

認知過程に考慮し、並び替えを目指す高校情報科の アルゴリズムプログラミング教育の開発と実践

太田 剛^{†1}

概要：新学習指導要領の情報Iにおいてアルゴリズムのプログラミング教育が必須となり、プログラミング経験の浅い教員がいることと、生徒の学習の困難さが指摘されることから、いろいろな実践が試行されている。但し、並び替え等のアルゴリズムのプログラム実践は十分に行われていない。筆者は、プログラミングに不慣れな教員でも、生徒が主体的にアルゴリズムのプログラミングの学習ができる授業を実施できることを目的に2018年度より教材・授業を、定期的な改善サイクルを行うデザインベース研究として開発してきた。2020年度は従来のフローチャートの利用から、よりプログラミングの認知過程を考慮したプログラミング部品の考え方をとり入れた教材を開発し、2022年度からの情報Iでのプログラミング教育用の教材・授業方法の整備に目途がついた。

キーワード：プログラミング教育、高校情報科、プログラミング的思考、デザインベース研究

Development and Practice of Algorithm Programming Lesson in Informatics Education for High Schools by Considering Programming Cognitive Process

GO OTA^{†1}

1. はじめに

新学習指導要領[1]において、高校情報科で必修となった科目「情報I」の内容に「コンピュータとプログラミング」が含まれることから、新高校生が2022年度よりプログラミングを学ぶこととなる。この実施にあたっては、情報科の教師にプログラミングの高度な内容が教育できるか問題視され、現在先行しているいろいろなプログラミングの授業の実践が始まっている。ただし、それらは簡単なプログラミングを扱ったものが多く、検索や並び替えを扱ったプログラミングの実践は十分に行われていないようである。

これに対して、筆者は2018年度より、普通の高校生が情報科で検索や並び替えのプログラミングを学習する授業をデザインベース研究の立場から毎年改良し実践している。2020年度からは、特に従来のフローチャートの利用を見直し、プログラムの認知過程に注目したプログラム部品の考えを取り入れた方法を試行している。本稿は、この授業・教材開発と、その実践結果を報告するものである。2章ではプログラミング教育の現状を、3章ではプログラミング部品を利用したプログラミングの考え方を示す。そして、4章では本実践の目的と開発・実践の経緯を、5章で2020年度の実践内容を紹介する。6章では、今後のプログラミング教育について示す。

2. プログラミング教育の現状

情報Iのプログラミングに対応して現在進められている

実践は、①Python/JavaScript等の汎用テキスト言語を使用して、分岐や繰り返しなどを学習する[2]、②ビジュアル言語を用いて分岐や繰り返しなどを学習する[3]、③アンプラグドで並び替えなどのアルゴリズムを学習する[4]、④データサイエンス[5]やWebなどのアプリケーションなどの実用プログラムを開発する[6]、⑤ロボットや他のフィジカルな装置を制御・計測する[7]、等に大きく分類されると考えられる。ただし、高校での検索や並び替えに関しては、文部科学省のプログラミング教育実践ガイドとして「基本的なアルゴリズムの学習」[8]の指導案は公開されているが、その中にある探索・並び替えについての具体的な実践報告は公開されていないようである。中西[9]の「検索はともかく整列は二重ループが使われるので、生徒にとって簡単なものではない」の言葉にあるように、普通の高校では並び替え等のアルゴリズムをプログラミングする実践は十分に行われていないようである。

これに対して高等教育では情報関連の学部・学科においてアルゴリズムのプログラミング教育は長年行われ、一般に配列・添え字による配列の操作・二重ループなどが学生にとって難しいものであることは、多くの指導者の経験知として認識されていると考えられる。そして、これらの問題に対応する支援システムや教授法が開発・研究されてきて、伊藤ら[10]は、これらを、大きく「説明生成(可視化)型」システムと、「診断・指導型」システムとに分類している。「診断・指導型」システムは、デバッガやトレーサなど

^{†1} 千葉県立生浜高等学校/関東第一高等学校
Chiba Prefectural Oihama High School/Kanto Daiichi High School

のプログラム作成支援ツール[11], アルゴリズムアニメーション, 変数の可視化ツール[12]等である. そして, 「診断・指導型」はプログラムの自動判定のフィードバックや穴埋め問題提示などのプログラムの作成補助ツール[13]である.

ただし, 多くのプログラミング授業では, 高岡ら[6]が示したように「教授者が学習者に対して演習課題を提示し, 学習者がある演習課題に取り組む形」であり, 「学習者が参考書やその他の資料などを参照し, 解決法などの本質的な理解をすることなく演習課題を完成させてしまうことがある。」というような問題もある.

