

プログラミングにおける時間の感覚を発想させる教授法の効果

中尾 尊洋・土井 康作

The Effect of Teaching Method to Think About Time Feeling in Programming

NAKAO Takahiro, DOI Kosaku

地域学論集（鳥取大学地域学部紀要） 第14巻 第3号 抜刷

REGIONAL STUDIES (TOTTORI UNIVERSITY JOURNAL OF THE FACULTY OF REGIONAL SCIENCES) Vol.14 / No.3

平成30年3月26日発行 March 26, 2018

プログラミングにおける時間の感覚を発想させる教授法の効果

中尾尊洋*・土井康作**

The Effect of Teaching Method to Think About Time Feeling in Programming

NAKAO Takahiro*, DOI Kosaku**

キーワード：プログラミング学習，試行錯誤，身体動作支援，深い理解

Key Words: programing learning, trial and error, programming support by body movement, deep understanding

I. はじめに

1. 中学校におけるプログラミング学習と研究目的

2012年度(平成24年度)の学習指導要領実施によって、中学校技術・家庭科技術分野（以下中学校技術科）では、プログラミング学習が必修化された¹⁾。この中で、プログラムによる計測・制御について指導する事項として、「情報処理の手順を考え、簡単なプログラムが作成できること」とされている。このことを踏まえて、制御教材を用いてプログラミング学習を実践する報告が多く見られる²⁾。例えば、自動ドアを教材化したもの、LED制御させる製品、ライントレースカーなど、実際の場面を想定した教材を用いた実践である。これらの教材をみると、学習者に製品を制御する手順を工夫することが求められており、フローチャートを作成させたり、プログラミングエディタを使わせたりするなど、学習者自らプログラミングできるように環境が作り出され、支援されていた。その結果、自らが思考することによって、プログラムの作成ができる力の育成が可能になったと報告されている。

このような実践を詳細にみると、学習者が条件分岐やループ等の基本的な処理を用いて手順を作り出し、プログラムを作成している。このことは、基本的な処理に関して、その機能を学習者が理解していることが前提である。しかし、基本的な処理である条件分岐やループ等の命令の意味は、自ら思考して導き出し、理解されたものではない。授業の中で授業者が知識を伝達して教え込んでいるものとみられる。つまり、命令の意味については、単に課題に対して必要となることから、記憶させているだけにとどまるのではないかと考えた。

認知心理学の知見³⁾によると、深い理解を伴う学習には、単なるスキルや事項の記憶以上のものが必要であることが、熟達化研究の成果によって示されたとしている。そして、学習方法として、授業における事象とそれを明確に定義している原理の両方を理解させることを提案している。そのために、既に関連知識を持っているということを学習者に気づかせることを通して、自立した問題解決者として学習者を育成する視点を持つことが重要であるとしている。

このような知見から、基本的な処理の学習場面では、学習者が文脈に沿った必要性から、既有知識をもとに命令や基本的な処理を発想し、プログラムを作成する授業構成が望ましいと考えられる。しかし、学習者が基本的な処理を複合的に用いてプログラムを構成することは、日常生活経験で

*鳥取大学附属中学校

**鳥取大学地域学部地域教育学科

はあまり考えない思考であるため、困難が予想される。本稿では、その中でも、時間の感覚がプログラムと日常生活では大きく異なることに着目した。すなわち、プログラムの動作速度は、CPUを通じた電氣的な速度に対応しており、この速度が、人間が通常感じる時間の感覚とは違って非常に高速であるため、プログラムの作成に関する思考を阻害していると考えられるのである。従って、学習者自らがプログラムを構成できるようになるには、時間の感覚を捉える支援が必要となる。

このようなことから、本稿では、プログラミング学習において、基本的な処理の命令を学習者が文脈に沿った必要性から導き出し、プログラムを作成する授業を実践した。その際、学習者の発想を阻害すると考えられる時間の感覚について、それを捉えさせる支援を導入し、その効果を検証した。

2. プログラミング学習の先行研究と教授方法

一般的に行われていると考えられるプログラミング学習の方法としては、岡本ら⁴⁾が述べているように、サンプルプログラムを使って、プログラムの記述とその実行結果から各処理概念を学んでいく経験学習の方略が用いられる。授業者が提示したサンプルプログラムを模倣して入力、実行し、その結果を確認する方法である。この方法は、テキストベースのコンピュータ言語を用いてプログラム作成する技術を育成する観点からは、有効な方法であると考えられる⁵⁾。テキスト入力によるプログラミングは、初学者にとって、綴り間違いや空白の有無等によるエラーを起しやすいため、論理的なアルゴリズムの思考以前に、いわゆる言語習得が必要だからである。ところが、近年では、テキスト入力をほとんど必要としないビジュアルプログラミング言語が開発され、これまで必須と考えられていたテキストプログラミング言語の習得なしに、プログラムを作成、実行できるようになってきた。これによって、アルゴリズムの思考に絞ったプログラミング学習が可能となっている。このようなプログラミング環境の変化は、基本的な処理を記憶させるのではなく、自ら思考して導き出すことに重点を置いた学習を可能にする。基本的な処理について、生活経験や動作の内観をもとに、気づかせることができるのではないかということである。

中学校技術科におけるプログラミング学習の先行研究としては、題材開発に関するものや設計段階における支援に関するものが多い。荻嶺ら⁶⁾は、題材タイプの違いによる生徒の反応について研究し、市川⁷⁾により、それをもとにした実践が報告されている。また、藤田ら⁸⁾は、フローチャートの作成を前提としないプログラムの設計学習に、より高い効果を示すことを明らかにしており、増田ら⁹⁾はそれをもとにした、プログラムの設計場面における状態遷移図を用いたプログラム動作の可視化支援の実践を報告している。これらの研究は、プログラミング学習の効果的な支援を実現してはいるが、学習者がプログラムの命令をどのように理解してアルゴリズムを構成しているのかを明らかにしていない。

このように、先行研究を概観すると、プログラムの命令の意味や基本的な処理である条件分岐やループ等の基本的な処理の学習に関して、授業者が知識を伝達することで学習者に知識を与え、その上で学習者が様々なプログラムを作成していると考えられる。このような方法は、どのような命令のつながりが、どのような動作になるのかというプログラムの知識、技能の獲得を効率的にするだろう。しかし、一方で、知識を伝達されることで命令や基本的な処理の持つ意味を固定的に捉えてしまうことが懸念される。

中学校の授業という学習空間では、問題の答えを知った時点で、学習者が満足してしまうことがよく見られる。このことに関して、稲垣ら¹⁰⁾は、教師と学習者の間には、評価者と被評価者という

関係性が築かれており、評価者である教師により評価を求める、つまり教師の持つ解答を提示することに、学習者の関心が注がれてしまうことが一因と述べている。プログラミング学習の場面において、サンプルプログラムを提示して写させるという手段は、まさに教師のもつ解答を提示する行為といえる。学習者は、プログラムの内容を理解しようとするよりも、動作させて評価を得ることに意識を奪われてしまうのではないかと危惧される。実際、以前筆者が実践していたサンプルプログラムを写させる授業でも、プログラムを入力し、動作を確認した時点で達成感を持ち、学習を終了してしまう学習者が散見されていたことを記憶している。

