

論文の要旨

題目 Computational Thinking を育成する学習支援システムの開発に関する研究
(A Study of the Development of Learning Support System for the Promotion of Computational Thinking)

氏名 福井 昌則

高度情報化が加速する 21 世紀のデジタル時代において、Computational Thinking を身につけることが重要であると指摘されている。Computational Thinking とは、「問題をコンピュータで解決できる形で整理し、表現するための思考プロセス」のことであり、各国の Computer Science 教育などにおいて育成が求められている重要な概念である。我が国のプログラミング教育では、Computational Thinking 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「プログラミング的思考」の育成が掲げられており、その思考をプログラミングの授業のみならず、各教科内で育成することが求められている。

ここで、プログラミングを用いて Computational Thinking 育成を目指すこと、コーディングを必須としない形で、かつ各教科の内容で Computational Thinking を育成することの 2 つのアプローチについてそれぞれ検討していくことが求められる。

本研究では、「プログラミング教育において Computational Thinking を育成すること」、「各教科内でコーディングを用いずに Computational Thinking を育成すること」を実現するために、以下の 2 つの研究を実施した。

1 点目の「プログラミング教育において Computational Thinking を育成すること」についての事例は多く見られ、Scratch などのシステムを用いた事例が見られ、プログラミングのシステムも多数存在している。一方、Computational Thinking を身につけることで、プログラミングスキル以上のものを習得することが期待できる Wong and Cheung は、児童生徒がプログラミング活動を用いて Computational Thinking を身につけることができると述べており、Computational Thinking を育成する上でプログラミングを活用することは有用である。そして Computational Thinking を身につけることでプログラミングのコーディングスキル以上のスキルを身につけることが期待できる。プログラミングは、他の教科より自由度が高く、学習者が問題を自由に変更することや様々な実現・実装方法などが存在するため、学習者それぞれが自身で考えたことを実現するために有用な手法の一つである。よってコーディングスキル以上のスキルとして創造性、柔軟性に着目する。Rotem et al., Doleck et al. は、Computational Thinking と創造性には関連性があると指摘している。また、プログラミングで Computational Thinking の育成が可能であること、プログラミングで創造性を育成する事例など多く報告されている。しかし、我が国のプログラミング教育において育成が期待されている、プログラミングに対する様々な意識と創造性の関連性については明らかにされていない。よって、これらの詳細について把握することは重要であると考えられる。本研究では、プログラミングに対する様々な意識と創造的態度の関連性について把握する。

一方、2 点目の「各教科内でコーディングを用いずに Computational Thinking を育成すること」については、アンプラグドコンピューティングなど、コーディングを伴わない紙面上で行う実践が多い。また、各教科内容とは別の課題を準備した上で Computational Thinking の育成を目指すことが多い。各教科で Computational Thinking を育成するためには、既存の教科内で扱われている題材を活用する方法を確立することが必要である。そのために、どの教科においても適用可能な Computational Thinking 育成教材として、分類課題に着目する。「分類」は小学校においても教えられている。そして、論理的で網羅

的な分類が論理的思考の基礎として重要である。しかし、分類に関する学習が適切には成立していない可能性があり、分類についての適切な思考経験は十分に得られていないことが想定される。つまり、分類は教えられてはいるものの、それについての適切な思考経験は与えられておらず、その結果として、分類のための論理的思考が十分に習得されていないことが想定される。本研究では、分類課題を **Computational Thinking** の課題として捉え直すために、ベン図、Yes/No チャートを活用し、図形を網羅的に分類させることを目指す。分類を **Computational Thinking** の観点から見れば、物事を分類するために、**Computational Thinking** の「パターン抽出（一般化）」と「抽象化」が不可欠である。分類の実行には、**Computational Thinking** の「手続化」が必要となる。また、分類を段階的に行っていくことは、分類という課題に対して **Computational Thinking** の「分解」を行うことである。よって分類課題は、**Computational Thinking** の基本要素をすべて含む課題となっていると考えられ、各教科における分類課題を **Computational Thinking** 育成課題として捉えることで、各教科内で **Computational Thinking** を育成することが実現できる。その実現のために、ベン図、Yes/No チャートを用いたシステムを開発し、学習者それぞれの **Computational Thinking** 育成を目指す。

ここで本研究では、21 世紀のデジタル時代に必要なスキルである **Computational Thinking** をプログラミング教育および各教科内で育成することを目指す。この実現に向けた第一歩として、「プログラミングと創造性の関連性の把握」、「各教科内での **Computational Thinking** 育成」という 2 点について研究を行った。

本研究では、以下の知見が得られた。1 点目の「プログラミング教育において **Computational Thinking** を育成すること」に関して、プログラミング教育において創造性を育成するために、レディネスとしての創造的態度に着目し、創造的態度とプログラミングに対する様々な意識との関連性についての調査研究を行い、男子の方が女子よりも創造的態度とプログラミングに対する様々な意識の各項目が高いこと、性別によって関連性がある創造的態度とプログラミングに対する様々な意識の項目に相違があることを示した。そして、性別によって創造的態度を高めるプログラミング教育の方法について差異がある可能性が示唆された。以上のことから、よって、創造的態度を高めるプログラミングの実践方法に性別ごとに適した方法が存在する可能性があることが示唆された。

2 点目の「各教科内でコーディングを用いずに **Computational Thinking** を育成すること」に関して、**Computational Thinking** を育成するために、各教科で用いられている題材の活用を前提とした上で、図形の分類課題を **Computational Thinking** 育成課題として設定した。その育成のために、ベン図と Yes/No チャートを導入し、自動判定機能を実装したベン図、Yes/No チャート学習システムを開発した。その上で、Yes/No チャート学習システムの学習効果を、経験学習を取り入れた実践モデルに基づいて測定した。その結果、Yes/No チャート学習システムが **Computational Thinking** 育成に有効であることを示した。また、システム経験前の **Computational Thinking** テストの得点によって学習効果が異なることも合わせて示した。そして、Yes/No チャート学習システムを用いた **Computational Thinking** 育成にそれまでのプログラミング経験が影響しないことも合わせて示した。

本論文の構成を示す。第 1 章では、本研究の位置付けと意義について述べる。第 2 章では、先行研究及び各概念について整理する。第 3 章では、創造的態度とプログラミングに対する意識との関連性について述べる。第 4 章では、**Computational Thinking** を育成するための題材を開発し、**Computational Thinking** を育成するベン図、Yes/No チャート学習システムについて述べる。そして、Yes/No チャート学習システムの学習効果、プログラミング経験が Yes/No チャート学習システムに与える影響について述べる。第 5 章では、これらの研究を総括する。