3. プログラミングの認知過程を考慮したプログラム部品を中核とする教材の改良

高等教育でもプログラミングが出来る学生と出来ない学生がいることは, ふたごぶらぐダ問題[14]と認識されていることである. そして, 高校の情報 I で生徒がプログラミングを学習する場面において, 個人のプログラミングの適性の有無に関係なく, すべての生徒がどのようにしたらプログラミングを習得できるか考える必要がある. そのためには, 三宅[15]の「プログラミング学習において, 何らかの認能力を伸ばそうとするのなら, そもそもプログラミングがどのような認知過程なのか, そこから調べていかなくてはならない」というアプローチが必要であると考える..

従来の演習問題によるプログラミング教育では, 図 1 に示すような, 問題解決として, 課題からアルゴリズムを作成しプログラミングする指導を行ってきた. これは課題を順次, 分岐, 繰り返しを基本構造として分析するようなト

ップダウン的な認知過程を想定しているとも考えられる. ただし, 単純な課題では, このアプローチは有効であるが, 並び替えのような難しい問題では有効では無いと考えられ, そのため生徒・学生はプログラムをなんとか完成しても, 類似した問題が出来ないというように応用力が身に付かない問題が指摘されていると考えられる.

従来の学習方法は三宅の指摘したプログラミングの本質的な認知活動を考慮したものであったか疑問に思う. 筆者も 2020 年前期までは, トップダウン的にフローチャートを用いて, アルゴリズムを理解し, プログラムを作ることを教材に取り入れていたが, ① 並び替えなどの二重ループになるとフローチャートでの表現・理解が難しくなる. ② フローチャートを見ても, それから一連の命令に置き換えるという作業が困難であるという生徒の状況を見ていて, 従来の考え方に行き詰っていた.

図 1 とは異なるモデルを考える上で, 筆者は子供のプログラミングを観察した中では, 子供はアルゴリズムを考えて, それを実現するために個々の命令を使うよりは, ブロック(命令)を組み合わせた機能単位(プログラム部品)を利用することによってプログラミングを学習していることも分かっていた[16]. 同様な見方として, 遠藤は, 「競技プログラミングというのは(中略), 彼らは, 問題を見た時に自分の頭に入っているアルゴリズムを応用できるかどうか<<宝さがし>>のような感じだというのだ」と述べている[17]. 実際の研究では, 吉池らは学習者が複数の命令をまとめた部品として認識することにより, プログラムの意味理解や作成に役立つことを指摘している[18].