また、サンプルプログラムを写させることによって、学習者がプログラムについて理解しているのかということが見えにくくなるという問題もある。プログラムについて理解を伴わなくても、写すことは可能だからである。写させるのではなく、工夫すべき点に気づき、学習者自らの思考によってプログラムを完成に導くことができれば、それは、学習者がプログラムについて理解していると見なすことができる。

これらのことから、中学校におけるプログラミング教育において重要な、プログラムの思考等の育成を目的としたプログラミング学習の場面で、サンプルプログラムを写させる学習方法では、その効果が不十分だと思われる。正しいプログラムを完成させることではなく、試行錯誤を伴わせ、不完全にしる、自分で動作手順を考えてプログラムを作成し、動作させることが重要ではなからうか。こうしたプログラミング学習における命令や基本的な処理の意味理解、そして、アルゴリズムの構成について、知識や概念を獲得する手段についての議論は、まだ十分ではないと考える。

3. プログラミングにおける時間の感覚について

コンピュータプログラムの動作速度は、CPUを通じた電氣的な速度に対応している。この速度は、人間が通常感じる時間の感覚とは違って非常に高速である。複数の動作であっても一瞬で実行するため、同時に実行されているように感じる¹¹⁾。こうした一瞬で実行されるということを利用して、プログラミングの際には、様々な処理が同時に行われるような動作をつくり出す。このプログラム動作において、人間には捉えられない瞬間に処理される感覚を、本稿では、プログラミングにおける時間の感覚としている。

プログラミングにおける時間の感覚を得ることで、通常の生活の中で並列的に実行される行為等の同時という認識が、瞬間的な時間によって分割、実行されたものであるという認識に変わる。その認識をプログラミングの際に働かせることで、同時処理をさせる場面において、順次的に命令を構成するというプログラム方法の発想へと進ませる。しかし、プログラミング初心者である中学生段階の学習者にとって、プログラミングにおける時間の感覚は日常的に感じる事ができないため、その発想は容易ではない。したがって、この感覚を発想させる支援なしに、学習者がプログラムを作成することは困難と考える。

4. 身体動作によるプログラム理解の支援

学習者にプログラミングにおける時間の感覚を発想させる支援として、視認できないプログラムの動作と対応させた身体動作を行わせることとした。視認できない動作と、視認できる動作の対比からプログラムにおける時間の感覚を発想させることをねらったものである。具体的には、プログラムの動作に類似した身体の動作を行い、その動作を細かい命令群の集合と捉えさせ、プログラムの命令や手順の組み合わせと対比させて、プログラムの構造を発想させるようにした。プログラム

の動作は、様々な命令の組み合わせであり、身体の動作を細かく分解したものと同様の構造を持つと考えたからである。身体動作という、自分で意識できる動作とプログラムの構造とが類似した構造であると捉えることで、プログラムを構成することに関して、難しさを感じにくくするねらいがある。

II. 研究の方法

1. 授業実践の概要

1.1. 教材

授業であつかう教材として、ドットマトリクス LED (8×8 ドット) を Arduino 互換基板で操作するものを用いた(図 1)。これは、鳥取市中学校技術・家庭科研究会が開発したものである。PC と USB 接続することで、プログラムの受信と電源供給が行われる。これを操作するプログラムエディタは、Arduino 社が提供している sketch 等を用いることができる。プログラムをアップロードした後は、電池駆動により USB からの電源供給に頼ることなく、単独動作させることができる。実践授業における学習者の操作場面では、PC でプログラムを作成、アップロードすることにより、ドットマトリクス LED の点灯、消灯の操作をさせた。

このドットマトリクス LED への出力は、一度に 8 灯分しかできない。ドットマトリクス LED64 灯分を出力しようとする、出力系が不足するため、高速に順次的に点灯させる、ダイナミック点灯をさせる必要がある。プログラミングにおける時間の感覚が得られていないと使いこなせない教材であることから、この教材を選定した。

学習者が操作するために用いたプログラムエディタは、sketch 上で動作する ardublock というプラグインを利用した(図 2)。これは、テキストでコーディングするスタイルではなく、命令に相当するブロックをメニューから取り出し、つなぎ合わせることで、プログラミングを可能にする。プログラミング教育に期待されていることは、コーディングする知識や技能を獲得させることではなく、プログラミング的思考などを育むことにある。したがって、テキストでコーディングすることが重要ではなく、目的とする動作を実現するために、命令をどのようにつなげるかという点が重要であるといえる。ブロックをつなげるスタイルでプログラミングが可能な ardublock は、テキストの入力ミスを起こさず、命令のつながりに焦点化してプログラミングができる。この点を捉えて、中学校での授業実践に適していると判断し、選定した。

