreams/8690a9c6-a265-4715-b1ed-7f67f0767f80/content(2024 年 8 月 12 日参照)

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: Programming for all. Communications of the ACM, 52(11), 60-67. https://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf　(2024 年 8 月 12 日参照)

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. Education and Information Technologies, 18(2), 351–380.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada.

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: Understanding teacher experiences and challenges. Computer Science Education, 26(4), 235-254.

Smith, K. A., Sheppard, S. D., Johnson, D. W., & Johnson, R. T. (2005). Pedagogies of engagement: Classroom-based practices. Journal of Engineering Education, 94(1), 87-101.

Eisenberg, M. B., & Berkowitz, R. E. (1990). Information problem-solving: The big six skills approach to library & information skills instruction. Ablex Publishing.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? Computers in Human Behavior, 41, 51–61.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. Educational Researcher, 42(1), 38–43.

付記

・本研究報告は，小澤・原田が全章協議を重ねた上で，次のように行った。

Ⅰ：小澤，Ⅱ：小澤，Ⅲ：小澤・原田，Ⅳ：小澤・原田，Ⅴ：小澤

資料1 情報活用能力とプログラミング教育のモデルカリキュラム

情報・プログラミング教育モデルカリキュラム

校種		初等部				中等部			高等部	
学年		1年生	2年生	3年生	4年生	5年生	6年生	7年生	8年生	9年生
フェーズ		コンピュータに慣れ親しむ		プログラミングに慣れ親しむ		プログラミングを身に付け活用する			情報の技術を深め活用する	
指導のポイント		コンピュータに慣れ親しませる。各教科の学びとコンピュータの良さを融合させ生活の一部と認識させる。		プログラミング的思考に慣れ親しませる。プログラミングを通して各教科と社会生活に繋がることを感じさせる。		プログラミングの基礎基本を身に付けさせる。プログラミングで学んだことを各教科の考え方や解決の仕方に活かせることを感じ取らせる。			プログラミング言語を活用することができる。また、情報の技術の見方・考え方を通して、様々な場面で課題の発見や解決ができることに気づきそれを駆使して新たな課題で有効活用できるようにする。	
情報活用・操作能力		→：操作を体験する　○：操作を習得する　■ 習得済								
基本操作	サインイン・シャットダウン	○								
	クラウドアプリ（「classroom」「ロイロノート）等に学校・家から繋ぐ事ができる	○								
	タップ操作・マウスや タッチパッド操作	○								
	指やペンで文字・絵を書く（ロイロノート・50音キーボード）	→	○							
	キーボードによる文字入力（ローマ字入力）			→	○					
	10分間での文字入力数	註）（）内はワープロ検定の合格目安である		200文字以上（4級）	300文字以上（3級）	400文字以上（準2級）	500文字以上（2級）	600文字以上（準1級）	700文字以上（1級）	800文字以上(初段)
記録・編集 活用①	「ロイロノート」等のカメラ機能を使って、写真、音声、動画を記録する	→	○							
	「ロイロノート」等を使って写真等の編集や記録を行う	→	→	→	○					
	目的に応じて「カメラ」、「フォト」、「ビデオ」等のアプリを活用する			→	→	→	○	→	→	○