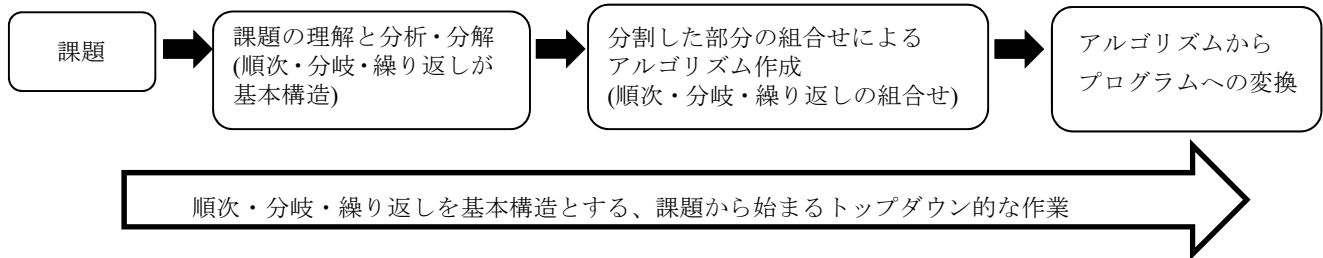


図 1 従来の演習課題の実施の流れと, それから想定されるプログラミングの認知過程

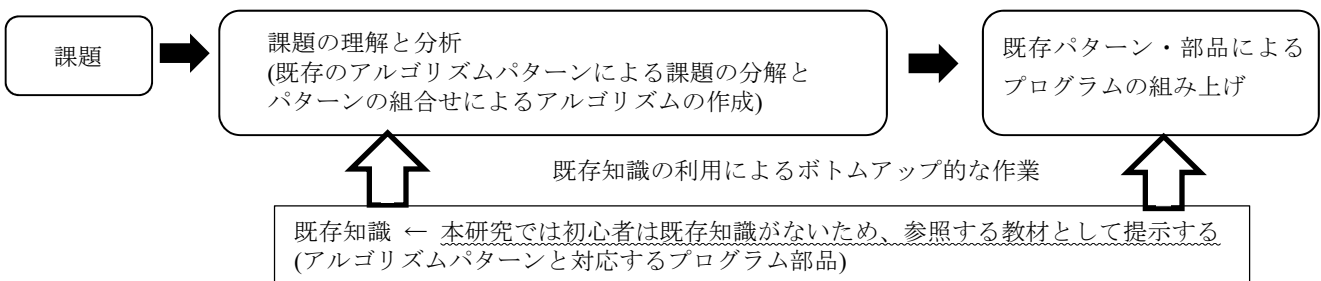


図 2 プログラム部品を想定した問題解決としてのプログラミングの認知過程モデル

これから、ある程度プログラミングが出来る人は、プログラムを作る時に順次・分岐・繰り返しなどの基本構造の組合せではなく、もっと大きなアルゴリズムのパターンやそれに対応したプログラム部品から課題の理解・分解が始まり、それらを組み合わせるといのではないかと考え、図 2 に示すモデルを想定した。ここにおいて、プログラミングとは、すでに知識としてあるプログラミング部品を組み合わせて問題を解決するボトムアップ的な認知過程であると再定義できる。実際の教材や学習方法は、4 章に詳細を示すが、プログラミングの初心者である生徒には既存の知識がないため、それを参照情報として利用できるプログラミング部品を提示し、それを組み合わせながら学習する教授方略をとることにした。

4. 本開発の目的・内容と過去の実践内容

本開発・実践の目標は情報 I におけるプログラミング教育の実現が主目的であり、まず、つぎのような条件を考えた。

条件 a: 6 時限の授業で、多くの生徒が並び替えのプログラミングを学習できる。

情報 I の年間カリキュラムを考えた場合、アルゴリズムのプログラミングに配置できるのは 6 時限程度と想定している。また、情報科目が大学共通テストに採用されることから、過去の大学共通テストや私学の入試問題を分析した結果、ある程度難易度の高いアルゴリズムの出題も多く、それらの基礎となる並び替えを目標として学習する。

条件 b: プログラミング経験の少ない教員でも授業が実施できて、生徒も主体的に学習できること。

生徒は用意された資料をもとに、個人のペースでプログ

ラムを作成し、教師はできない生徒の支援を行う、「教えないプログラミング学習」という授業スタイルとした。実際、生徒間の教えあいで解決する問題も多く、教師も初年度は生徒といっしょに学習してもいいかもしれない。

条件 c: 単なるお手本を写して作るではなく、考えてプログラミングする。

一般に、提示されたサンプルプログラムを模写・入力・実行確認を行う写経型学習は必要な場面もある[19]が、すべての学習が写経だけならば、生徒はプログラミングする能力を獲得するのは難しいと考える。そのため、何らかの課題からプログラミング完成の間に生徒が考える内容を取り入れる。

また、従来のプログラミングの実践研究では、特定の教授方法や支援システムの有効性を検証するため対照実験的な方法がとられるが、本研究ではデザインベース研究の考え方から授業や教材を毎年改善するサイクルを継続することで、より効果的な教授法を創り出すことを目指している。

このような条件と研究で実践した経緯を以下に示す。

(1) 2018 年度開発と実践

2018 年度は、先行実施されている高校・大学等のプログラミング授業を参考に、以下のような点を考慮して教材開発し、高校の情報専門科目「アルゴリズムとプログラム(8 時限)」の授業で実践した。

① 可視化とトレース

変数などの情報を可視化することが有効であることから、ビジュアル言語である Scratch を使用することとした。そして、授業時には、繰り返しにおいて、変数の値の変化を紙に書くようにトレースの指導を行った。

表 1 アルゴリズムプログラム学習チェックシート(2020 年度版)

	No	内容	[理解]	[写経]	[開発]	
ここまで、 やってみよう。	1	準備運動 1: Scratch の四則演算	<input type="checkbox"/>	<input type="checkbox"/>		
	2	プログラムに名前をつけて保存/新しいプログラム	<input type="checkbox"/>	<input type="checkbox"/>		
	3	準備運動 2: 円から韓国ウォンに変換	<input type="checkbox"/>	<input type="checkbox"/>		
	4	課題 1: 入力した 2 個の数で四則演算			<input type="checkbox"/>	
	5	打ち込み 1: 合格判断	<input type="checkbox"/>	<input type="checkbox"/>		
	6	プログラムに名前をつけて保存	<input type="checkbox"/>			
	7	課題 2: 合格不合格判断			<input type="checkbox"/>	
	8	課題 3: 成績 A~C			<input type="checkbox"/>	
	9	打ち込み 2: 1 から 10 までの数を言う	<input type="checkbox"/>	<input type="checkbox"/>		
	10	打ち込み 3: 1 から 10 までの合計を言う	<input type="checkbox"/>	<input type="checkbox"/>		
	11	課題 4: 2 から X までの偶数の合計			<input type="checkbox"/>	
	12	打ち込み 4: おみくじをつくる	<input type="checkbox"/>	<input type="checkbox"/>		
	13	打ち込み 5: リストを使った 5 つの数の合計	<input type="checkbox"/>	<input type="checkbox"/>		
	14	最後目標の課題の説明: 数の並び替え	<input type="checkbox"/>			
	15	変数/リストシートを使って考えよう	<input type="checkbox"/>			
	16	課題 5: リストの中から数を探す			<input type="checkbox"/>	
	19	課題 6: リストの中の一番小さい数を見つける			<input type="checkbox"/>	
	21	課題 7: 一番小さい数を配列の先頭に入れ替える			<input type="checkbox"/>	
	22	打ち込み: 二重繰り返しで九九に挑戦	<input type="checkbox"/>	<input type="checkbox"/>		
	23	打ち込み: 複雑な二重繰り返しに挑戦	<input type="checkbox"/>	<input type="checkbox"/>		
	24	最後のチャレンジ: 数の並び替え			<input type="checkbox"/>	
	もっと、や ろう	25	発展課題 1: Fizz Buzz			<input type="checkbox"/>
		27	発展課題 2 バブルソート			<input type="checkbox"/>
		30	発展課題 3: 並び替えの方法の処理速度の違い			<input type="checkbox"/>