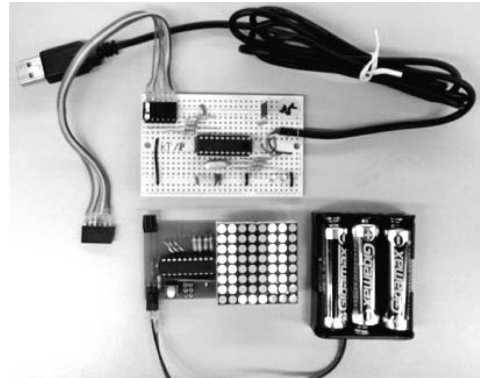


図 1 授業で用いた教材

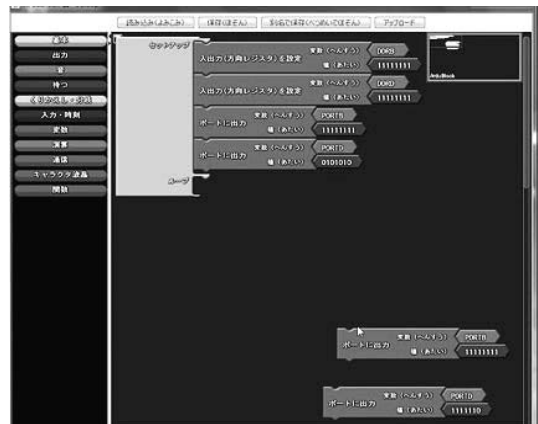


図 2 授業で用いたプログラムエディタ

1.2. 授業実践のねらい

授業実践では、学習者に試行錯誤を促し、サンプルプログラムの提示を極力行わずにプログラムを完成させることをねらった。提示する問題は、ドットマトリクス LED の縦一列表示と、横一列表示を交互に点灯させ、自動車のワイパーのような動きをアニメーション表示することとした (図 3)。この問題は、学習者が、一時的にプログラムを止める命令の必要性に気づくことができなければ、指定するアニメーション動作が完成しない。単純に縦一列と横一列の点灯、消灯を順次的にプログラミングしただけでは、視認できる表示は L 字になってしまう。プログラムの実行速度は非常に速いので、縦一列表示と横一列表示が同時に行われているように見えるダイナミック点灯になってしまうことが理由である (図 4)。したがって、プログラムにおける時間の感覚が得られていなければ、自らの力でプログラムを作成できない。このような教材の特徴を生かした上で、学習者が自分で作成したプログラムを実行し、その動作のデバッグを繰り返す中で、プログラムにおける時間の感覚に気づかせようと考えた。そのため、学習者のプログラミングスキルに応じて、3 段階の授業構成とした (図 5)。ある程度の知識をもっている学習者であれば、既にプログラムにおける時間の感覚を踏まえて正しいプログラムを作成できたり、プログラミングとデバッグを繰り返してプログラムにおける時間の感覚に気づけたりすると考えられる。そこで、1 段階は、縦、横のアニメーションを表示させるという問題のみ提示し、既有知識のみで試行錯誤させる時間を確保した。2 段階は、正しいプログラムを作成できない学習者に対して、プログラムにおける時間の感覚の獲得に向けた支援をし、その上で試行錯誤させる時間を確保した。3 段階は、1,2 それぞれの段階を経てもプログラムを作成できない学習者に対して、授業者が正しいプログラムを提示し、その動作を解説した上で、プログラムの動作を確認させるようにした。

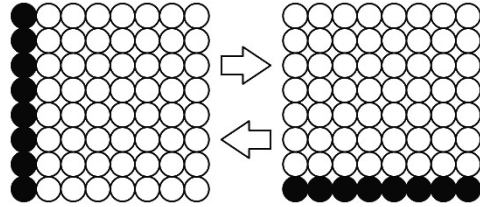


図 3 ドットマトリクスのアニメーション表示

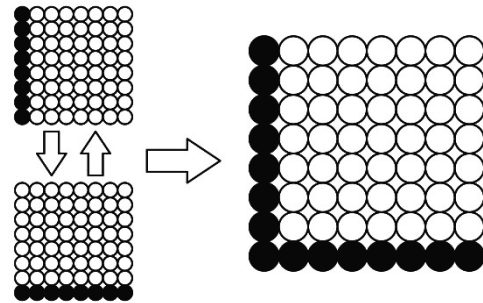


図 4 縦横表示のみでの見え方

その動作のデバッグを繰り返す中で、プログラムにおける時間の感覚に気づかせようと考えた。そのため、学習者のプログラミングスキルに応じて、3 段階の授業構成とした (図 5)。ある程度の知識をもっている学習者であれば、既にプログラムにおける時間の感覚を踏まえて正しいプログラムを作成できたり、プログラミングとデバッグを繰り返してプログラムにおける時間の感覚に気づけたりすると考えられる。そこで、1 段階は、縦、横のアニメーションを表示させるという問題のみ提示し、既有知識のみで試行錯誤させる時間を確保した。2 段階は、正しいプログラムを作成できない学習者に対して、プログラムにおける時間の感覚の獲得に向けた支援をし、その上で試行錯誤させる時間を確保した。3 段階は、1,2 それぞれの段階を経てもプログラムを作成できない学習者に対して、授業者が正しいプログラムを提示し、その動作を解説した上で、プログラムの動作を確認させるようにした。

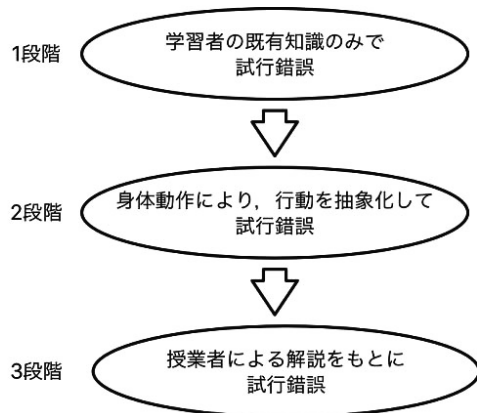


図 5 設定した授業段階

1.3. 既習事項

本稿における授業実践を実施するまでに、既習事項をまとめたものを表1に示す。コンピュータのハードウェアについての学習や、デジタルとアナログの違いについての学習、ドットマトリクスLED教材の扱い方や arduino の扱い方について学習している。

arduino を用いた学習に関しては、ドットマトリクスの1点を点灯、消灯させるプログラム、縦、または横一列に点灯、消灯させるプログラムについて扱った。

表1 本研究における授業実践までの既習事項

時間	授業	学習内容
1	情報伝達の工夫と情報処理の仕組み	デジタルとアナログの違い・情報伝達の仕組み
1	情報処理とコンピュータの構成	情報処理の4要素・コンピュータデバイスの構成
1	コンピュータを用いた情報処理	LEDの点灯方法・プログラムエディタの使い方
1	文字と画像の特徴と利用方法	8×8ドットマトリクスLEDの列表示方法

1.4. 授業の内容

授業実践の流れを図6に示した。授業でプログラミングさせる動作は、縦、横表示をアニメーションさせるといものである。まず始めに、学習者に対して、授業で目指す動作内容を説明した。最初の活動として授業者が何も支援をせずにプログラムを考えさせる時間を10分程度確保した。プログラムの作成経験がある学習者に、自分の力でプログラムを完成させ、また、経験がない学習者には、自分の考えを深めさせるためである。

その後、身体動作によるプログラム理解の支援を行った。テーマがワイパー動作であることから、自分の腕の肘から先を縦、横に動かす動作をさせた(図7)。その際、プログラミング経験がない学習者が作成するプログラムは、動作速度を考慮できないと考えられることから、単純に縦表示と横表示を順次的に並べるものになると予想される。必要な処理としては、プログラムを一時的に止めるウェイト処理である。このことに気づかせるために、作成したプログラム通りに腕を動かすよう指示し、実際の動作との違いを対比させるようにした。こうした支援を通して再度、学習者にプログラムを考えさせる時間を10分程度確保した。

活動の最後には、授業者がプログラムを提示し、解説を行った上でプログラム作成させる時間として5分程度確保した。これまでの時間では作成できなかった学習者に対して、プログラムの構造に気づかせること、及び、すでに作成できている学習者に対して、理解を定着させることをねらった。

