

高校生の簡単な数値計算プログラミングにおける フローチャート作成時の思考過程の構造的把握

Structural Analysis on Thinking Processes of Making a Flowchart in an Easy Numerical Calculation Programming in Senior High School Students

森山 潤* 前田 康則** 福井 昌則*** 黒田 昌克****
MORIYAMA Jun MAEDA Yasunori FUKUI Masanori KURODA Masakatsu

中尾 尊洋**** 小倉 光明**** 山下 義史*****
NAKAO Takahiro OGURA Mitsuaki YAMASHITA Yoshifumi

本研究の目的は、プログラミング初学者の高校生を対象に、簡単な数値計算プログラミングの情報処理手順を構想するフローチャート作成時の思考過程を構造的に把握し、今後のプログラミング教育における授業改善に向けた示唆を得ることである。A県内の公立専門高校3年生116名を対象に、1からnまでの整数和を求めるプログラム、1カ月間の株価データを読み込み、最も高い株価を表示させるプログラムのフローチャートを作成させ、事後に質問紙調査を実施した(有効回答率82.7%)。その結果、数値計算課題のフローチャート作成時の思考過程として「エラーの探索と修正」因子、「大局的プランニング」因子、「部分的プランニング」因子、「変数内の値のチェック」因子、「処理構造のチェック」因子の5因子が抽出された。また、これら5因子のうち、フローチャート作成の課題達成には、「大局的プランニング過程」が重要な役割を果たしていることが示された。「大局的プランニング過程」は、コンピューショナル・シンキングの概念において抽象化、分解、アルゴリズム的思考と関連するものと考えられ、これらの思考を促す授業改善が重要であることを指摘した。

キーワード：プログラミング的思考、コンピューショナル・シンキング、フローチャート、数値計算課題、プログラミング教育、高校生

Key words : programming thinking, computational thinking, flow chart, numerical calculation task, programming education, senior high school students

1. はじめに

本研究の目的は、プログラミング初学者の高校生を対象に、簡単な数値計算課題の情報処理手順を構想するフローチャート作成時の思考過程を構造的に把握し、今後のプログラミング教育における授業改善に向けた示唆を得ることである。

我が国では、小学校段階におけるプログラミング教育(以下、小学校プログラミング教育)が必修化される¹⁾等、体系的な情報教育の充実を図られている。中学校技術・家庭科技術分野(以下、技術科)においてもプログラミングに関する学習内容が拡充され²⁾、高等学校でも共通教科情報科(以下、情報科)において、情報I、情報IIが設置され、プログラミングに関する学習内容が大幅

に拡充された³⁾。

小学校段階におけるプログラミング教育では、コーディングを必ずしも必要としないものの、プログラミングの思考を育成すること、そしてそれらの内容を各教科内で実施することが求められている。プログラミング的思考は、「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」と定義されている。これは、コンピューショナル・シンキング(以下、CT)の概念を踏まえつつ、プログラミングと論理的思考との関係を整理

*兵庫教育大学大学院人間発達教育専攻生活・健康・情報系教育コース 教授

令和2年10月21日受理

**神戸大学大学院経済学研究科

***兵庫教育大学教員養成・研修高度化センター 特命助教

****兵庫教育大学大学院連合学校教育学研究科(博士課程)教科教育実践学専攻生活・健康系教育連合講座

*****兵庫教育大学大学院学校教育研究科人間発達教育専攻(修士課程)生活・健康・情報系教育コース

しながら提言されたものである⁴⁾。CTは、我が国で言うところの情報教育に相当する Computer Science 教育において、国際的に重要な概念として認知されており⁵⁾⁶⁾、小学校だけでなく中学校や高校においても CT の概念を意識する必要があると考えられる。