② アルゴリズムとプログラミング言語の学習の分離

江木ら[11]は、「プログラミング教育には、プログラミング言語、アルゴリズム、構築技術の3つの主要な側面がある、学習者にとっては、これら3つは不可分でありほぼ同時進行で学習が進む。」と述べているが、生徒にとって同時進行での学習は難しく、アルゴリズムのプログラミングの前に、Scratchを使用した自由なプログラミング作成の授業を実施し構築技術を習得する時間とした。

③ 問題解決的要素を含む演習課題

生徒が学習する手順書として Web 上に置かれた pdf に、説明・課題・ヒントなどを用意した。また表 1 に示すチェックリストを用意して、生徒が自分で進捗を記入するようにした。課題は、後半部分は教科書に沿った探索、並び替えとしたが、前半部分は基本的な順次、選択、繰り返しをフローチャートと対比させながらプログラムを開発するものであった。

結果としては、8 時限授業であったが、並び替えまで完成する生徒はいなかった。そして、教材自体の問題点として、入力や変数、配列の説明が不十分であり、先に進めない生徒が多かった、例題や課題などの粒度が大きく課題実施が難しかった、筆者にとって初めての授業で生徒がどの程度プログラミング理解できるかの情報がなかったためヒントなどが不十分だった、などがあげられる。

(2) 2019 年度開発と実践

2018 年度の経験を生かし、教材を次のような点で改良し、高校の情報専門科目「表現メディアの編集と表現(6 時限)」の授業で実践した。

① タンジブルな変数環境の利用

従来はトレースなどを紙に書きだすことや、画面上に可視化することで支援することが行われた変数や配列について、具体的に手に取って操作でき、変数のイメージに近い箱型の模型をタンジブルな変数環境として用意することとした。そして、繰り返しのカウンターとして使用する変数については、その操作がよりイメージしやすいように計数機を使うこととした(図 3 参照)。さらに手順書も生徒がプログラムのイメージをつかみやすいようにするため、タンジブルな変数、フローチャートとプログラムを明確に関連付けるようにした(図 4 参照)。

② スモールステップであるが考える内容

2018 年度の実践から、生徒がプログラミングのどのような点でつまずくか判ってきたため、生徒がスモールステップで確実にプログラミングできるように教材の粒度を細かくした。ただし、すぐに生徒が回答を見つけたり、単に例を写経のようにコピーするのではなく、ある程度考えて作ることも考慮した。例えば、1~n の合計を作るプログラムを完成後、1~n の偶数の合計を求めるプログラムでも生徒にとってハードルが高いため、偶数をプログラムで

作るヒントなど追加した。また、入力と変数に関する問題や説明も大幅に追加した。

実践結果としては、生徒 16 名中 2 名が並び替えのプログラミングを完成させた。ただし、入力・変数・分岐で、躓く生徒は多く、また、フローチャートをヒントで出しても、それが Scratch の命令に変換できない生徒も多かった。

(3) 2020 年度開発

前述したようにプログラム部品の考え方を取り入れて大幅に教材を改良した。

① プログラム機能部品カードの作成

並び替えのプログラムを作成するまでに必要な機能を 11 個定義し(表 2 参照)、その構成や使い方を示したカードを作成した。カードは生徒に渡し、いつでも参照できるようにした(図 5 参照)。また手順書のプログラム機能部品を組み合わせで作成するように直した。同一課題のヒントをフローチャートで提示したものと、プログラム機能部品を使用したものも図 4 に示す。

② 課題とヒントの精査

入力や四則演算について、課題を追加するとともに、分岐については、三角形の判断は時間がかかるため、単純な合格・不合格の判定にした。また、並び替えのプログラムを作成するために単純な二重ループなどの写経用のプログラムを追加した。なお、2019 年度に使用したタンジブルな変数環境については、準備に時間がかかることから普及版として紙に変数を印刷したシート(変数シート)を用意した。