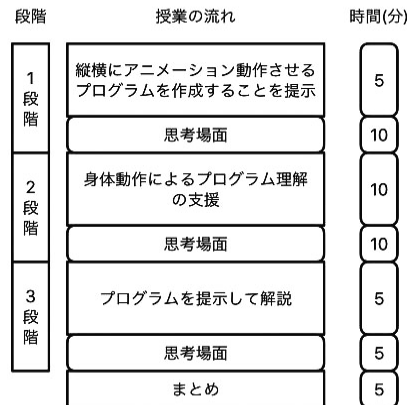


図6 授業実践の流れ

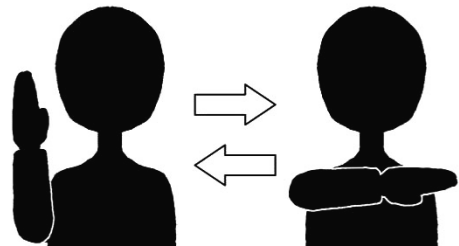


図7 LEDの動きを再現させた腕の動作

2. 調査期間及び調査対象

2015年11月中旬に実施された1時間の授業を研究対象とした。後期の授業が開始されて約1ヶ月半後の時期にあたる。対象被調査者は鳥取市立K中学校の3年生52名である。K中学校の3年生は、5クラスの編成である。2クラスは前期に技術の授業を終了している。後期に技術の授業を実施した残り3クラスのうち、調査者が立ち会うことのできた2クラス、かつ、データが得られた被調査者を対象としている。データが得られなかった被調査者が存在してしまった理由は、PCの不具合等により、画面記録が保存されなかったことによる。

3. 調査方法

授業実践時の被調査者の操作を、画面キャプチャアプリケーションにより、1秒につき、約1枚記録、保存した。授業の中で、プログラムが完成したとみなされる授業の段階をもとに、上位群、中位群、下位群に分け、それぞれの操作数、及び、マイコンへのプログラムアップロード数をカウントし、その増減などから、授業の段階における各群のプログラム作成活動を分析した。また、各群より無作為に3名抽出し、操作画面を観察することで、各群における思考のプロセスを分析した。

なお、プログラムの完成時点の定義として、プログラムにおける時間の感覚を考慮してLEDのアニメーション表示プログラムが作成できた時点とした。具体的には、縦一列表示と、横一列表示の間に、プログラムを一時的に止める命令を組み込んでマイコンへアップロードし、動作を確認した時点とした。

III. 結果と考察

1. 各群の人数分布及び操作数、アップロード数の変化について

1.1. 各群の人数

上位群、中位群、下位群それぞれの人数及び割合を表2に示した。

何も支援を受けずにプログラムを作成できた上位群と、身体動作によるプログラム理解の支援を行ってプログラムを作成できた中位群で、あわせると調査対象となる被調査者全体の92.3%を占める。中学生というプログラムに関する既有知識が少ない段階でも、本稿の実践において、多くの被調査者が自らの力でプログラムを作成することができたといえる。内訳を見ると、中位群が69.2%と最も高く、次いで上位群の23.1%、下位群の7.7%となった。

上位群の23.1%は、テーマのみを提示されてプログラムを作成できていたことから、命令や手順の組み方、プログラムにおける時間の感覚を得ている被調査者であるといえる。上位群は、プログラムに関する既有知識を有する、または、生活経験等をプログラムに関する知識へと自ら転移させることができた被調査者と考えられる。

中位群の69.2%は、ある程度思考した後に、身体動作によるプログラム理解の支援をうけて、プログラムを作成できた。自らの思考のみではプログラム作成には至らなかったが、支援により、プ

表2 各群の人数及び割合

(N=52)

	上位群	中位群	下位群
人数(人)	12	36	4
割合(%)	23.1	69.2	7.7

プログラムの命令や手順の組み方、プログラムにおける時間の感覚に気づき、プログラムの完成に導くことができたといえる。中位群の割合が最も高いことから、授業前からプログラムに関する既知知識をもっていた被調査者は多くないと見ることができる。その上で、支援によってプログラムを作成できた被調査者の割合が高いことから、身体動作によるプログラム理解の支援は、被調査者が自らプログラムを作成することに高い効果を示したと考えられる。

下位群の7.7%は、身体動作によるプログラム理解の支援を通して、プログラムを作成することができず、その後のプログラムの提示と解説によって完成させた。この被調査者には、身体動作によるプログラム理解の支援が効果を示さず、自らの力でプログラムを作成することができなかった。

1.2. 各群の操作数の変化

上位群(N=12)、中位群(N=36)、下位群(N=4)における、身体動作によるプログラム支援導入前後の操作数の時間経過による変化を表3に示し、グラフにしたものを図8に示した。

身体動作によるプログラム理解の支援を導入する前後の、自ら試行錯誤していると考えられる時間帯について、1分あたりの操作数をカウントし、各群の被調査者全員の平均値を算出した。

グラフの縦軸は操作数の平均であり、横軸が時間経過である。身体動作によるプログラム理解の支援を導入する前から、上位群の操作数が伸びていることがわかる。次いで、中位群の操作数が多

表3 身体動作によるプログラム支援導入前後の1分あたりの操作数平均

群	導入前経過時間(分)											導入後経過時間(分)							
	1	2	3	4	5	6	7	8	9	10	11	1	2	3	4	5	6	7	8
上位群	2.75	5.00	3.36	5.45	4.45	2.82	3.33	4.92	3.67	5.17	1.08	1.80	3.82	5.00	4.08	3.58	3.92	2.67	2.25
中位群	2.91	3.14	3.82	3.68	4.49	3.37	3.51	2.66	3.77	3.08	0.31	1.95	5.80	5.83	3.44	3.92	3.81	3.69	2.58
下位群	3.00	2.50	2.25	2.00	2.50	4.25	2.50	4.00	0.75	1.50	0.25	2.50	3.25	3.50	3.75	3.25	2.25	4.25	3.50

単位:回

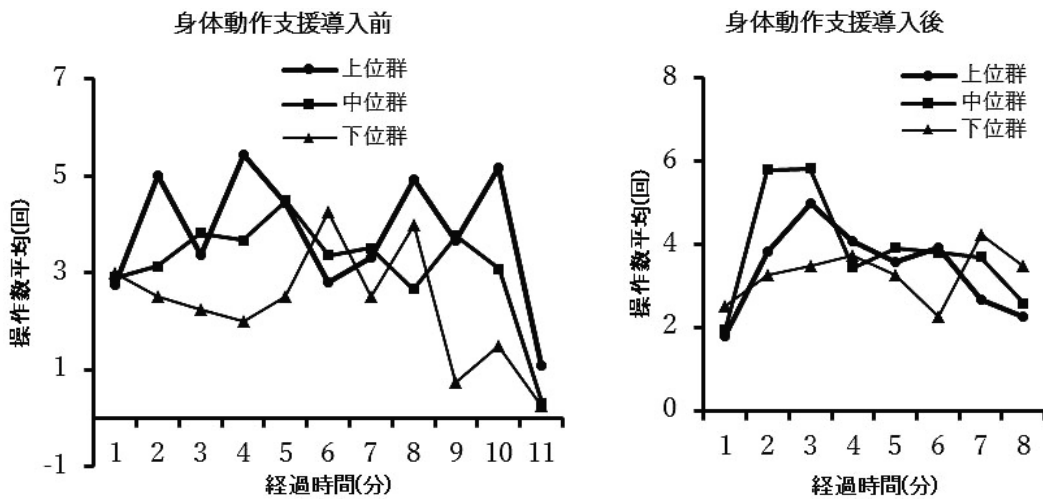


図8 身体動作によるプログラム支援導入前後の1分あたりの操作数平均

く、下位群の操作数が最も少ない。上位群は、何も支援していない状態にもかかわらず、様々な操作をしていることがわかった。上位群は、テーマを提示された時点で、ある程度の命令の手順について発想が浮かんだことが、操作数が多い要因と推察する。必要なプログラムの手順に関して、既知知識を転移させることが、ある程度できているものとみられる。中位群は、プログラムを完成させられていない中で、様々な操作をしているといえる。漠然とした方向性をもって操作しているか、または、とにかくいろいろな方法を試しているものと見られる。下位群の操作数が少ないのは、プログラムの手順に関するイメージが湧かず、どのようにプログラム作成をすればよいのか、わからない中で操作しているのではないかと推察する。