	目的に応じて文書作成、表計算、プレゼン作成等のアプリを活用する			→	→	→	→	→	→	○
ファイル操作 活用②	「ロイロノート」の「ノート」や「カード」の取り出しや提出、保存を行う。	→	○							
	「ロイロノート」の「マイフォルダ」等の管理や、検索、送受信を行う		→	→	○					
	「classroom」等を使って、ファイルの管理、検索、送受信を行う				→	→	○			
	ファイルの圧縮、暗号化、バックアップをして管理する						→	→	○	
情報検索 活用③	「Yahoo!きっず」等のキッズ用検索サイト、「ロイロノート」から検索したり保存をする	→	○	→	○					
	ホームページの「お気に入り登録」をする 検索サイトでキーワード検索する	→	→	→	○					
	インターネット・情報セキュリティ・ネットワーク社会の問題に関する知識					→	→	→	→	○
	情報検索に関する知識とデータベース等の理解をする								→	○
その他学習 活用④	リモートのコミュニケーション活動を経験・参加・発表をする	→	○	→	○	→	→	→	→	○
	PCを通してクラウドの保存・管理、協働作業が使用できる			→	→	→	→	→	→	○
	ネットワークやOSの設定ができる								→	○
プログラミングに関する知識・能力		→：操作を体験する　○：操作を習得する　■ 習得済								
「育成すべき資質・能力」プログラミングに関する	知識・技能	コンピュータの基本的な操作ができる		問題を解決するために、順次、分岐、反復の3つの処理手順を知る。	問題解決に必要な3つの処理手順を、記号や図（流れ図）に表す良さに気づく。	条件分岐、データや変数を含んだプログラムの作成、改善ができる。	意図した処理を行うための最適なプログラムの作成・評価・改善ができる。	問題解決において、要素分解できるスキルを身に付ける。情報のデジタル化や処理の自動化などに関わる技術の仕組みを理解する。	情報通信ネットワークの構成と情報セキュリティに関わる仕組みを理解する。プログラムの製作、動作の確認及びデバッグ等ができる。	情報のシステム化の科学的な理解、問題の発見、解決のプログラム制作とモデル化ができる。計測・制御システムの仕組みの理解、双方向コンテンツのプログラミングが理解できる。
	思考力・判断力・表現力	コンピュータを使用して学習した内容の表現ができる		課題を振り返り、自分の考えを言葉で表現・説明することができる。	課題の解決について、他者の考えと比較して、さらによい解決方法がないかを考え、改善することができる。	課題を解決するために見通しをもって、学習した内容を有効的に使用しながら表現手段の組み合わせを考えることができる。	課題の解決を通して新たな課題を設定したり、解決するために協働的に取り組む態度をもつことができる。	課題の解決をする中で、新たな表現方法を見つけ、自ら有効的な解決策を論理的に見出すことができる。	情報の技術の見方・考え方に気づき、課題を設定し効果的な解決に向け計画を立案、評価、改善することができる。	情報の技術を評価し、適切に選択、管理・運用したり、新たな視点に基づいて課題の発見や解決することができる。
	学びに向かう力 人間性	コンピュータの便利さや楽しさを知ろうとする		プログラミングの考えが様々な教科や生活に結びついていることを見つけようとしている。	具体物（ロボット等）を利用してプログラミングで自分の考えた動きを意欲的に表現しようとする。	プログラミングで学習したことを意識して取り入れようとしている。また、他者と協力することができる。	プログラミングの考えを使用しながら、有効な解決方法を考え、物事を批判的に考察しようとすることができる。	主体的に知識・技能を高めようとする態度をもつことができる。条件を踏まえて計画を立て、思考しようとすることができる。	進んで情報の技術と関わり、主体的に理解し、技術を身に付けようとする態度をもつことができる。	自分が興味をもったことに対して、新たな問いを見つけようとすることができる。また、多様な視点から評価・改善・創造しようとする態度をもつことができる
プログラミングの 知識・技能等	基本的なプログラミングの概念の理解			→	○					
	情報モラルと情報セキュリティの理解			→	→	→	→	→	→	○
	基本的な概念の理解（条件分岐・変数・関数・配列）					→	→	○		
	for・while文やif文などの基本トピック						→	→	→	○
	フローチャートを使用した簡単な読み書きとアルゴリズムの理解				→	→	○（基礎）	→	→	○（応用）
	データ構造（リスト・辞書）・デバッグ等								→	○
プログラミング言語				scratch等（ビジュアル）	scratch等（ビジュアル）	（簡易言語）	（簡易言語）	python（スクリプト）	python（スクリプト）	python・JavaScript（スクリプト）