5. 2020 年度の実践と結果・考察

(1) 実践方法

「社会と情報」の授業の中でアルゴリズムとプログラミングの単元として実施。なお、6 時限で行ったが、コロナ禍の短縮授業のため、40 分の 2 コマ連続授業を 3 回行い実質的には、40 分 x 6 回で 240 分であり、通常の 50 分授業の 5 時限分に満たない時間であった。1 年生 40 名を対象としたが、欠席者や学習データの使用を望まない生徒を除いた 30 名を分析の対象とした。なお、生徒の 21%が中学までにプログラムの経験があった。前述したプログラム部品カード、チェックリスト、変数シートを個々の生徒に配り、生徒は Web 上の pdf である手順書をみながら、Web 版の Scratch で開発を行った。

また、本授業の前に、筆者の作成した Scratch のサンプルプログラムのカードをもとにした 2 時限の授業で自由に Scratch プログラムを作る授業を行った。

(2) 実践結果

① 課題達成状況

各生徒がどこまで課題を達成したか、チェックリスト

を



図 3 タンジブルな変数環境(2019 年度用)

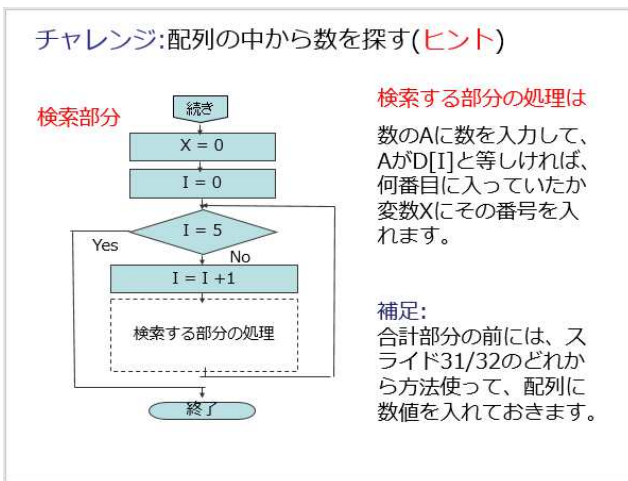
もとにした結果を図 6 に示す。なお、大まかに、どの課題に時間がかかっているか把握するため、各 2 時限の授業ごとに実施状況がわかるように記録していた。

最終的に 10%の生徒が並び替え、87%がリストの検索まで達成していた。そして全員がリストの合計までは達成した。学習状態を観察すると、偶数の合計に時間がかかる生徒が多く、そして以前の実践と同様に、入力・四則演算・分岐に時間がかかる生徒もいた。

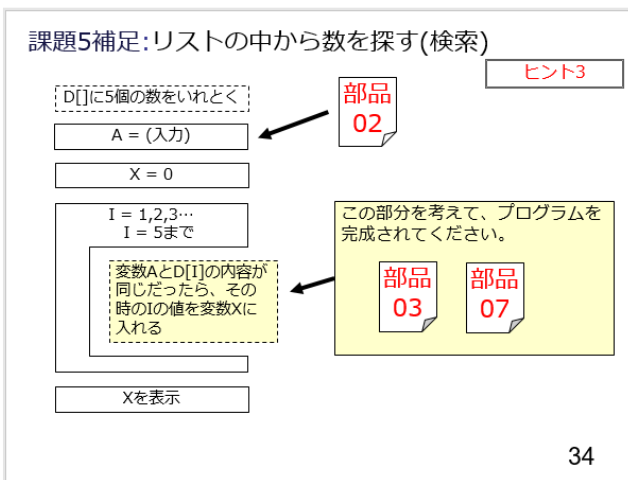
② プログラミングと授業に対する評価

授業の終了時に、プログラミングや授業に対する評価に関するアンケートを実施した(表 3 参照)。

全体として授業やプログラミングは難しいが、プログラミングは楽しいという評価が多いと考えられる。ただし、今後もプログラミングをしてみたいという意見はやや多い程度である。授業については、友達同士での教えあい



a. 2019 年度版(フローチャート利用)



b. 2020 年度版(プログラム部品利用)

図 4 手順書の指示内容例 (どちらも検索のプログラムのヒント)

表 2 プログラム機能部品一覧

プログラム機能部品	
01	四則演算と変数への代入
02	キーボードから数値や文字の入力
03	条件を判断して、命令を実行
04	条件を判断して、○と×で違う命令を実行
05	複数の条件を判断して違う命令を実行
06	数をカウントアップしながら繰り返す
07	リストの処理(取り出し、入れ替え、追加)
08	二つの変数の内容を入れ替える
09	繰り返しとリスト利用の組み合わせ
10	二重の繰り返し
11	二重の繰り返し(内側の繰り返しの開始を変更)