身体動作によるプログラム理解の支援を導入直後は、中位群の操作数が伸びている。次いで、上位群の操作数が多く、下位群の操作数が最も少ない。中位群は、身体動作による支援により、必要なプログラムの手順に関して、身体動作の手順が転移されて、発想が進んだものと推察する。上位群は、プログラム完成後も操作を継続している。下位群は、依然としてプログラムの手順に関するイメージが湧いていないのではないかと推察されるが、後半に操作数が伸びていることから、いろいろと操作していく中で、イメージが捉えられる被調査者も出てきている可能性がある。

各群のプログラム理解の支援を導入した前後で、操作数の増減について確認するため、全被調査者の支援前後の操作数の平均値に対して *t* 検定を行い、表 4 に示した。

表 4 身体動作によるプログラム支援導入前後の比較

(N=52)

群	導入前平均(回)	導入後平均(回)	<i>t</i> 値	
上位群	3.89	3.44	<i>t</i> (11)=0.63	n.s.
中位群	3.05	3.99	<i>t</i> (35)=3.88	*
下位群	2.21	3.25	<i>t</i> (3)=1.27	n.s.

p*<.05, *p*<.10, n.s.: 非有意

中位群において、プログラム理解の支援を導入した前に比べて、導入した後に有意に操作数が増加している。中位群では、身体動作による支援が効果を示し、導入後の操作数増加に影響を与えたものと考えられる。このことから、身体動作による支援の効果が示されたといえる。上位群、及び下位群には、有意差が認められなかった。

1.3. 各群のアップロード数の変化

上位群(N=12)、中位群(N=36)、下位群(N=4)における、身体動作によるプログラム支援導入前後のアップロード数の時間経過による変化を表 5 に示し、グラフにしたものを図 9 に示した。

身体動作によるプログラム理解の支援を導入する前後の自ら試行錯誤していると考えられる時間帯について、1分あたりのアップロード数をカウントし、各群の被調査者全員の平均値を算出した。

表 5 身体動作によるプログラム支援導入前後の1分あたりのアップロード数平均

経過時間(分)	導入前											導入後							
	1	2	3	4	5	6	7	8	9	10	11	1	2	3	4	5	6	7	8
上位群	0.25	0.71	0.45	1.18	1.55	0.82	0.58	1.08	1.25	0.83	0.50	1.75	1.45	1.00	0.92	1.08	0.75	0.83	0.50
中位群	0.18	0.59	0.86	0.74	0.89	1.03	0.94	1.00	1.31	1.19	0.19	0.41	1.00	1.17	0.89	1.06	1.03	0.92	0.97
下位群	0.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	0.50	0.75	0.00	0.00	0.50	0.50	1.00	0.75	1.00	0.50	0.50

単位: 回

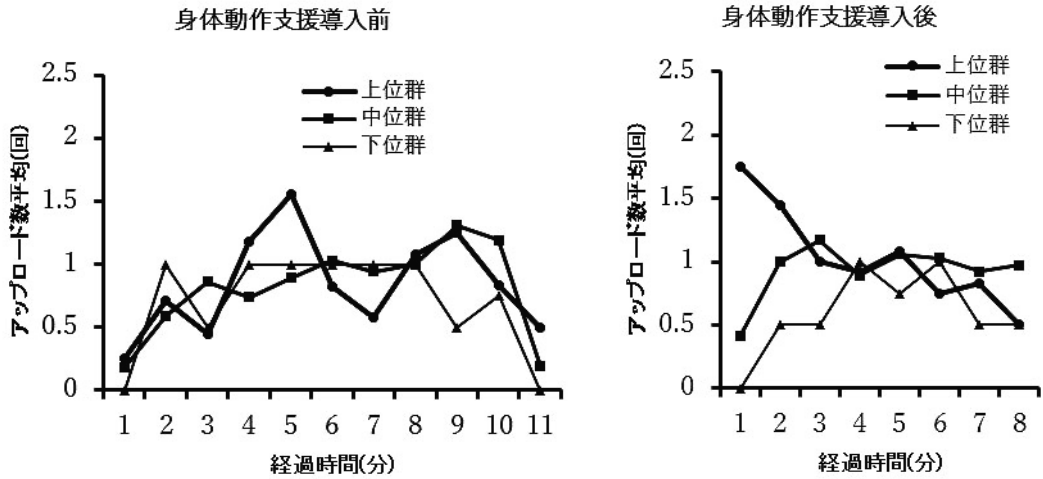


図9 身体動作によるプログラム支援導入前後の1分あたりの操作数平均

グラフの縦軸はアップロード数の平均であり、横軸が時間経過である。グラフをみると、身体動作によるプログラム理解の支援を導入する前のアップロード数に関して、各群の差はあまり見られない。アップロード数は動作を確認している回数でもある。上位群では、他の群に比べて操作数が多かったが、アップロード数は他の群と比較して変わらないことがわかった。このことから、上位群では、支援をする以前から多くの操作をしてから動作確認をしているといえる。上位群は、プログラムの構成にある程度の推測をして組み立てているのではないかと推察する。身体動作によるプログラム理解の支援を導入した後も、アップロード数に関しては、各群の差は大きくない。身体動作による支援の導入直後の操作場面において、上位群に多くのアップロードがみられているが、より理解が進んで動作を再確認したのではないかと推察する。

各群のプログラム理解の支援を導入した前後で、アップロード数の増減について確認するため、全被調査者のアップロード数の平均値に対して t 検定を行い、表6に示した。

表6 身体動作によるプログラム支援導入前後の比較

(N=52)

群	導入前平均(回)	導入後平均(回)	t 値	
上位群	0.90	0.94	$t(11)=0.35$	n.s.
中位群	0.83	0.95	$t(35)=1.24$	n.s.
下位群	0.70	0.64	$t(3)=-0.21$	n.s.