Wing (2006) によれば、CT 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」と定義されている⁷⁾。また、ISTE & CSTA (2011) は、CT を問題解決のプロセスと捉えた上で、「コンピュータや他のツールを使って問題を解決できるように問題を策定すること」、「データを論理的にまとめ、分析すること」、「モデル化やシミュレーションなどの抽象化によるデータ表現」、「アルゴリズム的思考による解決の自動化（一連の順序づけられたステップ）」、「ステップとリソースの最も効果的で効果的な組み合わせを達成するという目標を持った、可能なソリューションの特定・分析・実装」、「問題解決過程の一般化と広い範囲にわたる問題への移行」といった特徴を持つ概念としている⁵⁾。また、BBC Bitesize (2020) は CT を構成する概念として、論理、アルゴリズム的思考、分解、パターン化、抽象化、評価の6つの要素があるとしている⁸⁾。森山ら (2020) は、小学校段階でプログラミング的思考が育成されていることを前提とし、中学校以降はコーディングの活用も視野に入れ、さらに実践的で高度な内容を学習することによって、複雑な状況に対応させることや創造的で実践的に問題解決させること、プログラミング的思考に留まらず CT を育成することが重要であると指摘している⁹⁾。

このようなプログラミング的思考や CT を体系的なプログラミング教育において育成していくためには、児童生徒の発達段階に応じた学習支援が重要となる。その際、小学校、中学校、高等学校という学校段階の進行に伴うプログラミング課題の変化を踏まえる必要がある。小学校では、プログラミングに慣れ親しむことが主眼となる。そのため、児童が取り組む課題の多くは、教育用のブロックプログラミング環境を使用し、単純な命令を順次処理で組み合わせるものが主体となる。キーボードからの入力を受け付けたり、ロボット系教材でセンサーを活用したりする場合などでは、反復処理や条件分岐処理が用いられることもあるが、数値を計算対象とするケースはほとんどない。中学校になると技術科では、ネットワークを利用した双方向性のあるコンテンツのプログラミング、計測・制御システムのプログラミングが取り上げられる。ここでは、教育用のブロックプログラミング環境に加えて、日本語のテキスト型プログラミング環境や、社会で使用されている本格的なプログラミング環境を利用することも想定される。また、生徒は、小学校で慣れ親しんだ3つの処理構造を概念として正しく理解した上で、自らの問題解決にこれらを活用したよ

り複雑なプログラミングに取り組むことになる。その際、プログラミングの中では、インクリメントなどの処理回数の演算やセンサーの閾値の扱い、配列変数などで数値を扱うケースが生じる。高等学校になると社会で使用される本格的なプログラミング言語の使用が前提となる。また、情報 I では、ソートのアルゴリズムやシミュレーションなど、より複雑に数値を扱う課題が取り扱われるようになる。情報 II ではさらに、Python や R などを用いて、データサイエンスと関連づけたプログラミングへと発展していくことが標榜されている。

このように、学校段階の進行に応じたプログラミング課題の変化には、プログラミング環境の変化など、様々な要素の高度化があるが、その一つとして数値に対する処理の重要性が段階的に増していく点が挙げられる。特に中学校から高等学校にかけては、プログラミングの中で、変数に格納された値に対する数的な演算処理を適切に行えるかどうか、課題の達成を左右する重要な要因となっていく。数的な演算処理では、プログラミング言語の文法理解やコーディングスキルだけでなく、変数に対する数的な情報処理の手順を正しく構想する力が求められる。今後、中学校や高等学校におけるプログラミング教育を推進していくためには、生徒の数的な演算処理に関わる力を高めたり、つまづきを回避したりする適切な学習指導方法の構築が重要になることが予測される。

プログラミングにおける学習者の思考過程については、これまでもいくつかの先行研究が見られる。宮川ら (2006) は、オブジェクト指向イベントドリブン型のプログラミング問題解決過程の構成因子について検討し、「オブジェクト機能化」、「エラー修正」、「論理エラー検索」、「構文エラー検索」、「動作チェック」、「知識要求」の6因子からなることを報告している¹⁰⁾。また、黒上・堀田 (2017) は、CT の構成概念である分解やアルゴリズム的思考、一般化、抽象化に着目しながら問題解決能力を育成するためには、適切な部品に分解し、物事を筋道立てて考え、説明することなどが重要となると述べている¹¹⁾。しかし、これらの先行研究を含めて、数的な演算処理に着目したプログラミングの思考過程については十分な検討がなされていないのが現状である。