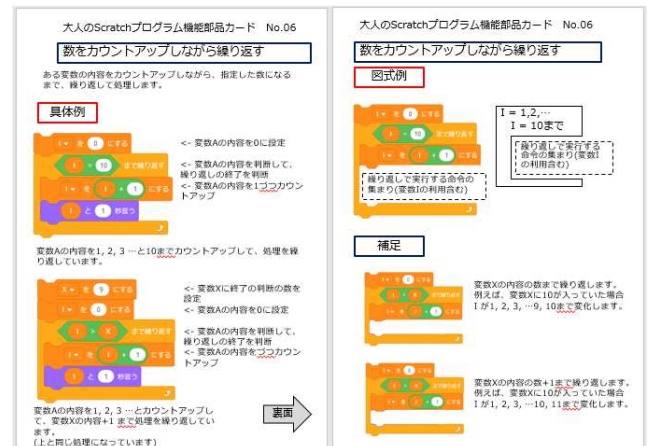


図 5 プログラム機能部品カード例 (A5 で両面印刷で作成)

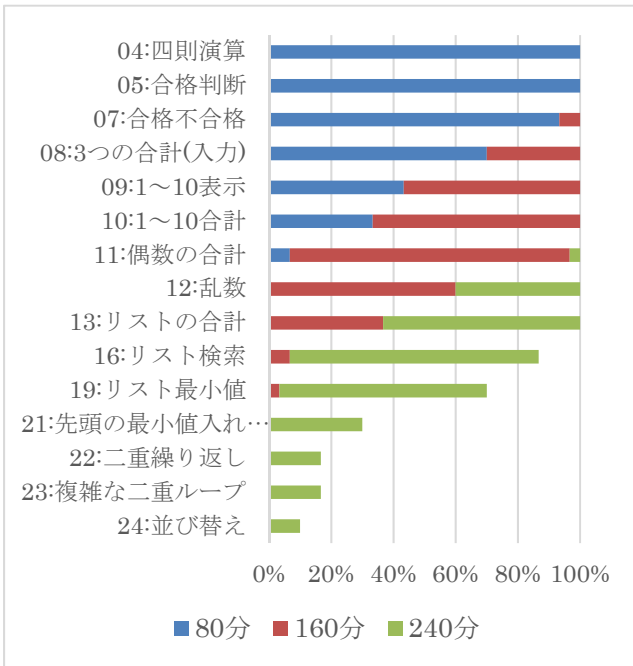


図 6 課題達成状況(2020 年度)

表 3 授業に関する生徒の評価(2020 年度)

評価項目	そう思う-----普通-----思わない				
	5	4	3	2	1
今回の授業は難しかったと思う。	70%	30%	0%	0%	0%
プログラムは難しかったと思う。	73%	23%	3%	0%	0%
プログラムが動いた時、出来た時はうれしかった、楽しかったと思う。	63%	23%	7%	7%	0%
他のプログラミングもやってみたいと思う。	30%	27%	40%	3%	0%
従来の授業と比べて、自分で学習したと思う。	37%	30%	23%	10%	0%
友達に教えたり、教えてもらったりして学習したと思う。	63%	17%	17%	3%	0%
プログラム機能カードは役立ったと思う。	30%	53%	13%	3%	0%

表 4 学習方法(教材)に関する状況

a. 変数シートの利用について

変数シートを使わなかった	87%
変数シートを使って考えた	13%

b. 手順書を見る方法について

主にパソコンで見えていた。	40%
パソコンとスマホの両方で見えていた。	27%
主にスマホで見えていた。	33%

などが多かったことを示しているが、自分で学習したという意見はやや多い程度である。また、プログラミング機能部品カードも役立ったかについては、やや多い程度である。

③ 学習方法(教材)について

本授業では、生徒は手順書、プログラミング機能部品カード、変数シートなどの多様な資料を使い、PC上でプログラミングするという複雑な学習環境であったため、その利用方法もアンケートで調査した(表 4 参照)

本授業では特に、変数シートを使用することを明示しなかったため、それを使ってプログラムの動作などを考えた生徒は 13% だった。そして、手順書もプログラミングをしている PC 上の同一画面でみる必要があるため、60% の生徒はスマホに指示書を表示する方法で学習していた。

(3) 考察と今後の改良の検討

今回の実践は短縮授業のため合計 240 分であったため、本来の 50 分授業であれば、より多くの生徒が並び替えの課題を完成させることができたと予想できるが、まだまだ不十分である。2020 年度の実験結果から今後の改良点などについて以下に示す。

① 入力・四則演算・分岐などの教材の工夫

まだ、多くの生徒がプログラミングの基本的なところでつまづいている。特に入力について、Scratch は会話型で作成するプログラムのため、プログラムの作成と実行の区別がついていない生徒もいるようで、そのため実行時に入力するということが理解しにくいかもしれない。このため、この作成と実行を明確に理解するための教材や解説動画を追加する必要がある。