* $p<.05$, ** $p<.10$, n.s.: 非有意

表6より、各群において、有意差は認められない。このことより、身体動作によるプログラム理解の支援の導入前後に、アップロードによる動作確認回数の増減は見られなかったといえる。

中位群では、操作数は増加していたが、アップロード数の増加は見られなかった。このことから、中位群は、身体動作によるプログラム理解の支援によって、ある程度のプログラムの推測が行われ、まとまった操作を行ってから、アップロードして確認しているのではないかと推察される。

2. 各群の操作内容について

2.1. 抽出被調査者について

各群の操作内容について検証するため、記録したキャプチャ画面から、どのような操作が行われているのかを確認した。確認する被調査者を各群から無作為に3名抽出し、操作画面を観察した。観察のポイントとして、以下の点に着目した。

- ①縦表示、及び横表示の方法が理解されているか。
- ②縦表示と横表示の命令を順番に並べる、順次性を意識したプログラミングがどの段階でできているか。
- ③プログラム動作を一時的に止める「待つ」という命令がどの段階で出現し、どの段階で正しい位置に組み込んでいるか。
- ④テーマとして提示した、縦表示と横表示のアニメーションプログラムを発展させるようなプログラミングをしようとしているか。

それぞれの観察ポイントは、プログラムに関する命令やその手順について、理解できていることが確認できるものである。①は、前時までの学習内容であり、縦一列、及び横一列表示させるプログラム方法が理解されていて、プログラミングできる段階と考えられる。②のプログラムが作成できている段階は、プログラムの順次性について理解できている段階と考えられる。③に関しては、「待つ」が必要と認識できるということは、プログラムにおける時間の感覚に気づき、一時的にプログラム動作を止める必要性が理解できている段階と考えられる。また、それを正しい位置に組み込めるということは、順次性についての理解を踏まえた上で、プログラムを一時的に停止させる位置が理解できている段階と考えられる。④に関しては、プログラムの順次性や時間の感覚についてわかった上で、自分なりに活用してプログラムを発展させようとしている段階と考えられる。具体的には、アニメーション速度を変更したり、点灯位置を変更したりすることが考えられる。

2.2. 上位群の操作内容

上位群の抽出被調査者に見られた、プログラミングの特徴を表7に示した。

上位群被調査者に共通するプログラミングの特徴を見ると、身体動作によるプログラム理解の支援を導入する前の段階で、プログラムの順次性や「待つ」命令の必要性に気づいているといえる。プログラム作成の開始直後から、縦表示と横表示を順番に作成する様子が見られ、順次性を意識してプログラミングできている様子が確認できた。さらに、「待つ」命令を加えることもできた(図10)。プログラムにおける時間の感覚に即時的に気づいていると考えられる。上位群被調査者は、このようなプログラムの作成プロセスから、自らプログラムにおける時間の感覚に気づき、プログラムを完成させているとみられる。即時的に時間の感覚に気づいていることから、プログラム経験があつてプログラムの構成を想起したか、時間の感覚に関する既有知識を瞬間的に転移させてプログラムの構成に結びつけたものと推察する。

身体動作によるプログラム理解の支援を導入した後は、アニメーション表示に関して、「待つ」の時間を変更して確認したり、「待つ」の配置を変更して確認したりする活動が見られた(図11)。目的としたプログラムが既に完成しており、「待つ」の効果を再確認する意図があるものと推察する。また、アニメーションを増やすなどの、プログラムの改変操作が上位群被調査者には見られており、自分なりのプログラムを作成しようとする意欲が高まり、プログラムを発展させていることも推察される。

表7 上位群抽出被調査者のプログラミングの特徴

抽出被調査者	身体動作導入前	身体動作導入後
抽出被調査者 1	<ul style="list-style-type: none"> 縦表示, 横表示ができています。 縦表示, 横表示を順次的にプログラミングしている。 「待つ」命令を正しく組み込んで, プログラムを作成。 	<ul style="list-style-type: none"> アニメーションの回数を増やすプログラムを自主的に作る。
抽出被調査者 2	<ul style="list-style-type: none"> 縦表示, 横表示ができています。 縦表示, 横表示を順次的にプログラミングしている。 「待つ」命令を正しく組み込んで, プログラムを作成。 	<ul style="list-style-type: none"> 「待つ」の配置を変更して, 動作確認を繰り返す。
抽出被調査者 3	<ul style="list-style-type: none"> 縦表示, 横表示ができています。 縦表示, 横表示を順次的にプログラミングしている。 「待つ」命令を正しく組み込んで, プログラムを作成。 	<ul style="list-style-type: none"> 「待つ」の長さの数値を変更して, 動作確認を繰り返す。 メニューから様々な命令を取り出して, プログラムに組み込み, 動作確認を繰り返す。
共通した特徴	<ul style="list-style-type: none"> 表示技能が習得されていて, アニメーション表示させるために順次的に配列させる方法が理解できている。「待つ」命令を正しく組み込んで, プログラムを作成できている。 	<ul style="list-style-type: none"> 命令や数値変更を繰り返したり, アニメーションを増やしたりするなど, 新しいプログラム作成に挑み動作確認する活動が行われている。

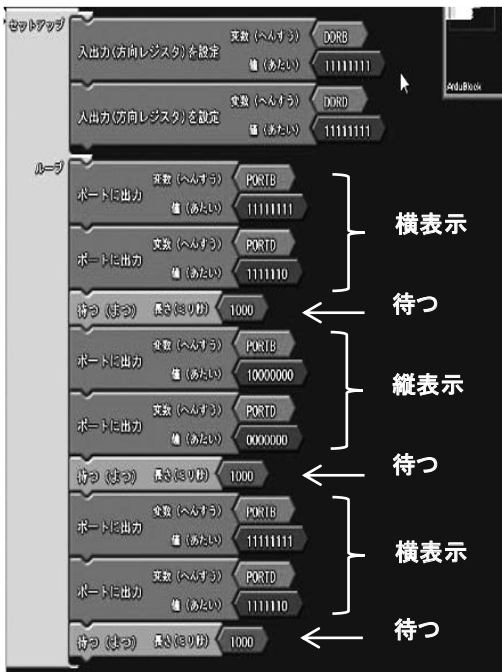


図10 上位群抽出被調査者1の支援導入前のプログラミング



図11 上位群抽出被調査者1の支援導入後のプログラミング

2.3. 中位群の操作内容

中位群の抽出被調査者に見られた、プログラミングの特徴を表 8 に示した。

表 8 中位群抽出被調査者のプログラミングの特徴

抽出被調査者	身体動作導入前	身体動作導入後
抽出被調査者 1	<ul style="list-style-type: none"> ・縦表示, 横表示ができている。 ・縦表示, 横表示を順次的にプログラミングしている。 ・表示命令の並び順を変えて動作確認。 ・表示命令の数値を変更して動作確認。 	<ul style="list-style-type: none"> ・「待つ」命令を正しく組み込んで, プログラムを作成。 ・アニメーションの回数を増やすプログラムを自主的に作る。
抽出被調査者 2	<ul style="list-style-type: none"> ・縦表示, 横表示ができている。 ・縦表示, 横表示を順次的にプログラミングしている。 ・表示命令の並び順を変えて動作確認。 ・様々な命令を組み込んで動作確認。 	<ul style="list-style-type: none"> ・「待つ」命令を正しく組み込んで, プログラムを作成。 ・「待つ」の長さの数値を変更して, 動作確認を繰り返す。
抽出被調査者 3	<ul style="list-style-type: none"> ・縦表示, 横表示ができている。 ・表示命令の数値を変更して動作確認。 ・「待つ」を不適切に組み込んで, プログラムを作成。 	<ul style="list-style-type: none"> ・縦表示, 横表示を順次的にプログラミングしている。 ・「待つ」命令を正しく組み込んで, プログラムを作成。
共通した特徴	<ul style="list-style-type: none"> ・縦表示, 横表示の方法が理解できている。 ・「待つ」命令を正しく使ったプログラムを作ることができていない。 	<ul style="list-style-type: none"> ・「待つ」を正しく組み込んでプログラムを作成できている。 ・数値変更したり, アニメーションを増やしたりするなど, 新しいプログラム作成に挑み動作確認する活動が行われている。