数的な演算処理を取り扱うプログラミング課題として、数値計算課題がある。数値計算課題は、データの統計量を計算したり、数学的な変換処理を行ったりするものである。生徒の持つ数的な演算処理に関わる力を把握したり、高めたりする場合には、数値計算課題を用いたプログラミングを取り上げることが考えられる。筆者らは前報⁹⁾にて、初学者の変数操作に関するプログラムのフローチャート作成時における情報処理手順に関する思考過程についてプロトコル分析を実施し、問

題解決の過程を構成する計 26 カテゴリを抽出した。その上で、これらのカテゴリが「P 表象・プラン」, 「D 実行」, 「C 確かめ」, 「R 修正」の 4 過程, 問題解決の操作子は「A アルゴリズムの構成」, 「B 変数操作」の 2 操作子に分類できたことを報告している⁹⁾。

そこで本研究では、前報で抽出した計 26 カテゴリを援用し、高校生を対象に、簡単な数値計算プログラミングにおける情報処理手順を構想する際の思考過程を明らかにすることとした。具体的には、平均や最大値など、簡単なデータの統計量を求める数値計算課題を設定し、高校生にフローチャートを作成させる。その際に思考したことを前報で抽出した 26 カテゴリを用いて把握し、因子分析を用いてその構造を探索する。その上で、課題の達成状況に影響する思考過程を同定することを通して、今後のプログラミング教育における学習支援の方策について考察することとした。

なお、ここでフローチャート作成を課題としたのは、前述したように、数的な演算処理では変数に対する数的な情報処理の手順を正しく構想する力が重要であるとの認識から、プログラミング言語の文法理解やコーディングスキルの影響を排除して生徒の思考過程を把握するためである。

2. 研究の方法

2.1 被験者

被験者（以下、生徒）は、A 県内の公立専門高校の情報科および機械科の高校生 3 年生 116 名（男子 96 名、女子 20 名）とした。うち、有効回答は 96 名（男子 76 名、女子 20 名）であり、有効回答率は 82.7% であった。生徒は、基礎的な情報処理の授業を履修しており、フローチャートの図記号や順次・反復・条件処理などの基本的な処理について知識を有しているものの、具体的なプログラミングを伴う情報実習は未履修であった。

2.2 実験課題および質問項目

2.2.1 実験課題

実験課題は、順次・反復・条件処理などの基本的な処理を用いて具体的にプログラミングする力を把握する内容として、指定した処理のフローチャートを完成させる課題 1, 課題 2 の 2 つの課題を設定した。課題 1 は、1 から n までの整数和を求めるプログラムについて、空欄を含むフローチャートを示し、指定された処理条件を満たす動作を解答群から選択させる 3 つの問題とした（図 1）。課題 2 は、企業の 1 か月間の株価データを読み込み、最も高い株価を表示させるプログラムのフローチャートを作成させる問題とした（図 2）。

2.2.2 質問項目

質問項目は、フローチャート作成に関する経験や意識

課題 1 から n までの数を入力し、処理条件に従って図 1 のように表示する流れ図を考えなさい。

処理条件

- (1) 整数 n をテキストボックスに入力し、計算ボタンを押すと、以下の処理が実行される。
- (2) 1 ~ n までの整数の和を合計して Label3 に表示する。
- (3) (2) の処理を行うフローチャートを考えて、あてはまる言葉を下の解答群から記号なさい。

解答群

ア 0→乗計 イ 0→和計
 ウ 和計+n→和計
 エ $n \times 2$ →答え
 オ 乗計+n→乗計
 カ 和計+k→和計
 キ 2のk乗→乗数
 ク 乗計+乗数→乗計
 ケ kは1から1ずつ増やしてnまで
 コ mは1から1ずつ増やしてkまで

図 1 課題 1

課題 2 ある企業の 1 か月間の株価データを読み、最も高い株価、最も安い株価を求めて、図 1 のような実行形式のラベルに表示する流れ図を考えなさい。

処理条件

- (1) 計算ボタンをクリックすると以下の処理が実行される。
- (2) 最も高い株価を求め、Label2に表示される。
- (3) 最も安い株価を求め、Label3に表示される。