② 変数シートの利用

変数シートの利用が少ない理由として、シート自体が汎用的な利用を想定していて単に変数の箱があるだけなので、どう利用していいかわからないことが想定される。このため、個々の課題に対応するような明確なシートが必要かもしれない。

③ 課題のヒント自体の精査

例えば、偶数の合計は 1 から 10 までの合計のプログラムを少し変更すれば完成するのに、そこに気がつかず多くの時間を使っている生徒が多い。これらに対しては、適切なヒントを明示するように改良する必要がある。

④ プログラム機能部品カードの利用

本実践の前の 2 時限の Scratch の授業でも同様のカード型の教材を使用していたため、授業の初めで部品カードの内容をそのまま入力すれば良いと勘違いする生徒が多くいた。この点も含め、部品カードの適切な使い方を教材で明記する必要がある。

⑤ 生徒の達成感の考慮

授業中の観察では、明らかに合計の計算や偶数の合計の問題あたりから難しいと考えて学習を進めない生徒が

出てくる。これらの生徒に対して、どのように問題ができるようになるかのヒントの出し方と、達成感を感じて学習を継続する問題のレベルなどを考慮する必要がある。

⑥ 利用しやすい手順書の準備

同一 PC 画面上で手順書を表示して、プログラミングすることは生徒にとって困難なようであり、多くの生徒がスマホで手順書を表示していた。このことから手順書は印刷した資料で準備して利用しやすくする必要がある。

6. まとめと今後の課題

(1) 情報 I のプログラミング教育として

本開発・実践の主目的である、情報 I におけるプログラミング教育の実現については、前述したような教材・授業方法の改善を行い、今後 2021 年度に実践を行うことで、そのモデル教材・授業を確立することが出来ると考えられる。

例えば、「自分で学習したと思う。」の評価がそれほど高くないのは、友達に教えてもらって課題を達成した生徒がいるからだと考えられる。これは対話的な学習としては意味があるが、個々の生徒が本当にプログラミングを理解しているかということには疑問が残るだろう、また今回提案したプログラミング機能カードが生徒の学習に役立っているかという点や、今回学習した内容が他のプログラミングの作成にも応用できるような知識になっているかという疑問も当然残る。

ただし、プログラミングに不慣れな情報科の教員が多い、アルゴリズムの有効な授業実践事例がほとんどない現状において、実質的に授業を行うことについて、有効な教材や授業方法を提供できると考える。ただし、さらなる目標として、現在は生徒用の教材としてプログラミング部品カードや指示書を紙で印刷して準備する必要がある手間がかかるという問題がある。Web 上だけの教材で学習できるような手軽さが望まれると考える。

また、2025 年からの大学入試では高度なプログラミングの問題が出題されることも予想されて、この授業内容では不十分である。これに対しては情報 II の授業を想定して Python を使用した、大学入試想定問題集の開発も行っている。

(2) プログラミングの認知過程の研究として

今まで曖昧なままのプログラミングの認知過程について、図 2 に示したモデルの提示は、今後、認知過程を意識した研究を促進する契機になると考える。ただし、本研究は上記のように情報 I のプログラミング教育を実現するための、役に立つ教育方法を開発することが目的であり、プログラミングの認知過程自体についての内容や検証については不十分なものである。例えば

・プログラミングに適性のある人とならない人の認知過程は、本研究で提案したモデルで明らかに違いがあるか、そして正しいモデルと考えられるか。

・モデルが正しいとして、既存知識であるアルゴリズムパターンやプログラム部品が学習により獲得できるものなのか、そして、それらが別の課題場面で転移して利用できるものなのか。

・モデルが正しいとして、プログラミングに適正のある人は、外的な支援がなくて、どのように既存知識のアルゴリズムパターンやプログラム部品を獲得していくか、そして、それらで対応できない課題に対して、どのように新しい知識を構築するのか、もしくは獲得するのか。

・オブジェクト指向プログラミングの教育ではオブジェクトの設計やカプセル化の学習が行われている、これらは本モデルにおける既存知識に近い性質のものであり、これらのオブジェクト指向の考えとモデルの関連には興味がある。などについて、今後は対照実験的な研究が必要であると考えられる。

ただし、今後、プログラミングの認知過程が明かになることで単なる表面的な支援でなく、応用力のあるプログラミング能力を獲得することを目指した適切な教育が実施できると考える。