中位群の抽出被調査者に共通するプログラミングの特徴を見ると、身体動作によるプログラム理解の支援を導入する前で、LED を縦、横に表示する方法については理解できていることがわかった。また、抽出被調査者 3 名のうち 2 名は、縦の表示と横の表示を順番に並べることができており、順次性を意識してプログラミングできる段階に達していた(図 12)。残りの 1 名は順次性を意識したプログラミングはできていないが、「待つ」を表出させることができしており、プログラムにおける時間の感覚は、明確ではないものの捉えつつある段階と考えられる。このことから、中位群被調査者における、身体動作によるプログラム理解の支援を導入する以前では、順次性に関しては理解できているものの、プログラムにおける時間の感覚には気づけていない段階、もしくは、プログラムにおける時間の感覚を捉えかけてはいるが明確に捉えられていない段階と推察する。

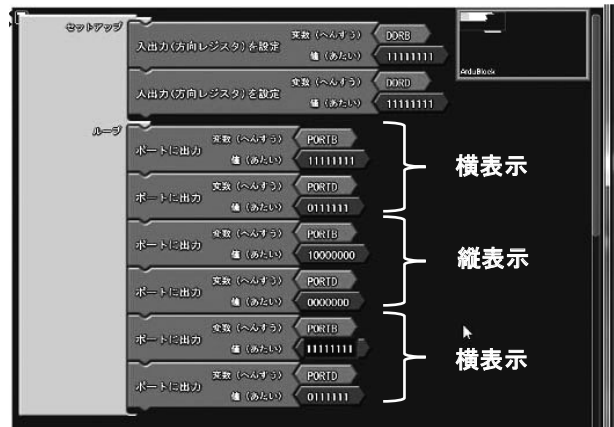


図 12 中位群抽出被調査者 1 の支援導入前のプログラミング

身体動作によるプログラム理解の支援を導入した後は、抽出被調査者は共通して、順次性を意識し、かつ「待つ」命令を適切に用いてプログラムを作成できていた。このことから、プログラム

の順次性、及びプログラムにおける時間の感覚に気づき、「待つ」命令を適切に組み込むことで、アニメーションプログラムが実現できたと考えられる。身体動作を通してプログラムの手順を確認したときに、順次性が間違っていたわけではないことに気づき、さらに身体の動作認識を細分化したことによって、動作時間の感覚がアニメーションさせるために必要な手順に関わるものとして意識されたと推察する。つまり、身体動作によって、プログラムの配置の誤りではなく、高速動作による視覚的影響であることに気づき、プログラムの動作を視認させるために一時的にプログラムを停止させる「待つ」という命令の必要性への気づきにつながったのではないかと考える。

プログラムの完成後は、中位群抽出被調査者においても、上位群被調査者同様に「待つ」の時間を変更して確認

したり、アニメーションを増やしたりするなどの傾向が見られた(図 13)。プログラムの順次性、及びプログラムにおける時間の感覚への理解が深まり、自分なりのプログラムを作成しようとする意欲が高まり、プログラムを発展させていると推察する。



図 13 中位群抽出被調査者 1 の支援導入後のプログラミング

2.4. 下位群の操作内容

下位群の抽出被調査者に見られた、プログラミングの特徴を表 9 に示した。

下位群抽出被調査者に共通するプログラミングの特徴を見ると、身体動作によるプログラム理解の支援を導入する前では、3 名とも LED の縦、横表示がうまく行えていない様子が見られた(図 14)。

表 9 下位群抽出被調査者のプログラミングの特徴

	身体動作導入前	身体動作導入後
抽出被調査者 1	<ul style="list-style-type: none"> ・不規則に表示命令を並べてプログラム作成。 ・操作が止まる 	<ul style="list-style-type: none"> ・「待つ」を不適切に組み込んで、プログラムを作成。
抽出被調査者 2	<ul style="list-style-type: none"> ・不規則に表示命令を並べてプログラム作成。 	<ul style="list-style-type: none"> ・「待つ」を不適切に組み込んで、プログラムを作成。
抽出被調査者 3	<ul style="list-style-type: none"> ・不規則に表示命令を並べてプログラム作成。 	<ul style="list-style-type: none"> ・「待つ」を不適切に組み込んで、プログラムを作成。
共通した特徴	<ul style="list-style-type: none"> ・縦表示、横表示がプログラミングできない。 ・「待つ」命令を正しく使ったプログラムを作ることができていない。 	<ul style="list-style-type: none"> ・「待つ」命令を正しく使ったプログラムを作ることができていない。

このことから、順次性を意識したプログラミングをする以前に、表示させることの理解が不十分であることがわかる。いろいろと操作はしているものの、目的に向かって構成していくというよりは、気が向くまま無作為に操作しているように見えた。このことから、下位群被調査者は、本授業時間に最低限として持つ必要がある LED を縦、横表示させる、という技能を習得できていないことが推察される。LED を縦、横表示させる処理の学習は前授業時間の内容であり、その段階で躓いているといえる。したがって、この授業時間に目標とする動作に向かうことができず、意図したプログラム作成ができなかったものと考えられる。

身体動作によるプログラム理解の支援を導入した後は、「待つ」命令をメニューから取り出して、無作為に組み込む様子が見られたものの、LED の縦、横表示を順次性を意識して構成するプログラムは、最後まで作成できなかった(図 15)。「待つ」命令を取り出せている点のみを捉えれば、下位群被調査者もプログラムにおける時間の感覚に気づけたと考えられる。しかし、下位群被調査者の操作からは、最後までプログラムの順次性を意識して構成できていないことが観察から確認された。本授業において、順次性の理解なくしてはプログラムにおける時間の感覚について気づいているとは考えにくいことから、「待つ」命令の必要性に気づいたというよりは、周囲のプログラミングの様子を見て模倣しようとしたと推察する。下位群抽出被調査者が最後までプログラムを完成させることができなかったことについて、LED を点灯させる処理について理解できていないということが、最後まで被調査者の理解を妨げたと考える。