(1) ~ (3) の処理を行うフローチャートを考えなさい。
(計算ボタンを押す処理は省く)

図 2 課題 2

を問う項目として 3 項目、実験課題に対する難易度意識を把握する項目として 1 項目、フローチャート作成時の思考過程を把握する項目として 26 項目を作成した。

フローチャート作成に関する経験や意識を問う項目は 4 件法にて実施し、フローチャートの作成経験 (4. よく書いたことがある, 3. 少し書いたことがある, 2. あまり書いたことがない, 1. 全然書いたことがない)、フローチャート作成に対する好嫌意識 (4. 大好き, 3. 少し好き, 2. 少し嫌い, 1. 大変嫌い)、フローチャート作成に対する得意・不得意 (4. 大変得意, 3. 少し得意, 2. 少し不得意, 1. 大変不得意) に関して回答を求めた。

実験課題に対する難易度意識を把握する項目も、4 件法 (4. 大変難しかった, 3. 少し難しかった, 2. 少し易しかった, 1. 大変易しかった) にて実施し、取り組んだ実験課題に対して感じた難しさについて回答を求めた。

フローチャート作成時の思考過程を把握する項目は、前報のプロトコル分析より得られたカテゴリ⁹⁾に基づき、高校において情報教育の経験を持つ熟練教員 2 名で協議してワーディングした。質問項目を表 1 に示す。

これらの各項目について、4件法（4.とても考えた、3.少し考えた、2.あまり考えなかった、1.まったく考えなかった）にて実施し、実験課題の実施時における思考を振り返らせて回答を求めた。

表1 実験課題解決時の思考過程を把握する質問項目

1	課題の条件から、全体の情報処理の手順を大まかに構想し、どのような処理構造(反復や条件分岐など)が必要になるか、考えた。
2	課題の条件から、どのような変数の設定が必要か、考えた。
3	フローチャートの中で、どのような処理のまとまりを反復させればよいか、考えた。
4	フローチャートの中で、どのような条件で処理の流れを分析すればよいか、考えた。
5	フローチャートの中で、どのような値を入力変数にすればよいか考えた。
6	フローチャートの中で、入・出力変数以外に、どのような変数が必要か考えた。
7	チャートの中で、どのような対応を出力変数にすればよいか考えた。
8	フローチャートの中に、必要な反復処理の位置と内容を決めて、設定した。
9	フローチャートの中に、必要な条件分岐処理の位置と内容を決めて、設定した。
10	フローチャートの中に、必要な入力変数を決めて、設定した。
11	フローチャートの中で、入・出力変数以外に必要な変数を決めて、設定した。
12	フローチャートの中に、必要な出力変数を決めて、設定した。
13	課題の条件から、作成したフローチャート全体の処理の流れを追って確認した。
14	フローチャートの中で、反復が正しく設定できているかどうか、処理の流れを確認した。
15	チャートの中で、条件分岐が正しく設定されているかどうか、処理の流れを確認した。
16	作成したフローチャートの処理手順によって、各変数の中に課題の条件を満たす正しい値が適切に格納されているか、確認した。
17	作成したフローチャートの中で、入力変数に正しい値が格納されているかどうか確認した。
18	作成したフローチャートの中で、入・出力変数以外の変数に正しい値が格納されているかどうか確認した。
19	作成したフローチャートの中で、出力変数に正しい値が格納されているかどうか確認した。
20	作成したフローチャートは課題の条件を満たしていないので、間違いを探した。
21	フローチャートの中で、反復処理の仕方に関連を見つけ、修正した。
22	フローチャートの中で、条件分岐処理の仕方に関連を見つけ、修正した。
23	作成したフローチャートの中で、入力変数の設定の仕方が間違っていたので、修正した。
24	作成したフローチャートの中で、入・出力変数以外の変数の設定の仕方が間違っていたので、修正した。
25	作成したフローチャートの見直しによって見つけた間違いを、課題の条件を満たすように修正した。
26	課題の条件を満たすように、作成したフローチャートの問題点を改善したり、より良くするために、異なる解決方法をいろいろ考え、試してみた。