(3) プログラミング的思考を考える

小学校ではプログラミング的思考の習得を目的にプログラミング教育が始まっている。例えば、堀田[20]は「間違っただけではないのは、コンピュータによるプログラミング体系を一度もしていない児童にとっては、仮に割り算の筆算アルゴリズムを論理的にトレースする学習をおこなっても、それは児童の頭の中でコンピュータに意図した処理を行わせるイメージに繋がっていない観点において、プログラミング教育とは言えないということである。」と述べているように、単なるアルゴリズムの学習とプログラミン的思考(コンピュータに意図した処理を行うイメージ)の根本的な違いが現状では明確に論議・定義されていないようである。プログラミングの認知過程が明かになれば、この「頭の中でコンピュータに意図した処理を行わせるイメージ」が明かになり、プログラミング的思考とは具体的に何であるかを示すことができると考えている。

補足

2020 年度に使用した教材などについて下記のサイトにて公開している。

高校「情報科」の教材・指導案作ってみました。

Unit05 アルゴリズムとプログラム

<http://beyondbb.jp/>

参考文献

- [1] 文部科学省. 高等学校学習指導要領(平成 30 年告示), https://www.mext.go.jp/a_menu/shotou/new-cs/1384661.htm, (参照 2020-2-19).
- [2] 間辺広樹, 長島和平, 並木美太郎, 長慎也, 兼宗進. 高等学校における複数言語によるプログラミング教育の提案. 情報処理学会論文誌教育とコンピュータ(TCE), 2017, Vol.3, no.3, p.29-41.

- [3] 伊藤一成. ピクトグラミング—人型ピクトグラムを用いたプログラミング学習環境. 情報処理学会論文誌教育とコンピュータ (TCE), 2018, Vol.4, no.2, p.47-61.
- [4] 間辺広樹, 神藤健朗, 並木美太郎, 兼宗進. コンピュータ・アルゴリズムの「発見・記述・伝達」を導く授業の実践と評価. 情報処理学会論文誌教育とコンピュータ (TCE), 2016, Vol.2, no.1, p.10-24.
- [5] 兼宗進, 白井詩沙香, 竹中一平, 長瀧寛之, 小林史弥, 島袋舞子, 田邊則彦. データベースと情報システムを学習する授業の提案と実践. 情報処理学会論文誌教育とコンピュータ (TCE), 2017, Vol.3, no.3, p.18-28.
- [6] 高岡詠子, 山内崇裕, 滑川敬. 章高等学校における実用的プログラミングの教育実践. 情報処理学会論文誌教育とコンピュータ (TCE), 2016, Vol.2, no.2, p.37-52.
- [7] 富永浩之, 中井智己, 辻健人, 劉世博, 花川直己. 高大連携の導入講座としての LEGO プログラミング演習の実践. 情報処理学会論文誌教育とコンピュータ (TCE), 2018, Vol.4, no.2, p.1-13.
- [8] 文部科学省. プログラミング教育実践ガイド, https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1408013.htm, (参照 2020-2-19).
- [9] 中西渉. 高校におけるプログラミング教育. 情報処理, 2016, Vol.57, no.4, p.358-361.
- [10] 伊藤良二, 小西達裕, 伊東幸宏. プログラムの問題領域上での動作説明を行うプログラミング学習支援システムの構築. 人工知能学会誌, 2000, Vol.15, no.2, p.362-375.
- [11] 江木鶴子, 竹内章. プログラミング初心者にトレースを指導するデバッグ支援システムの開発と評価. 日本教育工学会論文誌, 2009, Vol.32, no.4, p.369-381.
- [12] 大城正典, 永井保夫. 初学者向けプログラミング学習のための初等アルゴリズム視覚化システム. 「情報教育シンポジウム (SSS2018)」 予稿集, 2018, p.104-111.
- [13] 掛下哲郎, 柳田峻, 太田康介. 穴埋め問題を用いたプログラミング教育支援ツール pgtracer の開発と評価. 情報処理学会論文誌教育とコンピュータ (TCE), 2016, Vol.2, no.2, p.20-36.
- [14] Saeed D., Richard B.. The camel has two humps, 2006, Middlesex University, Working Paper.
- [15] 三宅なほみ. コンピュータを教える ; 岩波講座 教育の方法教育と機械, 1987, 岩波書店, p120-159
- [16] 太田剛, 加藤浩, 森本容介. コンピュータショナル・シンキング概念に基づくプログラム自動評価機能を持つ Scratch 用学習支援システムの開発, 2018, 教育システム情報学会誌, 35(2), p.204-214.
- [17] 遠藤諭. プログラミング的思考がわからない, 情報処理, 2021, Vol.52, No.2, p.130-133.
- [18] 古池 謙人, 東本 崇仁, 堀口 知也, 平嶋 宗. プログラミングの構造的理解を指向した部品の段階的拡張手法の提案と支援システムの開発・評価, 教育システム情報学会誌, 2019, Vol. 36, No. 3, p.190-202.
- [19] 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, p.49-53.
- [20] つくば市教育局総合教育研究所, 赤堀 侃司, 久保田 善彦. これならできる小学校教科でのプログラミング教育, 2018, 東京書籍