図 14 下位群抽出被調査者 1 の支援導入前のプログラミング



図 15 下位群抽出被調査者 1 の支援導入後のプログラミング

IV. まとめ

これまでのプログラミング学習では、英単語をベースとしたプログラム言語の命令とその使い方を理解し、テキストでプログラムを入力しなければならなかった。当然、タイプミスなどによる、プログラムの誤動作も多く見られた。また、プログラムでは、タイプミスが無かったとしても、目的とは違う動作をしてしまうことがある。処理の手順を間違えたり、一部不足したりした場合である。岡本ら¹²⁾も、これらの点を初学者がつまづく要因として挙げている。こうした中で、サンプルプログラムの模写、動作確認を行う学習方法は、正しいプログラムを学習者に提示するために、タイピングや命令、構文の理解を進めやすいと捉えられていたと考えられる。しかし、英単語の意味や使い方を理解しなくても論理構造のみでプログラミング可能なビジュアルプログラミング言語が開発されている今、本稿では、中学生を対象とした授業において、学習者が自ら命令や基本的な処理の意味を探り、プログラミングすることが可能であることを明らかにした。

それを可能にしたのは、命令が日本語表記され、メニューによって直感的に配列されたプログラミングエディタのインターフェースと、授業で実施した身体動作の支援であると推察する。これまでの英単語のテキスト入力を主体としたプログラミングと違い、複雑な使い方の理解も、英単語のタイプミスに気を遣う必要もない。日本語表記なので、命令の意味も感覚的に理解できる。このようなツールを用いることで、論理構造の思考のみに焦点をあてた授業構成が可能になった。また、プログラムの命令と対比して身体を動作させることで、身体動作の細分化とその可視化を促し、プログラム上で見落としていた処理に気づかせた。

このように本稿では、プログラミングの経験が少ない中学生でも、ビジュアルプログラミング言語を用い、身体動作による支援を行うことで、自ら思考してプログラムを作成できるという結論を得た。このことは、既有知識や概念がないということで、正しいサンプルプログラムを模写、動作確認を行う教授法が適切と考えられていたプログラミングの学習方法に、一石を投じるものである。学習者の生活経験や身体動作に関して認識を深めていくことで、類似した構造のプログラムを学習者が自ら気づき、自身の思考によってプログラミングは可能になるのである。

プログラミングを教育に取り入れる意義について、山本ら¹³⁾は、以下のように整理している。

- ①新たなものを生み出したり、難しいものに挑戦しようとする探求力が身につく。
- ②アルゴリズム的思考、論理的思考力が身につく。
- ③物事や自己の知識に関する理解力が身につく。
- ④自分の考えや感情が発信できる表現力や説得力などのコミュニケーション力が身につく。
- ⑤知恵を共有したり他者の理解や協力して物事を進めたりする力が身につく。
- ⑥プログラミングを通して、情報(技術)的なもの見方や考え方が身につく。
- ⑦プログラミングを学ぶことを通して、プログラミングが設計された基礎にある事象(現実)のとらえ方(見方・考え方)を身につけることができる。

これらの意義を、より効果的に実現するためには、学習者が単に記憶に頼ってプログラムを作成するべきではない。文脈に応じて、命令やその構成をどのようにするのかを、自ら思考しつつ、プログラムを組み立てる経験を重ねるべきである。本稿での実践は、学習者に命令や基本的な処理の意味を探究させつつ、思考させてプログラミングを実践する学習方法であり、今後のプログラミング学習の目指すべき方向性を提示できたと考える。

V. 今後の課題

本稿では、身体動作による支援、及びビジュアルプログラミング言語の利用によって、自ら思考しつつ、命令の意味やアルゴリズムを理解し、プログラムを作成できることを明らかにした。しかし、どのような過程を経て、命令の意味やアルゴリズムを理解し、プログラムを作成しているのか明確とはいえない。思考過程をより精緻に観察し、思考過程を明確にしていくことが、プログラムの命令の意味や基本的な処理を理解する過程の解明につながると考える。今後も同様の教授方法を用いた授業を実践し、詳細に観察し、中学生段階でのプログラミング学習の思考過程を明らかにするとともに、効果的な教授方法を探る必要があると考えている。また、プログラミングという作業自体が持つ、問題解決への方策と論理的手順の検討、という特徴を踏まえると、土井¹⁴⁾の「段取り」研究で述べられているように、学習者の技術的能力の形成過程の解明、さらに日常的な生活場面における意思決定の解明に結びつくものであると考えている。

VI. 参考文献

- 1) 文部科学省. 中学校学習指導要領解説 技術・家庭科編. 2012.
- 2) 全日本中学校技術・家庭科研究会. 中学校技術・家庭科理論と実践No.54. 全日本中学校技術家庭科研究会, 2016, pp. 52-59.
- 3) 米国学術研究推進会議. 授業を変えるー認知心理学のさらなる挑戦. 北大路書房, 2002, pp. 245-258.
- 4) 岡本雅子, 村上正行, 吉川直人, 喜多一. プログラミングの写経型学習過程を対象としたつまずきの分析とテキスト教材の改善. 京都大学高等教育研究. 2013, vol. 19, pp. 47-57.
- 5) 中田豊久. 写経プログラミングの学習効果に関する考察. 人工知能学会全国大会論文集, vol.27, 2013, pp. 1-3.
- 6) 荻嶺直孝, 宮川洋一, 森山潤. 中学校技術科「プログラムによる計測・制御」の学習における題材タイプの違いによる生徒の反応の差異. 日本産業技術教育学会誌. 2013, vol. 55, no. 3, pp. 181-190.
- 7) 市川貴子. 教育現場からの実践報告 中学校技術科における計測・制御学習教材の開発と授業実践 : 生徒一人ひとりが課題を解決する製作活動. 日本産業技術教育学会誌. 2015, vol. 57, no. 1, pp. 59-64.
- 8) 藤田眞一, 加賀江孝信, 三浦吉信. 小型コンピュータを用いたプログラムによる計測・制御学習の効果. 日本産業技術教育学会誌. 2013, vol. 55, no. 3, pp. 191-198.
- 9) 増田麻人, 大村基将, 片田宗一郎, 紅林秀治. 状態遷移図を利用したプログラムによる計測・制御教材の開発. 日本産業技術教育学会誌. 2015, vol. 57, no. 2, pp. 93-101.
- 10) 稲垣佳世子, 羽多野諄余夫. “学校化された学びのゆがみ”. 岩波講座3現代の教育 授業と学習の転換. 岩波書店, 1998, pp. 70-91.
- 11) エルンスト・ペッペル. 意識の中の時間. 岩波書店, 1995.
- 12) 岡本雅子, 喜多一. プログラミングの「写経型学習」における初学者のつまずきの類型化とその考察. 滋賀大学教育学部附属教育実践総合センター紀要. 2014, vol. 22, pp. 49-53.
- 13) 山本利一, 本郷健, 本村猛能, 永井克昇. 初等中等教育におけるプログラミング教育の教育的意義の考察. 教育情報研究. 2014, vol. 32, no. 2, pp. 3-11.
- 14) 土井康作. 組立課題における作業段取りの効果について. 教育心理学研究. 1998, vol. 46, no. 1, pp. 68-76.