2.3 手続き

まず、調査対象校の生徒に、授業時間内（情報関連科目）において、2つの実験課題に取り組みさせた。所要時間は15分とした。その後、作成した質問紙による調査を実施した。

実験課題の集計は、生徒の解答に対して正誤の判定をし、得点化した。課題1は3つの問題それぞれを採点し、正答数を求めた。課題2は、記入されたフローチャートの処理構造が課題の条件を満たしている場合に正答、それ以外は誤答とした。この際、フローチャートの図記号や文字の書き方や綺麗さなどは正答の条件としなかった。

質問紙調査の回答は、回答に合わせて1～4点に順次得点化し集計した。フローチャート作成に関する経験や意識を問う項目、実験課題に対する難易度意識を把握する項目に関しては、各回答の頻度を算出し、フローチャートの作成経験や意識、実験課題に対する難易度の意識について傾向を確認した。

フローチャート作成時の思考過程を把握する項目に関しては、主因子法及びプロマックス回転を用いた因子分析を行った。因子分析では、初期解で得られた固有値のスクリープロットより、1.0以上を対象に因子軸の回転を施した（カイザー・ガットマン基準）。回転後に、各項目の因子負荷量の絶対値0.35以上を基準に同一因子とみなし、解釈・命名した。次に、得られたフロー

チャート作成時の思考過程を構成する因子別に尺度の平均値を求め、課題1、課題2の正答・誤答の群間で、*t*検定にて課題における得点差を比較した。

3. 結果及び考察

3.1 生徒の実態

フローチャート作成に関する経験や意識を問う項目の集計結果を表2、フローチャート作成に対する好嫌意識を問う項目の集計結果を表3、フローチャート作成に対する得意・不得意を問う項目の集計結果を表4、実験課題に対する難易度意識の集計結果を表5にそれぞれ示す。

これらの結果から少なくともフローチャート作成の経験のある者が約2割程度に留まり、本実験に参加した生徒は、大半はフローチャート作成に対する興味・関心がなく、不得意意識を有する生徒である実態が把握された。また、被験者は実験課題に対して、やや難易度が高いと感じていることも把握された。実験計画の段階では、高校生の大部分は中学校段階でなんらかのフローチャート作成を経験していることを想定していたが、中学校段階でのフローチャート作成に関する学習が適切に実施されていない、あるいは実施されているが生徒の経験として認知されていないと推察された。上記の実態を持つ生徒の反応ということを留意した上で以下の分析を進めることとする。

表2 フローチャートの作成経験

	人数	割合 (%)
4. よく書いたことがある	3	3.1
3. 少し書いたことがある	12	12.5
2. あまり書いたことがない	37	38.5
1. 全然書いたことがない	44	45.8

n=96

表3 フローチャート作成に対する好嫌意識

	人数	割合 (%)
4. 大変好き	1	1.0
3. 少し好き	30	31.2
2. 少し嫌い	46	47.9
1. 大変嫌い	19	19.8

n=96

表4 フローチャート作成に対する得意・不得意

	人数	割合 (%)
4. 大変得意	1	1.0
3. 少し得意	9	9.4
2. 少し不得意	45	46.9
1. 大変不得意	41	42.7

n=96

表6 因子負荷量

(主因子法・プロマックス回転)

	因子1	因子2	因子3	因子4	因子5
22フローチャートの中で、条件分岐処理の仕方に間違いを見つけ、修正した	0.99	-0.10	-0.07	-0.07	-0.07
24作成したフローチャートの中で、入・出力変数以外の変数の設定の仕方が間違っていたので、修正した	0.93	-0.21	0.04	0.07	0.06
25作成したフローチャートの見直しによって見つけた間違いを、課題の条件を満たすように修正した	0.90	0.70	-0.19	0.05	0.70
21フローチャートの中で、反復処理の仕方に間違いを見つけ、修正した	0.89	-0.10	0.15	-0.14	0.13
23作成したフローチャートの中で、入力変数の設定の仕方が間違っていたので、修正した	0.48	-0.07	-0.04	0.07	0.09
26課題の条件を満たすように、作成したフローチャートの問題点を改善したり、より良くするために、異なる解決方法をいろいろ考え、試してみた	0.63	0.36	-0.20	0.06	-0.19
20作成したフローチャートは課題の条件を満たしていないので、間違いを探した	0.53	0.13	0.16	-0.07	0.31
18作成したフローチャートの中で、入・出力変数以外の変数に正しい値が格納されているかどうか確認した	0.50	-0.03	-0.11	0.55	0.10
1多大な条件から、全体の情報処理の手順を大まかに構想し、どのような処理構造(反復や条件分岐など)が必要になるか、考えた	-0.21	0.91	-0.14	0.02	0.21
4フローチャートの中で、どのような条件で処理の流れを分析すれば良いか、考えた	0.14	0.89	-0.04	-0.31	0.20
2課題の条件から、どのような変数の設定が必要か、考えた	-0.07	0.87	-0.18	0.23	-0.20
5フローチャートの中で、どのような値を入力変数にすれば良いか考えた	-0.15	0.79	0.11	0.03	0.12
3フローチャートの中で、どのような処理のまとまりを反復させればよいか、考えた	-0.14	0.72	0.23	0.00	0.09
6フローチャートの中で、入・出力変数以外に、どのような変数が必要か考えた	0.24	0.67	0.10	-0.08	-0.21
7チャートの中で、どのような対応を出力変数にすれば良いか考えた	0.03	0.54	0.23	0.04	-0.17
9フローチャートの中に、必要な条件分岐処理の位置と内容を決めて、設定した	-0.01	-0.03	0.99	-0.06	0.05
8フローチャートの中に、必要な反復処理の位置と内容を決めて、設定した	-0.04	0.00	0.89	-0.20	0.22
10フローチャートの中に、必要な入力変数を決めて、設定した	-0.11	0.08	0.75	0.18	-0.09
12フローチャートの中に、必要な出力変数を決めて、設定した	0.07	-0.14	0.63	0.28	-0.02
11フローチャートの中で、入・出力変数以外に必要な変数を決めて、設定した	0.22	0.12	0.52	0.14	-0.31
17作成したフローチャートの中で、入力変数に正しい値が格納されているかどうか確認した	0.14	-0.03	-0.05	0.80	0.06
16作成したフローチャートの処理手順によって、各変数の中に課題の条件を満たす正しい値が適切に格納されているか、確認した	0.22	-0.02	-0.08	0.75	0.04
19作成したフローチャートの中で、出力変数に正しい値が格納されているかどうか確認した	0.13	-0.06	0.10	0.66	0.16
15チャートの中で、条件分岐が正しく設定されているかどうか、処理の流れを確認した	-0.17	0.10	0.27	0.60	0.15
13課題の条件から、作成したフローチャート全体の処理の流れを追って確認した	0.15	0.19	0.02	0.20	0.68
14フローチャートの中で、反復が正しく設定できているかどうか、処理の流れを確認した	0.29	0.14	0.07	0.16	0.47

回転後の因子の相関行列

	因子1	因子2	因子3	因子4
因子2	0.51			
因子3	0.58	0.65		
因子4	0.73	0.52	0.61	
因子5	0.19	0.07	0.08	0.03

群それぞれに対し、抽出された5因子について回答を得点化した平均値を算出し、*t*検定によりその得点差を比較した。課題1の結果を表9に、課題2における結果を表10にそれぞれ示す。

課題1では、1問以上正答している生徒が62名(64.6%)、全ての問題で誤答だった生徒が34名(35.4%)の間で、「大局的プランニング」($t_{(94)}=2.16, p<.05$)、「部分的プランニング」($t_{(94)}=1.78, p<.10$)の平均値にそれぞれ有意な差が認められた。課題2では、正答者10名(10.4%)と誤答者86名(89.6%)の間で、F2「大局的プランニング」の平均値に有意な差が認められた($t_{(94)}=3.01, p<.01$)。

3.5 考察

以上の結果から、本実験の条件下で生徒が数値計算を主とするフローチャートを作成する際には、「エラーの探索と修正」、「大局的プランニング」、「部分的プランニング」、「変数内の値のチェック」、「処理構造のチェック」といった思考過程の構造が生起していたことが示唆された。また、課題1、課題2の両方において、正答できた生徒の「大局的プランニング」因子の平均値が、誤答だった生徒より有意に高く、フローチャート作成の課題達成には、情報処理の手順を構想する大局的プランニングの過程が重要な役割を果たしていることが示

唆された。

これらのことから、プログラミングに未習熟の生徒に、数的な演算処理を含む情報処理手順の構想力を求める授業を実施する場合は、生徒の大局的なプランニングの過程を活性化させる手立てが重要ではないかと考えられる。大局的なプランニングの過程は、課題の意味的な理解を情報処理手順に関する手続き的な理解へと変換する過程である。これは、CTの抽象化(課題の意味をモデル化して捉える)や分解(課題をさらに下位の課題に分解する)、アルゴリズム的思考(論理的に手順を組み立てる)と関連するものと考えられる。言い換えれば、CTの抽象化、分解、アルゴリズム的思考などの要素は、それぞれを個別に育成するのではなく、これらを情報処理手順構想時の大局的なプランニングの過程に位置づけて、相互に関連づけながら育成することの重要性が指摘できる。

このような観点から、大局的なプランニングの過程を活性化させる手立ての具体例を挙げると、フローチャート作成の前段階で問題をより小さく単純化された問題に分解させる、課題の解決に必要な処理構造や変数を書き出させ互いの関連性を図に表す、フローチャートの形式に拘らず図や文章で大まかな処理の流れを表現させる等が考えられる。これらの手立てを意図的にフローチャート作成に組み込むことで、抽象化、分解、アルゴ

リズム的思考を伴って大局的なプランニングが活性化され、数的な演算処理に関わる情報処理手順の構想力を高めることができるのではないかと期待される。また、特定の問題で作成したフローチャートを別の問題にどのように適用するかについて検討させることも、大局的なプランニング過程を活性化できる可能性があるのではないかと考えられる。

しかし、本研究では、平均や最大値を求めるといった簡単な数値計算を課題としたため、これらの手立ての汎用性については、今後の詳細な検討が必要である。また、実際に問題解決で取り扱われる課題は、本研究で使った数値計算課題のように数的な演算処理が顕在化していないケースが考えられる。例えば、「壁に当たると跳ね返る」のような課題を変数とその値の変化や比較として捉えなおす過程（モデル化）に、生徒のつまづきが生じる可能性も高い。本研究においても課題1に対して課題2の達成状況が芳しくなかったことは、このような課題の持つ文脈の効果による可能性もある。これについては、問題のモデル化に着目した新たな検討が必要である。これらの成果を相互に結び付けて、小学校から中学校、高等学校へと至る体系的なプログラミング教育において、課題の高度化にスムーズに対応しうる適切な学習支援の方策を構築していく必要がある。

表9 課題1の達成状況と尺度平均値の関連性

因子	1つ以上正解 (n=62)		すべて誤答 (n=34)		t検定
	平均	SD	平均	SD	
F1: エラーの探索と修正	1.94	0.59	1.93	0.69	$t_{(94)}=0.074$ n. s.
F2: 大局的プランニング	2.52	0.63	2.22	0.67	$t_{(94)}=2.159$ *
F3: 部分的プランニング	2.22	0.59	1.99	0.62	$t_{(94)}=1.775$ †
F4: 変数内の値のチェック	2.00	0.62	2.04	0.67	$t_{(94)}=0.291$ n. s.
F5: 処理構造のチェック	2.26	0.79	2.15	0.67	$t_{(94)}=0.680$ n. s.

*: $p < .05$, †: $p < .10$, n. s.: 非有意

表10 課題2の達成状況と尺度平均値の関連性

因子	正答 (n=10)		誤答 (n=86)		t検定
	平均	SD	平均	SD	
F1: エラーの探索と修正	1.85	0.41	1.95	0.65	$t_{(94)}=0.471$ n. s.
F2: 大局的プランニング	2.99	0.54	2.35	0.64	$t_{(94)}=3.007$ **
F3: 部分的プランニング	2.22	0.56	2.13	0.61	$t_{(94)}=0.441$ n. s.
F4: 変数内の値のチェック	1.78	0.53	2.04	0.64	$t_{(94)}=1.223$ n. s.
F5: 処理構造のチェック	2.45	0.83	2.19	0.74	$t_{(94)}=1.027$ n. s.

** : $p < .01$, n. s.: 非有意

4. まとめと今後の課題

以上、本研究では、プログラミング初学者の高校を対象に、簡単な数値計算課題の情報処理手順を構想するフローチャート作成時の思考過程を構造的に把握した。その結果、本研究の条件下で以下の知見が得られた。

1. 高校生が数値計算を主とするフローチャート作成に取り組む問題解決では、F1「エラーの探索と修正」因子、F2「大局的プランニング」因子、F3「部分的プランニング」因子、F4「変数内の値のチェック」因子、F5「処理構造のチェック」因子の5因子からなる思考過程が生じていることが示唆された。

2. これらの5因子と課題の達成度との関連性を検討した結果、正答・誤答群間でF2「大局的プランニング」因子の平均値に有意な差が認められたことから、フローチャート作成の課題達成には、情報処理の手順を構想する大局的なプランニングの過程が重要な役割を果たしていることが示唆された。
3. これらの結果から今後のプログラミング教育では、数的な演算処理を含む情報処理手順の大局的なプランニングの過程がCTの抽象化、分解、アルゴリズム的思考と関連するものである点に着目した授業改善が重要であることを指摘した。

今後は、本研究で得られた5つの因子とCTの諸概念とを対応づけ、中学校技術科や高校情報科の教科書などに掲載されている課題における生徒の思考過程を評価しうる基準を作成する必要がある。そのためには、本研究で抽出した5因子とCTの諸概念間との関連性を確証的因子分析によって検証する必要があると考えられる。その上で、本研究で提案した学習支援の方策について、その効果を実践的に検証する必要がある。また、同様の問題意識から、例えば、プログラムの構造化など、数的な演算処理以外の課題要因について取り上げ、必要な学習支援の方策について検討を進めていく必要がある。本研究で得られた知見に対する追試を含めて、これらについては今後の課題とする。

文献

- 1) 文部科学省：小学校学習指導要領，東洋館出版社，(2018)
- 2) 文部科学省：中学校学習指導要領解説技術・家庭編，開隆堂出版，(2018)
- 3) 文部科学省：中学校学習指導要領解説情報編，開隆堂出版，(2019)
- 4) 文部科学省 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：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（議論の取りまとめ），(2016) https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm (2020.10.11 アクセス確認)
- 5) The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) :Operational Definition of Computational Thinking, (2011) <https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf> (2020.10.11 アクセス確認)
- 6) National Curriculum in England: Computing Programmes of Study, (2013)

<https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study> (2020.10.11 アクセス確認)

- 7) J.M. Wing: Computational Thinking, Communications of the ACM, 49 (3), pp.33-35, (2006) (日本語訳, 中島秀之: 計算論的思考, 情報処理, 56 (6), pp.584-587, (2015))
- 8) BBC Bitesize: Introduction to Computational Thinking, (2020) <https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1> (2020.10.11 アクセス確認)
- 9) 森山潤, 前田康則, 福井昌則, 小倉光明, 中尾尊洋, 近澤優子, 山下義史, 黒田昌克: 初学者のプログラミングにおける変数操作に関する情報処理手順を構想する思考過程の分析～中学校技術科内容「D 情報の技術」におけるプログラミング教育の改善に向けて～, 兵庫教育大学学校教育学研究, 33, pp.1-8, (2020)
- 10) 宮川洋一, 森山潤, 松浦正史: オブジェクト指向イベントドリブン型のプログラミングにおける問題解決過程の構造分析 - プログラム作成能力との関連に焦点を当てて -, 教育情報研究, 22 (2), pp.3-11, (2006)
- 11) 黒上晴夫, 堀田龍也: プログラミング教育導入前に知っておきたい思考のアイデア, 小学館, (2017)